Root Cause Analysis of Miscommunication Hotspots in Spoken Dialogue Systems

Spiros Georgiladakis[1], *Georgia Athanasopoulou*[1], *Raveesh Meena*[2], *José Lopes*[2], *Arodami Chorianopoulou*[3], *Elisavet Palogiannidi*[3], *Elias Iosif*[4,4], *Gabriel Skantze*[2], *Alexandros Potamianos*[1,4]

[1]School of Electrical & Computer Engineering, National Technical University of Athens, Greece
[2] KTH Speech, Music and Hearing, Stockholm, Sweden
[3]School of Electrical & Computer Engineering, Technical University of Crete, Chania, Greece
[4]"Athena" Research and Innovation Center, Athens, Greece

{sgeorgil,gathanasop,iosife,potam}@central.ntua.gr,
{raveesh,jdlopes,gabriel}@kth.se, {achorianopoulou,epalogiannidi}@isc.tuc.gr

Abstract

A major challenge in Spoken Dialogue Systems (SDS) is the detection of problematic communication (hotspots), as well as the classification of these hotspots into different types (root cause analysis). In this work, we focus on two classes of root cause, namely, erroneous speech recognition vs. other (e.g., dialogue strategy). Specifically, we propose an automatic algorithm for detecting hotspots and classifying root causes in two subsequent steps. Regarding hotspot detection, various lexico-semantic features are used for capturing repetition patterns along with affective features. Lexico-semantic and repetition features are also employed for root cause analysis. Both algorithms are evaluated with respect to the Let's Go dataset (bus information system). In terms of classification unweighted average recall, performance of 80% and 70% is achieved for hotspot detection and root cause analysis, respectively.

Index Terms: miscommunication detection, miscommunication root causes, spoken dialogue systems

1. Introduction

Despite recent progress in spoken dialogue system (SDS) technologies, one major roadblock in commercial SDS prototyping is the significant effort and expertise required for the enhancement of the performance of deployed services. The iterative enhancement process is often performed with little automation by inspecting data logs and partially transcribed calls. A wide-array of technologies has emerged under the umbrella term *speech analytics* that facilitates the automatic or semi-automatic extraction of relevant information from large amount of speech data, e.g., audio mining for keywords and topics, affective analysis, analysis of speaker population characteristics, attitudes and behaviors.

Unlike human-human interaction, the detection and resolution of miscommunication is not trivial for the case of SDS [1]. A challenging speech analytics task is the detection of problematic communication in dialogue turns, as well as the identification of the SDS components to which such problems may be attributed. In this work, we refer to such problematic turns as *hotspots*, while the cause identification process is termed as *root cause analysis*. One of the earliest approaches for the detection of miscommunication in SDS was proposed in [2] via supervised learning, exploiting features derived from the logs, e.g., ASR output, logs of the components dealing with natural language understanding and dialogue management etc. The detection was formulated as a classification problem where rule learning algorithms were applied. This formulation was also followed in later research efforts, such as [3]. A set of shallow linguistic features (e.g., part-of-speech labels) was utilized in [4], along with word statistics and turn- and dialogue-level properties, in order to predict the success of turns and entire dialogues. Based on the observation that prosodic speech is associated with problematic dialogues [5], various prosodic features have been employed for hotspot detection. For example, in [6] prosodic features were used for the classification of problematic dialogues with respect to elicited speech and a Wizard-of-Oz scenario. Despite the saliency of such features, the authors suggest the incorporation of additional feature types (e.g., repetitions). The detection results have been used for tasks such as the computation of the optimal strategy for routing the call to a human operator [7]. A related study was presented in [8] dealing with the tuning of dialogue management and strategies, based on massive data collected from large-scale applications.

The approach that is most related to the present work is [1], where two different techniques are proposed for the detection of hotspots, namely, online and offline detection. We focused on the development of an offline detection model via supervised learning, exploiting features extracted from manually annotated system logs. In this work, we follow the offline approach presented in [1] by incorporating new features for hotspot detection. The proposed feature set includes various lexico-semantic and affective features. Most importantly, we perform the additional step of automatically identifying the type of hotspot, i.e., performing root cause analysis.

2. Problem definition

According to [9], miscommunication in dialogues can be studied with respect to non-understanding and misunderstanding. In general, this distinction applies both to human-human and human-system conversational interactions. This is of great relevance for SDS, where the system is expected to both recognize the speech input and understand the underlying user's intent. This is exemplified in the dialogue excerpt presented in Table 1. An example of non-understanding can be found in turn 3, where the system fails to make any hypothesis for the user's

Table 1: *Example of miscommunication.*

1	S: *What type of restaurant are you looking for?*
2	U: I'm looking for a Greek takeaway restaurant.
3	S: *Sorry, I didn't get that.*
4	U: Greek takeaway.
5	S: *I recommend Pane that serves Italian pizza.*
6	U: Greek restaurant.

input in turn 2. As it is shown in turn 5, the system misunderstands the user's intent expressed in turn 4. The misunderstanding can be identified in turn 6, where the user rephrases their choice. It was observed that non-understandings are typically detected right away, while misunderstandings are typically spotted at later dialogue turns [1]. According to this observation, the detection of miscommunication errors can be defined as "early" and "late" [10]. In early detection, the current system hypothesis about the most recent user utterance is used, while for late detection, a number of previous turns is taken into consideration.

A major task for the enhancement and tuning of SDS is the identification of cases where the dialogue policy may trigger ASR errors. For example, a widely-used strategy for handling non-understanding is to prompt the user to repeat. In such a scenario, the user may get frustrated and use hyper-articulated speech, which in turn is likely to cause new ASR errors. Motivated by the observation that erroneous ASR constitutes one of the most frequent causes of miscommunication problems in SDS [11], we propose an automatic process as a first step towards the aforementioned task, outlined as follows: 1) the detection of hotspots, and 2) the subsequent classification of hotspots' root causes with respect to ASR vs. other components of SDS (i.e., root cause analysis). While there is a large body of literature regarding the prediction of ASR errors (e.g., [12, 13]), root cause analysis is a much less researched problem. Root cause analysis is also interesting because it is formulated upon established miscommunication events [14].

3. Hotspot detection

In this section, we briefly present various features used for the detection of hotspots. The features are distinguished into three types, namely, lexical, semantic, and affective, which are presented in Table 2 along with the respective modalities. Lexical and semantic features were used for identifying repetitions in system prompts and user utterances, based on the observation that miscommunication incidents are characterized by such repetitions. The affective content of user utterances was taken into consideration, based on the hypothesis that problematic communication is likely to cause negative emotions.

Table 2: *Hotspot: feature types.*

Feature type	Modality	To capture
Lexical	text	repetitions
Semantic	text	repetitions
Affective	speech, text	emotion

Lexical features: Given a pair of transcribed system prompt(s) and/or user utterance(s), denoted as t_1 and t_2, the following lexical features were computed: 1) the Levenshtein distance of t_1 and t_2, 2) two Dice coefficients computed according to the common (i) words and (ii) character bigrams shared between t_1 and t_2.

Semantic features: The set of semantic features includes: 1) a semantic similarity score between t_1 and t_2 estimated using a state-of-the-art semantic model [15], 2) a binary value indicating whether t_1 is a paraphrase of t_2 [16], 3) a score of semantic concreteness for each chunk computed by averaging the respective word-level scores retrieved from the MRC Psycholinguistic Database [17].

Affective features: In order to capture the affective content of the user's speech input, a set of appropriate low-level descriptors (LLDs) were used. This set includes prosody (pitch and energy), short-term spectral (Mel Frequency Cepstral Coeficients, MFCCs) and voice quality (Jitter) [18]. The LLDs were extracted using a fixed window size of 30 ms with a 10 ms frame update. The following statistics were computed for each of the LLDs and used in the feature set: percentiles, extremes, moments, and peaks. In addition, for each transcribed user utterance, the following steps were performed: (i) a score was retrieved for each constituent word from an affective lexicon, and (ii) a feature set was constructed including a set of basic statistics (mean, median, min, max, variance) computed over the word-level scores. This was done for the three basic affective dimensions, namely, valence, arousal and dominance In this work, we used the English affective lexicon presented in [19], which was created via the automatic expansion of a manually-crafted seed lexicon (ANEW [20]).

4. Root cause analysis

In this section, we present the features used for root cause analysis Given a pair of a system prompt t_s and a user utterance t_u, along with their immediate preceding turn exchange t_s^p and t_u^p, the following features were computed.

Log-derived features: Features extracted from the interaction logs, namely: 1) the ASR confidence score for t_u, 2) the timestamp of the turn exchange, computed wrt. the total dialogue duration, 3) the task (dialogue act) that the user is aiming to achieve [1].

Lexical features: The set of lexical features includes: 1) the Levenshtein distance, 2) two Dice coefficients based on common words and character bigrams. The distance and coefficients were computed between a) t_u^p and t_u, b) t_u^p and t_s, and c) t_s^p and t_s.

N-gram features: We extracted a) unigrams, b) bigrams, c) trigrams, and d) sentences from turn exchanges associated with root cause, as well as from the preceding turn exchange, i.e., n-grams were extracted from t_s^p, t_u^p, t_s and t_u. The extracted n-grams were subsequently ranked according to their class-conditional entropy, computed as

$$E(n) = \sum_{i}^{k} -p(c_i \mid n) log p(c_i \mid n), \quad (1)$$

where $p(c_i \mid n)$ is the probability of class c_i given n-gram n, and k stands for the number of classes. We selected the n-grams with the lowest score (see Section 5.2) to constitute our n-gram features.

[1] In this work, the dialogue acts were manually annotated.

5. Experimental setup

5.1. Hotspot detection

The detection of hotspots was formulated as a two-class classification problem, i.e., each turn was classified as "problematic" (hotspot) vs. "non-problematic". We utilized the Let's Go (LG) '09, '12, and '14 datasets[2], described in detail in [21]. The number of turn exchanges with hotspot information, along with the percentage of problematic turn exchanges, is displayed in Table 3. All datasets were preprocessed by lowercasing and removing punctuation from user utterances and prompts. We applied the JRip classifier. The evaluation was performed using 10-fold cross validation (10-FCV), while the performance is reported in terms of Unweighted Average Recall (UAR). As the baseline for our experiments, we used the *majority class*.

Table 3: *LG '09, '12 and '14 hotspot detection datasets.*

Dataset	# Turn exchanges	Problematic %
LG '09	792	38.3%
LG '12	985	60.2%
LG '14	1344	44.9%

5.2. Root cause analysis

For this task, we also utilized the LG '09, '12, and '14 datasets. Turn exchanges that were manually annotated as problematic (hotspots) were further annotated for root cause analysis by an expert annotator with one of the following root cause types:

- *ASR*: error by the speech recognizer
- *DP*: dialogue policy error
- *EP*: endpoint error
- *OOD*: out-of-domain utterance
- *SLU*: spoken language understanding failure
- *PROMPT*: prompt design error
- *BE*: back-end error

The number of turn exchanges associated with a root cause and the percentage of root cause annotations in the datasets are presented in Table 4. The percentage of each root cause type in turn exchanges associated with root cause is displayed in Table 5. All datasets were preprocessed by lowercasing and removing punctuation from user utterances and system prompts. Two systems were setup, comprising of combinations of the LG datasets. According to the first system, data from previous system versions were used for training (LG '09 and '12), while the test was performed in interaction logs of the latest version (LG '14). For the second system, data from all three versions were used for training and testing via 10-FCV.

Table 4: *LG '09, '12 and '14 root cause analysis datasets.*

Dataset	# Turn exchanges	RC %
LG '09	305	38.1%
LG '12	568	31.8%
LG '14	684	31.1%

[2] http://www.speech.cs.cmu.edu/letsgo/

Table 5: *Percentage of root cause types wrt. turn exchanges.*

Root Cause	LG '09	LG '12	LG '14	Total
ASR	78.7%	57.4%	67.3%	65.9%
DP	27.5%	29.6%	13.6%	22.2%
EP	0%	17.6%	16.2%	13.6%
OOD	7.5%	5.3%	4.5%	5.4%
SLU	5.6%	1.9%	3.4%	3.3%
PROMPT	4.9%	0.8%	0.6%	1.5%
BE	0%	0.1%	0.7%	0.3%

Since ASR errors are the most frequent types in the datasets, we focused on classifying erroneous states as ASR vs. other SDS errors. To this end, we defined two categories for predicting root cause for a given turn exchange: 1) *ASR*: consisting of ASR errors, and 2) *Non-ASR*: consisting of errors associated with DP and/or EP. We used the features described in Section 4, while the top 2% of the total n-grams were utilized after being ranked in ascending order according to (1).

As the baseline for our experiments, we used the *majority class*. We applied the Random Forest [22] and SVM [23, 24] classifiers, as well as BoosTexter, an implementation that is based on a collection of boosting algorithms which can be trained from raw textual input [25]. UAR was used as the evaluation metric. In addition, for the case of BoosTexter we used the classification confidence score, computed by the classifier (as described in [25]), for setting a threshold. This approach was used for discarding the classifications that were scored with confidence falling below the threshold. For this setting, the measurements of precision Pr, recall Rc and F-measure Fm were used.

6. Evaluation results

In Section 6.1 we present the evaluation results for hotspot detection. The evaluation results of root cause analysis are presented in Section 6.2.

6.1. Hotspot detection

The performance yielded by the feature types described in Section 3 is presented in Table 6 for the LG '09, '12 and '14 datasets. Regarding the individual feature types, the highest UAR score is yielded by the lexical features (72.3%, 74.6%, and 77.2% for the LG '09, '12 and '14, respectively). The best performance is achieved when all feature types are exploited (via the concatenation of the individual feature vectors) and equals to 79.4%, 76.1% and 81.8% UAR for the LG '09, '12 and '14, respectively. Regarding LG '09, the top UAR score (79.4%) does not exceed the best score (88.0%) reported in [1]. This difference can be attributed to the use of different feature

Table 6: *Hotspot detection: feature evaluation (UAR %).*

Feature type	LG '09	LG '12	LG '14	Avg.
Majority class	50.0	50.0	50.0	50.0
Lexical	72.3	74.6	77.2	74.7
Semantic	68.1	71.2	76.9	72.1
Affective (text)	71.4	61.7	65.5	66.2
Affective (speech)	59.2	50.1	50.7	53.3
All	**79.4**	**76.1**	**81.8**	**79.1**

types. However, in preliminary experiments using a subset of LG '09 we found that the fusion of the feature set utilized in this work with the features used in [1] yields higher performance (85.0%) compared to the performance of the individual feature sets (76.0% for the features of the present work and 83.0% for the features of [1]).

6.2. Root cause analysis

The evaluation results with respect to different classifiers are presented in Table 7. When using the first system (i.e., training

Table 7: *Root cause analysis: classifier evaluation (UAR %).*

Training Set Test Set	LG '09+'12 LG '14	LG '09+'12+'14 10-FCV
Majority class	50.0	50.0
Random Forest	**63.2**	63.9
SVM	58.1	**71.3**
BoosTexter (thres = 0)	60.9	63.7

with the LG '09 and '12), the best performance is achieved by Random Forest (63.2%), followed by BoosTexter. All classifiers perform better when using the second system (i.e., training with all LG datasets). The top performance is achieved by SVM (71.3%). BoosTexter and Random Forest obtain comparable performance (63.7% and 63.9%, respectively).

The performance for the case of BoosTexter is shown in Fig. 1 as a function of the classification confidence score used as threshold. This is displayed when using LG '09, '12 and '14 for training and testing with 10-FCV. We experimented with values ranging from 0 to 0.04. It is observed that the precision slightly improves as the threshold value increases, however, the recall drops. Similar observations are also made for the first system.

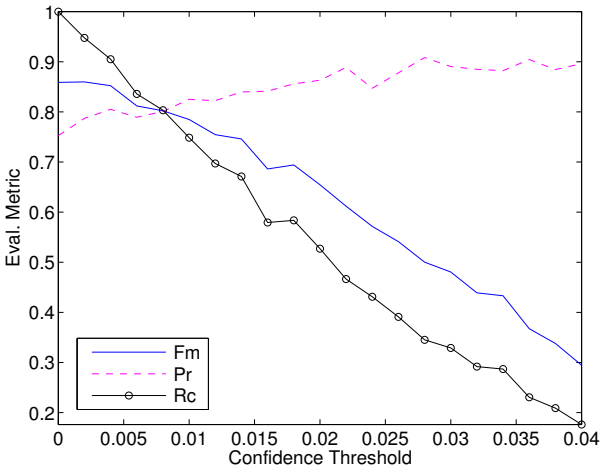

Figure 1: *Performance of BoosTexter (Precision Pr, Recall Rc, F-measure Fm) on different confidence threshold values.*

In Table 8, we present the performance of the features described in Section 4. The most salient feature types are those based on n-grams. The lowest performance is observed for the log-derived and lexical features. Also, it is shown that the most appropriate point in the dialogue for feature extraction is the previous turn exchange. User utterances and system prompts

Table 8: *Root cause analysis: feature evaluation (UAR %).*

Feature type	UAR (%)
Majority class baseline	50.0
Log-derived	50.0
Lexical	50.0
N-grams: prev. turn exchange	70.0
N-grams: curr. turn exchange	58.6
N-grams: user utterance	59.1
N-grams: system prompt	59.6
All	**71.3**

appear to yield comparable performance. Overall, the top UAR score (71.3%) was achieved when all feature types were used.

7. Conclusions

In this work, we proposed a two-stage automatic algorithm for detecting and analyzing the root cause of problematic communication (hotspots). For the task of hotspot detection, approximately 80% UAR was achieved via the exploitation of lexico-semantic and affective features. Approximately 70% UAR was achieved for the classification of root cause, using n-grams extracted from turn exchanges as well as lexico-semantic features. Regarding hotspot detection, the lexical features were the highest performing feature type. For the case of root cause analysis, the best performance was achieved when using n-gram features extracted from the preceding turn exchange.

Our ongoing work deals with the fusion of the proposed features with other feature types reported in the literature for hotspot detection, e.g., [1]. In the future, we will also investigate a more fine grained classification scheme for the task of root cause analysis. Last but not least, our end goal is to integrate the presented features and algorithms into a toolkit for speech analytics, aimed to aid the enhancement and tuning of SDS.

8. Acknowledgements

The authors would like to thank Prof. Joakim Gustafson for fruitful discussions. This work has been partially supported by the SpeDial project supported by the EU FP7 with grant number 611396.

9. References

[1] R. Meena, G. Skantze, and J. Gustafson, "Automatic detection of miscommunication in spoken dialogue systems," in *Proc. of SIGdial*, 2015.

[2] M. Walker, I. Langkilde, J. Wright, A. Gorin, and D. Litman, "Learning to predict problematic situations in a spoken dialogue system: experiments with how may I help you?" in *Proc. of NAACL*, 2000, pp. 210–217.

[3] A. van den Bosch, E. Krahmer, and M. Swerts, "Detecting problematic turns in human-machine interactions: rule-induction versus memory-based learning approaches," in *Proc. of ACL*, 2001, pp. 82–89.

[4] S. Steidl, C. Hacker, C. Ruff, A. Batliner, E. Nöth, and J. Haas, *Text, Speech and Dialogue*. Springer Berlin Heidelberg, 2004, ch. Looking at the Last Two Turns, I'd Say This Dialogue Is Doomed – Measuring Dialogue Success, pp. 629–636.

[5] E. Krahmer, M. Swerts, M. Theune, and M. Weegels, "Error detection in spoken human-machine interaction," *International journal of speech technology*, vol. 4, no. 1, pp. 19–30, 2001.

[6] A. Batliner, C. Hacker, S. Steidl, E. Nöth, and J. Haas, "User states, user strategies, and system performance: how to match the one with the other," in *Proc. of ISCA Tutorial and Research Workshop on Error Handling in Spoken Dialogue Systems*, 2003.

[7] T. Paek and E. Horvitz, "Optimizing automated call routing by integrating spoken dialog models with queuing models," in *Proc. of HLT-NAACL*, 2004, pp. 41–48.

[8] D. Suendermann, J. Liscombe, J. Bloom, G. Li, and R. Pieraccini, "Large-scale experiments on data-driven design of commercial spoken dialog systems." in *Proc. of Interspeech*, 2011, pp. 813–816.

[9] G. Hirst, S. McRoy, P. Heeman, P. Edmonds, and D. Horton, "Repairing conversational misunderstandings and non-understandings," *Speech communication*, vol. 15, no. 3, pp. 213–229, 1994.

[10] G. Skantze, "Error handling in spoken dialogue systems," Ph.D. dissertation, KTH - Royal Institute of Technology, School of Computer Science and Communication, Department of Speech, Music and Hearing, 2007.

[11] D. Bohus and A. Rudnicky, "A principled approach for rejection threshold optimization in spoken dialog systems," in *Proc. of Interspeech*, 2005, pp. 2781–2784.

[12] D. J. Litman, M. A. Walker, and M. S. Kearns, "Automatic detection of poor speech recognition at the dialogue level," in *Proc. of ACL*, 1999, pp. 309–316.

[13] J. Hirschberg, D. Litman, and M. Swerts, "Prosodic and other cues to speech recognition failures," *Speech Communication*, vol. 43, no. 1, pp. 155–175, 2004.

[14] R. Meena, "Data-driven methods for spoken dialogue systems: Applications in language understanding, turn-taking, error detection, and knowledge acquisition," Ph.D. dissertation, KTH - Royal Institute of Technology, School of Computer Science and Communication, Department of Speech, Music and Hearing, 2016.

[15] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, and A. Y. Ng, "Grounded compositional semantics for finding and describing images with sentences," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 207–218, 2014.

[16] R. Socher, E. H. Huang, J. Pennin, C. D. Manning, and A. Y. Ng, "Dynamic pooling and unfolding recursive autoencoders for paraphrase detection," in *Proc. of Advances in NIPS*, 2011, pp. 801–809.

[17] M. Wilson, "MRC psycholinguistic database: Machine-usable dictionary, version 2.00," *Behavior Research Methods, Instruments, & Computers*, vol. 20, no. 1, pp. 6–10, 1988.

[18] C. Busso, M. Bulut, and S. Narayanan, *Social emotions in nature and artifact: emotions in human and human-computer interaction*. Oxford University Press, 2010, ch. Toward effective automatic recognition systems of emotion in speech, pp. 110–127.

[19] E. Palogiannidi, E. Iosif, P. Koutsakis, and A. Potamianos, "Valence, arousal and dominance estimation for English, German, Greek, Portuguese and Spanish lexica using semantic models," in *Proc. of Interspeech*, 2015.

[20] M. M. Bradley and P. J. Lang, "Affective norms for English words (ANEW): Instruction manual and affective ratings," Center for Research in Psychophysiology, University of Florida, Tech. Rep., 1999.

[21] A. Raux, B. Langner, D. Bohus, A. W. Black, and M. Eskenazi, "Let's go public! Taking a spoken dialog system to the real world," in *Proc. of Interspeech*, 2005, pp. 885–888.

[22] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[23] J. C. Platt, "Fast training of Support Vector Machines using Sequential Minimal Optimization," in *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1998, pp. 185–208.

[24] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy, "Improvements to Platt's SMO algorithm for SVM classifier design," *Neural Computation*, vol. 13, no. 3, pp. 637–649, 2001.

[25] R. E. Schapire and Y. Singer, "Boostexter: A boosting-based system for text categorization," *Machine learning*, vol. 39, no. 2, pp. 135–168, 2000.