Active Error Detection and Resolution for Speech-to-Speech Translation

Rohit Prasad, Rohit Kumar, Sankaranarayanan Ananthakrishnan, Wei Chen,
Sanjika Hewavitharana, Matthew Roy, Frederick Choi, Aaron Challenner,
Enoch Kan, Arvind Neelakantan, Prem Natarajan

Speech, Language, and Multimedia Business Unit, Raytheon BBN Technologies
Cambridge MA, USA
{rprasad,rkumar,sanantha,wchen,shewavit,mroy,fchoi,achallen,ekan,aneelaka,prem}@bbn.com

Abstract

We describe a novel two-way speech-to-speech (S2S) translation system that *actively* detects a wide variety of common error types and resolves them through *user-friendly* dialog with the user(s). We present algorithms for detecting out-of-vocabulary (OOV) named entities and terms, sense ambiguities, homophones, idioms, ill-formed input, etc. and discuss novel, interactive strategies for recovering from such errors. We also describe our approach for prioritizing different error types and an extensible architecture for implementing these decisions. We demonstrate the efficacy of our system by presenting analysis on live interactions in the English-to-Iraqi Arabic direction that are designed to invoke different error types for spoken language translation. Our analysis shows that the system can successfully resolve 47% of the errors, resulting in a dramatic improvement in the transfer of problematic concepts.

1. Introduction

Great strides have been made in Speech-to-Speech (S2S) translation systems that facilitate cross-lingual spoken communication [1][2][3]. While these systems [3][4][5] already fulfill an important role, their widespread adoption requires broad domain coverage and unrestricted dialog capability. To achieve this, S2S systems need to be transformed from *passive conduits* of information to *active participants* in cross-lingual dialogs by detecting key causes of communication failures and recovering from them in a user-friendly manner. Such an active participation by the system will not only maximize translation success, but also improve the user's perception of the system.

The bulk of research exploring S2S systems has focused on maximizing the performance of the constituent automatic speech recognition (ASR), machine translation (MT), and text-to-speech (TTS) components in order to improve the rate of success of cross-lingual information transfer. There have also been several attempts at joint optimization of ASR and MT, as well as MT and TTS [6][7][8]. Comparatively little effort has been invested in the exploration of approaches that attempt to detect errors made by these components, and the interactive resolution of these errors with the goal of improving translation / concept transfer accuracy.

Our previous work presented a novel methodology for assessing the severity of various types of errors in our English/Iraqi S2S system [9]. These error types can be broadly categorized into: (1) out-of-vocabulary concepts; (2) sense ambiguities due to homographs, and (3) ASR errors caused by mispronunciations, homophones, etc. Several approaches, including implicit confirmation of ASR output with barge-in and back-translation [10], have been explored for preventing such errors from causing communication failures or stalling the conversation. However, these approaches put the entire burden of error detection, localization, and recovery on the user. In fact, the user is required to infer the potential cause of the error and determine an alternate way to convey the same concept – clearly impractical for the broad population of users.

To address the critical limitation of S2S systems described above, we present novel techniques for: (1) automatically detecting potential error types, (2) localizing the error span(s) in spoken input, and (3) interactively resolving errors by engaging in a clarification dialog with the user. Our system is capable of detecting a variety of error types that impact S2S systems, including out-of-vocabulary (OOV) named entities and terms, word sense ambiguities, homophones, mispronunciations, incomplete input, and idioms.

Another contribution of this paper is the novel strategies for overcoming these errors. For example, we describe an innovative approach for cross-lingual transfer of OOV named entities (NE) by splicing corresponding audio segments from the input utterance into the translation output. For handling word sense ambiguities, we propose a novel constrained MT decoding technique that accounts for the user's intended sense based on the outcome of the clarification dialog.

A key consideration for making the system an active participant is deciding how much the system should talk, i.e. the number of clarification turns allowed to resolve potential errors. With that consideration, we present an effective strategy for prioritizing the different error types for resolution and also describe a flexible architecture for storing, prioritizing, and resolving these error types.

2. Error Types Impacting S2S Translation

We focus on seven types of errors that are known to impact S2S translation. Table 1 shows an example of each of these error types. Out-of-vocabulary names (*OOV-Name*) and Out-of-vocabulary non-name words (*OOV-Word*) are some of the errors introduced by the ASR in S2S systems. OOV words are recognized as phonetically similar words that do not convey the intended concept. *Word sense ambiguities* in the input language can cause errors in translation if a target word/phrase does not correspond to the user's intended sense.

Disclaimer: This paper is based upon work supported by the DARPA BOLT Program. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

Distribution Statement A (Approved for Public Release, Distribution Unlimited)

Homophone ambiguities and *mispronunciations* are two other common sources of ASR error that impact translation. *Incomplete utterances* are typically produced if the speaker abruptly stops speaking or due to a false-release of the push-to-talk microphone button. Finally, unseen *idioms* often produce erroneous literal translations due of the lack of appropriate transfer rules in the MT parallel training data.

Table 1: Examples of Types of Errors

Error Type	Example
OOV-Name	**My name is Sergeant Gonzales.** *ASR*: my name is sergeant guns all us
OOV-Word	**The utility prices are extortionate.** *ASR*: the utility prices are extort unit
Word Sense	**Does the town have enough tanks.** *Ambiguity:* armored vehicle | storage unit
Homophone	**Many souls are in need of repair.** *Valid Homophones*: soles, souls
Mispron.	**How many people have been harmed by the water when they wash.** *ASR*: how many people have been harmed by the water when they worse
Incomplete	**Can you tell me what these**
Idiom	**We will go the whole nine yards to help.** *Idiom*: the whole nine yards

3. Approach for Active Error Detection and Resolution

Figure 1 shows the architecture of our two-way English to Iraqi-Arabic S2S translation system. In the English to Iraqi direction, the initial English ASR hypothesis and its corresponding translation are analyzed by a suite of *error detection* modules discussed in detail in Section 3.3. An *Inference Bridge* data structure supports storage of these analyses in an interconnected and retraceable manner. The potential classes of errors and their associated spans in the input are identified and ranked in an order of severity using this data structure. A resolution strategy, discussed in detail in Section 3.4, is executed based on the top ranked error.

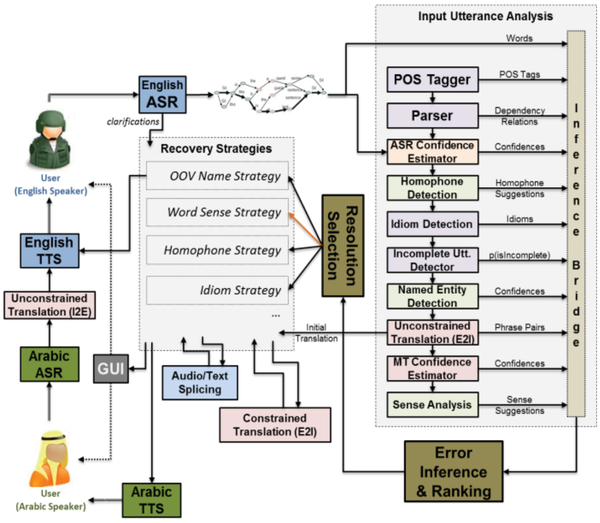

Figure 1: BBN English/Iraqi-Arabic S2S System with Error Recovery in English to Iraqi-Arabic direction

The strategies use a combination of automated and user-mediated interventions to attempt recovery of the concepts associated with the error span. At the end of a strategy, the Arabic speaker may be presented with a translation of the user's input utterance with appropriate corrections; or the English speaker may be informed of the system's inability to translate the sentence along with an explanation of the cause of this failure. With this information, the English speaker can choose to rephrase the input utterance so as to avoid the potential failure. At all times, the English speaker has the option to force the system to proceed with its current translation by issuing the "Go Ahead" command. Our system may be regarded as *high-precision* due to its ability to prevent the transfer of erroneously translated concepts to Arabic speakers. This increased precision comes at the cost of increased effort by the English speaker in terms of performing clarifications and rephrasals. The metrics and results presented in Section 4 study this compromise.

The Arabic to English direction of the system implements a traditional loosely coupled pipeline architecture comprising of the Arabic ASR, Arabic-English MT, and English TTS.

3.1. Baseline ASR System

Speech recognition was based on the BBN Byblos ASR system. The system uses a multi-pass decoding strategy in which models of increasing complexity are used in successive passes in order to refine the recognition hypotheses [11]. In addition to the 1-best and N-best hypotheses, our ASR engine generates word lattices and confusion networks with word posterior probabilities. The latter are used as confidence scores for a variety of error detection components.

The acoustic model was trained on approximately 150 hours of transcribed English speech from the DARPA TRANSTAC corpus. The language model (LM) was trained on 5.8M English sentences (60M words), drawn from both in-domain and out-of-domain sources. LM and decoding parameters were tuned on a held-out development set of 3,534 utterances (45k words). With a dictionary of 38k words, we obtained 11% WER on a held-out test set of 3k utterances.

3.2. Baseline MT System

Our statistical machine translation (SMT) system was trained using a corpus derived from the DARPA TRANSTAC English-Iraqi parallel two-way spoken dialogue collection. The parallel data (773k sentence pairs, 7.3M words) span a variety of scenarios including force protection, medical diagnosis and aid, maintenance and infrastructure, etc.

Table 2: SMT performance for different configurations

System	BLEU	100-TER
Baseline	**16.1**	**35.8**
Boosted	**16.0**	**36.3**
PAC	**16.1**	**36.0**

Phrase translation rules were extracted from bidirectional IBM Model 4 word alignment [12] based on the heuristic approach of [13]. The target LM was trained on Iraqi transcriptions from the parallel corpus and the log-linear model tuned with MERT [14] on a held-out development set (~44.7k words). Table 2 summarizes translation performance on a held-out test set (~38.5k words) of the baseline English to Iraqi SMT system for vanilla phrase-based, boosted alignment [15], and phrase alignment confidence (PAC) [16] systems. We used the PAC SMT models in our system.

3.3. Input Analysis & Error Detection

3.3.1. Automatic Identification of Translation Errors

In order to automatically detect mistranslated segments of the input, we built a confidence estimation system for SMT (similar to [17]) that learns to predict the probability of error for each hypothesized target word. In conjunction with SMT phrase derivations, these confidence scores can be used to identify input segments that may need to be clarified. The confidence estimator relies on a variety of feature classes:

- *SMT-derived features* include forward and backward phrase translation probability, lexical smoothing probability, target language model probability, etc.
- *Bilingual indicator features* capture word co-occurrences in the generating source phrase and the current target word and are obtained from SMT phrase derivations.
- *Source perplexity* is positively correlated with translation error. We used the average source phrase perplexity as a feature in predicting probability of translation error.
- *Word posterior probability* was computed for each target word in the 1-best hypothesis based on weighted majority voting over SMT-generated *N*-best lists.

Reference labels for target words (*correct* vs. *incorrect*) were obtained through automated TER alignment on held-out partitions of the training set (10-fold jack-knifing). The mapping between above features and reference labels was learned with a maximum-entropy (MaxEnt) model. We also exploited the "bursty" nature of SMT errors by using a joint lexicalized label (n-gram) LM to rescore confusion networks generated by the pointwise MaxEnt predictor. Table 3 summarizes the prediction accuracy of correct and incorrect hypothesized Iraqi words on the MT test set (~38.5k words).

Table 3: Incorrect target word classification performance

Method	Dev set	Test set
Majority (baseline)	51.6%	52.6%
MaxEnt + Lexicalized LM	70.6%	71.1%

3.3.2. OOV Named Entity Detection

Detecting OOV names is difficult because of the unreliable features resulting from tokens misrecognized by ASR in the context of an OOV word. We use a MaxEnt model to identify OOV named-entities (NE) in user input [18]. Our model uses lexical and syntactic features to compute the probability of each input word being a name. We trained this model on Gigaword, Wall Street Journal (WSJ), and TRANSTAC corpora consisting of approximately 250K utterances (4.8M words). This includes 450K occurrences of 35K unique named-entity tokens. On a held-out clean (i.e. no ASR error) test set consisting of only OOV named-entities, this model detects 75.4% named-entities with 2% false alarms.

While the above detector is trained on clean text, our real test cases are noisy due to ASR errors in the region of the OOV name. To address this mismatch, we use word posteriors from ASR in two ways. First, an early fusion technique weighs each feature with the word posterior associated with the word from which the feature is derived. This attenuates unreliable features at runtime. Second, we use a heuristically-determined linear combination of ASR word posteriors and the MaxEnt named-entity posterior to compute a score for each word. This technique helps in further differentiating OOV named-entity words since the ASR word posterior term serves as a strong OOV indicator.

Contiguous words with NE posteriors greater than a specified threshold are considered as candidate OOV names. These spans are filtered through a list of known NEs. If a sizeable span (>0.33 seconds) contains at least one non-stopword unknown name token, it is considered for OOV name resolution.

We evaluated our OOV NE detector on an offline set comprising of 2,800 utterances similar in content to the evaluation scenarios described in Section 4.1. We are able to detect 40.5% OOV NEs with 39.1% precision. Furthermore, an additional 19.9% OOV NEs were identified as error spans using the detector described in the next section.

3.3.3. Error Span Detection

We use a heuristically derived linear combination of ASR and MT confidence for each input word in the source language to identify source words that are likely to result in poor translations. We use this error detector to identify a variety of errors including unknown/unseen translation phrases, OOV Word (non-names), user mispronunciations and ASR errors. All consecutive words (ignoring stop words) identified by this detector are concatenated into a single span.

3.3.4. Improving Translation of Multiple Word Senses

Phrase-based SMT is susceptible to word sense translation errors because it constructs hypotheses based on translation rules with relatively limited context. We address this issue through a combination of (a) constrained SMT decoding driven by sense-specific phrase pair partitions obtained using a novel semi-supervised clustering mechanism, and (b) a supervised classifier-based word sense predictor.

3.3.4.1 Semi-supervised phrase pair clustering

The use of constraints for clustering phrase pairs associated with a given ambiguity class into their senses significantly reduces clustering noise and "bleed" across senses due to lack of sufficient context in the phrase pairs. Constraints are obtained in three different ways.

1. *Key-phrase constraints*: Manually annotated key-phrases are used to establish an initial set of constraints between each pair of translation rules corresponding to a given ambiguity class. Two phrase pairs are related by a *must-link* constraint if their source phrases both contain key-phrases associated with the same sense label; or by a *cannot-link* constraint if they contain key-phrases corresponding to different sense labels.
2. *Instance-based constraints*: The word alignment of a sentence pair often allows extraction of multiple phrase pairs spanning the same ambiguous source word. All of these phrase pairs refer to the same sense of the ambiguous word and must be placed in the same partition. We enforce this by establishing *must-link* constraints between them.
3. *Transitive closure*: The process of *transitive closure*

ensures that the initial set of constraints is propagated across all two-tuples of phrase pairs. This leads to a set of constraints that is far larger than the initial set, leading to well-formed, noise-free clusters. We implemented transitive closure as a modified version of the Floyd-Warshall algorithm. We used the transitive closure over key-phrase and instance-based constraints to partition phrase pairs for a given ambiguity class into their respective senses using constrained *k*-means [19].

3.3.4.2 Constrained SMT decoding

Constrained decoding is a form of dynamic pruning of the hypothesis search space where the source phrase spans an ambiguous word. The decoder must then choose a translation from the partition corresponding to the intended sense. We used the partitioned inventories to tag each phrase pair in the SMT phrase table with its ambiguity class and sense identity.

At run time, the constrained SMT decoder expects each input word in the test sentence to be tagged with its ambiguity class and intended sense identity. Unambiguous words are tagged with a generic class and sense identity. When constructing the search graph over spans with ambiguous words tagged, we ensure that phrase pairs covering such spans match the input sense identity. Thus, the search space is constrained only in the regions of non-generic ambiguity classes, and unconstrained elsewhere. By naturally integrating word sense information within the translation model, we preserve the intended sense and generate fluent translations.

Table 4: Concept transfer for ambiguous words

Method	Yes	No	unk
Unconstrained	95	68	1
Constrained	108	22	34
Improvement	**13.7%**	**66.2%**	**n/a**

We evaluated the constrained decoder on a balanced offline test set of 164 English sentences covering all in-vocabulary senses of 73 ambiguity classes that appeared in multiple senses in our training data. Each test sentence contains exactly one ambiguous word. We presented each input sentence and its translation to a bilingual judge, with the ambiguous source word and the target word(s) due to it both highlighted. The judge passes a binary judgment; *yes*, implying that the sense of the source word is preserved, or *no*, indicating an incorrect sense substitution. Non-dominant senses of an ambiguity class may not be translatable if the corresponding partition does not possess sufficient contextual coverage. We count the number of untranslatable ambiguous source concepts separately from correct or incorrect sense transfer. Table 4 summarizes these results.

3.3.4.3 Supervised word sense disambiguation

Complementary to the above framework is a supervised word sense disambiguation system that uses MaxEnt classification to predict the sense of an ambiguous word. Sense predictions by this component are integrated with user input in our mixed-initiative interactive system to identify the appropriate phrase pair partitions for constrained decoding.

We selected up to 250 representative sentences for each ambiguity class from the training corpus and had human annotators (a) assign an identity and description for up to five different senses, and (b) label each instance with the appropriate sense identity. Based on these annotations, we trained separate maximum entropy classifiers for each ambiguity class, with sense identities as target labels. Classifiers were trained for 110 ambiguity classes using *contextual* (window-based), *dependency* (parent/child of ambiguous word), and corresponding *part-of-speech* features.

We performed an offline evaluation of the sense classifiers by using them to predict the sense of the ambiguity classes in held out test sentences. The most frequent sense of an ambiguity class in the training data served as a baseline (chance level) for that class. The baseline word sense predication accuracy rate over 110 ambiguity classes covering 2,324 sentences containing ambiguous words was 73.7%. This improved to 88.1% using the MaxEnt sense classifiers.

3.3.5. *Homophone Detection and Correction*

A common problem with ASR is the substitution of a different word that sounds identical to the spoken word (e.g. “role” vs. “roll”). To alleviate this problem, we developed a state-of-the-art automatic homophone detection and correction module based on MaxEnt classification. We induced a set of homophone classes from the ASR lexicon such that the words in each class had identical phonetic pronunciation. For each homophone class, we identified training examples containing the constituent words. A separate classifier was trained for each homophone class with the correct variants as the target labels. This component essentially functions as a strong, local, discriminative language model. The features used for the homophone corrector are identical to those used for supervised word sense disambiguation (Section 3.3.4.3).

We evaluated this component by simulating, on a held-out test set for each homophone class, 100% ASR error by randomly substituting a different variant for each homophone constituent in these sentences. We then used the classifier to predict the word variant for any slot corresponding to a homophone class constituent. The overall correction rate over 223 homophone classes covering 174.6k test sentences containing homophone classes was 95.8%. Similarly, the false correction rate (simulated by retaining the correct homophone variant in the test set) was determined to be 1.3%.

3.3.6. *Idiom Detection*

Idioms unseen in SMT training usually generate incomprehensible literal translations. To detect and pre-empt translation errors originated from idioms, we harvested a large list of English idioms from public domain sources to use in a simple string matching front-end. However, the harvested idioms are usually in a single canonical form, e.g. “give *him* a piece of my mind”. Thus, simple string match would not catch the idiom “give *her* a piece of my mind”. We used two approaches to expand coverage of the idiom detector.

1. *Rule-based idiom expansion:* We created rules for pronoun expansion (e.g. “his” → “her”, “their”, etc.) and verb expansion (e.g. “give her a piece of my mind” → “gave her a piece of my mind”), being conservative to avoid explosion and creation of nonsense variants.
2. *Statistical idiom detector:* We trained a binary MaxEnt classifier that predicts whether any input n-gram is an idiom. We used 3.2k gold standard canonical idioms as positive samples and all 15M non-idiom n-grams in our data as negative samples. On a balanced set containing

unseen idiom variants and non-idioms, this classifier gave us a detection rate of 33.2% at 1.8% false alarm.

3.3.7. *Incomplete Utterance Detection*

In order to detect user errors such as intentional aborts after mis-speaking, or unintentional pushing or releasing of the "record" button, we built an incomplete utterance detector (based on a MaxEnt classifier) that identifies fragments with ungrammatical structure in recognized transcriptions. Training data for incomplete utterances were automatically generated using an error simulator that randomly removed words from the beginning and/or end of a clean, fully-formed sentence. A number of lexical and syntactic features were used to train and evaluate the incomplete utterance classifier.

We trained a binary classifier on approximately 771k fully formed sentences and varied the number of automatically generated incomplete utterances. We evaluated the classifier on a balanced test set of 1,000 sentences with 516 auto-generated sentences that were verified by hand to be positive examples of incomplete sentences. At a false alarm rate of 5%, the incomplete utterance detector demonstrated a detection rate of 41%. Syntactic and part-of-speech features were particularly powerful at identifying this error type.

3.4. Error Resolution Strategies

Our implementation of error resolution strategies follows a multi-expert architecture along the lines of Jaspis [20] and Rime [21]. Each strategy has been manually designed to resolve one or more types of errors discussed in Section 2.

Figure 2 illustrates 9 interaction strategies used by our system. Each strategy is comprised of a sequence of steps which include actions such as TTS output, user input processing, translation (unconstrained or constrained) and other error type specific operations.

The *OOV Name* and *ASR Error* strategies are designed to interactively resolve errors associated with OOV entities (names and non-names), ASR errors and MT errors. When a span of words is identified as an OOV named-entity, the user is asked to confirm whether the audio segment spanning those words actually corresponds to a name (Excerpt A), following which the segment is spliced in place of the target phrases corresponding to that span. In the case where a (non-name) error span is detected by the detector described in Section 3.3.3, the user is asked to rephrase the utterance. This strategy

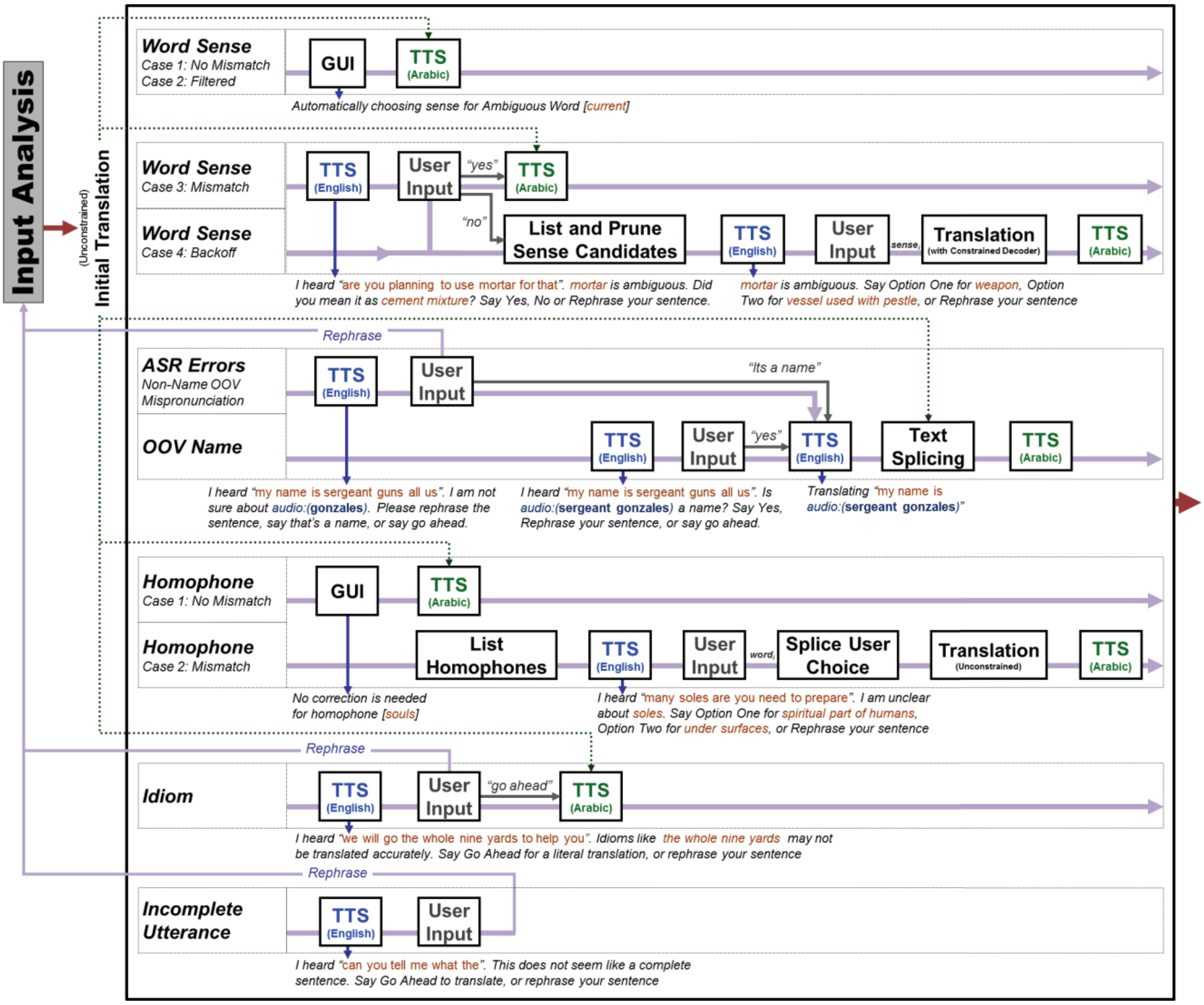
Input Analysis
(Unconstrained)
Initial Translation
Word Sense
Case 1: No Mismatch
Case 2: Filtered
GUI
TTS (Arabic)
Automatically choosing sense for Ambiguous Word [current]
Word Sense
Case 3: Mismatch
TTS (English)
User Input
"yes"
TTS (Arabic)
"no"
Word Sense
Case 4: Backoff
List and Prune Sense Candidates
TTS (English)
User Input
sense,
Translation (with Constrained Decoder)
TTS (Arabic)
I heard "are you planning to use mortar for that". mortar is ambiguous. Did you mean it as cement mixture? Say Yes, No or Rephrase your sentence.
mortar is ambiguous. Say Option One for weapon, Option Two for vessel used with pestle, or Rephrase your sentence
Rephrase
ASR Errors
Non-Name OOV
Mispronunciation
TTS (English)
User Input
"Its a name"
OOV Name
TTS (English)
User Input
"yes"
TTS (English)
Text Splicing
TTS (Arabic)
I heard "my name is sergeant guns all us". I am not sure about audio:(gonzales). Please rephrase the sentence, say that's a name, or say go ahead.
I heard "my name is sergeant guns all us". Is audio:(sergeant gonzales) a name? Say Yes, Rephrase your sentence, or say go ahead.
Translating "my name is audio:(sergeant gonzales)"
Homophone
Case 1: No Mismatch
GUI
TTS (Arabic)
Homophone
Case 2: Mismatch
List Homophones
TTS (English)
User Input
word,
Splice User Choice
Translation (Unconstrained)
TTS (Arabic)
No correction is needed for homophone [souls]
I heard "many soles are you need to prepare". I am unclear about soles. Say Option One for spiritual part of humans, Option Two for under surfaces, or Rephrase your sentence
Rephrase
Idiom
TTS (English)
User Input
"go ahead"
TTS (Arabic)
I heard "we will go the whole nine yards to help you". Idioms like the whole nine yards may not be translated accurately. Say Go Ahead for a literal translation, or rephrase your sentence
Rephrase
Incomplete Utterance
TTS (English)
User Input
I heard "can you tell me what the". This does not seem like a complete sentence. Say Go Ahead to translate, or rephrase your sentence

Figure 2. Interaction Strategies for Error Resolution

Table 5: Examples of Error Recovery Strategies

A. Out of Vocabulary Name	
User Utt.:	My name is Sergeant Gonzalez
ASR:	**my name is sergeant guns all us**
Sys.:	I heard "my name is sergeant guns all us". Is (audio:Sergeant Gonzalez) a name? Say Yes or Rephrase your sentence, or Go Ahead.
ASR:	**yes**
Sys.:	Translating "my name is (audio: Sergeant Gonzalez)".
B. OOVs and other ASR/MT error	
User Utt.:	Are your wages enough to subsist on?
ASR:	**are your wages enough to sub system**
Sys.:	I heard "are your wages enough to sub system". I am not sure about (audio: subsist). Please rephrase the sentence, say that's a name, or say Go Ahead.
User:	**do you make enough money**
Sys.:	Translating "do you make enough money"
C. Word Sense	
User Utt.:	I am looking for the man in charge.
ASR:	**i am looking for the man in charge**
Sys.:	I heard "i am looking for the man in charge". Charge is ambiguous. Say Option One for in command of, Option Two for payment, Option Three for accusation, or Rephrase your sentence.
User:	**option one**
Sys.:	Translating "i am looking for the man in charge" using charge as in command of.
D. Homophone	
User Utt.:	How much do you pay for the fare?
ASR:	**how much do you pay for the fair**
Sys.:	I heard "how much do you pay for the fair". I am unclear about fair. Say Option One for price of transportation, Option Two for free from bias, Or Rephrase your sentence
ASR:	**option one**
Sys.:	Translating "how much do you pay for the fare"
E. Improper Input/Fragment	
User Utt.:	I am going for the-
ASR:	**i am going for the**
Sys.:	I heard "i am going for the". This does not seem like a complete sentence. Say Go Ahead to translate or Rephrase your sentence.
ASR:	**i am going to the village**
Sys.:	Translating "i am going to the village"
F. Idiom	
User Utt.:	What can we do about this dead space here?
ASR:	**what can we do about this dead space here**
Sys.:	I heard "what can we do about this dead space here". Idioms like dead space may not be translated accurately. Say Go Ahead for literal translation, or Rephrase your sentence.
ASR:	**go ahead**
Sys.:	Translating "what can we do about this dead space here"

is suitable for handling multiple error types including OOVs, mispronunciations, and ASR/MT errors. Additionally, the *ASR Errors* strategy has been designed to capture a large fraction of the OOV name false negatives (i.e. missed detections) by allowing the user to indicate if the error span is a name (Excerpt B). Because of the similar nature of the errors handled by these two strategies, we have found it beneficial to maintain reciprocity between them to resolve all the errors handled by these strategies.

The four *Word Sense* (WS) disambiguation strategies resolve sense ambiguity errors. The underlying principle behind the strategies is that the sense of an ambiguous word must be confirmed by at least two of four possible independent sources. These four sources include (a) the translation system (sense lookup corresponding to phrase pair associated with the ambiguous word), (b) sense-inventory that lists source phrase keywords, (c) sense predicted by supervised model for sense-class and (d) sense specified by the user. Some of these sources may not be available for certain words. *Case 2: Filtered* strategy corresponds to the case where (a) and (b) agree. In this case, the user is shown a message using the GUI and the system proceeds to present the translation to the Arabic speaker. Similarly, *Case 1: No Mismatch* strategy correspond to the case where (a) and (c) agree. If these three sources are unable to resolve the sense of the word, the user is asked to confirm the sense identified by source (a) following the *Case 3: Mismatch* strategy. If the user rejects that sense, a list of senses is presented to the user (*Case 4: Backoff* strategy). The user-specified sense drives constrained decoding to obtain an accurate translation which is then presented to the Arabic speaker. An example of this case is shown in Excerpt C of Table 5.

Albeit simpler, the two homophone (HP) resolution strategies mimic the WS strategies in principle and design. The observed homophone variant produced by the ASR must be confirmed either by the MaxEnt model (*Case 1: No Mistmatch*) of the corresponding homophone class or by the user (*Case 2: Mismatch*) as shown in Excerpt D. The input utterance is modified (if needed) by substituting the resolved homophone variant in the ASR output which is then translated and presented to the Arabic speaker.

Strategies for resolving errors associated with idioms and incomplete utterances (Excerpts E and F) primarily rely on informing the user about the detection of these errors. The user is expected to rephrase the utterance to avoid these errors. For idioms, the user is also given the choice to force a literal translation when appropriate.

At all times, the user has the ability to rephrase the initial utterance as well as to force the system to proceed with the current translation. This allows the user to override system false alarms whenever suitable. The interface also allows the user to repeat the last system message which is helpful for comprehension of long prompts presented by the system.

4. Experimental Results

In this section, we present results from a preliminary evaluation for measuring the benefit of active error detection and resolution capability in S2S systems. Note that this evaluation does not contrast the various design choices involved in our implementation. Instead, we focus on a holistic evaluation of the system.

4.1. Evaluation Approach and Metrics

Multiple English speaking human subjects interacted with the system to communicate 20 scenarios to an Arabic speaker. Each scenario consists of 5 "starting" utterances. The subject speaks one English starting utterance at a time and is allowed to freely respond to any interactive recovery dialog initiated by the system. Interaction corresponding to each starting utterance comes to an end when the system presents an Arabic translation. Each starting utterance has been designed to pose exactly one of the seven error types discussed in Section 2. This is often compounded by unexpected ASR errors.

Prior to the start of the experiment, each speaker was trained using five scenarios (25 starting utterances) to allow the speakers to familiarize themselves with the system prompts. In all, we were able to collect interactions corresponding to 103 starting utterance for this evaluation. The primary measure of success of a S2S system is its ability to accurately communicate concepts across the language pair. High Level Concept Transfer (HLCT) [22] has been used in the past for multi-site S2S system evaluations under the DARPA TRANSTAC program.

In this paper, we adapt HLCT to focus on the concept associated with the erroneous span (word/phrase) in each starting utterance. We consider only the span associated with the intended error. Each erroneous concept is considered as transferred if it is conveyed accurately in the translation. The benefit of using active error detection and recovery is measured as the improvement in HLCT between the initial translation (i.e. before recovery) and final translation (i.e. after recovery). This is demonstrated in Table 6. In addition to improvement in concept transfer, we also present error detection accuracy metrics as well as analysis of number of clarification turns.

Table 6: Example of HLCT for Erroneous Concept

User Utt: i have heard that the utility prices are extortionate
Before Clarification
ASR: **i have heard that the utility prices are extort unit**
MT: آنيسمعتإنهالخدماتالأسعاروحدة
Gloss: I heard that services all prices are same
Concept Transferred? **No** ✗
After Clarification
ASR: **the price for utilities seems very high**
MT: السعرالخدماتمبينكلشعالية
Gloss: the price of services seem to be very high
Concept Transferred? **Yes** ✓

4.2. Results

Table 7: HLCT for Erroneous Spans
(#: count of utterances transferred, %: percentage transferred)

Intended Error	Count	Initial Transfer		Final Transfer		Change
		#	%	#	%	%
OOV-Name	12	1	8.33	5	41.67	33.33
OOV-Word	46	3	6.52	20	43.48	36.96
Word Sense	18	4	22.22	10	55.56	33.33
Homophone	15	4	26.67	5	33.33	6.67
Mispronunciation	5	1	20.00	2	40.00	20.00
Idiom	2	0	0.00	1	50.00	50.00
Incomplete	5	0	0.00	5	100.00	100.00
All	**103**	**13**	**12.62**	**48**	**46.60**	**33.98**

ASR WER for the utterances used in this evaluation was 23%. Table 7 shows the initial, final and change (improvement) in HLCT for the erroneous span for each of the error types. Overall, our S2S system equipped with active error detection and recovery is able to improve the transfer of erroneous concepts by 33.98%. This improvement is more prominent in the case of certain types of errors such as OOVs.

Table 8 shows the detection accuracy within our evaluation set for each type of error. Two different detection accuracy metrics are shown. First, %correct is the fraction of errors that were identified as the intended error. Second, %recoverable is the fraction of errors that were identified as an error whose strategy supports recovery from the intended error. For example, an OOV-Name incorrectly identified as an error span is still recoverable because the strategy allows the user to inform the system that the span is a name. Note that %recoverable is always greater than or equal to %correct because correctly identified errors is considered recoverable in this analysis. Overall, 33% of errors are identified correctly and 59.2% are identified as a potentially recoverable error. Of these, as shown in Table 7, 46.6% errors are actually recovered by our recovery strategies. On average, the recovery strategies require 1.4 clarification turns.

Table 8: Error Detection Accuracy
(*Intended and Actual Errors may differ)

Intended Error	%Correct	%Recoverable
OOV-Name	41.7	75.0
OOV-Word	37.8	75.6
Word Sense*	16.7	16.7
Homophone*	31.3	50.0
Mispronunciation	60.0	60.0
Idiom	0.0	0.0
Incomplete	20.0	80.0
All	**33.0**	**59.2**

5. Discussion and Future Work

Error recovery strategies have been shown to be effective at improving task success in several applications [23][24]. However, their application to S2S systems has been limited [10][25]. In [25], the authors developed a wide range of repair strategies for narrow domain S2S. However, this implementation did not have any active error detection. Instead, it was delegated to the user who was asked to highlight erroneous words resulting from ASR errors.

The active error detection and interactive recovery strategies described in this paper go well beyond user confirmation [10] and repair strategies of [25]. As seen in the results presented in Section 4, well-designed error-specific recovery strategies can significantly improve (34%) the communication of erroneous concepts despite moderate error detection capabilities (33%). We also note that this state-of-the-art implementation is able to recover only about 46.6% erroneous concepts. This suggests a significant scope for improvement of S2S systems in this line of investigation.

While our current system has demonstrated an effective approach for enhancing eyes-free S2S systems with active error detection and recovery, this system implements these capabilities in only one direction (English to Arabic). Developing similar capabilities in both directions of S2S presents exciting challenges. In particular, the participation of the foreign language speaker in the error recovery activity offers both opportunities for developing novel interaction strategies as well as challenges such as addressee detection, speaker diarization and prompt targeting in addition to addressing increased computational needs for bi-directional error detection.

In addition to extending our system to a 2-way implementation, further scientific inquiry to evaluate the effectiveness of error recovery in S2S systems is necessary. Specifically, evaluation presented in this paper has two shortcomings. First, each utterance in the evaluation scenarios is designed to have one of the 7 expected errors. This was necessary in these preliminary evaluations to gather a representative sample of each of types of error within a reasonable number of utterances collectable with a small number of human subjects. However, in practice, many utterances may have none or multiple expected errors. While our current system is capable of dealing with these situations, the evaluation presented here does not measure system performance under such conditions.

Second, in a practical S2S system, often the two speakers are able to perform limited amount of error recovery. While this form of error recovery is often expensive in terms of user time and effort, a thorough evaluation should compare this form of recovery to automated error recovery.

6. References

[1] Wahlster, W., “Verbmobil: translation of face-to-face dialogs”, *Proc. of European Conf. on Speech Comm. And Tech., 1993, p. 29-38*

[2] Nakamura, S., Markov, K., Nakaiwa, H., Kikui, G., Kawai, H., Jitsuhiro, T., Zhang, J.S., Yamamoto, H., Sumita, E., and Yamamoto, S. "The ATR multilingual speech-to-speech translation system," *IEEE Trans. on Audio, Speech, and Language Processing, 14.2 p. 365-376, 2006*

[3] Stallard, D., Prasad, R., Natarajan, P., Choi, F., Saleem, S., Meermeier, R., Krstovski, K., Ananthakrishnan, S., and Devlin, J. “The BBN TransTalk Speech-to-Speech Translation System”, *Speech and Language Technologies, InTech, 2011, p. 31-52*

[4] Google Translate, http://translate.google.com/

[5] Eck, M., Lane, I., Zhang, Y., and Waibel, A. "Jibbigo: Speech-to-speech translation on mobile devices," *IEEE Wksp. on SLT, 2010, p.165-166*

[6] Zhang, R., Kikui, G., Yamamoto, H., Watanabe, T., Soong, F., and Lo, W. K. “A unified approach in speech-to-speech translation: integrating features of speech recognition and machine translation”, *Proc. of 20th COLING, Stroudsburg, PA, USA, 2004*

[7] He, X. and Deng, L. “Optimization in Speech-Centric Information Processing: Criteria and techniques”, *Proc. of ICASSP, 2012, p. 5241-5244*

[8] Matsoukas, S., Bulyko, I., Xiang, B., Nguyen, K., Schwartz, R. and Makhoul, J. "Integrating Speech Recognition and Machine Translation," *Proc. of ICASSP, 2007, p. 1281- 1284*

[9] Stallard, D., Kao, C. Krstovski, K., Liu, D., Natarajan, P., Prasad, R., Saleem, S., and Subramanian, K., “Recent improvements and performance analysis of ASR and MT in a speech-to-speech translation system”, *Proc. of ICASSP, 2008, p. 4973-4976*

[10] Prasad, R., Natarajan, P., Stallard, D., Saleem, S., Ananthakrishnan, S., Tsakalidis, S., Kao, C.-L., Choi, F., Meermeier, R., Rawls, M., Devlin, J., Krstovski, K., Challenner, A. “BBN TransTalk: Robust multilingual two-way speech-to-speech translation for mobile platforms,” *Computer Speech & Language, 2011*

[11] Nguyen L., and Schwartz, R. “Efficient 2-pass N-best decoder,” *Proc. of Eurospeech, Rhodes, Greece, 1997, p. 167–170.*

[12] Brown, P. E., Della Pietra, V. J., Della Pietra, S. A., and Mercer, R. L. “The Mathematics of Statistical Machine Translation: Parameter Estimation”, *Computational Linguistics, 19, 1993, p. 263–311*

[13] Koehn, P., Och, F. J., and Marcu, D. “Statistical Phrase-based Translation”, *NAACL-HLT, 2003, p. 48–54*

[14] Och, F. J., “Minimum Error Rate Training in Statistical Machine Translation”, *Proc. of 41st ACL, Stroudsburg, PA, USA, 2003, pp. 160-167*

[15] Ananthakrishnan, S., Prasad, R., and Natarajan, P. “An Unsupervised Boosting Technique for Refining Word Alignment”, *Proc. of IEEE Wksp. on SLT, Berkeley, CA, 2010*

[16] Ananthakrishnan, S., Prasad, R., and Natarajan, P., “Phrase Alignment Confidence for Statistical Machine Translation”, *Proc. of Interspeech, Makuhari, Japan, 2010, p. 2878-2881*

[17] Bach, N., Huang F. and Al-Onaizan, Y. “Goodness: A Method for Measuring Machine Translation Confidence”, *Proc. of 49th ACL-HLT, 2011, Portland, OR, USA*

[18] Kumar, R., Prasad, R., Ananthakrishnan, S., Vembu, A. N., Stallard, D., Tsakalidis, S., Natarajan, P. “Detecting OOV Named-Entities in Conversational Speech”, *Proc. of Interspeech, 2012, Portland, OR, USA*

[19] Wagstaff, K., Cardie, C., Rogers, S., and Schrödl, S. “Constrained k-means Clustering with Background Knowledge”, *Proc. of 18th ICML, 2001, San Francisco, CA, USA, p. 577–584*

[20] Turunen, M. and Hakulinen, J. “Jaspis - An Architecture for Supporting Distributed Spoken Dialogues”, *Proc. of Eurospeech, Geneva, Switzerland, 2003*

[21] Nakano, M., Funakoshi, K., Hasegawa, Y., Tsujino, H., “A Framework for Building Conversational Agents Based on a Multi-Expert Model”, *Proc. of 9th SigDial Workshop on Discourse and Dialog, Columbus, Ohio, 2008*

[22] Weiss, B. A., Schlenoff, C.I., Sanders, G. A., Steves, M. P., Condon, S., Phillips, J. and Parvaz, D. "Performance Evaluation of Speech Translation Systems", *Proc. of 6th LREC, 2008*

[23] Turunen, M. and Hakulinen, J. “Agent-based Error Handling in Spoken Dialogue Systems”, *Proc. of Eurospeech, 2001*

[24] Bohus, D. and Rudnicky, A. I. "Sorry, i didn’t catch that!- an investigation of non-understanding errors and recovery strategies", *Proc. of SIGDial, 2005, p. 128-143*

[25] Suhm, B., Myers, B. and Waibel, A. "Interactive recovery from speech recognition errors in speech user interfaces," *Proc. of 4th ICSLP, 1996. p.865-868*