Confidence Measures for Statistical Machine Translation

Nicola Ueffing Klaus Macherey Hermann Ney

Lehrstuhl für Informatik VI – Computer Science Department
RWTH Aachen – University of Technology
{ueffing,kmach,ney}@cs.rwth-aachen.de

Abstract

In this paper, we present several confidence measures for (statistical) machine translation. We introduce word posterior probabilities for words in the target sentence that can be determined either on a word graph or on an N best list. Two alternative confidence measures that can be calculated on N best lists are proposed. The performance of the measures is evaluated on two different translation tasks: on spontaneously spoken dialogues from the domain of appointment scheduling, and on a collection of technical manuals.

1 Introduction

With the rising number of applications of machine translation (MT), the demand for the ability to spot erroneous words also increases. A method for labeling the generated words as either correct or incorrect enables the system to signal possible errors to the user or to propose only those words as translations that are likely to be correct.

Confidence measures are widely used in speech recognition (see e.g. (Weintraub et al., 1997; Wessel et al., 2001)), but until recently they have not been applied in the area of MT. (Gandrabur and Foster, 2003) introduced confidence measures for a translation prediction task in an interactive environment. They estimate confidence for up to four predicted words. Unlike this, our approach allows for the calculation of confidence for each word in a sentence generated by the system. Thus, it can be applied to interactive systems like the one described in (Och et al., 2003), e. g. to mark words with a low confidence for correction.

We present methods for the calculation of confidence measures for MT that rely only on information contained in the output of an MT system. They are based on word graphs and N best lists. For each word in a target sentence, the posterior probability is computed and employed as confidence measure. Furthermore, alternative confidence measures computed on N best lists are introduced and compared to the word posterior probabilities.

The remainder of the paper is organized as follows: A short introduction to Statistical Machine Translation (SMT) is given in Section 2; and Section 3 presents the proposed confidence measures. Section 3.2 explains the computation of word posterior probabilities on word graphs, and Section 3.3 that on N best lists. The two alternative N best list based confidence measures are introduced in Section 3.4. In Section 4, the different confidence measures are evaluated on two different translation tasks. The conclusions are given in Section 5.

2 Statistical Machine Translation

The goal of machine translation is the translation of an input string $f_1 \ldots f_J$ in the source language into a target language string $e_1 \ldots e_I$. We choose the string that has maximum probability given the source string, $Pr(e_1^I|f_1^J)$. Applying Bayes' decision rule yields the following criterion:

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

The correspondence between the words in the source and the target string is described by alignments which can be viewed as mappings $B : i \rightarrow B_i \subset \{1, \ldots, J\}$ assigning a set B_i of source positions to each target position i (including the empty position zero). Note that this conception is different from the one introduced in (Brown et al., 1993) where the alignment assigns (exactly) one target position to each source position.

The search (denoted by the $\arg\max$ operation in

Eq. 1) explores the space of all possible target language strings e_1^I and all possible alignments B_0^I between the source and the target language string to find the one with maximum probability. Applying the maximum approximation, this yields

$$(\hat{e}_1^I, \hat{B}_0^I) \;=\; \arg\max_{e_1^I,\, B_0^I} Pr(e_1^I,\, B_0^I \mid f_1^J)\,. \tag{2}$$

This decision criterion aims at the minimization of the expected number of *sentence* errors.

For descriptions of SMT systems see for example (Och and Ney, 2002; Vogel et al., 2000; Takezawa et al., 1998; Yamada and Knight, 2002).

3 Confidence Measures

For a given translation produced by an MT system, we want to measure the confidence of being correct for each generated word. That is, for each word in the target hypothesis, the confidence is to be calculated and compared to a tagging threshold which has been optimized on a development corpus. All words whose confidence is above this threshold are tagged as correct and all others are tagged as false.

Unlike the criterion given in Eq. 2, this approach aims at the minimization of the expected number of *word* errors instead of sentence errors.

3.1 Word Posterior Probabilities

In speech recognition, confidence measures based on word posterior probabilities as proposed in (Wessel et al., 2001) have proven to be among the very effective methods known. We transfer this concept to the area of machine translation and show that it can be applied successfully as well.

We compare two different approaches to the calculation of word posterior probabilities:

1. We regard the target position i in which the word e occurs in a specific hypothesis and determine the word posterior probability $p_i(e|f_1^J)$.
2. Word posterior probabilities $p_B(e|f_1^J)$ are defined for a target word e, considering the set B of source positions that is aligned to this word in a specific hypothesis.

The difference between the two approaches is illustrated in the example in Table 1 which will be explained in detail at the end of this subsection.

The word posterior probability can be determined by summing up the posterior probabilities of all sentences which contain this specific word in position i or aligned to the source positions B. Let

$$\begin{aligned} & p(e_1^I, B_0^I, f_1^J) = \\ & = \prod_{k=0}^{I} \Big\{ p(f_{B_k}|e_k) \cdot p(B_k|B_{k-1}) \cdot p(e_k|e_1^{k-1}) \Big\}\,, \end{aligned}$$

where $p(f_{B_k}|e_k) = \prod_{j \in B_k} p(f_j|e_k)$ is the probability of translating the source words in B_k with e_k, $p(B_k|B_{k-1})$ is the alignment probability, and $p(e_k|e_1^{k-1})$ is the language model probability.

For criterion 1 introduced above, we obtain the word posterior probability by the following summation

$$\begin{aligned} & p_i(e|f_1^J) = \\ & = \frac{1}{p(f_1^J)} \sum_{(B_0^I,\, e_1^I):\, e_i = e} p(e_1^I, B_0^I, f_1^J)\,. \end{aligned} \tag{3}$$

Applying criterion 2 instead leads to

$$\begin{aligned} & p_B(e|f_1^J) = \\ & = \frac{1}{p(f_1^J)} \sum_{\substack{(B_0^I,\, e_1^I): \\ \exists i:\, (e_i, B_i) = (e, B)}} p(e_1^I, B_0^I, f_1^J)\,. \end{aligned} \tag{4}$$

Table 1 shows a German input sentence together with four different translations. The aligned source positions of each generated target word are given as index. We see that the position of a word in the target sentence can differ due to insertions, deletions and reordering of words. If we want to determine the confidence of the word '?' in Translation 1, the two rightmost columns show the sentences that are taken into account for the summation in Eqs. 3 and 4, respectively. For criterion 2, we have more sentences that match, yielding a more reliable confidence estimation (as the experiments in Section 4 will show).

The posterior probability can directly be used as a measure of confidence as described above. We denote these two confidence measures by

$$\begin{aligned} \mathcal{C}_{target}(e) &= p_i(e|f_1^J) \\ \text{and} \quad \mathcal{C}_{source}(e) &= p_B(e|f_1^J)\,. \end{aligned}$$

These confidence measures are probabilistic and exploit only information which is contained in the output of an SMT system.

Table 1: Illustration of Eqs. 3 and 4: Sentences taken into account for the calculation of the word posterior probability of the word '?' in Translation 1 are marked with a '+'.

Source	was hast du gesagt ?	target (criterion 1)	source (criterion 2)
Translation 1	what$_1$ did$_2$ you$_3$ say$_4$?$_5$	$+$	$+$
Translation 2	what$_1$ have$_2$ you$_3$ said$_4$?$_5$	$+$	$+$
Translation 3	what$_1$ did$_2$ you$_3$ just$_0$ say$_4$?$_5$		$+$
Translation 4	what$_1$ you$_3$ did$_2$ say$_4$.$_5$		

3.2 Word Posterior Probabilities on Word Graphs

Using an SMT system, we construct a word graph as described in (Ueffing et al., 2002): The nodes represent sets of covered source sentence positions and differentiate between different language model histories. An edge is labeled with the target word generated as a translation of the covered source position(s). The edges are also annotated with the probabilities of the different translation submodels. Each path from the source of the graph (i.e. the node with zero covered source sentence positions) to the sink (i.e. the node where all source sentence positions are covered) is an alternative target sentence hypothesis. Thus, the word graph contains the most probable sentence hypotheses that survived the pruning during search and can be used to approximate the word posterior probabilities as defined in Eqs. 3 and 4.

The word graphs are then compressed by merging all nodes which have the same coverage vector and the same source position covered last into one.

The information relevant for determining the word probabilities of a word e in the best generated target sentence is the following:

- the translation probability $p(f_B|e)$, where f_B is the set of source words aligned to e,
- for $\mathcal{C}_{target}$: the position i of word e in the target hypothesis,
- for $\mathcal{C}_{source}$: the set B of source positions aligned to e,
- the alignment probability $p(B|B')$, where B' is the set of source positions covered by the predecessor target word of e.

The language model probabilities can be calculated on the fly.

The computation of word posterior probabilities on word graphs can be performed by applying a forward-backward algorithm. Assume we use a trigram language model. The formulae we are going to present can be extended to capture for longer histories as well, but in order to keep the presentation understandable, we will show them only for trigrams.

Target position dependency

Let C_i be the coverage set of the partial hypothesis $e_1 \ldots e_i$, i.e. $C_i = \bigcup_{k=1}^{i} B_k$. Throughout the paper, we denote the complement of a set C by $\overline{C}$.

If we apply $\mathcal{C}_{target}$ and keep the *target* position fix, we can compute the forward probability as follows:

$$\begin{aligned}\Phi_i(e_{i-1}, C_{i-1}; e_i, B_i) &= \qquad (5)\\ = \; p(f_{B_i}|e_i) \cdot \sum_{e_{i-2}} \sum_{B_{i-1}} \Big\{ & p(B_i|B_{i-1}) \\ & \cdot p(e_i|e_{i-1}e_{i-2}) \\ & \cdot \Phi_{i-1}(e_{i-2}, C_{i-1} \setminus B_{i-1}; e_{i-1}, B_{i-1}) \Big\} \,,\end{aligned}$$

which is calculated recursively in ascending order of i.

For the computation of backward probabilities, the probability of completing a sentence from word e in position i is

$$\begin{aligned}\Psi_i(e_i, B_i; e_{i+1}, \overline{C_i}) &= \qquad (6)\\ = \; p(f_{B_i}|e_i) \cdot \sum_{e_{i+2}} \sum_{B_{i+1}} \Big\{ & p(B_{i+1}|B_i) \\ & \cdot p(e_{i+2}|e_{i+1}e_i) \\ & \cdot \Psi_{i+1}(e_{i+1}, B_{i+1}; e_{i+2}, \overline{C_{i+1}}) \Big\} \,.\end{aligned}$$

This can be recursively determined in descending order of i.

Using the forward-backward algorithm, the word posterior probability can be calculated according to

$$p_i(e_i|f_1^J) = \frac{1}{p(f_1^J)} \sum_{e_{i-1}} \sum_{e_{i+1}} \sum_{C_{i-1}} \sum_{B_i} p(e_{i+1}|e_i e_{i-1}) \cdot \frac{\Phi_i(e_{i-1}, C_{i-1}; e_i, B_i) \cdot \Psi_i(e_i, B_i; e_{i+1}, \overline{C_i})}{p(f_{B_i}|e)}$$

with the normalization term

$$\begin{aligned} p(f_1^J) &= \sum_{I} \sum_{e_I} \sum_{B_I} \sum_{e_{I-1}} \Phi_I(e_{I-1}, \overline{B_I}; e_I, B_I) \\ &= \sum_{e_1} \sum_{B_1} \sum_{e_2} \Psi_1(e_1, B_1; e_2, \overline{B_1}) \\ &\qquad \cdot p(e_2|e_1) \cdot p(e_1)\ . \end{aligned}$$

Here, I is the last position in the generated target sentences.

Source position dependency

Instead of regarding the position of word e in the target sentence, we determine the posterior probability according to the set B of source positions that e is aligned to. This approach allows for the direct comparison of translation hypotheses of different lengths.

Applying this view, we cannot calculate a posterior probability for zero fertility words, because there are no source words that generate them. Therefore, we assign them a posterior probability of one.

Analogously to Eq. 5, the forward probability can be computed according to

$$\begin{aligned} &\Phi_B(e', C'; e) = \\ &\quad = p(f_B|e) \cdot \sum_{e''} \sum_{B' \subseteq C'} \Phi_{B'}(e'', C' \setminus B'; e') \\ &\qquad\qquad \cdot p(e|e'e'') \cdot p(B|B')\ , \end{aligned}$$

where B' is the set of source positions covered by the predecessor word e', and $C' \subseteq \overline{B}$.

The backward probability is determined as

$$\begin{aligned} &\Psi_B(e; \tilde{e}, \tilde{C}) = \\ &\quad = p(f_B|e) \cdot \sum_{\tilde{\tilde{e}}} \sum_{\tilde{B} \subseteq \tilde{C}} \Psi_{\tilde{B}}(\tilde{e}; \tilde{\tilde{e}}, \tilde{\tilde{C}}) \\ &\qquad\qquad \cdot p(\tilde{\tilde{e}}|\tilde{e}e) \cdot p(\tilde{B}|B)\ , \end{aligned}$$

where $\tilde{C} \subseteq \overline{B}$. This yields the word posterior probability formula

$$p_B(e|f_1^J) = \frac{1}{p(f_1^J)} \sum_{e'} \sum_{\tilde{e}} \sum_{C' \subseteq \overline{B}} p(\tilde{e}|ee') \cdot \frac{\Phi_B(e', C'; e) \cdot \Psi_B(e; \tilde{e}, \overline{C' \cup B})}{p(f_B|e)}$$

with the normalization term

$$\begin{aligned} p(f_1^J) &= \sum_{I} \sum_{e_I} \sum_{B_I} \sum_{e_{I-1}} \Phi_{B_I}(e_{I-1}, \overline{B_I}; e_I) \\ &= \sum_{e_1} \sum_{B_1} \sum_{e_2} \Psi_{B_1}(e_1; e_2, \overline{B_1}) \\ &\qquad \cdot p(e_2|e_1) \cdot p(e_1)\ , \end{aligned}$$

where B_I is the set of source positions covered by the last word of the generated target sentences, and B_1 those covered by the first word, respectively.

3.3 Word Posterior Probabilities on N Best Lists

From a word graph, we can easily extract an N best list containing the N target sentences that obtained the highest probability by the translation and language model. They can be used as a representation of the possible target sentences as well, and we can determine the sum in Eq. 3 over the sentences in the N best list.

As mentioned already in Section 3.1, the generated target sentences may have different lengths and the position of a word within different sentences in the N best list might differ due to reorderings, deletions, and insertions. Thus, we determine the Levenshtein alignment (Levenshtein, 1966) on the sentences according to the best target sentence. That is, for each word $\hat{e}_i$ in the best target sentence $\hat{e}_1^I$, we determine the corresponding word w in any of the other sentences in the N best list. We denote the Levenshtein alignment of two sentences $\hat{e}_1^I$ and $w_1^{I_n}$ by $\mathcal{L} = \mathcal{L}(\hat{e}_1^I, w_1^{I_n})$ and that of word $\hat{e}_i$ by $\mathcal{L}_i(\hat{e}_1^I, w_1^{I_n})$ for $n = 2, \ldots, N$.

Using this Levenshtein alignment, we can easily compute word posterior probabilities for each word $\hat{e}_i$ in the best target sentence. We sum over the probabilities of all sentences containing the word in a position that is aligned to i in the Levenshtein alignment:

$$\begin{aligned} &p_i(\hat{e}_i|f_1^J, \hat{e}_1^I, \mathcal{L}) = \\ &\quad = \frac{\sum\limits_{n=1}^{N} p(f_1^J|w_1^{I_n}) \cdot p(w_1^{I_n}) \cdot \delta(\hat{e}_i, \mathcal{L}_i(\hat{e}_1^I, w_1^{I_n}))}{\sum\limits_{n=1}^{N} p(f_1^J|w_1^{I_n}) \cdot p(w_1^{I_n})}\ , \end{aligned}$$

where $\delta(.,.)$ is the Kronecker function.

This yields the confidence measure

$$\mathcal{C}_{prob}(\hat{e}_i) = p_i(\hat{e}_i|f_1^J, \hat{e}_1^I, \mathcal{L})\ .$$

3.4 Alternative Confidence Measures on N Best Lists

Apart from the word posterior probabilities as presented in Section 3.3, we investigated two simple confidence measures on N best lists: The relative frequency of a word in the list and the sum of the ranks of the target sentences containing this word. Those values can be calculated on N best lists produced by any MT system which does not have to be statistical.

Relative Frequency

For each word $\hat{e}_i$ in the best target sentence, we determine the number of sentences in the N best list containing this word in a position aligned to i. Then, we take the relative frequency of word $\hat{e}_i$ in the N best list with respect to the Levenshtein alignment directly as a confidence measure:

$$\mathcal{C}_{rel}(\hat{e}_i) = \frac{1}{N} \sum_{n=1}^{N} \delta(\hat{e}_i, \mathcal{L}_i(\hat{e}_1^I, w_1^{I_n})).$$

Rank sum

Another simple confidence measure that can be computed on N best lists is the sum of the ranks of those target sentences containing word $\hat{e}_i$ as $\mathcal{L}_i(\hat{e}_1^I, w_1^{I_n})$ (normalized by the total rank sum):

$$\mathcal{C}_{rank}(\hat{e}_i) = \frac{\sum\limits_{n=1}^{N} (N-n) \cdot \delta(\hat{e}_i, \mathcal{L}_i(\hat{e}_1^I, w_1^{I_n}))}{\frac{N}{2}(N+1)} .$$

Since we want ranks near to the top of the list to score better, we sum $N - n$ instead of the rank n.

3.5 Scaling of the Probabilities

During the translation process, the language model probability is raised to the power of 0.8 in order to give the language model probabilities a higher weight in the decision process. For the calculation of word posterior probabilities, we varied this scaling factor and optimized it on a development corpus in order to minimize the decision errors.

4 Results

4.1 Corpora

We performed experiments on two different corpora. One is the trilingual corpus which is successively built within the LC-STAR project. It comprises the languages English, Spanish, and Catalan. Experiments were carried out for all six translation directions. At the time of our experiments, we had about 13k sentences per language available; the statistics are given in Table 2. The corpus consists of transcriptions of spontaneously spoken dialogues in the domain of appointment scheduling and travel arrangements.

The second task we worked on is the TransType2 corpus. It consists of technical manuals like user guides, operating guides, and system administration guides for different devices. It comprises three language pairs, each of which contains English (E). The other three languages are French (F), Spanish (S), and German (G). The corpus statistics can be seen in Table 3.

4.2 Experimental Setup

We performed translation experiments with an implementation of the IBM-4 translation model (Brown et al., 1993). A description of the system can be found in (Tillmann and Ney, 2003).

The experimental setup for the two corpora is described in Table 4. It shows the baseline word error rate (WER), the graph error rate (GER), and the word graph density (WGD) for the different language pairs. The WER is based on the Levenshtein distance and computes the minimum number of substitution, insertion, and deletion operations that have to be performed to convert the generated string into the reference string. The GER is computed by determining the sentence in the word graph that has the minimum Levenshtein distance to a given reference. Thus, it is a lower bound for the word error rate. The WGD is computed as the total number of word graph edges divided by the number of words in the reference sentence.

Table 4: Experimental setup

Corpus		WGD	GER	WER
LC-STAR	C–E	12.6	19.8	31.7
	S–E	10.7	20.6	31.3
	C–S	24.7	8.5	18.4
	S–C	20.2	10.1	20.4
	E–C	21.0	19.7	33.5
	E–S	19.8	18.8	33.5
TransType2	E–S	14.6	22.8	31.2
	S–E	16.1	20.2	30.5
	F–E	33.8	43.0	58.1
	E–F	29.5	46.0	62.0
	G–E	45.8	47.2	63.6
	E–G	46.7	53.2	69.3

Table 2: Statistics of the LC-STAR corpus

		English	Spanish	Catalan
Training	Sentences	13 352		
	Words	123 454	118 534	118 137
Vocabulary	Size	2 154	3 933	3 572
Develop	Sentences	272		
	Words	2 267	2217	2211
Test	Sentences	262		
	Words	2 626	2 451	2 470

Table 3: Statistics of the TransType2 corpora

		Spanish	English	French	English	German	English
Training	Sentences	54 806		51 797		50223	
	Words	747 918	670 207	732 348	641 897	586 613	618 637
Vocabulary	Size	11 932	8 196	10 323	7 973	17 788	7 877
Develop	Sentences	1 012		994		964	
	Words	16 262	14 701	12 903	11 345	11 141	11 152
Test	Sentences	1 128		984		995	
	Words	11 481	9 969	12 723	11 576	12 416	12 613

4.3 Evaluation Metrics

After computing the confidence, 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 two different metrics:

- **C**onfidence **E**rror **R**ate

The CER is defined as the number of incorrectly assigned tags divided by the total number of generated words in the translated sentence. The baseline CER is given by the number of substitutions and insertions, divided by the number of generated words. The CER strongly depends on the tagging threshold. Therefore, the tagging threshold is adjusted beforehand on a development corpus *distinct* from the test set.

- **D**etection **E**rror **T**radeoff curve

The DET curve plots the *false rejection rate* versus the *false acceptance rate* for different values of the tagging threshold. The false rejection rate is defined as the number of correctly translated words that have been tagged as wrong, divided by the total number of correctly translated words. It is also referred to as *type I error*. The false acceptance rate (or *type II error*) is calculated as the number of incorrect words that have been accepted, divided by the total number of incorrectly translated words. If the type I error is restricted by a given $\alpha > 0$, the type II error usually cannot be restricted; both error rates depend on each other.

4.4 Experimental Results

Confidence Error Rates

Table 5 contains the CER for all language pairs of the two tasks. We compared the confidence measures determined on word graphs and those on N best lists. For each language pair, the minimal CER is shown in bold face.

We see that $\mathcal{C}_{source}$, the word posterior probabilities calculated on word graphs with respect to the aligned source position(s), outperform $\mathcal{C}_{target}$ and the N best list based measures in most of the cases. Nevertheless, the N best list based confidence measure $\mathcal{C}_{rank}$ significantly decreases the CER, likewise, and shows best performance among all measures for some of the language pairs.

Regarding the results of the confidence measures computed on N best lists, we see that the performance of the word posterior probabilities $\mathcal{C}_{prob}$ stays behind that of the simple measures. Here, the rank based criterion $\mathcal{C}_{rank}$ produces best results among the N best list based measures.

For some of the confidence measures and some language pairs, the CER does not decrease, but even slightly grows compared to the baseline. This is the case for the word graph based measure that depends on the target position of the word, $\mathcal{C}_{target}$, and for the probabilistic N best list measure, $\mathcal{C}_{prob}$. We believe that this is due to the fact that both of the measures refer to the position of the generated word in the target sentence which might differ between alternative hypotheses – even if the N best list method tries to compensate for this by determining the Levenshtein alignment.

Detection Error Tradeoff Curves

The DET curves for two language pairs are shown in Figures 1 and 2. They support the analysis presented above: The word graph based confidence measure $\mathcal{C}_{source}$ produces best results, closely followed by the N best list based measure $\mathcal{C}_{rank}$. The confidence measure $\mathcal{C}_{prob}$ clearly performs worst.

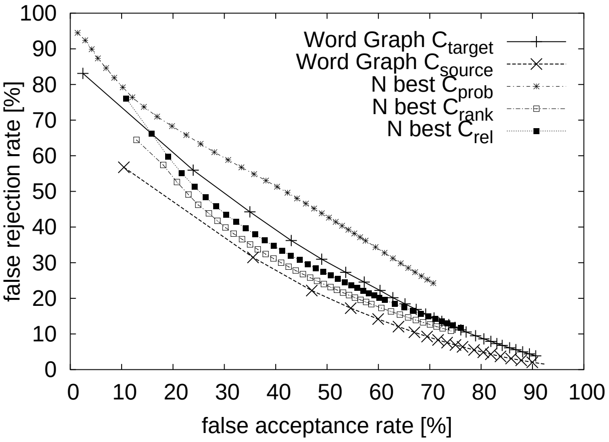

Figure 1: DET curves for LC-STAR C–S (word graph and 1000 best list)

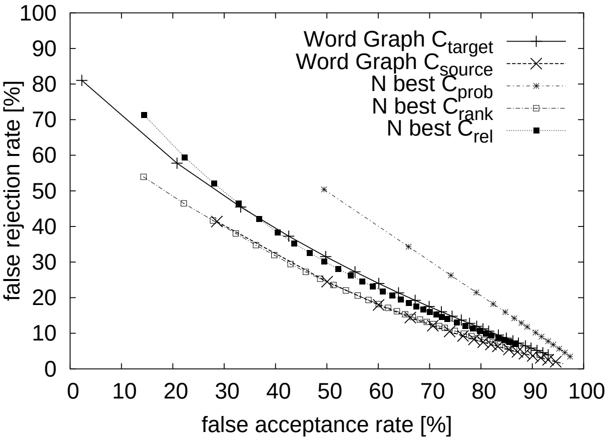

Figure 2: DET curves for TransType2 E–G (word graph and 1000 best list)

5 Conclusion

We presented several concepts of confidence measures for (statistical) MT systems. Applying them, the words in the generated target sentence can be tagged as correct or false, e.g. to facilitate post-editing or work in an interactive translation environment.

Word posterior probabilities were computed from information contained in the output of an SMT system, either from a word graph or an N best list. Furthermore, two alternative confidence measures computed on N best lists were introduced and their performance was compared with that of the word posterior probabilities. Those two methods are applicable to non-statistical MT systems as well.

A systematic evaluation was presented on the LC-STAR corpus which consists of spontaneously spoken dialogues in the domain of appointment scheduling and travel arrangement, and the TransType2 corpus comprising technical manuals.

Experiments showed that the word graph based confidence measure $\mathcal{C}_{source}$, depending on the source position(s) covered by the target word, yields best results. Nevertheless, a simple rank based criterion calculated on N best lists also performed well. Both of them significantly reduce the confidence error rate.

6 Acknowledgements

This work was partly supported by the projects LC-STAR (IST–2001–32216) and TransType2 (IST–2001–32091) by the European Community.

Table 5: Word Graph and N best list CER for LC-STAR and TransType2 [%]

Corpus		Baseline	Word Graph		1000 best list		
			$\mathcal{C}_{target}$	$\mathcal{C}_{source}$	$\mathcal{C}_{prob}$	$\mathcal{C}_{rank}$	$\mathcal{C}_{rel}$
LC-STAR	C–E	*16.7*	16.3	15.8	16.7	**15.5**	16.0
	S–E	*14.8*	14.4	14.0	14.7	**13.8**	14.1
	C–S	*10.3*	**9.3**	10.2	10.3	10.1	10.3
	S–C	*11.9*	11.1	**11.0**	11.9	11.9	11.9
	E–C	*19.2*	19.0	**17.0**	19.2	17.3	18.1
	E–S	*19.0*	18.7	18.4	18.9	**16.6**	17.3
TransType2	S–E	*13.7*	13.6	13.2	13.7	**13.1**	13.4
	E–S	*17.4*	17.3	**16.1**	17.4	16.8	17.0
	F–E	*32.3*	28.9	**26.3**	32.3	27.3	28.1
	E–F	*37.0*	31.8	**29.9**	36.9	31.0	32.1
	G–E	*37.4*	**29.5**	30.2	35.9	30.2	31.0
	E–G	*44.5*	31.6	**29.4**	36.1	30.3	32.3

7 References

P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

S. Gandrabur and G. Foster. 2003. Confidence estimation for text prediction. In *Proc. Conf. on Natural Language Learning (CoNLL)*, pp. 95–102, Edmonton, Canada, May.

V. I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–71, Feb.

F. J. Och and H. Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proc. 40th Annual Meeting of the Assoc. for Computational Linguistics (ACL)*, pp. 295–302, Philadelphia, PA, July.

F. J. Och, R. Zens, and H. Ney. 2003. Efficient search for interactive statistical machine translation. In *Proc. 10th Conf. of the Europ. Chapter of the Assoc. for Computational Linguistics (EACL)*, pp. 387–393, Budapest, Hungary, April.

T. Takezawa, T. Morimoto, Y. Sagisaka, N. Campbell, H. Iida, F. Sugaya, A. Yokoo, and S. Yamamoto. 1998. A Japanese-to-English speech translation system: ATR-MATRIX. In *Proc. 5th Int. Conf. on Spoken Language Processing (ICSLP)*, pp. 2779–2782, Sydney, Australia, Nov.

C. Tillmann and H. Ney. 2003. Word re-ordering and DP beam search for statistical machine translation. *Computational Linguistics*, 1(29):97–133.

N. Ueffing, F. J. Och, and H. Ney. 2002. Generation of word graphs in statistical machine translation. In *Proc. Conf. on Empirical Methods for Natural Language Processing (EMNLP)*, pp. 156–163, Philadelphia, PA, July.

S. Vogel, F. J. Och, C. Tillmann, S. Nießen, H. Sawaf, and H. Ney. 2000. Statistical methods for machine translation. In W. Wahlster, editor, *Verbmobil: Foundations of Speech-to-Speech Translation*, pp. 377–393. Springer Verlag: Berlin, Heidelberg, New York.

M. Weintraub, F. Beaufays, Z. Rivlin, Y. Konig, and A. Stolcke. 1997. Neural - network based measures of confidence for word recognition. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 887 – 890, Munich, Germany, April.

F. Wessel, R. Schlüter, K. Macherey, and H. Ney. 2001. Confidence measures for large vocabulary continuous speech recognition. *IEEE Trans. on Speech and Audio Processing*, 9(3):288–298, March.

K. Yamada and K. Knight. 2002. A decoder for syntax-based statistical MT. In *Proc. 40th Annual Meeting of the Assoc. for Computational Linguistics (ACL)*, pp. 303–310, Philadelphia, PA, July.