

Bayes Decision Rules and Confidence Measures for Statistical Machine Translation

Nicola Ueffing, Hermann Ney

Lehrstuhl für Informatik VI - Computer Science Department
RWTH Aachen University
Ahornstrasse 55
52056 Aachen, Germany
{ueffing,ney}@cs.rwth-aachen.de

Abstract. In this paper, we re-visit the foundations of the statistical approach to machine translation and study two forms of the Bayes decision rule: the common rule for minimizing the number of string errors and a novel rule for minimizing the number of symbol errors. The Bayes decision rule for minimizing the number of string errors is widely used, but its justification is rarely questioned.

We study the relationship between the Bayes decision rule, the underlying error measure, and word confidence measures for machine translation. The derived confidence measures are tested on the output of a state-of-the-art statistical machine translation system. Experimental comparison with existing confidence measures is presented on a translation task consisting of technical manuals.

1 Introduction

The statistical approach to machine translation (MT) has found widespread use. There are three ingredients to any statistical approach to MT, namely the Bayes decision rule, the probability models (trigram language model, HMM, ...) and the training criterion (maximum likelihood, mutual information, ...).

The topic of this paper is to examine the differences between string error (or *sentence* error) and symbol error (or *word* error) and their implications for the Bayes decision rule. The error measure is referred to as loss function in statistical decision theory. We will present a closed representation of different word error measures for MT. For these different word error measures, we will derive the posterior risk. This will lead to the definition of several confidence measures at the word level for MT output.

Related Work: For the task of MT, statistical approaches were proposed at the beginning of the nineties [3] and found widespread use in the last years [12, 14]. To the best of our knowledge, the 'standard' version of the Bayes decision rule, which minimizes the number of sentence errors, is used in virtually all approaches to statistical machine translation (SMT). There are only a few research groups that do not take this type of decision rule for granted.

In [8], an approach to SMT was presented that minimized the posterior risk for different error measures. Rescoring was performed on 1,000-best lists produced by an SMT system. In [11], a sort of error related or discriminative training was used, but the decision rule as such was not affected. In other research areas, e.g. in speech recognition, there exist a few publications that consider the word error instead of the sentence error for taking decisions [6].

2 Bayes Decision Rule for Minimum Error Rate

2.1 The Bayes Posterior Risk

Knowing that any task in natural language processing (NLP) is a difficult one, we want to keep the number of wrong decisions as small as possible. To classify an observation vector y into one out of several classes c , we resort to the so-called statistical decision theory and try to minimize the posterior *risk* $R(c|y)$ in taking a decision. The posterior risk is defined as

$$R(c|y) = \sum_{\tilde{c}} Pr(\tilde{c}|y) \cdot L[c, \tilde{c}] ,$$

where $L[c, \tilde{c}]$ is the so-called *loss function* or *error measure*, i.e. the loss we incur in making decision \tilde{c} when the true class is c . The resulting decision rule is known as *Bayes decision rule* [4]:

$$y \rightarrow \hat{c} = \arg \min_c R(c|y) = \arg \min_c \left\{ \sum_{\tilde{c}} Pr(\tilde{c}|y) \cdot L[c, \tilde{c}] \right\} .$$

In the following, we will consider two specific forms of the error measure, $L[c, \tilde{c}]$. The first will be the measure for *sentence* errors, which is the typical loss function used in virtually all statistical approaches. The second is the measure for *word* errors, which is the more appropriate measure for machine translation and also speech recognition.

In NLP tasks such as Part-of-Speech tagging, where we do not have the alignment problem, the optimal decision is the following: compute the Bayes posterior probability and accept if the probability is greater or equal to 0.5. We omit the proof here. Following this, we formulate the Bayes decision rule for two different word error measures in MT. From those we can derive word confidence measures for MT according to which the words in MT output can be either accepted as correct translations or rejected.

2.2 Sentence Error

For machine translation, the starting point is the observed sequence of words $y = f_1^J = f_1 \dots f_J$, i.e. the sequence of words in the source language which has to be translated into a target language sequence $c = e_1^I = e_1 \dots e_I$.

The first error measure we consider is the sentence error: two target language

sentences are considered to be identical only when the words in each position are identical (which naturally requires the same length I). In this case, the error measure between two strings e_1^I and \tilde{e}_1^I is:

$$L[e_1^I, \tilde{e}_1^I] = 1 - \delta(I, \tilde{I}) \cdot \prod_{i=1}^I \delta(e_i, \tilde{e}_i) ,$$

with the Kronecker delta $\delta(.,.)$. In other words, the errors are counted at the *string* (or sentence) level and not at the level of single symbols (or words). Inserting this cost function into the Bayes risk (see Section 2.1), we obtain the following form of *Bayes decision rule for minimum sentence error*:

$$\begin{aligned} f_1^J \rightarrow (\hat{I}, \hat{e}_1^{\hat{I}}) &= \arg \max_{I, e_1^I} \{Pr(I, e_1^I | f_1^J)\} \\ &= \arg \max_{I, e_1^I} \{Pr(I, e_1^I, f_1^J)\} . \end{aligned} \quad (1)$$

This is the starting point for virtually all statistical approaches in machine translation. However, this decision rule is only optimal when we consider *sentence* error. In practice, however, the empirical errors are counted at the *word* level. This inconsistency of decision rule and error measure is rarely addressed in the literature.

2.3 Word Error

Instead of the *sentence* error rate, we can also consider the error rate of *symbols* or single *words*. In the MT research community, there exist several different error measures that are based on the word error. We will investigate the *word error rate (WER)* and the *position independent word error rate (PER)*.

The symbol sequences in Figure 1 illustrate the differences between the two error measures WER and PER: Comparing the strings 'ABCB D' and 'ABBCE', WER yields an error of 2, whereas the PER error is 1.

	WER:	PER:
string 1	A B C B D	A B C B D
	\	X
string 2	A B B C E	A B B C E

Fig. 1. Example of the two symbol error measures WER and PER: The string 'ABCB D' is compared to 'ABBCE'.

For NLP tasks where there is no variance in the string length (such as Part-of-Speech tagging), the integration of the symbol error measure into Bayes decision

rule yields that a maximization of the posterior probability for each position i has to be performed [10]. In machine translation, we need a method for accounting for differences in sentence length or word order between the two strings under consideration, e.g. the Levenshtein alignment (cf. WER).

- **WER** (word error rate): The word error rate is based on the Levenshtein distance [9]. It is computed as the minimum number of substitution, insertion, and deletion operations that have to be performed to convert the generated sentence into the reference sentence. For two sentences e_1^I and $\tilde{e}_1^{\tilde{I}}$, the Levenshtein alignment is denoted by $\mathcal{L}(e_1^I, \tilde{e}_1^{\tilde{I}})$; for a word e_i , the Levenshtein aligned word in $\tilde{e}_1^{\tilde{I}}$ is denoted by $\mathcal{L}_i(e_1^I, \tilde{e}_1^{\tilde{I}})$ for $i = 1, \dots, I$. In order to keep the presentation simple, we only consider substitutions and deletions of words in e_1^I and omit insertions in $\tilde{e}_1^{\tilde{I}}$. The error measure is defined by

$$L[e_1^I, \tilde{e}_1^{\tilde{I}}] = \sum_{i=1}^I \left[1 - \delta(e_i, \mathcal{L}_i(e_1^I, \tilde{e}_1^{\tilde{I}})) \right] .$$

This yields the posterior risk

$$\begin{aligned} R(I, e_1^I | f_1^J) &= \sum_{\tilde{I}, \tilde{e}_1^{\tilde{I}}} Pr(\tilde{I}, \tilde{e}_1^{\tilde{I}} | f_1^J) \cdot \sum_{i=1}^I \left[1 - \delta(e_i, \mathcal{L}_i(e_1^I, \tilde{e}_1^{\tilde{I}})) \right] \\ &= \sum_{i=1}^I \left(1 - \sum_{\tilde{I}, \tilde{e}_1^{\tilde{I}}: \mathcal{L}_i(e_1^I, \tilde{e}_1^{\tilde{I}}) = e_i} Pr(\tilde{I}, \tilde{e}_1^{\tilde{I}} | f_1^J) \right) . \end{aligned}$$

In Section 3.2 we will see that this is related to the word posterior probabilities introduced in [15]. The *Bayes decision rule for minimum WER* is obtained by minimizing the risk.

- **PER** (position independent word error rate): A shortcoming of the WER is the fact that it does not allow for movement of words or blocks. The word order of two target sentences can be different even though they are both correct translations. In order to overcome this problem, the position independent word error rate compares the words in the two sentences *without* taking the word order into account. Words that have no matching counterparts are counted as substitution errors, missing words are deletion and additional words are insertion errors. The PER is always lower than or equal to the WER.

To obtain a closed-form solution of the PER, we consider for each word $e = 1 \dots E$ in the target vocabulary the number n_e of occurrences in sentence e_1^I , i.e. $n_e = \sum_{i=1}^I \delta(e_i, e)$. The number of occurrences of word e in sentence $\tilde{e}_1^{\tilde{I}}$ is denoted by \tilde{n}_e , respectively. The error can then be expressed as

$$L[e_1^I, \tilde{e}_1^{\tilde{I}}] = \max(I, \tilde{I}) - \sum_e \min(n_e, \tilde{n}_e) .$$

Thus, the error measure depends only on the two sets of counts $n_1^E := n_1 \dots n_e \dots n_E$ and $\tilde{n}_1^E := \tilde{n}_1 \dots \tilde{n}_e \dots \tilde{n}_E$. The integration of this error measure into the posterior risk yields [16]

$$R(n_1^E | f_1^J) = \frac{1}{2} \sum_e \sum_{\tilde{n}_e} |n_e - \tilde{n}_e| \cdot Pr_e(\tilde{n}_e | f_1^J) + \frac{1}{2} \sum_{\tilde{I}} |I - \tilde{I}| \cdot Pr(\tilde{I} | f_1^J) \quad (2)$$

where $Pr_e(n_e | f_1^J)$ is the posterior probability of the count n_e of word e .

3 Confidence Measures for Machine Translation

3.1 Introduction

In many applications of machine translation, a method for labeling the generated words as either correct or incorrect is needed. To this purpose, each word in the generated target sentence is assigned a so-called confidence measure. This confidence measure can be used e.g. in interactive systems to report possible errors to the user or to propose translations only when they are likely to be correct.

Confidence measures have been extensively studied for speech recognition, but are not well known in other areas. Only recently have researchers started to investigate confidence measures for machine translation [1, 2, 5, 15].

We apply word confidence measures in MT as follows: For a given translation produced by an MT system, we calculate the confidence of each generated word and compare it to a threshold. All words whose confidence is above this threshold are tagged as correct and all others are tagged as incorrect translations. As stated before, this approach is related to the minimization of the expected number of *word* errors instead of sentence errors.

In this section, we will shortly review some of the word confidence measures that have proven most effective, and show their connection with the Bayes risk as derived in Section 2.1. In addition, we will introduce new confidence measures and give an experimental comparison of the different methods.

3.2 Word Posterior Probabilities

In [15], different variants of word posterior probabilities which are applied as word confidence measures are proposed. We study three types of confidence measures:

Target position: One of the approaches to word posterior probabilities presented in [15] can be stated as follows: the posterior probability $p_i(e | f_1^J, \hat{I}, \hat{e}_1^{\hat{I}} \setminus \hat{e}_i)$ expresses the probability that the target word e occurs in position i (given the other words in the target sentence $\hat{e}_1^{\hat{I}} \setminus \hat{e}_i$). In Section 2.3, we saw that the (modified) word error measure WER directly leads to this word posterior probability. Thus, we study this word confidence measure here.

The word posterior probability can be calculated over an N -best list of alternative translations that is generated by an SMT system. We determine all sentences that contain the word e in position i (or a target position Levenshtein aligned to i) and sum their probabilities, i.e.

$$p_i(e|f_1^J, \hat{I}, \hat{e}_1^{\hat{I}} \setminus \hat{e}_i) = \frac{p_i(e, f_1^J, \hat{I}, \hat{e}_1^{\hat{I}} \setminus \hat{e}_i)}{\sum_{e'} p_i(e', f_1^J, \hat{I}, \hat{e}_1^{\hat{I}} \setminus \hat{e}_i)} ,$$

where

$$p_i(e, f_1^J, \hat{I}, \hat{e}_1^{\hat{I}} \setminus \hat{e}_i) = \sum_{\tilde{I}, \tilde{e}_1^{\tilde{I}}: \mathcal{L}_i(\tilde{e}_1^{\tilde{I}}, \tilde{e}_i^{\tilde{I}})=e} p(\tilde{I}, \tilde{e}_1^{\tilde{I}}, f_1^J) . \quad (3)$$

This probability depends on the target words $\hat{e}_1^{\hat{I}} \setminus \hat{e}_i$ in the generated string, because it is based on the Levenshtein alignment $\mathcal{L}_i(\hat{e}_1^{\hat{I}}, \tilde{e}_1^{\tilde{I}})$.

Average target position: Due to the reordering of words which takes place in translation, the same target word may appear in different positions in the generated translations. The word posterior probabilities based on target positions presented above partially compensate for this effect by determining the Levenshtein alignment over the N -best list. Nevertheless, this cannot handle all reordering within the sentence that may occur. Therefore, we also introduce a new version of word posterior probabilities that determines the *average* over all posterior probabilities based on target positions:

$$p_{\text{avg}}(e|f_1^J) = \frac{p_{\text{avg}}(e, f_1^J)}{\sum_{e'} p_{\text{avg}}(e', f_1^J)} , \quad p_{\text{avg}}(e, f_1^J) = \frac{1}{I^*} \sum_{\tilde{I} \geq i, \tilde{e}_1^{\tilde{I}}: \tilde{e}_i^{\tilde{I}}=e} p(\tilde{I}, \tilde{e}_1^{\tilde{I}}, f_1^J) \quad (4)$$

where I^* is the maximum of all generated sentence lengths. The idea is to determine the probability of word e occurring in a generated sentence at all - without regarding a fixed target position. Note that here no Levenshtein alignment is performed, because the variation in sentence positions is accounted for through the computation of the arithmetic mean.

Word count: In addition to the word posterior probabilities described above, we also implemented a new variant that can be derived from Eq. 2 (Sec. 2.3), taking the counts of the words in the generated sentence into account, i.e. we determine the probability of target word e occurring in the sentence n_e times:

$$p_e(n_e|f_1^J) = \frac{p_e(n_e, f_1^J)}{\sum_{n'_e} p_e(n'_e, f_1^J)} ,$$

where

$$p_e(n_e, f_1^J) = \sum_{\tilde{n}_1^E: \tilde{n}_e=n_e} p(\tilde{n}_1^E, f_1^J) = \sum_{\tilde{I}, \tilde{e}_1^{\tilde{I}}: \tilde{n}_e=n_e} p(\tilde{I}, \tilde{e}_1^{\tilde{I}}, f_1^J) . \quad (5)$$

Implementation: As already stated above, the word posterior probabilities can be calculated over N -best lists generated by an SMT system. Thus, the sum over all possible target sentences \tilde{e}_1^I is carried out over the alternatives contained in the N -best list. If the list is long enough, this approximation is not harmful. In our experiments, we used 1,000-best lists.¹

Since the true probability distribution $Pr(I, e_1^I, f_1^J)$ is unknown, we replace it by a *model distribution* $p(I, e_1^I, f_1^J)$. This model distribution is the one from the SMT baseline system (see Section 4.2).

3.3 IBM-1

We implemented another type of confidence measure that determines the translation probability of the target word e averaged over the source sentence words according to Model 1 introduced by IBM in [3]. We determine the probability according to the formula²

$$p_{\text{IBM-1}}(e|f_1^J) = \frac{p_{\text{IBM-1}}(e, f_1^J)}{\sum_{e'} p_{\text{IBM-1}}(e', f_1^J)}, \quad p_{\text{IBM-1}}(e, f_1^J) = \frac{1}{J+1} \sum_{j=0}^J p(e|f_j) \quad (6)$$

where f_0 is the 'empty' source word [3]. The probabilities $p(e|f_j)$ are word based lexicon probabilities, i.e. they express the probability that e is a translation of the source word f_j .

Investigations on the use of the IBM-1 model for word confidence measures showed promising results [1, 2]. Thus, we apply this method here in order to compare it to the other types of confidence measures.

4 Results

4.1 Task and Corpus

The experiments are performed on a French-English corpus consisting of technical manuals of devices such as printers. This corpus is compiled within the European project TransType2 [13] which aims at developing computer aided MT systems that apply statistical techniques. For the corpus statistics see Table 1.

¹ In the area of speech recognition, much shorter lists are used [7]. The justification is that the probabilities of the hypotheses which are lower in the list are so small that they do not have any effect on the calculation of the word posterior probabilities. Nevertheless, we use longer N -best lists here to be on the safe side.

² Note that this probability is different from the one calculated in [1]; it is normalized over all target words e' . Nevertheless, both measures perform similarly well.

Table 1. Statistics of the training, development and test set.

		French	English
Training	Sentences	52 844	
	Words + Punctuation Marks	691 983	633 770
	Vocabulary Size	14 831	13 201
	Singletons	4 257	3 592
Develop	Sentences	994	
	Words	11 731	10 903
Test	Sentences	984	
	Words	11 800	11 177

4.2 Experimental Setting

As basis for our experiments, we created 1,000-best lists of alternative translations using a state-of-the-art SMT system. The system we applied is the so-called alignment template system as described in [12]. The key elements of this approach are pairs of source and target language phrases together with an alignment between the words within the phrases.

This system – like virtually all state-of-the-art SMT systems – applies the Bayes decision rule in Eq. 1, i.e. it takes the decision based on *sentence* error.

4.3 Word Error Measures

It is not intuitively clear how to classify words in MT output as correct or incorrect when comparing the translation to one or several references. In the experiments presented here, we applied WER and PER for determining which words in a translation hypothesis are correct. Thus, we can study the effect of the word posterior probabilities derived from the error measures in Section 2.3 on the error measures they are derived from.

These error measures behave significantly different with regard to the percentage of words that are labeled as correct. WER is more pessimistic than PER and labels 58% of the words in the develop and test corpus as correct, whereas PER labels 66% as correct.

4.4 Evaluation Metrics

After computing the confidence measure, each generated word is tagged as either *correct* or *false*, depending on whether its confidence exceeds the tagging threshold that has been optimized on the development set beforehand. The performance of the confidence measure is evaluated using three different metrics:

- **CAR** (**C**onfidence **A**ccuracy **R**ate): The CAR is defined as the number of correctly assigned tags divided by the total number of generated words in the translation. The baseline CAR is given by the number of correct words in the

generated translation, divided by the number of generated words. The CAR strongly depends on the tagging threshold. Therefore, the tagging threshold is adjusted on a development corpus *distinct* from the test set.

- **ROC (Receiver Operating Characteristic curve)** [4]: The ROC curve plots the *correct rejection rate* versus *correct acceptance rate* for different values of the tagging threshold. The correct rejection rate is the number of incorrectly translated words that have been tagged as wrong, divided by the total number of incorrectly translated words. The correct acceptance rate is the ratio of correctly translated words that have been tagged as correct. These two rates depend on each other: If one of them is restricted by a lower bound, the other one cannot be restricted. The further the ROC curve lies away from the diagonal, the better the performance of the confidence measure.
- **AROC (Area under ROC curve)**: This value specifies twice the size of the area between the ROC curve and the diagonal; it ranges from 0 to 1. The higher this value, the better the confidence measure discriminates.

4.5 Experimental Results

We studied the performance of the word confidence measures described in Sections 3.2 and 3.3. The results are given in Tables 2 and 3. For both the word error measure PER as well as WER, the best performance in terms of CAR is achieved by the IBM-1 based word confidence measure.

Table 2. CAR [%] and AROC [%] on the test corpus. The error counting is based on PER (see Sec. 4.3).

	CAR	AROC
Baseline	64.4	-
averaged target position (Eq. 4)	64.8	6.6
target position (Eq. 3)	67.2	28.3
word counts (Eq. 5)	68.1	29.8
IBM-1 (Eq. 6)	71.6	21.5

When comparing the CAR for the word posterior probabilities given in Eqs. 3,4,5 and the IBM-1 based confidence measure, it is surprising that the latter performs significantly better (with regard to both word error measures WER and PER). The IBM-1 model is a very simple translation model which does not produce high quality translations when applied in translation. Thus it was interesting to see that it discriminates so well between good and bad translations. Moreover, this method relies only on one target hypothesis (and the source sentence), whereas the word posterior probabilities take the whole space of possible translations (represented by the N -best list) into account.

In contrast to the good performance in CAR, the IBM-1 based confidence measure yields a much lower AROC value than two of the other measures. Looking

Table 3. CAR [%] and AROC [%] on the test corpus. The error counting is based on WER (see Sec. 4.3).

	CAR	AROC
Baseline	55.9	-
averaged target position (Eq. 4)	59.3	9.1
target position (Eq. 3)	64.1	26.4
word counts (Eq. 5)	62.0	20.8
IBM-1 (Eq. 6)	66.3	18.7

at the ROC curve³ in Figure 2 we find the reason for this: there is a small area on the left part of the curve where the IBM-1 model based confidence measure actually discriminates better than all of the other confidence measures. Nevertheless, the overall performance of the word posterior probabilities based on target positions and of those based on the word count are better than that of the IBM-1 based confidence measure.

We assume that the better performance of the IBM-1 based confidence measure is due to the fact that the involved lexicon probabilities do not depend on the specific N -best lists, but on a translation model that is trained on the whole training corpus. Thus, they are more exact and do not rely on an approximation as introduced by the Levenshtein alignment (cf. Eq. 3). Moreover, the tagging threshold can be estimated very reliably, because it will be the same for the develop and the test corpus. In order to verify this assumption, we analyzed the CAR on the develop corpus. This is used for the optimization of the tagging threshold. Thus, if the other measures indeed discriminate as well as or better than the IBM-1 based confidence measure, their CAR should be higher for the optimized threshold. Table 4 presents these experiments. We see that indeed the word posterior probabilities based on target positions and those based on word counts have a high accuracy rate and show a performance similar to (or better than) that of the IBM-1 based confidence measure.

Table 4. Comparative experiment: CAR [%] on the develop set (threshold optimized). Results for error counting based on PER and WER (see Sec. 4.3).

	word error measure	
	WER	PER
Baseline	60.5	67.1
averaged target position (Eq. 4)	62.1	67.5
word counts (Eq. 5)	67.3	72.1
target position (Eq. 3)	69.0	71.3
IBM-1 (Eq. 6)	67.8	72.5

³ We only present the ROC curve for the word error measure PER here. The curve for WER looks very similar.

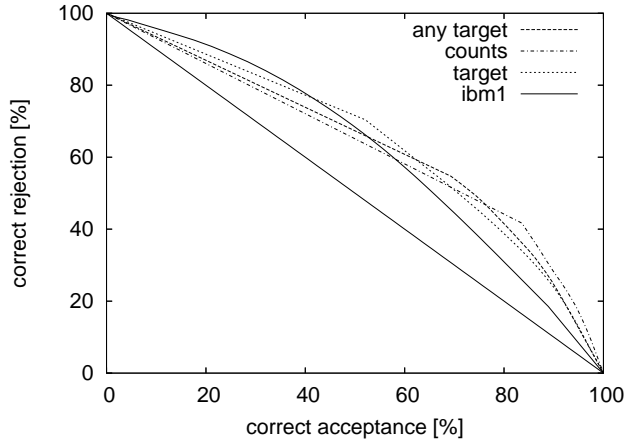


Fig. 2. ROC curve on the test set. The error counting is based on PER (see Sec. 4.3).

5 Outlook

We saw in the derivation from Bayes risk that word posterior probabilities are closely related to *word* error rate minimization. Moreover, we found that they show a state-of-the-art performance as confidence measures on the word level. Therefore, we plan to apply them directly in the machine translation process and study their impact on translation quality. One possible approach would be to combine them with the sentence error based decision rule that is widely used for rescoreing N -best lists.

6 Conclusion

In this work, we have taken first steps towards studying the relationship between Bayes decision rule and confidence measures. We have presented two forms of Bayes decision rule for statistical machine translation: the well-known and widely-applied rule for minimizing sentence error, and one novel approach that aims at minimizing word error. We have investigated the relation between two different word error measures and word confidence measures for SMT that can be directly derived from Bayes risk.

This approach lead to a theoretical motivation for the target position based confidence measures as proposed in [15]. In addition, we derived new confidence measures that reduced the baseline error in discriminating between correct and incorrect words in MT output by a quarter. Other studies report similar reductions for Chinese–English translation [1, 2].

Acknowledgement

This work was partly supported by the National Science Foundation under Grant No. 0121285, and by the RTD project TransType2 (IST-2001-32091) funded by the European Commission.

References

1. Blatz, J., Fitzgerald, E., Foster, G., Gandrabur, S., Goutte, C., Kulesza, A., Sanchis, A., Ueffing, N.: Confidence Estimation for Machine Translation. Final report of the JHU/CLSP Summer Workshop. (2003) <http://www.clsp.jhu.edu/ws2003/groups/estimate/>
2. Blatz, J., Fitzgerald, E., Foster, G., Gandrabur, S., Goutte, C., Kulesza, A., Sanchis, A., Ueffing, N.: Confidence Estimation for Machine Translation. Proc. 20th Int. Conf. on Computational Linguistics (COLING). (2004)
3. Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., Mercer, R. L.: The Mathematics of Statistical Machine Translation: Parameter Estimation. Computational Linguistics, Vol. 19 (2). (1993) 263–311.
4. Duda, R. O., Hart, P. E., Stork, D. G. Pattern classification. John Wiley & Sons, New York (2001)
5. Gandrabur, S., Foster, G.: Confidence Estimation for Text Prediction. Proc. Conf. on Natural Language Learning (CoNLL). Edmonton, Canada (2003) 95–102
6. Goel, V., Byrne, W.: Minimum Bayes-Risk Automatic Speech Recognition. In: W. Chou and B. H. Juang (eds.): Pattern Recognition in Speech and Language Processing. CRC Press (2003)
7. Komatani, K., Kawahara, T.: Flexible Mixed-Initiative Dialogue Management using Concept-Level Confidence Measures of Speech Recognizer Output. Proc. 18th Int. Conf. on Computational Linguistics (COLING). Saarbrücken, Germany (2000) 467–473
8. Kumar, S., Byrne, W.: Minimum Bayes-Risk Decoding for Statistical Machine Translation. Human Language Technology conference/North American chapter of the Association for Computational Linguistics annual meeting (HLT/NAACL). Boston, MA (2004)
9. Levenshtein, V. I.: Binary Codes Capable of Correcting Deletions, Insertions and Reversals. Soviet Physics Doklady, Vol. 10 (8) (1966) 707–710
10. Ney, H., Popović, M., Sündermann, D.: Error Measures and Bayes Decision Rules Revisited with Applications to POS Tagging. Proc. Conf. on Empirical Methods for Natural Language Processing (EMNLP). Barcelona, Spain (2004)
11. Och, F. J.: Minimum Error Rate Training for Statistical Machine Translation. Proc. 41th Annual Meeting of the Assoc. for Computational Linguistics (ACL). Sapporo, Japan (2003)
12. Och, F. J., Ney, H.: The alignment template approach to statistical machine translation. *To appear in* Computational Linguistics (2004)
13. TransType2 – Computer Assisted Translation. RTD project TransType2 (IST-2001-32091) funded by the European Commission. <http://tt2.sema.es/>
14. Vogel, S., Zhang, Y., Huang, F., Tribble, A., Venugopal, A., Zhao, B., Waibel, A.: The CMU Statistical Machine Translation System. Proc. MT Summit IX. New Orleans, LA (2003)
15. Ueffing, N., Macherey, K., Ney, H.: Confidence Measures for Statistical Machine Translation. Proc. MT Summit IX. New Orleans, LA (2003) 394–401
16. Ueffing, N., Ney, H.: Confidence Measures for SMT and Bayes Decision Rule. Unpublished results. RWTH Aachen University, Computer Science Department (2004)