RETRIEVING THE SYNTACTIC STRUCTURE OF ERRONEOUS ASR TRANSCRIPTIONS FOR OPEN-DOMAIN SPOKEN LANGUAGE UNDERSTANDING

Frédéric Béchet, Benoit Favre, Alexis Nasr, Mathieu Morey

LIF/CNRS, Aix-Marseille University, 163 avenue de Luminy, Marseille, France
{firstname.lastname}@lif.univ-mrs.fr

ABSTRACT

Retrieving the syntactic structure of erroneous ASR transcriptions can be of great interest for open-domain Spoken Language Understanding tasks in order to correct or at least reduce the impact of ASR errors on final applications. Most of the previous works on ASR and syntactic parsing have addressed this problem by using syntactic features during ASR to help reducing Word Error Rate (WER). The improvement obtained is often rather small, however the structure and the relations between words obtained through parsing can be of great interest for the SLU processes, even without a significant decrease of WER. That is why we adopt another point of view in this paper: considering that ASR transcriptions contain inevitably some errors, we show in this study that it is possible to improve the syntactic analysis of these erroneous transcriptions by performing a joint error detection / syntactic parsing process. The applicative framework used in this study is a speech-to-speech system developed through the DARPA BOLT project.

***Index Terms*—** Automatic Speech Recognition, Spoken Language Understanding, Dependency Parsing, Confidence Measures

1. INTRODUCTION

Open Domain Spoken Language Understanding aims at enriching spoken transcriptions with structural and semantic information without using a domain ontology limited to a specific application (e.g. flight booking). Such information includes finding sentence boundaries, parsing sentence and dialogue structure, spotting semantic concepts and predicate-argument entities, and distinguishing relevant from superfluous pieces of information.

The recent generalization of speech technology to a large range of applications such as voice search, speech-to-speech translation, personal assistant or multimedia interfaces have highlighted the need for domain-independent SLU models that can process speech input and output structured representation of messages. These structured representations can be seen as an interface between the word transcriptions (1-best, n-best, word lattice) output by ASR systems and application-specific input representations. In some cases the output of the open-domain SLU can be directly mapped to the application-domain representation; in other cases this first level of *understanding* is used to "*clean*" ASR output in order to correct or at least reduce the impact of ASR errors and spontaneous speech disfluencies on the final application.

The applicative framework used in this study is a speech-to-speech system developed through the DARPA BOLT project [1]. The originality of this system is to be able to perform clarification dialogs with a user when an error or an ambiguity is detected before sending the ASR transcription to the translation module. Retrieving the syntactic dependency structure of erroneous ASR transcriptions can be of great interest in such a use-case as clarification dialogs can take advantage of the structure predicted in order to ask more meaningful questions.

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

2. RELATED WORK

A lot of methods have been proposed for limited domain SLU, following early works on the ATIS corpus (see [2] for a review of SLU methods and models). Regardless of the paradigm chosen for performing SLU (parsing, classification, sequence labelling), the domain-ontology concepts and relations are always directly predicted from the ASR word transcriptions, sometimes with features coming from a linguistic analysis based on generic syntactic or semantic models. For open-domain SLU, it is necessary to choose an abstract level of representation that can be applied to a large range of domains and applications, therefore syntactic and semantic models developed in the Natural Language Processing community for processing text input are good candidates.

Despite its usefulness, syntactic parsing is not always considered when building a Spoken Language Understanding (SLU) system dedicated to process spontaneous speech because of two main issues:

1. firstly spontaneous speech transcriptions are often difficult to parse using a grammar developed for written text due to the specificities of spontaneous speech syntax (agrammaticality, disfluences such as repairs, false starts or repetitions);
2. secondly, transcriptions obtained through an Automatic Speech Recognition (ASR) process contain errors, the amount of errors increasing with the level of spontaneity in speech.

The first issue can be partially tackled by using new approaches to parsing. Syntactic parsing aims to uncover the word relationships (e.g. word order, constituents) within a sentence and support the semantic layer of the language-processing pipeline. Parsing is traditionally tightly connected to rewriting grammars, usually context free grammars, used together with a disambiguation model. Many current state-of-the-art text parsers are built on this model, such as [3]. Shallow syntactic processes, including part-of-speech and syntactic chunk tagging, are usually performed in the first stage. This traditional view of parsing based on context-free grammars is not suitable for processing non-canonical text such as automatic speech transcripts: due to ungrammatical structures in this kind of text, writing a generative grammar and annotating transcripts with that grammar remains difficult.

New approaches to parsing based on dependency structures and discriminative machine learning techniques [4] are much easier to adapt to non-canonical text for two main reasons: they need less training data; the annotation with syntactic dependencies of spoken transcripts is simpler than with syntactic constituents. Other advantages are the fact that partial annotation can be performed [5] and the parses generated are much closer to meaning than constituent trees, which eases semantic interpretation.

For the second issue of ASR errors and syntactic parsing, most of the work have addressed this problem from a different point of view: using syntactic features during ASR to help reducing Word Error Rate (WER). This can be done by directly integrating parsing and ASR language models [6] or keeping them as separate processes through a reranking approach using both ASR and parsing features [7, 8]. The improvement in ASR transcriptions obtained by adding syntactic features to the models is often rather small, however the structure and the relations between words obtained through parsing can be of great interest for the SLU processes, even without a significant decrease of WER. That is why we are adopting another point of view in this paper: considering that ASR transcriptions contain inevitably some errors, we show in this study that it is possible to improve the syntactic analysis of these erroneous transcriptions by performing a joint error detection / syntactic parsing process.

3. DEPENDENCY PARSING OF ERRONEOUS ASR TRANSCRIPTIONS

3.1. Characterizing ASR errors

ASR errors can have multiple sources, such as: language model ambiguities (*I ran* / ***Iran***); out-of-vocabulary words (*priest in* / ***pristine***); non-canonical word pronunciation; noise; voice quality and sub-optimal search due to real-time constraints.

We have presented in [9] a study on ASR error segment detection integrating various ASR and linguistic features. We proposed a model focusing on detecting *significant* error segments. An error is considered as *significant* if a clarification dialog is needed in order to correct it. For example out-of-vocabulary words or errors on names belong to this category, unlike confusion between a verb and its past participle (*e.g. call/called*). This system was integrated into the *SRI ThunderBOLT* system participating to the 2013 DARPA BOLT challenge [1]. In this new study we focus now on characterizing each error segment, from a syntactic point of view, in order to ask more meaningful questions during the clarification dialog.

One of the main difficulties is the fact that we are in an open domain framework (no precise modeling of expected semantics) and that we are targeting certain types of errors (errors that can be repaired with a clarification strategy).

This is illustrated in table 1 on two examples from the BOLT corpus. In example 1, there is only one error segment corresponding to an OOV word: ”`acetaminophen`” replaced by ”`I see them in a fan`”. The main difficulty here comes from the fact that only the first and the last word of the error segment have low ASR posteriors.

Example 2 highlights the differences between error segments for the clarification strategy: the ASR 1-best contains several errors, some minor ones like ”`travel/traveled`” or ”`cross/across`”, and two main errors ”`to desert`” instead of ”`the desert`” and ”`camera back`” instead of ”`camelback`”. In the context of a speech-to-speech translation application, such as the BOLT one, these two errors can bring a serious misunderstanding problem and possibly lead the dialog into a dead-end.

There is no information in ASR posteriors about the importance of errors. That is why we propose in this study to characterize each error segment hypothesis according to its syntactic role within the whole sentence. By retrieving the underlying structure of the original utterance, even if all the words are not recovered, we can help selecting the most important error segments for further processing, such as clarification dialogs as in the BOLT project. Five features are estimated for each error segment detected:

- 4 syntactic features: Part-Of-Speech (POS), dependency tag, dependency link, syntactic category of the chunk containing the error segment (either noun phrase or verbal phrase);
- 1 Named Entity tag (either `NONE`, `HUMAN` or `LOCATION`);

The process estimating these features is presented in the next subsection.

3.2. Parsing with error segments

The tagger and syntactic parser we use in this study come from the MACAON tool suite [10]. The POS-tagger is based on a linear-chain CRF as implemented in the CRFsuite library [11]. The syntactic parser is a first-order graph-based dependency parser trained using the discriminative perceptron learning algorithm with parameter averaging [4]. It uses the same first-order features as [12]. Compared to transition-based parsers, graph-based parsers are particularly interesting for ASR transcriptions because they have a more even distribution of errors and are less prone to error propagation [13]. This can be explained by the fact that transition-based parsers typically use a greedy inference algorithm with rich features, whereas graph-based parsers typically use exhaustive search algorithms with limited-scope features.

The POS-tagger and parser are trained with the following process:

- the tagging and parsing models are first trained on a ”*speechified*” versions of the CoNLL 2009 dataset for English [14] containing about 1 million words of newswire, where punctuation and capitalization have been removed and numbers replaced;
- these models are then used to annotate the ASR training corpus of the BOLT project;
- this annotated corpus is used to retrain the tagging and parsing models after replacing singleton words (words occurring only once in the corpus) by a special symbol **XX**. This symbol will be used at decoding time to represent each error segment automatically detected.

At decoding time, following [9], the ASR transcriptions are preprocessed to automatically detect ASR error segments. Then all words in an error segment hypothesis are collapsed and replaced by the single special token **XX**. The resulting transcriptions are annotated with part-of-speech tags and syntactic dependency trees using MACAON. The POS-tagger and parser handle **XX** tokens as regular unknown words: only contextual features, e.g. lemmas and POS of words preceding and following the unknown word, are used.

An example of the output of this process is given in table 2. From the automatic annotation obtained, we estimate the four syntactic features of the error segment ”`your dishes that`”: *POS=noun*, *dependency tag=object*, *dependency link=2* (*need*), *chunk=noun phrase*.

The Named Entity tag is given by an NE tagger applied on the transcription with the **XX** symbol.

example 1													
ASR 1-best	is	the	town	low	on	supplies	of	**I**	**see**	**them**	**in**	**a**	**fan**
ASR posteriors	1	1	1	0.9	1	1	1	**0.1**	0.9	0.9	0.9	0.8	**0.6**
Reference	is	the	town	low	on	supplies	of	***acetaminophen***					

example 2									
ASR 1-best	they	**travel**	**to**	**cross**	**to**	desert	on	**camera**	**back**
ASR posteriors	0.7	0.8	0.8	0.8	0.8	0.9	0.9	0.7	0.8
Reference	they	***traveled***		***across***	***the***	desert	on	***camelback***	

Table 1. Example of erroneous ASR transcription with OOV words

4. JOINT ERROR DETECTION AND CHARACTERIZATION

This section describes the decision module used to select the *most important* error segments in an ASR transcription. As presented in the previous section, not all ASR errors have the same impact on the processes following ASR. Therefore our goal is to bias our error segment detector toward errors that would have the biggest impact on the syntactic structure of ASR transcriptions.

To this purpose we introduce new features in the error segment scoring function, representing the *linguistic consistency* of a transcription (with possibly an error segment **XX**). These features are quite similar to those used in [7, 8]. We consider POS n-grams and dependency chains. On the BOLT training corpus processed by the MACAON tagger and parser as described in the previous section, we extract 3 kinds of features: all sequences of 3-grams and 4-grams POS tags, dependency chain up to 2 links with POS, dependency chain up to 2 links with words. For example, on the sentence of table 2, we have the following dependency chains: *(we-SBJ-need)*, *(judicious-NMOD-men)*, *(judicious-NMOD-men-OBJ-need)*.

The *linguistic consistency* of an ASR transcription W is estimated at the word level: for a word w_i we enumerate the POS 3,4-grams containing w_i in W as well as the dependency chains involving w_i in W. By checking the occurrence of these features in the set of similar features extracted from the entire BOLT training corpus, we obtain binary features that can be easily integrated in the error segment scoring function. The linguistic consistency of w_i for the 3 levels of syntactic features is expressed as follows:

1. $C_1(w_i, W) = \frac{\sum_{j=i-3}^{i} T(w_j \dots w_{j+3}) + \sum_{j=i-2}^{i} T(w_j \dots w_{j+2})}{7}$
2. $C_2(w_i, W) = \frac{T(D^1_{pos}(w_i), W) + T(D^2_{pos}(w_i, W))}{|D^1_{pos}(w_i, W)| + |D^2_{pos}(w_i, W)|}$
3. $C_3(w_i, W) = \frac{T(D^1_{word}(w_i, W)) + T(D^2_{word}(w_i, W))}{|D^1_{word}(w_i, W)| + |D^2_{word}(w_i, W)|}$

With $T(x)$ a binary function returning 1 if the event x occurs in the training corpus and 0 otherwise; $D^n(w, W)$ is the list of dependency chains of length n including word w in sentence W, the chain is given at the POS level for $D^n_{pos}(w, W)$ and at the word level for $D^n_{word}(w, W)$.

At decoding time the joint error detection and characterization process consists of the following steps:

- The ASR 1-best with posterior confidence scores is processed by the error segment detector presented in [9]. The output of this process is an n-best of error segments with confidence scores.
- From this n-best we produce an n-best of ASR transcriptions, each one containing an error segment collapsed and replaced by the symbol **XX**.
- Each transcription is processed by the MACAON tagger and parser and the linguistic consistency features are estimated for each word.
- Finally a reranker using both error segment confidence scores and linguistic consistency scores is used to choose the best hypothesis in the transcription n-best.

The fact that we process only one error segment for each hypothesis is justified by our applicative framework: the clarification dialog can only address one portion of the ASR 1-best at a time. However a single error segment can contain several words.

5. EXPERIMENTS

5.1. Experimental setup

These experiments have been made within the DARPA BOLT project. We consider here the English-Iraqi Arabic speech-to-speech translation task presented in [15]. Only the English ASR side is considered in this paper. The ASR system used is the SRI *Dynaspeak* system [16] adapted to the task.

The error detector used is the one described in [9]. The main novelty at the error detection level is the use of a new method developed at SRI for estimating ASR word confidence. The SRI word confidence is trained using a sigmoidal neural network with two binary outputs. Error labels are obtained by aligning a manual reference with an ASR hypothesis. Typical input features include maximum/mean/standard deviation of word posteriors from a confusion network. To capture long-distance word context within and across previous utterances, forward and backward recurrent neural network language model features are derived. Other features coming from a complimentary GMM ASR system are also combined to the NN features. With 36 input features, a neural network is trained using backpropagation. Details can be found in [17].

We used 3 corpora in this study:

- train: the BOLT language model training corpus, containing English utterances collected during the TRANSTAC project, is used to train ASR and collect the linguistic consistency features described in the previous section. This corpus contains 766K utterances and 7.6M words.
- development: we use a set of 745 utterances (7.4K words) recorded at SRI during the BOLT project for training the error segment tagger and tuning the decision process. This corpus contains sentences in the same domains as the training one, however most of the sentences were designed to contain at least one issue that can be either an OOV word, a mispronounced in-vocabulary word or a translation ambiguity, leading to a high WER (30%).
- test: the test corpus is similar to the development one, it is made of 887 utterance (6.4K words) with also a high WER

			Reference		Error rate
word	we	need	judicious	men	0%
POS	PRP	VBP	JJ	NN	0%
synt. dep.	SBJ(need)	ROOT()	NMOD(men)	OBJ(need)	0%

			1-best (no error processing)			Error rate
word	we	need	**your**	**dishes**	**that**	75%
POS	PRP	VBP	**PRP**	NN	**WDT**	50%
synt. dep.	SBJ(need)	ROOT()	**NMOD(dishes)**	OBJ(need)	**NONE()**	50%

			1-best with error processing	Error rate
word	we	need	**XX**	50%
POS	PRP	VBP	NN	25%
synt. dep.	SBJ(need)	ROOT	OBJ(need)	25%

Table 2. Example of word/POS/dep error rate evaluation with and without error processing

(27.9%) due to the specific ASR and translation issues added to the sentences.

5.2. Evaluation metrics

The main question addressed by this study is the capacity of our system to recover the syntactic structure of erroneous ASR transcriptions. Therefore we designed an evaluation framework illustrated in table 2. Each sentence from our test corpus is represented by 3 levels of annotation: the word sequence, the POS sequence and the sequence of syntactic dependencies of each word. The reference annotations for the word level are obtained manually. They are automatic for the two other levels: we run the MACAON parser on these manual transcriptions and consider the sequences of POS and dependencies obtained as the reference annotations to retrieve. Of course this is an approximation as the parser makes errors even on true transcriptions, however we consider this bias acceptable considering the cost of manually annotating data with syntactic information.

As we can see in table 2, correctly detecting an error segment, even without any reparation, can help reducing the Word Error Rate by reducing the insertion rate. Correctly predicting the POS and the dependency for the error segment reduces also the error rate at the POS and dependency level.

5.3. Results

Table 3 presents the results obtained with this metric on our test corpus. We compare 4 conditions:

1. **1-best** is the baseline, no error processing is used, and the syntactic parser is directly applied to the ASR 1-best.
2. **oracle** corresponds to a "*cheating*" experiments where the real error segments, obtained through an alignment between the reference and the 1-best transcriptions, are replaced by the symbol **XX**. This is the upper bound that our models can achieve.
3. **error conf.** is a version of our system where the decision is taken only from the confidence given by the error segment detector;
4. **joint** corresponds to the decision module presented in the previous section where the linguistic consistency features are added to the error confidence features in order to select the *best* error segment.

As we can see in table 3, the error processing module reduces significantly all error rates, even the WER by reducing the insertion rate. The POS and dependency error rates are reduced by around 4% in absolute and the joint processing add another little improvement.

condition/error rate	Word	POS	Synt. Dep.
1-best	27.9	21.4	44.6
oracle	21.2	14.5	34.2
error conf.	25.2	17.5	40.6
joint	25.0	16.9	39.9

Table 3. Word/POS/Dependency error rate without (1-best) and with syntactic recovery (oracle, posteriors and joint)

If these metrics are a good way of evaluating globally the performance of our models at the sentence level, independently from any applicative framework, they have the problem of mixing together correct and missed detection of erroneous words. For example, if we have an error segment made of 1 word confusion, and if the tagger predicts the correct POS for this confused word, we will get no reward for correctly replacing the erroneous word by the **XX** symbol.

Therefore another way to evaluate the performance of our system is to check the reliability of the features predicted for each error segment as presented in section 3. Table 4 presents such results on our test corpus with the two decision strategies: error confidence alone or joint decision with linguistic consistency. As we can see, in this case, the joint decision increases the recall of the correct detection and prediction, with only a very small loss in precision.

error confidence			
level	Precision	Recall	F-measure
POS	80.4	46.1	56.8
dependency	68.4	39.2	49.3
Named Entity	78.0	32.2	45.5
joint			
level	Precision	Recall	F-measure
POS	79.0	53.2	63.6
dependency	68.7	46.3	55.3
Named Entity	77.9	37.6	50.7

Table 4. Correct detection and prediction on error segments with and without the joint decision process for 3 levels of prediction: POS, dependency label+link and Named Entity

6. CONCLUSION

We have shown in this study that it is possible to improve the syntactic parsing of erroneous ASR transcription by jointly detecting and characterizing error segments. The results obtained, even if they are still far from the oracle ones, show a significant improvement over a baseline without error processing. This system has been integrated in the SRI ThunderBOLT system participating to the 2013 DARPA BOLT challenge.

7. REFERENCES

[1] Necip Fazil Ayan, Arindam Mandal, Michael Frandsen, Jing Zheng, Peter Blasco, Andreas Kathol, Frédéric Béchet, Benoit Favre, Alex Marin, Tom Kwiatkowski, et al., ""can you give me another word for hyperbaric?: Improving speech translation using targeted clarification questions," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 8391–8395.

[2] Gokhan Tur and Renato De Mori, *Spoken language understanding: Systems for extracting semantic information from speech*, John Wiley & Sons, 2011.

[3] Slav Petrov and Dan Klein, "Learning and inference for hierarchically split pcfgs," in *Proccedings of the National Conference on Artificial Intelligence*. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007, vol. 22, p. 1663.

[4] Ryan McDonald, Koby Crammer, and Fernando Pereira, "Online large-margin training of dependency parsers," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2005, pp. 91–98.

[5] Frédéric Béchet and Alexis Nasr, "Robust dependency parsing for spoken language understanding of spontaneous speech," in *Proc. Interspeech*. ISCA, 2009.

[6] Ciprian Chelba and Frederick Jelinek, "Structured language modeling," *Computer Speech & Language*, vol. 14, no. 4, pp. 283–332, 2000.

[7] Benjamin Lambert, Bhiksha Raj, and Rita Singh, "Discrimintively trained dependency language modeling for conversational speech recognition," in *Proc. Interspeech*. ISCA, 2013.

[8] Michael Collins, Brian Roark, and Murat Saraclar, "Discriminative syntactic language modeling for speech recognition," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2005, pp. 507–514.

[9] Frédéric Béchet and Benoit Favre, "Asr error segment localization for spoken recovery strategy," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 6837–6841.

[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] Naoaki Okazaki, "Crfsuite: a fast implementation of conditional random fields (crfs) `http://www.chokkan.org/software/crfsuite/`," 2007.

[12] Bernd Bohnet, "Very high accuracy and fast dependency parsing is not a contradiction," in *Proceedings of the 23rd International Conference on Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 89–97.

[13] Ryan T McDonald and Joakim Nivre, "Characterizing the errors of data-driven dependency parsing models.," in *EMNLP-CoNLL*, 2007, pp. 122–131.

[14] Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, et al., "The conll-2009 shared task: Syntactic and semantic dependencies in multiple languages," in *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*. Association for Computational Linguistics, 2009, pp. 1–18.

[15] 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.

[16] 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.

[17] Y. C. Tam, Y. Lei, J. Zheng, and W. Wang, "Asr error detection using recurrent neural network language model and complementary asr," in *submitted to ICASSP 2014*, 2014.