A Real-Time Speech Command Detector for a Smart Control Room

Daniel Reich\textsuperscript{1}, Felix Putze\textsuperscript{1}, Dominic Heger\textsuperscript{1}, Joris Ijsselmuiden\textsuperscript{2}, Rainer Stiefelhagen\textsuperscript{2}, Tanja Schultz\textsuperscript{1}

\textsuperscript{1}Cognitive Systems Laboratory, Karlsruhe Institute of Technology, Germany
\textsuperscript{2}Fraunhofer Institute of Optronics, System Technologies and Image Exploitation, Germany

daniel.reich@student.kit.edu
{felix.putze, dominic.heger, tanja.schultz}@kit.edu
{joris.ijsselmuiden, rainer.stiefelhagen}@iosb.fraunhofer.de

Abstract

In this work we present an always-on speech recognition system that discriminates spoken commands directed to the system from other spoken input. For discrimination we integrated various features ranging from prosodic cues and decoding features to linguistic information. The resulting “Speech Command Detector” provides intuitive hands-free user interaction in a Smart Control Room environment where voice commands are directed toward a large interactive display. Based on a recognition vocabulary of 259 words with more than 10k possible commands, the Speech Command Detector detected 88.3\% of the commands correctly maintaining a very low False Positive Rate of 1.5\%. In a cross-domain setup the system was evaluated on a Star Trek episode. With only minor adjustments, our system achieved very promising results with 91.2\% command detection rate at a False Positive Rate of 1.8\%.

Index Terms: always-on spoken command detection, smart environment, prosodic and confidence-based features

1. Introduction

For a natural and intuitive experience when interacting with computer systems, it is vital to provide speech interfaces which are trigger-free, always on and can be operated in uncontrolled and dynamic environments. Such systems include in-car voice applications, smart environments or human-robot dialog systems to name only a few. For this purpose, a speech recognition system is needed that is constantly online and able to process user queries anytime. Consequently, discrimination between irrelevant speech segments and actual user requests is necessary. Irrelevant segments can contain ambient noise, monologs or dialogues with other people that might falsely be recognized as commands.

For related work, a lexicalized filler-model for key-phrase verification was proposed by Kawahara et al. in [1]. A filler model is created based on word sequences that are most common for a particular speaking style. Examined hypotheses are verified by comparing scores of the hypothesis and the best sequence of filler phrases contained in the filler model. Very good results were achieved in a presentation task where a speaker operates a slide projector by speech. Usable vocabulary contained 56 words.

In a smart home environment setup, where at least two people were present, Obuchi et al. tried to discriminate conversation and other common home environmental noise from 17 command phrases for controlling a TV Set using several prosody and recognition confidence-based scores [2].

In the area of human-robot interaction, Yamagata et al. [3] studied a scenario where humans have conversations with each other and occasionally make system requests to a mobile robot. This mobile robot was able to understand 18 commands. Acoustic features calculated from head, tail and main part of an utterance were used to detect system requests. In later works using the same setup, Yamagata et al. proposed a multi-resultion analysis using one-dimensional Gabor wavelet transform to examine a log-scale Mel-frequency filter-bank feature that is calculated on the input waveform [4]. By using only acoustic features, evaluation reached a 0.926 F-score.

In a similar scenario involving two people doing conversation and sporadically giving commands to a robot, Katzenmaier et al. [5] combined speech-based features with head-pose estimation to determine the addressee of an utterance.

Furthermore, we consider research concerning prosody in speech type and dialog act classification. A large-scale prosody feature evaluation was done in [6] for the task of dialog act classification. Several acoustic features were investigated for their efficiency in the task of classifying clear and conversational speech of linguistically identical utterances in [7].

The Speech Command Recognition System that we present in this paper was evaluated in the Smart Control Room for crisis response and management at the Fraunhofer IOSB in Karlsruhe [8]. Various input modalities work side by side to create a natural way of interacting with a large video wall. Among other methods, gesture recognition, head pose estimation and speech recognition are used to create a perceptive user interface that makes system control on the video wall highly intuitive. Usable commands contain sentences like “show street map”, “send 2 ambulances to section 4” and “center operations area 3”.

For this application, we developed a classification system for system requests which 1) combines prosody and decoding-based features for robust classification, 2) is integrated in an online ASR system and does feature extraction and classification in real-time, 3) is able to distinguish between conversational sentences and commands that share the same terminology, 4) can handle a high number of commands, and 5) works user-independently.

2. System Layout

We segment the continuous audio stream input using an energy and zero-crossing-based segmentation algorithm. These audio segments are input to our speech recognizer which is based on the system that was used in [9]. It was built with the JRTk [10]. The recognizer’s acoustic model (AM) uses fully continuous
HMMs and consists of about 3200 context dependent models with 48 Gaussians each. MFCC feature dimensionality is 32. We use a context-free grammar as language model (LM).

Dictionary and grammar were constructed from used commands in our data. The grammar was designed to be able to accept a variety of logical variations of the used commands. It consists of 584 nodes, 905 arcs, 18 rules, has a vocabulary of 259 words and accepts more than 10k different commands.

For out-of-vocabulary (OOV) detection, we use generalized word models [11] which can replace every non-terminal in the grammar during decoding.

We use a second LM in parallel to the grammar. This trigram LM was trained on a large German dialog corpus and interpolated with various news texts concerning crisis scenarios like floods, fires and earthquakes.

After standard ASR on an audio segment, the segment is classified as either command or non-command as follows.

1. Check ASR output hypothesis for parseability and OOVs: First, we check whether or not a given hypothesis can be fully produced by our grammar or contains OOV-words. The hypothesis is rejected, if it is incomplete or contains OOVs, as clean and complete grammar sentences are required for further processing.

2. Calculate and classify feature vector: We then calculate a feature vector for the segment based on the raw audio file and the decoding result. This vector is classified based on the maximum likelihood method using two multivariate normal distributions, one for commands and one for non-commands. Parameters for the distributions are estimated from training data.

3. Send hypothesis to Smart Control Room modules: Only ASR outputs that have been classified as commands are sent to the Smart Control Room for further processing, such as fusion with other modalities and generation of scripted actions on the screen.

3. Collected Data

Recordings for training and evaluation were collected in the same Smart Control Room that the system runs in now. All data was recorded using lapel microphones. In each session, two participants were given the task to act as operators-in-charge of a crisis response scenario involving a flood threatening a German town. The video wall showed a town map and a table of available units and materials. We asked the participants to discuss a course of action in this fictive crisis scenario and interact with the video wall by using gestures and speech. The system could not react to input as this was a dry-run simulation. We also did not hand out a list of possible input commands so that participants were free to use whatever commands came to their mind. Since participants were consulting each other about their course of action, conversations shared domain and terminology of the commands. To investigate the effect of segmentation errors, we studied both automatically and manually segmented data.

We recorded two data sets. Data Set 1 (DS1) consists of four sessions and is about 27 minutes long for each of the two male speakers. Data Set 2 (DS2) consists of three sessions recorded by two different male speakers and is about 14 minutes long for each speaker. The ratio of commands to non-commands for DS1 is 60/333 for manually segmented data and 49/782 for automatically segmented data. For DS2 we have a ratio of 73/281 and 61/463 respectively.

Recordings in DS2 differ strongly from DS1 in terms of speaking styles (hectic vs calm), vocabulary size for commands (259 vs 56), commands (large variations vs many similar commands) and are in general much more natural and spontaneous. This makes the design of a system which works well for both data sets a challenging task.

4. Feature Description

We group features into two categories, decoding features and prosody features. Some of the features (e.g. acoustic scores of senones in each frame) consist of multiple values for one segment. First, each of these values is classified as belonging to a command or non-command speech act by a preliminary maximum likelihood classification. Then, the final feature value for the segment is determined as the resulting percentage of values that have been classified as command.

4.1. Decoding Features

Decoding features are computed based on ASR-output, i.e. generated transcription, lattice or scores. They are motivated by the fact that decoding of speech segments that contain in-grammar commands yields more robust recognition results than decoding attempts for other sound segments. Thus, the features can be used as confidence measures for the recognized hypothesis.

The acoustic score of each frame, i.e. the senone scores of the best hypothesis, are used as a feature. Extracted values are normalized by session-wide mean and standard deviation (SD) of the senone scores. Also, we calculate the range within the senone scores of an utterance.

We use the number of backpointers (BP) of a decoded speech segment to measure how well the preprocessed features of a speech segment fit into the GMMs of the acoustic HMMs. The total BP count for a hypothesis is normalized by the duration, i.e. the number of frames of the speech segment.

We count the number of all lattice word-nodes traversed during decoding and normalize by duration. In-grammar hypotheses usually have a considerably smaller number of nodes in their corresponding lattice as they better match the AM.

We use the Levenshtein distance between the hypotheses of the grammar-LM-based and the trigram-LM-based ASR at phoneme level as a feature.

We use filler words to handle hesitations and non-verbal noise that may occur in spontaneous speech. They may occur in the hypothesis between any two words. We use the total filler count within an utterance normalized by duration and the ratio of filler words to total duration within an utterance as features.

Phoneme count normalized by total frame count of the utterance is used as an approximation of speaking rate.

4.2. Prosody Features

We asked a Japanese person who had almost no knowledge of the German language to detect commands in our German data by listening to DS2. He was able to detect about 75% of the commands with a single false alarm. This indicates that there are some prosodic characteristics that make commands distinguishable from normal conversation between humans. Features that try to capture prosodic characteristics of speech are listed in this category. Their calculation is based on the raw sound file.

Fundamental Frequency Variation (FFV) features [12] consist of 7 continuous coefficients and represent instantaneous change in fundamental frequency at frame-level. The coefficients are computed for each frame in the speech segment.
Since it is not captured by FFV, we also use absolute fundamental frequency (F0) as a feature. Our features are: All F0 frame values, F0 range, F0 mean and F0 SD. F0 frame values, F0 range, mean, and SD of the first and last 15 frames of a sound segment. Features are z-normalized in each utterance based on session-wide mean and SD. When considering only the head and tail part of an utterance, we perform z-normalization based on the examined utterance’s F0 mean and SD.

Only voiced frames are used for Root-Mean-Squared (RMS) Energy features and RMS is z-normalized over the whole session for each speaker. The following features are examined: All RMS frame values, RMS range, RMS mean and RMS SD. RMS frame values, RMS range, mean and SD of the first and last 15 frames of a sound segment.

Furthermore, we count RMS peaks of an utterance that have a minimum height of half the SD of the session’s RMS above the session’s mean RMS. This results in two features: Total number of peaks found in a segment normalized by (voiced) frame count and width of each gap between peaks.

5. Evaluation

For evaluation, we first investigated each feature separately on DS1. Then, we selected a set of the most promising features, trained a complete system on DS1 and evaluated it on DS2.

All sound segments were randomly distributed into four evenly partitioned sets on which we evaluated each feature using cross-validation, resulting in 12 evaluation runs for each feature. We performed all evaluations with OOV and parseability checks as a preprocessing step. This step reduced the number of non-command segments but not the number of commands.

We used 4 evaluation setups: Manually and automatically segmented data was each evaluated using an adapted and unadapted AM for ASR. Adaptation of the AM was performed using MLLR based on command utterances from 3 of the 4 available sessions for each speaker. When using the adapted AM, decoding features were extracted from the omitted session. Prosody features remained unaffected by adaptation since they are calculated on the raw sound file. F-score means over all 12 evaluation runs for all features were calculated.

Classification results for prosody features declined drastically from manually to automatically segmented data. Mean F-scores, taken over all results for prosody features, fell from 0.69 to 0.44, both with a SD of about 0.17. Further examination revealed that the biggest difference between the resulting segments lies in their varying beginnings and ends. Manually segmented data was usually cut cleanly at the boundaries of speech parts. The segments from automatic processing on the other hand had sloppier cuts and thus longer heads and tails surrounding the main part of a sound segment. These parts of an utterance seem to account for most of the information that can make prosody features useful. However, this observation does not apply to RMS-peak-based features as there were no RMS peaks at the beginning and end of the segments.

From the decoding features, lattice and BP features proved particularly useful. Most of the decoding features were not affected as strongly by the different segmentation types (mean F-score for manual vs automatic segmentation: 0.82 vs 0.68) as prosody features and delivered better results when working with adapted AM (mean F-score for unadapted vs adapted AM: 0.82 vs 0.85 and 0.68 vs 0.70 respectively).

To find the best subset of features for our classifier, we used sequential Forward Feature Selection (FFS) to find feature subsets within each category and from all features pooled together, based on DS1. The features from both categories which were picked most often were number of lattice nodes, RMS peaks and normalized phoneme count. Using those feature sets, we achieved a mean F-score between 0.89 and 0.99 on the development data (see last column of Table 1). All obtained feature sets from FFS on DS1 were evaluated on DS2. Evaluation was performed running our whole ASR system under quasi-online conditions with audio streamed from a recorded file.

We found a significant difference between the models of commands and non-commands when analyzing the spectral norm of their co-variance matrices. The spectral norm ratio of the classes of the decoding feature set in the automatic segmentation / adapted AM setup, for instance, was roughly 1.5 (commands vs non-commands). For this reason, we introduced a tuning parameter Q as a threshold when comparing the output of each Gaussian given a feature vector. If the ratio between the outputs of the Gaussian for the command class and the one for the non-command class is larger than Q, the utterance will be labeled as command (cf. normalization using background model scores in speaker verification). The threshold Q was determined for each system setting from a small development set. For our evaluations, we set Q according to the threshold that has the highest Recall while maintaining a False Positive Rate (FPR) of at most 2% in each evaluation setting and feature-set type. We consider a low FPR to be more important than Recall given the application in crisis scenarios. FPR, Precision, Recall and F-scores for all four setups, each evaluated with a classifier built from subsets of prosody and decoding features and all features combined, are listed in Table 1.

5.1. Results Discussion

As expected, decoding feature sets performed particularly well in setups with adapted AM. But even in the two unadapted setups, they did fairly well, never falling below a F-score of 0.8.

Prosody-based features worked well on DS1 (see last column of Table 1) but results declined for DS2. This suggests that these features are strongly dependent on speaking style and speaker identity and do not generalize as good as the decoding features. When analyzing the classification results from the prosody feature sets, we found a small set of misclassified noncommands which consisted mostly of short, non-verbal sound segments. Since these segments strongly resemble commands in their prosodic properties, correct classification is not feasible using that feature set and a FPR of at most 2% results in a low Recall, which is shown in the ROC curve in Fig. 1. Also, even though we used normalized F0 features throughout these experiments, they are still more person-dependent than RMS features and could be another reason for the poor performance on DS2. For these two reasons, we constructed a new feature set built only from RMS-based features and excluded F0 features from FS All. To illustrate the full potential of the prosody features, we also list their scores for a slightly higher FPR of 5%.

In the most relevant setup to our application (automatic segmentation / adapted AM), our system found 88.3% of all commands while maintaining a low FPR of 1.5% using all features.

5.2. The Star Trek Experiment

We used episode 131 of Star Trek - The Next Generation to test our system for cross-domain portability. In Star Trek, any crew member can verbally communicate with the computer system at any time. In some of the scenes, several crew members take turns in making system requests while also talking to each other.

For this experiment, we only made some minor adjustments...
to cope with the special challenges that an audio stream from a TV show comes with: We adjusted the segmentation algorithm to be able to segment the audio stream despite the background music and sound effects and used a different threshold Q to fit this new scenario. Also, we added all commands that appeared in the episode to our existing grammar, which added another 92 words to the vocabulary. We used the setup of automatic segmentation with unadapted AM and used the feature set created from all features, since it had delivered the best results in that setup. Segments of the 44 minutes long audio track were divided into 34 commands spoken by 7 different actors and 552 non-commands. Results show that the system performs admirably even under these harsh conditions. For a FPR of 0.018, the system reached a Recall of 0.912 and a Precision of 0.756, resulting in a total F-score of 0.827.

Table 1: False Positive Rate (FPR), Precision, Recall and F-score of the evaluation on DS2 for different feature sets. For comparison, the last column contains the achieved F-score on the development set of data set DS1. FS Prosody* does not contain F0 features and is allowed a FPR of 5%.

<table>
<thead>
<tr>
<th>Feature Set</th>
<th>FPR</th>
<th>Prec.</th>
<th>Recall</th>
<th>F</th>
<th>DS1 F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual Segmentation, Unadapted AM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FS Prosody</td>
<td>0.018</td>
<td>0.688</td>
<td>0.162</td>
<td>0.262</td>
<td>0.915</td>
</tr>
<tr>
<td>FS Prosody*</td>
<td>0.049</td>
<td>0.769</td>
<td>0.735</td>
<td>0.752</td>
<td>0.889</td>
</tr>
<tr>
<td>FS Decoding</td>
<td>0.011</td>
<td>0.947</td>
<td>0.794</td>
<td>0.864</td>
<td>0.960</td>
</tr>
<tr>
<td>FS All</td>
<td>0.011</td>
<td>0.947</td>
<td>0.794</td>
<td>0.864</td>
<td>0.960</td>
</tr>
<tr>
<td>Manual Segmentation, Adapted AM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FS Prosody</td>
<td>0.018</td>
<td>0.844</td>
<td>0.370</td>
<td>0.515</td>
<td>0.934</td>
</tr>
<tr>
<td>FS Prosody*</td>
<td>0.046</td>
<td>0.824</td>
<td>0.836</td>
<td>0.830</td>
<td>0.896</td>
</tr>
<tr>
<td>FS Decoding</td>
<td>0.007</td>
<td>0.970</td>
<td>0.877</td>
<td>0.921</td>
<td>0.980</td>
</tr>
<tr>
<td>FS All</td>
<td>0.007</td>
<td>0.970</td>
<td>0.849</td>
<td>0.904</td>
<td>0.983</td>
</tr>
<tr>
<td>Automatic Segmentation, Unadapted AM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FS Prosody</td>
<td>0.019</td>
<td>0.775</td>
<td>0.574</td>
<td>0.660</td>
<td>0.892</td>
</tr>
<tr>
<td>FS Prosody*</td>
<td>0.047</td>
<td>0.646</td>
<td>0.778</td>
<td>0.706</td>
<td>0.892</td>
</tr>
<tr>
<td>FS Decoding</td>
<td>0.013</td>
<td>0.875</td>
<td>0.778</td>
<td>0.824</td>
<td>0.957</td>
</tr>
<tr>
<td>FS All</td>
<td>0.011</td>
<td>0.898</td>
<td>0.772</td>
<td>0.830</td>
<td>0.971</td>
</tr>
<tr>
<td>Automatic Segmentation, Adapted AM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FS Prosody</td>
<td>0.019</td>
<td>0.786</td>
<td>0.550</td>
<td>0.647</td>
<td>0.928</td>
</tr>
<tr>
<td>FS Prosody*</td>
<td>0.047</td>
<td>0.667</td>
<td>0.750</td>
<td>0.706</td>
<td>0.928</td>
</tr>
<tr>
<td>FS Decoding</td>
<td>0.015</td>
<td>0.883</td>
<td>0.883</td>
<td>0.883</td>
<td>0.989</td>
</tr>
<tr>
<td>FS All</td>
<td>0.015</td>
<td>0.883</td>
<td>0.883</td>
<td>0.883</td>
<td>0.989</td>
</tr>
</tbody>
</table>

6. Conclusions

In this work we presented an online ASR system with a built-in speaker-independent classifier that discriminates between commands directed to an operable screen and other speech and sound segments. For development of our classifier, we examined two kinds of features, prosody and decoding features. While prosody features develop most of their use for a slightly higher allowable FPR, decoding features more than compensated for this and made a practical system with a FPR of lower than 2% possible. While maintaining a low FPR of 1.5%, our classifier was able to detect 88.3% of all commands on our test data for the most important setting of an online system that uses automatic segmentation and adapted AM.

Cross-domain portability was proven by applying our system to the task of command recognition for an episode of Star Trek. The task was well accomplished with an F-score of 0.827.

7. References