Automated Radio Operator Utterance Recognition for US Navy Training

Morgan Ulinski[1], Ethan Medjuck, Kellen Bixler[1], Henry Phillips[2]
[1]Soar Technology LLC **[2]Advanced Distributed Learning (ADL) Initiative**
Orlando, FL **Orlando, FL**
morgan.ulinski@soartech.com, ethanmedjuck589@gmail.com,
kellen.bixler@soartech.com, henry.phillips.ctr@adlnet.gov

ABSTRACT

Advances in speech recognition technology have enabled the automation of grading for speech-based training courses that have historically required a trained instructor. This paper describes an effort conducted to apply this technology to Navy shipboard radio operator training, which called for simulation of two-way radio communications between the student and various Non-Player Characters (NPCs).

The Navy training use case calls for instances where a student is required to provide a *verbatim* radio response to a prompt, consisting of a set of words delivered in a specific order, as well as *non-verbatim* radio responses, whereby students are allowed to include extra words provided they start with a salutation and include a phrase from each required keyword slot.

In place of an instructor, these verbatim and non-verbatim instance courses use speech recognition to grade students during prompted interactions with NPCs in classroom training environments. To evaluate this system, the authors generated validation corpora of human utterances representing true positive (correct) and true negative (incorrect) elements of the verbatim and non-verbatim coursework utterances.

For coursework grading, we designed and built a system based on an all-in-one speech recognition and parsing tool. We created custom language models to recognize the required verbatim and non-verbatim utterances and refined these models based on our validation corpus. The team then conducted hyperparameter tuning through integration of an open-architecture testing system that leveraged formal methods, knowledge-based testing, and optimized search.

This paper outlines the details of our approach for corpus generation, language model construction, and model validation. Results are presented achieving 97% recognition accuracy on our validation corpora of over 10,000 accurate and inaccurate utterances. Results suggest that a finely tuned and trained speech recognition and parsing model can work extremely well for the use case of supporting radio operator training.

ABOUT THE AUTHORS

Morgan Ulinski is a Research Scientist at Soar Technology, LLC, focused on natural language processing. She received her B.A. in Computer Science and Linguistics from Cornell University in 2009 and her Ph.D. in Computer Science from Columbia University in 2019. Her previous work has applied natural language processing techniques to low resource and endangered languages, spatial language interpretation, and detection of linguistic hedges.

Ethan Medjuck is a computer engineer and entrepreneur. Ethan graduated from Florida Polytechnic University with a Bachelor of Science in Computer Engineering. He went on to work for Wind Talker Innovations Inc. and then Soar Technology, LLC. While at SoarTech he specialized in NGTS simulation and speech AI.

Kellen Bixler is an AI Engineer at Soar Technology, LLC, focused on swarming autonomy. He received his B.S.E in Computer Science from University of Michigan in 2017. His previous work includes research and development using artificial and swarm intelligence for distributed control of complex systems ranging from multi-air/ground vehicle control.

Henry L. Phillips IV is Program Manager for the Advanced Distributed Learning (ADL) Initiative. He served 20 years on active duty as a Naval Aerospace Experimental Psychologist including service as Executive Officer of the Naval Air Warfare Center Training Systems Division. He holds a PhD in Industrial/Organizational Psychology from the University of Houston.

Automated Radio Operator Utterance Recognition for US Navy Training

Morgan Ulinski[1], Ethan Medjuck, Kellen Bixler[1], Henry Phillips[2]
[1]Soar Technology LLC **[2]Advanced Distributed Learning (ADL) Initiative**
Orlando, FL **Orlando, FL**
morgan.ulinski@soartech.com, ethanmedjuck589@gmail.com,
kellen.bixler@soartech.com, henry.phillips.ctr@adlnet.gov

INTRODUCTION

Advances in automated speech recognition (ASR) technology have enabled the automation of grading for speech-based training courses that have historically required a trained instructor. This paper describes an effort conducted to apply this technology to Navy shipboard radio operator training, which called for simulation of two-way radio communications between the student and various Non-Player Characters (NPCs).

ASR refers to the use of Machine Learning or Artificial Intelligence (AI) technology to process human speech into readable text. The text is often further processed to produce machine-readable outputs that can be integrated into software applications. ASR applications in operational training environments must be developed and deployed with awareness of critical human systems integration questions including but not limited to functional interface design, speech controls, incorporation of the ASR capability into guidelines for training and instructional design, and trainee and user willingness to use and rely on such a system, among others (Noyes & Haas, 2010). The question of how such an application can be seamlessly integrated into the training process to add value and avoid disruption is paramount. The focus of this paper is on a single narrow component of these larger issues: *development of a speech recognition component with high enough accuracy that it can potentially be employed in place of a human grader*. While this component was ultimately integrated with a larger simulation-based training application, the details of that system are out of scope for the work presented here.

The particular Navy training use case for this project calls for two types of radio responses: instances where a student was required to provide a *verbatim* radio response to a prompt (a response consisting of a fixed set of words delivered in a specific order with no additions or deletions) as well as *non-verbatim* radio responses (whereby students must use a phrase from each provided keyword slot and are allowed to include unlimited extra words). In addition, the Navy requirement for this project was that the speech recognition system should maintain an average 97% accuracy rate when judging whether a radio response is correct or incorrect. The 97% accuracy threshold required our system to maintain an extremely low word-error rate. To meet the requirement, we must maintain a 0% word-error rate in 97% of cases. For comparison, OpenAI's best Whisper model achieves a word-error rate of 2.5% (Radford, et al., 2023). Achieving such a low error rate is particularly difficult in specialized domains because existing ASR systems have not been trained on their unique vocabulary. In the Navy training use case, the radio responses include many acronyms and other specialized speech patterns. If large amounts of audio data in the specialized domain are available, it is possible to mitigate this issue by applying traditional fine-tuning techniques to an off-the-shelf ASR system. However, in most cases, there is not enough recorded audio data to make this approach feasible. Instead, our approach combines an existing ASR system with a human-defined domain-specific language model.

In the following sections, we describe the speech recognition and parsing system we used, the process for creating an intent model, collection of a data set of audio recordings, our approach for testing and grading recorded utterances, hyperparameter tuning of the underlying ASR model, and our results. We show that our system exceeded the 97% accuracy requirement and achieved word error rates well below state of the art.

SPEECH RECOGNITION AND PARSING SYSTEM

The team used an existing all in one speech recognition and analysis/parsing tool called SpeechZero for the purpose of this study (Stensrud, Taylor, & Crossman, 2006; Lebanoff, et al., 2021). An important detail in ASR is the translation of speech into speaker intentions. Similar or even identical strings of words can have very different intended

meanings, which can be parsed from the setting, context, or words/word patterns surrounding the word in question. This translation of words by context is defined as a computer-interpretable *intent model* that classifies natural language utterances by speaker intention. A related process is *semantic parsing*, which converts a natural language sentence or phrase into a formal representation of its meaning that a machine can understand. SpeechZero combines an off-the-shelf ASR engine with a manually-created intent model to provide accurate speech recognition and semantic parsing in specialized domains. In this project, the team used QuartzNet (Kriman, et al., 2020) for the end-to-end ASR engine.

An intent model compatible with the SpeechZero tool consists of the following:

- **Intents:** Represent the set of intentions that a speaker might have.
- **Atoms:** Components of intents that are too small to have unique contextual meaning. Atoms can have multiple meanings across different intents and are used to define and differentiate among intents.
- **Categories:** A set of atoms and/or intents grouped under a common name.

An intent model is used in two ways by the tool. First, it is used to generate an n-gram language model for use in ASR decoding. An n-gram language model is a statistical model of language based on the assumption that the probability of the next word in a sequence depends on a fixed size window of previous (n – 1) words. Second, it is used as a grammar to semantically parse the ASR output. Figure 1 outlines the architecture for this system, including a representation of how grading of student utterances was incorporated.

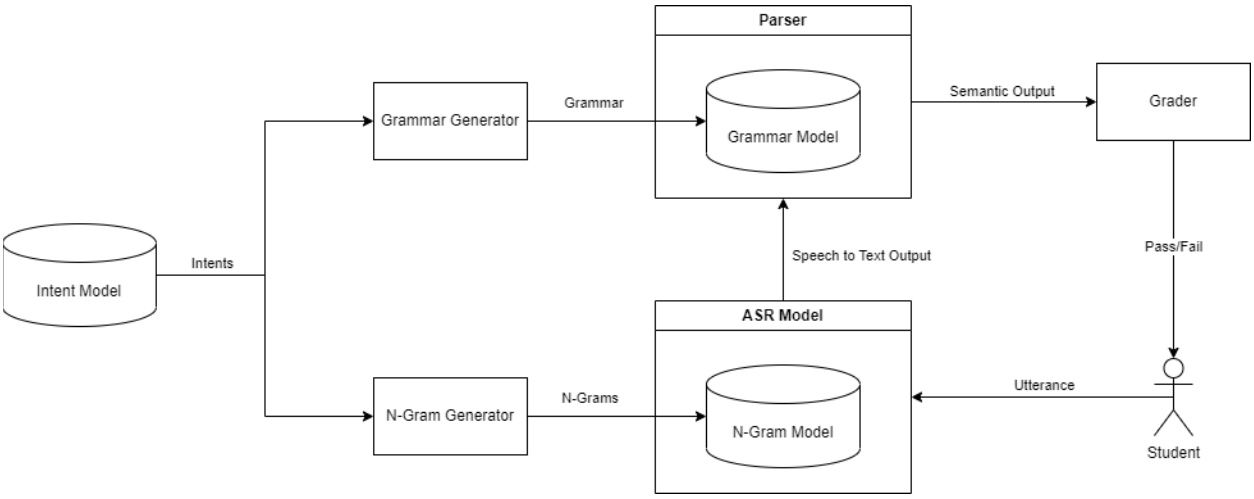

Figure 1. Speech recognition & parsing architecture

The typical use case for this tool is that, given a spoken utterance, the system should identify the most likely intent for that utterance and return a semantic parse, defined as a set of semantic roles mapped to elements in the utterance. Our use case is slightly different: given a spoken utterance and the intent that it should match, the system should identify which of the required components of that intent are present or absent. To do this, we added a way to constrain the set of intents that the system would consider as candidates so that the specified intent would always be selected. Then, we defined the intent model in such a way that the "semantic parse" includes all required words and phrases in an utterance rather than traditional semantic roles.

INTENT MODEL CONSTRUCTION FOR VERBATIM AND NON-VERBATIM UTTERANCES

Verbatim

In this paper, a *verbatim utterance* is defined as an utterance that includes a set of specific mandated words, which may be subject to variations in the pronunciation of acronyms, and/or individual letters and numbers.

A system intent model for verbatim utterances was constructed procedurally in a Python script. The authors were provided with a list of 3,515 verbatim utterances used in shipboard Navy training. These data were cleaned and prepared in the following ways to make them suitable for use in the intent model:

- Correcting misspellings and typos, e.g. starboaord => starboard, accirdancewith => accordance with
- Adding spaces between acronym characters, e.g. MZD => M Z D
- Converting numbers to English words, e.g. 27 => twenty seven
- Converting dashes to "tac" when appropriate, e.g. 5-4-0-W => five tac four tac zero tac whiskey

Once all the utterances were processed and duplicates merged, the team was left with 1,650 unique utterances. Each of these was used as the basis for a single intent in the model.

The next step of the processing was to break down the utterances into slots for an intent. We started with an extremely simple intent model where each unique utterance (intent) was considered a single atom. As we acquired more data, however, we found that this would not be sufficient. Using only one atom per intent, we found that adding additional pronunciations or spellings of individual words could not be adequately encoded without creating complicated permutations of each entire utterance. We therefore changed our strategy to define the slots as individual words or short phrases. This allowed us to easily encode variations in pronunciations (*turban* for *turbine*) and spellings (*stand by* vs. *standby*).

Splitting into words and phrases was done using the Natural Language Toolkit (NLTK) (Bird, Loper, & Klein, 2009) multi-word expression (MWE) tokenizer. A MWE is natural language expression made up of at least 2 words, that acts as a single unit at some level of linguistic analysis. For example, *in accordance with* acts as a preposition and *gas turbine* is a compound noun. We used a custom lexicon of MWEs extracted from the verbatim utterances. Each utterance was tokenized using the MWE tokenizer, and each resulting token is considered a slot for its corresponding intent. Each slot in an intent can be one of the following:

- an *atom*, representing single words like *and, the, number, tank,* etc. The atoms in the intent model consist of the set of unique words across all utterances.
- an *intent fragment*, used to represent phrases like *in accordance with, combining gear, boost shaft,* etc. The intent fragments are further broken down into slots in their own definitions so that all types decompose to atoms.
- a *category*, used in cases where multiple variations of words or phrases are acceptable, such as "*m m r"* and "*main machinery room"* as acceptable representations of the term "MMR". A category can include both atoms and intent fragments.

As we processed actual audio data through the system, we found that selected atoms needed to be augmented with additional values to counter the effect of homophone errors in the ASR output and to handle words with multiple valid pronunciations. For example, the ASR often outputted *i* instead of the word *aye*, in student utterances, so we added the value *i* to the *aye* atom. Similarly, we added the value *turban* to the *turbine* atom, to help the system recognize these as two acceptable pronunciations of the atom representing the idea "turbine." ASR errors were accommodated at the phrase level as well. For example, the ASR output tended to include "*loo boil"* in place of the intended utterance of the term "*lube oil*." To handle these, we represented the phrase *lube oil* as a category with two different intent fragments as its values. These mappings were encoded in the Python script as dictionaries.

Non-verbatim

In this paper, a *non-verbatim utterance* is defined as an utterance that includes a set of specific required words and phrases beginning with a salutation that otherwise can be spoken in any order, and that may be accompanied by an unlimited number of additional words.

The authors were provided with a list of 1,093 non-verbatim utterances. Each of these had an "Expected Communication", which looked similar to a verbatim utterance, and a set of required keyword slots, each of which contained one or more words or phrases. These were intended to represent utterances that could be repeated correctly as multiple combinations of the keywords of which they were composed. A correctly spoken non-verbatim utterance must include at least one word or phrase from every keyword slot. One keyword slot might contain the words *electric*

and *electrical*, while another might contain both *DAU* and *Data Acquisition Unit*. Each expected communication in the non-verbatim utterance list was comprised of a set of valid keywords but contained additional words that made the utterance into a grammatically correct English sentence(s).

Like the verbatim intent model, the non-verbatim intent model was built procedurally in a Python script. The difference is that instead of breaking down a *complete verbatim utterance* into atoms, the basis of the non-verbatim intents was the *keyword slots*. Both the expected communications and the keywords were preprocessed and cleaned with the same methodology used for the verbatim utterances. However, in the case of the non-verbatim utterances, the merging of utterances was done based on identical keyword sets and not the words comprising the expected communications. After preprocessing and merging utterances, there were 553 unique non-verbatim utterances remaining. Each of these unique non-verbatim utterances was used as the basis for an intent. In a single intent, each keyword slot was represented as a category containing multiple values. Each possible value was represented as an intent fragment (for phrases) or an atom (for single words). The individual words in each intent fragment were represented as atoms.

Iteration on the non-verbatim intent model followed a similar pattern to that of the verbatim model. Although many of the variations in pronunciation were already encoded into the atoms based on our work with the verbatim utterances, ASR errors on the non-verbatim validation data prompted further changes. For example, the team found that non-verbatim accuracy was improved by encoding "RCO" as both its individual letters "r c o" and spelling it phonetically as "arceo."

AUDIO DATA COLLECTION

In this section, we describe our process for collecting recordings of verbatim and non-verbatim utterances. A total of 13 speakers participated in the recording process. This number as well as the number of recordings obtained were dictated by availability of speakers during the limited timeframe we had to obtain recordings.

Verbatim Utterances

The team collected multiple audio recordings for each of the 1,650 unique verbatim utterances, all spoken aloud with the goal of serving as correct examples. After a qualitative pass through the audio recordings to remove any with mispronounced or unclear words, there remained a total of 3,310 correct audio recordings contributed by 10 speakers.

The team also collected audio recordings for intentionally incorrect examples. To produce these examples, each of the 1,650 verbatim utterances was processed in one of three ways:

1. **Deletion**: a random word or phrase was removed from the utterance
2. **Substitution**: a random word or phrase was replaced by a different word or phrase extracted from the vocabulary of all verbatim utterances.
3. **Insertion**: A random word or phrase from the vocabulary of all verbatim utterances was inserted at a random position in the utterance.

The team then collected audio recordings for the incorrect examples. After a qualitative pass to make sure no audio recording was accidentally correct, there remained a total of 1,646 incorrect audio recordings contributed by 3 speakers.

Non-verbatim Utterances

Given the variations possible in non-verbatim utterances based on the government rules stipulating how non-verbatim utterances were to be evaluated, the team implemented three variations of the audio recordings for non-verbatim recognition task:

- **Non-verbatim Correct, expected communications:** The first type was recordings of the expected communication for each utterance. More precisely, each utterance was a set of specific required words and phrases beginning with a salutation, spoken in the order targeted in the training documentation.
- **Non-verbatim Correct, randomized:** The second type were random combinations of the required keywords intermixed with extra words. These contained a set of specific required words and phrases beginning with a

salutation, but which are spoken in a randomized order. To generate the text for these utterances, a keyword was randomly selected from each required keyword slot. These selections were then shuffled into a random order. Between each keyword, 0, 1, or 2 extra words were added. The extra words were sampled from the vocabulary of the verbatim utterances and the non-verbatim expected communications. Note that according to the rules of how non-verbatim utterances were to be scored, all these recordings of the second type were intended to serve as correct or acceptable variations of the targeted non-verbatim utterance.

- **Non-verbatim Incorrect, randomized:** The third type were incorrect versions of the *non-verbatim correct* utterances. These included a set of specific required words and phrases beginning with a salutation, spoken in a randomized order, but from which one or more atoms included in the targeted utterance are omitted. These were generated by randomly excluding exactly one of the required keyword slots and verifying that any extra words added were not a match to any of the keywords in that slot.

The scripts for these recordings were distributed across 9 speakers. Of these speakers, 7 speakers contributed a total of 1,632 expected communications recordings, 8 speakers contributed a total of 2,208 randomized correct recordings, and 4 speakers contributed a total of 1,301 randomized incorrect recordings.

Data Collection Challenges

One of the biggest challenges involved in this effort was obtaining satisfactory datasets for evaluation. While the team was able to include more than one single human-spoken accent for every verbatim and non-verbatim utterance, our final per-utterance sample size is smaller than we would ideally have liked. We were limited both by time and by availability of speakers to record utterances. Availability of a larger quantity of quality human-spoken data would have been desirable, particularly data captured using the specific microphone and headset intended for use in the operational environment and including a wider variety of accents. The team's internally produced human-spoken recordings were recorded in reasonably controlled acoustic environments. This allowed for our speakers to use whatever microphone they had on hand regardless of whether it had noise cancelling capabilities or not. This enabled us to produce and process a significant quantity of recordings from a variety of speakers in a short amount of time but may not be a perfect reflection of the recording quality yielded in the operational environment.

BUILDING A TEST HARNESS/GRADER

As we began iterating on the intent model and collecting audio data, we needed a way to easily test different versions of intent models on a variety of validation datasets, with the ability to vary parameters of the ASR system as needed. To meet these requirements, we developed a configurable test harness for configuring and managing the various models and model parameters as well as processing voice data to produce graded results.

To combat the somewhat malleable nature of potential recognition validation requirements, we associated each utterance with a specific *validation configuration*. In the process of validating voice files, the validation configuration for the associated utterance is referenced for more precise requirements of what constitutes a passing score on the recognition of that utterance. This approach yields flexibility in terms of tailoring recognition requirements on an utterance-by-utterance basis as some utterances can be more challenging to recognize than others.

Algorithm for Grading Verbatim Utterances

Verbatim means word for word. Our system follows this guideline when grading. For all the recordings intended to represent correct utterances of verbatim statements, we allowed for no tolerance for either redundant word or missing slots. It was necessary, however, to allow for multiple configurations of a single validation configuration to be defined as "passing." The two options defined for this purpose were "redundant words" and "missing slots". Using these configurations, the grading system followed the following rules:

1. For each extra word the **redundant** counter will get incremented. Afterwards any excess words will get marked into the **missing slot** section.
2. For each missing word the **missing slot** counter will get incremented. Once the counter reaches its maximum, the test has failed.
3. Reorder of a word will result in incrementation of the **missing slot** counter. Once the counter reaches the maximum value, the test will be failed.

Algorithm for Grading Non-Verbatim Utterances

Multiple iterations were conducted on the validation process while analyzing and grading the non-verbatim output. From the beginning, the structure of the validation process had to be somewhat reconfigurable since multiple unique responses could be considered consistent with the utterance requirements. We therefore needed a new mechanism to grade and validate the utterances in question.

During this process, we introduced two new algorithms to recognize non-verbatim utterances. The first was a heuristic comparison algorithm which compared an utterance segment to a potential slot. The second algorithm is native to the test tool and allows for validation checks over a hierarchical web of slots. Iterating between these two approaches in parsing yielded high accuracy levels grading non-verbatim utterances.

One of the biggest issues in parsing and grading non-verbatim utterances had to do with collision of slots recognized within a single non-verbatim utterance recording. The underlying atoms of a slot frequently contained similar phrases to atoms in another slot present in the same utterance, which would confound the grading process unless controlled for. To mitigate this problem, we developed a procedure to select the correct atom for each slot: a comprehensive selective parser that evaluated the anticipated permutations – themselves a subset of the list of all possible permutations – of atom appearance within a given non-verbatim utterance recording. This helped avoid possible confusion of the identification of a given sound with multiple slots targeted throughout a single utterance.

The solution implemented allowed for different permutations of the same utterance keywords while also retaining the high success rate otherwise yielded by the system for both correct and incorrect utterances.

HYPERPARAMETER TUNING

A *hyperparameter* is a configuration variable used by a machine learning model; in the instance reported here, hyperparameters helped the models used balance the relative contributions of different solution sources and to vary the precision attached to certainty estimates attached to matched utterances. *Hyperparameter tuning* is a method by which hyperparameters are selected and tested on actual data to achieve optimal model results. To facilitate hyperparameter tuning (Liu & Wang, 2021), a particle-based optimizer was employed. *Particle-based optimization* is a computational method to find optimum solutions to a problem, inspired by swarm behavior observed in nature. Candidate solutions, or *particles*, are iteratively guided around in the search-space toward best known positions influenced by the entire swarm. Tuning was conducted primarily on the datasets representing correct utterances and was conducted separately for verbatim and non-verbatim data. Performance improvement due to tuning rarely improved the baseline by more than ~1% recognition accuracy. However, recognition accuracy did vary quite widely across the hyperparameter space overall, with poor choices of hyperparameter values yielding drastically worse results.

Hyperparameters of interest for tuning included the Alpha and Beta values of the beam search decoder of the underlying ASR system. Alpha controls the importance weighting of the acoustic model versus the language model, and Beta provides a conciseness weighting. As such, differences in Alpha and Beta had wider implications on development of the intent model. Changes in the intent model often required additional hyperparameter tuning and improved tuning often provided more control and additional room for refinement within the intent model.

DEFINITION OF TESTING CONDITIONS

The following is the working set of test conditions used for evaluation of the test harness' recognition accuracy.

Pronunciation Requirements for usable utterance recordings

- Cadence and enunciation reflecting some minimum and human-detectable pauses between words.
- Human-perceptible breaks in speaking at utterance points containing commas or periods.
- Human-perceptible distinction between syllables
- Recording must be understandable by a human when played at normal volume.

Usable utterance recordings do include those created by speakers with different accents, of different genders, of different ages, and of different ethnicities, so long as those speakers meet the criteria listed above.

Test conditions for measuring accuracy

The four possible outcomes of an utterance test are as follows:

- **True Positive (TP)**: Correctly spoken utterance that is consistent with pronunciation requirements and audible requirements that matches its targeted utterance attempt and is graded as an utterance match during the test event.
- **True Negative (TN)**: Utterance that meets all pronunciation and audibility requirements, but that does not match the entry it is intended to represent from the verbatim or non-verbatim list and is graded as a non-match during the test event.
- **False Positive (FP)**: An utterance that does not match its targeted utterance attempt, but which is graded as an utterance match during the test event.
- **False Negative (FN)**: An utterance that does match its targeted utterance attempt, but which is not scored as an utterance match during the test event.

These were all assessed from among a set of utterance recordings that accurately reflect the words, numbers, and letters in specific verbatim and non-verbatim utterance attempts by a speaker that are consistent with pronunciation requirements and audible requirements. This means that utterances that were mispronounced, slurred, spoken too rapidly, or for which audio levels were too low were NOT counted as part of any of these four groups (TP, TN, FP, FN) for purposes of calculating recognition accuracy. Many such excluded utterances could readily have been treated as TN utterances but were not included in these analyses since the specific deficiencies in these flawed utterances were not quantified.

Recognition rate requirement

Recognition rate is defined as the proportion of utterance cases included in a test that are correctly identified as TP or TN. A requirement to achieve a targeted recognition rate is defined as the correct identification of some proportion of the utterances included in a test as TP or TN. In addition, it is worth noting that recognition rate as used here is not used to define accuracy of recognition of atoms or words within a single utterance. The authors are not evaluating whether the tool correctly recognizes 97% of the contents of any specific utterance in any multi-utterance test, but rather the proportion of utterances correctly identified as matches to a targeted validation configuration.

RESULTS

We present results for each type of recording for two baseline intent models and our final intent model, consistently implemented as follows. The first baseline intent model included no alternative pronunciations or spellings for words and phrases. The second baseline intent model included a single alternative spelling, namely, allowing *i* in place of the word *aye*. The final intent model introduced includes all alternative pronunciations and spellings.

First, we present recognition rate results for the baseline models and the final model. All results are presented as both counts and recognition rates; that is, the proportion of cases for which our grading system made the correct classification of the utterance in question. For correct examples, we present the proportion of true positives, and for incorrect examples we present the proportion of true negatives.

Next, we present word error rate (WER) and slot error rate (SER) results for our final model on each of our datasets. WER is a representation of the number of recognition errors in an utterance (i.e., words or statement(s)) divided by the total number of words. These errors can include substitution – recognition of an intended word as a different word not intended by the speaker, insertion – addition of a word to a recognized utterance that was not intended by the speaker, or deletion – the omission of an intended word from a system-detected utterance. WER rate is a commonly used metric for evaluating speech recognition systems and fits well within the context of verbatim recognition. Since non-verbatim recognition allows extra words, we define the SER metric similarly to WER but excluding extra words (insertions) when counting errors. As such, errors in the context of SER primarily count missing slots.

Verbatim recognition rate results

Table 1(a) shows the results of analysis of the 3,310 correct verbatim recordings, Table 1(b) shows the results of analysis of the 1,646 incorrect verbatim recordings, and Table 1(c) shows combined results over all verbatim datasets. The final model achieved an overall 97.5% recognition rate on the verbatim datasets.

Table 1(a): Results on verbatim correct recordings

Intent model	True Positives	False Negatives	Recognition Rate (%)
Baseline 1	1702	1608	51.4199
Baseline 2	2979	331	90.0000
Final model	3227	83	97.4924

Table 1(b): Results on verbatim incorrect recordings

Intent model	False Positives	True Negatives	Recognition Rate (%)
Baseline 1	9	1637	99.4532
Baseline 2	28	1618	98.2989
Final model	41	1605	97.5091

Table 1(c): Overall recognition rate averaged across both verbatim datasets

Intent model	Recognition Rate (%)
Baseline 1	67.3728
Baseline 2	92.7562
Final model	97.4979

Non-verbatim recognition rate results

Table 1(a) shows the results of analysis of the 1,632 expected communications recordings, Table 1(b) shows results for the 2,208 correct randomized recordings, Table 1(c) presents results of analysis of the 1,301 incorrect randomized recordings, and Table 1(d) shows combined results over all non-verbatim datasets. Our final model achieved an overall 97.9% recognition rate on the non-verbatim datasets.

Table 2(a): Results on non-verbatim (correct) expected communications recordings

Intent model	True Positives	False Negatives	Recognition Rate (%)
Baseline 1	1465	167	89.7672
Baseline 2	1464	168	89.7059
Final model	1602	30	98.1618

Table 2(b): Results on non-verbatim correct randomized recordings

Intent model	True Positives	False Negatives	Recognition Rate (%)
Baseline 1	1903	305	86.1866
Baseline 2	1905	303	86.2772
Final model	2143	65	97.0562

Table 2(c): Results on non-verbatim incorrect randomized recordings

Intent model	False Positives	True Negatives	Recognition Rate (%)
Baseline 1	9	1292	99.3082
Baseline 2	10	1291	99.2314
Final model	11	1290	99.1545

Table 2(d): Overall recognition rate averaged across all non-verbatim datasets

Intent model	Recognition Rate (%)
Baseline 1	90.6438
Baseline 2	90.6438
Final model	97.9381

Word error rate (WER) and slot error rate (SER) results

Table 3 shows the WER and SER results for each of our datasets, using our final intent model.

Table 3: WER and SER results

Dataset	Result (%)
Verbatim Correct (WER)	0.1624%
Non-verbatim Expect Comms (SER)	0.1567%
Non-verbatim Randomized Correct (SER)	0.2782%

Summary

Results described in the sections above clearly indicate the team exceeded its target recognition rate of 97% for both verbatim (97.5%) and non-verbatim (97.9%) utterances. Word error rates on this task are also significantly lower than reported rates for state-of-the-art ASR models. In a paper that compares existing end-to-end ASR systems, word error rates range from 1.4% to 16.8% (Li, 2022). OpenAI's best Whisper model achieves a WER of 2.5% (Radford, et al., 2023). On our verbatim data set, our final intent model yielded a WER of 0.16%.

The inclusion of incorrect utterances in these analyses will also serve to ensure the generalizability of our results effectiveness in an operational setting. The inclusion of both expected communication patterns and randomized presentation of non-verbatim utterances in correct and incorrect use cases should also strengthen the generalizability of these recognition levels.

We found that accuracy was lower on the randomized utterances than on the expected communications. One possible explanation for this is that the underlying ASR model was trained on grammatically spoken English and the randomized utterances did not follow this pattern. Another possible explanation is that the randomized utterances tended to be somewhat longer due to the number of extra words being inserted.

CONCLUSION

This paper introduced and described an ASR-based grading system designed for Navy radio operator training. We discussed the speech recognition system, including our approach for constructing custom language models for the task and hyperparameter tuning. We described our process for collecting a high-quality set of recordings to use as a dataset and our methodology for grading and testing each utterance. Finally, we presented results illustrating that this approach not only met the 97% recognition accuracy requirements for this task but are also highly competitive with state-of-the-art ASR systems.

As mentioned in the introduction, our focus was specifically on developing the grading component and we therefore do not provide details about the larger training system in which this is embedded, except to say that the system in question did undergo joint integration testing with the government and met all requirements. These results show that a finely tuned and trained speech recognition and parsing model can work extremely well for the use case of radio operator training. We expect that this approach would generalize well to other speech-based training systems.

ACKNOWLEDGEMENTS

This material was based upon work contracted by Cubic Defense Applications, Inc. to Soar Technology, LLC under subcontract 12-1872, and performed by Cubic Defense Applications, Inc. under contract N61340-20-F-0149. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the US Navy.

REFERENCES

Bird, S., Loper, E., & Klein, E. (2009). *Natural Language Processing with Python.* O'Reilly Media Inc.

Kriman, S., Beliaev, S., Ginsburg, B., Huang, J., Kuchaiev, O., Lavrukhin, V., . . . Zhang, Y. (2020). Quartznet: Deep Automatic Speech Recognition with 1D Time-Channel Separable Convolutions. *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* Barcelona, Spain.

Lebanoff, L., Newton, C., Hung, V., Atkinson, B., Killilea, J., & Liu, F. (2021). Semantic Parsing of Brief and Multi-Intent Natural Language Utterances. *Proceedings of the Second Workshop on Domain Adaptation for NLP*, (pp. 255–262).

Li, J. (2022). Recent advances in end-to-end automatic speech recognition. *APSIPA Transactions on Signal and Information Processing, 11*(1).

Liu, X., & Wang, C. (2021). An Empirical Study on Hyperparameter Optimization for Fine-Tuning Pre-trained Language Models. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (pp. 2286–2300). Online: Association for Computational Linguistics.

Noyes, J. M., & Haas, E. (2010). Military Applications: Human Factors Aspects of Speech-Based Systems. In F. J. Chen, *Speech Technology: Theory and Applications* (pp. 251–270). New York, NY: Springer.

Radford, A., Kim, J. W., Xu, T., Brockman, G., Mcleavey, C., & Sutskever, I. (2023). Robust Speech Recognition via Large-Scale Weak Supervision. *Proceedings of the 40th International Conference on Machine Learning*, (pp. 28492--28518).

Stensrud, B., Taylor, G., & Crossman, J. (2006). IF-Soar: A Virtual, Speech-Enabled Agent for Indirect Fire Training. *Proceedings of 25th Army Science Conference.* Orlando, FL.