



# Punctuating Confusion Networks for Speech Translation

Roldano Cattoni, Nicola Bertoldi, Marcello Federico

Fondazione Bruno Kessler - IRST  
I-38050 Povo (Trento), Italy

## Abstract

Translating from confusion networks (CNs) has been proven to be more effective than translating from single best hypotheses. Moreover, it is widely accepted that the availability of good punctuation marks in the input can improve translation quality. At present, no ASR systems can generate punctuation marks in the word graphs, therefore CNs miss punctuation. In this paper we investigate the problem of adding punctuation marks into confusion networks. We investigate different punctuation strategies and show that the use of multiple hypotheses improves translation quality in a large-vocabulary speech translation task.

**Index Terms:** speech translation, statistical machine translation, punctuation restoration

## 1. Introduction

The recent years have seen a dramatic progress in the field of automatic speech translation through the application of the statistical approach. A major challenge of research is nowadays the translation of long audio streams containing speech in a given language into a text in another language, that can be possibly passed to a speech synthesizer. As general and widely accepted requirements, the output text should contain capitalization and punctuation marks. This requirement sounds natural in the case of text translation, given that for many Western languages pairs capitalization and punctuation can be quite easily transferred from the source to the target language [1]. Translating speech rather than text is generally more difficult, for several reasons: spoken language is less constrained, errors in the source are introduced by automatic speech recognition, punctuation and capitalization are not explicitly represented in the source, and, finally, input to be translated cannot be organized into syntactically consistent segments but is rather a continuous stream of words.

This paper addresses the problem of integrating punctuation information into a large-vocabulary speech translation task, namely the translation of political speeches from English to Spanish held at the European Parliament. With respect to conversational speech, this task offers the advantages of dealing with well planned speech and of having plenty of training data at disposal under form of so called *final text editions*, namely polished versions of both source and target languages.

The paper is organized as follows. Section 2 introduces our speech translation baseline based on confusion network decoding. Section 3 discusses approaches for integrating punctuation into speech translation, and our original method based on confusion networks. Section 4 presents the experimental settings, the used evaluation criteria, and the questions addressed by our investigation. Experimental results corresponding to each single addressed question are finally reported and discussed.

## 2. Speech Translation Approach

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 approximate criterion:

$$\mathbf{e}^* = \arg \max_{\mathbf{e}} \max_{\mathbf{f} \in \mathcal{F}(\mathbf{o})} \exp \left\{ \sum_{r=1}^R \lambda_r h_r(\mathbf{e}, \mathbf{f}, \mathbf{o}) \right\} \quad (1)$$

where the source language sentence  $\mathbf{f}$  is an hidden variable representing any transcription hypothesis.

The well established log-linear framework has been considered, because it enables the use and the balance of any kind of features  $h_r(\mathbf{e}, \mathbf{f}, \mathbf{o})$ , regarded as important for the sake of translation and suitable for the type of input. Currently, better performance is achieved by defining features in terms of *phrases* [2, 3, 4] instead of single words.

### 2.1. Confusion Network

Although a word graph contains all transcription alternatives considered during the ASR process, its topology is very complex. Hence, a simpler and more compact way of representing these alternatives is achieved through a confusion network (CN) [5], also known as *sausage*, is preferable. A CN is still a weighted directed graph with the peculiarity that each path from the start node to the end node goes through all the other nodes, and that words and posterior probabilities are associated to edges. In principle, a CN can contain more hypotheses than the word graph. A CN is represented as a table of words whose columns have different depths as shown in Figure 1. Any path within the CN represents a transcription alternative and is scored with a posterior probability. Trivially, a CN can be used to represent plain text as well; hence, the SLT decoder can be applied to single transcriptions, too.

The extraction of a CN from a word graph can also produce special empty-words  $\epsilon$  in some columns. These empty-words permit to generate transcription hypotheses of different length and are treated differently from regular words only at the level of feature functions. CNs are computed by means of the `lattice-tool`, available in the SRILM toolkit [6].

### 2.2. Model and CN decoder

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

1. A word-based 5-gram target LM.
2. A reordering model defined in terms of the distance between the first column covered by current span and the

i. <sub>9</sub>	cannot. <sub>8</sub>	ε. <sub>7</sub>	say. <sub>6</sub>	ε. <sub>7</sub>	anything. <sub>8</sub>	at. <sub>9</sub>	this. <sub>8</sub>	point. <sub>7</sub>	are <sub>1</sub>	there. <sub>8</sub>	ε. <sub>8</sub>	any. <sub>7</sub>	comments. <sub>7</sub>
hi. <sub>1</sub>	can. <sub>1</sub>	not. <sub>3</sub>	said. <sub>2</sub>	any. <sub>3</sub>	thing. <sub>1</sub>	ε. <sub>1</sub>	these. <sub>1</sub>	points. <sub>1</sub>		the. <sub>1</sub>	a. <sub>1</sub>	new. <sub>1</sub>	comment. <sub>2</sub>
	ε. <sub>1</sub>		say. <sub>1</sub>		things. <sub>1</sub>		those. <sub>1</sub>	ε. <sub>1</sub>		their. <sub>1</sub>	air. <sub>1</sub>	a. <sub>1</sub>	commit. <sub>1</sub>
			ε. <sub>1</sub>					pint. <sub>1</sub>				ε. <sub>1</sub>	

Figure 1: Example of confusion network.

last column of the previous span. (In the current implementation, we did not distinguish between regular and empty words.)

- Four phrase-based lexicon models exploiting statistics at word- and phrase-level. These models remove any empty-word in the source side.
- Phrase and word penalty models, i.e. counts of the number of phrases and words in the target string.
- The posterior probability of the covered span.

A decoding algorithm for CN-based speech translation was first proposed in [7]. Recently, a more efficient implementation was developed during the JHU Summer Workshop 2006 and integrated into an open source toolkit, called `moses` [8, 9, 10].

### 3. Punctuation Insertion

The problem of punctuation restoration for ASR-provided utterances has been investigated in [11]. For predicting punctuation the authors explored three alternative strategies, on the basis of the presence/absence of punctuation in the phrase tables:

- Restore punctuation on target language:* the training corpus does not include any punctuation marks; translation is performed from un-punctuated input to un-punctuated output; restoration of punctuation is performed on the target language after decoding.
- Restore punctuation implicitly:* the training corpus includes punctuation marks in the target language only; translation is performed from un-punctuated input to punctuated output; restoration of punctuation is performed implicitly during decoding.
- Restore punctuation on source language:* the training corpus include punctuation marks on both languages; translation is performed from punctuated input to punctuated output; restoration of punctuation is performed on the source language before decoding.

For punctuation prediction either in the source or in the target language, they used a so-called *hidden-event language model*. Briefly, this approach exploits an hidden Markov model in which hidden states model text including punctuation marks, while observations do only include words. State transitions are modeled with an ordinary  $n$ -gram language model. Hidden-event language model can be implemented with the `hidden-ngram` tool available in the SRILM Toolkit [6].

In this paper we use the same three strategies as starting point of our investigation, with some relevant differences, though. First, the type of input provided by the ASR is different: our purpose is to cope with the multiple hypotheses encoded in the CNs, while in [11] the (textual) first-best hypothesis is taken into account. The second difference regards the use of the `hidden-ngram` tool: we aim at exploiting as well the multiple hypotheses on hidden events (i.e. punctuation marks) that such

tool can provide. These multiple hypotheses are represented with CNs as well.

In our work, the use of CNs is investigated at two independent levels: at the transcription level CNs encode uncertainty by the ASR decoder; at the level of punctuation, CNs capture uncertainty by the `hidden-ngram` decoder.

As far as the strategy of restoring punctuation on the source language is concerned, the method we implemented to insert punctuation marks in a CN includes a sequence of steps:

- the consensus decoding is extracted from the CN;
- multiple hypotheses of punctuation marks are generated by means of the `hidden-ngram` model on the consensus decoding;
- a CN with punctuation marks is created from the multiple hypotheses provided by the tool;
- this punctuated CN is merged with the original CN.

Figure 2 shows the method applied to the example CN reported in Figure 1. The upper part presents the consensus decoding extracted from the CN: it is a simple sentence without punctuation marks. In the second part of the figure the 10-best hypotheses generated by `hidden-ngram` are shown: each hypothesis consists of a score and of the sentence enriched with punctuation marks. In this example the hidden vocabulary is composed by strong punctuation marks only (namely full stop, question and exclamation marks). The marks have been possibly added only in three positions of the sentence: after the words "anything" and "point", and at the end. The third part of the figure shows the CN that encodes the multiple hypotheses on punctuation: the posterior probability of a word – including punctuation marks – is the normalized sum of the probabilities of the hypotheses containing such word. Therefore, for non-punctuation words it is always 1.0. Concerning punctuation marks, there is high probability for a full stop to appear after the word "point" and for a question mark at the end of the sentence. On the other side, the probability that a full stop appears after the word "anything" is quite low. The figure finally shows the CN obtained by merging the punctuated CN with the original CN of Figure 1.

## 4. Experiments

### 4.1. Setup

We evaluated the proposed punctuation method on the TC-STAR 2006 Evaluation<sup>1</sup> task for English-to-Spanish speech translation.

Training data consist of the Final Text Edition (FTE) of recordings of political speeches acquired during some European Parliament Plenary Sessions (EPPS). The corpus contains a total of 36M English and 38M Spanish running words; English and Spanish dictionaries contain 116K and 149K words, respectively. The same EPPS corpus was used to estimate English and Spanish 3-gram language models: the former was used to

<sup>1</sup><http://www.tcstar.org>

i	cannot	say	anything	at	this	point	are	there	any	comments
---	--------	-----	----------	----	------	-------	-----	-------	-----	----------

(i)

NBEST.0	-15.2699	i cannot say anything	at this point	.	are there any comments	
NBEST.1	-15.3172	i cannot say anything	at this point	.	are there any comments	?
NBEST.2	-16.2751	i cannot say anything	at this point	.	are there any comments	?
NBEST.3	-16.3224	i cannot say anything	at this point	?	are there any comments	?
NBEST.4	-17.829	i cannot say anything	at this point	.	are there any comments	.
NBEST.5	-18.2841	i cannot say anything	at this point	?	are there any comments	.
NBEST.6	-18.3313	i cannot say anything	at this point	.	are there any comments	.
NBEST.7	-18.4734	i cannot say anything	.	at this point	are there any comments	.
NBEST.8	-18.5207	i cannot say anything	.	at this point	are there any comments	?
NBEST.9	-18.8342	i cannot say anything	at this point	.	are there any comments	.

(ii)

i <sub>1</sub>	cannot <sub>1</sub>	say <sub>1</sub>	anything <sub>1</sub>	ε <sub>.9</sub>	at <sub>1</sub>	this <sub>1</sub>	point <sub>1</sub>	. <sub>.7</sub>	are <sub>1</sub>	there <sub>1</sub>	any <sub>1</sub>	comments <sub>1</sub>	? <sub>.6</sub>
				. <sub>.1</sub>				ε <sub>.2</sub>					ε <sub>.3</sub>
								? <sub>.1</sub>					. <sub>.1</sub>

(iii)

i <sub>.9</sub>	cannot <sub>.8</sub>	ε <sub>.7</sub>	say <sub>.6</sub>	ε <sub>.7</sub>	anything <sub>.8</sub>	ε <sub>.9</sub>	at <sub>.9</sub>	this <sub>.8</sub>	point <sub>.7</sub>	. <sub>.7</sub>	are <sub>1</sub>	there <sub>.8</sub>	ε <sub>.8</sub>	any <sub>.7</sub>	comments <sub>.7</sub>	? <sub>.6</sub>
hi <sub>.1</sub>	can <sub>.1</sub>	not <sub>.3</sub>	said <sub>.2</sub>	any <sub>.3</sub>	thing <sub>.1</sub>	. <sub>.1</sub>	ε <sub>.1</sub>	these <sub>.1</sub>	points <sub>.1</sub>	ε <sub>.2</sub>		the <sub>.1</sub>	a <sub>.1</sub>	new <sub>.1</sub>	comment <sub>.2</sub>	ε <sub>.3</sub>
	ε <sub>.1</sub>		say <sub>.1</sub>	ε <sub>.1</sub>	things <sub>.1</sub>			those <sub>.1</sub>	ε <sub>.1</sub>	? <sub>.1</sub>		their <sub>.1</sub>	air <sub>.1</sub>	a <sub>.1</sub>	commit <sub>.1</sub>	. <sub>.1</sub>
			ε <sub>.1</sub>					pint <sub>.1</sub>						ε <sub>.1</sub>		

(iv)

Figure 2: The method for punctuation marks restoration applied to the example CN shown in Figure 1: first the consensus decoding is extracted from the original CN (i); a 10-best list with punctuation hypotheses is generated by `hidden-ngram` (ii); a punctuated CN is created from the 10-best list (iii); the final CN is obtained by merging the punctuated CN with the original CN (iv).

train the *hidden-ngram* model for punctuation restoration on the source language, the latter both for decoding and for punctuation restoration on the target language.

Two test sets are used for the evaluation: the 2006 development (dev06) and evaluation (eval06) sets. For both sets, human verbatim transcriptions and automatic ASR transcriptions are available. Concerning the source side, the number of running words on the human transcriptions is about 30K for each set, while the dictionary size is about 3.9K word. In the ASR transcriptions, running words of the consensus decoding reduce to 28K for the dev06 set and to 29K for the eval06, mostly due to missing punctuation marks. The average depth of the CNs is 1.94 and 2.15, respectively. Regarding the target language, there are two reference translations for each set.

For the experiments only a single-pass search was used (i.e. the MT decoder). Moreover, translation evaluation was performed with automatic scores in the case-insensitive condition to point out impact of punctuation insertion only. As automatic scores we employed the BLEU, NIST, word error rate (WER), and position-independent error rate (PER).

## 4.2. Evaluation

Our experiments aimed at answering the following questions:

1. Where is it better to add punctuation marks: in the source language, in the target language or implicitly in the phrase tables?
2. Do multiple punctuation hypotheses help to improve translation performance?
3. What is the effect of using multiple recognized speech hypotheses, enhanced with punctuation marks, on translation quality?

set	punct. method	BLEU	NIST	WER	PER
dev06	target	43.91	9.89	45.46	33.46
	implicit	44.86	9.90	42.81	31.73
	source	46.42	10.02	42.26	30.88
eval06	target	42.23	9.72	46.12	34.38
	implicit	42.85	9.67	43.93	32.98
	source	44.92	9.84	42.84	31.77

Table 1: Translation results of the three different methods for punctuation marks restoration on clean text.

The first question was already addressed by [11], but directly on speech input (first-best recognized hypotheses): here, differently, we performed experiments on verbatim human transcriptions, in order to rule out the effect of recognition errors. Table 1 reports the translation results of the three different methods for punctuation marks restoration. The result is that the strategy of adding punctuation in the source language outperforms the other two methods on both sets and with all considered scores. In particular, the gain in BLEU score is at least 1.5 points absolute.

Going on with the most promising strategy, namely adding punctuation on the source language, the experiments for the second question were again performed on verbatim transcripts, for the same reason. Table 2 reports the translation results with different number of punctuation hypotheses. The use of 1000-best hypotheses appears to produce better quality translations with respect to using the first-best only: even though the PER score slightly decreases and the NIST score remains practically constant, the BLEU and WER scores improve in both sets.

Moving from clean text, i.e. human transcriptions, to ASR provided input, Table 3 presents the results of the experiments

dev06				
# punctuation hypotheses	BLEU	NIST	WER	PER
1	46.42	10.02	42.26	30.88
1000	46.62	10.01	42.15	31.13

eval06				
# punctuation hypotheses	BLEU	NIST	WER	PER
1	44.92	9.84	42.84	31.77
1000	45.33	9.83	42.58	31.59

Table 2: Translation results when using a different number of hypotheses in restoring punctuation marks on the source side by means of the `hidden-ngram` tool. Multiple hypotheses are encoded with CNs. Results are on human transcriptions.

dev06					
ASR type	# punct. hyp	BLEU	NIST	WER	PER
1-best	1	38.61	8.89	51.34	38.33
	1000	39.08	8.96	50.93	38.21
CN	1	39.10	8.97	50.31	37.80
	1000	39.51	9.00	50.06	37.73

eval06					
ASR type	# punct. hyp	BLEU	NIST	WER	PER
1-best	1	35.62	8.37	57.15	44.56
	1000	36.01	8.41	56.78	44.39
CN	1	36.22	8.46	56.39	44.37
	1000	36.45	8.49	56.17	44.19

Table 3: Translation results on ASR input. For each type of input provided by the ASR engine (1-best hypothesis or lattice encoded with CN), different number of punctuation hypotheses (single or multiple hypotheses encoded with CNs) have been tested.

aimed at answering the third and final question: what is the effect of using multiple hypotheses on ASR input? First row of both set presents the simplest case: adding the first-best punctuation mark hypothesis to the first-best ASR input. In the next rows multiple hypotheses are taken into account with increasing complexity: in the second row the 1000-best punctuation hypotheses are considered for the first-best ASR input. Multiple hypotheses on the recognized speech (CN) with first-best punctuation mark hypothesis are employed for the third row. Finally, in the fourth row multiple hypotheses are used for both recognized speech and punctuation marks insertion. Results show that the higher the complexity the better the performance, for all the scores. The total increment on BLEU score is at least 2.2% relative.

From the presented experiments we can draw some conclusions. First, CNs are an effective tool to capture uncertainty on the punctuation marks provided by the `hidden-ngram` decoder. Translation quality improves in all the conditions – clean text, first-best ASR input and ASR word graph – when using multiple hypotheses on punctuation marks. Second, CNs are also effective to encoded uncertainty on automatic transcriptions provided by the ASR decoder. The translation scores obtained with CNs are always better than those obtained with first-best ASR input. Finally, the two ways to employ the CNs (to encode uncertainty by ASR and `hidden-ngram` decoders) do not conflict: the best result we get with ASR input derives from the combined application of the two techniques.

## 5. Conclusions

We discussed automatic methods for enriching speech translations with punctuation marks. Starting from previous work on speech translation through confusion network decoding, we presented a method for augmenting a confusion network, that represents multiple ASR hypotheses, with multiple punctuation hypotheses generated by an `hidden-ngram` model. We presented results on a large-vocabulary speech translation task from English to Spanish.

In the future, we would like to investigate the portability of the approach to other language pairs. Also the use of other decodings in addition to the consensus one appears to be promising.

## 6. 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>).

## 7. References

- [1] W. Wang, K. Knight, and D. Marcu, “Capitalizing machine translation,” in *Proc. of HLT/NAACL, Main Conference*, New York City, USA, 2006, pp. 1–8.
- [2] R. Zens, F. J. Och, and H. Ney, “Phrase-Based Statistical Machine Translation,” in *Annual German Conference on AI, KI 2002*, September 2002, pp. 18–32.
- [3] P. Koehn, F. J. Och, and D. Marcu, “Statistical Phrase-Based Translation,” in *Proc. of HLT/NAACL 2003*, Edmonton, Canada, 2003, pp. 127–133.
- [4] M. Federico and N. Bertoldi, “A word-to-phrase statistical translation model,” *ACM Transactions on Speech and Language Processing (TSLP)*, vol. 2, no. 2, pp. 1–24, December 2005.
- [5] L. Mangu, E. Brill, and A. Stolcke, “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.
- [6] A. Stolcke, “SRILM - an extensible language modeling toolkit,” in *Proc. of ICSLP*, Denver, Colorado, 2002.
- [7] N. Bertoldi and M. Federico, “A New Decoder for Spoken Language Translation based on Confusion Networks,” in *Proc. of ASRU*, San Juan, Puerto Rico, December 2005.
- [8] W. Shen, R. Zens, N. Bertoldi, and M. Federico, “The JHU Workshop 2006 IWSLT System,” in *Proc. of the International Workshop on Spoken Language Translation*, Kyoto, Japan, 2006, pp. 59–63.
- [9] N. Bertoldi, R. Zens, and M. Federico, “Speech Translation by Confusion Network Decoding,” in *Proc. of ICASSP*, Honolulu, Hawaii, USA, April 2007.
- [10] P. Koehn, et al., “Moses: Open source toolkit for statistical machine translation,” in *Proc. of ACL, demonstration session*, Prague, Czech Republic, June 2007.
- [11] E. Matusov, A. Mauser, and H. Ney, “Automatic Sentence Segmentation and Punctuation Prediction for Spoken Language Translation,” in *Proc. of IWSLT*, Kyoto, Japan, 2006.