"CAN YOU GIVE ME ANOTHER WORD FOR *HYPERBARIC*?": IMPROVING SPEECH TRANSLATION USING TARGETED CLARIFICATION QUESTIONS

Necip Fazil Ayan[1], *Arindam Mandal*[1], *Michael Frandsen*[1], *Jing Zheng*[1], *Peter Blasco*[1], *Andreas Kathol*[1]
Frédéric Béchet[2], *Benoit Favre*[2], *Alex Marin*[3], *Tom Kwiatkowski*[3], *Mari Ostendorf*[3]
Luke Zettlemoyer[3], *Philipp Salletmayr*[5]*, *Julia Hirschberg*[4], *Svetlana Stoyanchev*[4]

[1] SRI International, Menlo Park, USA
[2] Aix-Marseille Université, Marseille, France
[3] University of Washington, Seattle, USA
[4] Columbia University, New York, USA
[5] Graz Institute of Technology, Austria

ABSTRACT

We present a novel approach for improving communication success between users of speech-to-speech translation systems by automatically detecting errors in the output of automatic speech recognition (ASR) and statistical machine translation (SMT) systems. Our approach initiates system-driven targeted clarification about errorful regions in user input and repairs them given user responses. Our system has been evaluated by unbiased subjects in live mode, and results show improved success of communication between users of the system.

Index Terms— Speech translation, error detection, error correction, spoken dialog systems.

1. INTRODUCTION

Interacting via natural language between a human and machine to accomplish a task, such as speech translation, poses two key technical challenges to human-machine communication systems:

1. The complexity, ambiguity, and informality of natural spoken language have a significant negative impact on the overall performance of existing dialog systems.
2. User intent is conveyed not only by what is explicitly said but also by how and in what context it is said.

State-of-the-art linguistic tools are most successful on formal, edited data (e.g., news stories) and tend to break down with spontaneous, unedited natural language input in a speech-to-speech human-machine communication system. ASR systems are constantly improving, but their performance degrades significantly in the presence of out-of-vocabulary (OOV) words and ambiguous words. In a speech-to-speech translation setting, such system errors result in loss of communication between users and in a poor user experience. Thus, it is critical to detect and fix these errors, preferably before translation, to avoid cascaded errors in dialog.

Most existing human-machine communication systems follow simple clarification mechanisms to address certain types of errors in recognition, usually in the form of generic "please rephrase" questions. While this strategy is useful to recover from some errors, it frequently fails because the users do not have a clear indication of which portion of their original utterance is causing the recognition system to fail, and they do not have clear instructions for rephrasing their original utterance. Therefore, the ability to detect misrecognized portions of user speech and to ask targeted clarification questions to address these misrecognitions is important for success in human-machine communication systems.

*Author performed the work while visiting Columbia University.

Our previous work on speech-to-speech translation systems has shown that there are seven primary sources of errors in translation:

- ASR named entity OOVs: *Hi, my name is Colonel* **Zigman**.
- ASR non-named entity OOVs: *I want some* **pristine** *plates.*
- Mispronunciations: *I want to collect some* **+de-MOG-raf-ees** *about your family?* (demographics)
- Homophones: *Do you have any* **patients** *to try this medication?* (patients vs. patience)
- MT OOVs: *Where is your* **father-in-law***?*
- Word sense ambiguity: *How many men are in your* **company***?* (organization vs. military unit)
- Idioms: *We are* **clear as a bell** *on the level of supplies.*

In this work, we present a new architecture for a speech-to-speech translation system with machine-initiated clarification functionality to resolve these sources of errors in ASR and MT output. The novelty of the proposed architecture is its ability to localize the errors and ambiguities in ASR and MT outputs, ask targeted questions to address only those regions of errors and ambiguities, and merge user responses with the original utterance before translation.

2. SYSTEM OVERVIEW

Figure 1 presents the architecture of our speech-to-speech translation system with machine-initiated clarification capability. More specifically, this work is focused on speech-to-speech translation between English and Iraqi-Arabic. The user speech is first recognized by an ASR system to produce a 1-best ASR hypothesis, a word confusion network, a lattice, and several prosodic features. Next, the error detection module's OOV detection component processes the word confusion networks generated by the ASR system, and identifies OOV candidates—both named and non-named entities. Then a confidence prediction component assigns a probability of misrecognition to each word in the 1-best ASR hypothesis. This probability score is generated by a statistical classifier trained on a combination of ASR posterior probabilities, prosodic features, and syntactic features associated with each word. The ASR error detection module makes a final decision about errorful segments. It employs a conditional random field (CRF) tagger, a dependency parser, and an error hypothesis re-ranker to process all the features and scores generated by the previous components, and generates a set of candidate error segments, which might span multiple words.

The answer extraction and merging module combines the initial user input with subsequent user answers to system-initiated clarification questions by extracting the relevant portions from user responses to compose the final answer that will be translated. If no

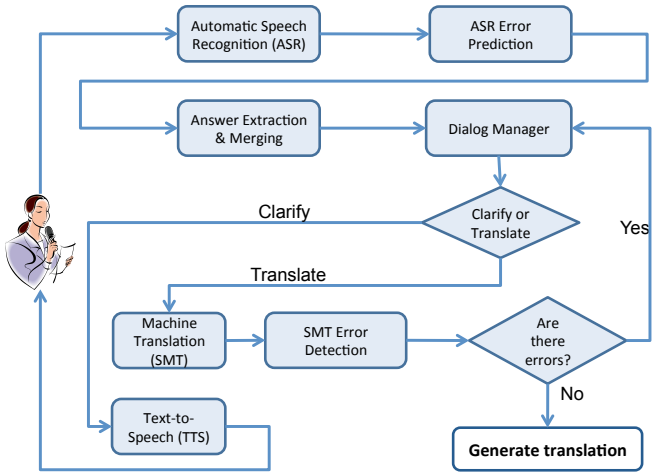

Fig. 1. Clarification dialog in a speech translation system: The user speech is passed through error detection modules. If the system finds an error in the input, it initiates a clarification dialog with the user. Once all errors are resolved, the system translates the best input sentence generated by the answer-merging module over several turns.

ASR errors are detected in the merged output, the dialog manager sends the merged output to the translation engine. The MT error detector analyzes the translation and checks for errors, such as words unknown to the translation engine. If no errors are found in the translation, the system sends this final translation to the other party in the conversation. If there are any ASR or MT errors, however, the dialog manager formulates a question according to the type of error detected, and plays back this question to the user through a text-to-speech synthesizer (TTS) to finish the current turn. Figure 2 illustrates how the system detects errors, initiates a dialog with the user, and generates the final translation.

3. RELATED WORK

Early work in OOV detection in ASR hypotheses for constrained-domain applications used OOV word classes in the language model [1, 2]. Better results were obtained using multiple sub-word fragments as words [3], especially in highly inflected languages and for spoken term detection. Another approach is based on word confidence estimation techniques, using ASR word posteriors with contextual features such as word lattice topology, and predicting OOVs instead of or in addition to ASR errors [4, 5]. ASR error detection is considered as a generalization of OOV detection where word-level errors predictors are trained on ASR hypotheses [6, 7]. It is achieved by learning from lexical, syntactic and prosodic features [8, 9, 10], as well as ASR confidence metrics such as word posteriors and depth of confusion networks [5] and by decoupling the most probable acoustic and linguistic hypotheses [3]. Error recovery strategies developed for multimodal scenarios [11, 12] allows editing of recognized words using speech commands such as *select, correct, spell that* [13]. Fusion of multiple sentences has been studied in the context of summarization, by aligning and merging dependency parse trees with rule-based and integer linear programming methods [14, 15]. Clarification based error correction in speech-to-speech systems is a relatively new approach and only one other work is reported in [16].

Starting Sentence	We need to coordinate the rescue efforts at Hawija Street
Speech Recognition	we need to coordinate the rescue efforts at all i just street
Translation	احنا نحتاج ننسق الانقاذ جهود بس اني شارع
Gloss	we (احنا) need to (نحتاج) coordinate (ننسق) the rescue (الانقاذ) efforts (جهود) only (بس) I (اني) street (شارع)
Clarification Question	I think I heard a name when you said PLAY_USER_SPEECH(Hawija). Is that right?
User Response	yes
Clarification Question	OK, can you please spell that name?
User Response	harry adam william ike jim adam
Translation Input	we need to coordinate the rescue efforts at **$name{Hawija→حويجه}** street
Translation	احنا نحتاج ننسق الانقاذ جهود بشارع حويجه
Gloss	we (احنا) need to (نحتاج) coordinate (ننسق) the rescue (الانقاذ) efforts (جهود) at (ب) **Hawija (حويجه)** street (بشارع)

Fig. 2. Sample Dialog between Human and Machine: The system identifies a named entity OOV error and initiates a clarification dialog. The name provided by the user using a standard spelling alphabet is transliterated by the system and merged with the original user input. After all the errors are resolved, the system translates the merged user input.

4. SYSTEM COMPONENTS

This section describes in detail each component outlined in Figure 1.

4.1. Automatic Speech Recognition

Our ASR system uses standard acoustic models that have cross-word triphones modeled as hidden Markov models with output distributions modeled by Gaussian mixture distributions trained discriminatively using the minimum phone error criterion. The front-end signal processing uses Mel-frequency cepstral coefficients features, which are transformed using several linear operations to adapt to speakers and environments. To speed up ASR decoding we used standard Gaussian selection shortlists. The ASR decoding graph is created using a unigram language model (LM) using highly optimized weighted finite state transducer composition and expanded using a modified on-the-fly LM rescoring approach with a 4-gram, Kneser-Ney smoothed LM. Detailed information about our ASR system can be found in [17].

4.2. Error Detection Module

The error detection module consists of three components: an OOV detector, an ASR confidence predictor and an ASR error detector. We describe each component next.

4.2.1. OOV Detection

The OOV detection component detects regions in the output of an ASR system that correspond to an OOV word in the utterance (see Figure 3 for an illustration). Since ASR systems operate with a fixed vocabulary, they inevitably run into problems when they encounter an OOV word and hypothesizes words or sequences of words that have similar phonetic content but that are often syntactically and/or semantically anomalous.

The OOV detection component takes as input a word confusion network (WCN), which is a flattened lattice, consisting of a sequence of slots with each slot comprising a list of words and their confidences. The output is a WCN annotated with OOVs. The OOV

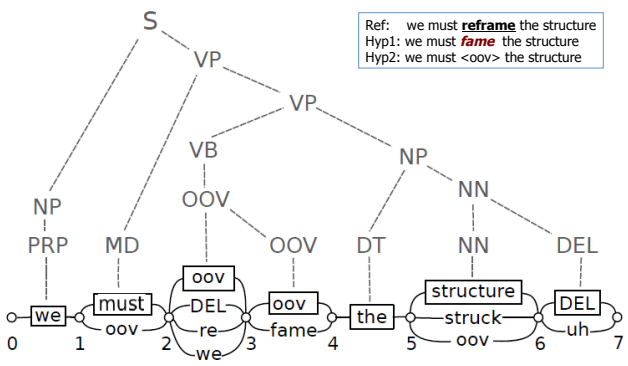

Fig. 3. OOV detection: Example best parse of a WCN for the utterance *"We must reframe the structure"*, where the word *"reframe"* is OOV. Note that *"reframe"* has been mapped, by the ASR system, onto two slots in the WCN. These are modeled by the parser as an OOV region with category VB (verb), allowing the same syntactic parse that would result if the word *"reframe"* had been recognized.

classification is performed in three stages: (1) Predict the slot-level posterior OOV probability using maximum entropy (MaxEnt) classifiers, and renormalize other word posteriors; (2) Choose the best path through the WCN allowing OOV spans, using a probabilistic context free grammar (PCFG) parser modified to incorporate deletions (DEL) and multiple OOV tokens; and (3) Refine OOV region prediction with a second MaxEnt classifier with syntactic/semantic features, leveraging the output of the previous stages.

A system with stage 1 alone corresponds to the standard approach used in previous work. For a task with a relatively high OOV rate, using just stage 1 reduces word error rate (WER) from 14.1% to 12.9%, and adding the subsequent stages further reduces the WER to 11.7%[1]. The additional stages improve OOV detection to 68.7 F-score over the baseline performance of 58.2 F-score. Analyzing performance over a range of detection thresholds, we find that the multi-stage system combines the strengths of the different approaches in different regions of the precision-recall curve. Further details of this component are described in [18].

4.2.2. ASR Confidence Prediction

We developed a set of features that capture information for predicting confidence in the correctness of each word in the ASR hypotheses [19]. These features included prosodic information (min, max, mean, and stddev of F0 and energy, proportion of voiced segments, duration, and measures of speaking rate) as well as lexical and syntactic information (part-of-speech tag (POS) ngrams, parse and OOV hypothesis confidence scores), together with confusion network-based posterior probability scores on the 1-best hypothesis produced by the decoder. Using classifiers trained on acoustic data collected between 2005 and 2009 under DARPA's TRANSTAC program and ASR hypotheses from SRI's IraqComm system [20], we predict if an an utterance and its constituent words has been misrecognized. Our method improved the F-measure for predicting misrecognized utterances by 13.3% when compared to using ASR posteriors alone, and prediction of misrecognized F-measure by 40%.

[1]The reduction in WER is due to replacing OOV words in the references of the held-out development set and the OOV predictions in the ASR hypotheses with a unique token in both sets.

4.2.3. ASR Error Detection

Lastly, the ASR error segment detection component hypothesizes error segments with associated confidence scores and sends them to the dialog manager to generate the next clarification question. The labels attached to each error segment contain the type of word expected to correct the erroneous segments (noun, verb, named-entity) and a dependency link to another word in the utterance. The first step in the process is to label each word of the ASR 1-best transcription as error/non-error, using a CRF tagger that uses a set of confidence measures as well as lexical and syntactic features considering not only OOV errors but also ASR insertions or substitutions error. The tagger produces a lattice of sequences of error/non-error labels. Since the clarification strategy can deal with only one error at a time, we filter the lattice by keeping paths that contain only one error segment and imposing a minimum length for a segment to be retained. A n-best list of error segment hypotheses is then produced by enumerating the n-best paths in the filtered lattice. Each error segment in each hypothesis in the n-best list is replaced by the symbol "*X*" in the ASR 1-best transcription.

The second step consists of applying a statistical dependency parser similar to the MATE [21] parser to each "*X*" hypothesis. The dependency parser is trained on a corpus that contained "*X*" symbols for very low-frequency words and hypothesizes the most likely POS category, dependency link and type according to the context of "*X*" in the sentence. The last step re-ranks the n-best lists of error segments using the estimated POS labels and dependency information from the parser and a classifier that is trained on a large set of confidence scores and linguistic features. The re-ranking process is biased toward precision rather than recall to limit the number of false alarms in a clarification dialog. For instance, we can increase the error detection precision by 20% (from 60 to 80%) while losing only 10% of recall compared to the use of word posterior confidence measures only. For each error segment detected, the correct POS tag and syntactic link are predicted in 42% and 60% of the cases, respectively.

4.3. Answer Extraction and Merging

The answer extraction and merging module is responsible for creating a corrected utterance given the initial user utterance and the answer to a clarification question. This task can get complicated, depending on the complexity of the user responses to clarification questions, with several possibilities: (1) answer exactly fits the error segment; (2) answer is anchored to original words; (3) some original words are rephrased in answer; (4) answer has filler phrases and (5) answer may have ASR errors.

We adopt a finite-state transducer approach to align the answer with the original words, which directly leads to the corrected utterance as shown in Figure 4. Let O and A be acceptors that respectively represent the original and answer utterances, and $<error>$ and $<ins>$ are special symbols that match respectively the error segment or an insertion. Using these special symbols, we augment O and A with paths that optionally map any word in ASR hypotheses to part of a multi-word spanning error segment or alternatively allow them to be matched before or after an error segment. The corrected utterance is the shortest path in the transducer produced by the composition of O and A ($O \circ A$). To address the above possibilities, this framework is enriched with paraphrase paths in the original [22], alternate error segment boundaries, and matching costs for $<error>$ and $<ins>$ symbols. ASR errors can be accounted for by using word lattices for O and A.

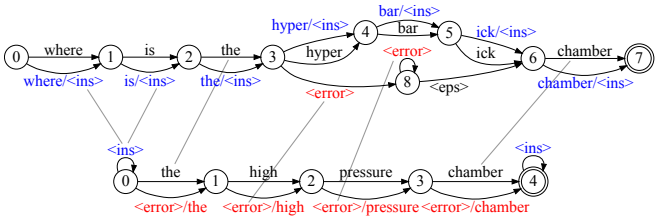

Fig. 4. Example of transducers used for merging. The error segment covers the misrecognized word "hyperbaric" and the resulting edited utterance is "where is the high pressure chamber."

4.4. Dialog Manager

Our dialog manager is rule-based, employing templates for different types of errors detected in the user input and/or translation. It accepts multiple templates for each error type, and automatically choose one during run-time. The templates might include references to user speech for playback purposes, and specification of prosodic information on any word. There are five primary types of clarification question templates: (1) Ask for a spelling of a word (named entity OOVs); (2) Ask for another word or phrase (non-named-entity OOVs and MT OOVs); (3) Provide alternatives and ask for disambiguation (homophones and word senses); (4) Substitute erroneous words/phrases with "*what*," "*who*," "*where*," "*which*," or "*what kind of*," depending on the POS tag and dependency information attached to erroneous words (mispronunciations, non-OOV errors); and (5) Confirm a name, a spelling, or an entire sentence. Figure 5 presents examples of clarification questions for different templates.

User: Hello, my name is Captain **Hill**.
System: I think I heard a name when you said [play-user-speech:*Hill*]. Could you please spell that name?
User: We recently got a large **philanthropic** donation.
System: Can you please tell me another word or phrase for this: [PLAY-USER-SPEECH:*philanthropic*]
User: We need to collect some **demographees** about your family.
System: collect some what?
User: We need some **patients** to use this medication.
System: Did you mean *patience* as in "tolerance of delay" or *patients* as in "people who require medical care?" Please say one or two.
User: We should not have any **unforeseen** difficulties.
System: I do not know how to translate the word "*unforeseen*". Can you give me another word for it?

Fig. 5. Sample questions generated by the dialog manager based on the types of detected errors: named-entity and non-named entity ASR OOV, mispronunciation, homophone, and MT OOV.

4.5. Statistical MT and Name Transliteration

Our machine translation systems from English to Iraqi-Arabic employ phrase-based models [23] for name transliteration, and hierarchical phrase-based models [24] for final translation.

The translation models are log-linear models that combine several features, including a language model score, conditional phrase probabilities in both directions, lexical phrase probabilities in both directions, word penalty, and phrase/rule penalty scores. For the final translation system, we generated a 6-gram language model using the SRILM toolkit [25], on the target-language side of the MT training data. For name transliteration system, we used a 10-gram character-based language model trained on nearly 10K names collected from the MT training data. The word alignments were generated using GIZA++ [26] on the entire training data in both directions, and merging those uni-directional alignments using heuristic-based methods [23]. During phrase extraction, we limited the maximum phrase length to ten words (or characters for name transliteration) on either side. Both MT systems were optimized using an in-house implementation of the minimum-error-rate trainer [27] to maximize BLEU score [28] on the tuning set. The training data consisted of 760K sentences (6.8M tokens) for translation and 6.6K names for transliteration. The translation and transliteration systems were optimized and tested on nearly 3500 sentences and 1700 names respectively.

5. RESULTS

Our English ASR and English-to-Iraqi-Arabic SMT systems were trained on data released under DARPA's TRANSTAC program. The ASR word error rate was in the range of 4.4-10.5% on four "online" test sets, and between 13.5-20.8% on three "offline" test sets collected under TRANSTAC. The SMT BLEU scores on tuning and test sets, with one reference translation, were 17%.

Clarification Success	Translation Success	Number of Sentences
Unsuccessful	Incorrect	33
Successful	Incorrect	11
	Partially correct	23
	Correct	54

Table 1. Number of sentences with unsuccessful and successful clarification and translation for the targeted error regions.

As part of evaluation of our system under DARPA's BOLT program, we measured the utility of clarification dialog in the English-to-Iraqi-Arabic direction only by analyzing performance on 121 English sentences collected by NIST, which "targeted" one error region drawn from the error categories listed in Section 1. Table 1 shows the distribution of number of sentences where the system clarified and translated the targeted error region successfully or failed to do so. On 33 out of 121 sentences the system did not detect an error or the clarification attempt failed, resulting in an incorrect translation for the error region. For the remaining 88 sentences, the system was successful at clarifying the error region and produced a transcription for that region with a meaning equivalent to the original utterance. Among those 88 sentences the system produced correct translation on 54, partially correct translation on 23, and incorrect translation on 11 sentences. Overall, our system was able to get the correct transcription in 73% (88/121) of the test sentences through a successful clarification attempt. In 88% (77/88) of the cases where clarification was successful, the system was able to generate the correct (or partially correct) translation.

Acknowledgments: This work is supported by the Defense Advanced Research Projects Agency (DARPA) under contract number HR0011-12-C-0016. Approved for Public Release, Distribution Unlimited.

6. REFERENCES

[1] Ayman Asadi, Rich Schwartz, and John Makhoul, "Automatic detection of new words in a large vocabulary continuous speech recognition system," in *Proceedings of ICASSP*, 1990, vol. 1, pp. 125–128.

[2] Thomas Schaaf, "Detection of OOV words using generalized word models and a semantic class language model," in *Proceedings of Eurospeech*, 2001, pp. 2581–2584.

[3] Ariya Rastrow, Abhinav Sethy, and Bhuvana Ramabhadran, "A new method for OOV detection using hybrid word/fragment system," in *Proceedings of ICASSP*, 2009, pp. 3953–3956.

[4] Steve Young, "Detecting misrecognitions and out-of-vocabulary words," in *Proceedings of ICASSP*, 1994, vol. ii, pp. II.21–II.24.

[5] Benjamin Lecouteux, Georges Linarès, and Benoit Favre, "Combined low level and high level features for out-of-vocabulary word detection.," in *Proceedings of Interspeech*, 2009, pp. 1187–1190.

[6] Timothy Hazen and Issam Bazzi, "A comparison and combination of methods for oov word detection and word confidence scoring," in *Proceedings of ICASSP*, 2001, vol. 1, pp. 397–400.

[7] Carolina Parada, Mark Dredze, Denis Filimonov, and Fred Jelinek, "Contextual information improves OOV detection in speech," in *Proceedings of HLT-NAACL*, 2010, pp. 216–224.

[8] Yongmei Shi, *An investigation of linguistic information for speech recognition error detection*, Ph.D. thesis, University of Maryland, Baltimore, 2008.

[9] Julia Hirschberg, Diane Litman, and Marc Swerts, "Prosodic and other cues to speech recognition failures," in *Speech Communication*, 2004, vol. 43, pp. 155–175.

[10] Elizabeth Shriberg and Andreas Stolcke, "Prosody modeling for automatic speech recognition and understanding," in *Proceedings of the Workshop on Mathematical Foundations of Natural Language Modeling*, 2002, pp. 105–114.

[11] David Huggins-Daines and Alexander 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*, 2008, pp. 17–19.

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

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

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

[15] Katja Filippova and Michael Strube, "Sentence fusion via dependency graph compression," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2008, pp. 177–185.

[16] Rohit Prasad, Rohit Kumar, Shankar Ananthakrishnan, Wei Chen, Sanjika Hewavitharana, Matthew Roy, Frederick Choi, Aaron Challenner, Enoch Kan, Arvind Neelakantan, and Prem Natarajan, "Active error detection and resolution for speech-to-speech translation," in *Proceedings of International Workshop on Spoken Language Translation (IWSLT)*, 2012.

[17] Jing Zheng, Arindam Mandal, Xin Lei, Michael Frandsen, Necip Fazil Ayan, Dimitra Vergyri, Wen Wang, Murat Akbacak, and Kristin Precoda, "Implementing SRI's Pashto speech-to-speech translation system on a smartphone," in *Proceedings of IEEE Workshop on Spoken Language Technology (SLT)*, 2010, pp. 121–126.

[18] Alex Marin, Tom Kwiatkowski, Mari Ostendorf, and Luke Zettlemoyer, "Using syntactic and confusion network structure for out-of-vocabulary word detection," in *Proceedings of IEEE Workshop on Spoken Language Technology (SLT)*, 2012.

[19] Svetlana Stoyanchev, Philipp Salletmayr, Jingbo Yang, and Julia Hirschberg, "Localized detection of speech recognition errors," in *Proceedings of IEEE Workshop on Spoken Language Technology (SLT)*, 2012.

[20] Murat Akbacak, Horacio Franco, Michael Frandsen, Saŝa Hasan, Huda Jameel, Andreas Kathol, Shahram Khadivi, Xin Lei, Arindam Mandal, Saab Mansour, Kristin Precoda, Colleen Richey, Dimitra Vergyri, Wen Wang, Mei Yang, and Jing Zheng, "Recent advances in SRI's IraqComm Iraqi Arabic-English speech-to-speech translation system," in *Proceedings of ICASSP*, 2009, pp. 4809–4813.

[21] Bernd Bohnet, "Top accuracy and fast dependency parsing is not a contradiction," in *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)*, 2010, pp. 89–97.

[22] Matthew Snover, Nitin Madnani, Bonnie J. Dorr, and Rich Schwartz, "TER-Plus: paraphrase, semantic, and alignment enhancements to translation edit rate," *Machine Translation*, vol. 23, no. 2, pp. 117–127, 2009.

[23] Philipp Koehn, Franz J. Och, and Daniel Marcu, "Statistical phrase-based translation," in *Proceedings of HLT-NAACL*, 2003.

[24] David Chiang, "Hierarchical phrase-based translation," *Computational Linguistics*, vol. 33, no. 2, pp. 201–228, 2007.

[25] Andreas Stolcke, "SRILM – an extensible language modeling toolkit," in *Proceedings of the International Conference on Spoken Language Processing (ICSLP'02)*, 2002, vol. 2, pp. 901–904.

[26] Franz J. Och and Hermann Ney, "Improved statistical alignment models," in *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL 2000)*, 2000, pp. 160–167.

[27] Franz J. Och, "Minimum error rate training in statistical machine translation," in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL 2003)*, 2003, pp. 160–167.

[28] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*, Philadelphia, PA, 2002, pp. 311–318.