Integration of Speech Recognition and Machine Translation in Computer-Assisted Translation

Shahram Khadivi and Hermann Ney, *Senior Member, IEEE*

***Abstract*—Parallel integration of automatic speech recognition (ASR) models and statistical machine translation (MT) models is an unexplored research area in comparison to the large amount of works done on integrating them in series, i.e., speech-to-speech translation. Parallel integration of these models is possible when we have access to the speech of a target language text and to its corresponding source language text, like a computer-assisted translation system. To our knowledge, only a few methods for integrating ASR models with MT models in parallel have been studied. In this paper, we systematically study a number of different translation models in the context of the N-best list rescoring. As an alternative to the N-best list rescoring, we use ASR word graphs in order to arrive at a tighter integration of ASR and MT models. The experiments are carried out on two tasks: English-to-German with an ASR vocabulary size of 17 K words, and Spanish-to-English with an ASR vocabulary of 58 K words. For the best method, the MT models reduce the ASR word error rate by a relative of 18% and 29% on the 17 K and the 58 K tasks, respectively.**

***Index Terms*—Computer-assisted translation (CAT), speech recognition, statistical machine translation (MT).**

I. Introduction

THE GOAL of developing a computer-assisted translation (CAT) system is to meet the growing demand for high-quality translation. A desired feature of CAT systems is the integration of human speech, as skilled human translators are faster in dictating than typing the translations [1]. In addition, another useful feature of CAT systems is to embed a statistical machine translation (MT) engine within an interactive translation environment [2], [3]. In this way, the system combines the best of two paradigms: the CAT paradigm, in which the human translator ensures high-quality output, and the MT paradigm, in which the machine ensures significant productivity gains. In this paper, we investigate the efficient ways to integrate automatic speech recognition (ASR) and MT models so as to increase the overall CAT system performance.

In a CAT system with integrated speech, two sources of information are available to recognize the speech input: the target language speech and the given source language text. The target language speech is a human-produced translation of the source language text. The overall schematic of a speech-enabled computer-assisted translation system is depicted in Fig. 1.

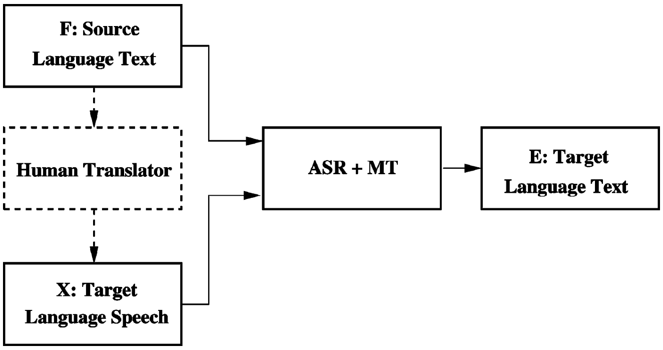

Fig. 1. Schematic of automatic text dictation in computer-assisted translation.

The idea of incorporating ASR and MT models in a CAT system has been early studied by researchers involved in the TransTalk project [4], [5], and researchers at IBM [1]. In [1], the authors proposed a method to integrate the IBM translation model 2 [6] with an ASR system. The main idea was to design a language model which combines the trigram language model probability with the translation probability for each target word. They reported a perplexity reduction, but no recognition results. In the TransTalk project, the authors improved the ASR performance by rescoring the ASR N-best lists with a translation model. They also introduced the idea of *dynamic vocabulary* in an ASR system, where the dynamic vocabulary was derived from a MT system for each source language sentence. The better performing of the two is the N-best rescoring.

Recently, [7], [8] and [9] have studied the integration of ASR and MT models. In the first paper, we showed a detailed analysis of the effect of different MT models on rescoring the ASR N-best lists. The other two papers considered two parallel N-best lists, generated by MT and ASR systems, respectively. They showed improvement in the ASR N-best rescoring when several features are extracted from the MT N-best list. The main concept among all features was to generate different kinds of language models from the MT N-best list. All of the above methods were based on an N-best rescoring approach. In [10], we studied different methods for integrating MT models to ASR word graphs instead of the N-best list.

In [11], the integration of speech into a CAT system is studied from a different point of view. To facilitate the human and machine interactions, the speech is used to determine the acceptable partial translations by reading parts of the target sentence offered by the system in addition to keyboard and mouse. In this

Manuscript received December 21, 2007; revised June 08, 2008. This work was supported by the European Union under the RTD project TransType2 (IST 2001 32091) and the integrated project TC-STAR—Technology and Corpora for Speech to Speech Translation—(IST-2002-FP6-506738). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ruhi Sarikaya.

The authors are with the Department of Computer Science, RWTH Aachen University, 52056 Aachen, Germany (e-mail: khadivi@cs.RWTH-Aachen.de; ney@cs.RWTH-Aachen.de).

Digital Object Identifier 10.1109/TASL.2008.2004301

scenario, the ASR output is constrained to the translation provided by the MT system, i.e., the ASR language model has to be adapted to the MT output.

This paper makes the following four contributions.

1) All previous methods have employed an N-best rescoring strategy to integrate ASR and MT models. Here, we will take another step towards a full single search for the integration of ASR and MT models by using ASR word graphs. A full single search means to generate the most likely hypothesis based on ASR and MT models in a single pass without any search constraints.
2) Furthermore we will investigate several new techniques to integrate ASR and MT systems, a preliminary description of some of these integration techniques were already presented in our previous paper [10].
3) In all previous works, a phrase-based MT had the least impact on improving the ASR baseline. In this paper, we will study the reason for this failure concluding in a solution for this problem.
4) To our knowledge, up to now no experiments have been reported in this field on a large task. Here, we will perform our experiments using the N-best rescoring method and the word graph rescoring method on a standard large task, namely the European parliament plenary sessions.

The remaining part is structured as follows. In Section II, a general model for parallel integration of ASR and MT systems is described. In Section III, the details of the MT system and the ASR system are explained. In Section IV, different methods for integrating MT models into ASR models are described, and in Section V the experimental results are discussed.

II. Combining ASR and MT Models

In parallel integration of ASR and MT systems, we are given a source language sentence $F = f_1^J = f_1, \ldots, f_j, \ldots, f_J$, which is to be translated into a target language sentence $E = e_1^I = e_1, \ldots, e_i, \ldots, e_I$, and an acoustic signal $X = x_1^T = x_1, \ldots, x_t, \ldots, x_T$, which is the spoken target language sentence. Among all possible target language sentences, we will choose the sentence $\hat{E}(X,F)$ with the highest probability

$$\begin{aligned}(X,F) &\rightarrow \hat{E}(X,F)\\ &= \operatorname*{argmax}_E \{P(E|X,F)\} \qquad (1)\\ &= \operatorname*{argmax}_E \{P(E,F,X)\} \qquad (2)\\ &= \operatorname*{argmax}_E \{P(E,F)\cdot P(X|E)\} \qquad (3)\\ &= \operatorname*{argmax}_E \{P(E)\cdot P(F|E)\cdot P(X|E)\}. \qquad (4)\end{aligned}$$

Equation (2) is decomposed into (4) by assuming no conditional dependency between X and F. The decomposition into three knowledge sources allows for an independent modeling of the target language model $P(E)$, the translation model $P(F|E)$, and the acoustic model $P(X|E)$. The general decision rule for the (pure) ASR and the (pure) MT system are as follows:

$$\text{ASR}: X \rightarrow \hat{E}(X) = \operatorname*{argmax}_E \{P(E)\cdot P(X|E)\}$$

$$\text{MT}: F \rightarrow \hat{E}(F) = \operatorname*{argmax}_E \{P(E)\cdot P(F|E)\}.$$

Then, the integrated model can be described as introducing the acoustic model into the MT system or as introducing the MT model into the ASR system.

Another approach for modeling the posterior probability $P(E|F,X)$ is direct modeling by using a log-linear combination of different models

$$P(E|F,X) = \frac{\exp\left[\sum_{m=1}^{M} \lambda_m h_m(E,F,X)\right]}{\sum_{E'} \exp\left[\sum_{m=1}^{M} \lambda_m h_m(E',F,X)\right]} \qquad (5)$$

then the decision rule can be written as

$$\hat{E}(F,X) = \operatorname*{argmax}_E \left\{\sum_{m=1}^{M} \lambda_m h_m(E,F,X)\right\}. \qquad (6)$$

Each of the terms $h_m(E,F,X)$ denotes one of the various models which are involved in the recognition procedure. Each individual model is weighted by its scaling factor λ_m. The direct modeling has the advantage that additional models can be easily integrated into the overall system. The model scaling factors λ_1^M are trained on a development corpus according to the final recognition quality measured by the word error rate (WER) [12]. This approach has been suggested by [13] and [14] for a natural language understanding task, by [15] for an ASR task, and by [16] for an MT task.

In a CAT system, by assuming no direct dependence between F and X, we have the following three main models:

$$\begin{aligned}h_1(E,F,X) &= h_{\text{LM}}(E) = \log P(E)\\ &= \log \prod_i p\left(e_i|e_{i-n+1}^{i-1}\right)\\ h_2(E,F,X) &= h_{\text{AM}}(E,X) = \log P(X|E)\\ &= \log \prod_i p(\text{“}x_i\text{”}|e_i)\\ h_3(E,F,X) &= h_{\text{MT}}(E,F).\end{aligned}$$

In these equations, $p(e_i|e_{i-n+1}^{i-1})$ is a language model (LM) n-gram probability and $p(\text{“}x_i\text{”}|e_i)$ represents an acoustic model (AM) probability, where “x_i” is the most likely segment of X corresponding to e_i. The definition of the machine translation (MT) model $h_{\text{MT}}(E,F)$ is intentionally left open, as it will be later defined in different ways, e.g., as a log-linear combination of several sub-models.

The argmax operator in (6) denotes the *search*. The search is the main challenge in the integration of the ASR and the MT models in a CAT system. The search in the MT and in the ASR systems are already very complex; therefore, a full single search to combine the ASR and the MT models will considerably increase the complexity. In addition, a full single search becomes more complex since there is no any specific model nor any training data to learn the alignment between F and X.

To reduce the complexity of the search, the recognition process is performed in two steps. At first, the baseline ASR system generates a large word graph for a given target language speech X. Second, the MT model rescores each word graph based on the associated source language sentence. Thus, for each target language speech, the decision regarding the most likely target sentence is based on the ASR and the MT models. It is also possible to rescore a MT word graph by using ASR models, but in practice the ASR system results in a much better output accuracy than the MT system; therefore, the rescoring of the ASR word graphs with MT models is more reasonable.

We can make the integration process even simpler, if we extract the N-best list from the ASR word graph, and then rescore the N-best list by using a log-linear combination of the ASR and the MT models.

Finally, the general decision rule for a CAT system can be written as

$$\hat{E}(F,X) = \underset{E\in\mathcal{E}}{\operatorname{argmax}}\Big\{\lambda_{\mathrm{LM}}h_{\mathrm{LM}}(E) + \lambda_{\mathrm{AM}}h_{\mathrm{AM}}(E,X) + \lambda_{\mathrm{MT}}h_{\mathrm{MT}}(E,F)\Big\} \quad (7)$$

where $\mathcal{E}$ is a finite set of possible target sentences. In a full single search, the set $\mathcal{E}$ includes all possible sentences in the target language. In the word graph rescoring approach, the set $\mathcal{E}$ includes all possible paths through the ASR word graph, and in the N-best rescoring approach, the set $\mathcal{E}$ is limited to the ASR N-best hypotheses. In this paper, we study different ways to model $h_{\mathrm{MT}}(F,E)$ under different search constraints, i.e., different definitions of $\mathcal{E}$.

III. Baseline Components

In this section, we briefly describe the basic system components, namely the ASR and the MT systems.

A. Automatic Speech Recognition System

In this paper, we work with two languages as target language: German and English. Both the German and the English ASR systems are state-of-the-art systems. The German ASR system was mainly developed within the Verbmobil project [17], and was further developed and optimized for the specific domain of this paper, which is different from the Verbmobil domain. The English ASR system is the system that is used and developed within the TC-Star project. In this paper we used the system which has been used in TC-Star 2006 evaluation [18].

The ASR systems employed here produce word graphs in which arcs are labeled with start and end time, the recognized entity (word, noise, hesitation, silence), the probability of acoustic vectors between start and end time given the entity, and the language model probability. We have further postprocessed the ASR word graph. First, we mapped all entities that were not spoken words onto the empty arc label ε. As the time information is not used in our approach, we removed it from the word graphs and compressed the structure by applying ε-removal, determinization, and minimization. We have also merged the LM and the AM scores to one score by using the log-linear combination method. This means the ratio of $\lambda_{\mathrm{AM}}/\lambda_{\mathrm{LM}}$ in (7) is a constant value, which is determined in the ASR optimization process. For all of these operations, we have employed the finite-state transducer toolkit of [19] which efficiently implemented them on demand. This step significantly reduces the runtime without changing the results.

B. Machine Translation System

In [6], a set of *alignment models* was introduced to model $P(F|E)$, namely IBM-1 to IBM-5. These models are based on similar principles as in hidden Markov models (HMMs) for ASR, we rewrite the translation probability by introducing the *hidden alignments* A for each sentence pair $(F;E)$

$$P(F|E) = \sum_{A} P(F,A|E).$$

1) IBM-1,2, and Hidden Markov Models: The first type of alignment models is virtually identical to HMMs and is based on a mapping $j \rightarrow i = a_j$, which assigns a source position j to a target position $i = a_j$. Using suitable modeling assumptions [6], [20], we can decompose the probability $P(F,A|E)$ with $A = a_1^J$

$$P(f_1^J, a_1^J|e_1^I) = p(J|I) \cdot \prod_{j=1}^{J} [p(a_j|a_{j-1}, I, J) \cdot p(f_j|e_{a_j})]$$

with the length model $p(J|I)$, the alignment model $p(i|i', I, J)$ and the lexicon model $p(f_j|e_i)$. The alignment models IBM-1 and IBM-2 are obtained in a similar way by allowing only zero-order dependencies.

2) IBM-3,4, and 5 Models: For the generation of the target sentence, it is more appropriate to use the concept of *inverted alignments* which perform a mapping from a target position i to a set of source positions j, i.e., we consider mappings B of the form

$$B : i \rightarrow B_i \subset \{1, \ldots, j, \ldots, J\}$$

with the constraint that each source position j is covered exactly once. Using such an alignment $A = B_1^I$, we rewrite the probability $P(F,A|E)$

$$P(f_1^J, B_1^I|e_1^I) = p(J|I) \cdot \prod_{i=1}^{I} \Big[p(B_i|B_1^{i-1}) \cdot \prod_{j\in B_i} p(f_j|e_i)\Big].$$

By making suitable assumptions, in particular first-order dependencies for the inverted alignment model $p\left(B_i|B_1^{i-1}\right)$, we arrive at what is more or less equivalent to the alignment models IBM-3, 4, and 5 [6], [20]. To train the above models, we use the GIZA++ software [20].

3) Phrase-Based MT: In all preceding models, the translation is based on single words. In [21] and [22], the authors show a significant improvement in translation quality by modeling *word groups* rather than *single words* in both the alignment and the lexicon models. The method is known in general as the *phrase-based (PB) MT*. Here, we make use of the RWTH phrase-based statistical MT system [23], which is a state-of-the-art phrase-based MT system. This system directly models the posterior probability $P(E|F)$ using a log-linear combination of several models [16]. To arrive at our MT model, we first perform a segmentation of the source and target sentences into K *blocks* $k \longrightarrow d_k \equiv (i_k; b_k, j_k)$, for $k = 1, \ldots, K$. Here, i_k denotes the last position of the kth target phrase; we set $i_0 := 0$. The pair (b_k, j_k) denotes the start and end positions of the source phrase which is aligned to the kth target phrase; we set $j_0 := 0$. Phrases are defined as nonempty contiguous sequences of words. We constrain the segmentations so that all words in the source and the target sentence are covered by exactly one phrase. Thus, there are no gaps and there is no overlap. For a given sentence pair $\left(f_1^J, e_1^I\right)$, and a given segmentation d_1^K, we define the bilingual phrases $(\widetilde{f}_k, \widetilde{e}_k)$ as [22]

$$\widetilde{f}_k := f_{b_k}, \ldots, f_{j_k}, \quad \widetilde{e}_k := e_{i_{k-1}+1}, \ldots, e_{i_k}.$$

Note that the segmentation d_1^K contains the information on the phrase-level reordering. The segmentation d_1^K is introduced as a hidden variable in the MT model. Therefore, it would be theoretically correct to sum over all possible segmentations. In practice, we use the maximum approximation for this sum.

The decision rule of this MT system can be written as shown in (8) at the bottom of the page, with weights $\lambda_i, i = 1, \ldots, 8$. The weights λ_1 and λ_3 play a special role and are respectively used to control the number I of words (word penalty) and number K of segments (phrase penalty) for the target sentence to be generated. In addition to the word penalty and the phrase penalty, the following models are used in the phrase-based MT:

- $p\left(e_i | e_{i-n+1}^{i-1}\right)$: word-based n-gram language model;
- $p(\widetilde{e}_k|\widetilde{f}_k)$ and $p(\widetilde{f}_k|\widetilde{e}_k)$: phrase-based models;
- $p(f_j|e_i)$ and $p(e_i|f_j)$: single word models;
- $q(b_k|j_{k-1}) = e^{|b_k - j_{k-1} - 1|}$: reordering model.

The reordering model is based on the distance from the end position of a phrase to the start position of the next phrase. A full search in MT is possible but time consuming. Therefore, a beam search pruning technique is obligatory to reduce the translation time.

If we model h_{MT} in (7) with the phrase-based MT, then it can be written as

$$h_{\mathrm{MT}}(E, F) = h_{\mathrm{PB}}(E, F) = \max_{K, d_1^K} \log P_{\mathrm{PB}}\left(f_1^J, e_1^I; d_1^K\right). \tag{9}$$

This phrase-based MT can produce word graphs as the output of the translation instead of a single-best translation. The MT word graph is a recorded track of the MT search [24].

IV. ASR AND MT INTEGRATION

In this section, we introduce several approaches to integrate the MT models with the ASR models. The common assumption through all methods described in this section is that we are given a large word graph generated by the ASR system for the input target language speech.

A. Word Graphs Product

At the first glance, a practical method to combine the ASR and the MT systems is to combine them at the level of word graphs. This means the ASR system generates a large word graph for the input target language speech, and the MT system also generates a large word graph for the source language text.

As both MT and ASR word graphs are generated independently, the general decision rule of (7) can be tailored for this method by defining $h_{\mathrm{MT}}(E, F) = h_{\mathrm{PB}}(E, F)$ (9) and $\mathcal{E} = \Pi(\mathrm{WG}_{\mathrm{ASR}}) \cap \Pi(\mathrm{WG}_{\mathrm{MT}})$, where $\Pi(\mathrm{WG}_{\mathrm{ASR}})$ and $\Pi(\mathrm{WG}_{\mathrm{MT}})$ are the set of all possible paths through the ASR and the MT word graphs, respectively. If the intersection of (the existing paths in) two word graphs is zero, then the most likely hypothesis is selected only based on the language and the acoustic models; since the WER of the ASR system is usually much lower than the WER of the MT system. We call this integration method *word graphs product*. A block diagram representation of this method is shown in Fig. 2.

$$\begin{aligned}
F \rightarrow \hat{E}(F) &= \operatorname*{argmax}_{I, e_1^I} \left\{ \max_{K, d_1^K} \log P_{\mathrm{PB}}\left(f_1^J, e_1^I; d_1^K\right) \right\} \\
\log P_{\mathrm{PB}}\left(f_1^J, e_1^I; d_1^K\right) &= \sum_{i=1}^{I} \left(\lambda_1 + \lambda_2 \cdot \log p(e_i | e_{i-n+1}^{i-1})\right) \\
&\quad + \sum_{k=1}^{K} \left(\lambda_3 + \lambda_4 \cdot \log p(\widetilde{e}_k | \widetilde{f}_k) + \lambda_5 \cdot \log p(\widetilde{f}_k | \widetilde{e}_k) \right. \\
&\qquad + \lambda_6 \cdot \log \prod_{j=b_k}^{j_k} \sum_{i=i_{k-1}+1}^{i_k} p(f_j | e_i) \\
&\qquad \left. + \lambda_7 \cdot \log \prod_{i=i_{k-1}+1}^{i_k} \sum_{j=b_k}^{j_k} p(e_i | f_j) + \lambda_8 \cdot \log q(b_k | j_{k-1}) \right)
\end{aligned} \tag{8}$$

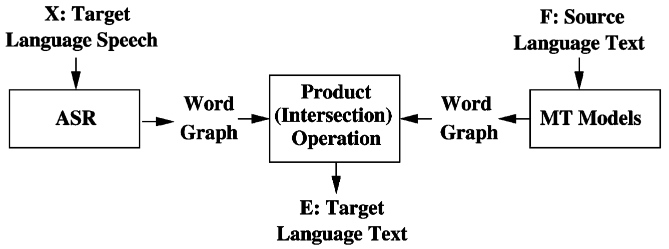

Fig. 2. Schematic of word graphs product method.

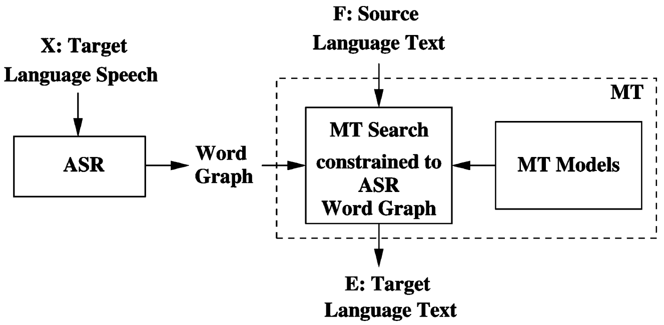

Fig. 3. Schematic of ASR-constrained search method.

The ASR and the MT word graphs can be assumed as two finite-state automata (FSA), then using FSA theory, we can implement the *word graphs product* method by using the *composition* operation. To do this, we use the finite-state transducer toolkit of [19] which is an efficient implementation of the FSA operations on demand.

In the integration of the ASR and the MT models, we define the term of *success rate* (SR). SR means how many percent of the cases the MT models are successfully integrated with the ASR models. When talking about SR, it must be kept in mind that we perform beam search rather than full search both for MT and ASR. Thus, the beam search pruning affects the SR.

In Section V, it will be shown that the intersection of the ASR and the MT word graphs are zero for a large fraction of sentences. This means the SR of this method is much smaller than one. A reason of the low SR for this method might be due to the word graph size, i.e., if we increase the word graph size, the integration success rate of this method will be increased. In Section IV-B, a solution for this problem is introduced.

B. ASR-Constrained Search

Instead of integrating the ASR and the MT word graphs which are generated independently, here the ASR word graph is integrated into the search of the MT system. To adjust the general decision rule of (7), we define $h_{\mathrm{MT}}(E,F) = h_{\mathrm{PB}}(E,F)$ (9) where the n-gram LM is excluded, and $\mathcal{E} = \Pi(\mathrm{WG}_{\mathrm{ASR}})$. Thus, the decision rule implies that the MT system is forced to generate only those hypotheses which exist in the ASR word graph. This results in a higher overlap between the ASR word graph and the MT word graph compared to the previous *word graphs product* method. It is also possible to integrate the MT word graph into the recognition process of the ASR system. However, as it will be shown in Section V, the ASR system is able to produce much better results than the MT system. Therefore, integration of ASR word graphs into the generation process of the MT system is more reasonable.

To implement this method, we replace the n-gram language model of the phrase-based MT with the ASR word graph, and the MT search is appropriately modified to be adapted with the ASR word graph as an LM (Fig. 3).

The experimental results show that this integration method also fails for a large number of sentences, since the MT system is not able to generate any target sentence that exists in the ASR word graph. A possible reason that needs to be investigated is the large difference of WERs between the ASR and the MT systems. The question is whether we can achieve better results if the MT and ASR systems have much closer WERs to each other or not.

C. Adapted LM

To answer the question initiated in the previous subsection, we need to modify the MT system to generate better results. Since we know that the accuracy of the ASR system is higher than the MT system, the MT system can be improved by adapting its language model to the ASR system output. To do this, we derive an n-gram LM from each ASR word graph, then we have a specific n-gram LM for each source sentence and we use these n-gram LMs in the MT system. To derive an n-gram LM from an ASR word graph, we first extract a large list of best hypotheses from the word graph, and then we build an n-gram LM on these hypotheses, by using the SRI language modeling toolkit [25]. In the next step, the MT word graph is generated by using this sentence specific LM. Finally, this MT word graph, which has much lower WER compared to the standard MT system, is integrated with the ASR word graph by using the word graphs product method. We adjust the general decision rule of (7) by setting $\mathcal{E} = \Pi(\mathrm{WG}_{\mathrm{ASR}}) \cap \Pi(\mathrm{WG}_{\mathrm{MT}})$ and defining h_{MT} as

$$h_{\mathrm{MT}}(E,F) = \max_{K,d_1^K} \log P_{\mathrm{PBALM}}\left(f_1^J, e_1^I; d_1^K\right)$$

where P_{PBALM} has the same definition as phrase-based MT (8), but the term $\lambda_2 \cdot \log p\left(e_i|e_{i-n+1}^{i-1}\right)$ is replaced with $\lambda_2 \cdot \log p\left(e_i|e_{i-n+1}^{i-1}, X\right)$. We call this method *adapted LM*. The overall schematic of this method is depicted in Fig. 4.

In Section V, the MT results will show a considerable reduction in the WER by using the adapted LM method, although they are not better than the ASR results. In addition, the experiments will show that the SR of this method is lower than the SR of the ASR-constrained search method. All three methods, which have been introduced so far, will have the same SR if we do not apply any pruning in the MT search.

D. MT-Derived LM

Another integration method for the MT and the ASR models is to rescore the ASR word graph with a language model that is derived from the MT system. It is similar to the idea of the preceding subsection. There, only the LM derived from ASR is used in the MT search, but here we use the LM derived from

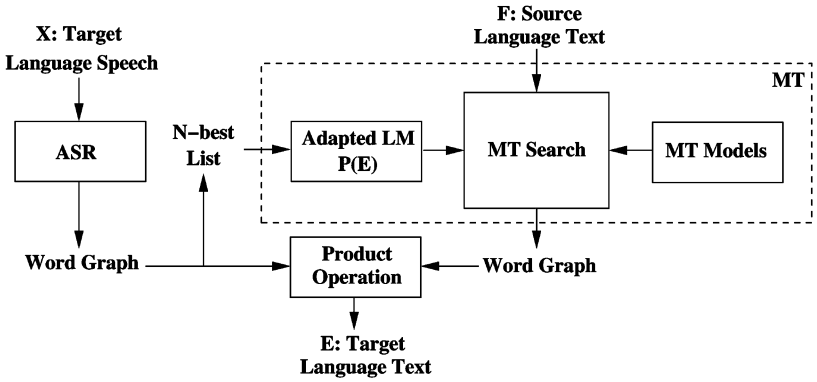

Fig. 4. Schematic of adapted LM method.

MT as a secondary LM in the ASR system since the MT output quality is much lower than the ASR output quality.

In the past, the ASR systems were based on a two-pass approach, where a bigram LM was used during the search and the generation of the word graph and in the next step a trigram LM was used to rescore the generated word graph [26]. We use the MT-derived LM in a similar way. The general decision rule of (7) can be adjusted for this method by defining $\mathcal{E} = \Pi(\mathrm{WG}_{\mathrm{ASR}})$ and h_{MT} as follows:

$$h_{\mathrm{MT}}(E,F) = \log \prod_{i=1}^{I} p\left(e_i | e_{i-2}^{i-1}, F\right).$$

A schematic representation of this method is represented in Fig. 5.

To implement this method, we represent a trigram MT-derived LM in the FSA format. Therefore, the rescoring of the ASR word graph using this LM can be implemented by using the *composition* operation. A similar method is presented in [8] and [9], in which the MT-derived LM is interpolated with their standard ASR LM and then the interpolated LM is used during the ASR search.

E. Phrase-Based With Relaxed Constraints (PBRC)

A reason for the failure of all previous methods to efficiently integrate ASR and MT models is the existence of unknown words or phrases in a sentence pair. The basic assumptions in the phrase-based MT system is that all source words have to be covered, and each target phrase corresponds to at least one source phrase, and a source phrase is at least one word. Thus, the reason of the failure for the case of unknown words is obvious. In addition, the phrase-based MT system might not find a set of phrases that cover both source and target sentences, even if there will be no unknown words neither in the source nor in the target side. This means it is not possible to find any sequence of phrases that covers all words only once in the source and target sentences, even if we do not apply any pruning in the MT search. It is important to understand that the phrase-based MT system is able to handle any input sequence, but it is not able to produce each possible target sentence, e.g., the sentence which the user has intended.

In this section, we tailor the standard phrase-based MT system in several ways to make it usable for our task. First, the target phrases are assumed to correspond to all source words instead of only one source phrase. Second, the requirement of the source sentence word coverage is removed. Finally, if there are still some words that cannot be covered with the set of target phrases, we assign the probability of $p(e|F)$ to those words, the probability can be estimated by using a more general translation model like IBM Model 1. We adjust the general decision rule of (7) for this method by defining $\mathcal{E} = \Pi(\mathrm{WG}_{\mathrm{ASR}})$ and h_{MT} as follows:

$$h_{\mathrm{MT}}(E,F) = \max_{K,\tilde{e}_1^K} \left\{ \sum_{k=1}^{K} \log p\left(\tilde{e}_k | f_1^J\right) \right\}$$

where $\tilde{e}_1^K$ is a segmentation of a target sentence into K blocks. To estimate $p\left(\tilde{e}|f_1^J\right)$, we use the following smoothing principle: For a target phrase $\tilde{e}$ in the target sentence, if there is any phrase pair $(\tilde{f}, \tilde{e})$ in the phrase table that the source side $\tilde{f}$ exists in the source sentence f_1^J, then the highest probability of such a phrase pair is assigned to $p\left(\tilde{e}|f_1^J\right)$, otherwise $p\left(\tilde{e}|f_1^J\right)$ is estimated using IBM Model 1 translation model. Thus, $\log p\left(\tilde{e}|f_1^J\right)$ can be defined as shown in the equation at the bottom of the next page, where ν is a number close to one, e.g., 0.99.

To implement this method, we design a transducer to rescore the ASR word graph. For each source language sentence, we extract all possible phrase pairs from the word-aligned training corpus. The transducer is formed by one node and a number of self loops for each individual target side of the phrase pairs $\tilde{e}$. The weight of each arc is $\log p(\tilde{e}|f_1^J)$. To implement the back-off term, we add the ASR vocabulary to this transducer with the score of $(1-\nu)\log 1/(J+1)\sum_{j=0}^{J} p(e|f_j)$, more details about the estimation of IBM Model 1 by using a transducer is given in Section IV-F.

This transducer is an approximation of a non-monotone phrase-based MT system. Using the designed transducer it is possible that some parts of the source texts are not covered or covered more than once. Then, this model can be compared to the IBM-3 and IBM-4 models, as they also have the same characteristic in covering the source words.

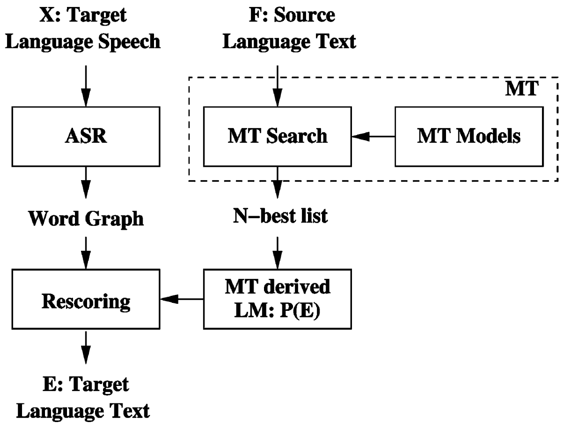

Fig. 5. Schematic of ASR word graph rescoring using MT-derived LM method.

F. IBM Model 1: Dynamic Lexical Weighting

The idea of a dynamic vocabulary, restricting the word lexicon of the ASR according to the source language text, was first introduced in [5]. The idea was also seen later in [9]; they extract the words of the MT N-best list to restrict the vocabulary of the ASR system. Here, we extend the dynamic vocabulary idea by weighting the ASR vocabulary based on the source language text and the MT models. We use the lexicon model of the HMM and the IBM MT alignment models. Based on these lexicon models, we assign to each possible target word e the probability $P(e|F)$. One way to compute this probability is inspired by IBM Model 1:

$$p_{\mathrm{IBM1}}(e|F) = \frac{1}{J+1}\sum_{j=0}^{J} p(e|f_j).$$

The general decision rule of (7) can be tailored for this method by defining $\mathcal{E} = \Pi(\mathrm{WG}_{\mathrm{ASR}})$ and h_{MT} as follows:

$$\begin{aligned} h_{\mathrm{MT}}(E,F) &= \log p_{\mathrm{IBM1}}(E|F) \\ &= \log \frac{1}{(J+1)^I} \prod_{i=1}^{I}\sum_{j=0}^{J} p(e_i|f_j). \end{aligned}$$

To implement this integration method, we design a simple transducer (or more precisely an acceptor) to efficiently rescore all paths (hypotheses) in the ASR word graph by using IBM Model 1. The transducer is formed by one node and a number of self loops for each target language word. In each arc of this transducer, the input label is a target word e and the weight is $w = \log p_{\mathrm{IBM1}}(e|F)$. Due to its simplicity, this model can be easily integrated into the ASR search. It is a sentence specific unigram LM. The overall schematic of this integration model is the same as the *ASR-constrained search* method (Fig. 3) if the MT models are limited to only the lexicon model.

G. IBM Model 3 Clone: Single Word-Based MT

In this section, we follow the idea of Brown *et al.* [1] to integrate ASR and IBM Model 2, but here we employ a more complex model, IBM Model 3. It is also true to say that we follow the concept of the ASR-constrained search introduced in Section IV-B, but here we use a single word-based (SWB) MT instead of phrase-based MT. Thus, the overall schematic of this method is depicted in Fig. 3, with this note that the MT models are all single word-based MT models.

In [6], three alignment models are described which include fertility models, these are IBM Models 3, 4, and 5. The fertility-based alignment models have a more complicated structure than the simple IBM Model 1. The fertility model estimates the probability distribution for aligning multiple source words to a single target word. The fertility model provides the probabilities $p(\phi|e)$ for aligning a target word e to ϕ source words. Here, we make use of IBM Model 3 that is defined as

$$\begin{aligned} p_{\mathrm{IBM3}}(F|E) = \max_{a_1^J} \Bigg\{ & \binom{J-\varphi_0}{\varphi_0} \cdot P_1^{\varphi_0} \cdot P_0^{J-2\varphi_0} \\ & \cdot \prod_{i=1}^{I} P(\varphi_i|e_i) \cdot \prod_{i=0}^{I} \varphi! \cdot \prod_{j=1}^{J} P(f_j|e_{a_j}) \cdot \frac{1}{J!} \Bigg\} \\ \simeq \max_{B_0^I} \Bigg\{ & \prod_{i=1}^{I} \left[P(\varphi_i|e_i) \cdot \prod_{v=1}^{\varphi_i} P(f_{b_{iv}}|e_i) \right] \\ & p\Big(\varphi_0|e_0; J - \sum_{i=1}^{I}\varphi_i\Big) \cdot \prod_{v=1}^{\varphi_0} \Big(p(f_{b_{iv}}|e_0)\Big) \Bigg\} \end{aligned}$$

where $B_i = \{b_{i1}, \ldots, b_{iv}, \ldots, b_{i\varphi_i}\}$ is the inverted alignment, and $\varphi_i = |B_i|$. Now, we have to adjust the general decision rule of (7) by defining $\mathcal{E} = \Pi(\mathrm{WG}_{\mathrm{ASR}})$ and $h_{\mathrm{MT}}(E,F) = \log p_{\mathrm{IBM3}}(F|E)$.

To implement the IBM Model 3, we follow the idea introduced in [27] to make a SWB MT, which is a finite-state implementation of the IBM Model 3. This MT system consists of a cascade of several simple transducers: lexicon, null-emitter, and fertility. The lexicon transducer is formed by one node and a number of self loops for each target language word. On each arc of the lexicon transducer, there is a lexicon entry: the input label is a source word f, the output label is a target word e, and the weight is $\log p(f|e)$. The null-emitter transducer, as its name states, emits the null word with a predefined probability after each target word. The fertility transducer is also a simple

$$\begin{cases} (1-\nu) \cdot \log \frac{1}{(J+1)^{|\widetilde{e}|}} \prod\limits_{i=1}^{|\widetilde{e}|} \sum\limits_{j=0}^{J} p(e|f_j) & \textbf{if } p(\widetilde{f}, \widetilde{e}) = 0 \\ \nu \cdot \max\limits_{\widetilde{f} \in F} \Bigg\{ \lambda_1 \cdot \log p(\widetilde{e}|\widetilde{f}) + \lambda_2 \cdot \log p(\widetilde{f}|\widetilde{e}) + \lambda_3 \log \prod\limits_{j=1}^{|\widetilde{f}|} \sum\limits_{i=0}^{|\widetilde{e}|} p(f_j|e_i) + \lambda_4 \log \prod\limits_{i=1}^{|\widetilde{e}|} \sum\limits_{j=0}^{|\widetilde{f}|} p(e_i|f_j) \Bigg\} & \textbf{otherwise} \end{cases}$$

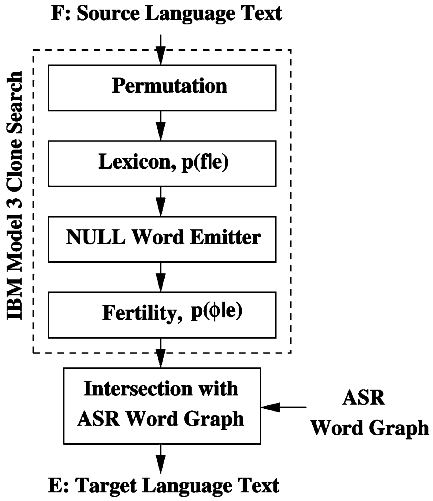

Fig. 6. Sequence of finite-state transducers to form the search in the SWB MT.

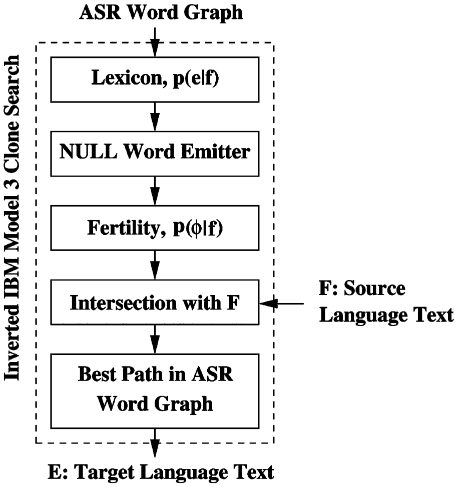

Fig. 7. Sequence of finite-state transducers to form the search in the inverted SWB MT.

transducer to map zero or several instances of a target word to one instance of the source word.

The search in this MT system can be then implemented by successive application of the described transducers to the source language text, as depicted in Fig. 6. As a part of the search, the source language text is first permuted and then it is successively composed with the lexicon, the null-emitter, the fertility transducers, and finally with the ASR word graph. In practice, especially for long sentences, a full (global) permutation of the source sentence text considerably increases the computation time and memory consumption. Therefore, to reduce the number of permutations, we need to apply some reordering constraints. Here, we use the IBM reordering [28], or the local reordering [29] constraints. The idea behind the IBM constraints is to postpone the translations of a limited number of words. This means, at each state we can translate any of the first l yet uncovered word positions. In local reordering, the next word for the translation is selected from a window of l positions (covered or uncovered) counting from the first yet uncovered position. The permutations obtained by the local constraints are a subset of the permutations defined by the IBM constraints.

Due to the higher degree of generation power (larger freedom) of the single word-based MT compared to the phrase-based MT, the integration of the ASR system with this model is usually successful. In the case of the integration failure, which is less than 6% of the sentences in our experiments, we use the ASR output.

1) Inverted IBM Model 3 Clone: We may also apply the SWB MT from the target to the source. This means to translate the ASR word graph into the source language text. In this approach, the best recognition result is the best path in the input ASR word graph which has the highest translation likelihood to the source language text. The general decision rule of (7) for the inverted IBM Model 3 is specified by defining $\mathcal{E} = \Pi(\mathrm{WG}_{\mathrm{ASR}})$ and $h_{\mathrm{MT}}(E,F) = \log p_{\mathrm{IBM3}}(E|F)$. To implement this method by using the cascade of finite-state transducers, we need some reorganization in the search compared to the standard translation direction as shown in Fig. 7. The main difference is that the source input, which is the ASR word graph, is not permuted, i.e., the translation is monotone in respect to the source input. This is because we are completely sure that the word order in the ASR word graph is correct, and we are looking for *the best path* in the ASR word graph that has the highest likelihood to the corresponding source sentence. However, to model the non-monotonicity between the source sentence F and its corresponding translation E, we permute the word orders in the source sentence F by using IBM or local reordering constraints, although the source sentence F is treated here as the target language sentence. Please note that this difference between the inverted model and the standard model may result in a rough approximation of IBM Model 3 for the inverted direction.

An interesting point concerning the inverted model, when we model $P(E|F)$, is the similarity of the decision rule to the speech translation, where the word graph is used as the interface between ASR and MT systems. The decision rule in a speech translation system (to translate the target spoken language to the source language text) is as follows:

$$\hat{F}(X) = \operatorname*{argmax}_{F}\Big\{\lambda_{\mathrm{AM}}h_{\mathrm{AM}}(E,X) + \lambda_{\mathrm{LM}}h_{\mathrm{SLM}}(E) + \lambda_{\mathrm{MT}}h_{\mathrm{MT}}(E,F) + \lambda_{\mathrm{TLM}}h_{\mathrm{TLM}}(F)\Big\}$$

where the first two terms form the ASR system, the second two terms form the MT system, $h_{\mathrm{MT}}(E,F)$, is usually defined as $\log P(E|F)$, and SLM and TLM refer to source and target LM, respectively. Therefore, we can describe the parallel integration of the ASR and the MT systems as similar to a speech translation system, except that we aim to get the best ASR output (the best path in the ASR word graph) rather than the best translation. Since the best translation is already given, we do not need a language model for the target language (here F), i.e., $\lambda_{\mathrm{TLM}} = 0$.

H. ASR and MT Integration: n-Best List Approach

Here, we aim to rescore the N-best lists that are extracted from the ASR word graphs, instead of rescoring the ASR word graphs directly. To rescore a word graph we have some limitations in implementing the rescoring models, e.g., a rescoring model needs to be implemented in a form of a transducer. However, to rescore an N-best list, we can use any model to rescore a pair of source and target sentences.

The decision rule for N-best rescoring approach can be derived from (7), by specifying $\mathcal{E}$ as all hypotheses in the ASR N-best list and $h_{\mathrm{MT}}(E, F)$ can be a single MT model or a combination of MT models including single-word based, or phrase-based MT models.

1) Single-Word Based MT Models: Here, we rescore the ASR N-best lists with the standard HMM [30] and IBM [6] MT models. To rescore a source-target sentence pair, specially when the target side is produced by the ASR system, we should carefully take into account the unknown words—unseen words in MT models. In the case of single-word based MT models, we assume a uniform probability distribution for unknown words.

2) Phrase-Based MT Model: We can rescore the ASR N-best list with the PB MT system described in Section III-B. We should note that the PB MT system is able to rescore a sentence pair only if it is able to generate the target sentence from the source sentence. Therefore, when we use phrase-based MT models in rescoring, there is a specific problem that we have to take into account. As described in Section IV-E, the phrase-based models are able to match any source sentence but not to generate any target sentence. As we will show in Section V, the SR of all previous methods built upon phrase-based MT have a SR much lower than one. The reason is that the PB MT models used in the rescoring do not have a nonzero probability for all hypotheses in the word graph. An efficient way to solve this problem is to introduce a model which is able to penalize and omit the words, either in the source or in the target side of a sentence pair, which cannot be covered by any sequence of phrases from the phrase table. With such a model, the system would prefer the hypothesis with a minimum number of uncovered words. The number of uncovered words can be used as another model in addition to the current models of the phrase-based MT system. Using this method, *PB MT with omission*, every sentence pair can be successfully rescored. We will show the results of the ASR N-best list rescoring with standard PB MT and with PB MT with omission in Section V.

Each of these described models assigns at least one score to each entry of the ASR N-best list. These models are integrated with the existing ASR score (acoustic model score plus scaled language model score) by using the log-linear approach. The scaling factors of all models are optimized on a development corpus with respect to the WER, the optimization algorithm is Downhill Simplex [31].

V. Results

The ASR and MT integration experiments are carried out on two different tasks: English–German Xerox manual translation which is a small domain task, and a large vocabulary task which is the Spanish–English parliamentary speech translation.

The English–German Xerox task consists of technical manuals describing various aspects of Xerox hardware and software installation, administration, usage, etc. The German pronunciation lexicon was derived from the VerbMobil II corpus [17].

TABLE I
Statistics of the Speech Recognition Train Corpus

Task	Xerox	EPPS
Language	German	English
Acoustic data [h]	61.5	87.5
# Running words	701 K	705 K
Vocabulary size	17 K	58 K
# Segments	36 K	67 K
# Speakers	857	154

The language and MT models are trained on the part of the English–German Xerox corpus which was not used in the ASR test corpus [7]. A bilingual test corpus including 1562 sentences was randomly extracted, and the target side of these sentences were read by ten native German speakers where every speaker uttered on an average 16 min of test data. Recording sessions were carried out in a quiet office room. We divide this ASR test corpus into two parts, the first 700 utterances as the development corpus and the rest as the evaluation corpus.

We also tested the integration methods on a large vocabulary task, namely the European Parliament Plenary Sessions (EPPS) task. The training corpus for this task has been collected in the framework of the European research project TC-Star. In this project, an open speech-to-speech translation evaluation was conducted in March 2006, including Spanish-to-English and English-to-Spanish language pairs. Here, we use the RWTH systems of the English ASR [18] and the Spanish-to-English MT systems that were employed in TC-Star 2006 evaluation campaign. The English pronunciation lexicon was derived from the British English Example Pronunciations Dictionary (BEEP), and the acoustic model of the English ASR system is trained on the transcribed recordings from the EPPS [18]. TC-Star evaluation results show that the RWTH systems (ASR and MT) have a very good performance among other state-of-the-art systems. In Table I, some statistics of the ASR training sets for both tasks are presented.

We should note that we need to modify the speech-to-speech experiment conditions in order to be appropriate for the experiments of this paper. For this purpose, we only need to change the MT direction. This means, in order to have the English speech in the target language, we have to use the Spanish-to-English MT system. Thus, the development and evaluation sets also need to be switched, and as we have two reference translations, we assume the first reference as the source language sentence. The development and evaluation sets of the EPPS task are taken from the TC-Star 2005 evaluation campaign, since only for this evaluation an explicit segmentation of the speech signal corresponding to the target text sentences is given. This explicit segmentation is useful in our experiments as the implicit segmentation will introduce another source of errors into the parallel integration system.

The statistics of both Xerox and EPPS corpora are depicted in Table II. The term OOVs in the table denotes the total number of occurrences of unknown words, the words which were not seen in the training corpus. In both tasks, we make use of a backoff n-gram language model using the SRI language modeling toolkit [25]. We use a 3-gram LM in the German ASR system and a 4-gram LM in the English ASR system as well as in both Xerox and EPPS MT systems. The applied smoothing is

TABLE II
MACHINE TRANSLATION CORPUS STATISTICS

Task		Xerox		EPPS	
		English	German	Spanish	English
Train	Sentences	47 619		1 167 627	
	Running words	529 K	468 K	35.3 M	33.9 M
	Vocabulary size	9 816	16 716	159 080	110 636
	Singletons	2 302	6 064	63 045	46 121
Dev	Sentences	700		1 750	
	Running words	8 823	8 050	22 174	23 429
	OOVs	56	108	64	83
Eval	Sentences	862		792	
	Running words	11 019	10 094	19 081	19 306
	OOVs	58	100	43	45

TABLE III
DEVELOPMENT AND EVALUATION WORD GRAPHS STATISTICS

Task			Xerox		EPPS	
			Dev	Eval	Dev	Eval
ASR		WER[%]	19.3	21.3	14.6	11.5
		BLEU[%]	74.7	71.9	77.0	81.3
		avg. density	18	32	1908	1249
		graph error rate	10.9	11.8	4.0	2.6
MT	Standard	WER[%]	46.9	49.9	43.9	48.5
		BLEU[%]	47.6	45.2	38.3	34.5
		avg. density	1017	1038	1120	1389
		graph error rate	28.3	29.7	18.2	20.8
	Adapted LM	WER[%]	33.7	36.1	21.1	21.0
		BLEU[%]	61.0	58.6	69.2	69.6
		avg. density	859	906	713	711
		graph error rate	23.0	24.7	10.9	11.6

modified Kneser–Ney discounting with interpolation. To train the LMs, the target side of the bilingual corpora (Table II) are used, but for the English ASR, the transcriptions of the acoustic training data are also included into the training corpus [18].

A. Word Graph-Based Approach Results

Here, we study the performance of different word graph rescoring methods proposed in this paper. Table III shows the statistics of both MT and ASR word graphs for both Xerox and EPPS tasks, where *adapted LM* refers to the MT word graphs generated with the adapted LM integration method. *BLEU* is a translation accuracy measure which is widely used in the MT community [32]. The *graph error rate* (GER) is the oracle WER of the word graph. This means, it can be computed by determining the sentence in the word graph that has the minimum Levenshtein distance to a given reference. Thus, it is a lower bound on the WER that can be obtained by rescoring the word graph.

The recognition results using word graph rescoring methods are shown in Table IV, where the results corresponding to the EPPS task are case-insensitive. In Table V, the integration success rate of the ASR and the MT models are shown.

The ASR system generates much better results than the MT system. Thus, the baseline recognition/translation WERs are 21.3% and 11.5% on the evaluation sets of the Xerox and the EPPS tasks, respectively. Now we discuss the experimental results of word graph-based methods. First, we conducted a set of experiments to integrate the ASR and the MT systems by using the *word graphs product* method. We obtain a WER of 20.9% and 11.4% for the evaluation sets of the Xerox and the EPPS tasks, respectively. A detailed analysis, as shown in Table V, reveals that only 28.2% and 13.5% of sentences in the Xerox and the EPPS evaluation sets have identical paths in the two ASR and MT word graphs, respectively. The SR of this method is directly related to the word graph size, i.e., if we increase the word graph size, the SR of this method will be increased. We study the effect of the MT word graph size on the SR using the *ASR-constrained search* method, in which the ASR word graph is integrated into the MT search and we allow a large search space for the MT. Thus, it is equivalent to using a very large MT word graph in the *word graphs product* method.

TABLE IV
RECOGNITION WER [%] USING WORD GRAPH RESCORING METHOD

Task			Xerox		EPPS	
			Dev	Eval	Dev	Eval
PB MT			46.9	49.9	43.9	48.5
ASR (baseline)			19.3	21.3	14.6	11.5
ASR + PB MT	word graphs product		19.0	20.9	14.0	11.4
	ASR-constrained search		19.0	20.7	13.7	11.3
	adapted LM		19.0	20.8	12.9	10.9
	MT-derived LM		18.8	20.8	13.4	11.3
	PBRC		18.6	20.3	13.7	10.9
ASR + SWB MT	IBM Model 1,	INV	17.3	19.0	13.8	10.9
	IBM Model 3	STD	17.9	20.1	13.3	10.7
	clone	INV	17.5	18.5	12.9	10.3

TABLE V
RECOGNITION SUCCESS RATE (SR) [%] ON THE WORD GRAPH RESCORING METHOD

Task			Xerox		EPPS	
			Dev	Eval	Dev	Eval
ASR + PB MT	word graphs product		33.4	28.2	36.3	13.5
	ASR-constrained search		45.8	41.6	74.0	49.1
	adapted LM		38.0	33.9	61.2	36.4
	MT-derived LM		52.7	50.0	56.9	30.0
	PBRC		100		100	
ASR + SWB MT	IBM Model 1,	INV	100		100	
	IBM Model 3	STD	93.4	94.0	95.7	95.4
	clone	INV	93.4	94.0	95.5	95.3

Using the *ASR-constrained search* method, the number of identical paths in MT and ASR word graphs are increased to 41.6% and 49.1% of the sentences in the evaluation set of the Xerox and the EPPS tasks, respectively. The results show that this integration method also fails for more than half of the sentences on the evaluation sets, even with a large search space. A possible reason that needs to be investigated is the large difference of WERs between the ASR and the MT systems.

Now we study whether we can achieve better integration results if the MT and the ASR word graphs have less difference in WER. Using the *adapted LM* method, we first generate the MT word graphs for all source sentences which are all non-empty. We obtain a relative 27.6% and 56.7% WER reduction on the Xerox and the EPPS MT tasks (Table III), respectively. Second, we integrate these MT word graphs with the ASR word graphs using the *word graphs product* method. We obtain a remarkable improvements over the baseline system in the EPPS task. The results also show that the *adapted LM* method is able to generate more compact and more accurate word graphs.

We do not obtain better recognition results using the *MT-derived LM* method compared to the previous methods. The *PBRC*

TABLE VI
DEVELOPMENT AND EVALUATION N-BEST LISTS STATISTICS

Task	Xerox		EPPS	
	Dev	Eval	Dev	Eval
# utterances	700	862	1 750	792
ave. N per utterance	1 534	1 904	5 599	8 463
max N per utterance	5 000	5 000	10 000	10 000
single best WER[%]	19.3	21.3	14.6	11.5
oracle recognition WER[%]	11.2	12.4	5.1	4.1

method results in WER of 20.3% and 10.9% for the evaluation sets of the Xerox and the EPPS tasks, respectively. The *PBRC* method is a fairly simplified version of the phrase-based MT system, and therefore the main advantage of this method is its simplicity. The SR of this method is 100%, as this method is always able to rescore the ASR word graphs.

The *PBRC* and the *adapted LM* methods are the best performing methods among those which are based on phrase-based MT. Although the improvements achieved over the ASR baseline are statistically significant at the 99% level [33], they are not large in terms of the WER. Looking at the results, we can also see that a higher SR does not necessarily mean a better WER, although a high SR is a prerequisite for a successful integration.

So far, all these integration methods are based on phrase-based MT; now we study the integration results using single word-based MT. First, we conduct experiments using the *IBM Model 1* method. The results are shown in Table IV, where STD and INV mean different translation directions, $p(F|E)$ and $p(E|F)$, respectively. The results are promising, especially regarding the simplicity of the model. The *IBM Model 3 clone* and its inverted variant perform well in integrating MT and ASR models. We did not observe significant changes in the recognition results by using different methods and settings to permute the source text. The results reported here are achieved by using the local reordering method with a window size of 4. It is possible to use the submodels, i.e., the lexicon and the fertility models, of the more complex alignment models (IBM Model 4 and 5) in the *IBM Model 3 clone* method, since more complex alignment models can provide a better estimate of their submodels including those which are in common with IBM Model 3. We conduct a set of experiments to investigate the effect of using the submodels of more complex alignment models on the recognition quality. The experiment results do not show any remarkable differences from the results of the table for the INV direction. However, for the STD direction, we observe the WER of **18.9%** and **9.8%** for the evaluation set of the Xerox and the EPPS tasks, respectively.

The SR in the SWB models are significantly higher than the phrase-based MT models; it is mainly due to the higher degree of freedom in the generation process of the SWB models compared to the usual phrase-based MT.

B. N-Best Rescoring Results

Here, we present the results of the N-best list approach to integrate the ASR and the MT models. Table VI shows the statistics of the Xerox and the EPPS N-best lists. The best possible hypothesis achievable from the N-best list has the WER (oracle WER) of 12.4% and 4.1% for the evaluation set of Xerox and EPPS tasks, respectively.

TABLE VII
RECOGNITION WER [%] USING N-BEST RESCORING ($N = 50$) METHOD

Task			Xerox		EPPS	
			Dev	Eval	Dev	Eval
PB MT			46.9	49.9	43.9	48.5
ASR (baseline)			19.3	21.3	14.6	11.5
oracle recognition WER			12.9	14.4	7.4	6.6
ASR + MT	IBM1	STD	17.4	19.3	11.9	9.4
		INV	17.8	19.3	13.8	10.9
		BOTH	17.1	18.7	11.9	9.4
	HMM	STD	17.4	19.3	11.9	9.4
		INV	17.8	19.3	13.8	10.9
		BOTH	17.1	18.7	11.9	9.4
	IBM3	STD	18.3	20.1	13.5	10.9
		INV	17.3	18.2	13.0	10.3
		BOTH	17.4	18.6	12.6	10.2
	IBM4	STD	16.9	18.9	11.1	9.1
		INV	17.2	18.3	12.4	9.8
		BOTH	16.4	17.8	11.0	9.0
	IBM5	STD	17.0	19.0	11.4	9.2
		INV	17.3	18.3	12.4	9.8
		BOTH	16.6	18.1	11.2	9.1
	PB MT		18.7	20.6	14.0	11.4
	PB MT with omission		16.8	18.5	11.6	10.2

The N-best list recognition results are summarized in Table VII, where the size of the N-best is limited to 50 hypotheses per source sentence ($N = 50$). For each MT model, the N-best lists are rescored by using standard direction $p(F|E)$ (STD), and inverted direction $p(E|F)$ (INV) translation probabilities of the specific model and the probabilities of the acoustic and the language models. The scaling factors of all models are optimized on a development corpus.

To rescore the ASR N-best list by using the *PB MT* and the *PB MT with omission* methods, we employ all phrase-based models described in Section III-B, and for the latter, we also use the number of uncovered words in the sentence pair. In previous works [7], [10], only the results of the *PB MT* were reported. The significant improvements of the *PB MT with omission* method over the *PB MT* method is due to the use of a more flexible model which allows the existence of uncovered words (by all possible entries in the phrase table) either in the source or in the target side of a sentence pair. However, this flexibility results in a higher computational requirements, which is critical to rescore a large N-best list.

To study the effect of the N-best list size on the integration results, we repeat the N-best rescoring experiments with much larger N-best lists, a maximum of 5000 and 10 000 hypotheses per sentence for the Xerox and the EPPS tasks, respectively. The results are shown in Table VIII. In this table, the results of the *PB MT with omission* method for the EPPS task are missing due to a high computational requirement.

Comparing Tables VII and VIII, we observe some improvements by using a very large N-best list. However, a larger N-best list needs more processing time. To find the optimum N-best list size, we conduct a sequence of experiments with different sizes of N-best list. The integration results are depicted in Figs. 8 and 9 for the Xerox and the EPPS tasks, respectively. The optimum N-best list size depends on the rescoring model, more sophisticated models like IBM-4 and IBM-5 models are able to profit from a larger N-best list. The maximum size of the N-best list for the EPPS task is

TABLE VIII
RECOGNITION WER [%] USING N-BEST RESCORING METHOD ($N = 5$ K FOR THE XEROX AND $N = 10$ K FOR THE EPPS TASKS)

Task			Xerox		EPPS	
			Dev	Eval	Dev	Eval
PB MT			46.9	49.9	43.9	48.5
ASR (baseline)			19.3	21.3	14.6	11.5
oracle recognition WER			11.2	12.4	5.1	4.1
ASR + MT	IBM1	STD	17.1	19.0	11.5	9.2
		INV	17.4	18.9	13.7	10.9
		BOTH	16.7	18.3	11.5	9.2
	HMM	STD	18.1	19.5	12.3	9.5
		INV	18.1	19.2	13.6	10.7
		BOTH	17.4	18.5	12.1	9.3
	IBM3	STD	18.1	20.0	13.4	10.8
		INV	17.1	18.1	12.8	10.0
		BOTH	16.9	18.4	12.4	9.9
	IBM4	STD	16.6	18.7	10.5	8.2
		INV	16.8	17.9	12.3	9.5
		BOTH	16.0	17.4	10.4	8.1
	IBM5	STD	16.7	18.7	10.9	8.4
		INV	16.8	18.0	12.2	9.3
		BOTH	16.1	17.8	10.5	8.3
	PB MT		18.7	20.6	13.9	11.4
	PB MT with omission		16.5	17.8	N/A	N/A

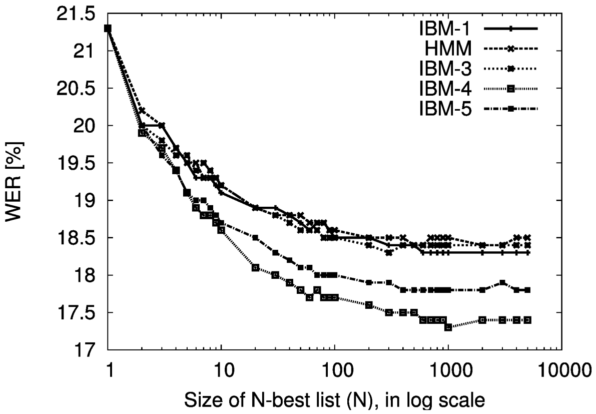

Fig. 8. N-best rescoring results for different N-best sizes on the Xerox evaluation set.

10 000. We do not go beyond 10 000 hypotheses because the experiments are computationally very expensive. In addition, as shown in Fig. 9, when the maximum size of the N-best list is increased from 5000 to 10 000, the absolute improvement is only 0.1% and 0.05% for IBM Model 4 and 5, respectively.

In general, N-best rescoring is a simplification of word graph rescoring. As the size of the N-best list is increased, the results obtained by N-best list rescoring converge to the results of the word graph rescoring. However, we should note that this statement is correct only when we use exactly the same model and the same implementation to rescore the N-best list and word graph. In this paper, this is true for the PB MT and the inverted IBM Model 1 methods and, to certain approximation, for the standard and the inverted IBM Model 3. Table IX summarizes the results of these models for which we can directly compare the word graph rescoring and the N-best rescoring methods. In the table, the methods built upon PB MT in word graph rescoring approach are compared with the PB MT method in N-best rescoring, while PB MT with omission employs a smoothing method which has not been used in none of the word graph rescoring methods. The differences between the N-best list and word graph approaches are not statistically significant at the 99% level [33], except for the inverted IBM Model 3 clone. The reason why the inverted IBM Model 3 clone is an exception here might be that its implementation for word graphs is not a good approximation of IBM Model 3, as pointed out in Section IV-G.

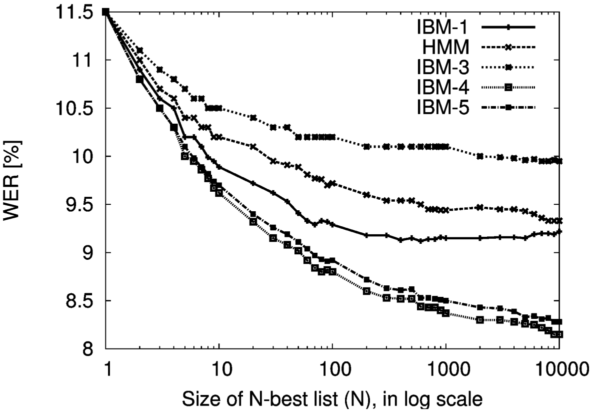

Fig. 9. N-best rescoring results for different N-best sizes on the EPPS evaluation set.

TABLE IX
RECOGNITION WER [%] ON THE EVALUATION CORPUS OF THE XEROX AND THE EPPS TASKS BY USING COMPARABLE METHODS OF THE N-BEST RESCORING APPROACH AND THE WORD GRAPH RESCORING APPROACH

Approach			word graph		N-best	
Task			Xerox	EPPS	Xerox	EPPS
ASR+ PB MT	word graphs product		20.9	11.4	20.6	11.4
	ASR-constrained search		20.7	11.3		
	MT-derived LM		20.8	11.3		
ASR+ SWB MT	IBM Model 1,	INV	19.0	10.9	18.9	10.9
	IBM Model 3 clone	STD	20.1	10.7	20.0	10.8
		INV	18.5	10.3	18.1	10.0

VI. CONCLUSION

We have studied different approaches to integrate the MT and ASR models into a CAT system at the level of word graphs and N-best lists. One of the main goals in this research was to take another step towards a full single search for the integration of ASR and MT models; to extend the work presented in [1], where ASR models were integrated to the IBM translation model 2 and only the perplexity reduction was reported. As the word graph is a real representation of the search space in the ASR, the rescoring of the ASR word graph with the MT models would be an acceptable simulation of a full single search. We have proposed several new methods to rescore the ASR word graphs with IBM translation model 1 and 3, and phrase-based MT. All improvements of the combined models are statistically significant at the 99% level with respect to the ASR baseline system.

The best integration results are obtained by using the N-best list rescoring approach because of the flexibility of the N-best list approach to use more complex models like IBM Model 4 or 5 in an accurate and efficient way. However, an advantage of the word graph rescoring is the confidence of achieving the best possible results based on a given rescoring model. Another advantage of working with ASR word graphs is the capability to pass on the word graphs for further processing. For instance, the resulting word graph can be used in the prediction engine of a CAT system [3].

We have also shown that the phrase-based MT system can also be effectively integrated to the ASR system, whereas in the previous works, phrase-based MT had the least impact on improving the ASR baseline. We showed that smoothing of the phrase-based MT is necessary in order to effectively integrate it to the ASR models. The rescoring of the ASR N-best list with smoothed phrase-based MT, *PB MT with omission*, outperforms all phrase-based MT rescoring methods in the word graph rescoring approach, even when the size of N-best list is small ($N = 50$).

We conducted the experiments on the Xerox task and on a standard large task, the EPPS task. To our knowledge, these are the first experiments that have been done on a large task so far. We have obtained a relative 18% and 29% error rate reduction, respectively, on the Xerox and the EPPS tasks using IBM Model 4 (in both directions) in the N-best rescoring approach. The largest improvements obtained by the word graph rescoring approach for Xerox and EPPS tasks were a relative 13% and 10%, respectively, using IBM Model 3 in inverted direction for the Xerox task and in standard direction for the EPPS task.

Acknowledgment

The authors would like to thank T. Szilassy for his efforts in providing the orthographic transcription of the Xerox ASR test set. They would also like to thank A. Zolnay and J. Lööf for providing the ASR word graphs for the Xerox and the EPPS tasks, respectively. This paper has also profited from discussions with R. Zens and E. Matusov.

References

[1] P. F. Brown, S. F. Chen, S. A. D. Pietra, V. J. D. Pietra, A. S. Kehler, and R. L. Mercer, “Automatic speech recognition in machine-aided translation,” *Comput. Speech Lang.*, vol. 8, no. 3, pp. 177–187, Jul. 1994.

[2] G. Foster, P. Isabelle, and P. Plamondon, “Target-text mediated interactive machine translation,” *Mach. Translation*, vol. 12, no. 1, pp. 175–194, 1997.

[3] F. J. Och, R. Zens, and H. Ney, “Efficient search for interactive statistical machine translation,” in *Proc. EACL03: 10th Conf. Eur. Chap. Assoc. Comput. Linguist.*, Budapest, Hungary, Apr. 2003, pp. 387–393.

[4] M. Dymetman, J. Brousseau, G. Foster, P. Isabelle, Y. Normandin, and P. Plamondon, “Towards an automatic dictation system for translators: the TransTalk project,” in *Proc. ICSLP’94*, Yokohama, Japan, 1994, pp. 193–196.

[5] J. Brousseau, C. Drouin, G. Foster, P. Isabelle, R. Kuhn, Y. Normandin, and P. Plamondon, “French speech recognition in an automatic dictation system for translators: The transtalk project,” in *Proc. Eurospeech*, Madrid, Spain, 1995, pp. 193–196.

[6] P. F. Brown, S. A. D. Pietra, V. J. D. Pietra, and R. L. Mercer, “The mathematics of statistical machine translation: Parameter estimation,” *Comput. Linguist.*, vol. 19, no. 2, pp. 263–311, Jun. 1993.

[7] S. Khadivi, A. Zolnay, and H. Ney, “Automatic text dictation in computer-assisted translation,” in *Proc. Interspeech’05—Eurospeech, 9th Eur. Conf. Speech Commun. Technol.*, Lisbon, Portugal, 2005, pp. 2265–2268.

[8] M. Paulik, S. Stüker, C. Fügen, T. Schultz, T. Schaaf, and A. Waibel, “Speech translation enhanced automatic speech recognition,” in *Proc. Automatic Speech Recognition and Understanding Workshop (ASRU)*, San Juan, Puerto Rico, 2005, pp. 121–126.

[9] M. Paulik, C. Fügen, S. Stüker, T. Schultz, T. Schaaf, and A. Waibel, “Document driven machine translation enhanced ASR,” in *Proc. Interspeech”05—Eurospeech, 9th Eur. Conf. Speech Commun. Technol.*, Lisbon, Portugal, 2005, pp. 2261–2264.

[10] S. Khadivi, R. Zens, and H. Ney, “Integration of speech to computer-assisted translation using finite-state automata,” in *Proc. COLING/ACL Main Conf., Companion Vol.*, Sydney, Australia, Jul. 2006, pp. 467–474.

[11] E. Vidal, F. Casacuberta, L. Rodrìguez, J. Civera, C. D. M. Hinarejos, and S. Processing, “Computer-assisted translation using speech recognition,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 3, pp. 941–951, May 2006.

[12] F. J. Och, “Minimum error rate training in statistical machine translation,” in *Proc. 41th Annu. Meeting Assoc. Comput. Linguist. (ACL)*, Sapporo, Japan, Jul. 2003, pp. 160–167.

[13] K. A. Papineni, S. Roukos, and R. T. Ward, “Feature-based language understanding,” in *EUROSPEECH*, Rhodes, Greece, Sep. 1997, pp. 1435–1438.

[14] K. A. Papineni, S. Roukos, and R. T. Ward, “Maximum likelihood and discriminative training of direct translation models,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Seattle, WA, May 1998, vol. 1, pp. 189–192.

[15] P. Beyerlein, “Discriminative model combination,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Seattle, WA, May 1998, vol. 1, pp. 481–484.

[16] F. J. Och and H. Ney, “Discriminative training and maximum entropy models for statistical machine translation,” in *Proc. 40th Annu. Meeting Assoc. Comput. Linguist. (ACL)*, Philadelphia, PA, Jul. 2002, pp. 295–302.

[17] A. Sixtus, S. Molau, S. Kanthak, R. Schlüter, and H. Ney, “Recent improvements of the RWTH large vocabulary speech recognition system on spontaneous speech,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Istanbul, Turkey, Jun. 2000, pp. 1671–1674.

[18] J. Lööf, M. Bisani, C. Gollan, G. Heigold, B. Hoffmeister, C. P. R. Schlüter, and H. Ney, “The 2006 rwth parliamentary speeches transcription system,” in *Proc. 9th Int. Conf. Spoken Lang. Process. (ICSLP)*, Pittsburgh, PA, Sep. 2006, pp. 105–108.

[19] S. Kanthak and H. Ney, “FSA: An efficient and flexible C++ toolkit for finite state automata using on-demand computation,” in *Proc. 42nd Annu. Meeting Assoc. Comput. Linguist. (ACL)*, Barcelona, Spain, Jul. 2004, pp. 510–517.

[20] F. J. Och and H. Ney, “A systematic comparison of various statistical alignment models,” *Comput. Linguist.*, vol. 29, no. 1, pp. 19–51, Mar. 2003.

[21] F. J. Och, C. Tillmann, and H. Ney, “Improved alignment models for statistical machine translation,” in *Proc. Joint SIGDAT Conf. Empirical Methods in Natural Lang. Process. Very Large Corpora*, College Park, MD, Jun. 1999, pp. 20–28.

[22] F. J. Och and H. Ney, “The alignment template approach to statistical machine translation,” *Comput. Linguist.*, vol. 30, no. 4, pp. 417–449, Dec. 2004.

[23] R. Zens, O. Bender, S. Hasan, S. Khadivi, E. Matusov, J. Xu, Y. Zhang, and H. Ney, “The RWTH phrase-based statistical machine translation system,” in *Proc. Int. Workshop Spoken Lang. Translation (IWSLT)*, Pittsburgh, PA, Oct. 2005, pp. 155–162.

[24] N. Ueffing, F. J. Och, and H. Ney, “Generation of word graphs in statistical machine translation,” in *Proc. Conf. Empirical Methods for Natural Lang. Process. (EMNLP)*, Philadelphia, PA, Jul. 2002, pp. 156–163.

[25] A. Stolcke, “Srilm—an extensible language modeling toolkit,” in *Proc. Int. Conf. Speech Lang. Process. (ICSLP)*, Denver, CO, Sep. 2002, vol. 2, pp. 901–904.

[26] S. Ortmanns, H. Ney, and X. Aubert, “A word graph algorithm for large vocabulary continuous speech recognition,” *Comput., Speech Lang.*, vol. 11, no. 1, pp. 43–72, Jan. 1997.

[27] K. Knight and Y. Al-Onaizan, “Translation with finite-state devices,” in *AMTA*, ser. Lecture Notes in Computer Science, D. Farwell, L. Gerber, and E. H. Hovy, Eds. New York: Springer Verlag, 1998, vol. 1529, pp. 421–437.

[28] A. L. Berger, P. F. Brown, S. A. D. Pietra, V. J. D. Pietra, J. R. A. S. K. Gillett, and R. L. Mercer, “Language translation apparatus and method of using context-based translation models,” U.S. patent 5,510,981, Apr. 1996.

[29] S. Kanthak, D. Vilar, E. Matusov, R. Zens, and H. Ney, "Novel reordering approaches in phrase-based statistical machine translation," in *Proc. 43rd Annu. Meeting Assoc. Comput. Linguist.: Proc. Workshop Building and Using Parallel Texts: Data-Driven Machine Translation, and Beyond*, Ann Arbor, MI, Jun. 2005, pp. 167–174.

[30] S. Vogel, H. Ney, and C. Tillmann, "Hmm-based word alignment in statistical translation," in *Proc. COLING'96: 16th Int. Conf. Comput. Linguist.*, Copenhagen, Denmark, Aug. 1996, pp. 836–841.

[31] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C++*. Cambridge, U.K.: Cambridge Univ. Press, 2002.

[32] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguist. (ACL)*, Philadelphia, PA, Jul. 2002, pp. 311–318.

[33] M. Bisani and H. Ney, "Bootstrap estimates for confidence intervals in ASR performance evaluationx," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Montreal, QC, Canada, May 2004, pp. 409–412.

Shahram Khadivi was born in Esfahan, Iran, in 1974. He received the B.S. and M.S. degrees in computer engineering from Amirkabir University of Technology, Tehran, Iran, in 1996 and 1999, respectively, and the Ph.D. degree in computer science from RWTH Aachen University, Aachen, Germany, in 2008.

Since 2002, he has been with the Human Language Technology and Pattern Recognition Research Group, RWTH Aachen University, where he has also been pursuing his doctoral study in computer science since 2004. His research interests include pattern recognition, computational natural language processing, machine learning, and machine vision.

Hermann Ney (SM'07) is a Full Professor of computer science at RWTH Aachen University, Aachen, Germany. Before, he headed the Speech Recognition Group at Philips Research. His main research interests lie in the area of statistical methods for pattern recognition and human language technology and their specific applications to speech recognition, machine translation, and image object recognition. In particular, he has worked on dynamic programming for continuous speech recognition, language modeling, and phrase-based approaches to machine translation. He has authored and coauthored more than 350 papers in journals, books, conferences, and workshops.

Prof. Ney was a member of the Speech Technical Committee of the IEEE Signal Processing Society from 1997 to 2000. In 2006, he was the recipient of the Technical Achievement Award of the IEEE Signal Processing Society.