
Proceedings Paper:
Casanueva, I., Hain, T. orcid.org/0000-0003-0939-3464 and Green, P. (2016) Improving generalisation to new speakers in spoken dialogue state tracking. In: Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH. Interspeech 2016, 08-12 Sep 2016, San Francisco, USA. , pp. 2726-2730.

https://doi.org/10.21437/Interspeech.2016-404

Improving generalisation to new speakers in spoken dialogue state tracking

Iñigo Casanueva, Thomas Hain, Phil Green

Department of Computer Science, University of Sheffield, United Kingdom

{i.casanueva, t.hain, p.green}@sheffield.ac.uk

Abstract

Users with disabilities can greatly benefit from personalised voice-enabled environmental-control interfaces, but for users with speech impairments (e.g. dysarthria) poor ASR performance poses a challenge to successful dialogue. Statistical dialogue management has shown resilience against high ASR error rates, hence making it useful to improve the performance of these interfaces. However, little research was devoted to dialogue management personalisation to specific users so far. Recently, data driven discriminative models have been shown to yield the best performance in dialogue state tracking (the inference of the user goal from the dialogue history). However, due to the unique characteristics of each speaker, training a system for a new user when user specific data is not available can be challenging due to the mismatch between training and working conditions. This work investigates two methods to improve the performance with new speakers of a LSTM-based personalised state tracker: The use of speaker specific acoustic and ASR-related features; and dropout regularisation. It is shown that in an environmental control system for dysarthric speakers, the combination of both techniques yields improvements of 3.5% absolute in state tracking accuracy. Further analysis explores the effect of using different amounts of speaker specific data to train the tracking system.

Index Terms: dialogue state tracking, dysarthric speakers

1. Introduction

Due to the rapidly growing demand on spoken interfaces for electronic devices, the development of these interfaces has become a key research topic in speech technology [1]. Dialogue state tracking (DST) is a key requirement in these interfaces, as it maps the dialogue history up to the current dialogue turn (Spoken language understanding (SLU) output, actions taken by the device, etc.) to a probabilistic representation called the *dialogue state* or *belief state*. This representation will later be the input used by the dialogue policy to decide which action should be taken next [2, 3]. Recently, the Dialogue State Tracking Challenges (DSTC) [4, 5, 6] were held, where it was shown that data driven discriminative models for DST outperform generative models. One of the reasons for this is the capacity of discriminative models to use higher dimensional, possibly correlated, input features, by directly modelling the conditional probability of the dialogue state given the input features [7]. The DSTCs also defined standard DST scoring metrics and provided annotated corpora for further research. However, these corpora were gathered in a specific domain (information gathering) where many users interacted with a system once or a few times. Therefore, the corpora are not suitable to study system adaptation to specific speakers. On the other hand, spoken interfaces to digital devices are likely to be used by a single user over many interactions. Speaker adaptation of ASR acoustic models is commonly used [8], but little research investigated user adaptation of dialogue management or state tracking [9, 10, 11].

Personalisation of dialogue interfaces can bring a large improvement to voice enabled environmental control interfaces for assistive technologies. For instance, users with dysarthria face various problems when using conventional spoken interfaces due to high error rates of the ASR. This is caused by the unusual characteristics of their speech with respect to conventional and other dysarthric speakers. However, using a small amount of data from the target speaker to adapt the acoustic model greatly improves ASR performance for dysarthric speakers [12, 13]. In dialogue management, extending the input features of the dialogue policy with speaker specific features (extracted from the acoustic signal and the ASR) showed improvements in dialogue reward [10]. If the usual DST input features (the SLU output and the last system action) are extended with these extra speaker features, the capacity of the discriminative trackers to handle a richer set of input features can increase the benefit obtained from using these features.

When developing an environmental control interface designed for dysarthric speakers, such as homeService [14], the following scenario is likely to be found: a system for a new *target* user must be set up, in which only data from other *source* users is available. This will result in a mismatch between the training and the evaluation data, which was one of the main problems machine learning-based dialogue state trackers faced in the DSTCs [4, 6]. In order to solve this, techniques that lead to generalization to unseen data have to be applied or the performance with the target user will be poor. This paper proposes two techniques aiming to improve generalization to data from unseen speakers, to be used with an LSTM-based state tracker: First, the previously mentioned input feature augmentation with speaker specific features (*iVectors* and ASR-related), which helps to find similarities between the target and the source speakers. Second, *dropout* regularization [15], which helps to not only generalize to unseen speakers, but also increases the performance improvement of the tracker when using the augmented input features. In a further analysis, it is shown that the effect of these generalization techniques increases when a small amount of target speaker data is available.

2. Dialogue state tracking

In each dialogue turn, the dialogue manager decides which action to take depending on the *dialogue state*, a representation of what the user has stated up to the current turn. Therefore, a component in charge of inferring the dialogue state in each turn is needed, *the dialogue state tracker*. This component takes the dialogue history as input (the collection of ASR-SLU observations, machine actions, etc. up to the current turn) and estimates the distribution over the dialogue state, also known as the *belief state*. Historically, machine learning approaches to DST used generative models [2, 16], which need to model all the correlations in the input features. This forced the generative models to

make many conditional independence assumptions and to use just the *dialogue features* (SLU output plus last system action) as input features in order to maintain tractability. On the other hand, in the DSTCs it was shown how discriminative models outperform generative ones in DST, because of their capability to incorporate a rich set of features without worrying about their dependencies on one another. Most models used very high dimensional input features generated from the dialogue features [17, 18, 19] and others even extracted the features directly from the ASR output [20].

2.1. RNN-LSTM for DST

Recurrent neural networks (RNNs) are sequence classification models, composed of a neural network with one or more recurrent connections. Each time step t, the value of one or more layers of the network, known as the *state* $\mathbf{h}$, is updated by a function depending on the current input $\mathbf{x}_t$ and the value of the layer itself at the previous time step $\mathbf{h}_{t-1}$:

$$\mathbf{h}_t = \sigma(\mathbf{W}_h \mathbf{h}_{t-1} + \mathbf{W}_x \mathbf{x}_t); \quad (1)$$

where $\mathbf{W}_h$ and $\mathbf{W}_x$ are weight matrices and σ is an element-wise sigmoid function. This lets the network to "encode" all the inputs of the previous time steps into a fixed dimensional vector. From a dialogue management perspective, this can be interpreted as encoding all dialogue history up to the current turn. RNNs have been shown to be a powerful DST model performing competitively in the DSTCs [20, 19]. One of the shortcomings of RNNs is the difficulty in learning long-term dependencies due to the issue known as vanishing gradient [21]. Long-short term memory networks (LSTM) [22] address this issue by maintaining an additional *cell* state to store long term information and using a series of gates depending on $\mathbf{h}_{t-1}$ and $\mathbf{x}_t$ to update the information stored in the cell. LSTMs have been applied to DST with promising results [23].

2.1.1. DST feature extension

Historically, dialogue management has used a very defined data flow (shown in Fig. 1 as the continuous line), starting with the user utterance (acoustic signal), being transformed to a string of words by the ASR and to a set of concepts by the SLU, then being feed to the state tracker of the dialogue manager, and so on. In this architecture, each module reduces the data dimensionality. However, some useful information could be lost in each step. One motivation for this architecture was the need to obtain an input feature set small and decorrelated, to maintain the generative state tracker independence assumptions [16]. As discriminative models are better able to handle high dimensional, possibly correlated, input features, the tracker's input features can be augmented with features extracted in previous modules of the dialogue system. The ability to handle high dimensional input features is especially interesting in personalised dialogue management, since the dialogue features can be extended with user specific features such as acoustic or ASR-related features. These features give useful information that represent a certain type of speaker behaviour, which allows to relate it to the behaviour observed on "similar" source speakers. We propose to modify the usual dialogue data flow including features extracted directly from the acoustic signal and from the ASR (Fig. 1). In dysarthric user oriented state tracking of a environmental control interface, *iVectors* [24] are used as acoustic features and ASR performance-related features as ASR features (sec. 3.3).

2.1.2. Dropout regularization

The dialogue data gathered from the source speakers used for training might have been generated following a different distribution than the data the target speaker will generate. As mentioned in section 1, if the distribution between the source (training) data and the target (test) data differ, the tracker might overfit to the source data, thus performing poorly on the target. To address this issue, we propose to use *dropout* regularization [15], which has been proven to be a powerful regularization technique for neural networks. A percentage of neurons is randomly "deactivated" in each layer at every training iteration, forcing neurons to learn activation functions independent of other neurons. But as RNNs and especially LSTMs are difficult to train, dropout can make it more complicated to learn long term dependencies [25]. To avoid this issue, dropout is only applied in the non-recurrent connections between layers as proposed in [26].

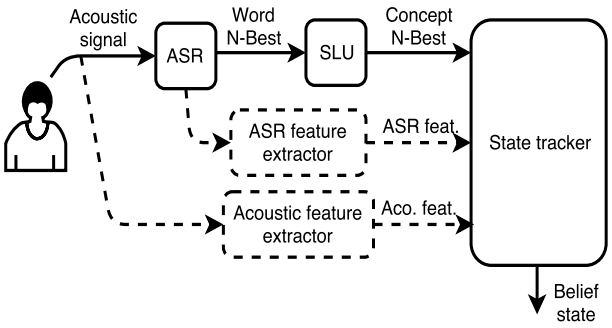

Figure 1: *Typical dialogue data flow (continuous line) and proposed extended dialogue data flow (dashed line)*

3. Experimental setup

To test the system in a scenario with high variability between the dynamics of the speakers, the experiments are performed within the context of a voice-enabled control system designed to help speakers with dysarthria to interact with their home devices. The user can interact with the system in a mixed initiative way, speaking single-word commands[1] from a set of 36 commands. As the ASR is configured to recognise single words, the SLU operates a direct mapping from the ASR output, an N-Best list of words, to an N-Best list of commands. The dialogue state of the system is factorized into three slots, with the values of the first slot representing the devices to control (TV, light, bluray...), the second slot its functionalities (channel, volume...) and the third slot the actions that these functionalities can perform (up, two, off...). The slots have 4, 17 and 15 values respectively, and the combination of the values of the three slots compose the joint goal (e.g. TV-channel-five, bluray-volume-up). The set of valid[2] joint goals $\mathcal{G}$ has a cardinality of 63, and the belief state for each joint goal g is obtained by multiplying the slot probabilities of each of the individual slot values and normalising:

$$P(g) = \frac{P_{s1}(g_1)P_{s2}(g_2)P_{s3}(g_3)}{\sum_{h \in \mathcal{G}} P_{s1}(h_1)P_{s2}(h_2)P_{s3}(h_3)} \quad (2)$$

where $P_{sx}(g_x)$ is the probability of the value g_x in slot s_x and $g = (g1, g2, g3)$.

3.1. Dialogue corpus collection

One of the main problems in dialogue management research is the lack of annotated dialogue corpora and the difficulty of using data from one domain for training a system in a different domain. The corpora released for the first three DSTCs aimed

[1] Severe dysarthric speakers cannot articulate complete sentences.

[2] Take into account that many combination of slot values wont be valid, e.g. light-channel-on

to mitigate this problem. However, they have been collected in a scenario where many different speakers interact only a few times, thus making adaptation to specific speakers infeasible. Furthermore, there is no acoustic data available, hence, features extracted from the acoustics cannot be used. For these reasons, a large part of dialogue management research relies on *simulated users* (SU) [27, 28, 29] to collect the data needed. The dialogue corpus used in the following experiments has been generated with simulated users interacting with a rule based dialogue manager. To simulate data collected from several dysarthric speakers during a large number of interactions from each user, a set of SUs with dysarthria has been created. As stochastic factors influence the corpus generation (Simulated user, stochastic policy), three different corpora have been generated with different random seeds. To reduce the effects introduced by the random components, the results presented are the mean results of the tracking evaluation on the three corpora. 1200 dialogues are collected for each speaker for each seed.

3.1.1. Simulated dysarthric users

Each SU is composed of a *behaviour simulator* and an *ASR simulator*. The *behaviour simulator* decides on the commands uttered by the SU in each turn. It is rule-based and depending on the machine action, it chooses a command corresponding to the value of a slot or answers a confirmation question. To simulate confusions by the user, it uses a probability of producing a different command, or of providing a value for a different slot than the requested one. The probabilities of confusion vary to simulate different expertise levels with the system. Three different levels are used to generate the corpus to increase its variability.

The *ASR simulator* generates an ASR N-Best list given the true user action. It is data driven and to train the ASR simulator for users with different dysarthria severities, data from a dysarthric speech database (UASpeech database [30]) has been used. This database includes data from 15 speakers with dysarthria severities clustered in 4 groups depending on their intelligibility: 4 very low, 3 low, 3 medium and 5 high. For more details on the ASR simulator, the reader may refer to [31].

3.1.2. Rule-based state tracker

One of the trackers used in the DSTCs as baseline [32] has been used to collect the corpus. This baseline tracker performed competitively in the DSTCs, proving the difficulty for data driven trackers when the training and test data are mismatched. The state tracking accuracy of this tracker is also used as the baseline in the following experiments.

3.1.3. Rule-based dialogue policy

The dialogue policy used to collect the corpus follows simple rules to decide the action to take in each turn: For each slot, if the maximum belief of that slot is below a threshold the system will ask for that slot's value. If the belief is above that threshold but bellow a second one, it will confirm the value, and if the maximum beliefs of all slots are above the second threshold it will take the action corresponding to the joint goal with the highest probability. The thresholds values are optimized by grid search to maximize the dialogue reward. In addition, the policy implements a stochastic behaviour to induce variability in the collected data; choosing a different action with probability p and requesting the values of the slots in a different order. The corpus is collected using two different policy parameter sets.

3.2. LSTM-based state tracker

The methods proposed in section 2.1.1 and 2.1.2 to improve generalization to new speakers are tested on a set of LSTM-based state trackers. To simulate the setting up of a system where dialogue data from the target speaker is not available, the tracker for each speaker is trained on data from the remaining 14 source speakers. 1200 dialogues are used for each source speaker (with a 0.9-0.1 train-validation split) and is tested with 1200 target speaker dialogues. In a second set of experiments, target speaker dialogues are included in the training data in different amounts. The target speaker dialogues used for training and testing are independent. The models are trained for 100 iterations with stochastic gradient descent and the five models corresponding to the five iterations performing best in the validation set are combined to get the slot output distribution.

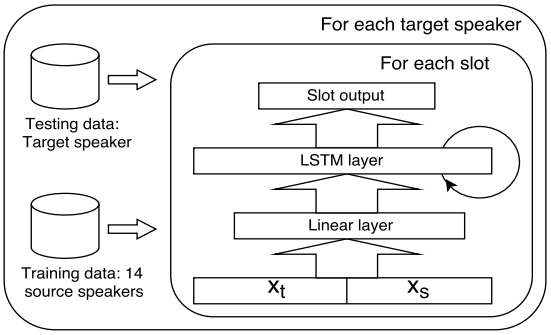

Figure 2: *Topology of the LSTM-based tracker*

3.2.1. Different LSTM models

The topology of the network is shown in Fig. 2, in which for each slot the turn input (the N-Best ASR output concatenated with the speaker features (if any), see section 3.3) is put to a linear projection layer that in turn feeds into a recurrent LSTM layer. The output of this layer is the input to a softmax layer with a size equal to the number of slot values. Two different linear-LSTM layer sizes have been tested[3]: 25-75 (SML) and 75-150 (LRG). Each model is evaluated with and without using dropout in training, with dropout rates of 20% in the input connections and 50% in the rest. This defines a total of four LSTM-based trackers evaluated in section 4, named SML, SML-DO, LRG and LRG-DO respectively.

3.3. Extended input features

The standard input features of the tracker in each turn $\mathbf{x}_t$ are the dialogue features, i.e. the N-best list of commands outputted by the ASR plus the system action in turn t. In addition, the models are evaluated concatenating the dialogue features with the following speaker features $\mathbf{x}_s$:

IV: In [33] it was shown that *iVectors* [24] can be used to predict the intelligibility of a dysarthric speaker. For each speaker s, $\mathbf{x}_s$ is a 50 dimensional vector corresponding to the mean *iVector* extracted from each utterance from that speaker. For more information on the *iVector* extraction, refer to [34].

APW: The statistics of the ASR can be used as speaker features. In this paper, the accuracy per word (command) is used, defining $\mathbf{x}_s$ as a 36 dimensional vector where each element is the ASR accuracy for each of the 36 commands[4].

IV+APW: The concatenation of *APW* and *IV* features.

[3]The reason to compare LSTMs with different sizes is because dropout reduces the effective size of the network [15], thus optimal network sizes might vary depending on the dropout rate. Several network sizes were tested, and the two with better performance are presented.

[4]This is computed on the *enrolment data*, a small set of commands recorded from the user when the system is set up [14].

Tracker		Speaker features: no feat.	IV	APW	IV+APW
Baseline	acc.	64.85	-	-	-
	L2	0.667	-	-	-
SML	acc.	66.93	65.21	66.60	67.17
	L2	0.482	0.501	0.484	0.483
SML-DO	acc.	67.31	68.77	70.17	70.60
	L2	0.451	0.427	0.418	0.408
LRG	acc.	66.12	66.24	66.50	68.63
	L2	0.497	0.505	0.489	0.464
LRG-DO	acc.	67.42	69.72	69.75	70.05
	L2	0.459	0.427	0.424	0.417

Table 1: *State tracking accuracy (%) and L2 results for the different trackers using different speaker features. SML (25-75) and LRG (75-150) are the size of the layers, and DO indicates that dropout is used. IV are iVectors and APW accuracy per word features.*

4. Results

The performance of the state trackers is evaluated on 8 different SUs corresponding to the speakers with ASR accuracy between 40% and 90%[5]. State tracking accuracy and L2 measure are used as metrics, following scoring *schedule 2* of the DSTCs [5].

Table 1 shows results when the trackers are trained with data from the source speakers only. The first row is the performance of the baseline tracker and the rows below compare the 4 LSTM-based trackers. The columns denote the features used in the input. It can be seen that the performance of the baseline tracker is only between 1% and 3% absolute below the performance of all the LSTM trackers when not using speaker specific features. This shows that the baseline tracker can compete with machine learned models in mismatched train-test data conditions, even in challenging ASR environments. Without dropout, using APW and IV features degrades the tracker performance in the case of SML network, and shows insignificant improvement for LRG. However, the concatenation of both features increases the performance slightly in SML and for more than 2% in LRG. Analysing the results speaker by speaker (not included in this paper for space reasons), it can be observed that, depending on the speaker, APW and IV features independently can degrade the performance. This suggests that for some speakers the best similarity measure with other speakers are APW features, and IVs for others. By combining both, the LSTM tracker is able to learn which features work better for a certain type of speaker. Dropout regularization improves the results of SML-DO and LRG-DO without using speaker features, but the performance increase is considerably more pronounced when APW or IV features are used, with improvements between 1.5% and 3% absolute. Combining the features and dropout gives the largest improvement with respect to the baseline, 5.75%, and 3.67% with respect to the LSTM tracker not using dropout or extended features. This shows that extending the networks input increases the chance to overfit, because neurons learn co-adaptations that only work for the training data. By using dropout, these co-adaptations can not be learnt because the presence of any particular input is unreliable. The improvement on accuracy and L2 measures given by the extended features is highly correlated.

In figure 3, the performance of the SML and SML-DO trackers when different amounts of user specific dialogues are included in the training set is shown. The results are presented

[5]In [31] it was shown that, for high intelligibility speakers, the ASR accuracy is above 90% so the improvement obtained from dialogue management is small, and for some very low intelligibility speakers, the ASR accuracy is too low to get any useful performance.

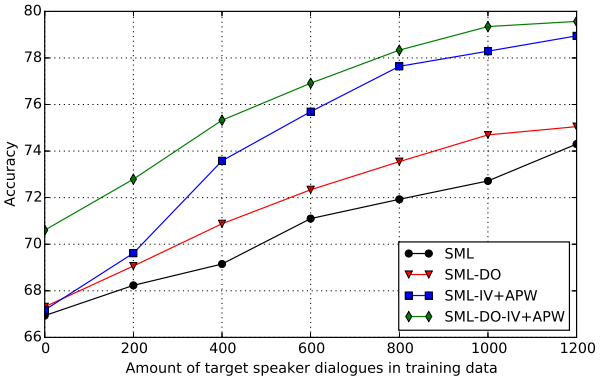

Figure 3: *Accuracy for SML tracker, using different amounts of target speaker dialogues in the training data. DO indicates that dropout regularization is applied and IV+APW that the concatenation of IV and APW features is used.*

with no speaker features and IV+APW features. The improvement obtained from speaker features increases when the target speaker dialogues are included in the training set, obtaining more than 4% absolute improvement compared with not using speaker features for any amount above 400 dialogues. When a small number of target speaker dialogues are included in the training set, the gain obtained from the combination of speaker specific features and dropout regularization (SML-DO-IV+APW) is significantly higher than any of these approaches alone (e.g. 3% with 200 dialogues). As more target speaker data is included in the training set, the gain obtained from the IV+APW features is increased with respect to the gain obtained from dropout, even if SML-DO-IV+APW still performs around 1% better. This shows how dropout helps to generalize when little or no speaker specific data is available for training, while, as more speaker specific data is included in the training set, the speaker features can work without the need for dropout.

5. Conclusions

In this paper, speaker specific features extracted from the raw acoustics and from the ASR were used to train an LSTM-based state tracker personalised to a target speaker, when training data from that speaker is not available. Dropout regularization showed to significantly increase the DST accuracy gained by including the features. It was shown that the improvement obtained with speaker features is larger when small amounts of data from the target speaker become available. Results where presented for an environmental control system designed for dysarthric speakers, but the features have the potential to be used with normal speakers too. Data from only 15 different speakers was used in this study. Having access to data from more source speakers could increase the chance of finding speakers "similar" to the target, which might increase the effectiveness of this method. Two types of features were used in this study, both related to time-invariant speaker characteristics. Feature-rich discriminative DST opens up the possibility of using numerous different features extracted from the acoustics or the ASR, such as features related to the noise, to the quality of the utterance, or to words appearing in the ASR output.

6. Acknowledgements

The research leading to these results was supported by EPSRC Programme Grant EP/I031022/1 (Natural Speech Technology). The authors would like to thank David Martínez for providing the *iVectors* used in this paper.

7. References

[1] S. Young, M. Gašić, B. Thomson and J. D. Williams, "POMDP-Based Statistical Spoken Dialog Systems: A Review". *Proceedings of the IEEE*, 2013.

[2] J. Williams and S. Young. "Partially observable Markov decision processes for spoken dialog systems". *Computer Speech and Language*, 2007.

[3] M. Gašić and S. Young. "Gaussian Processes for POMDP-based dialogue manager optimisation". *IEEE Transactions on Audio, Speech and Language Processing*, 2014.

[4] J. Williams, A. Raux, D. Ramachandran, and A. Black. "The dialog state tracking challenge". *Proceedings of the SIGDIAL 2013 Conference*, 2013.

[5] M. Henderson, B. Thomson, and J. Williams. "The second dialog state tracking challenge". *15th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 2014.

[6] M. Henderson, B. Thomson, and J. D. Williams. "The third dialog state tracking challenge". *Spoken Language Technology Workshop (SLT)*, 2014.

[7] S. Lee and M. Eskenazi. "Recipe For Building Robust Spoken Dialog State Trackers: Dialog State Tracking Challenge System Description". *Proceedings of the SIGDIAL 2013 Conference*, 2013.

[8] C. J. Leggetter and P. C. Woodland. "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models". *Computer Speech and Language*, 1995.

[9] S. Chandramohan, M. Geist, F. Lefevre and O. Pietquin. "Coadaptation in Spoken Dialogue Systems". *Proceedings of the 4th International Workshop on Spoken Dialogue Systems (IWSDS)*, 2012.

[10] I. Casanueva, T. Hain, H. Christensen, R. Marxer, and P. Green. "Knowledge transfer between speakers for personalised dialogue management." *16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 2015.

[11] A. Genevay, and R. Laroche. "Transfer Learning for User Adaptation in Spoken Dialogue Systems". *Proceedings of the 2016 International Conference on Autonomous Agents and Multiagent Systems*, 2016.

[12] H. Christensen, S. Cunningham, C. Fox, P. Green, and T. Hain, "A comparative study of adaptive, automatic recognition of disordered speech". *Proceedings of Interspeech*, 2012.

[13] H, Christensen, I. Casanueva, S. Cunningham, P. Green, and T. Hain. "Automatic selection of speakers for improved acoustic modelling: recognition of disordered speech with sparse data". *Spoken Language Technology Workshop (SLT)*, 2014.

[14] H. Christensen, I. Casanueva , S. Cunningham, P. Green, and T. Hain, "homeService: Voice-enabled assistive technology in the home using cloud-based automatic speech recognition". *4th Workshop on Speech and Language Processing for Assistive Technologies(SLPAT)*, 2013.

[15] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov. "Dropout: A simple way to prevent neural networks from overfitting". *The Journal of Machine Learning Research*, 2014.

[16] B. Thomson, and S. Young. "Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems". *Computer Speech and Language*, 2010.

[17] S. Lee. "Structured discriminative model for dialog state tracking". *Proceedings of the SIGDIAL 2013 Conference*, 2013.

[18] J. Williams. "Web-style Ranking and SLU Combination for Dialog State Tracking". *Proceedings of the SIGDIAL 2014 Conference*, 2014.

[19] M. Henderson, B. Thomson and S. Young. "Robust dialog state tracking using delexicalised recurrent neural networks and unsupervised adaptation". *Spoken Language Technology Workshop (SLT)*, 2014.

[20] M. Henderson, B. Thomson and S. Young. "Word-Based Dialog State Tracking with Recurrent Neural Networks". *Proceedings of the SIGDIAL 2014 Conference*, 2014.

[21] Y. Bengio, P. Simard and P. Frasconi. "Learning long-term dependencies with gradient descent is difficult". *IEEE Transactions on Neural Networks*, 1994.

[22] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 1997.

[23] L. Zilka and F. Jurcicek. "Incremental LSTM-based dialog state tracker". *arXiv preprint arXiv:1507.03471.*, 2015.

[24] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel and P. Ouellet. "Front-end factor analysis for speaker verification". *IEEE Transactions on Audio, Speech, and Language Processing*, 2011.

[25] J. Bayer, C. Osendorfer, N. Chen, S. Urban and P. van der Smagt. "On fast dropout and its applicability to recurrent networks". *arXiv preprint arXiv:1311.0701*, 2013.

[26] W. Zaremba, I. Sutskever and O. Vinyals. "Recurrent neural network regularization". *arXiv preprint arXiv:1409.2329*, 2014.

[27] K. Georgila, J. Henderson and O. Lemon. "User simulation for spoken dialogue systems: learning and evaluation". *proceedings of INTERSPEECH*, 2006.

[28] J. Schatzmann, B. Thomson, K. Weilhammer, H. Ye and S. Young. "Agenda-based user simulation for bootstrapping a POMDP dialogue system". *Human Language Technologies*, 2007.

[29] B. Thomson, M. Gasic, M. Henderson, P. Tsiakoulis and S. Young. "N-best error simulation for training spoken dialogue systems". *Spoken Language Technology Workshop (SLT)*, 2012.

[30] H. Kim, M. Hasegawa-Johnson, A. Perlman, J. Gunderson, T. Huang, K. Watkin, and S. Frame, "Dysarthric speech database for universal access research". *Proceedings of Interspeech*, 2008.

[31] I. Casanueva, H. Christensen, T. Hain, and P. Green, "Adaptive speech recognition and dialogue management for users with speech disorders". *Proceedings of Interspeech*, 2014.

[32] Z. Wang, and O. Lemon. "A simple and generic belief tracking mechanism for the dialog state tracking challenge: On the believability of observed information". *Proceedings of the SIGDIAL 2013 Conference*, 2013.

[33] D. Martínez, P. Green, and H. Christensen. "Dysarthria intelligibility assessment in a factor analysis total variability space". *Proceedings of Interspeech*, 2013.

[34] D. Martínez, E. Lleida, P. Green, H. Christensen, A. Ortega, and A. Miguel. "Intelligibility Assessment and Speech Recognizer Word Accuracy Rate Prediction for Dysarthric Speakers in a Factor Analysis Subspace". *ACM Transactions on Accessible Computing (TACCESS) 6*, 2015.