A NEW DECODER FOR SPOKEN LANGUAGE TRANSLATION BASED ON CONFUSION NETWORKS

Nicola Bertoldi, Marcello Federico

ITC-irst - Centro per la Ricerca Scientifica e Tecnologica
38050 Povo, Trento - ITALY
{bertoldi,federico}@itc.it

ABSTRACT

A novel approach to Spoken Language Translation is proposed, which more tightly integrates Automatic Speech Recognition (ASR) and Statistical Machine Translation (SMT). SMT is directly applied on an approximation of the word graph produced by the ASR system, namely a confusion network. The decoding algorithm extends a conventional phrase-based decoder in that it can process at once a large number of source sentence hypotheses contained in the confusion network. Experimental results are presented on a Spanish-English large vocabulary task, namely the translation of the European Parliament Plenary Sessions. With respect to a conventional SMT decoder processing N-best lists, a slight improvement in the BLEU score is reported as well as a significantly lower decoding time.

1. INTRODUCTION

Research on Spoken Language Translation (SLT) has been strongly boosted in the last years. First attempts to tackle SLT were made by cascading Automatic Speech Recognition (ASR) and Machine Translation (MT). In particular, the best hypothesis produced by the ASR system was passed as a text to the MT system. Hence, supplementary information easily available from the ASR system were not exploited in the translation process, such as the list of N-best hypotheses, the word graph and the likelihoods of the acoustic model (AM) and the language model (LM). Such information is indeed expected to be effective for improving translation quality, if employed properly [1, 2].

More recently, steps have been made toward an effective integration of ASR and MT into a unique statistically sound framework, which takes into account multiple hypotheses generated by the ASR component. Two main directions have been followed: translating N-best lists [2, 3] and translating ASR word-graphs by means of finite-state transducers [4, 5, 6].

In this paper, we propose an alternative approach which lies in between. Translation is namely applied on an approximation of the original ASR word-graph, known as *confusion network* [7]. A specific log-linear translation model and and efficient decoding algorithm are proposed which take advantage of the topological properties of confusion networks. The decoder can be seen as an extension of a phrase-based beam-search algorithm [8], in that each input word can now admit a variable number of alternative hypotheses, including the *empty* word. While re-ordering capabilities of the original algorithm are fully preserved, the exponential growth of the number of hypotheses represented by the confusion network only impacts polynomially on the decoding time.

2. SPOKEN LANGUAGE TRANSLATION

SLT can be considered as an extension of SMT, since its goal is to find the best translation of a speech utterance, rather than of a text string.

Given the vector $\mathbf{o}$ representing the acoustic observations of the input utterance, let us define $\mathcal{F}(\mathbf{o})$ as a set of transcription hypotheses computed by an available ASR system and represented through a word-graph. The assumption is that due to approximations by the acoustic and language models, the word-graph is likely to contain more correct hypotheses than the one with largest probability.

The best translation $\mathbf{e}^*$ is searched among all strings in the target language $\mathcal{E}$ through the following criterion:

$$\mathbf{e}^* = \arg\max_{\mathbf{e}} \sum_{\mathbf{f} \in \mathcal{F}(\mathbf{o})} \Pr(\mathbf{e}, \mathbf{f} \mid \mathbf{o}) \qquad (1)$$

where $\mathbf{f}$ is an hidden variable representing any speech transcription hypothesis. This gives the freedom of generating the best speech translation by considering the contribution of all available transcription hypotheses. Unfortunately, the summation over $\mathcal{F}(\mathbf{o})$ introduces an additional level of complexity, with respect to text translation.

According to the framework of maximum entropy, the conditional distribution $\Pr(\mathbf{e}, \mathbf{f} \mid \mathbf{o})$ can be determined through suitable real valued feature functions $h_r(\mathbf{e}, \mathbf{f}, \mathbf{o})$ and real parameters λ_r, $r = 1 \ldots R$, and takes the parametric form:

$$p_{\boldsymbol{\lambda}}(\mathbf{e}, \mathbf{f} \mid \mathbf{o}) = \frac{1}{\mathcal{Z}(\mathbf{f})} \exp \left\{ \sum_{r=1}^{R} \lambda_r h_r(\mathbf{e}, \mathbf{f}, \mathbf{o}) \right\} \qquad (2)$$

...	recordamos $_{0.98}$	que $_{0.98}$	quienes $_{0.35}$	se $_{0.97}$	presenta $_{0.40}$	$\epsilon_{0.78}$	esas $_{0.86}$	$\epsilon_{0.93}$	elecciones $_{0.97}$	...
			quien $_{0.31}$	he $_{0.03}$	presentó $_{0.22}$	a $_{0.08}$	esa $_{0.03}$	esas $_{0.05}$	selecciones $_{0.03}$	...
			quién $_{0.12}$		presentan $_{0.06}$		$\epsilon_{0.04}$			
			...		...		...			

Fig. 1. Matrix representation of a confusion network generated from a portion a Spanish input utterance. Words and posterior probabilities are shown. The manual transcription of this utterance portion is "`... recordamos que quién se presentó a esas elecciones ...`".

where $\mathcal{Z}(\mathbf{f})$ is a normalization term. By default, we assume $\lambda_r = 1$.

Main advantage of the log-linear model defined in (2) is the possibility to use any kind of features, regarded as important for the sake of translation.

Recently, performance improvements have been achieved by defining features in terms of *phrases* $\tilde{e}$ instead of single words, and to find the best translation $\tilde{\mathbf{e}}^*$ among all strings of phrases in an augmented vocabulary $\tilde{\mathcal{E}}$.

A complete search over all transcription hypotheses $\mathbf{f}$ in $\mathcal{F}(\mathbf{o})$ is often hard to realize because the set $\mathcal{F}(\mathbf{o})$ is usually huge and its structure is complex to decode.

In order to efficiently overcome this issue, in this work two methods are compared. Section 3 describes a novel approach that exploits an approximation of the word-graph, called *confusion network*. Section 4 reviews, as a reference term, the conventional N-best translation approach.

3. CONFUSION NETWORK APPROACH

After formally introducing the concept of confusion network, this section describes a generative translation process starting from a confusion network. Hence, it explains in the order how empty words can be treated, all feature functions used by the translation model, the decoding algorithm, and methods to improve its efficiency.

3.1. Confusion Network

A Confusion Network (CN) $\mathcal{G}$ is a weighted directed graph with a start node, an end node, and word labels over its edges. The CN has the peculiarity that each path from the start node to the end node goes through all the other nodes. As shown in Figure 1, it can be represented as a matrix of words whose columns have different depths. Each word $w_{j,k}$ in $\mathcal{G}$ is identified by its column j and its position k in the column; word $w_{j,k}$ is associated to the weight $p_{j,k}$ corresponding to the posterior probability $\Pr(f = w_{j,k} \mid \mathbf{o}, j)$ of having $f = w_{j,k}$ at position j given $\mathbf{o}$. It is worth noticing that $\Pr(f \mid \mathbf{o}, j)$ defines a probability measure over all words of the j-th column of the CN.

A string $\mathbf{f} = f_1, \ldots, f_m$ is a realization of $\mathcal{G}$ if f_j is equal to any word in the column j, $j = 1, \ldots, m$. Viceversa, any choice of one word per column corresponds to a specific string. In the following, $\mathcal{F}(\mathcal{G})$ will denote the set of all realizations of $\mathcal{G}$.

A realization $\mathbf{f} = f_1, \ldots, f_m$ of $\mathcal{G}$ is associated with the probability $\Pr(\mathbf{f} \mid \mathbf{o})$ of having $\mathbf{f}$ given $\mathbf{o}$, which can be factorized in terms of $\Pr(f \mid \mathbf{o}, j)$ as follows:

$$\Pr(\mathbf{f} \mid \mathbf{o}) = \prod_{j=1}^{m} \Pr(f_j \mid \mathbf{o}, j) \tag{3}$$

Notice that the previous decomposition assumes stochastic independence between the posterior probabilities of the single words.

The generation of the CN from the ASR word-graph [7] can produce in some columns a special word ϵ which corresponds to the empty word. For the sake of simplicity, we assume that ϵ-words are completely indistinguishable from the other normal words, unless differently specified.

3.2. Generative Translation Process

It is assumed that a translation of l phrases $\tilde{\mathbf{e}} = \tilde{e}_1, \ldots, \tilde{e}_l$ is generated incrementally with $l + 1$ steps starting from the input CN.

At each step $i = 0, \ldots, l$: (i) a new phrase $\tilde{e}_i = e_1, \ldots, e_{k_i}$ is added, (ii) some yet uncovered columns of $\mathcal{G}$ are possibly covered, (iii) and one word per column is chosen and mapped to $\tilde{e}_i$.

According to the conventional notation for SMT [9], the tablet τ_i denotes a string of source words translated into $\tilde{e}_i$, and the fertility ϕ_i is the length of τ_i. π_i and ψ_i identify columns and positions within the CN that correspond to the words of τ_i. The target-to-source alignment a_i is a shorthand for $(\phi_i, \tau_i, \pi_i, \psi_i)$. The `null` word/phrase $\tilde{e}_0 = e_0$ copes with those words which cannot be translated.

Figure 2 shows a specific realization of the generative process. Notice that in the here considered phrase-based model, π_i is constrained to cover ϕ_i consecutive columns of the CN. However, different phrase-based models could be derived by modifying this constraint.

The generative process induces an alignment $\mathbf{a} = a_0, \ldots, a_l$ between $\mathcal{G}$ and $\tilde{\mathbf{e}}$, which identifies a specific realization $\mathbf{f}(\mathbf{a}) = f_1, \ldots, f_m$ of $\mathcal{G}$. Any triple $(\tilde{\mathbf{e}}, \mathcal{G}, \mathbf{a})$ corresponds to a solution of length l obtained through the generative process, and $(\tilde{e}_0^i, \mathcal{G}, a_0^i)$ its portion of length i. The set of all compatible alignments between $\mathcal{G}$ and $\tilde{\mathbf{e}}$ is denoted by $\mathcal{A}(\mathcal{G}, \tilde{\mathbf{e}})$. It is trivial to prove that $\mathcal{A}(\mathcal{G}, \tilde{\mathbf{e}}) = \bigcup_{\mathbf{f} \in \mathcal{F}(\mathcal{G})} \mathcal{A}(\mathbf{f}, \tilde{\mathbf{e}})$.

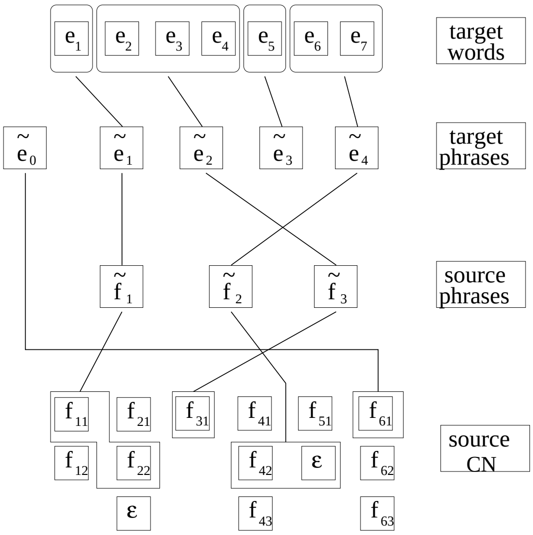

Fig. 2. The generative process of a translation produces an alignment between words of the input confusion network and words of the output string.

Hence, by introducing the alignment $\mathbf{a}$ as a *hidden* variable, the search criterion (1) can be rewritten as follows:

$$\tilde{\mathbf{e}}^* \approx \arg\max_{\tilde{\mathbf{e}}} \max_{\mathbf{a} \in \mathcal{A}(\mathcal{G}, \tilde{\mathbf{e}})} \Pr(\tilde{\mathbf{e}}, \mathbf{a} \mid \mathcal{G}) \tag{4}$$

where the sum over $\mathbf{f}$ is approximated with the maximum.

3.3. Handling ϵ words

An important issue arises from the presence of ϵ-words in $\mathcal{G}$. Whereas they do not affect the generative process, they have to be handled carefully in the definition of some feature functions.

Let us assume that $(\tilde{\mathbf{e}}, \mathcal{G}, \mathbf{a})$ is obtained during the generative process. The realization $\mathbf{f}$ of length m corresponding to this triple can comprise less real words because it possibly might contain some ϵ-words. For this reason, some feature functions should take into account the difference between real and ϵ-words.

In order to give a correct definition of the model, the variables $\hat{\phi}_i$, $\hat{\pi}_i$, $\hat{\psi}_i$, and $\hat{m}$ are introduced which are directly computed from ϕ_i, π_i, ψ_i, and m, respectively, by simply removing the ϵ words.

3.4. Feature functions

The log-linear model $\Pr(\tilde{\mathbf{e}}, \mathbf{a} \mid \mathcal{G})$ is composed by the following 9 feature functions.

- One feature is the logarithm of a 3-gram target LM:

$$\begin{aligned} h_1(\tilde{\mathbf{e}}, \mathcal{G}, \mathbf{a}) &= \log \Pr(\tilde{\mathbf{e}} = \tilde{e}_1, \ldots, \tilde{e}_l) \\ &= \sum_{i=1}^{l} \log p(k_i) p(\tilde{e}_i \mid \tilde{e}_{i-2}, \tilde{e}_{i-1}) \end{aligned} \tag{5}$$

where k_i is the length of $\tilde{e}_i$. The phrase-based 3-gram probabilities are further factorized by exploiting a conventional word-based 3-gram LM.

- Two features model the fertility of the target phrases and the `null` word. The former relies on statistics extracted from a sample of phrase pairs, while the latter is the logarithm of a binomial distribution.

$$h_2(\tilde{\mathbf{e}}, \mathcal{G}, \mathbf{a}) = \sum_{i=1}^{l} \log \frac{N(\hat{\phi}_i, \tilde{e}_i)}{N(\tilde{e}_i)} \tag{6}$$

$$h_3(\tilde{\mathbf{e}}, \mathcal{G}, \mathbf{a}) = \log b(\hat{\phi}_0 \mid \hat{m} - \hat{\phi}_0) \tag{7}$$

- Two feature functions cope with the reordering of source phrases: they are defined in terms of the distance d_i between the position $\pi_{i,1}$ of the first word of the source phrase $\tilde{f}(a_i)$ and the center $\bar{\pi}_{i-1}$ of the previous one. In the case of confusion networks, the true distance after the removal of ϵ-words should be taken into account. As in general this distance can be determined only when all intermediate positions have been covered, an expected value $\bar{d}_i$ is computed if this is not the case.

Monotone and non-monotone position coverage are, respectively, modelled by:

$$h_4(\tilde{\mathbf{e}}, \mathcal{G}, \mathbf{a}) = \sum_{i=1}^{l} t(\pi_i, \phi_i, \bar{\pi}_{i-1})\, \delta(\pi_i >= \bar{\pi}_i) \tag{8}$$

$$h_5(\tilde{\mathbf{e}}, \mathcal{G}, \mathbf{a}) = \sum_{i=1}^{l} t(\pi_i, \phi_i, \bar{\pi}_{i-1})\, \delta(\pi_i < \bar{\pi}_i) \tag{9}$$

where

$$t(\pi_i, \phi_i, \bar{\pi}_{i-1}) = -|\bar{d}_i| \, \log \prod_{k=2}^{\phi_i} \delta(\pi_{i,k} - \pi_{i,k-1} = 1)$$

Notice that the above definition inhibits the coverage of not contiguous columns of the CN.

- A feature assigns uniform probabilities to remaining positions covered by the `null` word

$$h_6(\tilde{\mathbf{e}}, \mathcal{G}, \mathbf{a}) = \log \frac{1}{\phi_0!} \tag{10}$$

- A feature models the phrase-based lexicon:

$$h_7(\tilde{\mathbf{e}}, \mathcal{G}, \mathbf{a}) = \sum_{i=0}^{l} \log \frac{N(\tilde{f}(a_i), \hat{\phi}_i, \tilde{e}_i)}{N(\hat{\phi}_i, \tilde{e}_i)} \tag{11}$$

where $\tilde{f}(a_i)$ is simply the phrase obtained by concatenating real words aligned with $\tilde{e}_i$. Phrase-pair statistics can be automatically extracted from a word-aligned parallel corpus in various ways (e.g. see [10]).

• A feature accounts for the length of a realization of the CN through the following function:

$$h_8(\tilde{\mathbf{e}}, \mathcal{G}, \mathbf{a}) \quad = \quad \sum_{i=0}^{l} \mid \tilde{f}(a_i) \mid \qquad (12)$$

• The last feature function measures how probable a realization $\mathbf{f}$ is within the CN $\mathcal{G}$. By rearranging the terms in (3) and taking the logarithm:

$$h_9(\tilde{\mathbf{e}}, \mathcal{G}, \mathbf{a}) \quad = \quad \sum_{i=0}^{l} \log \prod_{h=1}^{\phi_i} p_{\pi_{i,h}, \psi_{i,h}} \qquad (13)$$

The above feature functions permit to express the cost of generating a partial solution of length i in terms of the cost S of the corresponding steps of the generative process:

$$\sum_{r=1}^{R} h_r(\tilde{e}_0^i, \mathcal{G}, a_0^i) = \sum_{t=0}^{i} S(\mathcal{C}, \pi_t, \bar{\pi}_{t-1}, \psi_t, \tilde{e}_t, \tilde{e}_{t-1}, \tilde{e}_{t-2}) (14)$$

Notice that the cost of step 0 only depends on π_0 and ψ_0.

3.5. Decoding Algorithm

Through the log-linear model, the approximate search criterion (4) can be rewritten as:

$$\tilde{\mathbf{e}}^* \quad \approx \quad \arg\max_{\tilde{\mathbf{e}}} \max_{\mathbf{a} \in \mathcal{A}(\mathcal{G}, \tilde{\mathbf{e}})} \sum_{r=1}^{R} \lambda_r h_r(\tilde{\mathbf{e}}, \mathcal{G}, \mathbf{a}) \qquad (15)$$

According to the *dynamic programming* paradigm, the optimal solution can be computed through a recursive formula which expands and recombines previously computed partial theories. A theory can be described by its *state*, which only includes the information needed for its expansion; two partial theories sharing the same state are considered identical (indistinguishable) for the sake of expansion and are recombined.

More formally, let $Q_i(s)$ be the best score among all partial theories of length i sharing the state s, $pred(s)$ the set of partial theories which can be expanded into a theory of state s, and $G(s', s)$ be the cost of such expansion.

The score $Q_i(s)$ can be defined recursively with respect to the length i as follows:

$$Q_i(s) = \max_{th' \in pred(s)} Q_{i-1}(s(th')) + G(s(th'), s) \quad (16)$$

with a suitable initialization for $Q_0(s)$.

Given the log-linear model described in the previous section, the state $s(th)$ of a partial theory th includes the coverage set $\mathcal{C}$, the center of the last cept $\bar{\pi}$, and the last two output phrases $\tilde{e}'$ and $\tilde{e}$. A theory of state $s = (\mathcal{C}, \bar{\pi}, \tilde{e}', \tilde{e})$ can be only generated from one of state $s' = (\mathcal{C} \setminus \pi, \bar{\pi}', \tilde{e}'', \tilde{e}')$. In other words, a new output phrase $\tilde{e}$ is added with fertility $\phi = |\pi|$, columns π_i are covered and words ψ_i are selected. Notice that if $\phi = 0$ the center remains unaltered, i.e. $\bar{\pi}' = \bar{\pi}$. The possible initial states $s = (\pi_0, \bar{\pi}_0, \epsilon, \epsilon)$ correspond to partial theories with no target phrases and with all ϕ_0 words identified by π_0 and ψ_0 covered by the null word $\tilde{e}_0$. Notice that $\bar{\pi}_0$ is not used in the computation. Hence, eq. 16 relies on the following definitions:

$$G(s', s) \; = \max_{\psi_i} S(\mathcal{C}, \pi_i, \bar{\pi}_{i1-}, \psi_i, \tilde{e}_i, \tilde{e}_{i-1}, \tilde{e}_{i-2}) \quad (17)$$

$$Q_0(s) \; = \; \max_{\psi_0} S(\pi_0, \psi_0) \qquad (18)$$

The score Q^* of the optimal solution $\tilde{\mathbf{e}}^*$ can be searched among theories of any length i which are in a final state s, i.e. covering all columns of $\mathcal{G}$:

$$Q^* \quad = \quad \max_{\tilde{\mathbf{e}}} \max_{\mathbf{a}} \sum_{r=1}^{R} \lambda_r h_r(\tilde{\mathbf{e}}, \mathcal{G}, \mathbf{a}) \qquad (19)$$

$$= \quad \max_{i,s \text{ is final}} Q_i(s) \qquad (20)$$

Notice that a decoder for text strings is essentially identical, as any string can be seen as a CN with one word per column. The complexity of the algorithm is

$$O\left(2^m \; m^3 \; \phi_{max} \; \psi_{max}^{\phi_{max}} \begin{pmatrix} m \\ \phi_{max} \end{pmatrix} |\tilde{\mathcal{E}}|^3\right)$$

where ψ_{max} is the largest depth of the CN, and ϕ_{max} is the maximum fertility of the target phrases. Although the number of possible strings within the CN is super-polynomial with respect to its depth, the impact on the algorithm's complexity is only polynomial.

3.6. Complexity reduction

In order to reduce the huge number of theories to generate, four approximations are introduced in the algorithm:

- *Beam search*: at each expansion less promising theories are removed by applying *threshold* and *histogram* pruning criteria to all the theories covering the same set of source positions, and to all the theories with the same output length.

- *Reordering constraints*: columns to be covered are selected by applying the so-called IBM constraint; moreover, the maximum distortion is also limited to some value V. In this work a monotone search ($V = 1$) is performed. Notice that phrase-based translation permits anyway intra-phrase re-ordering.

- *Lexicon cutoff*: for each source phrase, only the most probable phrases are taken as translation alternatives, i.e. up to .95 probability and no more than 30.

- *Confusion network cutoff*: less input words are considered in the source CN by deleting terms $w_{j,k}$ which fall outside a given percentile.

4. N-BEST APPROACH

An alternative way to define the set $\mathcal{F}(\mathbf{o})$ is to take the N most probable hypotheses computed by the ASR system, i.e. $\mathcal{F}(\mathbf{o}) = \{\mathbf{f}_1, \ldots, \mathbf{f}_N\}$. By taking a maximum approximation over $\mathcal{F}(\mathbf{o})$, and assuming that $\Pr(\tilde{\mathbf{e}}, \mathbf{f} \mid \mathbf{o}) = \Pr(\mathbf{f} \mid \mathbf{o}) \Pr(\tilde{\mathbf{e}} \mid \mathbf{f})$ [1], we get the search criterion:

$$\tilde{\mathbf{e}}^* \quad \approx \quad \arg \max_{n=1,..,N} \Pr(\mathbf{f}_n \mid \mathbf{o}) \max_{\tilde{\mathbf{e}}} \Pr(\tilde{\mathbf{e}} \mid \mathbf{f}_n) \tag{21}$$

In the equation above one can isolate N problems of text translation (rightmost maximization), and the recombination of N results (leftmost maximization). Hence, the search criterion can be restated as:

$$\tilde{\mathbf{e}}^*_n \quad = \quad \arg \max_{\tilde{\mathbf{e}}} \Pr(\tilde{\mathbf{e}} \mid \mathbf{f}_n) \qquad n = 1, \ldots, N \tag{22}$$

$$\tilde{\mathbf{e}}^* \quad \approx \quad \arg \max_{n=1,..,N} \Pr(\mathbf{f}_n \mid \mathbf{o}) \Pr(\tilde{\mathbf{e}}^*_n \mid \mathbf{f}_n) \tag{23}$$

In plain words: first the best translation $\tilde{\mathbf{e}}^*_n$ of each transcription hypothesis $\mathbf{f}_n$ is searched; then, the best translation $\tilde{\mathbf{e}}^*$ is selected among $\{\tilde{\mathbf{e}}^*_1, \ldots, \tilde{\mathbf{e}}^*_N\}$ according to its score weighted by the ASR posterior probability $\Pr(\mathbf{f}_n \mid \mathbf{o})$.

A phrase-based log-linear model for text translation is employed which is very similar to the CN decoder. In particular, features h_8 and h_9 are respectively replaced with the AM and LM probabilities provided by the ASR system. Notice that the default value for the AM weight was empirically set to 0.05 to take into account the high dynamics of AM probabilities.

5. EXPERIMENTAL EVALUATION

The two presented SLT approaches have been experimentally compared on a large vocabulary translation task. The task consists in translating the European Parliament Plenary Sessions (EPPS) from Spanish into English. Statistics for the training, development and test data are given in Table 1. Two references per sentence were used both for the development and the test set. Performance of the two search methods are measured in terms of BLEU score [11] and decoding time.

Both approaches share the same modules for pre- and post-processing, extraction of phrase pairs, generation of N-best translations, and estimation of the log-linear parameter through minimum error training. Descriptions of these modules can be found in [3, 8, 10, 12].

Confusion networks and 1000-best ASR transcriptions were kindly provided by CNRS-LIMSI, France. From these, N-best lists ($N = 1, 5, 10, 20, 50, 100$) and CNs pruned with different percentiles (p =0, 50, 55, 60, 65, 70) were extracted. Notice that the CN `cn-p00` corresponds to the consensus decoding transcription [7]. Finally, for the sake of completeness, performance on the correct human transcriptions (`verbatim`) are also reported.

Table 1. Statistics of training, development and test data of the Spanish-English EPPS Task.

		Spanish	English
Train	Sentences	1 207 740	
	Words	31 360 260	30 049 355
	Vocabulary	139 587	93 995
Dev	Sentences	2643	
	Words	20 289	23 407
	Vocabulary	2932	2566
Test	Sentences	1073	
	Words	18 896	19 306
	Vocabulary	3302	2772

Weight optimization of the log-linear models was performed on the development set by applying a minimum error training procedure [12]. In particular, 100 alternative translations for each of the N-best transcriptions and 1000 for the confusion network were generated by the respective decoders. Notice that a separate optimization was performed for each N-best and CN condition.

Table 2 reports the average number of alternative ASR transcriptions processed in input (input aver. size) and the minimum word error rate (input WER) that was found in them. Figures are provided for the development and test sets. Notice that for the N-best case, the average input size is lower than N because for many short sentence the ASR search space was relatively small. Finally, BLEU score and average MT decoding time are reported.

Statistics of Table 1 show that sentences of the test set are longer than those of the development set, and this is reflected by the larger decoding time of both systems, and the larger size of the CNs. It is also worth noticing that the test set is significantly more difficult than the development set in terms of `WER` performance and, consequently, in terms of BLEU score.

Concerning translation performance, we can notice that BLEU score decreases on the test set by about 10% relative, when moving from `verbatim` transcription to ASR output.

The CN-based decoder performs slightly better than the one based on N-bests, but the difference is not significant. One advantage of the CN-based decoder is however its efficiency; in fact, with comparable decoding time, it translates a significantly larger amount of hypotheses with respect to the N-best decoder. Unfortunately, for some reason, the quality of these hypotheses results poorer than that of the N-best lists, as shown in `WER` column.

With respect to a conventional SLT translation system, only exploiting one ASR hypothesis, the best translation performance were achieved by the CN decoder. On the development set, the configuration `cn-p60` improved the BLEU

[1] This means that $\tilde{\mathbf{e}}$ is stochastically conditional independent from the acoustic observations $\mathbf{o}$, given $\mathbf{f}$.

Table 2. For each kind of input, statistics and performance about the development and the test sets are reported: average number of hypotheses, ASR `WER`, `BLEU` score, and MT decoding time.

	DEV				TEST			
	input aver. size	input `WER`	`BLEU`	decoding time	input aver. size	input `WER`	`BLEU`	decoding time
`verbatim`	1	0	45.78	0.6	1	0	40.84	1.7
`1-best`	1	11.77	40.17	0.6	1	14.60	36.64	2.1
`5-best`	4	8.12	40.63	2.8	5	11.90	36.47	10.5
`10-best`	8	6.99	40.83	5.3	9	11.02	36.75	20.4
`20-best`	13	6.19	41.03	9.8	16	10.20	36.55	38.9
`50-best`	25	5.40	40.85	20.6	34	9.47	36.66	84.2
`100-best`	38	5.07	40.87	33.2	56	9.09	36.68	135.3
`cn-p00`	1	11.67	40.30	4.0	1	14.46	36.54	28.4
`cn-p50`	4	9.42	41.06	5.8	32	11.86	37.14	31.2
`cn-p55`	13	8.93	41.21	6.3	150	11.32	37.23	34.7
`cn-p60`	194	8.41	41.24	6.7	1284	10.71	37.21	37.9
`cn-p65`	1,359	7.91	41.21	7.4	9816	35.07	37.05	43.9
`cn-p70`	15,056	7.53	41.23	27.4	228461	9.71	37.14	54.6

score from 40.17% to 41.24%, which on the test set corresponds to a BLEU score increment from 36.64% to 37.21%.

6. CONCLUSIONS

In this paper we presented a novel approach to tightly integrate ASR and SMT. The presented experiments focused on the search algorithm only, hence no re-scoring module was applied, which could be beneficial to improve performance. In particular, among the potentially useful features that could be applied after the CN decoder there is the source language LM, which could account for the linguistic plausibility of each realization within the CN. Moreover, the use of additional lexicon models and a 4-gram target language model for rescoring and/or decoding had been proved to be effective in very recent experiments [10]. Finally, the relatively high WER of the CN will be investigated and possible alternative ways to generate more accurate CNs will be considered.

7. ACKNOWLEDGMENTS

We wish to thank Hölger Schwenk and Jean-Luc Gauvain, CNRS-LIMSI France, for their kind support. This work was partially financed by the European Commission under the project TC-STAR - Technology and Corpora for Speech to Speech Translation Research (IST-2002-2.3.1.6, http://www.tc-star.org).

8. REFERENCES

[1] H. Ney, “Speech Translation: Coupling of Recognition and Translation”. Proc. *ICASSP*, Phoenix, AR, 1999.

[2] R. Zhang et al., “A Unified Approach in Speech-to-Speech Translation: Integrating Features of Speech Recognition and Machine Translation”. Proc. *COLING*, Geneve, Switzerland, 2004.

[3] V.H. Quan et al., “Integrated n-best re-ranking for spoken language translation”. Proc. *Interspeech*, Lisbon, Portugal, 2005.

[4] S. Bangalore and G. Riccardi, “Stochastic finite-state models for spoken language machine translation”. *Machine Translation*, 17(3), 2002.

[5] F. Casacuberta, et al., “Some approaches to statistical and finite-state speech-to-speech translation”. *Computer Speech and Language*, 18, 2004.

[6] S. Saleem et al., “Using word lattice information for a tighter coupling in speech translation system”. Proc. *ICSLP*, Jeju Island, Korea, 2004.

[7] L. Mangu et al., “Finding consensus among words: Lattice-based word error minimization”. Proc. *ISCA ECSCT*, Budapest, Hungary, 1999.

[8] N. Bertoldi et al., “The ITC-irst Statistical Machine Translation System for IWSLT-2004”. Proc. *IWSLT*, Kyoto, Japan, 2004.

[9] Peter F. Brown, et al., “The mathematics of statistical machine translation: Parameter estimation,” *Computational Linguistics*, 19(2), 1993.

[10] M. Cettolo et al., “A look inside the ITC-irst SMT system”. Proc. *MT Summit X*, Phuket, Thailand, 2005,

[11] K. Papineni et al., “Bleu: a method for automatic evaluation of machine translation.” RC22176, Thomas J. Watson Research Center, 2001.

[12] M. Cettolo and M. Federico, “Minimum Error Training of Log-Linear Translation Models”. Proc. *IWSLT*, Kyoto, Japan, 2004.