Efficient Evaluation of Task Oriented Dialogue Systems

Weiyi Lu
weiyilu@amazon.com

Yi Xu
yxaamzn@amazon.com

Li Erran Li
lilimam@amazon.com

Amazon Alexa AI
Seattle, WA 98101

Abstract

Smart voice assistants have gained much popularity in the past years. People can leverage them to accomplish a variety of daily tasks nowadays. To provide great services and ensure satisfactory user experiences, it is crucial to continuously measure and monitor how the assistant performs. One metric for such purposes is called goal success rate (GSR), which measures how often the assistant successfully fulfills a user's goal. In order to generate annotations for GSR calculation, human labelers need to examine randomly sampled goals from the users, where a goal consists of consecutive utterances in which the user attempts to direct the assistant to accomplish a particular task. A key challenge here is to identify all the relevant contextual utterances that make up a goal. As we will demonstrate, an existing rule based solution incurs substantial wasted labeling efforts, and also introduces potential bias into the GSR metric. Inspired by related work in question answering, we propose a BERT-based span prediction model to optimize the identification of relevant contextual utterances. Through our experiments, we show that the proposed model consistently reduces labeling waste by 5%-10%.

1 Introduction

In the past decade, we witnessed the debut of a variety of smart voice assistants (e.g. Google's Google assistant, Amazon's Alexa, Apple's Siri, and Microsoft's Cortana etc.), and since then their popularity has been rapidly growing. Now these assistants support a wide range of skills such as setting alarms, playing music, etc. In order to ensure the assistants deliver their skills smoothly, it is pivotal to measure and continuously monitor how well they perform. Goal success rate (GSR) is a metric designed towards such end, and it aims to measure how good an assistant is at achieving the users' goals. A goal is defined as a sequence of utterances where the user is trying to direct the assistant to accomplish some task. To calculate GSR, ideally we randomly sample goals from the traffic that the assistant receives, and then recruit human labelers to examine if the assistant succeeds in accomplishing each of the sampled goals. From the annotations, we can then derive an unbiased estimate of the assistant's performance.

However, a goal here is clearly just an artificial construct, and the traffic that the assistant receives is made up of individual utterances (snippets of transcribed speech). Therefore, we cannot directly sample goals from the traffic. Instead, the process starts by first identifying a set of relevant utterances. For example, if we are interested in understanding how well the assistant performs the task of setting alarms, we will first sample utterances where the user expresses an intention to set an alarm. We call these the target utterances. Then for each target utterance, we will look at its surrounding utterances and identify the ones that are relevant to the goal expressed in the target utterance. The target utterance and the relevant surrounding utterances are then bundled together as a goal.

NeurIPS 2020 Workshop on Human in the Loop Dialogue Systems (HLDS), Vancouver, Canada.

In order to maximize the efficiency of the annotation process, we would like to automate the goal sampling process. However, it is not trivial to correctly identify relevant contextual utterances for given target utterances. As a baseline approach, we have adopted a simple rule that expands from the target utterance to include surrounding utterances until the stopping criterion is met, which is determined by 3 parameters (details in Section 3). In practice, to avoid missing out relevant contexts for the target utterances, we have tuned the parameters such that the rule is overly inclusive. However, this creates significant labeling waste as it can bring in many irrelevant utterances. To mitigate this, we modified the annotation process as follows (rather than calling the outcome of this rule a goal, we call it a session). When the labelers are presented with such a session, instead of only focusing on the goal around the target utterance, we ask them to first identify and segment out all goals present in the session, and then perform GSR annotation for all the goals present.

However, before the labelers perform goal segmentation, they need to transcribe and annotate the intention for each utterance in the session. As a result, the annotation process is still prone to wasted labeling efforts. In particular, since each batch of the GSR annotation focuses on a specific goal type, if any of the goals in a session is not of the desired type, the labeling efforts are all wasted in such cases.

In addition, there is another drawback of using the rule-based approach to build sessions, which is that it could potentially introduce bias into the GSR metric. This is because even though the target utterances are sampled randomly from the traffic, when we expand from a target utterance into a session containing multiple goals, all of these goals come from the same user. This reduces the randomness in the sampling process and may be a potential source of bias.

To overcome these shortcomings, we propose an approach called Context Optimization Point (COP), which employs a deep learning model to intelligently identify relevant contextual utterances for a given target utterance. We still call the outcome of the model a session, and incorporate the model with the modified annotation workflow (where the labelers will perform goal segmentation before GSR annotation). But the objective here is for the session to only include the goal around the target utterance. This way, we expect to reduce the amount of wasted labeling, as well as eliminate the concern of a biased GSR metric.

We summarize our contributions as follows.

1. To the best of our knowledge, we are the first to propose a deep learning model for automatically identifying dialogue boundaries in the live traffic that voice assistants receive, thus improving the efficiency of the human evaluation process.
2. We formulate the problem of predicting contextual utterances as a span prediction task, and tackle it through a model leveraging pretrained BERT (Devlin et al., 2019). We show that the model is able to consistently reduce labeling waste (5%-10% increase in yield rate as shown in Section 4).
3. We identify a data imbalance problem whereby there are many more shorter goals, i.e. goals that consist of only a few utterances, than longer ones, which causes the performance of the initial model to quickly degrade as goal size increases. We propose a simple yet effective reweighting scheme, which fixes this issue.

2 Related Work

There has been a long history behind the study of evaluation methods for task oriented dialogue systems, and many automated methods have been proposed for efficient evaluation. They can be roughly grouped into two categories, namely user satisfaction modeling and user simulation (Deriu et al., 2020). The assumption behind user satisfaction modeling is that the performance of a dialogue system can be approximated by how satisfied the users are after interacting with the system. The goal is then to build models that can predict user satisfaction scores given the dialogues (Walker et al., 1997; Engelbrecht et al., 2009a; Higashinaka et al., 2010; Schmitt and Ultes, 2015; Hara et al., 2010). In user simulation, methods are developed to simulate the users' behavior, which are used to train the dialogue systems, as well as to evaluate how the system performs (Schatzmann et al., 2006, 2007; Kreyssig et al., 2018; Schatzmann et al., 2005; Engelbrecht et al., 2009b; Möller et al., 2006). For a comprehensive review of evaluation methods for task oriented dialogue systems, see Section 3 in Deriu et al. (2020) and references therein. The main difference between these previous works and our work is that we do not aim to build a model to perform evaluation automatically. Rather, the goal of our model is to identify dialogue boundaries in the live traffic that voice assistants receive, so that we can improve the efficiency of the human annotation process. The annotations in turn can be used to train the models from these previous works for automatically evaluation. In this sense, our work can be considered orthogonal to the previous works.

For our specific task, a relevant topic is coherence modeling (Barzilay and Lapata, 2008; Nguyen and Joty, 2017). In this task, the goal is to distinguish a coherent text from a random sequence of sentences, such that the coherent text binds the sentences together to express a meaning as a whole. Another related area is topic segmentation. Early work on this subject can be traced back to Marcus and Reynar (1998) and the references therein. Later work such as Purver (2011); Joty et al. (2013); Arnold et al. (2019) adopt various modeling techniques such as Lexical Cohesion-based Segmenter, Latent Dirichlet Allocation, and neural network models.

Finally, the topic that is most directly related to our proposed model is the question answering (QA) task (Bouziane et al., 2015; Rajpurkar et al., 2016). In a typical setup, the inputs given are a question and a paragraph of context. The task is to identify a span of text in the context that answers the question. In our case, we have a target utterance, and a sequence of utterances surrounding the target, from which we want to identify a span of utterances that together make up the goal that the target utterance belongs to. The main difference is that in QA, the boundary of the span is at the token level, while in our case it is at the utterance level.

3 Context Optimization Point Model

Baseline Approach As mentioned in the introduction, the baseline method that we have adopted for creating sessions for GSR annotation relies on a simple rule. Specifically, the rule starts from the target utterance and expands to surrounding utterances based on the following 3 parameters:

1. Maximum time difference (Stops when the time difference between the surrounding utterance and the target is beyond this number).
2. Maximum number of utterances to include.
3. Minimum number of utterances to include (This is not always used).

Given the above rule, the overall annotation workflow for GSR is as follows:

1. Decide on a goal type of interest and sample a set of target utterances. The selection is determined by a spoken language model that predicts the domain (related to goal) of each utterance.
2. For each target utterance, use the rule to bring in surrounding utterances and build a session.
3. For each session, labelers will first transcribe each utterance and annotate the user's intention in each utterance. Then based on the transcription and intention annotation, labelers will segment out all the goals present, and mark each goal as either fulfilled or not.

As we can see, the second step plays a key role in ensuring the efficiency and robustness of the annotation process. The rule-based approach often times brings in too many surrounding utterances when producing a session, leading to labeling wastes and bias in the GSR metric. Motivated by this, we propose a deep learning model that can intelligently identify relevant contexts for a given target utterance, which we describe below.

Problem Formulation As mentioned in the related work section, our proposed model is inspired by the span prediction models in the QA task. To format the inputs, we first treat the target utterance as the question. Then in order to get the context, we expand from the target and find utterances that come before and after the target on the same device within certain time frame (in practice, we use the session built by the simple rule for this purpose). As such, the problem becomes identifying a consecutive span of utterances around the target utterance that makes up the targeted goal. We name our model as the context optimization point (COP) model.

More formally, we formulate the problem as follows. The inputs we are given are (1) a sequence of utterances, $u_1, \ldots, u_m$ (where there is a total of m utterances and each utterance u_i consists of a sequence of words, which are outputs of an speech recognition model, and (2) the index of the target utterance t. The task is to predict a pair of indices (i, j) with $i \leq j$, which corresponds to the starting index and the ending index of utterances that belong to the same goal as the target utterance.

Our model is based on a BERT sequence-to-sequence model (Devlin et al., 2019), with modifications to the input and output layers. Figure 1 shows an overview of the architecture of the model through an example. Below, we provide more details on the input and output layers.

Model Architecture As an example, suppose we are given four utterances shown in Table 1, and the second utterance is the target. Figure 1 shows the architecture of our proposed COP model using the example utterances. Note that, wakeword is a word spoken by the customer to wake up the device. For example, for Google assistant it can be "Hey Google" or "OK Google". For Alexa, it can be "Alexa" or "Echo".

Index	Utterance
1	wakeword order some apples
2	wakeword play sanderson interlude by brent faiyaz (target utterance)
3	wakeword stop
4	wakeword turn on the light

Table 1: Example utterances. Wakeword can be "Hey Google", "Alexa", etc.

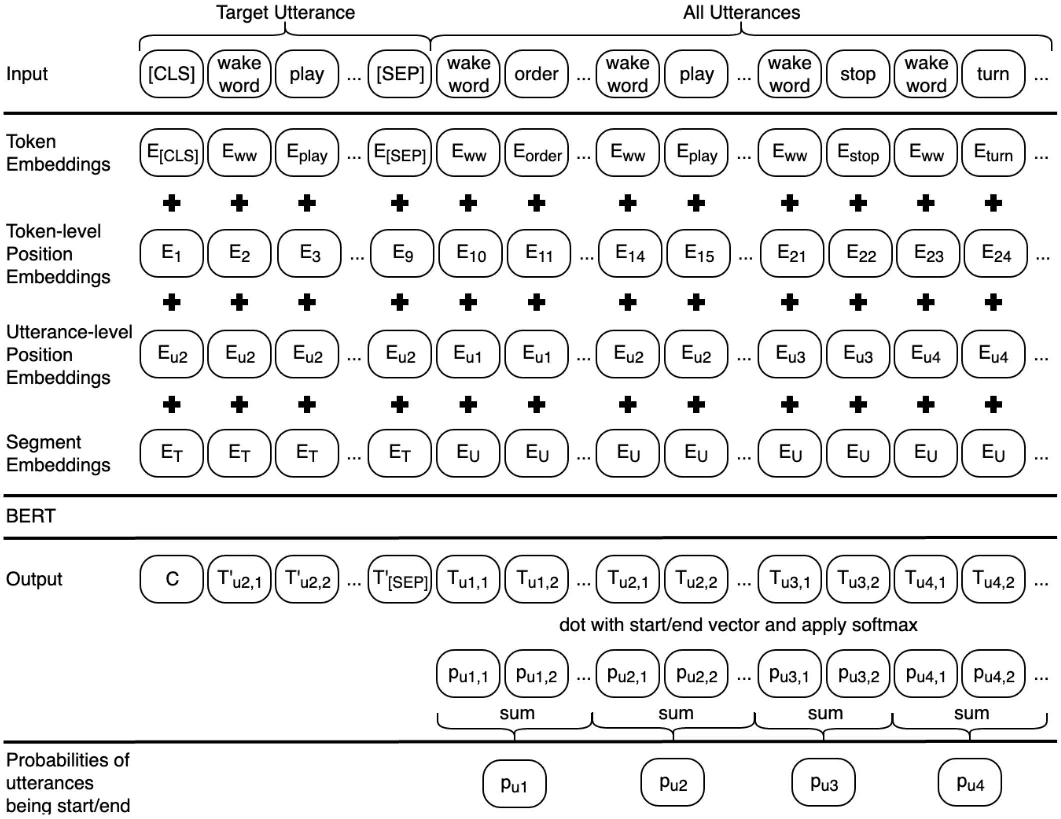

Figure 1: COP Model Architecture

Input Representation Given the input utterances, we will first format them into a sequence as shown in the first row of Figure 1. The sequence is made up of a [CLS] token, the target utterance, a [SEP] token, and finally a concatenation of all the utterances. Given this input sequence, four embedding sequences are generated:

1. Token embeddings: These are pretrained embeddings for the tokens in the input vocabulary.
2. Token-level position embeddings: These are embeddings for the absolute position of each token in the input sequence, which are also pretrained.
3. Utterance-level position embeddings: These are learnable embeddings for the absolute position of each utterance in the session.
4. Segment embeddings: These consist of two embedding vectors E_T and E_U, which correspond to the target utterance segment (all the tokens up until `[SEP]`) and the context segment (everything after `[SEP]`), respectively.

The four sequences are then summed and passed through BERT to obtain the output sequence.

Output and Objective Given the output sequence, we will only need the representations of the context segment, namely those corresponding to all the utterances. For utterance u_i, suppose the corresponding output representations are $T_{(u_i,1)}, \ldots, T_{(u_i,n_i)}$. In order to predict the start and end utterances, we make use of a start vector S and an end vector E, which will have the same dimension as the output representations.

Then, the probabilities of u_i being the start and being the end are given respectively by the following formula.

$$p_i^s = \frac{\sum_{j=1}^{n_i} \exp\left(S \cdot T_{u_i,j}\right)}{\sum_{i'=1}^{m} \sum_{j'=1}^{n_{i'}} \exp\left(S \cdot T_{u_{i'},j'}\right)} \tag{1}$$

$$p_i^e = \frac{\sum_{j=1}^{n_i} \exp\left(E \cdot T_{u_i,j}\right)}{\sum_{i'=1}^{m} \sum_{j'=1}^{n_{i'}} \exp\left(E \cdot T_{u_{i'},j'}\right)} \tag{2}$$

For training, suppose the correct index pair is (i', j'). The objective is defined as $\log p_{i'}^s + \log p_{j'}^e$. And for inferencing, we choose the pair of indices (i, j) such that $i \leq t \leq j$ where t is the index of the target, and $p_i^s + p_j^e$ is maximized.

4 Experiments and Results

Dataset We collect data from a commercial voice assistant company. Each instance is a session around some target utterance, and the session is produced by the simple rule described in Section 1. The sessions have been annotated by human labelers, where for each session the labelers segment out and annotate each goal present. The goal type of interest here is shopping, i.e. the GSR annotations on these sessions will be used to measure how well the assistant fulfills the customers' shopping intents. In the end, we created 2 datasets:

1. In the first set, all sessions come from the US region, and there are 1.65M sessions in total. We randomly split them into train/validation/test sets, which contain 1.2M, 0.3M, 0.15M sessions, respectively.
2. In the second set, the data come from various English-speaking regions, including US, UK, Canada (CA), India (IN), and Australia and New Zealand (ANZ). This is to test whether the model can produce consistent results across different regions. In total, there are 3.8M sessions, which are split randomly into train/validation/test sets, containing 2.6M, 0.6M, 0.6M sessions respectively.

For the encoder, we choose TinyBERT (Jiao et al., 2020). It consists of 4 transformer layers with 12 attention heads each, and the embedding dimension and hidden dimension are 312, while the feedforward dimension is 1200. The model is trained with Adam (Kingma and Ba, 2015) using the default parameters and the batch size is 1024. Training stops until the match ratio metric (explained below) stops increasing for 5 consecutive epochs, and we choose the checkpoint from the epoch that has the best match ratio.

Table 2 shows the results of the models as evaluated on the first dataset. Below are the definitions of the metrics reported:

- **Ingestion size** is the total number of utterances in the sessions created.

	Rule	COP	COP Weighted
Ingestion Size	340K	244K	303K
Avg Session Size	2.33	1.67	2.08
Yield Rate	58.14	63.50	62.25
Match Ratio	100.00	94.87	98.97

Table 2: Results on the first dataset, which contains data from the US region only.

- **Avg session size** is the average number of utterances in each session.
- **Yield rate** is the proportion of utterances in the sessions that are useful for GSR calculation. For this dataset specifically, since they are used for calculating GSR for shopping intents, an utterance is considered useful if it belongs to some shopping goal.
- **Match ratio** tracks whether the goal around each target utterance is entirely contained in the sessions. The targeted goal is considered matched if all the utterances in the goal are contained in the generated session, otherwise the goal is considered lost. And match ratio is the fraction of goals that are matched.

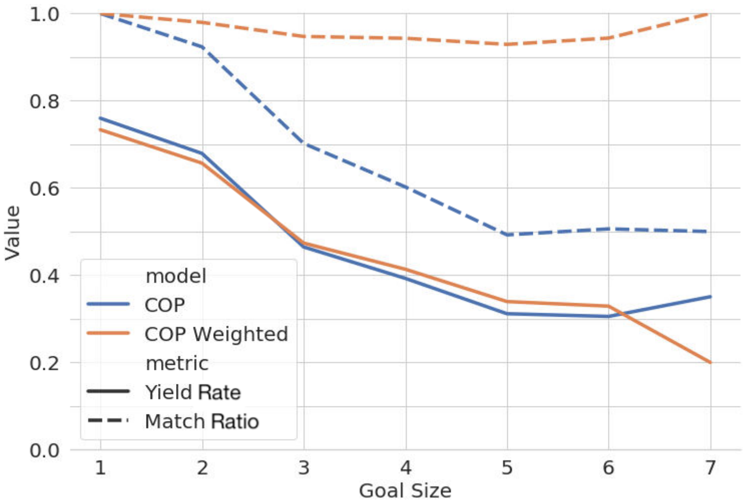

Figure 2: Metrics across different goal sizes. The size of a goal is defined as the number of utterances that the goal is made up of. Best viewed in color.

The first column in Table 2 contains the performance of the simple rule, which we want to compare against. The second column is our proposed COP model. We can see that our model achieves a 5.36% increase in yield rate, but there is a 5.13% drop in terms of match ratio. Upon further analysis, we find out that the drop is mainly due to the model being much worse at matching longer goals. In Figure 2, we plot the yield rate and match ratio against goal sizes. And as shown by the dashed blue line, the match ratio of the COP model decreases greatly as goals get longer. Therefore, we come up with a different version of the COP model where each training instance is reweighted such that longer goals would have higher weights. This results in the COP Weighted model in both Table 2 and Figure 2.

In order to calculate the weights, we first choose a base goal size (e.g. 7). For a goal with size greater than the base, we assign a weight of 1. For a goal with size i where i is less than or equal to the base size, the weight is determined by the following formula.

$$\text{weight}_i = \frac{\text{count of goals with size } i \text{ in training set}}{\text{count of goals with the base size in training set}} \tag{3}$$

This way, shorter goals are down-weighted. As we can see, COP Weighted is able to achieve comparable yield rate as COP, while improving substantially in match ratio (from 94.8% to 98.9%). Meanwhile, from the orange dashed line in Figure 2, we see that match ratio of COP Weighted is fairly consistent across different goal sizes, which means that the reweighting method is indeed effective in fixing the model's bias towards shorter goals.

Locale	Metric	Rule	COP Weighted
All regions	Ingestion Size	1.7MM	1.4MM
	Avg Session Size	2.86	2.40
	Yield Rate	54.27	61.71
	Match Ratio	100.00	98.41
US	Ingestion Size	744K	654K
	Avg Session Size	2.50	2.20
	Yield Rate	59.54	64.43
	Match Ratio	100.00	98.62
UK	Ingestion Size	412K	346K
	Avg Session Size	2.87	2.41
	Yield Rate	55.64	64.13
	Match Ratio	100.00	98.64
CA	Ingestion Size	199K	162K
	Avg Session Size	2.72	2.21
	Yield Rate	49.91	58.70
	Match Ratio	100.00	98.64
IN	Ingestion Size	351K	271K
	Avg Session Size	4.12	3.18
	Yield Rate	45.56	55.59
	Match Ratio	100.00	97.15
ANZ	Ingestion Size	47K	37K
	Avg Session Size	3.67	2.95
	Yield Rate	42.08	49.08
	Match Ratio	100.00	97.50

Table 3: Results of the rule baseline and our COP weighted model evaluated on the second dataset, which contains data from various English-speaking regions.

Finally, the results on the second dataset are shown in Table 3. Again our model outperforms the baseline by a large margin, with a 7.44% increase in yield rate in all English regions combined, while maintaining a match ratio of 98.41%. The improvement is also consistent across different regions, and noticeably in the IN region, there is a 10.03% increase in yield rate with a match ratio of 97.15%.

5 Conclusions and Future Work

In this paper, we tackle the key challenge to efficiently evaluate the performance of a task oriented dialogue using the GSR metric, which boils down to optimizing the contextual utterances given a target utterance. We propose a QA-style span prediction model based on BERT, and demonstrate its efficacy via performance comparison to the rule based baseline. We also address the imbalance in goal sizes in the data through a simple yet effective reweighting scheme. Having such a model allows us to save considerable labeling costs, as well as to reduce the bias in the GSR metric. In the future, we plan to address the issue when the model produces sessions that are too short and fail to completely include the goal around the target utterance. One potential solution is to modify the workflow such that the labelers are not restricted to seeing the utterances in the model-generated sessions only, but are able to expand and inspect utterances beyond the session. This would complement our proposed COP model and further improve the robustness of the annotation process.

References

Sebastian Arnold, Rudolf Schneider, Philippe Cudré-Mauroux, Felix A. Gers, and Alexander Löser. 2019. Sector: A neural model for coherent topic segmentation and classification. In *Association for Computational Linguistics*.

Regina Barzilay and Mirella Lapata. 2008. Modeling local coherence: An entity-based approach. In *Association for Computational Linguistics*.

Abdelghani Bouziane, Djelloul Bouchiha, Noureddine Doumi, and Mimoun Malki. 2015. Question answering systems: Survey and trends. In *Procedia Computer Science*.

Jan Deriu, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak. 2020. Survey on evaluation methods for dialogue systems. In *Artificial Intelligence Review*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Klaus-Peter Engelbrecht, Florian Gödde, Felix Hartard, Hamed Ketabdar, and S. Möller. 2009a. Modeling user satisfaction with hidden markov models. In *Special Interest Group on Discourse and Dialogue*.

Klaus-Peter Engelbrecht, Michael Quade, and Sebastian Möller. 2009b. Analysis of a new simulation approach to dialog system evaluation. In *Speech Communication*.

Sunao Hara, Norihide Kitaoka, and Kazuya Takeda. 2010. Estimation method of user satisfaction using n-gram-based dialog history model for spoken dialog system. In *International Conference on Language Resources and Evaluation*.

Ryuichiro Higashinaka, Yasuhiro Minami, Kohji Dohsaka, and Toyomi Meguro. 2010. Issues in predicting user satisfaction transitions in dialogues: Individual differences, evaluation criteria, and prediction models. In *International Workshop on Spoken Dialogue System Technology*.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xusong Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. Tinybert: Distilling bert for natural language understanding. In *Conference on Empirical Methods in Natural Language Processing*.

Shafiq R. Joty, Giuseppe Carenini, and Raymond T. Ng. 2013. Topic segmentation and labeling in asynchronous conversations. In *Journal of Artificial Intelligence Research*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Florian Kreyssig, Iñigo Casanueva, Pawel Budzianowski, and Milica Gasic. 2018. Neural user simulation for corpus-based policy optimisation for spoken dialogue systems. In *Special Interest Group on Discourse and Dialogue*.

Mitchell P. Marcus and Jeffrey C. Reynar. 1998. Topic segmentation: algorithms and applications. In *IRCS Technical Reports Series*.

Sebastian Möller, Roman Englert, Klaus Engelbrecht, Verena Hafner, Anthony Jameson, Antti Oulasvirta, Alexander Raake, and Norbert Reithinger. 2006. Memo: towards automatic usability evaluation of spoken dialogue services by user error simulations. In *Conference of the International Speech Communication Association*.

Tien Dat Nguyen and Shafiq R. Joty. 2017. A neural local coherence model. In *Association for Computational Linguistics*.

Matthew Purver. 2011. Topic segmentation. In *Spoken language understanding: systems for extracting semantic information from speech*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *Conference on Empirical Methods in Natural Language Processing*.

Jost Schatzmann, Matthew N.Stuttle, Karl Weilhammer, and Steve Young. 2005. Effects of the user model on simulation-based learning of dialogue strategies. In *IEEE Workshop on Automatic Speech Recognition and Understanding*.

Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. 2007. Agenda-based user simulation for bootstrapping a pomdp dialogue system. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Jost Schatzmann, Karl Weilhammer, Matthew Stuttle, and Steve Young. 2006. A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. In *Knowledge Engineering Review*.

Alexander Schmitt and Stefan Ultes. 2015. Interaction quality: Assessing the quality of ongoing spoken dialog interaction by experts - and how it relates to user satisfaction. In *Speech Communication*.

Marilyn A. Walker, Diane J. Litman, Candace A. Kamm, and Alicia Abella. 1997. PARADISE: A framework for evaluating spoken dialogue agents. In *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*.