Improved Machine Translation of Speech-to-Text outputs

Daniel Déchelotte, Holger Schwenk, Gilles Adda, Jean-Luc Gauvain

LIMSI/CNRS, Bâtiment 508, Université Paris-Sud, France

dechelot,schwenk,gadda,gauvain@limsi.fr

Abstract

Combining automatic speech recognition and machine translation is frequent in current research programs. This paper first presents several pre-processing steps to limit the performance degradation observed when translating an automatic transcription (as opposed to a manual transcription). Indeed, automatically transcribed speech often differs significantly from the machine translation system's training material, with respect to caseing, punctuation and word normalization. The proposed system outperforms the best system at the 2007 TC-STAR evaluation by almost 2 points BLEU. The paper then attempts to determine a criteria characterizing how well an STT system can be translated, but the current experiments could only confirm that lower word error rates lead to better translations.

Index Terms: ASR, MT, segmentation, punctuation, normalization, joint optimization

1. Introduction

Automatic text translation is a challenging task that researchers and engineers have been tackling for more than 40 years. Yet, even though translating well-formed texts is still a largely unanswered problem, recent and current projects [1, 2, 3] raise further up the bar as they require the translation systems to process the output of automatic speech recognition (ASR) systems. This article addresses two problems that specifically arise when a system combines ASR and machine translation (MT). The first difficulty is the various discrepancies between the training material used by the MT system and the data being actually translated during testing. The second open question relates to the tuning of the ASR system in order to globally maximize the system's end-to-end performance from the audio to the text in the target language. Both problems are discussed below.

Translating ASR output is arguably quite a different problem than translating formal texts. The literature already provides several discussions on mitigating the impact of recognition errors, for instance by handling ASR's ambiguous output (e.g., under the form of word lattices). In the first part of this article, we describe how our system addresses the following three items:

1. The transcription may contain *disfluencies*, which naturally occur in speech. MT systems should recognize and ignore hesitations, repetitions and false starts.
2. Speech is not explicitly *segmented* into sentences, and recovering that segmentation automatically is a challenging task that involves both acoustical and linguistic information. Moreover, clear conventions are lacking with respect to the placement of finer *punctuation marks* (e.g., commas, columns) and ASR system are therefore often built and tuned without punctuation marks.
3. Finally, ASR's *text normalization* might differ from the one expected by the MT system, which could lead to suboptimal translation. Translating the output from external ASR systems, or the ROVER combination of various systems, is particularly subject to this pitfall.

The second part of this work aims at confirming or infirming the intuition that a STT system that *breaks phrases* might lead to poorer translations when translated by a phrase-based MT system than a STT system of comparable word error rate (WER) that preserves more phrases, as suggested by [4].

The translation performance (as measured with BLEU) of a STT+MT system correlates generally well with ASR's WER [5, 6], a result that matches the intuitive assumption that the best overall performance is obtained when translating the ASR output of lowest WER. However, [5] compared different outputs produced by a single system using consensus decoding across all experiments.

In this work, translating the ROVER combination of various speech-to-text (STT) systems is compared to translating the output of a single system, performing consensus decoding or not. The translation performance appears to remain strongly correlated with ASR WER, although a tiny "inversion" (worse ASR WER but better translation) was observed once.

The paper is organized as follows. Section 2 describes the data and the system used in all the experiments. Section 3 details the steps that make the ASR output resemble MT's training data, and Section 4 describes the different STT systems considered for translation. Lastly, Section 5 presents the results of the two sets of experiments and discusses them.

2. Experimental framework

2.1. Data

The task considered in this work is the translation of the European Parliament Plenary Sessions (EPPS) from English to Spanish, in the framework of the TC-STAR project. The latter is envisaged as a long-term effort to advance research in all core technologies for speech-to-speech translation. The proposed MT system participated to the 2007 TC-STAR international evaluation campaign, translating between Spanish and English (both ways), under the "verbatim" condition (manual transcriptions of the acoustic data) and the "ASR" condition (automatic transcriptions)[1].

The figures reported in this article were obtained on the 2006 development set and the 2007 evaluation set.

[1] See http://www.elda.org/en/proj/tcstar-wp4/ for details on the specifications and the available training data.

2.2. System description

The MT system used in this paper is built upon the open-source, state-of-the-art phrase-based decoder Moses[7], and was trained with the scripts distributed with the software package.

The translation process employs a two-pass strategy. In the first pass, Moses generates n-best lists—1000 distinct hypotheses are requested—with a standard 3-gram language model and provides eight partial scores for each hypothesis. In the second pass, the n-best lists are rescored with a 4-gram continuous space language model[8] and the final hypothesis is then extracted. Each of the two passes uses its own set of eight weights and is tuned separately.

3. Making ASR output resemble MT's training data

In this section, we present the steps that attempt to renormalize the ASR's output so that it matches more closely the translation training data. The ASR's output is available under the "time marked conversation" CTM format, which holds some time and duration information along with the words themselves.

3.1. Case and punctuation

STT systems historically produced a case insensitive output, without punctuation or sentence segmentation. Nowadays, even if the reference transcription is punctuated and in true case, the *de facto* reference score remains the case insensitive, unpunctuated WER, because of remaining standardization issues both in capitalization (for words like "Project", "Program", "Committee", etc) and punctuation (for all punctuation marks but especially the comma). As a consequence, in order to achieve better translations, it is often necessary to re-punctuate and optionally to re-case STT's outputs or their ROVER combination.

[9] preserves the segmentation provided by the ASR engine and inserts commas based on linguistic features (bi- and tri-gram probabilities). [10] explores different strategies; on the same task as this article, predicting first the sentence segmentation and then the punctuation performs well. In this work, we first remove any case information and punctuation marks the input CTM file may contain, and recompute them in one pass as follows.

Both the words and the timing information contained in the input CTM are used to generate a flat, consensus-like lattice. More precisely, each word in the CTM file leads to the creation of one node and three edges, to account for its three alternative capitalizations (all-lowercase, capitalized or all-uppercase). Between words, edges are created as shown in Figure 1 to optionally insert a period or a comma.

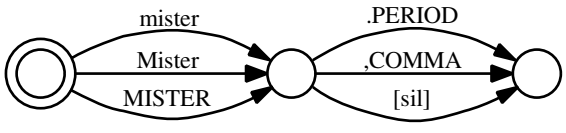

Figure 1: Sample lattice generated for one word of the input CTM file. "[sil]" is a spontaneous transition inserting no word.

The constituted lattice is then rescored by a special-purpose language model. This language model is created by interpolating several language models trained on the same data but using purposely different word normalizations with respect to compound words (prefixes and suffixes hyphenated or separated) and acronyms (spelled out or in one token). This allows the resulting language model to be able to restore the case and the punctuation for texts of unknown word normalization.

Frequencies of periods and commas have been computed over the EPPS training corpus and the available development data to provide characteristics to aim for. It was estimated that 3.5% of all tokens were periods and 5% were commas, amounting to roughly one period every 29 tokens and one comma every 20 tokens. Two quantities may be tuned in the proposed algorithm:

1. the penalty (or bonus) held by the edges with a punctuation mark (see lattice excerpt in Figure 1),
2. a duration τ used as follows: a mandatory period is inserted at pauses longer than τ.

These two quantities were tuned minimizing the sum of the quadratic errors (the differences between the target frequencies and the observed ones). A local optimum that achieved 3.8% of periods and 4.8% of commas was reached; at this optimum, $\tau = 1.6$ seconds.

3.2. Disfluency removal, and normalizations

Hesitations and filler words are easy to spot and remove—the complete list is `eh`, `uh`, `uhm`, `huh` and `mmm`. Additionally, repeated words are removed, leaving only the first occurrence.

The normalization required for the MT system might differ from the one of the STT system, for example for key words like `Mister` or `Mrs.`. Spelled out acronyms (`N. A. T. O.`) are consolidated into a single word, as they appear in the MT training data.

Lastly, STT's output might contain a relatively high frequency of contracted forms, such as `it's` or `can't`. It may indeed be beneficial for a STT system to output, e.g., `it's` since during scoring a global map (GLM) file will make it also match `it is` and `it has` as needed. Those contracted forms are present in the EPPS data, but at a much lower frequency, and are therefore expanded before translation. In this work, ambiguous forms were deterministically expanded to an arbitrary, but likely, form. For example, `it's` was systematically expanded to `it is` and `I'd` to `I would`.

3.3. Recomposition of compound words

Part of the normalization differences between an STT system and an MT system can be the way they deal with compound words. Some prefixes, like `pro-`, `anti-`, `trans-` and others, as well as some suffixes like `-like`, are extremely generative, meaning that they are susceptible to be associated with many words, enlarging significantly the vocabulary size. For an STT system, handling all these compounds can quickly become a burden, since pronunciations have to be generated for each of them, and the language model may face data sparseness issues. Moreover, the scoring tool usually splits words with hyphens, so there are no adverse effects of producing `pro-European` as one word or `pro European` as two words. MT systems on the other hand are expected to produce compounds in one word, hence they were not split in the training data.

Instead, a tool was designed to recover the compound words when needed. This tool uses n-gram counts extracted from the training data used by the MT system to, e.g., produce the compound word `pro-US` should this unigram be more frequent in the training data than the bi-gram `pro US`. Compounds with

up to three hyphens (such as `end-of-the-year`) may be recreated this way.

4. Translating the output of different STT systems

This section considers three STT systems and translates their output. In these experiments, all the processing steps described in the previous section are applied before translation.

4.1. STT systems

The first considered system is the ROVER combination of various ASR systems. The experiments on the test data uses the official TC-STAR ROVER combination, and a combination of the same systems was performed internally for the experiments on the development data.

The second system is the Limsi system [11], which performs consensus decoding [12] (CD) to produce its 1-best hypothesis. The third system consists of the same system using maximum a posteriori (MAP) decoding, without CD.

4.2. Evaluating the ASR outputs

In addition to the standard WER, a BLEU score [13] is computed against the reference transcription for each ASR output. The tool from [14] performed the necessary automatic resegmentation.

All scores are computed in a case insensitive manner and ignoring punctuation, which is consistent with the fact that the data is re-cased and re-punctuated before the actual translation.

5. Results and discussion

5.1. On the impact of weight tuning

As said earlier in this paper, our MT system may be tuned thanks to two sets of eight weights, one set per pass. Because tuning first-pass weights is time consuming, and since we had carefully tuned a system for the verbatim condition, we decided to re-use its first-pass weights for the ASR condition system. The second-pass weights were however tuned specifically for the ASR condition, yielding a consistent increase of 0.35 to 0.40 absolute %BLEU in all cases on the development set. The improvement was confirmed on unseen data, albeit with lower gains ranging from 0.15 to 0.20. Consequently, all results on the test set reported in this paper use the ASR-specific weights.

However, we informally reran the experiments on the test set with the verbatim weights and found that they consistently outperformed the ASR weights by 0.15 to 0.25 depending on the case. Although those discrepancies are well below the significance threshold, it might be worth investigating the stability of function points, especially with an inherently unreliable input such as an ASR output.

5.2. Experiments on renormalizing the ASR output

The three procedures described at Section 3 are evaluated systematically on the development data and the test data. Because the usefulness of the renormalizing scripts is likely to depend on the input type, their effects on the ROVER and the Limsi outputs[2] have been compared.

[2]Table 1 shows the ASR WER of the four CTMs translated in these experiments.

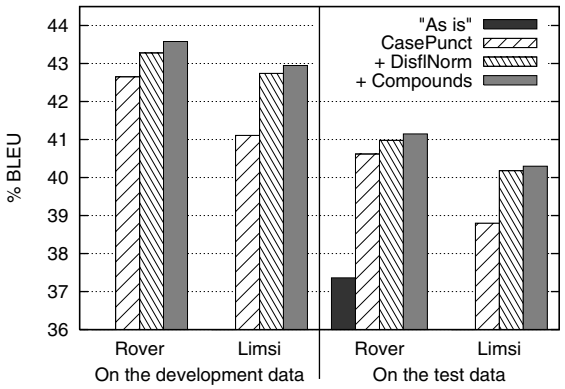

Figure 2: Relative performances of the preprocessing steps described at Section 3. "As is": translation of the CTM file provided by the evaluation committee, only available for the ROVER output on the evaluation data. CasePunct: prior to translation, restoration of case and punctuation. DisflNorm: disfluency removal and renormalization. Compounds: compound recomposition.

5.2.1. Impact of case and punctuation

Because the script used by the evaluation committee to restore the case and some punctuation is unknown, it was difficult to assess the usefulness of our own procedure before the evaluation took place. Afterwards, however, it became possible to compare the performances of translating the provided CTM file "As is" or after applying "CasePunct" (see Figure 2). With an increase greater than 3 points BLEU, from 37.4 to 40.6, this step appears to play a crucial role in reducing the mismatch between a typical ASR output and what our MT system expects to perform at its best[3].

To account for the performance discrepancy, we computed the frequencies of commas and periods on the two inputs "As is" and "CasePunct". The CTM file "As is" contained 3.18% of periods and a mere 0.46% of commas, as opposed to 4.09% of periods and 4.99% of commas in the CTM after "CasePunct". The lack of commas in the "As is" file has two effects. First, its translation does not contain enough tokens and its BLEU score is severely hit by a brevity penalty of 0.934, whereas it reaches 0.981 after "CasePunct". Second, even without applying the brevity penalty, "CasePunct" achieves better precision scores and obtains an unpenalized BLEU score of 41.4, although "As is" only scores 40.0. This indicates that not only the punctuation helps avoid the BLEU brevity penalty, it is also useful to pick the right phrases and, eventually, to produce a correct translation.

5.2.2. Impact of disfluency removal and renormalization

The impact of "DisflNorm" is positive in all cases, although its importance varies. On ROVER output, "DisflNorm" provides gains of 0.6 on the development data and 0.4 on the test data. Most of the changes consists of acronym restorations and contracted form expansions. On the Limsi output, the kinds of

[3]To put those number into perspective, our official system, which included preliminary versions of "DisflNorm" and "Compounds" but not "CasePunct", achieved a BLEU score of 37.6, and the best submitted system obtained 39.2.

changes are quite different, with the expansion of contracted form representing most of the changes, followed by the renormalization of `Mr.` and `Mrs.` and the handling of acronyms, yielding 1.6 points BLEU on the development data and 1.4 on the test data.

5.2.3. Impact of compound recomposition

"Compounds" modified half less words than "DisflNorm" did, and its impact in BLEU is even slighter, although always positive, with gains ranging from just above 0.1 to 0.3. It noticeably allowed the MT system to correctly translate numbers like `twenty two` into `veintidós` instead of `veinte dos`.

5.3. Experiments with different STT systems

Set	ASR			MT	
	System	WER	BLEU	# phr.	BLEU
Dev06	Rover	7.18	70.22	2231k	43.58
	Limsi CD	9.14	63.98	2260k	42.95
	Limsi MAP	9.53	63.92	2264k	43.05
Eval07	Rover	7.08	67.92	2103k	41.15
	Limsi CD	9.33	61.29	2123k	40.30
	Limsi MAP	9.66	61.14	2130k	40.19

Table 1: Translation of different STT systems. ASR columns: the system's name, its WER and its BLEU score against the manual transcription. MT columns: the size of the filtered phrase table and the translation BLEU score.

Table 1 gathers the results obtained in this series of experiments. The upper-half of the table (dev06) contains a surprising inversion: although Limsi MAP has as expected a higher WER than Limsi CD, its translation is better by a short margin. In an attempt to explain this inversion, we compared the number of phrase pairs whose source counterpart is included in the input text. Limsi MAP "recruits" slightly more phrase pairs than Limsi CD and ROVER, which is in no way an indicator of better translations but tends to confirm our intuition that performing CD on a single system "breaks phrases" and performing a ROVER combination even more so. We then computed the ASR-BLEU for all systems, since this score also takes into account *phrases* of up to four words. However, Limsi CD was found to score a higher BLEU than Limsi MAP. In addition, the inversion did not occur on the eval07 data, preventing us from drawing any definitive conclusion at that point, except that ASR-WER remains in these experiments the best indicator of the overall ASR+MT performance.

6. Conclusion

In the context of machine translation of automatic speech recognition output, we first proposed several processing steps that modify the ASR output so that it resembles the MT's training data with respect to caseing, sentence segmentation, punctuation and word normalization, allowing the described system to outperform the best system at the 2007 TC-STAR evaluation by almost 2 points BLEU. We then carried out experiments to determine a criteria characterizing how well an STT system can be translated, but we were only able to observe that lower ASR-WERs lead to better translations.

7. Acknowledgments

This work was partially funded by the European Union under the integrated project TC-STAR (IST-2002-FP6-506738).

8. References

[1] W. Wahlster, *Verbmobil: Foundations of Speech-to-Speech Translation*. Springer verlag, 2000.

[2] G. Kikui, E. Sumita, T. Takezawa, and S. Yamamoto, "Creating corpora for speech-to-speech translation," in *Proc. of Eurospeech*, 2003.

[3] "Human language technologies for Europe," Download: http://www.tc-star.org/pubblicazioni/D17_HLT_ENG.pdf, April 2006.

[4] M. Gales and al., "Speech recognition system combination for machine translation," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Honolulu, USA, 2007.

[5] D. Déchelotte, H. Schwenk, J.-L. Gauvain, O. Galibert, and L. Lamel, "Investigating translation of parliament speeches," in *Proc. of Automatic Speech Recognition and Understanding*, San Juan, Porto Rico, November 2005.

[6] H. A. Engelbrecht and T. Schultz, "Rapid development of an afrikaans-english speech-to-speech translator," in *Proc. of International Workshop on Spoken Language Translation*, Pittsburgh, USA, October 2005.

[7] P. Koehn and al., "Moses: Open source toolkit for statistical machine translation," in *Annual Meeting of the Association for Computational Linguistics (ACL), demonstration session*, Prague, Czech Republic, June 2007.

[8] H. Schwenk and J.-L. Gauvain, "Neural network language models for conversational speech recognition," in *Proc. of the International Conference on Spoken Language Processing*, 2004.

[9] Y.-S. Lee, S. Roukos, Y. Al-Onaizan, and K. Papineni, "IBM spoken language translation system," in *Proc. of the TC-STAR Workshop on Speech-to-Speech Translation*, Barcelona, Spain, 2006.

[10] E. Matusov, A. Mauser, and H. Ney, "Automatic sentence segmentation and punctuation prediction for spoken language translation," in *Proc. of the International Workshop on Spoken Language Translation*, Kyoto, Japan, 2006.

[11] L. Lamel and al., "The Limsi 2006 TC-STAR EPPS transcription systems," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Honolulu, USA, 2007.

[12] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus among words: Lattice-based word error minimization," in *In Proc. Eurospeech '99*, Budapest, 1999.

[13] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation." in *Proc. of the 40th Annual Meeting of the ACL*, University of Pennsylvania, 2002.

[14] E. Matusov, G. Leusch, O. Bender, and H. Ney, "Evaluating machine translation output with automatic sentence segmentation," in *Proc. of International Workshop on Spoken Language Translation*, Pittsburgh, USA, 2005.