System Combination for Machine Translation of Spoken and Written Language

Evgeny Matusov, *Student Member, IEEE*, Gregor Leusch, Rafael E. Banchs, Nicola Bertoldi, Daniel Déchelotte, Marcello Federico, Muntsin Kolss, Young-Suk Lee, José B. Mariño, *Member, IEEE*, Matthias Paulik, *Student Member, IEEE*, Salim Roukos, Holger Schwenk, *Member, IEEE*, and Hermann Ney, *Senior Member, IEEE*

***Abstract*—This paper describes an approach for computing a consensus translation from the outputs of multiple machine translation (MT) systems. The consensus translation is computed by weighted majority voting on a confusion network, similarly to the well-established ROVER approach of Fiscus for combining speech recognition hypotheses. To create the confusion network, pairwise word alignments of the original MT hypotheses are learned using an enhanced statistical alignment algorithm that explicitly models word reordering. The context of a whole corpus of automatic translations rather than a single sentence is taken into account in order to achieve high alignment quality. The confusion network is rescored with a special language model, and the consensus translation is extracted as the best path. The proposed system combination approach was evaluated in the framework of the TC-STAR speech translation project. Up to six state-of-the-art statistical phrase-based translation systems from different project partners were combined in the experiments. Significant improvements in translation quality from Spanish to English and from English to Spanish in comparison with the best of the individual MT systems were achieved under official evaluation conditions.**

***Index Terms*—machine translation, natural languages, speech processing, text processing.**

I. Introduction

IN THIS paper, we describe a new algorithm for computing a consensus translation from the outputs of multiple machine translation systems.

Combining outputs from different systems was shown to be quite successful in automatic speech recognition (ASR). Voting schemes like the ROVER approach of Fiscus [11] use edit distance alignment and time information to create confusion networks[1] from the output of different ASR systems for the same audio input. The consensus recognition hypothesis is generated by weighted majority voting.

The biggest challenge in application of system combination algorithms to machine translation (MT) is the need for word reordering. Different translation hypotheses from different systems may have different word order. This means that some hypotheses have to be reordered so that corresponding words can be aligned with each other in the confusion network.

In this paper, we show how the reordering problem in system combination for MT can be solved. Our approach to computing a consensus translation includes an enhanced alignment and reordering framework. In contrast to existing approaches [15], [34], the context of the whole corpus rather than a single sentence is considered in this iterative, unsupervised procedure, yielding a more reliable alignment.

The basic concept of the approach to be presented has been previously described in a conference publication [24]. Since then, the alignment and reordering framework was substantially improved. Also, the procedure of constructing the confusion network and computing the consensus translation from it was extended by important novel features. More precisely, several confusion networks are combined in a single lattice to improve performance. The majority voting on this lattice is performed using not only the prior probabilities for each system, but other statistical models such as a special n-gram language model. In this paper, the approach is thoroughly evaluated on a real-life translation task using the output of several state-of-the-art MT systems produced under conditions of an official evaluation. We present automatic and human evaluation results which show that the resulting consensus translation generally has better quality than the original translations and yet may be different from any of them.

This paper is organized as follows. In Section II, we will review the work related to the subject of system combination for machine translation. Section III will present our system combination algorithm in detail. In Section IV, we will describe and compare the MT systems involved in our experiments. These MT systems are research systems of the groups participating in the European Union project *TC-STAR* (*Technology and Corpora for Speech-to-Speech Translation* [38]). The experimental results, including the TC-STAR 2007 MT evaluation results, will

Manuscript received May 10, 2007; revised November 7, 2007. This work was supported in part by the European Union under 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. Bill Byrne.

E. Matusov, G. Leusch, and H. Ney are with RWTH Aachen University, 52056 Aachen, Germany (e-mail: matusov@cs.rwth-aachen.de; leusch@cs.rwth-aachen.de; ney@cs.rwth-aachen.de).

R. E. Banchs and J. B. Mariño are with the Universitat Politècnica de Catalunya (UPC), 08034 Barcelona, Spain (e-mail: rbanchs@gps.tsc.upc.edu; canton@gps.tsc.upc.edu).

N. Bertoldi and M. Federico are with Fondazione B. Kessler (FBK), 38100 Trento, Italy (e-mail: bertoldi@fbk.eu; federico@fbk.eu).

D. Déchelotte and H. Schwenk are with the Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur (LIMSI), 91403 Orsay cedex, France (e-mail: dechelot@limsi.fr; holger.schwenk@limsi.fr).

M. Kolss and M. Paulik are with the University of Karlsruhe (UKA), 76128 Karlsruhe, Germany (e-mail: kolss@ira.uka.de; paulik@ira.uka.de).

Y.-S. Lee and S. Roukos are with IBM Research, Yorktown Heights, NY 10598 USA (e-mail: ysuklee@us.ibm.com; roukos@us.ibm.com).

Digital Object Identifier 10.1109/TASL.2008.914970

[1]A confusion network is a weighted directed acyclic graph, in which each path from the start node to the end node goes through the same sequence of all other nodes. A matrix representation of a confusion network is shown in Fig. 3.

TER Alignment	HMM Alignment
I think that you know # you will be aware , I believe	
$ $ I think that you know will be aware , I you believe	I think that you $ $ know I believe , you will be aware
a huge fall in average prices # a decline strong in the prices means	
a huge fall in average prices $ a decline strong in the prices means	a huge fall in $ average prices a strong decline in the means prices

Fig. 1. Examples of the TER-based alignment in comparison with the alignment produced by the enhanced alignment and reordering algorithm of [24] (HMM alignment). In each example, the second translation is reordered to match the word order of the first one, given the alignment. The $ symbol denotes deletions/insertions in the alignment. The examples are from the TC-STAR evaluation data.

be presented in Section V. We will conclude with a summary in Section VI.

II. Related Work

Some research on multi-engine machine translation has been performed in recent years. The approaches can be divided in two main categories. The first set of methods are *selection* methods, i.e., for each sentence, one of the provided hypotheses is selected. Thus, the resulting translation comes from a set of already produced translations. The hypothesis selection is made based on the combination of different scores from n-gram language models [6], [27], but also from translation models and other features [32]. The best translation can also be selected from the combined N-best lists of the different MT systems. To be successful, such approaches require comparable sentence translation scores. However, the scores produced by most statistical machine translation (SMT) systems are not normalized and therefore not directly comparable. For some other MT systems (e.g., knowledge-based systems), the scores of hypotheses may not be even available. If scores are available, they have to be rescaled. Some suggestions how this can be done are found in [27], [34], and [39].

There is also a second set of approaches in which the system combination translation is created from subsentence parts (words or phrases) of the original system translations. The advantage of these approaches is that a possibly new translation can be generated that includes "good" partial translations from each of the involved systems. Some authors follow the idea of producing word alignments between the system translations, which then can be transformed into confusion networks so that a consensus translation can be computed in the style of [11]. Bangalore *et al.* [1] use the edit distance alignment extended to multiple sequences to construct a confusion network from several translation hypotheses. This algorithm produces monotone alignments only; hence, it is not able to align translation hypotheses with significantly different word order. Jayaraman and Lavie [15] try to overcome this problem. They introduce a method that allows for nonmonotone alignments of words in different translation hypotheses for the same sentence. However, this approach uses many heuristics and is based on the alignment that is performed to calculate a specific MT error measure; performance improvements have been reported only in terms of this measure. Recently, Rosti *et al.* [34] also followed a confusion network combination approach. They used the alignment based on translation error rate (TER, [37]). This alignment procedure computes the edit distance extended by allowing shifts of word blocks. Only exactly matching phrases can be shifted, and the shifts are selected greedily. The costs of aligning synonyms to each other are the same as those of aligning completely unrelated words. In many cases, the synonyms will not be matched to each other, but will be considered as insertions or deletions in their original positions. This is suboptimal for confusion network voting, for which it is important to align as many corresponding words as possible, considering reasonable reorderings of words and phrases.

Previous approaches for aligning multiple translations only exploited the alternative system hypotheses available for a particular sentence. In contrast, the enhanced hidden Markov model (HMM) alignment algorithm presented in [24] and explained in detail in this article makes the alignment decisions depend on probabilities iteratively trained on a whole corpus translated by the participating multiple MT systems. Thus, the alignment of synonyms and other related words can be learned automatically. Examples in Fig. 1 indicate that the alignments produced using this method (as well as the word reordering based on these alignments) compare favorably with the TER alignment used by Rosti *et al.* [34].

Finally, a few other system combination approaches do not perform the alignment between the hypotheses, but rather rely on the alignment with the source sentence. In one of the first publications on system combination in MT, Frederking and Nirenburg [13] create a chart structure where target language phrases from each system are placed according to their corresponding source phrases, together with their confidence scores. A chart-walk algorithm is used to select the best translation from the chart. More recently, Rosti *et al.* [34] show that a system combination translation can be produced by performing a new search with one of the involved phrase-based MT systems, but using only the phrases from the translation hypotheses provided by the participating systems. Syntactical phrases have to be flattened in order to pursue this approach. Although this method is superior to a selection approach, it is limited by the fact that all of the systems have to provide phrasal alignments with word sequences in the source sentence. In particular, this means that all the systems are required to work with the same preprocessing of the source sentence, which may reduce the

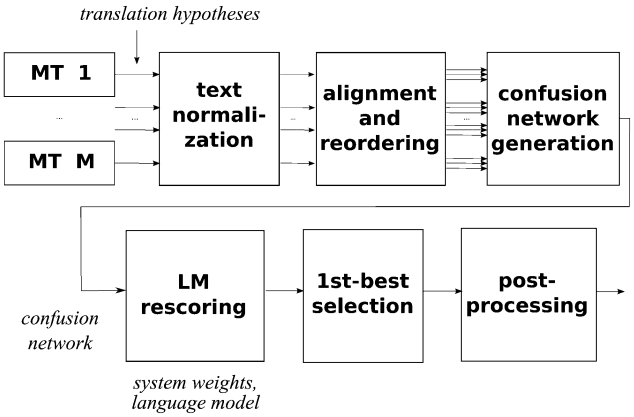

Fig. 2. System combination architecture.

diversity in their translations. Another limitation is that the final translation is generated by the "simple" phrase-based decoder, so that the system combination translation is bound to its structural restrictions.

III. System Combination Algorithm

In this section, we present the details of our system combination method. The notation is introduced in Section III-A, followed by the description of the algorithm. The algorithm consists of several steps. In the first step, the alignment between the hypotheses is determined as described in Section III-B. Then, the word order of some of the hypotheses is changed so that the alignment becomes monotone (Section III-C). From this alignment, a confusion network is created as explained in Section III-D. Section III-E describes the last step of the algorithm, in which the confusion network is scored with different statistical models, and the consensus translation is extracted. The algorithm also includes some important advanced features, which are described in detail in Section III-F. Fig. 2 gives an overview of the system combination architecture described in this section.

A. Notation

Given a single source sentence F in the test corpus, we combine M translation hypotheses $E_1, \ldots, E_m, \ldots, E_M$ coming from M MT engines. Each hypothesis $E_m (m = 1, \ldots, M)$ consists of I_m target language words

$$E_m := e_{m,1}, e_{m,2}, \ldots, e_{m,i}, \ldots, e_{m,I_m}.$$

In the following, we will also consider an *alignment* between two hypotheses E_n and E_m translating the same source sentence, $m, n \in \{1, \ldots, M\}; m \neq n$. In general, an alignment $A \subseteq I_n \times I_m$ is a relation between the words in each of the two hypotheses. Here, we will consider alignments which are functions of the words in E_m, i.e., $A : \{1, \ldots, I_m\} \rightarrow \{1, \ldots, I_n\}$.

B. Word Alignment

The proposed alignment approach is a statistical one. It takes advantage of multiple translations for a whole corpus to compute a consensus translation for each sentence in this corpus. It also takes advantage of the fact that the sentences to be aligned are in the same language.

For each source sentence F in the test corpus, we select one of its translations $E_n, n = 1, \ldots, M$ as the *primary* hypothesis. Then we align the *secondary* hypotheses $E_m (m = 1, \ldots, M; n \neq m)$ with E_n to match the word order in E_n. Since it is not clear which hypothesis should be primary, i.e., has the "best" word order, we let every hypothesis play the role of the primary translation, and align all pairs of hypotheses $(E_n, E_m); n \neq m$ (see Section III-E).

The word alignment is *trained* in analogy to the alignment training procedure in statistical MT. The difference is that the two sentences that have to be aligned are in the same language. We consider the conditional probability $Pr(E_m|E_n)$ of the event[2] that, given E_n, another hypothesis E_m is generated from the E_n. Then, the alignment between the two hypotheses is introduced as a hidden variable $\mathcal{A}$

$$Pr(E_m|E_n) = \sum_{\mathcal{A}} Pr(E_m, \mathcal{A}|E_n). \tag{1}$$

This probability is then decomposed into the alignment probability $Pr(\mathcal{A}|E_n)$ and the lexicon probability $Pr(E_m|\mathcal{A}, E_n)$

$$Pr(E_m, \mathcal{A}|E_n) = Pr(\mathcal{A}|E_n) \cdot Pr(E_m|\mathcal{A}, E_n). \tag{2}$$

As in statistical machine translation, we make modeling assumptions. We use the IBM Model 1 [5] and the HMM [44] to estimate the alignment model. The lexicon probability of a sentence pair is modeled as a product of single-word based probabilities of the aligned words

$$Pr(E_m|\mathcal{A}, E_n) = \prod_{j=1}^{I_m} p(e_{m,j}|e_{n,a_j}). \tag{3}$$

Here, the alignment a is a function of the words in the secondary translation E_m, so that each word $e_{m,j}$ in E_m is aligned to the word $e_{n,i}$ in E_n on position $i = a_j$.

The alignment training corpus is created from a test corpus[3] of N sentences (e.g., a few hundred) translated by the involved MT engines. However, the effective size of the training corpus is larger than N, since all pairs of different hypotheses have to be aligned. Thus, the effective size of the training corpus is $M \cdot (M-1) \cdot N$.

The single-word based lexicon probabilities $p(e|e')$ used in (3) are initialized from normalized lexicon counts collected over the sentence pairs (E_m, E_n) on this corpus. Since all of the hypotheses are in the same language, we count co-occurring identical words, i.e., if $e_{m,j}$ is the same word as $e_{n,i}$ for some i and j. In addition, we add a fraction of a count for words with identical prefixes. The initialization could be furthermore improved by using, e.g., a list of synonyms for the words involved.

The model parameters—the lexicon model $p(e|e')$ and the alignment model—are trained iteratively with the EM algorithm

[2]The notational convention will be as follows: we use the symbol $Pr(\cdot)$ to denote general probability distributions with (nearly) no specific assumptions. In contrast, for model-based probability distributions, we use the generic symbol $p(\cdot)$.

[3]A test corpus can be used directly because the alignment training is unsupervised and only automatically produced translations are considered.

system hypotheses	**0.25**	**would your like coffee or tea**							
	0.35	have you tea or coffee							
	0.10	would like your coffee or							
	0.30	I have some coffee tea would you like							
alignment and reordering	have|**would** you|**your** $|**like** coffee|**coffee** or|**or** tea|**tea**								
	would|**would** your|**your** like|**like** coffee|**coffee** or|**or** $|**tea**								
	I|**$** would\|**would** you\|**your** like\|**like** have\|**$** some|**$** coffee|**coffee** $|**or** tea|**tea**								
confusion network	**$**	**would**	**your**	**like**	**$**	**$**	**coffee**	**or**	**tea**
	$	have	you	$	$	$	coffee	or	tea
	$	would	your	like	$	$	coffee	or	$
	I	would	you	like	have	some	coffee	$	tea
voting	**$**	**would**	**you**	$	**$**	**$**	**coffee**	**or**	**tea**
	0.7	0.65	0.65	0.35	0.7	0.7	1.0	0.7	0.9
	I	have	your	**like**	have	some		$	$
	0.3	0.35	0.35	0.65	0.3	0.3		0.3	0.1
consensus translation	would you like coffee or tea								

Fig. 3. Example of creating a confusion network from monotone one-to-one word alignments (denoted with symbol |). The words of the primary hypothesis are printed in bold. The symbol $ denotes a null alignment or an ε-arc in the corresponding part of the confusion network.

using the GIZA++ toolkit [29]. The training is performed in the directions $E_m \rightarrow E_n$ and $E_n \rightarrow E_m$. The updated lexicon tables from the two directions are interpolated after each iteration.

The final alignments are determined using a cost matrix C for each sentence pair (E_m, E_n). The elements of this matrix are the local costs $C(j,i)$ of aligning a word $e_{m,j}$ from E_m to a word $e_{n,i}$ from E_n. Following [22], we compute these local costs by interpolating the negated logarithms of the state occupation probabilities[4] from the "source-to-target" and "target-to-source" training of the HMM. For a given alignment $A \subset I_n \times I_m$, we define the costs of this alignment $C(A)$ as the sum of the local costs of all aligned word pairs. The goal is to find a minimum cost alignment fulfilling certain constraints. Two different alignments are computed using the cost matrix C: the alignment $\tilde{a}$ used for reordering each secondary translation E_m, and the alignment $\bar{a}$ used to build the confusion network.

C. Word Reordering

The alignment $\tilde{a}$ between E_m and the primary hypothesis E_n used for reordering is determined under the constraint that it must be a function of the words in the secondary translation E_m with minimal costs. It can be easily computed from the cost matrix C as

$$\tilde{a}_j = \underset{i}{\operatorname{argmin}} C(j,i). \tag{4}$$

The word order of the secondary hypothesis E_m is changed. The words $e_{m,j}$ in E_m are sorted by the indices $i = \tilde{a}_j$ of the words in E_n to which they are aligned. If two or more words in E_m are aligned to the same word in E_n, they are kept in the original order.

After reordering each secondary hypothesis E_m and the rows of the corresponding alignment cost matrix according to the permutation given by the alignment $\tilde{a}$, we determine $M-1$ monotone *one-to-one* alignments between E_n as the primary translation and $E_m, m = 1, \ldots, M; m \neq n$. This type of alignment will allow a straightforward construction of the confusion network in the next step of the algorithm. In case of many-to-one connections in $\tilde{a}$ of words in E_m to a single word from E_n, we only keep the connection with the lowest alignment costs. This means that for each position i in E_n the unique alignment connection with a word in E_m is found with the following equation:

$$\bar{a}_i = \underset{j:\tilde{a}_j=i}{\operatorname{argmin}} C(j,i). \tag{5}$$

The use of the one-to-one alignment $\bar{a}$ implies that some words in the secondary translation will not have a correspondence in the primary translation and vice versa. We consider these words to have a null alignment with the empty word ε. In the corresponding confusion network, the empty word will be transformed to an ε-arc.

D. Building Confusion Networks

Given the $M-1$ monotone one-to-one alignments, their transformation to a confusion network can be performed. We follow the approach of Bangalore *et al.* [1] with some extensions. The construction of a confusion network is best explained by the example in Fig. 3. Here, the original $M = 4$ hypotheses are shown, followed by the alignment of the reordered secondary hypotheses 2–4 to the primary hypothesis 1 (shown in bold). The alignment is shown with the | symbol, where the words of the primary hypothesis are to the right of this symbol. The symbol $ denotes a null alignment or an ε-arc in the corresponding part of the confusion network.

Starting from an initial state s_0, the primary hypothesis is processed from left to right and a new state is produced for each word $e_{n,i}$. Then, an arc is created from the previous state to this state, for $e_{n,i}$ and for all words (or the null word) aligned to $e_{n,i}$. If there are insertions following $e_{n,i}$ (for example, "have some" in Fig. 3), the states and arcs for the inserted words are also created.

The difficulty in handling the insertions arises from the fact that several word sequences from different secondary translations can be inserted between two consecutive primary words $e_{n,i}$ and $e_{n,i+1}$. This is illustrated by the example in Fig. 4.

[4]These are marginal probabilities of the form $p_j(i, E_m|E_n) = \sum_{a:a_j=i} Pr(E_m, \mathcal{A}|E_n)$ normalized over target positions i.

system hypotheses	**I have coffee** I have liked hot coffee I have always liked coffee					
alignment	I|**I** I|**I**	have|**have** have|**have**	liked|**$** always\|**$**	hot|**$** liked\|**$**	coffee|**coffee** coffee|**coffee**	
CN1: (wrong)	**I** I I	**have** have have	**$** liked always	**$** hot liked	**coffee** coffee coffee	
CN2: (correct)	**I** I I	**have** have have	**$** $ always	**$** liked liked	**$** hot $	**coffee** coffee coffee

Fig. 4. Example of a wrong and a correct confusion network for the insertions w.r.t. the primary hypothesis.

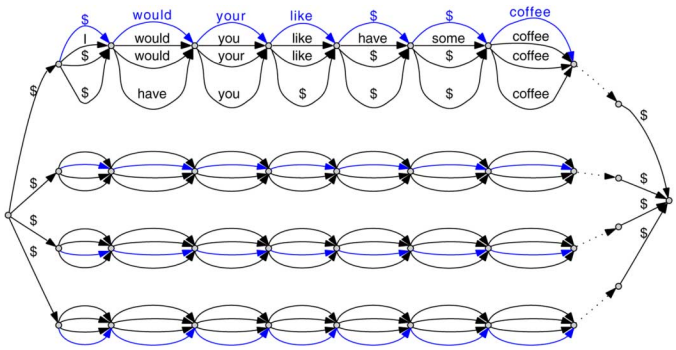

Fig. 5. Union of several confusion networks, including the one shown in Fig. 3.

Here, the primary hypothesis is "I have coffee." Since the words "always," "liked," and "hot" are all insertions, they can be considered "as is," yielding the confusion network CN1. However, a better correspondence can be achieved when we ensure that identical words are aligned with each other. To this end, we compute the edit distance alignment between all the insertions of the secondary translations. In the example Fig. 4, the edit distance alignment between "liked hot" and "always liked" is computed, yielding the confusion network CN2. Clearly, CN2 is better than CN1. For example, in contrast to CN1, the (very much) erroneous consensus translation "I have liked liked coffee" cannot be produced in CN2. More details regarding the edit distance alignment of multiple sequences can be found in Section III-F2.

E. Extracting Consensus Translation

In Section III-B, it was mentioned that each translation E_m for a sentence F is considered to be the primary hypothesis. Thus, we obtain a total of M confusion networks for each sentence. The consensus translation can be extracted only from one of these confusion networks, i.e., from the one in which the primary hypotheses was produced by a generally better performing MT system. However, the word order of the resulting consensus translation will follow the word order of the primary translation which may still be erroneous for some sentences. Because of that, a better strategy is to consider multiple primary hypotheses at once. Our experiments show that it is advantageous to unite the M confusion networks in a single lattice as shown in Fig. 5. Then, the consensus translation can be chosen from different alignment and reordering paths in this lattice.

The weighted majority voting on a single confusion network is straightforward and analogous to the ROVER system of [11]. First, we sum up the probabilities of the arcs which are labeled with the same word and have the same start and the same end state. More formally, this can be described as follows. For each state $s_k, k = 0, \ldots, K$ in the confusion network, we say that a word is at position k if an arc labeled with this word exists between the states s_{k-1} and s_k. Each word e_{mk} (including the empty word) at position k that is hypothesized by MT system m is assigned a weight γ_m. In our experiments, these weights give an *a priori* estimation of the translation quality of the MT system with the index m. These probabilities are adjusted based on the performance of the involved MT systems on a held-out development set in terms of an automatic MT evaluation measure. Generally, a better consensus translation can be produced if the words hypothesized by a better performing system get a higher probability. These global probabilities can also be interpolated with word- and position-specific confidence measures [42].

The probability for a unique word e to appear at position k is obtained with the following equation:

$$p_k(e|F) = \frac{\sum_{m=1}^{M} \gamma_m \cdot \delta(e_{mk}, e)}{\sum_{\tilde{e}} \sum_{m=1}^{M} \gamma_m \cdot \delta(e_{mk}, \tilde{e})}. \tag{6}$$

According to (6), the probability of a word e at position k is higher if the majority of the systems have produced e at this position.

Next, the consensus translation is extracted as the best path in the confusion network. The position-dependent probabilities $p_k(e|F)$ as given by (6) are used to score each path. We define the consensus translation as the sequence $\hat{e}_1^K := \hat{e}_1, \ldots, \hat{e}_k, \ldots, \hat{e}_K$[5] where, at each position k in the confusion network, the best word $\hat{e}_k$ is selected as given by the following equation:

$$\hat{e}_k = \underset{e}{\operatorname{argmax}} \{p_k(e|F)\}. \tag{7}$$

Note that the extracted consensus translation can be different from each of the original M translations.

In practice, the best path is extracted from the lattice which is a union of M confusion networks. Because of this, and also because ε-arcs are used, multiple identical word sequences can be extracted from the lattice. To improve the estimation of the score for the best hypothesis, we deviate from the (7) and sum the probabilities of identical partial paths. This is done through determinization of the lattice in the log semiring.[6] With this approach, we also can extract N-best hypotheses without duplicates. In a subsequent step, these N-best lists could be rescored with additional statistical models.

The lattice representing a union of several confusion networks can also be directly rescored with an n-gram language model (LM). The language models we used in our experiments are described in Section III-F1. In case of language model rescoring, a transformation of the lattice is required, since

[5]With the ε-arcs removed after extraction.

[6]Log semiring is a positive real semiring $(\mathbb{R} \cup \{-\infty, +\infty\}, \oplus_{\log}, +, +\infty, 0)$ with $a \oplus_{\log} b = -\log(e^{-a} + e^{-b})$.

LM history has to be memorized. The ε-arcs are removed as a result of this transformation. Therefore, the probabilities from (6) have to be redefined in terms of real words only and can be expressed with $p_i(e|F), i = 1, \ldots, i, \ldots, I$. Here, I is the length of a single full path in the transformed lattice. The probabilities $p_i(e|F)$ are then log-linearly interpolated with the language model probability. The following equation describes the modified decision criterion (7) when bigram LM probabilities $p_{\mathrm{LM}}(e_l|e_{l-1})$ are used:

$$(\hat{I}, \hat{e}_1^{\hat{I}}) = \underset{I, e_1^I}{\operatorname{argmax}} \left\{ \alpha^I \prod_{i=1}^{I} \left(p(e_i|F) \cdot p_{\mathrm{LM}}^{\lambda}(e_i|e_{i-1}) \right) \right\}. \tag{8}$$

Here, the maximization is performed over all of the paths in the LM-rescored lattice. λ is the LM scaling factor, and α is a word penalty that is used to avoid the bias towards short sentences. The parameters λ and α are optimized on the development set. Note again that (8) is an approximation, since in practice the probabilities are summed over identical partial paths when the rescored lattice is determinized.

F. Important Extensions

1) Improving Word Order: In most cases, the consensus translation produced as described in Section III-E is better than the individual system translations. This will be shown by experimental results in Section V. In particular, the lexical choice and thus the adequacy of the translations (i.e., the accuracy of the meaning they convey) improves dramatically. However, due to the nature of the proposed approach, the word order in the consensus translation sometimes may be even less correct than in the individual system translations. There are no constraints on the alignment-based reordering. This means that a good phrase translation may be broken up because it can happen that there is no good monotonic alignment of that phrase with the words in the primary translation. As a result, the system combination translation may possibly not be very fluent. For example, the fourth sentence in Fig. 3 is reordered to "I would you like have some coffee tea."

We have introduced several techniques to overcome this problem. First of all, we intend to avoid repetitions of identical words in the consensus translation. Such repetitions occur if two identical words from a secondary hypothesis E_m are aligned with the same word in the primary hypothesis E_n. This often happens with articles like "the," in cases when, e.g., the secondary system tends to overproduce the articles which are then all aligned to a single article from the primary translation. In order to avoid such word repetitions, we extend the simple algorithm for computing the alignment $\tilde{a}$ in (4) by introducing an additional constraint that identical words $e_{m,j} = e_{m,j'}$ in E_m cannot be all aligned to the same word $e_{n,i}$ in E_n. If two such connections are found, the one with the higher costs in the alignment cost matrix C is discarded (e.g., for $e_{m,j'}$) and another alignment point is determined. This is the point with the lowest costs in the same column of the matrix

$$\tilde{a}(j') = \underset{i' : i' \neq i}{\operatorname{argmin}} C(j', i'). \tag{9}$$

In combination with the extra alignment for insertions as illustrated by Fig. 4, this additional constraint helps to avoid almost all incorrect word repetitions in the produced consensus translations.

To further favor well-formed word sequences, we rescored the system combination lattice with a large n-gram language model ($n = 3$). However, no significant improvements in translation quality were achieved in our experiments. There may be several explanations for this fact, but we find the following the most probable. The confusion network, coupled with reordering of the secondary hypotheses, allows for many different (and mostly incorrect) word sequences. The LM trained on large amount of data can give high probabilities to n-grams in these sequences which are generally widely used, but have nothing to do with a correct translation of the particular source sentence.[7]

A novel idea which we tested experimentally was to train a trigram LM on the outputs of the systems involved in system combination. For LM training, we took the system hypotheses for the same test corpus for which the consensus translations are to be produced. Using this "adapted" LM for lattice rescoring thus gives bonus to n-grams from the original system hypotheses, in most cases from the original phrases. Presumably, many of these phrases have a correct word order, since they are extracted from the training data. Experimental results in Section V show that using this LM in rescoring together with a word penalty (to counteract any bias towards short sentences) notably improves translation quality, especially measured by automatic metrics sensitive to fluency like BLEU [31].

2) Handling of Long Sentences: System translations of long sentences pose a challenge for the presented system combination approach. In practice, the implementation of the alignment procedure described in Section III-B limits the maximum sentence length to 100 words. Also, rescoring "long" confusion networks for such sentences is computationally expensive due to the large number of paths which increases exponentially with sentence length.

To solve this problem, we developed a method for splitting long sentences based on punctuation information and monotone hypotheses alignment. For each sentence with the length of more than $L_{\max}$ words,[8] we first select a primary hypothesis among the system outputs. This can be the hypothesis of the generally best performing system. Then, we mark the split points in this hypothesis. We employ a recursive binary splitting algorithm that tries to find the best split at punctuation marks like period, comma, semicolon, etc. In case of multiple alternative split points, the algorithm selects the point which divides the considered word sequence in two parts of more or less the same size.

Then, we align the primary hypothesis with the other translations using the Levenshtein edit distance algorithm extended to multiple sequences. This approach is similar to the alignment used in [1]. First, we align two translations, then we align the third one to the alignment of the first two. If, e.g., a word in

[7]Preliminary experiments indicated that better consensus translations can be selected using the general LM scores in combination with the IBM model 1 scores which re-establish the dependency on the source sentence.

[8]We set $L_{\max} = 70$ in our experiments.

the third hypothesis is identical to one of the two words representing a substitution in the first alignment, this is considered a "match" with no costs. This alignment is not perfect because no reordering is considered. However, the punctuation marks (or the positions where they are missing in some of the systems) can be aligned rather well in most cases.

Finally, we transfer the split points from the primary hypothesis to the other ones based on the alignment and split all sentences. In this way, not only we simplify the processing of long sentences, but also enable the system combination approach to work on ASR output. In the TC-STAR project, the ASR output is provided to MT systems without sentence segmentation. Each of the individual MT systems uses an automatic algorithm to segment it into sentences and enrich with punctuation marks. Thus, the segmentation is different for different systems, and the sentence-level system combination approach presented here cannot be applied directly. However, if we consider each audio document (with several thousand words each) to be one single segment and split the system outputs as described above, we can compute the consensus translation based on this common automatic segmentation.

IV. Systems

In this section, we describe the individual MT systems of the TC-STAR project partners. All the systems used sentence-aligned Spanish–English transcripts of the European Parliament Plenary Sessions (EPPS) as bilingual training data. The outputs of these systems were used to create the consensus translations in our experiments. We begin with an overview of the features which all involved MT systems have in common.

A. Overview

All of the machine translation systems participating in the TC-STAR project are statistical MT systems. They all use the concept of *bilingual phrases* in order to better model the context dependency in the translation process. In this section, we will review the basic models used in state-of-the-art SMT systems.

In statistical machine translation, we are given a source language sentence $f_1^J = f_1 \dots f_j \dots f_J$, which is to be translated into a target language sentence $e_1^I = e_1 \dots e_i \dots e_I$. We directly model the posterior probability $Pr(e_1^I|f_1^J)$ with a log-linear model [28] and choose the translation with the highest probability according to the following decision criterion:

$$\begin{aligned}\hat{e}_1^I &= \operatorname{argmax}_{e_1^I} \Pr\left(e_1^I|f_1^J\right) \\ &= \operatorname{argmax}_{e_1^I} \left\{ \sum_{m=1}^{M} \lambda_m h_m\left(e_1^I, f_1^J\right) \right\}. \end{aligned} \tag{10}$$

In the following, we describe the basic models used as features $h_m(e_1^I, f_1^J)$ in (10). These models are used, with some modifications, by all of the TC-STAR machine translation systems. These models are: phrase translation model, single word-based translation model, target language model, as well as a word and/or phrase penalty. The scaling factors λ_1^M for the individual feature functions can be trained with respect to the final translation quality measured by some automatic error metric [30]. To this end, a development set is translated multiple times with different sets of scaling factors. The process is controlled by an optimization algorithm like the Downhill Simplex algorithm [33].

There exist several established search implementations for statistical phrase-based MT. Details can be found, e.g., in [45], [16], or [17].

1) Phrase-Based Model: To use bilingual phrase pairs in the translation model, we define a segmentation of a given sentence pair (f_1^J, e_1^I) into K nonempty nonoverlapping contiguous blocks

$$k \rightarrow s_k := (i_k; b_k, j_k), \text{ for } k = 1 \dots K. \tag{11}$$

Here, i_k denotes the last word 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 that is aligned to the kth target phrase; we set $j_0 := 0$. We constrain the segmentation so that all words in the source and the target sentence are covered by exactly one phrase.

For a given sentence pair (f_1^J, e_1^I) and a segmentation s_1^K, we define the bilingual phrases as

$$\tilde{e}_k := e_{i_{k-1}+1} \dots e_{i_k} \tag{12}$$

$$\tilde{f}_k := f_{b_k} \dots f_{j_k}. \tag{13}$$

Note that the segmentation s_1^K contains the information on the phrase-level reordering. It is introduced as a hidden variable in the translation model. Therefore, it would be theoretically correct to sum over all possible segmentations. In practice, we use the maximum approximation for this sum. As a result, the models $h(\cdot)$ in (10) depend not only on the sentence pair (f_1^J, e_1^I), but also on the segmentation s_1^K, i.e., we have models $h(f_1^J, e_1^I, s_1^K)$.

The phrase translation probabilities $p(\tilde{f}|\tilde{e})$ are estimated by relative frequencies

$$p(\tilde{f}|\tilde{e}) = \frac{N(\tilde{f}, \tilde{e})}{N(\tilde{e})}. \tag{14}$$

Here, $N(\tilde{f}, \tilde{e})$ is the number of co-occurrences of a phrase pair $(\tilde{f}, \tilde{e})$ in training. As in [45], we count all possible phrase pairs which are consistent with the word alignment. Two phrases are considered to be translations of each other, if the words are aligned only within the phrase pair and not to words outside. The word alignment is the IBM model 4 alignment [5] computed using the GIZA++ toolkit [29]. The marginal count $N(\tilde{e})$ in (14) is the number of occurrences of the target phrase $\tilde{e}$ in the training corpus. The resulting feature function is

$$h_{\text{Phr}}\left(f_1^J, e_1^I, s_1^K\right) = \log \prod_{k=1}^{K} p(\tilde{f}_k|\tilde{e}_k). \tag{15}$$

To obtain a more symmetric model, the inverse phrase-based model $p(\tilde{e}|\tilde{f})$ is also used.

2) Word-Based Lexicon Model: Relative frequencies are used to estimate the phrase translation probabilities (14). Most of the longer phrases occur only once in the training corpus. Therefore, pure relative frequencies overestimate the probability of those phrases. To overcome this problem, word-based

lexicon models are used to smooth the phrase translation probabilities.

The score of a phrase pair is computed in a way similar to the IBM model 1, but here, the summation is carried out only within a phrase pair and not over the whole target language sentence

$$h_{\mathrm{Lex}}\left(f_1^J, e_1^I, s_1^K\right) = \log \prod_{k=1}^{K} \prod_{j=b_k}^{j_k} \sum_{i=i_{k-1}+1}^{i_k} p(f_j|e_i). \quad (16)$$

The word translation probabilities $p(f|e)$ are estimated as relative frequencies from the word-aligned bilingual training corpus. The word-based lexicon model is also used in both directions $p(f|e)$ and $p(e|f)$.

3) Word and Phrase Penalty: Two simple heuristics, namely word penalty and phrase penalty, are often used in statistical phrase-based MT

$$h_{\mathrm{WP}}\left(f_1^J, e_1^I, s_1^K\right) = I, h_{\mathrm{PP}}\left(f_1^J, e_1^I, s_1^K\right) = K. \quad (17)$$

In practice, these two models affect the average sentence and phrase lengths. The model scaling factors can be adjusted to prefer longer sentences and longer phrases.

4) Target Language Model: A standard n-gram language model is an important feature of a statistical MT system. Its feature function is

$$h_{\mathrm{LM}}\left(f_1^J, e_1^I, s_1^K\right) = \log \prod_{i=1}^{I} p\left(e_i|e_{i-n+1}^{i-1}\right). \quad (18)$$

The smoothing technique applied is in most cases the modified Kneser–Ney discounting with interpolation.

B. Individual Systems

The individual systems participating in system combination all use the statistical models described in the previous section. Naturally, this introduces a limitation on the possible improvements in translation quality with the described system combination approach, since the output of these systems may often be similar. Nevertheless, the experimental results will show that the individual systems have enough modeling differences to make the system combination effective. The similarities and differences of the partner systems are described below.

1) IBM: Like all of the TC-STAR partner systems, the IBM statistical machine translation system is based on the log-linear model combination as given by (10). All basic models (see Section IV-A) are used. However, in contrast to most of the TC-STAR partners who trained the IBM model 4 [5] alignments with the GIZA++ toolkit, IBM trained an HMM-based alignment augmented by block acquisition algorithms [41], [18] and postprocessing [20]. Recently, the translation blocks derived from this alignment have been combined with those derived from the maximum entropy model-based word alignment [14].

IBM used reordering methods. One is local reordering based on part-of-speech (POS) templates to reorder consecutive word sequences [19]. The other method is nonlocal reordering based on parsing. The method is used to generate more accurate word order by reordering phrasal units identified by a parser in the preprocessing stage. This is done to capture the long-distance distortion between the source and the target language phrases. An HMM-based Spanish parser was developed and trained on a Spanish treebank containing about 90 k word tokens. The reordering rules for Spanish-to-English translation were manually acquired on the basis of error analysis on the TC-STAR 2005 development data set. The parsing-based reordering was applied only for the Spanish-to-English translation direction.

IBM used word 4-gram language models as well as word and POS trigram language models in translation.

2) FBK: The translations produced by the FBK (formerly ITC-irst) MT system were made using an open-source implementation of statistical phrase-based translation, called *Moses*. Moses had been developed at Johns Hopkins University during the JHU Summer Workshop 2006, with active participation of TC-STAR partners [17]. The Moses software implements a beam search decoder, including a log-linear phrase-based translation model able to process confusion networks. All of the basic models described in Section IV-A can be used in this implementation. For the TC-STAR evaluation, Moses was augmented with new data structures and training algorithms developed at FBK to handle large-scale language models [10]. Multiple language models were used already in the first-pass search; in particular, a very large 5-gram LM trained on the Gigaword corpus was used.

Additional models were also used in a *rescoring* step. To this end, N-best translations were produced for each sentence, supplied with individual model scores from the first-pass search ($N = 1000$). Then, these scores were combined with additional feature functions which were computed using the knowledge of the source sentence and a full translation hypothesis for this sentence. The scaling factors for all features were optimized on a development set. The additional models used by FBK include the IBM Model 1 and 3 word translation models, N-best rank of a hypothesis, and others.

In the evaluation, FBK focused on the ASR input condition, exploiting a more efficient search algorithm that allows to process large confusion networks [3]. The ASR output was also enriched with punctuation. This was achieved by inserting optional punctuation marks (periods, commas, etc.) into the confusion network. Thus, the decision whether or not to translate punctuation was left to the translation system.

3) LIMSI: LIMSI has participated only in the verbatim and ASR evaluation conditions for both translation directions. Like FBK, LIMSI has developed its phrase-based translation system using the open-source Moses decoder [17]. This baseline system was extended by innovative methods, in particular a continuous space target language model and word disambiguation using morpho-syntactic information.

a) Continuous Space Language Models: The LIMSI MT system uses a trigram back-off language model during decoding to generate N-best lists. These N-best lists are then rescored using a *continuous space* LM, also called *neural network* LM. The basic idea of this language model is to project the word indices onto a continuous space and to use a probability estimator operating on this space [2]. Since the resulting probability functions are smooth functions of the word representation, better generalization to unknown n-grams can be expected. A neural network can be used to simultaneously learn the projection of the words onto the continuous space and to estimate the n-gram

probabilities. This is still an n-gram approach, but the LM posterior probabilities are "interpolated" for any possible context of length n-1 instead of backing off to shorter contexts. This approach was successfully used in large-vocabulary continuous speech recognition [35].

LIMSI found that it was advantageous to only use a trigram during decoding, to generate 1000-best lists, and to rescore them with higher order back-off or the continuous space LMs. After rescoring, the coefficients of the eight feature functions were optimized using the publicly available CONDOR optimization toolkit [43]. The continuous space LM achieved significant improvements in the translation quality, in particular for the translation from English to Spanish. In addition, this approach showed a very good generalization behavior: the improvements obtained on the test data are as good or even exceed those observed on the development data.

b) Morpho-Syntactic Word Disambiguation: It is well known that syntactic structures vary greatly across languages. Spanish, for example, can be considered as a highly inflectional language, whereas inflection plays only a marginal role in English. LIMSI has investigated a translation model which enriches every word with its syntactic category. The enriched translation units are a combination of the original word and the POS tag. The translation system takes a sequence of enriched units as inputs and outputs. This implies that the test data must be POS tagged before translation. Likewise, the POS tags in the enriched output are removed at the end of the process to provide the final translation hypotheses which contain only a word sequence. This approach also allows to carry out a N-best reranking step using either a word-based or a POS-based language model [4], [36]. Combining morpho-syntactic word disambiguation with POS language model rescoring yielded a small improvement on the development data, but no significant change was observed on the test data. However, a human evaluation of some translations indicates that the proposed method seems to produce syntactically better output.

4) RWTH: The SMT system used at RWTH is based on the standard phrase-based log-linear statistical machine translation approach (see Section IV-A). Similar to most of the partners, a two-pass search is performed. In the first pass, the system generates the N-best list with a maximum of 10 000 hypotheses per sentence. In the second pass, the translation hypotheses are reranked using additional models.

In the first pass, the basic models described in Section IV-A were used. The language model was a 4-gram LM trained on the EPPS data only. In addition, phrase count features were employed. They allow to penalize or prefer phrase pairs that have frequencies above given thresholds. Three count features were used as described in [26]. Another extension was a trigram LM that had been trained using bilingual tuples. In contrast to the UPC approach (see Section IV-B6), the tuple LM was used to provide an additional score for each phrase pair that was determined with the standard approach described in Section IV-A1. The bilingual tuples were constructed using the word alignment within each phrase pair.

The second pass employed additional language models (clustered language models and sentence mixture language models) that model topic dependency of sentences. Also, an additional lexicon model together with additional word penalties and word deletion models were used. Finally, the sentence length and n-gram posterior probabilities [46] were included.

The ASR output was translated using a new method for punctuation prediction described in [25]. Having no punctuation in the input, the translation model and target language model are used to predict punctuation marks on the target side.

Part-of-speech-based reordering rules for nouns, adjectives, and adverbs were applied as a preprocessing step both in training and before translation. This helped to improve translation fluency for both translation directions.

5) UKA: The UKA phrase-based SMT system also uses the standard word alignment and translation models described in Section IV-A1. However, the phrase translation model is extended by smoothing of relative phrase frequencies with the method of Foster *et al.* [12]. The system also used two large language models, both trained on the same data: a 4-gram language model using modified Kneser–Ney smoothing, and a suffix array language model with arbitrary history lengths, with interpolation weights computed by minimum error training optimization towards BLEU [31].

Word reordering in the UKA system is accomplished by training reordering rules operating on part-of-speech tags of the source language side. In contrast to the manually derived rules used by IBM and RWTH, the reordering rules, or patterns, are automatically extracted from the bilingual word alignments. Prior to decoding, a fully reordered lattice is generated by applying these rules to the input utterance and inserting corresponding alternative paths annotated with distortion model scores. In the subsequent decoding step, a small additional local reordering window (size$=2$) is used.

6) UPC: UPC used the same system for all of the three translation conditions (see Section V-A). The SMT system of UPC is somewhat different from the other partner systems because it implements a translation model that is based on bilingual n-grams [21]. This translation model differs from the phrase-based translation model described in Section IV-A.1 in two basic issues: first, translation units are extracted from a monotonic segmentation of the training corpus; and, second, the model considers n-gram probabilities instead of relative frequencies. The tuple translation model, as well as other unique features of the UPC system, are described below.

a) Bilingual n-Gram Translation Model: The bilingual n-gram model constitutes a language model of bilingual units, referred to as tuples. This model approximates the joint probability between source and target language sentence by using 5-grams, as described by the following equation:

$$p((t,s)_1^K) \approx \prod_{k=1}^{K} p((t,s)_k | (t,s)_{k-1}, \ldots, (t,s)_{k-4}) \qquad (19)$$

where t refers to target, s to source, and $(t,s)_k$ to the kth tuple of a given bilingual sentence pair. Tuples are extracted from Viterbi alignments, which are automatically computed by using GIZA++ according to the following constraints: first, tuple extraction should produce a monotonic segmentation of bilingual sentence pairs; second, no smaller tuples can be extracted without violating the previous constraint.

TABLE I
CORPUS STATISTICS OF THE DEVELOPMENT AND TEST CORPORA (FINAL TEXT EDITIONS)

		EPPS English → Spanish		EPPS Spanish → English		CORTES Spanish → English	
Dev:	Sentences	1 122		699		753	
	Running Words	28 390	30 503	24 275	25 240	27 707	29 617
	Running Words without Punct. Marks	25 853	27 807	21 707	22 794	25 101	26 783
	Vocabulary Size	4 139	4 886	4 376	3 582	4 479	3 480
Test:	Sentences	1 130		828		642	
	Running Words	27 278	25 745	28 015	25 137	27 470	24 993
	Running Words without Punct. Marks	24 712	25 662	25 185	25 044	25 172	24 911
	Vocabulary Size	3 723	5 695	4 719	4 914	4 067	4 261

b) Spanish Morphology Reduction: The UPC n-gram based SMT system implements a morphology reduction of the Spanish language as a preprocessing step. As a consequence, training data sparseness due to Spanish morphology is reduced improving the overall performance of the translation system. In the case of English-to-Spanish translation, a postprocessing stage for Spanish morphology generation is required. This stage is implemented by means of a morphology reconstruction model which uses information about English and Spanish base forms as well as English full forms to infer the most probable Spanish full form [8].

c) Word Reordering Strategy: The UPC translation system uses a POS-based word reordering strategy. During training, reordering patterns are identified by looking at the alignment link crossings occurring in the bilingual corpus. Such patterns are then classified according to the corresponding POS-tags of the source words involved. Afterwards, all link crossings are unfolded by reordering the source words while keeping the target side of the corpus untouched. Then, from this new source-reordered bilingual corpus, translation tuples are extracted, and their n-gram model probabilities are trained. Finally, during the translation step, the input sentence is replaced by a word graph including all alternative paths provided by the POS reordering patterns learned during training. This word reordering procedure is further described in [7].

d) POS-Based Target Language Model: The UPC translation system also employs a 5-gram language model trained on target POS tags. In order to incorporate this model in the search, each bilingual tuple has to be extended to a triplet by adding a POS tag sequence corresponding to the words in the target part of the tuple. This POS information is used by the decoder only to score the alternative POS tag sequences associated with the competing partial translation hypotheses.

V. EXPERIMENTS

In this section, we will describe the experimental results for the presented system combination algorithm combining output from the TC-STAR partner systems described in the previous section.

A. Translation Conditions and Data

We evaluate consensus translations of the partner systems described in Section IV for both Spanish-to-English and English-to-Spanish translation directions. The official evaluation guidelines defined three conditions, or types of input. The first one is final text editions (FTE)—transcripts of speeches made in the European Parliament Plenary sessions which were additionally cleaned by professional editors to remove colloquial expressions, hesitations, repetitions, etc. The second one is verbatim transcriptions—original manually created transcripts that reflect exactly what has been said by a parliament speaker. The third condition is the output of an automatic speech recognition system (ASR) for the same speech segments for which the verbatim transcriptions had been produced.

For Spanish-to-English translations, apart from using the EPPS speeches as evaluation data, several speeches made in the Spanish parliament (the so called CORTES transcripts) were also translated under the above-mentioned conditions.

For the *primary track* considered in this paper, each of the involved systems used the manual transcripts of the EPPS sessions from 1993 to May 2006 and their manual translations as bilingual training data (about 37M running words). Additional monolingual data could be used for language modeling. Some systems, e.g., IBM, could achieve better translation results in the *secondary track*, where any publicly available bilingual and monolingual data could be used [38].

Table I gives an overview of the official 2007 TC-STAR evaluation development and test data used in our experiments. The statistics are given for the final text editions only, since the statistics for verbatim/ASR conditions for the same speeches are quite similar. For the target language, the statistics for one of the manual reference translations are specified.

B. Evaluation Criteria

Well-established automatic evaluation measures like the BLEU score [31], word error rate (WER), position-independent word error rate (PER, [40]), and the NIST score [9] were calculated to assess the translation quality. All measures were computed with respect to two available reference translations. For the ASR condition, the sentence segmentation of the hypotheses and the consensus translation was performed automatically and thus did not correspond to the sentence segmentation in the reference translations. The tool described in [23] was used to resegment the hypotheses based on the optimal edit distance alignment with the multiple reference translations. Then, the usual automatic error measures were computed.

Since the system combination algorithm takes lowercase translations as input and produces lowercase translations, we

TABLE II
INFLUENCE OF INDIVIDUAL SYSTEM COMBINATION COMPONENTS ON THE QUALITY OF THE CONSENSUS TRANSLATION (TC-STAR 2007 TEST SET, ENGLISH-TO-SPANISH TRANSLATION DIRECTION, VERBATIM CONDITION)

System	BLEU[%]	WER[%]	PER[%]	NIST
worst single system	49.3	39.8	30.0	9.95
best single system	52.4	36.7	27.9	10.45
consensus translation:				
single primary (uniform weights)	53.0	35.3	27.1	10.60
+ manual weight optimization	53.4	35.5	27.0	10.62
+ union of confusion networks	53.8	35.6	26.8	10.60
+ adapted LM	54.3	35.2	27.4	10.65
+ automatic weight optimization	54.5	35.5	27.5	10.62

TABLE III
INFLUENCE OF LANGUAGE MODEL RESCORING ON THE QUALITY OF THE SYSTEM COMBINATION TRANSLATION (TC-STAR 2007 TEST SET, ENGLISH-TO-SPANISH TRANSLATION DIRECTION, VERBATIM CONDITION)

System	BLEU[%]	WER[%]	PER[%]	NIST
best single system	52.4	36.7	27.9	10.45
selection (with rescoring)	53.6	36.5	27.8	10.50
consensus translation	53.8	35.6	26.8	10.60
rescoring (general LM)	53.7	36.2	27.4	10.49
rescoring (adapted LM)	54.3	35.2	27.4	10.65

report case-insensitive evaluation results to factor out the effect of truecasing of the English words from the effect of computing a consensus translation.

In addition to automatic evaluation, the TC-STAR 2007 evaluation also included human evaluation of English-to-Spanish translations by native speakers of Spanish. A translated sentence was judged in terms of adequacy and fluency by two evaluators, see [38] for details.

C. Results

1) Comparative Experiments: First, we performed comparative experiments to evaluate the influence of individual system combination features on MT quality. We report the results for these experiments for the translations of the Verbatim EPPS data from English to Spanish. Experiments for the other evaluation conditions lead to similar conclusions.

Table II presents the automatic MT error measures for translations of the evaluation data from the official TC-STAR 2007 evaluation. The six TC-STAR partner systems described in Section IV have submitted their translation output for the evaluation. Using these translations, a consensus translation was determined with the approach described in this article.

In the first (baseline) experiment, we created only one confusion network, using the enhanced alignment procedure and reordering as described in Section III. The translation hypothesis of the best performing system[9] was taken as the primary hypothesis. We used a uniform distribution for the global system probabilities, i.e., the consensus word at a given position was selected by a simple majority voting. In case of a tie, when, for example, two alternative words were used by three systems each, the preference was given to the word used by the best performing system. From Table II, it is clear that this setup already resulted in a substantial improvement of all error measures, e.g., 0.6% absolute in BLEU and 1.4% absolute in word error rate.

In the second experiment, we tuned the six system weights manually on a separate development set. The resulting weight distribution was only marginally different from the uniform distribution, but a slightly higher weight was given to the two systems with the highest BLEU score. Using manually tuned weights has improved the BLEU score by another 0.4% absolute.

In the next experiment, we combined the six confusion networks as described in Section III-E. We observed a slight improvement in the BLEU score and PER, which shows that the lexical choice in the consensus translation has improved. We attribute this to the fact that the quality of alignment and thus of the "voting" on the confusion network depends on the choice of the primary hypothesis. When all possible primary hypotheses are considered, the algorithm takes advantage of the "best" one on a sentence-by-sentence basis.

Another important improvement is language model rescoring. We used a trigram language model trained on the six system translations for each of the 1130 evaluation data sentences (i.e., on 6780 sentences). As explained in Section III-F1, we expected a language model trained on the systems' translations to give preference to the n-grams from the original phrases produced by the involved MT systems. Indeed, we observed an absolute improvement of e.g., 0.5% in BLEU by rescoring the union of confusion networks with this type of language model. It is worth noting that no improvement through LM rescoring was obtained when only one confusion network was used. This shows that the special language model has the power to discriminate between translations with good and bad word order. For this experiment, the system weights, the scaling factor for the language model, and the word penalty were tuned manually on the development set.

Table III compares the LM rescoring experiment from Table II with the rescoring by a regular 3-gram LM trained on the Spanish part of the bilingual training data. We observed that rescoring with a general LM did not improve the translation results. The added LM scores could not improve the word order of the consensus translations; this is reflected by the automatic metrics like BLEU and WER which are sensitive to fluency.

Table III also presents the results of a comparative experiment, in which we select one of the individual system translations by rescoring. We created a word lattice with only six paths representing the original system translations and scored this lattice with system weights, the adapted LM and the word penalty. The model weights were optimized on a development set separately for this experiment. From the results in Table III, we see that although this approach improves the overall translation quality in comparison with the best single system, it is inferior to the presented approach in which a consensus translation is computed. This is an expected result, since the selection approach is not able to generate output different from the individual system translations.

Finally, we were further able to slightly improve the BLEU score of the system combination translation by optimizing the parameters automatically on the development set (see Table II). The global system probabilities, as well as the LM factor and word penalty, were optimized for BLEU using the CONDOR

[9]as determined on a held-out development set

optimization tool [43]. For the optimization, the confusion networks can be kept fixed, since the parameters involved do not affect the alignment. In each step of the optimization algorithm, the confusion networks are scored using a set of system weights and the adapted LM, and the consensus translation is extracted by finding the best path through the rescored lattice. Then, the parameters are updated. The optimization algorithm converges after about 100–150 iterations.

Although the automatic optimization resulted in significant improvements in BLEU on the development set, this improvement was not significant on the evaluation data, and the other measures did not improve at all. We attribute this to overfitting: in fact, the weight for two of the involved systems was automatically determined to be very small. This can happen if a system's output for a subset of sentences is very similar. If this similarity diminishes on another data set, the role of these systems in determining the consensus translation may be underestimated.

While it may be of value to analyze the contribution of each participating system to the final system combination translation, we should keep in mind that the system combination approach in many cases produces a new translation which is different from each of the original hypotheses. Our experiments show that the quality of this new translation is often significantly better than any of the original translations. Thus, on the English-to-Spanish verbatim evaluation data, for 582 out of 1167 sentences new translations were generated.[10] Considering only these 582 sentences, the improvement due to the construction of a new consensus translation turned out to be from 49.1 to 50.7% in BLEU, whereas for the remaining 585 sentences, the improvement due to the mere selection of the "best" hypothesis was smaller: from 57.6% to 58.5%. Note that a genuine consensus translation is most often generated for sentences which are harder to translate.

In the next experiment, we tried to quantify the potential translation quality improvement that could be achieved with the presented system combination approach. To this end, we selected a subset of 300 sentences from the official 2007 TC-STAR evaluation data, Spanish-to-English verbatim condition. We then let human experts with fluent knowledge of English put together the "consensus" translation. They had access neither to the source sentences nor to the reference translations, but were given only the six system translations for each sentence. Also, the experts were only allowed to use the words appearing in the original system translations. This means that the produced human system combination hypothesis can be viewed as an upper bound for the performance of the automatic system combination approach.[11]

Table IV shows the MT error measures for this experiment. We see that the improvement due to the automatic system combination w.r.t. the best single system is similar to the improvement on the full evaluation data set (see Table VII). Naturally, the human system combination exhibits the best performance; however, automatic system combination is able to explore more than one fourth of this potential. We conclude from this result that the automatic system combination is an effective method, although refinements of the algorithm could have the potential to improve the translation quality.

[10]For each of the remaining sentences, the consensus translation turned out to be identical to one of the systems' translation.

[11]In practice, this upper bound cannot be even theoretically reached in every case because, e.g., a human can delete a word present in every system translation.

TABLE IV
POTENTIAL OF THE PRESENTED APPROACH (A SUBSET OF THE OFFICIAL TC-STAR 2007 EVALUATION DATA, SPANISH-TO-ENGLISH TRANSLATION DIRECTION, VERBATIM CONDITION)

System	BLEU[%]	WER[%]	PER[%]	NIST
worst single system	52.0	35.8	27.2	9.33
best single system	54.1	34.2	25.5	9.47
system combination	55.2	32.9	25.1	9.63
"human" system combination	58.2	31.5	24.3	9.85

TABLE V
TC-STAR 2007 EVALUATION RESULTS FOR THE ENGLISH-TO-SPANISH TRANSLATION DIRECTION

Input	System	BLEU[%]	WER[%]	PER[%]	NIST
ASR (WER: 6.9%)	syscombi	42.1	44.1	35.1	9.34
	RWTH	41.1	47.0	35.7	9.07
	FBK	40.8	45.4	35.7	9.19
	LIMSI	39.8	46.5	36.3	9.11
	IBM	38.8	49.1	37.0	8.83
	UKA	38.1	46.0	37.1	8.89
	UPC	38.0	48.3	37.4	8.84
Verbatim	syscombi	54.5	35.5	27.5	10.62
	FBK	52.4	36.7	27.9	10.45
	LIMSI	52.4	37.2	28.0	10.35
	RWTH	51.8	37.9	28.8	10.31
	UKA	51.7	36.4	28.2	10.40
	UPC	49.9	39.2	29.8	10.07
	IBM	49.3	39.8	30.0	9.95
Text	syscombi	56.9	33.3	26.3	10.81
	UKA	55.1	34.6	27.1	10.61
	FBK	54.3	34.6	27.3	10.59
	UPC	54.2	35.4	27.8	10.49
	RWTH	53.3	36.2	27.8	10.43
	IBM	52.0	37.6	29.1	10.17

2) TC-STAR Evaluation Results: Tables V–VII show the TC-STAR evaluation results for all of the participating partner systems, as well as for the presented system combination approach. The participating systems are sorted descending based on their performance in terms of the BLEU evaluation measure.

For system combination, all of the additional features described in Section III-F were used so that the result in Table V corresponds to the one in the last line of Table II. The system weights and the LM and word penalty scaling factors were optimized automatically, separately for the FTE and verbatim conditions. For the ASR condition, the optimal parameters for the verbatim condition were used. However, in some cases the optimization did not result in an additional improvement on the development set; in this case, we used manually tuned parameters.

From Table V one can infer that system combination performed especially well on the English text input, with a 1.8% absolute improvement in BLEU. For the other conditions, the improvement is also significant. In particular, system combination was helpful for the hardest condition of translating automatically recognized speech. Similar conclusions can be drawn from Tables VI and VII which show the performance of the partner systems and the system combination algorithm for the Spanish-to-English translation direction. For translations of the

TABLE VI
TC-STAR 2007 EVALUATION RESULTS FOR THE SPANISH-TO-ENGLISH TRANSLATION DIRECTION (CORTES TASK)

Input	System	BLEU[%]	WER[%]	PER[%]	NIST
ASR	syscombi	37.8	51.3	34.1	9.03
(WER: 8.8%)	FBK	37.2	52.3	34.8	8.91
	UKA	36.6	51.1	34.3	8.97
	LIMSI	36.5	51.8	34.4	8.88
	IBM	36.3	52.3	34.0	8.85
	RWTH	36.0	53.8	36.4	8.72
	UPC	35.2	53.9	36.1	8.66
Verbatim	syscombi	49.1	41.2	28.5	10.17
	RWTH	48.2	41.9	29.1	10.02
	UKA	48.1	41.3	28.6	10.14
	IBM	47.3	42.5	29.0	9.93
	FBK	47.3	43.1	29.6	9.89
	LIMSI	47.1	42.2	28.5	9.97
	UPC	46.6	43.5	30.7	9.80
Text	syscombi	47.2	43.0	29.5	9.96
	UKA	46.0	43.6	29.9	9.85
	FBK	46.0	44.5	30.5	9.79
	IBM	45.7	44.3	30.1	9.79
	RWTH	44.9	44.8	30.9	9.65
	UPC	44.6	45.6	31.9	9.56

TABLE VII
TC-STAR 2007 EVALUATION RESULTS FOR THE SPANISH-TO-ENGLISH TRANSLATION DIRECTION (EPPS TASK)

Input	System	BLEU[%]	WER[%]	PER[%]	NIST
ASR	syscombi	46.3	42.2	29.4	10.01
(WER: 5.9%)	FBK	44.8	43.9	30.5	9.81
	LIMSI	44.7	43.3	30.0	9.84
	IBM	44.5	43.9	30.2	9.76
	UKA	43.8	43.1	30.5	9.84
	UPC	43.4	44.8	31.9	9.65
	RWTH	43.1	45.3	31.9	9.56
Verbatim	syscombi	55.3	34.5	24.8	10.90
	IBM	54.4	35.5	25.1	10.77
	UKA	54.1	35.3	25.4	10.78
	LIMSI	53.5	35.9	25.0	10.69
	RWTH	53.1	36.3	25.7	10.64
	FBK	52.8	37.2	26.1	10.55
	UPC	52.2	37.2	26.7	10.52
Text	syscombi	55.4	36.3	25.4	10.88
	UKA	54.0	37.1	26.1	10.74
	IBM	53.5	37.8	26.3	10.67
	FBK	53.3	38.2	26.5	10.61
	RWTH	52.1	38.4	27.0	10.49
	UPC	51.9	38.9	27.6	10.44

out-of-domain CORTES data, all of the evaluation measures are lower than when the EPPS data is translated; see Table VI. However, the relative improvement due to system combination can be still observed. For translations of the EPPS data, the largest improvement is gained by producing a consensus translation for the ASR output condition.

On a representative subset (400 segments or 30%) of the English-to-Spanish evaluation data, a human evaluation was performed. Table VIII lists the average human scores for translation adequacy and fluency. The individual systems have similar average scores, although their outputs, of course, differ on a sentence-by-sentence basis. The system combination output is judged to be the best in terms of adequacy, and performs on the same level as the best individual system in terms of fluency.

TABLE VIII
HUMAN EVALUATION RESULTS ON A SUBSET OF THE VERBATIM ENGLISH-TO-SPANISH EVALUATION DATA, IN TERMS OF AVERAGE FLUENCY AND ADEQUACY SCORES

system	adequacy	fluency	BLEU [%]
IBM	3.54	3.24	48.87
FBK	3.60	3.35	51.26
LIMSI	3.57	3.32	51.35
RWTH	3.61	3.39	51.18
UKA	3.64	3.31	51.15
UPC	3.62	3.25	48.34
system combination	3.71	3.37	53.14
reference translation	4.39	4.24	100.0

The human evaluation has shown that the individual partner systems have a similar level of performance. The automatic error measures tell us the same: e.g., the difference in the BLEU score between the participating systems rarely exceeds 3% absolute. This is one of the prerequisites for good performance of the presented system combination algorithm. In general, our experiments show that the algorithm presented in this paper can obtain significant translation quality improvements with the produced consensus translation if the following criteria are (roughly) satisfied.

- The majority of participating systems have a similar level of performance as measured by (automatic) evaluation measures. If this is not the case, i.e., if, for example, only one system performs well, and two or three others are inferior to it, the words from the weaker systems will "outweigh" the words from the good-quality system.
- In spite of the similar performance level, the translations should be substantially different, i.e., different systems should ideally make different errors. Unfortunately, this requirement is quite hard to satisfy, especially if the individual systems already produce good-quality translations, which is the case for the TC-STAR systems. Nevertheless, for many sentences, in particular for sentences with unusual word choice and/or structure not observed in training, the systems do make different errors, and the consensus translation is able to effectively avoid them.
- When global system weights are used for scoring the confusion network, as it was the case for the TC-STAR evaluation, at least three systems are needed for the algorithm to work. The algorithm could be used with two systems only with word-specific confidence measures. However, the more systems are used, the better is the quality of the consensus translation.

Table IX shows examples of how the translation quality can be improved with system combination. Here, the consensus translation is compared with the translation of the best individual system, as well as with a human reference translation.

VI. CONCLUSION

In this paper, we described a comprehensive system combination approach for machine translation. It includes the enhanced

TABLE IX
EXAMPLES OF TRANSLATION QUALITY IMPROVEMENTS RESULTING FROM SYSTEM COMBINATION (SPANISH-TO-ENGLISH VERBATIM EVALUATION CONDITION)

single MT	it is the time to act and to make further forward the declarations of intent.
consensus MT	it is the time to act and leave for later the declarations of intent.
reference	it is time to act and to leave the declarations of intentions for the future.
single MT	history does not accept ... nor can expect us to other things happen.
consensus MT	history does not accept ... nor allows us to wait for other things happen.
reference	history does not accept ... nor allows us to wait for other things to take place.
single MT	we are seeing new forms of extremism wing of concern to all democrats.
consensus MT	we are seeing new forms of right-wing extremism concern to all democrats.
reference	we are observing new forms of right-wing extremism that worry all democrats.
single MT	are not two or three, ... they are transported , fifty-three companies.
consensus MT	they are not two or three, ... they are five hundred fifty three companies.
reference	they are not two or three, ... they are five hundred and fifty-three companies.
single MT	the commercial sky last month of July was clouded on Geneva.
consensus MT	the commercial sky is clouded on Geneva last July.
reference	the trade sky clouded over Geneva last month of July.

alignment procedure of Matusov *et al.* [24], as well as several novel and important extensions. For the first time, we formulated the theoretical basis for the developed system combination approach.

In contrast to previous approaches to system combination in MT, the presented method includes unsupervised training of the alignment between the translation hypotheses. The decision on how to align two translations of a sentence takes the context of a whole document into account. The high-quality alignment is used to reorder all but one of the translation hypotheses, which is considered to be the primary translation with correct word order. From the primary translation and the reordered secondary translations aligned to it in a monotone way, a confusion network is constructed. Since each of the M translations may have a good word order, we build M confusion networks and combine them in one lattice. The consensus translation is extracted from this lattice by weighted majority voting. A good-quality translation can be extracted, which is often different from each of the original translations. We showed that translation quality can be further improved by including a special language model to rescore this lattice. The language model is trained on the outputs of the individual translation systems on a test corpus in order to give bonus to the original phrases.

The quality of the produced consensus translations was evaluated in the TC-STAR speech translation project in the year 2007. We combined the outputs of all TC-STAR partner systems. These statistical phrase-based MT systems were used to translate speeches made in the European and Spanish parliaments. In this paper, we gave an overview of the base MT models used by all the partner systems. We also described the unique features of each participating system.

We were able to show experimentally that the presented system combination approach significantly improves translation quality. An improvement in all automatic and human evaluation measures was observed under all evaluation conditions for the Spanish-to-English and English-to-Spanish translation directions. We also performed a thorough analysis of how individual features of the algorithm influence the translation quality, and compared the overall performance with a reasonable upper bound given by the manually produced system combination translations.

In the future, we would like to include sophisticated word confidence estimations in the voting procedure, as well as further improve the alignment and language model rescoring steps by explicitly considering phrases and other syntactic and semantic structures.

ACKNOWLEDGMENT

The participating groups would like to thank and acknowledge the work of all their translation team members.

REFERENCES

[1] S. Bangalore, G. Bordel, and G. Riccardi, "Computing consensus translation from multiple machine translation systems," in *Proc. ASRU*, Madonna di Campiglio, Italy, Dec. 2001, CD-ROM.

[2] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, no. 2, pp. 1137–1155, 2003.

[3] N. Bertoldi, R. Zens, and M. Federico, "Speech translation by confusion network decoding," in *Proc. ICASSP*, Honolulu, HI, Apr. 2007, pp. IV-1297–IV-1300.

[4] H. Bonneau Maynard, A. Allauzen, D. Déchelotte, and H. Schwenk, "Combining morphosyntactic enriched representation with N-best reranking in statistical translation," in *Proc. NAACL-HLT Workshop Syntax and Structure in Statistical Translation*, Rochester, NY, Apr. 2007, pp. 65–71.

[5] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer, "The mathematics of statistical machine translation: Parameter estimation," *Comput. Ling.*, vol. 19, no. 2, pp. 263–311, 1993.

[6] C. Callison-Burch and R. Flournoy, "A program for automatically selecting the best output from multiple machine translation engines," in *Proc. Mach. Translation Summit VIII*, Sep. 2001, pp. 63–66.

[7] J. M. Crego and J. B. Mariño, "Improving SMT by coupling reordering and decoding," *Mach. Translation*, vol. 20, no. 3, pp. 199–215, Sep. 2006.

[8] A. de Gispert, "Introducing linguistic knowledge into statistical machine translation," Ph.D. dissertation, Universitat Politécnica de Catalunya, Barcelona, Spain, Oct. 2006.

[9] G. Doddington, "Automatic evaluation of machine translation quality using n-gram co-occurrence statistics," in *Proc. ARPA Workshop Human Lang. Technol.*, San Diego, CA, Mar. 2002, pp. 128–132.

[10] M. Federico and M. Cettolo, "Efficient handling of N-gram language models for statistical machine translation," in *Proc. ACL 2007, 2nd Workshop Statist.Mach. Translation*, Prague, Czech Republic, Jun. 2007.

[11] J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)," in *Proc. ASRU*, Santa Barbara, CA, Dec. 1997, pp. 347–354.
[12] G. Foster, R. Kuhn, and H. Johnson, "Phrasetable smoothing for statistical machine translation," in *Proc. EMNLP'06*, Jul. 2006, pp. 53–61.
[13] R. Frederking and S. Nirenburg, "Three heads are better than one," in *Proc. 4th Conf. Appl. Natural Lang. Process.*, Stuttgart, Germany, 1994, pp. 95–100.
[14] A. Ittycheriah and S. Roukos, "A maximum entropy word aligner for Arabic–English machine translation," in *Proc. HLT/EMNLP*, Vancouver, BC, Canada, Oct. 2005, pp. 89–96.
[15] S. Jayaraman and A. Lavie, "Multi-engline machine translation guided by explicit word matching," in *Proc. 10th Conf. Eur. Assoc. Mach. Translation*, Budapest, Hungary, 2005, pp. 143–152.
[16] P. Koehn, "Pharaoh: a beam search decoder for phrase-based statistical machine translation models," in *Proc. AMTA*, Washington, DC, 2004, pp. 115–124.
[17] P. Koehn *et al.*, "Moses: Open source toolkit for statistical machine translation," in *Proc. ACL 2007, Volume on Poster and Demo Sessions*, Prague, Czech Republic, Jun. 2007, pp. 177–180.
[18] Y.-S. Lee, "IBM statistical machine translation for spoken languages," in *Proc. IWSLT*, Pittsburgh, PA, Oct. 2005, pp. 86–93.
[19] Y.-S. Lee, "Morpho-syntax in statistical machine translation," presented at the TC-STAR OpenLab, Trento, Italy, Mar. 2006, unpublished.
[20] Y.-S. Lee, S. Roukos, Y. Al-Onaizan, and K. Papineni, "IBM spoken language translation system," in *Proc. TC-STAR Workshop Speech to Speech Translation*, Barcelona, Spain, Jun. 2006, pp. 13–18.
[21] J. B. Mariño, R. E. Banchs, J. M. Crego, A. de Gispert, P. Lambert, J. A. R. Fonollosa, and M. R. Cost-jussá, "N-gram based machine translation," *Comput. Linguistics*, vol. 32, no. 4, pp. 527–549, 2006.
[22] E. Matusov, R. Zens, and H. Ney, "Symmetric word alignments for statistical machine translation," in *Proc. COLING*, Geneva, Switzerland, 2004, pp. 219–225.
[23] E. Matusov, G. Leusch, O. Bender, and H. Ney, "Evaluating machine translation output with automatic sentence segmentation," in *Proc. IWSLT'05*, Pittsburgh, PA, Oct. 2005, pp. 148–154.
[24] E. Matusov, N. Ueffing, and H. Ney, "Computing consensus translation from multiple machine translation systems using enhanced hypotheses alignment," in *Proc. EACL*, Trento, Italy, Apr. 2006, pp. 33–40.
[25] E. Matusov, A. Mauser, and H. Ney, "Automatic sentence segmentation and punctuation prediction for spoken language translation," in *Proc. IWSLT*, Kyoto, Japan, Nov. 2006, pp. 158–165.
[26] A. Mauser, R. Zens, E. Matusov, S. Hasan, and H. Ney, "The RWTH statistical machine translation system for the IWSLT 2006 evaluation," in *Proc. IWSLT*, Kyoto, Japan, Nov. 2006, pp. 103–110.
[27] T. Nomoto, "2004. Multi-engine machine translation with voted language model," in *Proc. ACL*, Barcelona, Spain, 2004, pp. 494–501.
[28] F. J. Och and H. Ney, "Discriminative training and maximum entropy models for statistical machine translation," in *Proc. ACL*, Philadelphia, PA, Jul. 2002, pp. 295–302.
[29] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Comput. Linguistics*, vol. 29, no. 19, pp. 51–60.
[30] F. J. Och, "Minimum error rate training in statistical machine translation," in *Proc. ACL*, Sapporo, Japan, Jul. 2003, pp. 160–167.
[31] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A Method for automatic evaluation of machine translation," in *Proc. ACL*, Philadelphia, PA, Jul. 2002, pp. 311–318.
[32] M. Paul, T. Doi, Y. Hwang, K. Imamura, H. Okuma, and E. Sumita, "Nobody is perfect: ATR's hybrid approach to spoken language translation," in *Proc. IWSLT*, Pittsburgh, PA, 2005, pp. 55–62.
[33] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C++*. Cambridge, U.K.: Cambridge Univ. Press, 2002.
[34] A.-V. Rosti, N. F. Ayan, B. Xiang, S. Matsoukas, R. Schwartz, and B. Dorr, "Combining outputs from multiple machine translation systems," in *Proc. NAACL-HLT*, Rochester, NY, Apr. 2007, pp. 228–235.
[35] H. Schwenk, "Continuous space language models," *Comput. Speech Lang.*, vol. 21, pp. 492–518, 2007.
[36] H. Schwenk, D. Déchelotte, H. Bonneau Maynard, and A. Allauzen, "Modèles statistiques enrichis par la syntaxe pour la traduction automatique," *Traitement Automatique du Langage Naturel*, pp. 253–262, Jun. 2007.
[37] M. Snover, B. Dorr, R. Schwartz, L. Micciula, and J. Makhoul, "2006. A study of translation edit rate with targeted human evaluation," in *Proc. AMTA*, Cambridge, MA, 2006, pp. 223–231.
[38] "European Research Project Tc-Star—technology and corpora for speech-to-speech translation," Deliverable D30—Evaluation Rep. [Online]. Available: http://www.tc-star.org
[39] D. Tidhar and U. Küssner, "Learning to select a good translation," in *Proc. COLING*, 2000, pp. 843–849.
[40] C. Tillmann, S. Vogel, H. Ney, A. Zubiaga, and H. Sawaf, "Accelerated DP based search for statistical translation," in *Proc. Eur. Conf. Speech Commun. Technol.*, Rhodes, Greece, Sep. 1997, pp. 2667–2670.
[41] C. Tillmann, "2003. A projection extension algorithm for statistical machine translation," in *Proc. EMNLP*, Sapporo, Japan, Jul. 2003, pp. 1–8.
[42] N. Ueffing and H. Ney, "2005 Word-level confidence estimation for machine translation using phrase-based translation models," in *Proc. HLT/EMNLP*, Vancouver, BC, Canada, Oct. 2005, pp. 763–770.
[43] F. Vanden Berghen and H. Bersini, "CONDOR, a new parallel, constrained extension of powell's UOBYQA algorithm: Experimental results and comparison with the DFO Algorithm," *J. Comput. Appl. Math.*, vol. 181, pp. 157–175, 2005.
[44] S. Vogel, H. Ney, and C. Tillmann, "1996. HMM-based word alignment in statistical translation," in *Proc. COLING*, Copenhagen, Denmark, Aug. 1996, pp. 836–841.
[45] R. Zens and H. Ney, "Improvements in phrase-based statistical machine translation," in *Proc. HLT*, Boston, MA, May 2004, pp. 257–264.
[46] R. Zens and H. Ney, "N-gram posterior probabilities for statistical machine translation," in *Proc. NAACL-HLT 2006 Workshop Statist. Mach. Translation*, New York, Jun. 2006, pp. 72–77.

Evgeny Matusov (S'07) received the degree in computer science from RWTH Aachen University, Aachen, Germany, in 2003. He is currently pursuing the Ph.D. degree in the Department of Computer Science, RWTH Aachen University.

His research interests are in the area of statistical machine translation of text and speech, with the focus on combination of speech recognition and machine translation systems. He was the author or coauthor of more than 20 reviewed publications in international conferences.

Mr. Matusov was the recepient of the ISCA Best Student Paper Award in 2005.

Gregor Leusch received the degree in computer science from RWTH Aachen University, Aachen, Germany, in 2005, having written his diploma thesis on automatic evaluation measures for machine translation. He is currently pursuing the Ph.D. degree in the Department for Computer Science, RWTH Aachen University. His current research topics are the automatic evaluation and system combination for machine translation. The two topics share, in his view, a remarkably similar set of challenges compared to their speech recognition counterparts.

He is author or coauthor of several reviewed publications in international conferences.

Rafael E. Banchs received the B.S. and M.Sc. degrees in electronic engineering from Universidad Simón Bolívar, Caracas, Venezuela, in 1991 and 1993, respectively, and the Ph.D. in electrical engineering from the University of Texas, Austin, in 1998.

He then joined Intevep, the main research center of Venezuelan oil industry, where his research activities were focused on developing new technologies for data classification and parameter estimation in the context of oil reservoir characterization. In 2003, he was awarded a "Ramón y Cajal" fellowship from the Spanish Ministry of Education and Science; and, in 2004, he joined the TALP Research Center, Universitat Politécnica de Catalunya (UPC), Barcelona, Spain, to pursue research on statistical machine translation technologies. He was the author or coauthor of more than 50 publications in international conferences and journals. He has also taught several undergraduate and graduate courses in different universities around the world.

Nicola Bertoldi received the M.S. and Ph.D. degrees in mathematics from the University of Trento, Trento, Italy, in 2000 and 2005, respectively.

He is currently a Research Staff Member at Fondazione Bruno Kessler (FBK, formerly ITC-irst), Trento. He served as a program committee member for the International Workshop on Spoken Language Translation (IWSLT) in 2006-2007. His research interests include multilingual information retrieval, machine translation, and speech translation.

Daniel Déchelotte received the M.Sc. degree in telecommunication from the ENST, Paris, France, in 2002. He is currently pursuing the Ph.D. degree in information technology at the University of Paris-Sud, Orsay, France.

From April 2003 to April 2004, he worked as an intern at the IBM T. J. Watson Research Center, Yorktown Heights, NY. From July 2004 to December 2007, he worked as a Ph.D. student at the LIMSI-CNRS Laboratory, Orsay. His research interests include automatic text and speech translation, natural language processing, speech recognition, and machine learning in general.

Marcello Federico received the degree in computer science from University of Milan, Milan, Italy, in 1987 .

After that he joined Fondazione B. Kessler (FBK, formerly ITC-irst), Trento, Italy, where he has been a Permanent Researcher since 1991. His research interests include statistical machine translation, spoken language translation, statistical language modeling, information retrieval, and speech recognition. He is currently coresponsible of the Human Language Technology research unit at FBK and a Consulting Professor at the University of Trento.

Muntsin Kolss is currently pursuing the Ph.D. degree at the Department of Computer Science, University of Karlsruhe, Karlsruhe, Germany.

Young-Suk Lee received the M.S.E. degree in computer and information science and the Ph.D. degree in linguistics from the University of Pennsylvania, Philadelphia.

She has been working as a Research Staff Member at IBM T. J. Watson Research Center, Yorktown, NY, since November 2001, focusing on open domain statistical machine translation of text and speech. Before joining IBM, she was a Technical Staff Member at the MIT Lincoln Laboratory, Lexington, MA, responsible for various DARPA projects such as TIDES and Communicator. She has served on various professional committees including ACL, COLING, HLT conferences, Internationational Workshop on Spoken Language Translation (IWSLT).

José B. Mariño (M'74) was born in La Coruña, Spain, in 1950. He received the M.Sc. degree in telecommunication engineering from the Universidad Politécnica de Madrid, Madrid, Spain, in 1972 and the Ph.D. degree in signal processing from the Universitat Politécnica de Catalunya (UPC), Barcelona, Spain, in 1975.

He has been a Full Professor at UPC since 1981. He was Director of the Telecommunication Engineering School, Barcelona, from 1983 to 1986 and President of the Department of Signal Theory and Communications from 1986 to 1987, both at UPC. Currently, he is heading a research team in speech-to-speech translation at UPC. He has been managing several speech-related projects in Spain, and he has participated in several European Union research projects (SpeechDat, Orientel, Fame, and TC-Star among others). He has coauthored more than 200 technical papers in the field of signal processing, speech recognition, acoustic modeling, confidence measures, spoken dialogue systems, and speech-to-speech translation.

Matthias Paulik (S'05) received the M.S. degree in computer science from the University of Karlsruhe, Karlsruhe, Germany, in May 2005. He is currently pursuing the Ph.D. degree in the Department of Computer Science, University of Karlsruhe.

He works in the U.S. America part of the InterACT Research Laboratories, Carnegie Mellon University, Pittsburgh, PA. His research focuses on the development of speech translation systems by learning from human interpreters.

Holger Schwenk (M'03) received the M.S. degree from the University of Karlsruhe, Karlsruhe, Germany, in 1992 and the Ph.D. degree from the University Paris 6, Paris, France, in 1996, both in computer science.

He then did postdoctoral studies at the University of Montreal, Montreal, QC, Canada, and at the International Computer Science Institute, Berkeley, CA. From 1998 to 2007, he held an Assistant Professor position at the University of Paris 11, and he was a member of the Spoken Language Processing Group at LIMSI. He is now a Full Professor at the University of Le Mans, Le Mans, France. His research activities focus on new machine learning algorithms with application to human/machine communication, in particular large-vocabulary speech recognition, language modeling, and statistical machine translation. He has participated in several European- and DARPA-funded projects. He was in particular responsible for the activities on statistical machine translation at LIMSI in the framework of the European Tc-Star project on spoken language translation. He has over 50 reviewed publications.

Salim Roukos received the B.E. degree from the American University of Beirut, Beirut, Lebanon, in 1976 and the M.Sc. and Ph.D. degrees from the University of Florida, Gainesville, in 1978 and 1980, respectively.

He is a Senior Manager and CTO for Translation Technologies, IBM Research, Yorktown Heights, NY. His research areas at IBM have been in statistical machine translation, information extraction, statistical parsing, and statistical language understanding for conversational systems. He has served as Chair of the IEEE Digital Signal Processing Committee in 1988. He lead the group that created IBM's ViaVoice Telephony product in 2000, the first commercial software to support full natural language understanding for dialog systems, and more recently in 2003 the first statistical machine translation product for Arabic–English translation.

Hermann Ney (SM'07) received the Dipl. degree in physics from the University of Göttingen, Göttingen, Germany, in 1977 and the Dr.-Ing. degree in electrical engineering from the TU Braunschweig (University of Technology), Braunschweig, Germany, in 1982.

In 1977, he joined Philips Research Laboratories (Hamburg and Aachen, Germany), where he was appointed Head of the Speech and Pattern Recognition Group in 1985. From 1988 to 1989, he was a Visiting Scientist at AT&T Bell Laboratories, Murray Hill, NJ. In July 1993, he joined RWTH Aachen University, Aachen, Germany, as a Professor for computer science. His work is concerned with the application of statistical techniques and dynamic programming for decision-making in context. His current interests cover pattern recognition and the processing of spoken and written language, in particular signal processing, search strategies for speech recognition, language modeling, automatic learning, and translation of spoken and written language.

Dr. Ney, was on the Executive Board of the German section of the IEEE from 1992 to 1998. For the term 1997–2000, he was a member of the Speech Technical Committee of the IEEE.