SPEECH TRANSLATION BY CONFUSION NETWORK DECODING

Nicola Bertoldi

ITC-irst
Centro per la Ricerca
Scientifica e Tecnologica
I-38050 Povo (Trento), Italy
bertoldi@itc.it

Richard Zens

Lehrstuhl für Informatik 6
Computer Science Department
RWTH Aachen University
D-52056 Aachen, Germany
zens@cs.rwth-aachen.de

Marcello Federico

ITC-irst
Centro per la Ricerca
Scientifica e Tecnologica
I-38050 Povo (Trento), Italy
federico@itc.it

ABSTRACT

This paper describes advances in the use of confusion networks as interface between automatic speech recognition and machine translation. In particular, it presents an implementation of a confusion network decoder which significantly improves both in efficiency and performance previous work along this direction. The confusion network decoder results as an extension of a state-of-the-art phrase-based text translation system. Experimental results in terms of decoding speed and translation accuracy are reported on a real-data task, namely the translation of Plenary Speeches at the European Parliament from Spanish to English.

Index Terms— Machine Translation, Speech Translation, Natural Language Processing

1. INTRODUCTION

Machine translation input currently takes the form of simple sequences of words. However, there are increasing demands to integrate machine translation technology in larger information processing systems with upstream NLP/speech processing tools (such as named entity recognizers, speech recognizers, morphological analyzers, etc.). These upstream processes tend to generate multiple, erroneous hypotheses with varying confidence. Current MT systems are designed to process only one input hypothesis, making them vulnerable to errors in the input.

This work focuses on the speech translation case, where the input is generated by a speech recognizer. Recently, approaches have been proposed for improving translation quality through the processing of multiple input hypotheses. In particular, better translation performance have been reported by exploiting N-best lists [1, 2], word lattices [3, 4], and confusion networks [5].

This work improves the confusion network decoder discussed in [5], by developing a simpler translation model and a more efficient implementation of the search algorithm.

Finally, the here described decoder was implemented during the JHU Summer Workshop 2006 as an extension of `Moses`[1], a factored phrase-based beam-search decoder for machine translation.

2. SPOKEN LANGUAGE TRANSLATION

From a statistical perspective, SLT can be approached as follows. Given the vector $\mathbf{o}$ representing the acoustic observations of the input utterance, let $\mathcal{F}(\mathbf{o})$ be a set of transcription hypotheses computed by a speech recognizers and represented as a word-graph. 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 the source language sentence $\mathbf{f}$ is an hidden variable representing any speech transcription hypothesis. According to the well established log-linear framework, 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-valued parameters λ_r, $r = 1 \ldots R$, and takes the parametric form:

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

where $\mathcal{Z}(\mathbf{o})$ is a normalization term.

The 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. Currently, better performance are achieved by defining features in terms of *phrases* $\tilde{e}$ [6, 7, 8] instead of single words, and by searching the best translation $\tilde{\mathbf{e}}^*$ among all strings of phrases in a defined vocabulary of phrases.

The kind of representation used for the set of hypotheses $\mathcal{F}(\mathbf{o})$ clearly impacts on the implementation of the search algorithm. Here, we assume to have all hypotheses represented as a confusion network.

[1] Open source project web site http://www.statmt.org/moses.

se$_{.97}$	presenta$_{.40}$	$\epsilon_{.78}$	esas$_{.86}$	elecciones$_{.97}$
he$_{.03}$	presentó$_{.22}$	a$_{.08}$	$\epsilon_{.10}$	selecciones$_{.03}$
	presentan$_{.06}$	e$_{.07}$	esa$_{.04}$	
	...	en$_{.06}$		
		...		

Fig. 1. Example of confusion network.

3. CONFUSION NETWORKS

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, a CN 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 $\mathbf{o}$. A realization $\mathbf{f} = f_1, \ldots, f_m$ of $\mathcal{G}$ is associated with the probability $\Pr(\mathbf{f} \mid \mathbf{o})$, which is factorized as follows:

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

The generation of a CN from an ASR word-graph [9] can also produce special empty-words ϵ in some columns. These empty-words permit to generate source sentences of different length and are treated differently from regular words only at the level of feature functions.

3.1. Generative translation process

The following process describes how to incrementally generate a translation from $\mathcal{G}$.
While there are uncovered source columns:

i. A *span* of some yet uncovered and contiguous columns of $\mathcal{G}$ is is chosen and marked as covered.

ii. One word per column is chosen. This identifies a specific source phrase $\tilde{f}$ of the current span.

iii. A target phrase $\tilde{e}$ is chosen among the translation alternatives of $\tilde{f}$ and appended to the current translation.

The here presented statistical model could work on lattices, too; but unfortunately, lattices have a significantly more complex topology than CNs, and an efficient decoding algorithm for them has not been yet proposed. Main issues to be solved are related to word reordering and path overlaps:

- as words can be translated in any order, an asynchronous visit of the graph is required
- any path in the WG has to be visited even if there are many other similar paths, that is corresponding to similar transcriptions.

3.2. CN-based log-linear model

The log-linear model adopted for the CN decoder includes the following feature functions:

i. A word-based n-gram target LM.

ii. A reordering model defined in terms of the distance between the first column covered by current span and the last column of the previous span. (In the current implementation, we did not distinguish between regular and empty words.)

iii. Four phrase-based lexicon models compute the probability of $\tilde{f}$ given $\tilde{e}$ and viceversa in two ways: by relative frequency and through IBM Model 1. These models remove any empty-word in the source side.

iv. Phrase and word penalty models, i.e. counts of the number of phrases and words in the target string.

v. The CN posterior probability, see formula (3).

Notice that the above features can grouped into two categories: those which are expansion-dependent because their computation requires some knowledge about the previous step (i, ii), and those which are not (iii, iv, v).

3.3. Decoding algorithm

According to the *dynamic programming* paradigm, the optimal solution can be computed through expansions and recombinations of previously computed partial theories. With respect to translating a single input hypothesis, translating from a CN requires, in principle, exploring all possible input paths inside the graph. A key insight is that, due to their linear structure, CN decoding is very similar to text decoding. During the decoding, we have to look up the translation options of spans, i.e. some contiguous sequence of source positions. The main difference between CN and text decoding is that in text decoding there is exactly one source phrase per span, whereas in confusion network decoding there can be multiple source phrases per span. In fact, in a CN the number of source phrases per span is exponential in the span length, assuming its minimum depth is larger than one.
The decoding algorithm can be made much more efficient by pre-fetching translations for all the spans and by applying early recombination.

3.4. Early recombination

At each expansion step a span covering a given number of consecutive columns is generated. Due to the presence of empty-words, different paths within the span can generate the same source phrase, hence the same translations. The scores of such paths only impacts on the CN posterior feature (v). Additionally, it might happen that two different source phrases of the same span have a common translation. In this case, not only the CN posterior feature is different, but also

the phrase translation features (iii). This suggests that efficiency can be gained by pre-computing all possible alternative translations for all possible spans, together with their expansion-independent scores, and to recombine these translations in advance.

3.5. Pre-fetching of translation options

Concerning the pre-fetching of translations from the phrase-table, an efficient implementation can be achieved if we use a prefix tree representation for the source phrases in the phrase table and generate the translation options incrementally over the span length. So, when looking up a span (j_1, j_2), we can exploit our knowledge about the span $(j_1, j_2 - 1)$. Thus, we have to check only for the known prefixes of $(j_1, j_2 - 1)$ if there exists a successor prefix with a word in column j_2 of the CN. If all the word sequences in the CN also occur in the phrase table, this approach still enumerates an exponential number of phrases. So, the worst case complexity is still exponential in the span length. Nevertheless, this is unlikely to happen in practice. In our experiments, we do not observe the exponential behavior. What we observe is a constant overhead compared to text input.

4. N-BEST DECODER

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 $\Pr(\tilde{\mathbf{e}}, \mathbf{f} \mid \mathbf{o}) = \Pr(\mathbf{f} \mid \mathbf{o}) \Pr(\tilde{\mathbf{e}} \mid \mathbf{f})$, we get the search criterion:

$$\tilde{\mathbf{e}}^* \approx \arg \max_{n=1,..,N} \Pr(\mathbf{f}_n \mid \mathbf{o}) \max_{\tilde{\mathbf{e}}} \Pr(\tilde{\mathbf{e}} \mid \mathbf{f}_n) \quad (4)$$

In the equation above we can isolate N independent translation tasks (rightmost maximization), and the recombination of their results (leftmost maximization). Hence, the search criterion can be restated as:

$$\tilde{\mathbf{e}}_n^* = \arg \max_{\tilde{\mathbf{e}}} \Pr(\tilde{\mathbf{e}} \mid \mathbf{f}_n) \qquad n = 1, \ldots, N \quad (5)$$

$$\tilde{\mathbf{e}}^* \approx \arg \max_{n=1,..,N} \Pr(\mathbf{f}_n \mid \mathbf{o}) \Pr(\tilde{\mathbf{e}}_n^* \mid \mathbf{f}_n) \quad (6)$$

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 log-linear model for the N-best decoder is employed which is very similar to the CN decoder. Specifically, feature (v) is replaced with two features corresponding to the log-probability of the acoustic and language model scores provided by the ASR system.

5. EXPERIMENTAL RESULTS

Experiments were carried on one of the TC-STAR project tasks, namely the translation from Spanish to English of speeches from the European Parliament Plenary Sessions (EPPS). Statistics about the training, development and testing data are reported in Table 5. In particular, training of the lexicon models (phrase table) was performed with the `Moses` training tools, while training of the 4-gram target LM was performed with the IRST LM Toolkit. Sentences in the development and test sets are provided with two reference translations each.

		Spanish	English
Train	Words	37 M	36 M
	Vocabulary	143 K	110 K
	Phrase Pairs	83 M	
	Phrases	48 M	44 M
Dev	Utterances	2,643	
	Words	20,384	20,579
	Vocabulary	2,883	2,362
Test	Utterances	1,073	
	Words	18,890	18,758
	Vocabulary	3,139	2,567

Table 1. Statistics of the EPPS speech translation task. Word counts of dev and test sets sets refer to human transcriptions (Spanish) and the first reference translation (English).

5.1. Data preparation

Word lattices were kindly provided by CNRS-LIMSI, France. CNs and N-best lists were extracted by means of the `lattice-tool` package included in the SRILM Toolkit [10]. The resulting CNs have an average depth of 2.8 words. The consensus decoding [9] transcriptions were also extracted from the CN, by taking the most probable words of each column. Table 2 shows on its left side the average Word Error Rate (WER) of the oracle transcriptions of the CNs and the word lattices, of the consensus decoding transcriptions, and of the oracle transcriptions of various N-best lists.

5.2. Parameter tuning

Feature weights of all presented models were estimated by applying a minimum-error-rate training procedure which tries to maximize the BLEU score over the dev data. A special procedure was used for tuning the weights of the N-best translation system. First, a single best decoder was optimized over the dev set. Then M-best (M=100) translations were generated for each N-best input of the dev set. Hence, all NxM translations were merged and a new log-linear model including the ASR additional features was trained.

5.3. Results

Table 2 reports BLEU score, position-independent error rate (PER) and WER achieved by the decoder under different

Input		Output		
type	WER	BLEU	PER	WER
verbatim	0.0	48.00	31.19	40.96
wg-oracle	7.48	44.68	33.55	43.74
cn-oracle	8.45	44.12	34.37	44.95
cn	8.45	39.17	38.64	49.52
cons-dec	23.30	36.98	39.17	49.98
1-best	22.41	37.57	39.24	50.01
5-best	18.61	38.68	38.55	49.33
10-best	17.12	38.61	38.69	49.46

Table 2. Performance achieved with different inputs.

Input		Output		
type	WER	BLEU		
		[5]	[11]	Moses
verbatim	0.0	40.84	44.64	48.00
1-best	14.61	36.64	39.67	42.84
cons-dec	14.46	36.54	39.65	42.92
cn	11.61	37.21	40.00	43.51

Table 3. Comparison between Moses and previous implementations described in [5] and [11].

input conditions. Scores achieved on the textual inputs – i.e. verbatim, wg-oracle, cn-oracle, 1-best, and cons-dec– shows a strong correlation between WER and MT automatic scores. CN translation (cn) outperforms 1-best and consensus-decoding translations with respect to all translation metrics. CN decoding also performs better, in terms of BLEU score, than N-best decoding, which is significant given that all systems were trained to optimize the BLEU score. From the point of view of decoding speed, the advantage of CN decoding becomes even more important. With respect to 1-best decoding, CN decoding time is just 2.1 times higher (87.5 vs 42.5 seconds per sentence), i.e. it is comparable to 2-best decoding.

In Table 3, performance of Moses are compared against a previous implementation of a CN-decoder [5] and against a more recently developed decoder [11] which ranked top in the TC-STAR 2006 Evaluation Campaign. [5] uses only one phrase-based lexicon model, and a weaker recombination criterion than Moses. These additional experiments were conducted on the same task but exploited word lattices with smaller WERs and pruned CNs. It is evident that Moses outperforms all previous implementations of CN-decoder, which are also significantly slower (18 time factor with respect to 1-best decoding).

6. CONCLUSIONS

This work presented a new implementation of a phrase-based decoder for speech translation. The decoder exploits confusion networks as interface between speech recognition and machine translation. Confusion networks from one side permit to effectively represent a huge number of transcription hypotheses, from the other side they lead to a very efficient search algorithm for statistical machine translation. Comparisons against previous implementations showed significant gains in translation performance and decoding speed. The new implementation is part of an open source decoder, named Moses.

7. ACKNOWLEDGEMENTS

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), and by the JHU Summer Workshop 2006. We wish to thank all our workshop teammates Ondrej Bojar, Chris Callison-Burch, Alexandra Constantine, Christine Corbett Moran, Brooke Cowan, Chris Dyer, Evan Herbst, Hieu Hoang, Philipp Koehn, and Wade Shen.

8. REFERENCES

[1] R. Zhang, et al., "A unified approach in speech-to-speech tanslation: integrating features of speech recognition and machine translation," in *Proc. of COLING*, Geneve, Switzerland, 2004.

[2] V. H. Quan, et al., "Integrated n-best re-ranking for spoken language translation," in *Proc. of Interspeech*, Lisbon, Portugal, 2005.

[3] E. Matusov, et al., "On the integration of speech recognition and statistical machine translation," in *Proc. of Interspeech*, Lisbon, Portugal, 2005.

[4] L. Mathias and W. Byrne, "Statistical phrase-based speech translation," in *Proc. of ICASSP*, Toulouse, France, 2006.

[5] N. Bertoldi and M. Federico, "A new decoder for spoken language translation based on confusion networks," in *Proc. of IEEE ASRU*, San Juan, Puertorico, 2005.

[6] R. Zens, et al., "Phrase-based statistical machine translation," in *KI-2002, 25th Annual German Conference on AI*. 2002, vol. 2479 of *Lecture Notes in Artificial Intelligence*, pp. 18–32, Springer Verlag.

[7] P. Koehn, et al., "Statistical phrase-based translation," in *Proc. of HLT/NAACL 2003*, Edmonton, Canada, 2003.

[8] M. Federico and N. Bertoldi, "A word-to-phrase statistical translation model," *ACM Trans. on Speech and Language Processing (TSLP)*, vol. 2, no. 2, pp. 1–24, 2005.

[9] L. Mangu, et al., "Finding consensus in speech recognition: Word error minimization and other applications of confusion networks," *Computer, Speech and Language*, vol. 14, no. 4, pp. 373–400, 2000.

[10] A. Stolcke, "SRILM - an extensible language modeling toolkit," in *Proc. of ICSLP*, Denver, Colorado, 2002.

[11] N. Bertoldi, et al., "ITC-irst at the 2006 TC-STAR SLT Evaluation Campaign," in *Proc. of the TC-STAR Workshop on Speech-to-Speech Translation*, Barcelona, Spain, 2006.