Word-Level Confidence Estimation for Machine Translation

Nicola Ueffing*
RWTH Aachen University

Hermann Ney†
RWTH Aachen University

This article introduces and evaluates several different word-level confidence measures for machine translation. These measures provide a method for labeling each word in an automatically generated translation as correct or incorrect. All approaches to confidence estimation presented here are based on word posterior probabilities. Different concepts of word posterior probabilities as well as different ways of calculating them will be introduced and compared. They can be divided into two categories: System-based methods that explore knowledge provided by the translation system that generated the translations, and direct methods that are independent of the translation system. The system-based techniques make use of system output, such as word graphs or N*-best lists. The word posterior probability is determined by summing the probabilities of the sentences in the translation hypothesis space that contains the target word. The direct confidence measures take other knowledge sources, such as word or phrase lexica, into account. They can be applied to output from nonstatistical machine translation systems as well.*

Experimental assessment of the different confidence measures on various translation tasks and in several language pairs will be presented. Moreover, the application of confidence measures for rescoring of translation hypotheses will be investigated.

1. Introduction

The work presented in this article deals with confidence estimation for machine translation (MT). Because sentences generated by a machine translation system are often incorrect but may contain correct substrings, a method for identifying these correct substrings and finding possible errors is desirable. For this purpose, each word in the generated target sentence is assigned a value expressing the confidence that it is correct.

Confidence measures have been extensively studied for speech recognition. Only recently have researchers started to investigate confidence measures for machine translation (Gandrabur and Foster 2003; Ueffing, Macherey, and Ney 2003; Blatz et al. 2004; Quirk 2004). In this article, we will develop a sound theoretical framework for

* Now at National Research Council Canada, Interactive Language Technologies Group, Gatineau, Québec J8P 3G5, Canada. E-mail: nicola.ueffing@nrc.gc.ca.
† Lehrstuhl für Informatik VI, Computer Science Department, D-52056 Aachen, Germany. E-mail: ney@cs.rwth-aachen.de.

Submission received: 7 March 2006; revised submission received: 30 September 2006; accepted for publication: 3 October 2006.

calculating and evaluating word confidence measures. Possible applications of confidence measures include:

- marking words with low confidence as potential errors for post-editing
- improving translation prediction accuracy in TransType-style interactive machine translation (Gandrabur and Foster 2003; Ueffing and Ney 2005a)
- combining output from different machine translation systems: Hypotheses with low confidence can be discarded before selecting one of the system translations (Akiba et al. 2004), or the word confidence scores can be used in the generation of new hypotheses from the output of different systems (Jayaraman and Lavie 2005), or the sentence confidence value can be employed for reranking (Blatz et al. 2003).

The article is organized as follows: In Section 2, we briefly review the statistical approach to machine translation. The phrase-based translation system, which serves as the basis for one of the direct confidence measures, will be presented. Section 3 gives an overview of related work on confidence estimation for machine translation. Moreover, word posterior probabilities will be introduced, and we will explain how they can be used as word-level confidence measures. In Section 4, we describe so-called system-based methods for confidence estimation, which make use of the output of a statistical machine translation system, such as word graphs or N-best lists. In Section 5, we present confidence measures based on direct models. The combination of several confidence measures into one is described in Section 6. Experimental evaluation and comparison of the different confidence measures is provided in Section 7. Section 8 deals with the rescoring of translation hypotheses using confidence measures. The article concludes in Section 9.

2. Statistical Machine Translation

2.1 General

In statistical machine translation (SMT), the translation is modeled as a decision process: Given a source string $f_1^J = f_1 \ldots f_j \ldots f_J$, we seek the target string $e_1^I = e_1 \ldots e_i \ldots e_I$ with maximal posterior probability:

$$\hat{e}_1^{\hat{I}} = \underset{I,e_1^I}{\operatorname{argmax}} \left\{ Pr(e_1^I \,|\, f_1^J) \right\} = \underset{I,e_1^I}{\operatorname{argmax}} \left\{ Pr(f_1^J \,|\, e_1^I) \cdot Pr(e_1^I) \right\} \tag{1}$$

Through this decomposition of the probability, we obtain two knowledge sources: the translation model $Pr(f_1^J \,|\, e_1^I)$ and the language model $Pr(e_1^I)$. Both can be modeled independently of each other. The translation model is responsible for linking the source string f_1^J and the target string e_1^I. It captures the semantics of the sentence. The target language model captures the well-formedness of the syntax in the target language.

Nowadays, most state-of-the-art SMT systems are based on bilingual phrases (Och, Tillmann, and Ney 1999; Koehn, Och, and Marcu 2003; Tillmann 2003; Bertoldi et al. 2004; Vogel et al. 2004; Zens and Ney 2004; Chiang 2005). A more detailed description of

a phrase-based approach to statistical machine translation will be given in the following section.

2.2 Review of the Phrase-Based Translation System

For the confidence measures which will be introduced in Section 5.1, we use a state-of-the-art phrase-based translation approach as described in Zens and Ney (2004). The key elements of this translation approach are bilingual phrases. Note that these phrases are sequences of words in the two languages and not necessarily phrases in the linguistic sense. The bilingual phrases are extracted from a word-aligned bilingual training corpus.

In this translation approach, the posterior probability $Pr(e_1^I \,|\, f_1^J)$ is modeled directly using a weighted log-linear combination of a language model, a phrase translation model, and a word-based lexicon model. The translation models are used for both directions: $p(f \,|\, e)$ and $p(e \,|\, f)$. Additionally, a word penalty and a phrase penalty are applied. With the exception of the language model, all models can be considered as within-phrase models as they depend only on a single phrase pair, but not on the context outside the phrase.

In the following, we will present the generation criterion for the phrase-based translation approach. This will be done for a monotone search in order to keep the equations simple. The extension to the non-monotone case is straightforward. Let (j_0^K, i_0^K) be a segmentation of the source sentence into phrases, where $j_{k-1} < j_k$ and $i_{k-1} < i_k$ for $k = 1, \ldots, K$. The corresponding (bilingual) phrase pairs are denoted as

$$(\tilde{f}_k, \tilde{e}_k) = (f_{j_{k-1}+1}^{j_k}, e_{i_{k-1}+1}^{i_k}),\ k = 1, \ldots, K$$

Assume a trigram language model. The phrase-based approach to SMT is then expressed by the following equation:

$$\hat{e}_1^{\hat{I}} = \underset{j_0^K, i_0^K, I, e_1^I}{\operatorname{argmax}} \left\{ \prod_{i=1}^{I} \left[c_1 \cdot p(e_i \,|\, e_{i-2}^{i-1})^{\lambda_1} \right] \cdot \prod_{k=1}^{K} \Big[c_2 \cdot\ p(\tilde{f}_k \,|\, \tilde{e}_k)^{\lambda_2} \cdot p(\tilde{e}_k \,|\, \tilde{f}_k)^{\lambda_3} \cdot \right. \tag{2}$$

$$\left. \cdot \prod_{j=j_{k-1}+1}^{j_k} p(f_j \,|\, \tilde{e}_k)^{\lambda_4} \cdot \prod_{i=i_{k-1}+1}^{i_k} p(e_i \,|\, \tilde{f}_k)^{\lambda_5} \Big] \right\}$$

where $p(\tilde{f}_k \,|\, \tilde{e}_k)$ and $p(\tilde{e}_k \,|\, \tilde{f}_k)$ are the phrase lexicon models in both translation directions. The phrase translation probabilities are computed as a log-linear interpolation of the relative frequencies and the IBM model 1 probability. The single word–based lexicon models are denoted as $p(f_j \,|\, \tilde{e}_k)$ and $p(e_i \,|\, \tilde{f}_k)$, respectively. $p(f_j \,|\, \tilde{e}_k)$ is defined as the IBM model 1 probability of f_j over the whole phrase $\tilde{e}_k$, and $p(e_i \,|\, \tilde{f}_k)$ is the inverse model, respectively. c_1 is the so-called word penalty, and c_2 is the phrase penalty, assigning constant costs to each target language word/phrase. The language model is a trigram model with modified Kneser–Ney discounting and interpolation (Stolcke 2002). The search determines the target sentence and segmentation that maximize the objective function.

As Equation (2) shows, the sub-models are combined via weighted log-linear interpolation. The model scaling factors $\lambda_1, \ldots, \lambda_5$ and the word and phrase penalties are optimized with respect to some evaluation criterion (Och 2003) such as BLEU score.

The phrase-based translation model will be needed later to define the different confidence measures. We therefore introduce the following notation: Let $Q_{PM}(\tilde{f}_k, \tilde{e}_k)$ be the score of the phrase pair, which consists of the phrase penalty c_2, the phrase lexicon scores, and the two word lexicon model scores (see Equation (2)):

$$Q_{PM}(\tilde{f}_k, \tilde{e}_k) := c_2 \cdot p(\tilde{f}_k \,|\, \tilde{e}_k)^{\lambda_2} \cdot p(\tilde{e}_k \,|\, \tilde{f}_k)^{\lambda_3} \cdot \prod_{j=j_{k-1}+1}^{j_k} p(f_j \,|\, \tilde{e}_k)^{\lambda_4} \cdot \prod_{i=i_{k-1}+1}^{i_k} p(e_i \,|\, \tilde{f}_k)^{\lambda_5} \tag{3}$$

3. Confidence Measures for MT

3.1 Related Work

In many areas of natural language processing, confidence measures have scarcely been investigated. The exception is automatic speech recognition, where an extensive amount of research on the topic exists. Confidence measures are widely used in this area—for example, in dialogue systems and in unsupervised training. Recently, researchers have started to investigate confidence measures for machine translation (Blatz et al. 2003, 2004; Gandrabur and Foster 2003; Ueffing, Macherey, and Ney 2003; Quirk 2004; Sanchis 2004). This section gives an overview of confidence estimation for machine translation on the word level as well as the sentence level and discusses its applications.

The first work that studied confidence estimation for statistical machine translation was Gandrabur and Foster (2003). Their confidence measures consist of a combination of different features in a neural network. The confidence is estimated for a sequence of up to four words in an interactive machine translation environment. The probability of being a correct extension of a given sentence prefix is computed for this word sequence. The authors report significant improvement in quality of the predicted translations.

In 2003, a team at the yearly summer workshop at the Center for Language and Speech Processing (CLSP) at Johns Hopkins University, Baltimore, MD, developed confidence measures for machine translation. The combination of several confidence features using neural networks and a naive Bayes classifier was investigated. The workshop team studied confidence estimation on the word level as well as on the sentence level, though the focus was on the sentence level. The features applied included new features as well as those that had previously been developed by team members (Gandrabur and Foster 2003; Ueffing, Macherey, and Ney 2003). Among them were also some of the word posterior probabilities, which will be presented here. Additionally, heuristic and semantic features were studied. For a description of the features and results, see Blatz et al. (2003, 2004).

Following the work of the summer workshop team, Quirk (2004) presented an investigation of different approaches to sentence-level confidence estimation. A set of features is computed for each sentence generated by an MT system, and these features are combined using several different methods: modified linear regression, neural nets, support vector machines, and decision trees. Many of the sentence features are similar to

those presented in Blatz et al. (2003); the others are specific to the underlying MT system that generated the translations. Quirk (2004) also investigated the use of manually tagged data for training the confidence measures. The author found that using a small amount of manually labeled training data yields better performance than using large quantities of automatically labeled data.

Akiba et al. (2004) reported the application of confidence measures to the selection of output on N-best lists produced by different MT systems. Word-level confidence measures, namely the rank-weighted sum as described in Section 4.1 (and first introduced in Ueffing, Macherey and Ney [2003]), are used to discard low-quality system output before selecting a translation from the various MT systems.

Zens and Ney (2006) presented an extension of the word posterior probabilities presented in this article: Posterior probabilities are calculated not only on the word level, but also for n-grams, and are successfully applied to the rescoring of MT hypotheses.

3.2 Word Posterior Probabilities

The confidence of a target word can be expressed by its posterior probability, that is, the probability of the word occurring in the target sentence, given the source sentence. Word posterior probabilities are the basis of all approaches to confidence estimation presented here. The following explains how they can be determined. The different methods can be classified into two categories: **system-based** methods, which make use of system output such as word graphs or N-best lists; and **direct** methods, which use external knowledge sources such as statistical word or phrase lexica.

The system-based approaches derive the word posterior probability from the sentence posterior. The posterior probability of a sentence e_1^I can be approximated by the joint probability $p(f_1^J, e_1^I)$, which the statistical machine translation system assigns to a generated translation. The sentence probabilities employed in the search (see Equation (1)) are not normalized, which does not affect the result of the search. But for use in confidence estimation, they need to be normalized in order to obtain a probability distribution over all target sentences (see Equation (6)). From the sentence posterior probabilities, the word posterior probabilities can be calculated by summing up the probabilities of all sentences containing the target word. For an exact quantification of word posterior probabilities, we need to consider the following problem: How can we define a criterion for the occurrence of a word in a sentence? The answer to this question is not at all trivial. Due to ambiguities, the word position in the sentence is not fixed. Sentences can have different numbers of words because of deletions and insertions. Additionally, the words can be reordered in different ways during the translation process. The posterior probability of a target word e can depend on its occurrence in position i of the target sentence, for example, or on the number of times the word is contained in the sentence. Thus, several different definitions of posterior probabilities will be introduced and investigated in the following discussion. The basic concept of calculating the posterior probability will be explained for the target word e occurring in a fixed position i of the sentence. This is a rather strict and simple criterion; it will be used here mainly to illustrate the idea. Section 4 will describe several different concepts of word posterior probabilities that relax this condition.

Let $p(f_1^J, e_1^I)$ be the joint probability of source sentence f_1^J and target sentence e_1^I. Here, this is approximated by the probability that an SMT system assigns to the sentence pair (see Section 2). The word posterior probability of e occurring in position

i is calculated as the normalized sum of probabilities of all sentences containing e in exactly this position:

$$p_i(e\,|f_1^J) = \frac{p_i(e,f_1^J)}{\sum_{e'} p_i(e',f_1^J)} \tag{4}$$

$$\text{where} \qquad p_i(e,f_1^J) = \sum_{I,e_1^I} \delta(e_i,e) \cdot p(f_1^J,e_1^I) \tag{5}$$

Here $\delta(\cdot,\cdot)$ is the Kronecker function. The normalization term in Equation (4) is

$$\sum_{e'} p_i(e',f_1^J) = \sum_{I,e_1^I} p(f_1^J,e_1^I) \;=\; p(f_1^J) \tag{6}$$

This definition of word posterior probabilities raises the question of how to calculate the sums over the target sentences in Equations (5) and (6). This problem can be solved by approximating the summation space via a word graph or an N-best list. The summation is then performed explicitly over all sentences given in this restricted space. In the case of an N-best list, this is straightforward because the sentences are already listed. On a word graph, the forward–backward algorithm can be applied to carry out the summation efficiently. In these system-based approaches, the calculation depends on the output of the SMT system that generated the translations. The sentence probabilities summed in Equations (5) and (6) are the scores assigned by the underlying SMT system. The summation space is restricted to those hypotheses that are assigned a high probability by the SMT system, and the others are not considered.

The second approach to the calculation of word posterior probabilities is summation using direct models such as IBM model 1 or a phrase-based translation model. These methods do not consider the whole target sentence. The summation of probabilities is carried out over single words or phrases without context. These model-based word posterior probabilities are independent of the system generating the translations. They do not require the MT system to assign a probability to the translation hypothesis. Thus, they can also be used for confidence estimation on hypotheses from a non-statistical MT system or if only the single best translations without any scores are given.

3.3 Word Confidence Measures

The idea behind word-level confidence estimation is to be able to detect possible errors in the output of a machine translation system. Using confidence measures, individual words can be labeled as either correct or incorrect. This additional information can be used in, for example, interactive TransType-style machine translation systems (Gandrabur and Foster 2003; Ueffing and Ney 2005a).

Two problems have to be solved in order to compute confidence measures. First, suitable confidence features have to be computed. Second, a binary classifier has to be defined, which decides whether a word is correct or not. The word posterior probabilities introduced in Section 3.2 can be interpreted as the probability of a word being correct. That is, the probability can directly be used as a confidence measure. For this

purpose, it is compared to a threshold t. All words that have confidence above this threshold are tagged as correct, and all others are tagged as incorrect, translations. Thus, the binary classifier is defined as

$$\text{class}(e) = \begin{cases} \text{correct} & \text{if } p(e\,|\,f_1^J) \geq t \\ \text{incorrect} & \text{otherwise} \end{cases} \tag{7}$$

The threshold t is optimized on a distinct development set beforehand.

The question of how the correctness of a word in MT output is determined is not at all an easy one. We will address this issue in Section 7.2.

4. System-Based Confidence Measures

In this section, we will present confidence measures that are calculated over N-best lists or word graphs generated by an SMT system. Several different models for the occurrence of a target word in a sentence will be defined and experimentally evaluated. These are the models that proved most promising from a theoretical viewpoint and in the experimental evaluation:

- Target word e occurs in position i of the target sentence (see Section 4.1). The calculation of word posterior probabilities over word graphs and N-best lists is explained in detail for this concept.
- The word is considered if it occurs in a window around the position: $i \pm t,\ t \in \mathbb{N}$, for some position i (see Section 4.2).
- The Levenshtein alignments between the hypothesis under consideration and all other possible translations are determined. The target word e (in some position i) is taken into account if it is Levenshtein-aligned to itself (see Section 4.3).
- e is contained in the sentence at least n times, $n \in \mathbb{N}$ (see Section 4.4).

Section 4.5 will treat the issue of scaling the probabilities that the SMT system assigns to the translation hypothesis.

4.1 Approach Based on the Fixed Target Position

In this approach, the word posterior probability is determined for word e occurring in target position i as shown in Equation (4). This variant requires the word to occur exactly in the given position i. Hence, a probability distribution over the pairs (e, i) of target words e and positions i in the target sentence is obtained. This type of word posterior probability was first introduced in Ueffing, Macherey, and Ney (2003).

The concept of word posterior probabilities based on the fixed target position allows for easy calculation over word graphs and N-best lists. However, this concept is rather restrictive. In practice, the target position of a word varies between different translation alternatives. The method presented here is a starting point for more flexible approaches that perform summation over a window of target positions.

In the following, we will show how the word posterior probabilities based on fixed target positions are calculated over word graphs and over N-best lists.

Calculation using word graphs. A word graph represents the most promising hypotheses generated by the translation system (Ueffing, Och, and Ney 2002; Zens and Ney 2005). It has the advantage of being a compact representation of the translation hypothesis space, which allows for efficient calculation of word posterior probabilities. A word graph is a directed acyclic graph $G = (V, E)$ with vertex set V and edge set E. It has one designated root node $n_0 \in V$, representing the beginning of the sentence. Each path through the word graph represents a translation candidate. The nodes of the graph contain information such as the set of covered source positions and the language model history. Two hypotheses can be recombined if their information is identical. Recombination is carried out during decoding to accelerate the search process. If two hypotheses represent the same information with respect to translation and language models, they will be assigned the same probabilities by these models in the future. Therefore, the outcome of the search is not altered, but the processing time can be significantly decreased if only the more promising of the two hypotheses is considered for further expansion. If no recombination were carried out, the word graph would have the structure of a tree.

The edges in the word graph are annotated with target words. Additionally, they contain weights representing the part of the probability that is assigned to each particular word as part of the target hypothesis. When multiplying the scores along a path, the probability of the corresponding hypothesis is obtained. The sentence position of a word refers to the path length in the word graph: Consider an edge (n, n') that is annotated with word e. If a path leading from source node n_0 into n has i edges, then e will be the $(i + 1)$-th word in the corresponding sentence. Note that due to recombination this position is not unambiguous. If two hypotheses of different lengths i and i' are recombined in node n, then e will be in position $i + 1$ in the one resulting sentence, and in $i' + 1$ in the other sentence.

For an example of a word graph, see Figure 1. The source sentence is *Wir können das machen!*, and the reference translation is *We can do that!*. The leftmost node is the root node n_0. The other nodes represent different states with respect to the set of covered source positions and language model history. In this example, a trigram

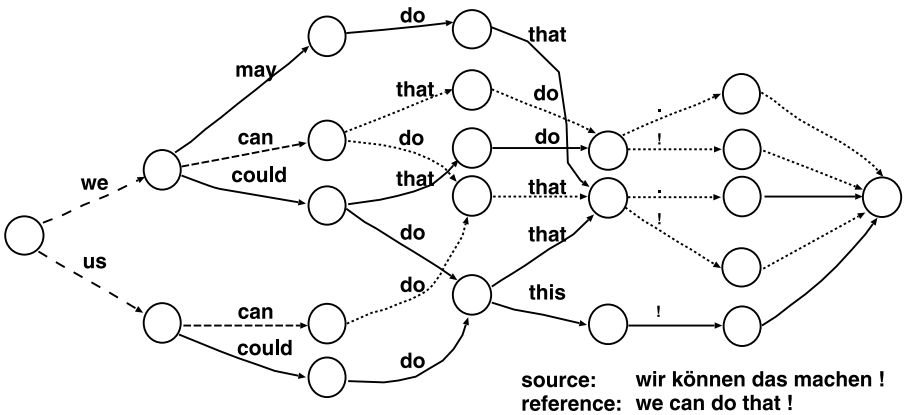

Figure 1
Example of forward–backward calculation on a word graph. The posterior probability of *can* in the second position is obtained by multiplying the total probability of all incoming paths (dashed lines) and outgoing paths (dotted), separately for the two edges, and summing the products.

language model is applied, that is, all paths leading into a node share the last two words. The translation alternatives contained in this word graph represent different reorderings of the words in the sentence: The monotone translation *that do* as well as the correctly reordered sequence *do that* occur. Note that in order to limit the size of the graph and keep the presentation simple, an example was chosen where all target sentences have the same length.

The posterior probabilities of word e in position i can be computed by summing up the probabilities of all paths in the graph that contain an edge annotated with e in position i of the target sentence. This summation is performed efficiently using the forward–backward algorithm (Jurafsky and Martin 2000). This algorithm also determines the total probability mass that is needed for normalization, as shown in Equation (6). In the following, we will present the exact equations for a word graph generated by the phrase-based translation system described in Section 2.2. In such a word graph, the first word of a target phrase is assigned the score for the whole phrase. That is, when translating a source phrase $\tilde{f}_k$ by a target phrase $\tilde{e}_k = e_{i_{k-1}+1} \ldots e_{i_k}$, the full contribution of all sub-models for this phrase is included for the first word $e_{i_{k-1}+1}$. All following words $e_{i_{k-1}+2} \ldots e_{i_k}$ are assigned probability 1.

The forward–backward algorithm works as follows: Let $Q_{PM}(\tilde{f}_k, \tilde{e}_k)$ be the phrase model score of a phrase pair as defined in Equation (3) in Section 2.2. In order to keep the notation simple, we assume a bigram language model. The extension to higher-order language models is straightforward. The **forward probability** $\Phi_i(e_i\,;\,\tilde{e}_k, \tilde{f}_k)$ of word e_i is the probability of reaching word e_i from the sentence start, where e_i occurs in position i of the sentence. It depends on the phrase pair $(\tilde{f}_k, \tilde{e}_k)$ for which $e_i \in \tilde{e}_k$. Because the full score of this phrase pair is included at the first word $e_{i_{k-1}+1}$, two cases have to be distinguished in the calculation: Either e_i is the first word in the phrase, that is, $i = i_{k-1} + 1$, or $i_{k-1} + 1 < i \leq i_k$. The forward probability can be determined by summing the probabilities of all partial hypotheses of length $i - 1$. This allows for recursive calculation in ascending order of i. We obtain the following formula:

$$\Phi_i(e_i\,;\,\tilde{e}_k, \tilde{f}_k) = \begin{cases} Q_{PM}(\tilde{f}_k, \tilde{e}_k) \cdot \prod_{i'=i+1}^{i_k} c_1 \cdot p(e_{i'} \,|\, e_{i'-1})^{\lambda_1} \cdot \sum_{\tilde{e}_{k-1}} p(e_i \,|\, e_{i_{k-1}})^{\lambda_1} \\ \qquad \cdot \sum_{\tilde{f}_{k-1}} \Phi_{i-1}(e_{i_{k-1}}\,;\,\tilde{e}_{k-1}, \tilde{f}_{k-1}) & \text{if } i = i_{k-1} + 1 \\ \Phi_{i-1}(e_{i-1}\,;\,\tilde{e}_k, \tilde{f}_k) & \text{if } i_{k-1} + 1 < i \leq i_k \end{cases}$$

The **backward probability** $\Psi_i(e_i\,;\,\tilde{e}_k, \tilde{f}_k)$ expresses the probability of completing a sentence from the current word on. It can be determined recursively in descending order of i. Again, we distinguish two cases:

$$\Psi_i(e_i\,;\,\tilde{e}_k, \tilde{f}_k) = \begin{cases} \sum_{\tilde{e}_{k+1}} \prod_{i'=i}^{i_{k+1}} c_1 \cdot p(e_{i'} \,|\, e_{i'-1})^{\lambda_1} \cdot \sum_{\tilde{f}_{k+1}} Q_{PM}(\tilde{f}_{k+1}, \tilde{e}_{k+1}) \cdot \Psi_{i+1}(e_{i+1}\,;\,\tilde{e}_{k+1}, \tilde{f}_{k+1}) \\ \qquad \text{if } i = i_k \\ \Psi_{i+1}(e_{i+1}\,;\,\tilde{e}_k, \tilde{f}_k) & \text{if } i_{k-1} < i < i_k \end{cases}$$

Using the forward–backward algorithm, the word posterior probability of word e in position i is determined by combining the forward and backward probabilities of all hypotheses containing e in this position. We carry out a summation over all corresponding phrase pairs $(\tilde{f}_k, \tilde{e}_k)$. This yields

$$p_i(e, f_1^J) = \sum_{\tilde{e}_k} \sum_{\tilde{f}_k} \Phi_i(e\,;\, \tilde{e}_k, \tilde{f}_k) \cdot \Psi_i(e\,;\, \tilde{e}_k, \tilde{f}_k) \tag{8}$$

To obtain a posterior probability, a normalization (as shown in Equation (4)) has to be performed. The normalization term $p(f_1^J) := \sum_{e'} p_i(e', f_1^J)$ corresponds to the probability mass contained in the word graph and can be calculated by summing the backward probabilities of all words that occur in the first sentence position:

$$p(f_1^J) = \sum_{\tilde{e}_1 = e_1 \ldots e_{i_1}} \sum_{\tilde{f}_1} \Psi_1(e_1; \tilde{e}_1, \tilde{f}_1)$$

Figure 1 illustrates the forward–backward algorithm. Assume the word posterior probability of the word *can* appearing in the second position of the target sentence is to be calculated. There are two edges in the graph that contain this word in the desired target position. Thus, the probabilities of the paths leading through these edges have to be summed. The forward probabilities are the probabilities of the incoming edges, shown by dashed lines. The backward probabilities are those of the paths marked by dotted lines. They are combined (separately for each edge) and then summed to obtain the word posterior probability of *can* in position 2.

Calculation using N*-best lists.* An N-best list contains the n most promising translation hypotheses generated by the statistical machine translation system. The N-best list is extracted from a word graph. The hypotheses are sorted by their probability in descending order. This representation allows for easy computation of the sum given in Equation (5). Furthermore, the calculation of more complex variants of word posterior probabilities, such as the approach based on Levenshtein alignment (see Section 4.3), is feasible.

Let $e_{n,1}^{n,I_n}$, $n = 1, \ldots, N$, be the target hypotheses in the N-best list. The word posterior probabilities presented in Equation (4) are calculated by summing the sentence probabilities of all sentences containing target word e in target position i. The sentence probability $p(f_1^J, e_{n,1}^{n,I_n})$ is given in the N-best list. The word posterior probability is then determined as

$$p_i(e \,|\, f_1^J) = \frac{\sum_{n=1}^{N} \delta(e_{n,i}, e) \cdot p(f_1^J, e_{n,1}^{n,I_n})}{\sum_{e'} \sum_{n=1}^{N} \delta(e_{n,i}, e') \cdot p(f_1^J, e_{n,1}^{n,I_n})}$$

The normalization term in the denominator equals the probability mass contained in the N-best list.

Instead of the sum of probabilities, one can also determine the relative frequency or the rank-weighted frequency of a word as follows: The relative frequency of e occurring in target position i in the N-best list is computed as

$$h_i(e\,|\,f_1^J) := \frac{1}{N}\sum_{n=1}^{N}\delta(e_{n,i},e) \tag{9}$$

The rank-weighted frequency is determined as

$$r_i(e\,|\,f_1^J) := \frac{2}{N(N+1)}\sum_{n=1}^{N}\delta(e_{n,i},e)\cdot(N+1-n) \tag{10}$$

Here, the inverted ranks $N+1-n$ are summed up because an occurrence of the word in a hypothesis near the top of the list will score better than one in the lower ranks. This value is normalized by the sum of all ranks in the list. Note that the values in Equations (9) and (10) could also be calculated over N-best lists that do not contain the sentence probability.

4.2 Approach Based on a Window over Target Positions

One way of accounting for slight variations in the target position i of word e is the introduction of a window $i \pm t$, $t \in \mathbb{N}$, around position i. The word confidence is determined as the sum of the word posterior probabilities calculated for the positions within this window. This leads to

$$p_{i,t}(e\,|\,f_1^J) = \sum_{k=i-t}^{i+t} p_k(e\,|\,f_1^J) \tag{11}$$

The window can easily be integrated both into the N-best list and the word graph–based implementation: The target position-dependent word posterior probabilities are calculated as stated in Equation (4), and the summation over the positions in the window is performed in an additional step.

4.3 Approach Based on the Levenshtein Alignment

Another way of accounting for variations in the target position of a word is to perform the Levenshtein alignment (Levenshtein 1966) between sentence e_1^I under consideration and the other possible target sentences. The summation in Equation (5) is then performed over all sentences containing e in a position Levenshtein-aligned to i (Ueffing, Macherey, and Ney 2003).

The implementation of this summation over N-best lists is straightforward: The Levenshtein alignment is performed between the hypothesis e_1^I and every sentence $e_{n,1}^{n,I_n}$ contained in the N-best list individually, and then the summation is carried out. For word graphs, no efficient way of determining the Levenshtein alignments and the resulting word posterior probabilities is known.

Let $\mathcal{L}(e_1^I, e_{n,1}^{n,I_n})$ be the Levenshtein alignment between sentences e_1^I and $e_{n,1}^{n,I_n}$, and $\mathcal{L}_i(e_1^I, e_{n,1}^{n,I_n})$ that of word e in position i in e_1^I. Consider the following example: Calculating the Levenshtein alignment between the sentences e_1^I ="A B C D E" and $e_{n,1}^{n,I_n}$ ="B C G E F" yields

$$\mathcal{L}(e_1^I, e_{n,1}^{n,I_n}) = \text{" } - \text{ B C G E"}$$

Using this representation, the word posterior probability of word e occurring in a position Levenshtein-aligned to i is given by

$$p_{\text{lev}}(e\,|\,f_1^J, e_1^I, \mathcal{L}) = \frac{p_{\text{lev}}(e, f_1^J, e_1^I, \mathcal{L})}{\sum_{e'} p_{\text{lev}}(e', f_1^J, e_1^I, \mathcal{L})} \tag{12}$$

$$\text{where} \qquad p_{\text{lev}}(e, f_1^J, e_1^I, \mathcal{L}) = \sum_{n=1}^{N} \delta(e, \mathcal{L}_i(e_1^I, e_{n,1}^{n,I_n})) \cdot p(f_1^J, e_{n,1}^{n,I_n})$$

The probability depends on all target words in the hypothesis e_1^I under consideration, because the Levenshtein alignment of the whole sentence, $\mathcal{L}(e_1^I, e_{n,1}^{n,I_n})$, is determined. This concept of word posterior probabilities is inspired by the error measure Word Error Rate (WER). It can be shown that the word posterior probabilities form a part of the Bayes risk for WER: Formulating the loss function and deriving the risk yields a minimization criterion consisting of the word posterior probabilities defined previously, one term representing the sentence length, and one for the deletion operations in the Levenshtein alignment. For more details, see Ueffing and Ney (2004) and Ueffing (2006).

4.4 Count-Based Approach

Inspired by Bayes risk for Position-independent Word Error Rate (PER), the word posterior probability can be defined by taking the counts of the words in the generated sentence into account (Ueffing and Ney 2004). The probability of target word e occurring in the sentence n times is determined as

$$p_e(n|f_1^J) = \frac{p_e(n, f_1^J)}{\sum_{n'=0}^{n_{\max}} p_e(n', f_1^J)} \tag{13}$$

$$\text{where} \qquad p_e(n, f_1^J) = \sum_{I, e_1^I} \delta(n_e, n) \cdot p(f_1^J, e_1^I)$$

Here, n_e is the count of word e in sentence e_1^I, and $n_{\max}$ is the maximal count that is observed. This term does not depend on the actual word sequence, but only on the counts of the target words. Let n_1^E be the counts of all target words $1, \ldots, E$ in sentence e_1^I. Analogously, $\tilde{n}_1^E$ denotes the count sequence for sentence $\tilde{e}_1^{\tilde{I}}$. In practice, many of these counts will be zero, of course. The posterior probabilities can then be expressed by the distribution over the count sequences:

$$p_e(n, f_1^J) = \sum_{n_1^E} \delta(n_e, n) \cdot p(n_1^E, f_1^J)$$

where the distribution over the count sequences is determined by summing up the probabilities of all sentences with these counts:

$$p(n_1^E, f_1^J) = \sum_{\tilde{I}, \tilde{e}_1^{\tilde{I}}} \delta(\tilde{n}_1^E, n_1^E) \cdot p(f_1^J, e_1^I)$$

Using this concept, the target position of the word is not taken into account, but the first occurrence of a word in the sentence will obtain a word posterior probability different from that of the second occurrence.

In Ueffing (2006), it is shown that the posterior risk for PER comprises one term related to the count-based word posterior probabilities defined here and one term related to the posterior probability of the sentence length. We can thus expect the count-based word posterior probabilities to perform especially well if the word correctness is defined on the basis of PER. The experimental results presented in Section 7.4 will confirm this assumption.

The summation in Equation (13) can be performed over N-best lists (analogously to the word posterior probability variants described so far), but it cannot efficiently be determined over the word graph. The problem is that the number of occurrences of a word on the whole path is needed. Because the word graph stores only local information, this count cannot be determined efficiently. The normalization term in Equation (13) corresponds to the total probability mass contained in the N-best list, because the case $n' = 0$ is also included.

4.5 Scaling the Probabilities

During the translation process, the different sub-models (such as the language model and the lexicon model) are weighted differently. These weights or scaling factors can be optimized with respect to some evaluation criterion (Och 2003). Nevertheless, this optimization determines only the relation between the different models, and not the absolute values of the scaling factors. The absolute values are not needed for the translation process, because the search is performed using the maximum approximation (see Equations (1) and (2)). In contrast to this, the actual values of the weights make a difference for confidence estimation, because the summation over the sentence probabilities is performed. To account for this and to find the optimal values of the scaling factors, a global weight Λ is introduced, which scales the sentence probability. The word posterior probability based on the fixed position i, for example, is then calculated according to

$$p_i(e \,|\, f_1^J) = \frac{\sum_{I, e_1^I} \delta(e_i, e) \cdot p^\Lambda(f_1^J, e_1^I)}{\sum_{e'} \sum_{I, e_1^I} \delta(e_i, e') \cdot p^\Lambda(f_1^J, e_1^I)} \tag{14}$$

When determining the system-based word posterior probabilities, this scaling factor is optimized with respect to some metric for confidence estimation on a development set distinct from the test set.

5. Confidence Measures Based on Direct Models

In the following, confidence measures based on direct models will be described. These approaches model the word posterior probability directly instead of summing the probabilities of sentences containing the target word. Confidence measures based on IBM model 1 and phrase-based translation models were developed and will be presented here. They make use of knowledge sources such as statistical word or phrase lexica for estimating the word confidence. Unlike the system-based word posterior probabilities presented so far, these confidence measures are completely independent of the target sentence position in which the word e occurs. They determine the confidence of e being contained anywhere in the sentence.

5.1 Direct Approach to Confidence Estimation Using Phrases

The statistical models presented in Section 2.2 can be used to estimate the confidence of target words as first described in Ueffing and Ney (2005b). In contrast to the approaches presented in Section 4, the direct phrase-based confidence measures do not use the context information at the sentence level, but only at the phrase level.

For a given source sentence f_1^J and a target word e, we want to determine a sort of marginal probability $Q(e,f_1^J)$. Therefore, we extract all source phrases f_j^{j+s} that occur in the given source sentence f_1^J. For these source phrases, we find the possible translations e_i^{i+t} in the bilingual phrase lexicon. The confidence of target word e is then calculated by summing over all phrase pairs (f_j^{j+s}, e_i^{i+t}) where the target part e_i^{i+t} contains e.

Let $Q_{PM}(\tilde{f}_k, \tilde{e}_k)$ be the phrase model score of a phrase pair as defined in Equation (3) in Section 2.2. Analogously, we define $Q_{LM}(e_i^{i+t})$ as the language model score of the target phrase together with the word penalty c_1 for each word in the phrase, that is,

$$Q_{LM}(e_i^{i+t}) := \prod_{i'=i}^{i+t} c_1 \cdot p(e_{i'} \mid e_{i'-2}^{i'-1})^{\lambda_1} \tag{15}$$

Note that this is the within-phrase language model probability, which does not include the context of the phrase. The language model probability at the phrase boundary is approximated by a unigram and bigram.

The (unnormalized) confidence of target word e is then determined by summing the product of the language model and the phrase model score of all phrase pairs containing e:

$$Q(e,f_1^J) := \sum_{j=1}^{J} \sum_{s=0}^{\min\{s_{\max},J-j\}} \sum_{e_i^{i+t}} \delta(e\,,\, e_i^{i+t}) \cdot Q_{LM}(e_i^{i+t}) \cdot Q_{PM}(f_j^{j+s}, e_i^{i+t}) \tag{16}$$

where $s \leq s_{\max}$ and t are source and target phrase lengths, $s_{\max}$ being the maximal source phrase length. $\delta(e, e_i^{i+t})$ denotes an extension of the Kronecker delta:

$$\delta(a,A) = \begin{cases} 1 \text{ if } a \in A \\ 0 \text{ otherwise} \end{cases}$$

The value calculated in Equation (16) is not normalized. In order to obtain a probability, this value is divided by the sum over the (unnormalized) confidence values of all target words:

$$p_{phr}(e\,|f_1^J) = \frac{Q(e,f_1^J)}{\sum_{e'} Q(e',f_1^J)} \tag{17}$$

As shown in Equations (3) and (15), the different sub-models of the phrase-based translation approach are combined in a log-linear manner. The weights $\lambda_1,\ldots,\lambda_5$ and the penalties c_1, c_2 are optimized in the translation process with respect to some evaluation criterion such as WER or BLEU. This is done using the Downhill Simplex algorithm (Press et al. 2002). The resulting values of the weights express the relation between the sub-models, but not their absolute values. They are usually normalized so that they sum to 1. For use in confidence estimation, two different aspects thus have to be considered:

- The relation of the sub-models that is optimal for translation quality is not necessarily optimal for classification performance. Therefore, the sub-model scaling factors are optimized with respect to some confidence evaluation measure (see Section 7.3). The direct phrase-based confidence measures provide a framework for optimizing the sub-model weights efficiently. The optimization is performed analogously to the procedure for machine translation: The confidence values are determined for all words in the development corpus. Then, classification is carried out as described in Section 3.3, and the result is evaluated. The weights are then modified and the confidence estimation is repeated, until optimal classification performance on the development set is achieved. Again, the Downhill Simplex algorithm is used for optimization.
- For MT, only the relation between the different sub-models, but not the actual values of the scaling factors, are important. Confidence measures, however, also depend on these actual values. In MT, the sub-model scaling factors are normalized such that they sum to 1. For the use in confidence estimation, the value of this sum,

  $$\Lambda := \sum_{i=1}^{5} \lambda_i + c_1 + c_2$$

  is also optimized. This Λ is analogous to the global scaling factor for the system-based confidence measures introduced in Section 4.5.

5.2 Confidence Measure Based on IBM Model 1

Another type of confidence measure that does not rely on system output and is thus applicable to any kind of machine translation system is the IBM model 1–based confidence measure that was introduced in Blatz et al. (2003). We modified this confidence measure because we found that the average lexicon probability used there is dominated

by the maximum. Therefore, we determine the *maximal* translation probability of the target word e over the source sentence words:

$$p_{\text{ibm1}}(e|f_1^J) = \max_{j=0,\ldots,J} p(e|f_j) \tag{18}$$

where f_0 is the "empty" source word (Brown et al. 1993). The probabilities $p(e|f_j)$ are word-based lexicon probabilities.

Investigations of the use of the IBM model 1 for word-level confidence estimation showed promising results (Blatz et al. 2003, 2004). Thus, we apply this method here and compare it to the other types of confidence measures. Ueffing and Ney (2005a) report on the use of this IBM model 1–based confidence measure in a TransType-style interactive MT system. The work presented there shows that even this relatively simple confidence measure yields a significant gain in the quality of the predictions proposed by the interactive system.

6. Combination of Confidence Measures

In related work in MT as well as in speech recognition, the combination of numerous confidence features has been suggested (Gandrabur and Foster 2003; Blatz et al. 2004; Quirk 2004; Sanchis 2004). Among the methods used for combination are multi-layer artificial neural networks, naive Bayes classifiers, and modified linear regression.

Because the combination of several confidence measures proved successful, the different word posterior probabilities proposed here were combined with each other. The combination was performed in a log-linear manner. Let $p_m(e\,|f_1^J, \ldots)$, $m = 1, \ldots, M$, be the word posterior probabilities of e determined using different approaches. The word confidence resulting from their combination is calculated as

$$c(e) = \exp\left\{ -\sum_{m=1}^{M} \lambda_m \cdot \log p_m(e\,|f_1^J, \ldots) \right\}$$

The interpolation weights λ_m are optimized with respect to some confidence evaluation metric on the development corpus using the Downhill Simplex algorithm (Press et al. 2002). With this approach, the confidence error rates were reduced over the best single confidence measure consistently on all corpora we examined. The experimental results will be presented in Section 7.4. This section also contains details on which confidence measures were combined.

However, the focus of this work is on word posterior probabilities as stand-alone confidence measures. It was shown that they are the best single features for confidence estimation (Blatz et al. 2004). Moreover, they are closely related to Bayes risk, which yields a sound theoretical foundation (Ueffing and Ney 2004).

7. Experiments

7.1 Experimental Setting

The experiments were performed on three translation tasks in different language pairs. The corpora were compiled in the EU projects TransType2 (TransType2 2005) and TC-STAR (TC-STAR 2005), and for the NIST MT evaluation campaign (NIST 2004). The

TransType2 corpora consist of technical manuals for Xerox devices such as printers. They are available in three different language pairs. This domain is very specialized with respect to terminology and style. The corpus statistics are given in Table 1. The TC-STAR corpus consists of proceedings of the European Parliament. It is a spoken language translation corpus containing the verbatim transcriptions of the speeches in the European Parliament Plenary Sessions (EPPS). The domain is basically unrestricted because a wide range of different topics is covered in the sessions. The translation direction is from Spanish into English. For corpus statistics, see Table 2. The NIST corpus was compiled for the yearly MT evaluation campaign carried out since 2001. Chinese news articles are translated into English. Similarly to the EPPS data, the domain is basically unrestricted, because a wide range of different topics is covered. However, the vocabulary size and the training corpus are much larger than in the EPPS collection, as the corpus statistics presented in Table 3 show. Additionally to the bilingual data, a monolingual English corpus consisting of 636M running words was used for language model training. The SMT systems that generated the translations for which confidence estimation was performed were trained on these corpora. The same holds for the probability models that were used to estimate the word confidences.

We translated the development and test corpora using several different MT systems for testing the confidence measures:

- The phrase-based translation system described in Section 2.2 (denoted as PBT in the tables); a large part of the results will be presented for output of this system.

Table 1
Statistics of the training, development, and test corpora for the TransType2 task.

		French	English	Spanish	English	German	English
TRAIN	Sentences	53,046		55,761		49,376	
	Running words	680,796	628,329	752,606	665,399	537,464	589,531
	Vocabulary	15,632	13,816	11,050	7,956	23,845	13,223
DEV	Sentences	994		1,012		964	
	Running words	11,674	10,903	15,957	14,278	10,462	10,642
TEST	Sentences	984		1,125		996	
	Running words	11,709	11,177	10,106	8,370	11,704	12,298

Table 2
Statistics of the training, development, and test corpora for the TC-STAR EPPS Spanish–English task. Both development and test corpus are provided with two English references.

		Spanish	English
TRAIN	Sentences	1,652,174	
	Running words	32,554,077	31,147,901
	Vocabulary	124,192	80,125
DEV	Sentences	2,643	
	Running words	20,289	40,396
TEST	Sentences	1,073	
	Running words	18,896	37,742

Table 3
Statistics of the training, development, and test corpora for the NIST Chinese–English task. Both development and test corpus are provided with four English references.

		Chinese	English
TRAIN	Sentences	7M	
	Running words	199M	213M
	Vocabulary	223K	351K
	Dictionary entries	82K	
DEV (2002 evaluation set)	Sentences	878	
	Running words	25K	105K
TEST (2004 evaluation set)	Sentences	1,788	
	Running words	52K	239K

- The alignment template system (Och and Ney 2004) (denoted as AT in the tables), which is also a state-of-the art phrase-based translation system.
- The Systran version available at `http://babelfish.altavista.com/tr` in June 2005. These hypotheses were used to investigate whether the direct confidence measures perform well on translations generated by a structurally different system.

The translation quality on the TransType2 task in terms of WER, PER, BLEU score (Papineni et al. 2002), and NIST score (NIST 2002) is given in Table 4. We see that the best results are obtained on Spanish to English translation, followed by French to English and German to English. The reason that Systran generates translations of much lower quality than the SMT systems is due to the fact that the technical manuals are very specific in terminology. The SMT systems were trained on similar corpora so that they are familiar with the terminology. The table additionally shows the translation quality achieved by the system PBT on the NIST test set.

Table 4
Translation quality of different MT systems on the TransType2 and the NIST test corpora.

Task	Language pair	System	WER[%]	PER[%]	BLEU[%]	NIST
TransType2	F → E	PBT	54.9	43.4	31.3	6.62
		AT	54.8	43.7	31.5	6.64
		Systran	81.5	71.7	12.5	4.23
	S → E	PBT	26.1	17.5	66.9	8.98
		AT	29.6	20.1	63.4	8.80
		Systran	78.0	62.3	23.4	4.77
	G → E	PBT	61.6	49.6	25.7	5.72
		AT	62.7	49.8	26.6	5.92
		Systran	79.2	66.4	12.0	4.09
NIST	C → E	PBT	61.8	42.9	31.1	8.47

C = Chinese; E = English; F = French; G = German; S = Spanish.

On the EPPS task from TC-STAR, the confidence measures were tested on output from the phrase-based translation system. The hypotheses are generated by the version of the system that participated in the TC-STAR evaluation round in March 2005 and that was ranked first there. The translation quality can be seen in Table 12 later in this article.

7.2 Word Error Measures

In order to evaluate the classifier built from the confidence measures as described in Section 3.3, reference tags are needed that define the true class of each word. In machine translation, it is not intuitively clear how to determine the correctness of a word. Therefore, a number of different measures for identifying the reference classes for single words in a translation hypothesis were implemented (Ueffing 2006). They are inspired by different translation evaluation measures like WER and PER. All of them compare the translation hypothesis to one or—if available—several references to determine the word errors. In this article, we will present results for the following error measures:

- **WER**: A word is counted as correct if it is Levenshtein-aligned to itself in one of the references.
- **PER**: A word is tagged as correct if it occurs in one of the references. The number of occurrences per word is taken into account, but the position of the word in the sentence is completely disregarded.

Both word error measures exist in two variants: First, each translation hypothesis is compared to the pool of all references (in case there exist different reference translations for the development and test corpus). Second, the reference with minimum distance to the hypothesis according to the translation evaluation measure under consideration is determined. The true classes of the words are then defined with respect to this nearest reference. For example, if the PER metric is applied, the pooled variant labels all those words as correct that occur in any of the references (with this count). The second variant considers as correct only those words that are contained in the nearest reference (with this count). The latter corresponds to the procedure used for m-WER and m-PER in MT evaluation (Nießen et al. 2000).

Table 5 shows the percentage of words that are labeled as correct according to the different error measures on the development and test corpora of the EPPS task. It can be seen that WER is the stricter error measure: It considers fewer words as correct than PER does. A comparison of the pooled and the nearest reference shows that the pooling yields a significant increase in the number of words labeled as correct. Note that the figures in the table do not directly correspond to the translation error rates for the system output. They are calculated only for the words contained in the generated

Table 5
Ratio of correct words (%) in the EPPS Spanish → English development and test corpora, according to different word error measures.

Error Measure	WER		PER	
	pooled	nearest	pooled	nearest
DEV	78.6	72.9	81.5	77.4
TEST	76.5	69.8	81.5	76.5

translation hypotheses and do not take deleted words into account. Moreover, they are normalized by the hypothesis lengths. If WER and PER are applied as translation evaluation measures (on the sentence level), deletions are counted as well, and the number of errors is divided by the number of reference words.

7.3 Evaluation Metrics

After computing the confidence measure, each generated word is tagged as either correct or incorrect, depending on whether its confidence exceeds the tagging threshold that was optimized on the development set beforehand. The performance of the confidence measures is evaluated using the following three measures:

- **C**lassification or **C**onfidence **E**rror **R**ate (CER): This is defined as the number of incorrect tags divided by the total number of generated words in the translated sentence. The baseline CER is determined by assigning the most frequent class (in the whole development or test corpus) to all words. Assume that the correct classes of the words are defined on the basis of WER. If the overall WER on the considered development or test corpus is below 50%, the baseline CER is calculated by tagging all words as correct. The baseline CER then corresponds to 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 (to minimize CER) on a development corpus distinct from the test set. Moreover, we will present significance bounds for the baseline CER. They were determined using the bootstrap estimation method described in Bisani and Ney (2004).[1]
- **R**eceiver **O**perating **C**haracteristic (ROC) curve (Duda, Hart, and Stork 2001):[2] The ROC curve plots the **correct rejection rate** versus the **correct acceptance rate** for different values of the tagging threshold. The correct rejection rate is the number of incorrectly translated words that were tagged as false, divided by the total number of incorrectly translated words. The correct acceptance rate is the ratio of correctly translated words that were 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 (and away from the point of origin), the better the performance of the confidence measure. Unlike the CER, the ROC curve is independent of the prior probability of the two classes correct and incorrect. This means that ROC curves from different data sets can be compared directly.
- **I**ntegral of the **ROC** curve (IROC): ROC curves provide for a qualitative analysis of classifier performance; a related quantitative metric is IROC, defined as the area under a ROC curve. The IROC takes on values in $[0, 1]$, with 0.5 corresponding to a random separation of correct and incorrect words, 1.0 corresponding to a perfect separation, and 0.0 the opposite.

1 The tool described in this paper is freely available from `http://www-i6.informatik.rwth-aachen.de/web/Software/`.

2 A variant of the ROC curve is the Detection Error Tradeoff (DET) curve which plots the *false* rejection rate versus the *false* acceptance rate.

7.4 Experimental Results

TransType2 task. Table 6 compares the classification performance of several confidence measures on the TransType2 French–English task. The CER and the IROC values are given for WER- and PER-based classification. Note that lower CER and higher IROC values express better performance. It is interesting to see that, in most of the cases, the tendencies are consistent for the two evaluation metrics: Lower CER is accompanied by higher IROC.

In general, one can see that the very simple approach that sums over sentences in the N-best list or word graph considering the fixed target position of the word clearly performs worst. This is to be expected, and the method was included only for comparison. It can be considered as a simple baseline method. The other system-based measures discriminate significantly better in both settings.

The system-based confidence measures show much better discriminative power than the direct IBM model 1. The N-best list based measure with Levenshtein alignment and the word posterior probabilities calculated over word graphs using a window perform similarly well. For WER-based classification, they are outperformed only by the direct phrase–based approach, which achieves the best CER and IROC values.

It is interesting to compare the two methods that were applied to both word graphs and N-best lists: the approach based on the fixed target position and the one summing over a window of positions. In both cases, the word graph–based calculation is slightly superior to that based on 10,000-best lists. However, the difference in CER is not significant.

The count-based method working on N-best lists is clearly the best confidence measure for PER-based classification. This result was to be expected because the count-based word posterior probability was derived from the Bayes risk for PER (Ueffing and Ney 2004). Even if its CER does not differ much from that of the direct phrase-based measure, there exists a clear predominance in terms of IROC. The IBM-1–based confidence measure performs rather poorly compared to the other methods. This is not surprising because the IBM model 1 is a very simple model.

Table 6
Classification performance in terms of CER (%) and IROC (%) for different confidence measures. TransType2 French → English test set. References based on WER and PER, confidence measures optimized accordingly. Hypotheses from the phrase-based system.

Model	WER		PER	
	CER	IROC	CER	IROC
baseline	42.2	–	34.2	–
99% confidence interval	±2.3	–	±2.0	–
10,000-best lists, fixed position	39.7	66.2	33.3	66.2
Levenshtein	31.3	72.6	28.1	74.8
window ±3	31.6	70.7	28.3	73.4
count-based	31.9	71.6	**27.0**	**76.5**
word graphs, fixed position	38.6	70.5	33.1	67.6
window ±3	31.1	72.4	27.3	75.4
IBM-1 (max.)	39.2	67.0	31.5	71.0
direct phrase-based	**30.6**	**74.4**	27.4	73.7

The comparison of the IROC values for WER- and PER-based classification shows that PER is easier to learn than WER: The IROC values for PER are higher for most confidence measures. This is consistent with the results obtained in the CLSP workshop (Blatz et al. 2003). The classifiers investigated there also show better discriminative power for reference classes based on PER than for WER.

To further illustrate the classification performance of the different confidence measures, the ROC curves for some of them are given in Figure 2. In each, the diagonal line refers to random classification of words as correct and incorrect. The left curve shows the results for WER-based classification, and the right one for PER, respectively. The *N*-best list-based method considering the fixed target position is again given for comparison. One can see that the IBM-1–based confidence measure is clearly better than this baseline for PER, but not for WER. The curves for the direct phrase-based model and the best *N*-best list-based method lie relatively close to each other. These two confidence measures clearly dominate all others.

Because the direct phrase-based confidence measures perform so well on the output of the phrase-based translation system, we were interested in finding out whether this is due to the fact that the translation system and the confidence measure explore the same statistical models. Therefore, the system-independent confidence measures (i.e., those based on IBM model 1 and the direct phrase-based method) were tested on output from different machine translation systems, including Systran as a non-statistical MT system. The experimental results are shown in Table 7. They can be summarized as follows:

- In all settings, both measures distinctly decrease the CER compared to the baseline. In one case (Spanish to English, Systran), the achieved CER is as much as 60% lower than the baseline CER.
- On French to English and German to English, all improvements are significant at the 1% level. On Spanish to English, which is the language pair yielding by far the lowest baseline CER, only the phrase-based measure achieves a reduction at this level of significance.
- In all but one case, the direct phrase-based approach outperforms the IBM-1–based method significantly. This tendency is consistent for both CER and IROC. The relative difference in CER is up to 20%.

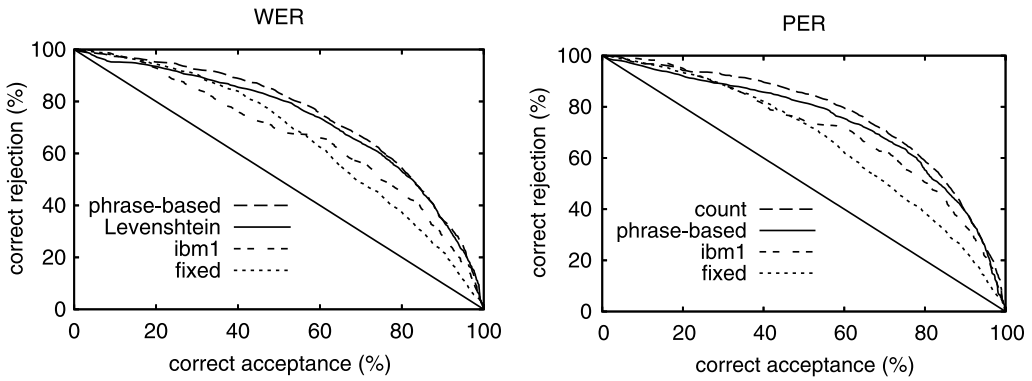

Figure 2
ROC curves for different confidence measures. TransType2 French → English test set. References based on WER (left) and PER (right). Hypotheses from the phrase-based system.

Table 7
Classification performance in terms of CER (%) and IROC (%) for different system-independent confidence measures. TransType2 test sets. Reference based on WER. Hypotheses from different MT systems.

Task	Model	AT		PBT		Systran	
		CER	IROC	CER	IROC	CER	IROC
F → E	baseline	42.5	–	42.2	–	32.8	–
	99% confidence interval	±2.3	–	±2.3	–	±1.7	–
	IBM-1 (max.)	34.1	68.3	35.6	66.9	26.0	81.3
	direct phrase-based	**30.2**	**73.0**	**30.6**	**74.4**	**22.7**	**83.2**
S → E	baseline	20.8	–	19.2	–	43.7	–
	99% confidence interval	±1.9	–	±2.0	–	±1.5	–
	90% confidence interval	±1.2	–	±1.3	–	±1.0	–
	IBM-1 (max.)	20.0	66.8	18.3	73.2	21.7	85.5
	direct phrase-based	**17.5**	**76.0**	**16.4**	**77.0**	**17.3**	**87.5**
G → E	baseline	49.2	–	48.4	–	37.4	–
	99% confidence interval	±2.2	–	±2.4	–	±1.4	–
	IBM-1 (max.)	32.7	73.3	32.8	72.2	**23.6**	80.7
	direct phrase-based	**27.6**	**79.1**	**26.4**	**80.3**	24.3	**81.4**

- On the German to English Systran hypotheses, both confidence measures discriminate similarly well. In terms of CER, the IBM model 1 is slightly better, whereas the phrase-based method achieves the highest IROC value.

EPPS task. Further experiments comparing the classification performance of the different confidence measures were carried out on the EPPS data task, which is structurally different from the Xerox task. The EPPS collection consists of speeches given in the plenary sessions of the European Parliament, translated from Spanish into English. The EPPS task is more challenging than the Xerox manuals because the domain is almost unrestricted and the translation has to cope with effects of spontaneous speech. The goal of these experiments is to find out whether the confidence measures perform equally well on this challenging task as on the Xerox task. The development and test set of the EPPS data are provided with two references each. This makes it possible to compare the two ways of handling multiple references: As explained in Section 7.2, the true class of a word can be determined either with respect to the pooled references or to the reference with minimal distance.

Table 8 presents the CER and IROC values for different confidence measures on the EPPS task. The classification with respect to m-WER and m-PER (i.e., considering only the nearest reference) as word error measures was investigated. The confidence measures based on the fixed position were not calculated because the previous experiments showed that they perform significantly worse than the other measures. It can be seen in the table that the word posterior probabilities derived from the Bayes risk for the word error measures perform best: The Levenshtein-based confidence measure discriminates best for m-WER and the count-based approach for m-PER. They are clearly superior to all other confidence measures, especially in terms of IROC. For WER-based classification, the word graph-based method performs similarly well to the Levenshtein-based measure in terms of CER, but significantly worse if IROC is considered.

The results achieved by the direct phrase-based approach on this task are not as good as on the Xerox data. The reason for this is that the domain of the EPPS collection

Table 8
Classification performance in terms of CER (%) and IROC (%) for different confidence measures. EPPS Spanish → English test set. Reference based on m-WER and m-PER, confidence measures optimized accordingly. Hypotheses from the phrase-based system.

Model	m-WER		m-PER	
	CER	IROC	CER	IROC
baseline	30.2	–	23.5	–
99% confidence interval	±1.2	–	±1.0	–
15,000-best lists, Levenshtein	25.7	**75.4**	21.6	74.2
window ±3	26.7	69.9	21.4	71.8
count-based	27.6	71.4	**21.2**	**78.3**
word graphs, window ±3	**25.6**	72.1	21.9	73.2
IBM-1 (max.)	27.7	68.7	21.5	72.5
direct phrase-based	26.8	67.5	**21.2**	70.9

is almost unrestricted. We found in a data analysis that the phrase models do not capture the data as well as they do in the Xerox domain (Ueffing 2006). Nevertheless, for m-PER–based classification, the direct phrase-based measures achieve the same reduction in CER over the baseline as the system-based method using count information. Because the direct phrase-based confidence measures completely disregard the target position of the word, they are better suited for PER-based classification than for WER.

As is to be expected, the IBM model 1–based confidence measure performs better for reference tags defined by m-PER than for m-WER. However, it is among the methods with the worst discriminative power in both cases.

In general, the improvements over the CER baseline are not as high on these EPPS data as on the TransType2 corpora. The relative gain in CER is 15% for the best confidence measure. But because the test corpora are large—with 20,000 running words they are about twice as big as the TransType2 test sets—all achieved improvements are significant at the 1% level. The IROC values are comparable to those achieved on TransType2 data. The fact that the IROC is independent of the baseline error allows for the conclusion that the confidence measures are well-suited for this challenging translation task as well.

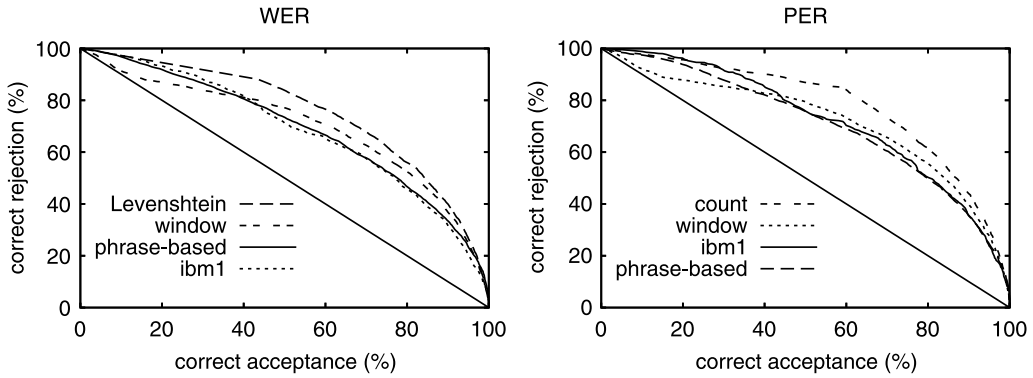

Figure 3
ROC curves for different confidence measures. EPPS Spanish → English test set. References based on m-WER (left) and m-PER (right). Hypotheses from the phrase-based system.

The ROC curves shown in Figure 3 further illustrate the classification performance of the different measures. The left curve shows the results for m-WER–based classification, and the right one for m-PER. One can see that for m-WER, the IBM-1–based and the direct phrase-based confidence measures perform very similarly. There is no clear difference between these two approaches and the one calculated over a window of target positions. The discriminative power of the direct model is higher for a lower correct acceptance ratio, whereas the system-based measure performs better for a high correct acceptance ratio. The Levenshtein-based word posterior probabilities are clearly superior to all other approaches. The ROC curve lies beyond the others over the whole range. For PER, the classifier based on word counts dominates all other confidence measures. The three other methods show relatively similar performance.

For all results presented so far, the reference tags were determined by comparing each hypothesis to the most similar reference. As mentioned in Section 7.2, it is also possible to pool the references instead. Table 9 presents an assessment of the discriminative power of different confidence measures for these reference tags. The conclusions from these results are the same as for those in Table 8: The Levenshtein-based method performs best for WER, and the count-based one for PER. All reported improvements in CER are significant at the 1% level. The IROC values for the pooled error measures are higher than for m-WER and m-PER for all confidence measures. Obviously, this method of error counting is easier to assess using confidence measures. The differences in CER are not as large here as in Table 8. However, the IROC values provide a clear indication of the differences in quality between the classifiers.

NIST task. The third translation task that was used for the evaluation of the confidence measures proposed in this article is part of the NIST MT evaluation campaign. The task here is the translation of news articles from Chinese into English. As with the EPPS data, the domain is basically unrestricted.

The experimental results are presented in Table 10. The confidence measures that perform best on the two other tasks were evaluated on the NIST data. The results support those achieved on the EPPS collection. All confidence measures reduce CER over the baseline with significance at the 1% level. For reference tags defined by m-WER, the confidence measure using Levenshtein alignment over N-best lists

Table 9
Classification performance in terms of CER (%) and IROC (%) for different confidence measures. EPPS Spanish → English test set. Reference based on pooled WER and PER, confidence measures optimized accordingly. Hypotheses from the phrase-based system.

Model	pooled WER		pooled PER	
	CER	IROC	CER	IROC
baseline	23.5	–	18.5	–
99% confidence interval	±1.1	–	±1.0	–
15,000-best lists, Levenshtein	**21.3**	**77.5**	17.0	76.4
window ±3	21.8	71.5	17.1	73.2
count-based	21.8	73.7	**16.7**	**80.6**
word graphs, window ±3	21.8	73.0	18.1	74.1
IBM-1 (max.)	21.7	70.2	16.9	74.3
direct phrase-based	**21.3**	69.5	16.8	69.3

Table 10
Classification performance in terms of CER (%) and IROC (%) for different confidence measures. NIST04 Chinese → English test set. Reference based on m-WER and m-PER, confidence measures optimized accordingly. Hypotheses from the phrase-based system.

model	m-WER		m-PER	
	CER	IROC	CER	IROC
baseline	46.2	–	32.7	–
99% confidence interval	±1.0	–	±0.6	–
10,000-best lists, Levenshtein	37.2	**67.4**	30.5	67.5
window ±3	39.2	64.4	30.5	67.0
count-based	**37.0**	66.0	28.4	**72.0**
IBM-1 (max)	42.9	58.0	31.9	59.9
direct phrase-based	37.3	66.7	**27.1**	71.8

performs best. Especially in terms of IROC, this method is clearly superior to the other confidence measures. The count-based method achieves a CER that is 0.2% lower, which is not significant. For classification with respect to m-PER, there are two methods that outperform the others: the count-based confidence measure calculated over N-best lists and the direct phrase-based approach. They achieve CER and IROC values that differ significantly from those of the other measures. However, neither of the two approaches is clearly superior to the other: The direct phrase-based confidence measure achieves a lower CER of 27.1%, whereas the count-based confidence measure calculated over N-best lists achieves a slightly higher IROC value. The confidence measure based on IBM model 1 shows by far the worst discriminative power for both m-WER- and m-PER-based classification. The CER obtained with this method is significantly higher than those of all other measures.

Combination of features. Because feature combination yields good results in the experiments reported in related work such as Blatz et al. (2003), we performed similar experiments. The confidence measures investigated here were combined log-linearly as described in Section 6. The resulting confidence measures were evaluated on all three translation tasks. The three single word posterior probabilities that perform best in each setting were used in the combination. For the confidence estimation with respect to reference tags defined by m-WER, these are:

- the system-based word posterior probabilities based on Levenshtein alignment
- the system-based word posterior probabilities performing windowing over target positions
- the direct phrase-based method

If the reference tags are determined by m-PER, the features used differ slightly, depending on the corpus. The measures that are combined are three of the following:

- the system-based word posterior probabilities based on the word count
- the system-based word posterior probabilities performing windowing over target positions

- the confidence measure based on IBM model 1
- the direct phrase-based method

The experimental results for the combined confidence measures are presented in Table 11. They show that the resulting confidence measure outperforms the best single method. The improvement in CER is up to 1.8% in absolute terms. In terms of IROC, the gain is up to 4.4 points. This is in the same range as the improvements achieved in the CLSP summer workshop (Blatz et al. 2003). However, there is one case in which the IROC decreases, namely the m-PER–based classification on EPPS Spanish to English. This can be explained by the fact that the combination was optimized with respect to CER. In order to avoid this type of inconsistency, the optimization could be performed considering a combination of CER and IROC as criterion.

8. Rescoring

8.1 Approach

This section reports on the use of word posterior probabilities for rescoring of N-best lists. The rescoring is performed as follows: For every hypothesis in the N-best list, the confidence of each word in the sentence is calculated. These word posterior probabilities are multiplied to obtain a score for the whole sentence. This sentence score is then used as an additional model for N-best list rescoring. It serves as an indicator of the overall quality of the generated hypothesis. Additionally, the minimal word posterior probability over the sentence is determined. This can be seen as an indicator of whether the hypothesis contains words that are likely to be incorrect. These new models are combined with the existing models (such as the score assigned by the underlying SMT system and additional language model scores) in a log-linear manner. The scaling factors of all models are optimized on the development corpus using the Downhill Simplex algorithm. This combination using the optimized factors is then applied and evaluated on the test set.

Table 11
Classification performance in terms of CER (%) and IROC (%) for a log-linear combination of word posterior probabilities. Test sets from all three tasks. References based on m-WER and m-PER. Hypotheses from the phrase-based system.

Reference tag		m-WER		m-PER	
Task	confidence measure	CER	IROC	CER	IROC
TransType2 F → E	baseline	42.2	–	34.2	–
	best single	30.6	74.4	27.0	76.5
	combination of 3	29.5	75.5	25.4	78.4
EPPS S → E	baseline	30.2	–	23.5	–
	best single	25.7	75.4	21.2	78.3
	combination of 3	25.7	76.1	20.1	76.9
NIST C → E	baseline	46.2	–	32.7	–
	best single	37.3	67.2	27.4	71.3
	combination of 3	35.5	68.0	25.8	75.7

8.2 Experimental Results

Rescoring was carried out on EPPS data using the direct phrase-based confidence measures. Within the project TC-STAR, an MT evaluation campaign was performed in March 2005 to compare the research systems of the consortium members (Ney et al. 2005). Different conditions concerning the input data were defined. In the following, rescoring results on the verbatim transcriptions will be presented. The translations that RWTH submitted to this evaluation were generated by the phrase-based translation system described in Section 2.2. *N*-best lists were generated for development and test corpus, with a maximum length of 20,000 and 15,000, respectively. These were then rescored with an IBM model 1, a 4-gram language model, and a deletion model based on IBM-1. The weights for all these models and for the sentence probability assigned by the SMT system were optimized with respect to BLEU score on the development corpus. For a detailed description of the system, see Vilar et al. (2005). This system was ranked first in the evaluation round according to all evaluation criteria (Ney et al. 2005).

Two different sets of rescoring experiments were performed. They differ only in their starting points: The first one starts from the baseline system without rescoring. The sub-model weights of this system were optimized with respect to BLEU on the development set, but no additional models were used for rescoring the *N*-best list. This experiment was performed to analyze the maximum improvement that can be achieved through rescoring with confidence measures. The second experiment starts from the system that has already been rescored with the three different models mentioned above. This is the system that was used in the TC-STAR evaluation campaign, and that was ranked first there. In this setting, it can be seen whether the rescoring with confidence measures manages to improve upon the *best* available system as well. Furthermore, it is possible to analyze whether the gains from all rescoring models are additive.

The results are shown in Table 12. The upper block evaluates the translation quality without considering case, and the second one contains the case-sensitive evaluation. These different figures are presented here in order to separate the effect of the translation and the true-casing process. The translation system was trained on a lower-cased corpus, and the true-casing is performed as an additional post-processing step.

Table 12
Translation quality for rescoring with confidence measures. EPPS Spanish → English test set. Optimized for BLEU.

case?	System	WER (%)	PER (%)	BLEU (%)	NIST
no	baseline	40.9	30.4	45.5	9.83
	+ direct phrase-based confidence measure	40.8	29.9	46.5	9.93
	+IBM-1+LM+deletion model	40.6	29.5	46.6	9.99
	+direct phrase-based confidence measure	**40.4**	**29.4**	**47.2**	**10.04**
yes	baseline	42.5	32.2	45.1	9.67
	+ direct phrase-based confidence measure	42.7	32.0	45.6	9.68
	+IBM-1+LM+deletion model	42.5	31.7	45.9	9.75
	+direct phrase-based confidence measure	**42.4**	**31.6**	**46.2**	**9.78**
	second best translation system	43.9	33.4	44.1	9.47

Let us first consider the case-insensitive results. The baseline is the single best output of the translation system. This system can be improved through rescoring with confidence measures by 1 BLEU point. This is only 0.1 BLEU points less than the gain achieved from rescoring with the *three* other models. The system from the second setup (rescored with IBM model 1, the language and the deletion model) improves the BLEU score by 1.1 points over the baseline. Another 0.6 BLEU points can be gained through additional rescoring with the direct phrase-based confidence measures. The improvement is consistent across all four automatic evaluation criteria. Naturally, the gain in BLEU score is higher than for the other measures, because the system was optimized with respect to BLEU.

In the TC-STAR evaluation campaign, case information was taken into account. The corresponding results are presented in the second block of the table. The overall translation quality is lower if case is considered. For all models applied here, the gain achieved through rescoring is not as big as in the case-insensitive evaluation. If only the confidence measures are used for rescoring, the BLEU score is increased by 0.5 points. The NIST score and the error measures change only slightly. However, when all four rescoring models are applied, the system is significantly improved. The models used in the TC-STAR evaluation yield an increase of 0.8 BLEU points. The word posterior probabilities add another 0.3 points to this. This change is rather small, but comparable to the contribution of each single rescoring model used in the evaluation campaign. For comparison, the translation quality of the second best system in this campaign is reported in the last row of the table. The difference in BLEU score between the RWTH system and the second best can be significantly improved through rescoring.

9. Conclusion

In this work, we set up a probabilistic framework for the computation of word posterior probabilities for machine translation. Within this framework, different concepts of word posterior probabilities were defined and analyzed. Several approaches to the calculation of word posterior probabilities were investigated and compared: system-based methods that explore information provided by the SMT system that generated the translations, and direct model-based methods that make use of statistical (translation) models.

The use of word posterior probabilities as confidence measures was studied, including their application in a rescoring scenario. The proposed confidence measures were systematically evaluated on different translation tasks and different language pairs. On all corpora, the best methods developed here reduce the confidence error rate significantly (at the 1% level). The direct confidence measures were also successfully applied to output from a non-statistical MT system.

The results of the experiments can be summarized as follows:

- The performance of the confidence measures depends heavily on the word error measure that defines the reference tags. Naturally, the word posterior probabilities derived from Bayes risk for this word error measure discriminate best. For WER, this is the approach based on the Levenshtein alignment, and for PER this is the method that considers the counts of the words.
- The direct phrase-based confidence measures perform very well on the restricted domain of the TransType2 corpora consisting of technical manuals. There, they outperform all other measures. However, this is

not the case for data from domains that are basically unrestricted, such as the EPPS and NIST corpora. There, the system-based measures discriminate better for reference tags given by WER. For PER-based confidence estimation, the direct phrase-based confidence measure and the count-based confidence measure calculated over N-best lists show the best performance.

- The confidence measures based on IBM model 1 normally perform worse than the system-based or direct phrase-based methods. The reason for this is that the IBM model 1 is a very simple model that does not consider the context of a target word at all.
- The combination of several different word posterior probabilities into one confidence measure yields better confidence estimation performance than the best single feature. However, the word posterior probabilities proposed here proved to be strong stand-alone features (see also experiments reported in Blatz et al. [2003]).
- Rescoring with confidence measures was shown to improve translation quality. The SMT system investigated here was the one that was ranked first in the TC-STAR evaluation campaign in March 2005. It was consistently improved through rescoring with confidence measures.

Acknowledgments

This work was partly funded by the European Union under the RTD project TransType2 (IST–2001–32091), and under the integrated project TC-STAR—Technology and Corpora for Speech to Speech Translation (IST-2002-FP6-506738). Nicola Ueffing would like to thank her former and current colleagues at RWTH Aachen University and the National Research Council Canada and everybody from the "CE for SMT" workshop team for their feedback and support, and the anonymous reviewers for their helpful comments on earlier versions of this article.

References

Akiba, Yasuhiro, Eiichiro Sumita, Hiromi Nakaiwa, Seiichi Yamamoto, and Hiroshi G. Okuno. 2004. Using a mixture of N-best lists from multiple MT systems in rank-sum-based confidence measure for MT outputs. In *Proceedings of COLING '04: The 20th International Conference on Computational Linguistics*, pages 322–328, Geneva, Switzerland.

Bertoldi, Nicola, Roldano Cattoni, Mauro Cettolo, and Marcello Federico. 2004. The ITC-irst statistical machine translation system for IWSLT-2004. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, pages 51–58, Kyoto, Japan.

Bisani, Maximilian and Hermann Ney. 2004. Bootstrap estimates for confidence intervals in ASR performance evaluation. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 409–412, Montreal, Canada.

Blatz, John, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2003. Confidence estimation for machine translation. Final report, JHU/CLSP Summer Workshop. http://www.clsp.jhu.edu/ws2003/groups/estimate/.

Blatz, John, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence estimation for machine translation. In *Proceedings of COLING '04: The 20th International Conference on Computational Linguistics*, pages 315–321, Geneva, Switzerland.

Brown, Peter F., Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Chiang, David. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of*

the 43rd Annual Meeting of the Association for Computational Linguistics (ACL), pages 263–270, Ann Arbor, MI.

Duda, Richard O., Peter E. Hart, and David G. Stork. 2001. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York.

Gandrabur, Simona and George Foster. 2003. Confidence estimation for text prediction. In *Proceedings of the Conference on Natural Language Learning (CoNLL)*, pages 95–102, Edmonton, Canada.

Jayaraman, Shyamsundar and Alon Lavie. 2005. Multi-engine machine translation guided by explicit word matching. In *Proceedings of the 10th Annual Conference of the European Association for Machine Translation (EAMT)*, pages 143–152, Budapest, Hungary.

Jurafsky, Daniel and James H. Martin. 2000. *Speech and Language Processing*. Prentice Hall, Upper Saddle River, NJ.

Koehn, Philipp, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the Human Language Technology Conference (HLT/NAACL)*, pages 127–133, Edmonton, Canada.

Levenshtein, Vladimir I. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710.

Ney, Hermann, Volker Steinbiss, Richard Zens, Evgeny Matusov, J. Gonzalez, Young-Suk Lee, Salim Roukos, Marcello Federico, Muntsin Kolss, and Rafael Banchs. 2005. TC-STAR deliverable no. D5: SLT progress report. Technical report, Integrated project TC-STAR (IST-2002-FP6-506738) funded by the European Commission. http://www.tc-star.org/.

Nießen, Sonja, Franz J. Och, Gregor Leusch, and Hermann Ney. 2000. An evaluation tool for machine translation: Fast evaluation for MT research. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC)*, pages 39–45, Athens, Greece.

NIST. 2002. Automatic evaluation of machine translation quality using N-gram co-occurrence statistics. http://nist.gov/speech/tests/mt/.

NIST. 2004. Machine translation evaluation Chinese–English. http://nist.gov/speech/tests/mt/.

Och, Franz J. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 160–167, Sapporo, Japan.

Och, Franz J. and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449.

Och, Franz J., Christoph Tillmann, and Hermann Ney. 1999. Improved alignment models for statistical machine translation. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-99)*, pages 20–28, University of Maryland, College Park, MD.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318, Philadelphia, PA.

Press, William H., Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. 2002. *Numerical Recipes in C++*. Cambridge University Press, Cambridge, UK.

Quirk, Chris. 2004. Training a sentence-level machine translation confidence metric. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC)*, pages 825–828, Lisbon, Portugal.

Sanchis, Alberto. 2004. *Estimación y aplicación de medidas de confianza en reconocimiento automático del habla*. Ph.D. thesis, Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, Valencia, Spain.

Stolcke, Andreas. 2002. SRILM—an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, volume 2, pages 901–904, Denver, CO.

TC-STAR. 2005. TC-STAR—Technology and corpora for speech to speech translation. Integrated project TC-STAR (IST-2002-FP6-506738) funded by the European Commission. http://www.tc-star.org/.

Tillmann, Christoph. 2003. A projection extension algorithm for statistical machine translation. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, pages 1–8, Sapporo, Japan.

TransType2. 2005. TransType2—Computer assisted translation. RTD project TransType2 (IST–2001–32091) funded

by the European Commission. http://tt2.atosorigin.es/.
Ueffing, Nicola. 2006. *Word Confidence Measures for Machine Translation*. Ph.D. thesis, Computer Science Department, RWTH Aachen University, Aachen, Germany.
Ueffing, Nicola, Klaus Macherey, and Hermann Ney. 2003. Confidence measures for statistical machine translation. In *Proceedings of the MT Summit IX*, pages 394–401, New Orleans, LA.
Ueffing, Nicola and Hermann Ney. 2004. Bayes decision rule and confidence measures for statistical machine translation. In *Proceedings of EsTAL—España for Natural Language Processing*, pages 70–81, Alicante, Spain. Lecture Notes in Computer Science, Springer Verlag.
Ueffing, Nicola and Hermann Ney. 2005a. Application of word-level confidence measures in interactive statistical machine translation. In *Proceedings of the 10th Annual Conference of the European Association for Machine Translation (EAMT)*, pages 262–270, Budapest, Hungary.
Ueffing, Nicola and Hermann Ney. 2005b. Word-level confidence estimation for machine translation using phrase-based translation models. In *Proceedings of the Human Language Technology Conference (HLT/EMNLP)*, pages 763–770, Vancouver, Canada.
Ueffing, Nicola, Franz J. Och, and Hermann Ney. 2002. Generation of word graphs in statistical machine translation. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, pages 156–163, Philadelphia, PA.
Vilar, David, Evgeny Matusov, Saša Hasan, Richard Zens, and Hermann Ney. 2005. Statistical machine translation of European parliamentary speeches. In *Proceedings of the MT Summit X*, pages 259–266, Phuket, Thailand.
Vogel, Stephan, Sanjika Hewavitharana, Muntsin Kolss, and Alex Waibel. 2004. The ISL statistical translation system for spoken language translation. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, pages 65–72, Kyoto, Japan.
Zens, Richard and Hermann Ney. 2004. Improvements in phrase-based statistical machine translation. In *Proceedings of the Human Language Technology Conference (HLT/NAACL)*, pages 257–264, Boston, MA.
Zens, Richard and Hermann Ney. 2005. Word graphs for statistical machine translation. In *43rd Annual Meeting of the Association for Computational Linguistics: Proceedings of the Workshop on Building and Using Parallel Texts: Data-Driven Machine Translation and Beyond*, pages 191–198, Ann Arbor, MI.
Zens, Richard and Hermann Ney. 2006. N-gram posterior probabilities for statistical machine translation. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL): Proceedings of the Workshop on Statistical Machine Translation*, pages 72–77, New York, NY.