Learning When to Listen: Detecting System-Addressed Speech in Human-Human-Computer Dialog

Elizabeth Shriberg, Andreas Stolcke, Dilek Hakkani-Tür, Larry Heck

Microsoft, Mountain View, CA, U.S.A.

{elshribe,anstolck,dilekha,lheck}@microsoft.com

Abstract

New challenges arise for addressee detection when multiple people interact jointly with a spoken dialog system using unconstrained natural language. We study the problem of discriminating computer-directed from human-directed speech in a new corpus of human-human-computer (H-H-C) dialog, using lexical and prosodic features. The prosodic features use no word, context, or speaker information. Results with 19% WER speech recognition show improvements from lexical features (EER=23.1%) to prosodic features (EER=12.6%) to a combined model (EER=11.1%). Prosodic features also provide a 35% error reduction over a lexical model using true words (EER from 10.2% to 6.7%). Modeling energy contours with GMMs provides a particularly good prosodic model. While lexical models perform well for commands, they confuse free-form system-directed speech with human-human speech. Prosodic models dramatically reduce these confusions, implying that users change speaking style as they shift addressees (computer versus human) within a session. Overall results provide strong support for combining simple acoustic-prosodic models with lexical models to detect speaking style differences for this task.

Index Terms: addressee detection, spoken dialog system, prosody, language model, GMM, boosting, logistic regression.

1. Introduction

Dialog systems are continually evolving to handle less constrained spoken input, interpret user intent, and engage in natural dialog to accomplish complex tasks. A fundamental capability for spoken dialog systems, especially those relying heavily on speech input, is *addressee detection*--the ability to detect whether or not user speech is directed toward the system. In single-user human-computer (H-C) contexts, the alternate addressee may be the user him- or herself (self-talk), or others in the environment who are not interacting with the system.

When multiple users interact jointly with a system, which we will refer to as H-H-C dialog, addressee detection becomes even more of a challenge. Human-human (H-H) conversation about the shared task can contain the same keywords a system would listen for. And when system-addressed utterances contain more than only commands or keywords, word sequences can begin to look more like those in H-H speech. Even other cues such as gaze can become less reliable (for example, when users are all looking at a system display, even while talking with each other).

Past research on addressee detection has focused on H-H settings (such as meetings), sometimes with multimodal cues [1]. Relatively little work has looked at the H-H-C scenario [2]. Early systems relied primarily on rejection of H-H utterances either because they could not be interpreted [3] or yielded low speech recognition confidence [4]. Some systems combine gaze with lexical and syntactic cues to detect H-H speech [5]. Others use relatively simple prosodic features based on pitch and energy in addition to those derived from automatic speech recognition (ASR) [6]. An interesting recent approach is the use of nonstandard acoustic features, such as multiscale Gabor wavelets [7]. Prior H-H-C studies involved H-C speech consisting only of "commands", which are lexically constrained and therefore relatively easy to detect. The present work looks at a scenario where computer-directed speech can be free-form and linguistically unconstrained. To deal with this challenge, we employ energy contour models which, either alone or in combination with more traditional prosodic and ASR-based features, give promising results. Also, with a view toward portability and future system integration, we limit ourselves to features that are independent of context and speaker.

2. Method

2.1. Data

Data come from interactions between two acquaintances and a dialog system using only spoken input. Subjects were brought into a room and seated about 5 feet away from a large TV screen and roughly 3 feet away from each other. They were told about the basic capabilities of the system and the domains it could handle. They were also given a small set of short commands; those relevant to addressee detection included commands to start a new interaction, pause, stop listening, or 'wake up' the system. Subjects were told to otherwise use open-ended natural language. For more information about the dialog system itself and the spoken language understanding approach, see [8][9].

The resulting corpus comprises 6.3 hours of recordings over 17 sessions with 2 speakers each from a set of 13 unique speakers. Session durations ranged from 5 to 40 minutes, as determined by users. Speech was captured by a Kinect microphone; endpointing and recognition used an off the shelf recognizer. Although the full interaction was recorded, we focus on speech in the recognized segments, or "segments", as described in Table 1.

Table 1. *Speech segment types, distribution, and grouping for binary classification purposes*

Addressee, Type	Abbreviation	% Total	Class
Computer, command	C command	39.9	C
Computer, noncommand	C noncommand	38.3	C
Mixed Computer/Human	M	2.7	C
Human	H	19.1	H

A total of 1802 segments containing 1.2 total hours of speech were hand-transcribed and annotated for addressee. Computer-addressed segments were also labeled as either command or noncommand. Segments containing both human- and computer-addressed speech (in any sequence) were marked as "mixed"; since these were also processed by the system they were grouped with the computer-addressed class for detection purposes.

2.2. Lexical features

Lexical features (N-grams). We used unigrams, bigrams, and trigrams of automatically recognized words, including start/end-of-utterance tags. The speech recognition system used had a word error rate of 19% and a sentence error rate of 28%. (In 5.5% of utterances the recognition hypothesis was empty.) For experiments to assess the best case scenario for N-gram performance, we also extracted N-grams from parallel human-produced reference transcripts

Maximum cosine similarity. This feature aims to capture whether the user's utterance refers to content displayed by the system. Assume $d_{i,1}$, ..., $d_{i,n}$ are the n items that are shown to the user after turn i, then maximum cosine similarity is defined as

$$\max_{k=1,\ldots,n} \text{cossim}\left(d_{i,k}, u_{i+1}\right)$$

where u_{i+1} is the user's utterance in the next turn, and cossim(x,y) is the cosine between vectors representing texts x and y, each of which is a binary vector of length $|V|$, the number of terms in the vocabulary V; each vector component is 0 or 1, depending on the absence or presence of the corresponding word in the utterance.

ASR confidence. As in past work in this area [4][6], we also include a real-valued number representing the utterance-level confidence score for the 1-best sequence output by the recognizer. The motivation was that computer-directed speech should be better matched to recognizer acoustic and language models.

2.3. Acoustic-prosodic features

We also explored acoustic-prosodic features. Here, unlike some past studies in related areas, we only examine features consistent with all three conditions below—to facilitate later integration in an online system:

1. **Word independent**: features do not rely on ASR.
2. **Context-independent**: features do not rely on system state or information from other segments in the session. For example, no session-level normalization is used.
3. **Speaker-independent**: features do not require any speaker normalization or modeling.

Segment-level features. We extracted acoustic-prosodic features at the level of the Kinect segment, designed to capture energy and speaking rate features that meet the conditions above. While pitch features showed some value in separate analyses, in particular in detecting computer-directed commands, we exclude them here because they are more complicated to model, especially with respect to speaker-independence.

One set of segment-level prosodic features is extracted from energy peaks, similar to [6] but including additional measures. We ran a peak-picking algorithm [10] on 10-ms-frame intensity output from Praat [11], after mean subtraction. Features include the peak count, rate, mean and max distance apart, mean/max/min/stdev intensity value, and the location and value for the highest peak. Another set of features uses speech activity information to describe speaking rate and duration information. In practice, our speech activity features are computed from the time-alignment of the word recognition output within the region that triggered speech activity detection, without making reference to the identity of the recognized words. The features include total waveform duration, lengths of initial and final nonspeech regions, and the total duration of nonspeech regions between words.

Energy contour features. In examining computer-directed speech from a separate collection, we noted that it often sounds more rhythmic or "sing-songy" than typical human-human conversation. We sought to capture this behavior by extracting energy-related features in fixed-length temporal windows and modeling DCT bases with Gaussian mixture models (GMMs). The approach utilizes 10-millisecond-frame c0 output from standard MFCCs, a 200-millisecond sliding window with a 50% shift, and the first 5 DCT bases for mean-subtraction-normalized c0 output. Similar results were found using intensity output [11] instead of c0. Appending the first 2 bases for c1 added a small improvement. We also tried adding pitch contours, separately or in the same model, but did not find appreciable gains.

The contour modeling approach itself is similar to methods used in prior work on speaker verification and language identification. Select studies successfully incorporated prosodic contour information for those tasks, using Legendre or DCT bases of pitch and/or energy, and modeling them for either syllable-like segments or fixed-length windows, e.g., [12][13]. What is interesting is that, in this task, the energy contour features appear to capture differences in two speaking styles (human-directed and computer-directed) that cooccur within the same user, language, acoustic environment, and session.

2.4. Classifiers and evaluation

A variety of machine learning approaches were used to model the features described above, and to obtain classifiers for addressee detection. All classifiers output a real value that can serve either as a detection score, or as a new feature to be fed into second-level classifiers.

Language models. We compute a log likelihood ratio of the two addressee classes from lexical N-grams by modeling each class with a standard trigram backoff language model. Witten-Bell discounting was used for smoothing. The detection score for an utterance w is computed as

$$\frac{1}{|w|} \log \frac{P(w|C)}{P(w|H)}$$

where $|w|$ is the number of recognized words in the test utterance.

GMMs. The energy contour features employ Gaussian mixture models (GMM) to compute a log likelihood ratio. Training feature vectors for each class are pooled and a GMM with full

Table 2. *System performance. EER=equal error rate, Error=classification error. Subscripts denote features: asrng=asr word ngrams, refng=reference ngrams, cosim=max cosine similarity, conf=asr confidence, energy=c0 DCT bases, segstats=segment-level prosody, * = human-transcribed words.*

	System Type	Model	EER	Error
	Chance	Random decision/Majority class	50.00	19.10
1	Lexical (ASR)	LM_{asrng}	28.95	17.44
2	Lexical (ASR)	LLR (LM_{asrng} , $Boost_{cosim,conf}$)	23.11	16.67
3	Prosodic (noASR)	$Boost_{segstats}$	16.03	11.83
4	Prosodic (noASR)	GMM_{energy}	13.93	11.21
5	Prosodic (noASR)	LLR ($Boost_{segstats}$, GMM_{energy})	12.63	10.17
6	Lexical (ASR) + Prosodic (noASR)	LLR (LM_{asrng} , $Boost_{cosim,conf}$, $Boost_{segstats}$, GMM_{energy})	11.08	9.06
7	Lexical (REF*)	LM_{refng}	10.16	8.88
8	Lexical (REF*) + Prosodic (noASR)	LLR (LM_{refng} , $Boost_{segstats}$, GMM_{energy})	6.72	5.06

co-variances is trained. The score of a test utterance with feature vectors *X* then becomes

$$\frac{1}{|X|}\log\frac{P(X|C)}{P(X|H)}$$

where $|X|$ is the number of vectors, and $P(X|class)$ is the aggregate GMM likelihood, assuming independence among the vectors. The energy contour features described earlier are modeled by a 20-mixture component GMM. Given large improvements from session variability compensation techniques in other work [14] it was natural to try these approaches. We applied eigenchannel compensation to the energy contours but found no improvement; this can be revisited when more data are available.

Boosting. Real-valued and binary utterance-level features are modeled by the Boostexter adaptive boosting algorithm [15] as implemented by [16]. Boosting induces a strong learner as a weighted combination of weak learners, each of which examines only a single feature of the input. The weighted combined score also serves as a detection score in our experiments. We use boosting to jointly model the segment-level acoustic-prosodic features, as well as max cosine similarity and ASR confidence. While it is also possible to include N-gram features in boosting, we found on our data that language models for those features give better results.

LLR. We use linear logistic regression (LLR) to calibrate and combine one or more detection scores (obtained by any of the methods described earlier). Given input scores x_1, …, x_n, the LLR model produces a new score $x = \text{sigmoid}(a_0 + a_1x_1 + \cdots + a_nx_n)$, where the function $\text{sigmoid}(x) = \frac{1}{1+e^{-x}}$ ranges between 0 and 1 and can be interpreted as a posterior probability of the target class. The parameters a_0, …, a_n are estimated on the training data to minimize the cross-entropy between the model's predictions *x* and the target labels.

Evaluation. We evaluate results using equal error rate (EER), actual error rate (broken down further by utterance type), and detection error tradeoff (DET) curves. Given the current size of this new data collection, we used cross-fold validation, in two stages. The available data was divided into 17 partitions, at session boundaries. The top-level classifiers for each experiment where then trained and tested using 17-fold cross-validation, using 16 sessions for training, and testing on the remaining one, round-robin until all sessions are used once for testing. Results aggregate results over the entire dataset. For experiments involving two levels of classifiers (e.g., a GMM producing scores used as input to LLR), the cross-validation was carried out in a hierarchical manner. Assume that classifier *A* produces inputs for classifier *B*, the top-level classifier, and that as part of the cross-validation classifier *B* is to be trained on sessions 1,…,16 and tested on session 17. To generate inputs for *B* using *A*, we would apply cross-validation to the subset 1,…,16, i.e., train *A* on 15 sessions and using its outputs on the 16th session, thereby cycling through the 16 training sessions without touching the 17th used as the test set for *B*.

3. Results

Table 2 summarizes the performance of various subsets of features and their combinations. *EER* is the value at which false detections and misses occur with the same probability relative to their true classes, a metric that is independent of the class priors. *Error* denotes the overall classification error on the class distribution seen in the data. Since class priors depend on many factors, we are primarily interested in the discriminative power of systems independent of prior class distribution. Figure 1 plots the detection error tradeoff (DET) between false alarm and miss errors for the eight systems in Table 2.

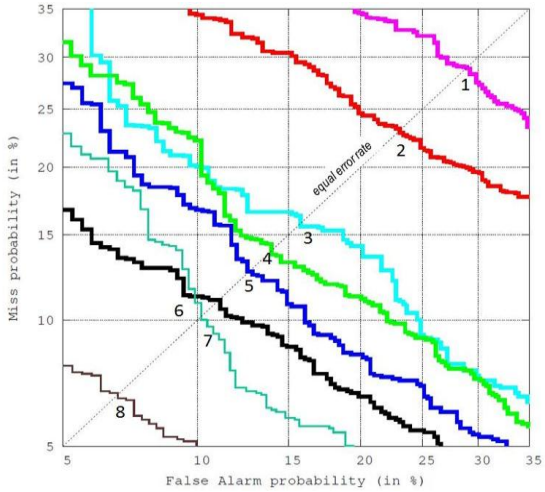

Figure 1. *DET curves for the systems in Table 2. Axes use a normal deviate scale, thin curves use REF words.*

As shown in Figure 1, all feature types give significant performance gains when combined with others, whether within or across feature types (lexical or prosodic). ASR confidence and max cosine similarity (system 2) add to word N-grams (1). The two individual prosodic models (3,4), despite similar Error (Table 2), combine well to reduce both Error and EER (5). Prosodic models alone (3,4,5) give far better results than lexical features alone (1,2) and also combine well with lexical features, yielding the best ASR-based performance (6). Prosodic features (5) even provide a 35% relative reduction in EER when added (8) to a system using reference words (7).

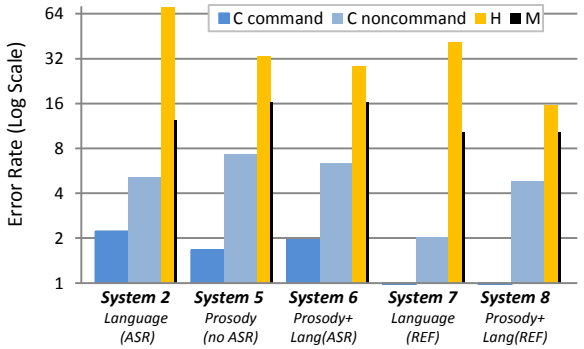

Figure 2. *Error rate by segment type and system. Note the error rates are on a log scale. Bar widths reflect relative frequencies of utterance types (see Table 1).*

Figure 2 breaks down performance by segment type. A clear pattern is the high error rates on the human-directed segments (H), especially noting the scale. For example, the H class error rate for System 2 is over 71% error. Prosody (System 5) greatly reduces this rate in absolute terms—from 71% to 33%, without large absolute error increases for commands or noncommands. With correct words alone (System 7), commands are detected as C almost perfectly, but the H class still has over 40% error. This is reduced to 15% for System 8, without adding errors on commands, and only slightly increasing error on noncommands in absolute counts. Mixed-type utterances have results intermediate between H and C types; they exhibit the least reduction in classification error, suggesting that they might require special treatment to achieve further improvements.

4. Discussion and Conclusions

Acoustic-prosodic features that do not require word recognition offer the possibility of reducing latency in a real-time system, and may facilitate portability across domains and even languages. Lexical features alone, even for improved ASR, quality can remain variable in the face of noise and various sources of model/data mismatch. The large difference (Table 2) found in N-gram model performance between ASR and REF means that it will be difficult to calibrate such a system. Results in Figure 2 show that even with reference words, lexical features still have trouble classifying human-addressed speech.

In summary, new challenges arise for addressee detection when users speak with each other (about the domain) *and* use unconstrained language to interact with a system. We find that in particular, natural unconstrained utterances to a system are confusable with speech between users. A combination of lexical and ASR-independent prosodic models yields large error reductions from a lexical model alone, holding up even if word recognition is perfect. As a prosodic model, we propose in particular a simple energy contour GMM that yields low error rates alone as well as in combination with other systems. Overall, we conclude that speakers modify their prosody not only in commands, but also in unconstrained computer-directed speech, and that this style difference can be harnessed for improved addressee detection in both H-C and H-H-C dialog.

5. Acknowledgments

We thank our colleagues G. Tur, A. Fidler, R. Iyer, P. Parthasarathy, M. Chinthakunta, L. Stifelman, and P. Greborio for useful discussions and for creating the infrastructure and data that enabled this research. L. Burget and L. Ferrer at SRI International provided valuable resources and advice on modeling questions.

6. References

[1] R. op den Akker & D. Traum, A comparison of addressee detection methods for multiparty conversations, *Proc. DiaHolmia*, pp. 99-106, 2009

[2] D. Bohus & E. Horvitz, Multiparty Turn Taking in Situated Dialog: Study, Lessons, and Directions, *Proc. ACL SIGDIAL*, 2011.

[3] T. Paek, E. Horvitz, & E. Ringger, Continuous listening for unconstrained spoken dialog. *Proc. ICSLP*, pp. 138-141, 2000.

[4] J. Dowding, R. Alena, W. J. Clancey, M. Sierhuis, & J. Graham, Are You Talking To Me? Dialogue Systems Supporting Mixed Teams of Humans and Robots, *Proc. AAAI Fall Symposium: Aurally Informed Performance*, Washington, DC, 2000.

[5] M. Katzenmaier, R. Stiefelhagen, & T. Schultz, Identifying the Addressee in Human-Human-Robot Interactions based on Head Pose and Speech, *Proc. 6th Intl. Conf. Multimodal Interfaces*, 2004.

[6] D. Reich, F. Putze, D. Heger, J. Ijsselmuiden, R. Stiefelhagen, & T. Schultz, A Real-Time Speech Command Detector for a Smart Control Room, *Proc. Interspeech*, pp. 2641-2644, 2011.

[7] T. Yamagata, T. Takiguchi, & Y. Ariki, System Request Detection in Human Conversation Based on Multi-Resolution Gabor Wavelet Features, *Proc. Interspeech*, pp. 256-259, 2009.

[8] D. Hakkani-Tür, G. Tur, & L. Heck, Research Challenges and Opportunities in Mobile Applications, *IEEE Signal Processing Magazine* 28(4), August 2011.

[9] D. Hakkani-Tür, G. Tur, L. Heck, & E. Shriberg, Bootstrapping Domain Detection Using Query Click Logs for New Domains. *Proc. Interspeech*, pp. 709-712, 2011.

[10] N. C. Yoder, PeakFinder (Matlab program), http://www.mathworks.com/ matlabcentral/fileexchange/25500-peakfinder, 2011.

[11] P. Boersma & D. Weenink, Praat: doing phonetics by computer (Version 5.1.05), http://www.praat.org/, 2009.

[12] C.-Y. Lin & H.-C. Wang, Language Identification Using Pitch Contour Information, *Proc. ICASSP*, vol. 1, pp. 601-604, 2005.

[13] N. Dehak, P. Dumouchel, & P. Kenny, Modeling Prosodic Features With Joint Factor Analysis for Speaker Verification, *IEEE Trans. Audio Speech and Language Processing* 15(7), 2095-2103, 2007.

[14] L. Burget, P. Matějka, P. Schwarz, O. Glembek, & J. Černocký, Analysis of feature extraction and channel compensation in GMM speaker recognition system, *IEEE Trans. Audio, Speech, and Language Processing* 15(7), 1979-1986, 2007.

[15] R. E. Schapire & Y. Singer, Boostexter: A Boosting-based System for Text Categorization, *Machine Learning* 39(2/3), 135-168, 2000.

[16] B. Favre, D. Hakkani-Tür, & S. Cuendet, icsiboost – Open-source implementation of Boostexter, http://code.google.come/p/icsiboost, 2007.