ASR ERROR SEGMENT LOCALIZATION FOR SPOKEN RECOVERY STRATEGY

Frédéric Béchet, Benoit Favre

LIF/CNRS, Aix-Marseille University, 163 avenue de Luminy, Marseille, France
{frederic.bechet,benoit.favre}@lif.univ-mrs.fr

ABSTRACT

Even though small ASR errors might not impact downstream processes that make use of the transcript, larger error segments like those generated by OOVs can have a considerable impact on applications such as speech-to-speech translation and can eventually lead to communication failure between users of the system. This work focuses on error detection in ASR output targeted towards significant error segments that can be recovered using a dialog system. We propose a CRF system trained to recognize error segments with ASR confidence-based, lexical and syntactic features. The most significant error segment is passed to a dialog system for interactive recovery in which rephrased words are reinserted in the original. 22% of utterances can be fully recovered and an interesting by-product is that rewriting error segments as a single token reduces WER by 17% on an adverse corpus.

Index Terms— Automatic Speech Recognition, Confidence Measure, Error Detection, Speech to Speech translation

1. INTRODUCTION

Current state-of-the-art speech recognition systems make errors that have to be identified in order to apply appropriate strategies for performing communicative and system actions, such as error correction and repair dialogs. In ASR, the decoding strategy finds a sequence of words which has the maximum posterior probability (i.e. confidence) of being conveyed by the speech signal. In applications such as speech-to-speech translation, if this word string contains important ASR errors (that affect semantics in a broad sense), the impact on the translation process can be huge, leading to a failure in the interaction. Detecting these *significant* ASR error segments before sending a word string to the translation process is crucial.

Once an error segment in an automatic transcription has been detected, it is possible to apply a recovery strategy that can use external knowledge, contextual information or a user interaction in order to recover the missing information from the original utterance. In the framework of the DARPA BOLT project we present a new method for detecting specifically these *significant* error segments in ASR hypotheses by considering this problem as a sequence labeling task and filtering the detections. An novel application of this method to a clarification dialog strategy for a Speech to Speech (S2S) task is also proposed and evaluated on TRANSTAC and BOLT data.

This work was partially funded by DARPA HR0011-12- C-0016 as an AMU subcontract to SRI International.

2. RELATED WORK

Estimating the confidence of an ASR hypothesis raises several issues: choosing the span of the confidence measures [1] (word, conceptual constituent or utterance), defining the set of features involved in the confidence estimation (ASR features, syntactic features, contextual information), combining efficiently the different features and choosing a decision strategy that takes into account all the features obtained [2]. The majority of the approaches share two basic steps: generate as many features as possible based on the ASR and/or automatic process of the transcriptions; estimate correctness probabilities with these features.

The problem of detecting ASR error segments is linked to the OOV segment detection task, since all OOV words necessarily generate at least one ASR error, and more often a sequence of ASR errors (the OOV word being replaced by short in-vocabulary words). State of the art OOV detectors [3] are based on a MaxEnt paradigm taking various input features corresponding to confidence scores produced by an ASR decoder (such as word and sub-word confusion networks) in addition to prosodic [4] or syntactic features. The problem is cast as binary classification where each word or confusion-network bin produced by the ASR module is classified as OOV or non-OOV.

More recently [5] proposed to consider OOV detection as a sequence labeling problem since OOV words tend to generate multiple ASR errors. A CRF-based tagger was used to find the best sequence of *begin OOV* `B_OOV` and *inside OOV* `I_OOV` tags. This approach lead to significant improvement over the single word classification approach.

We propose in this study to generalize this method to other ASR errors than OOV words. We keep the same approach based on a CRF tagger using various ASR, lexical and syntactic features in order to find ASR error segments. *Significant* error segments detected are sent to the clarification dialog module for recovery.

3. TASK

The task used in this study is the English-Iraqi Arabic speech-to-speech translation task presented in [6]. We will consider here the English ASR side only. Our goal is to detect error segments in the ASR output, before sending the transcription to the English-Iraqi translation module. The ASR system used is the SRI *Dynaspeak* system [7] adapted to the task.

We used two corpora to develop and validate our method:

- *Corpus 1*: this corpus contains English utterances recorded by NIST for evaluating English-Iraqi Arabic speech-to-speech systems with simulated dialogs in the military domain during the TRANSTAC project. The corpus is close to the training corpus of the IraqComm system, therefore the Word Error Rate on the automatic transcriptions is relatively low ($< 10\%$).

- *Corpus 2*: this corpus has been recorded within the BOLT project specifically for testing the ability of S2S systems to recover from ASR errors and ambiguities. It contains sentences in the same military domain as *Corpus 1*, however each sentence was designed to contain one problem that can be either an OOV word, a mispronounced in-vocabulary word or a translation ambiguity. Of course the WER on this corpus is much higher (35%).

These two corpora correspond to two very different situations: in *Corpus 1* we are dealing with transcriptions with a relatively high quality, containing *regular* ASR errors ; *Corpus 2* contains utterances very relevant to our task, since they are all likely to contain at least one *significant* ASR error segment, however they can be considered as artificial as they were explicitly designed for this purpose. It is therefore interesting to evaluate our methods on both to verify that we obtain good results on the *target* corpus without impacting the performance on the *regular* corpus.

Corpus	#words	#utt.	WER	avg. err. size	err. fertil.
Corpus 1	84405	6527	8.4	1.5	1.2
Corpus 2	4919	570	35.8	2.6	4.8

Table 1. Corpora description with size, WER, average error segment size and ASR error fertility

Both corpora have been processed by the Dynaspeak ASR system. The automatic transcriptions have been aligned with the reference transcriptions thanks to the *sclite* tool. From this alignment we can compute 3 figures: the WER, the average error segment size in the automatic transcriptions, the *fertility* of an ASR error, representing how many erroneous words in the ASR hypothesis are generated, on average, by 1 misrecognized word in the reference transcription (without considering the deletion). Table 1 shows these figures for both corpora. As expected *Corpus 2* contains much longer error segment than *Corpus 1*. The ASR fertility is very high since every non-deletion error in the reference transcription generates on average a segment of almost 5 erroneous words. This can be explained by the OOV names voluntarily added in *Corpus 2*.

4. ERROR SEGMENT LOCALIZATION

The Error segment localization method presented in this paper is based on a CRF tagger which is in charge of labeling each word of an ASR hypothesis thanks to a binary label: e for error and c for correct. This tagger is trained on a corpus of aligned automatic/reference transcriptions as presented in the previous section. Three levels of features attached to each word are used to train the CRF:

1. ASR features: we use as features the posterior probabilities provided by Dynaspeak during the ASR decoding. These values are discretized thanks to the method described in [8] and available at [9]. The posteriors of the current, previous and following word are used.

2. Lexical features: the current, previous and following words are used as features, as well as the length of the word and 3 binary features indicating if the 3 different 3-grams including the current word have been seen in the training corpus of the ASR language model.

3. Syntactic features: the transcriptions are processed by the MACAON NLP tool chain [10] that includes a POS tagger and a dependency parser. POS tags, dependency labels and word governors in the dependency tree are added as features to the CRF tagger.

The error segment CRF tagger is trained on the concatenation of *Corpus 1&2* with a 10-fold setting. We use the CRFSuite tool to train tagging models. At decoding time we use the CRF decoder implemented in MACAON that outputs a lattice of word/tags hypotheses. This error detection lattice is used in the recovery strategy presented in section 5. The results given in the next subsection are obtained with the 1-best of these lattices.

There are different ways for evaluating the performance of an error segment tagger. The simplest method consists in evaluating the error/correct tag prediction at the word level. This is the *correct* metric presented in table 2. However, since this is a detection task with a target on erroneous words, it is more relevant to give standard detection metrics such as precision (P), recall (R), False Alarm (FA) and Miss (Miss) detection. Table 2 presents all these results separately for both corpora, although at training time they were merged (and used in 10-fold train/test validation). Three settings are compared, corresponding to different kinds of features used in the CRF: ASR posteriors alone; + lexical features; + syntactic features.

As we can see, error prediction performance is better on *Corpus 2* which contains a lot of *significant* error segments.

Corpus 1					
features	*correct*	**P**	**R**	**FA**	**Miss**
ASR post.	94.8	51.1	22.0	48.9	78.0
+ lexical	95.1	55.1	32.1	44.9	67.9
+ syntactic	95.1	55.4	36.8	44.6	63.2
Corpus 2					
features	*correct*	**P**	**R**	**FA**	**Miss**
ASR post.	78.7	76.6	41.5	23.5	58.5
+ lexical	81.3	74.0	57.8	26.0	42.2
+ syntactic	82.4	74.0	63.7	26.0	36.3

Table 2. Error segment detection results on both corpora according to the set of features used

Adding lexical and syntactic features improves the recall measure for both corpora.

All the measures given so far estimate performance at the word level. However in this study we are interested in error *segment* localization. More precisely we want to check the ability of the system to detect segments of ASR errors generated for each error in the reference transcription (measured as ASR fertility in table 1). For this purpose we propose to simply collapse every contiguous sequence of errors in the ASR transcriptions into a single token *XX* (effectively reducing the number of insertions), then compare this new ASR transcription to the reference one using the standard WER measure. This new way of measuring error segment detection is interesting as it has a direct impact on transcription quality, even without any recovery strategy. For example if the reference text of an utterance is "*I saw that man at Izamni*" and the ASR hypothesis is "*I saw that man at **is on me***" we have a WER of 50%. If we correctly detect the erroneous segment, by collapsing "***is on me***" into a single token "*XX*", we obtain a WER of 16.6%.

Corpus/WER	**ASR**	**Oracle**	**P(e)=0.5**	**P(e)=0.8**
Corpus 1	8.4	7.5	9.2	8.3
Corpus 2	35.8	19.9	31.1	29.6

Table 3. WER scores obtained by collapsing all error segments into a single token before reference/hypothesis alignment

As we can see in table 3, the Oracle WER is much lower than the ASR 1-best WER. The automatic error segment detection results are provided with two operating points: one at $P(e) = 0.5$ meaning that we consider all error segments with a probability of 0.5 or higher according to the CRF tagger; and one at $P(e) = 0.8$. We can see that increasing the precision (by rising the $P(e)$ threshold) decreases the WER, mostly for Corpus 2, but we are still far from the Oracle value.

5. INTERACTIVE RECOVERY STRATEGY

The detection of ASR errors allows for automatic correction strategies (for example by acquiring out-of-vocabulary words from their most likely phoneme sequence), and for interactive recovery strategies. In this section, we detail a recovery strategy based on clarification dialogs for a hands-free, eyes-free speech-to-speech translation application.

While there is a large body of work on multimodal error correction [11, 12, 13], speech-only strategies developed for dictation systems mostly rely on the fact that the user can see the current transcript, editable with a set of speech commands such as *select, correct, spell that* [14]. Face-to-face speech translation requires an eyes-free setup so that speakers can keep eye contact.

Such setup is seen in dialog systems that make use of task driven error correction by asking implicit or explicit confirmation of information or commands [15]. However, most deployed dialog systems rely on domain specificity to constrain the concepts which can be corrected, an hypothesis that must be removed given the wide range of domains that might be addressed in speech-to-speech translation.

In our work, clarification dialogs are limited to three utterances in order not to hamper the fluidity of the conversation. This leads to a straight forward dialog strategy where only one error segment can be addressed, even though it can be larger than the actual error (the limit case is to ask for a complete rephrase of the sentence). Therefore, we focus on the most *significant* error segment generated by the detector.

In order to enforce this dialog management constraint, we search for the most likely hypothesis with a single error segment in the error lattice generated by the CRF. Let E be the transducer output by the CRF model, where each transition bears an ASR word as input and an error class in $\{c, e\}$ as output. Let F be an acceptor that recognizes the language $c^*e^*c^*$, which represent paths that only contain one error segment. The composition $E \circ F$ yields all error detections containing at most one error segment, and therefore the path of minimum log likelihood in this transducer is the best hypothesis from the CRF which respects dialog constraints.

Depending on the type of error segment detected, the confidence of the detector and the dialog context, the dialog manager might ask one of the following question: (1) confirmation of the transcript, (2) spelling for the error segment if it's an OOV name, (3) rephrase of the complete utterance, (4) rephrase of part of the utterance. While the implementation of (1) to (3) is straightforward, we detail how the original utterance is edited with the answer to partial rephrase queries to address (4). There are two solicitations implemented in the dialog manager: "please rephrase" followed by the recorded speech of the error segment, and a few context words before the error segment followed by "what?" (*to give the what?*).

Error recovery from rephrased speech is an application of text-to-text generation and draws inspiration from sentence

fusion, the merging of two or more sentences to obtain a shorter version in summarizers. Sentence fusion has been cast as a parse tree fusion problem with rules [16, 17] and language model rescoring, or machine learning from raw and edited material [18]. While sentence fusion was studied with well formed sentences as input, interactive error recovery involves ASR errors and partial sentences.

When answering partial rephrase questions, the user might adopt one or more of these behaviors:

1. The answer exactly fits the error segment.
2. The answer contains additional words which contextualize the editing operation
3. The edit might not fit the syntactic context of the original (*"the* `<error>` *plates"* $\Rightarrow$ *"plates without scratches"*)
4. Some original words might be rephrased for conciseness (i.e. use a pronoun in place of a noun phrase)
5. The user can use convenience phrases to introduce his answer (*"I said that . . . "*)

In addition, there can be ASR errors in the answer transcript [19] and the error segment may have false boundaries.

We adopt a finite state transducer approach to perform an alignment of the answer with the original words, which directly results in an edited utterance. See Figure 1 for an illustration of the process. Let O and A be acceptors that respectively represent the original and answer utterances. Let `<error>` and `<ins>` be special symbols that match respectively the error segment or an insertion. First, paths that recognize `<error>` loops are added to O at the error segment location. In addition, all word arcs of A are doubled with `<error>/word` arcs so that when composing O with A, words are either matched or recognized as part of the error segment. Then, all word arcs from O are doubled with `word/<ins>` transitions and `<ins>` loops are concatenated before and after A. That way, unmatched words from the original utterance can be inserted on both sides of the answer. The corrected utterance is the shortest path of $O \circ A$. In order to address rephrasing of already correct words, this framework is enriched with paraphrase paths in the original [20], alternate error segment boundaries, and matching costs for `<error>` and `<ins>` symbols.

This recovery strategy is evaluated on a subset of *Corpus 2* which was designed to exercise the error detection and recovery components. The following sources of errors are tackled: out-of-vocabulary words (nouns, adjectives and verbs), homophones word sequences, mispronunciations, incomplete utterances, and basic ASR errors. System output is recorded in interactive sessions where the user is given a starting sentence which contains an intended error, such as an OOV. The system has to detect ASR errors in the recorded utterance and

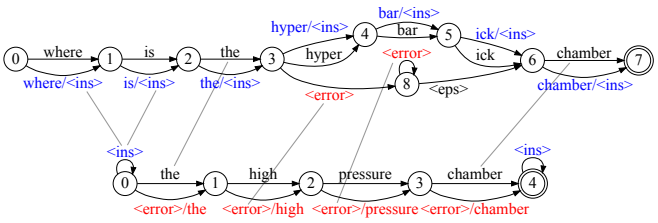

Fig. 1. Example of transducers used for merging. The error segment covers the misrecognized word "hyperbaric" and the resulting edited utterance is "where is the high pressure chamber".

User-intent	We don't want to create a **furor** over bad treatment.
Asr-output	we don't want to create a few or over bad treatment
Error-detect	few or
Sys-question	Can you rephrase AUDIO(few or)?
User-answer	Stink.
Edited	we don't want to create a stink over bad treatment

Fig. 2. Example of evaluation trial for OOV "furor."

ask for a rephrase of the largest error region. Then, the user utters an edition utterance which is supposed to fix the mistake. As he is free of using his own words, there is no guarantee that the new transcript will be free of ASR errors. An example of trial is given in Figure 2. Out of 100 trials performed in the BOLT project evaluation, the 59 trials that triggered error correction are used for evaluating the error recovery system.

To decouple the evaluation of error detection and recovery, we create two references: an error-segment reference given the ASR output, and an intended edited transcript given both what the user was supposed to say and how he rephrased the original utterance. For error detection, the complete segment accuracy is 57%. For error recovery, we compute a word-error-rate 27.57% of the new sentence compared to the intended edition, while a comparative baseline which always inserts the full answer in place of the error segment performs at a WER 29.36%. Out of 59 trials, 22% are fully recovered, a considerable result given that there may be error detection mistakes, ASR errors in the answer and merging errors.

6. CONCLUSION

In this study, we adapt error detection in ASR output to an interactive recovery strategy for a speech-to-speech translation application. The system makes use of a CRF error segment tagger based on acoustic, lexical and syntactic features, and the dialog system asks for a rephrase of the most *significant* error segment in order to edit the original sentence. Future work includes relaxing language model constraints in word lattices for getting more accurate error segment boundaries, using word lattices when editing the original in order to cope with ASR errors in rephrased error segments.

7. REFERENCES

[1] T.J. Hazen, S. Seneff, and J. Polifroni, "Recognition confidence scoring and its use in speech understanding systems," *Computer Speech & Language*, vol. 16, no. 1, pp. 49–67, 2002.

[2] R. Sarikaya, Y. Gao, M. Picheny, and H. Erdogan, "Semantic confidence measurement for spoken dialog systems," *Speech and Audio Processing, IEEE Transactions on*, vol. 13, no. 4, pp. 534–545, 2005.

[3] A. Rastrow, A. Sethy, and B. Ramabhadran, "A new method for oov detection using hybrid word/fragment system," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE, 2009, pp. 3953–3956.

[4] J. Hirschberg, D. Litman, and M. Swerts, "Prosodic and other cues to speech recognition failures," *Speech Communication*, vol. 43, no. 1, pp. 155–175, 2004.

[5] C. Parada, M. Dredze, D. Filimonov, and F. Jelinek, "Contextual information improves oov detection in speech," in *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2010.

[6] M. Akbacak, H. Franco, M. Frandsen, S. Hasan, H. Jameel, A. Kathol, S. Khadivi, X. Lei, A. Mandal, and S. Mansour, "Recent advances in sri's iraqcomm iraqi arabic-english speech-to-speech translation system," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE, 2009, pp. 4809–4812.

[7] H. Franco, J. Zheng, J. Butzberger, F. Cesari, M. Frandsen, J. Arnold, V.R.R. Gadde, A. Stolcke, and V. Abrash, "Dynaspeak: Sri's scalable speech recognizer for embedded and mobile systems," in *Proceedings of the second international conference on Human Language Technology Research*. Morgan Kaufmann Publishers Inc., 2002, pp. 25–30.

[8] U. Fayyad and K. Irani, "Multi-interval discretization of continuous-valued attributes for classification learning," 1993.

[9] "Discretized for CRF," `http://www.irisa.fr/texmex/people/raymond/Tools/tools.html`, 2012.

[10] A. Nasr, F. Béchet, J.F. Rey, B. Favre, and J. Le Roux, "Macaon: An nlp tool suite for processing word lattices," *Proceedings of the ACL 2011 System Demonstration*, pp. 86–91, 2011.

[11] D. Huggins-Daines and A.I. Rudnicky, "Interactive asr error correction for touchscreen devices," in *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Demo Session*. Association for Computational Linguistics, 2008, pp. 17–19.

[12] B. Suhm, B. Myers, and A. Waibel, "Multimodal error correction for speech user interfaces," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 8, no. 1, pp. 60–98, 2001.

[13] L. Hoste, B. Dumas, and B. Signer, "Speeg: a multimodal speech-and gesture-based text input solution," in *Proceedings of the International Working Conference on Advanced Visual Interfaces*. ACM, 2012, pp. 156–163.

[14] "Nuance Dragon Naturally Speaking," `http://nuance.com/dragon`, 2012.

[15] J. Shin, S. Narayanan, L. Gerber, A. Kazemzadeh, D. Byrd, et al., "Analysis of user behavior under error conditions in spoken dialogs," in *Proceedings of ICSLP*, 2002, vol. 2.

[16] R. Barzilay and K.R. McKeown, "Sentence fusion for multidocument news summarization," *Computational Linguistics*, vol. 31, no. 3, pp. 297–328, 2005.

[17] K. Filippova and M. Strube, "Sentence fusion via dependency graph compression," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2008, pp. 177–185.

[18] M. Elsner and D. Santhanam, "Learning to fuse disparate sentences," in *Proceedings of the Workshop on Monolingual Text-To-Text Generation*. Association for Computational Linguistics, 2011, pp. 54–63.

[19] C.M. Karat, C. Halverson, D. Horn, and J. Karat, "Patterns of entry and correction in large vocabulary continuous speech recognition systems," in *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*. ACM, 1999, pp. 568–575.

[20] M.G. Snover, N. Madnani, B. Dorr, and R. Schwartz, "Ter-plus: paraphrase, semantic, and alignment enhancements to translation edit rate," *Machine Translation*, vol. 23, no. 2, pp. 117–127, 2009.