DialogActs based Search and Retrieval for Response Generation in Conversation Systems

Nidhi Arora
Interactions, LLC
narora@interactions.com

Rashmi Prasad
Interactions, LLC
rprasad@interactions.com

Srinivas Bangalore
Interactions, LLC
sbangalore@interactions.com

Abstract

Designing robust conversation systems with great customer experience requires a team of design experts to think of all probable ways a customer can interact with the system and then author responses for each use case individually. The responses are authored from scratch for each new client and application even though similar responses have been created in the past. This happens largely because the responses are encoded using domain specific set of intents and entities. In this paper, we present preliminary work to define a dialog act schema to merge and map responses from different domains and applications using a consistent domain-independent representation. These representations are stored and maintained using an Elasticsearch system to facilitate generation of responses through a search and retrieval process. We experimented generating different surface realizations for a response given a desired information state of the dialog.

1 Introduction

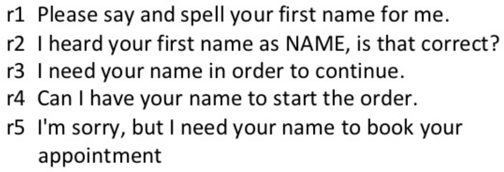
A good conversation system is the one that enables its users to converse freely in natural language text. To handle conversations in a robust manner, the system should have set of responses covering many possible ways the end-customer can interact with the system. Response generation is a challenging problem for spoken dialog systems, with the quality of the generator depending on factors such as adequacy, fluency, variation and readability (Stent et al., 2005). Partly because of the need to adapt to requirements of the specific domain for which the system is designed, many deployed applications, including the ones that provide context for our work, follow a template-based approach, in which response templates with possible slot fillers are manually authored by application designers. Such responses largely satisfy the quality measures of adequacy and fluency, where adequacy measures whether the response conveys the intended meaning completely, non-redundantly, and unambiguously, while fluency measures linguistic correctness and appropriateness of style. However, there can be challenges with respect to variation and readability.

Variation is intended to avoid repetitiveness so that the responses in multi-turn dialogs sound natural and human-like, while readability ensures that responses are interpretable in their dialog context. For example, asking users for basic personal information or task specific details is a common task across many business needs. However, repeatedly using the same small set of scripts such as `What is your account number?` or `Please share your account number` has the undesirable effect of sounding predictable and unnatural.

With a team of experts authoring prompts for diverse applications across different domains, collecting response variations for different response types for various stages of a conversation is effortless. The challenge however is that the collection is useful only if the variants are maintained with dialog state specific equivalence classes that are consistent across domains for authors to access and reuse. Fig 1 presents some alternative realizations for frequently occurring use cases of asking customers for their name. While $r1$ is simply querying the end user to provide their name, $r2$ acknowledges the capture of information and requests for confirmation of correctness. Responses $r3$ - $r5$ are used when the customer has not provided the desired information during the prior turn and needs to be prompted again; these are paraphrases providing reasons for why the information is necessary.

The motivation behind this work is to cluster these variations using a systematic and consistent

Proceedings of the 18th International Conference on Natural Language Processing, pages 564–572
Silchar, India. December 16 - 19, 2021. ©2021 NLP Association of India (NLPAI)



r1 Please say and spell your first name for me.
r2 I heard your first name as NAME, is that correct?
r3 I need your name in order to continue.
r4 Can I have your name to start the order.
r5 I'm sorry, but I need your name to book your appointment

Figure 1: Different variations of asking the end-user customer about their name.

framework to promote response sharing across different domains and to suggest possible variations to choose from. In this paper, we describe our preliminary work on dialog act classification of a large database of responses authored by a design team for different applications using a consistent and common domain independent annotation scheme. These response variants can then be used by the designers to provide possible alternative realizations of the content they want to convey. Furthermore, since selecting an appropriate variant is partly a function of the dialog context, our future goal is also to develop a context aware response suggestion model that can account for readability through the proper use of context-dependent elements such as referring expressions (e.g., *Please share your account number with me* vs. *Please share that with me*) and discourse markers (e.g., *What is your account number?* vs. *And what is your account number?*).

Our main contributions in this paper are:

- Creating a unified taxonomy for the communicative function of dialog acts that is universally applicable across different domains and covers probable agent/user tasks.

- Using Elasticsearch to maintain a repository of system responses and transforming context independent response generation problem into a search and retrieval task.

Section 2 presents our approach to Dialog Act (DA) classification, focusing on the taxonomy for the communication function (CF) component. Section 3 explains the use of Elasticsearch for maintaining a repository of utterances indexed along many different dimenions. Section 4 presents our preliminary experiments on classification and static response generation.

2 Communicative Function of Dialog Acts

The meaning of an utterance in dialog has long been characterized as a dialog act, designed to capture the communicative behavior of a participant in terms of speech acts (Austin, 1962; Searle, 1969). Many DA schemes have been developed over the years, but most were designed for specific domains and applications (Allen et al., 1994; Allen and Core, 1997a; Anderson et al., 1991; Alexandersson et al., 1997). Here, we adapt the ISO standard for DA annotation (Bunt et al., 2012) because it provides a domain independent representation that covers a broad range of intents for all aspects of a conversation state.

Based on the information-state update approach to meaning in dialogue (Bunt, 2000; Traum and Larsson, 2003), a DA in the ISO framework (ISO-DA) has two main components: a semantic content (SC), which describes the entities, events, actions, properties or relations that the DA is about, and a communicative function (CF), which specifies how addressees should update their information state with the semantic content. For example, the utterance *What is your account number?* has some representation of the customer's account number as the semantic content, while the CF should represent the fact that the value of this entity is not known and that it is being requested of the customer.

Since the focus here is on the CF classification of responses, we have adapted the ISO-DA CFs to reflect the commonly occurring functions in our data of approximately 37K unique response utterances. The following provides the CFs, with definitions and examples. Broadly, the functions can be classified as the ISO-DA categories of information-seeking functions, information-providing functions, commissives and directives. For some of the CFs, such as *query_info_intro*, further refinements are needed, however, our preliminary goal for this work is to explore the feasibility of the classification task with a coarse-grained taxonomy. Additionally, of the nine dimensions in the ISO-DA taxonomy (Bunt, 2006), we have focused here on classifying CFs in the task-related dimension. CFs that fall in other dimensions such as turn management, feedback, and social obligations are all treated as an *other* category but will be handled in future work.

query_info: Information-seeking function where the customer is asked to provide the unknown value of a specific entity, such as the request to provide the value of the check-out date in "*what date will you be checking out?*" or of the birthdate in "*please say or enter the 2-digt month, 2-digit day and 4-digit year of your birth.*"

query_info_open: Open-ended information-seeking function where the customer is asked to provide their intent, e.g., "*How may I help you?*" This includes requests for the intent related to a specific topic, such as "*what would you like to know about call blocking?*"

query_info_intro: In some dialog contexts, explicit requests for information are preceded by a statement that some information is needed, such as when a prior explicit elicitation for information was not successful for some reason, and an explanation is provided for the specific request E.g., "*I need your routing number in order to process your payment*". Because these responses are not explicit requests, we believe that their function is of the information-providing type rather than the information-seeking type.

query_confirm: Information-seeking function with an explicit request to confirm (or disconfirm) a proposition, such as "*I heard your credit card number as $NUM. Is this correct?*" or "*Just to be sure, I am about to cancel your annual subscription service. Is that correct?*" The expected response from the customer in such cases is a "yes" or "no".

query_disambig_yn: Information-seeking function to elicit a "yes" or "no" response from the customer, but unlike *query_confirm*, this does not elicit a confirmation. Utterances with this function are typically used to suggest an action to the customer to move the task in some direction in the dialog flow, for example, to invite the customer to transfer to a live agent for some task, as in "*Would you like to talk to someone about renewing?*", or to accept help via email, as in *Would you like me to send you an email to help you reset your password?*"

query_disambig_select: Information-seeking function to present choices for selection, such as "*Is this for a business, an educational institution or for a government entity?*" or "*Would you like to pay this with a debit card, credit card, or a different payment method?*" In the current version of the taxonomy, this does not distinguish between selection between entities and selection between actions.

promise: Commissive function committing to perform some action, such as "*I'll send a link to the email we have on file for you so you can reset your password.*"

offer: Commissive function also committing to perform some action, but unlike *promise*, the commitment here is contingent on some condition which may or not be specified. E.g., "*I can get you help with your login.*", "*I can get you to someone who will help with gift cards, but I just need bit more detail.*"

deflect_request: This is a special case of a commissive function that occurs frequently in our data, and involves deflecting a request from the customer while suggesting an alternative course of action. Typically, the deflected request is for a live agent, with examples such as "*I understand you want to speak to someone, but ...*"

instruct: Directive function specifying some action that the customer should undertake, such as "*Enter your username and password and click 'sign in'.*"

inform_issue: This is a special case of an information providing function to inform the customer of some contrary to expectation situation, such as when the customer's utterance in the prior turn was not understood or captured, e.g., "*I wasn't able to hear what you just said.*"

inform: This covers a broad class of information-providing utterances. Examples include "*Your confirmation number is $NUM.*", "*I see you have a tax appointment on $DATE at $TIME.*"

other: This category was used for utterances that could not be captured by any of the other CFs. As mentioned above, these include utterances with CFs from non task-related dimensions. We observed that most of these involve feedback CFs and social obligation CFs.

The CF taxonomy was developed first over a seed set of 200 utterances and validated over successive iterations as part of the active learning experiments described in Section 5. For example, the utterance "*I understand you want to speak to someone, but if you give me your credit card number, I can process your payment for you.*" has three functional segments with three CFs: *deflect_request*, *query_info_intro* and *offer*. In this stage of our work,

Field	Description
$uspan$	complete response as collected from our applications
cf	communicative function as annotated in Section 2.
$entities$	entities of interest such as bank account
$vertical$	domain names for ex. finance, tourism, hospitality etc.
$uspan\_vector$	dense vector representation for the utterance

Table 1: Description of indexing fields.

we ignore the ordering of the segments, and therefore, the classes as well.

3 Response Generation

For designing robust conversational systems, including ours, there exists a team of experts who list down alternative ways of how end customer might interact with the system and create responses for each such use case individually. The wording of the prompt has to be carefully chosen both to convey the desired message and to query for further information. This process is repeated from scratch for each new client and application, even though similar prompts may have been authored in the past for similar scenarios. For example, asking users for their personal information for authentication is a common task across many different applications. We observed many such situations where very similar system responses were present in different applications but were created from scratch because of no means available to access and re-use responses generated in the past.

In this section we describe the mechanism we devised for maintaining a repository of responses that can be used either for designing new conversation flows or for reuse directly as templates. As mentioned above each system response is characterized by both the communicative function and semantic content. We thought of providing search interface where designers could mention one or more of these dimensions to specify these requirements and access different realizations of the response they want to generate. One dimension is to provide a text span describing the theme of the current response such as validating gift cards, informing about longer wait times, calling about a new product launch, querying for name or date of birth. Communicative functions provides another dimension to search and filter responses by the desired intent. While we need an exact match to search for communicative functions, text based specifications should be able to retrieve responses that are semantically similar to the specified constraints.

Our requirements prompted us to use Elasticsearch because it facilitates both exact search as required for CFs as well as similarity based search in case designer describes a text phrase to specify key aspects of the content they wish to communicate. ElasticSearch (ES) is an open source search and analytics platform widely used for non-structured text data, hence it perfectly matches our requirements. Table 1 provides the list of fields indexed in ES for our task along with their definitions.

Each indexed field helps to filter responses according to the desired specifications, for example that the entity must be $billing\_address$ or $account\_number$. These filters help to obtain responses that are appropriate for a given context. The communicative function would be $query\_info$ when asking for $phone\_number$ or $first\_name$ of a person, however, *Please say and spell your first name* is more appropriate than *Please say and spell your phone number.* Since the current system implementation is context dependent, search fields such as $entities$ help to provide some context specific information thereby retrieving responses that are more relevant to the current context.

4 Models

In order to investigate response generation, we need to annotate a collection of system responses with the set of communicative functions defined above. The annotation task at hand essentially is a classification task, where given a system prompt we want to predict the relevant communicative function. For example, given system response " *Can you please tell me the phone number associated with this account*", output should be *query_info* and given input utterance as "*I can help with automatic payments, but first for security purposes please share the phone number linked to this account* " model should predict three labels as *offer, query_info_intro.*

Many machine learning classifiers are available in the literature for supervised multi-class classification problem such as SVM, KNN, and Gradient boosting etc, but being supervised algorithms they

cf	No of Prompts
Collections	1200
Communications	4000
Financial Services	9000
Food services	1909
Hospitality	5278
Insurance	6846
Retail Services	6266
Utilities	2160

Table 2: Distribution of

require annotated dataset for the models to learn patterns and make predictions whereas we had no reference dataset available with us. The only reference training dataset available with ISO annotation scheme is DialogBank (Bunt et al., 2016) which is very small to be used for training such classifiers and also more generic than ours.

We collected a set of approximately 37K system responses from twenty different applications across eight different domains providing enough response variations for commonly occurring modules. These responses are a subset of around 2 billion system responses being used by our conversation assistants on a daily basis for various clients across different domains. The distribution of different domains is present in Table 2.

BERT (Devlin et al., 2019) and T5 (Raffel et al., 2019) are two of the most widely adopted transfer learning approaches that are known to yield reasonably good performance even for a very small data set. For this study, we used Huggingface implementation of BertForSequenceClassification, where the final hidden state of the sequence is input to a fully connected softmax layer with cross-entropy loss function. There are two standard approaches to train multi-label classifier. The first being to train individual binary classifiers for each class in one-vs-rest approach and the second is to list down all possible combinations treating each one as an independent class. For example, if there are 3 unique classes, a, b and c, then we can have at most 7 distinct class labels (a), (b), (c), (a, b), (a, c), (b, c), (a, b, c) where each combination is treated as an atomic class. The latter approach though works well for smaller set of unique labels but becomes difficult as the number of classes increase and the distribution varies a lot.

Of the two approaches for solving multi-label classification problems, the preliminary set of experiments indicated that training ensemble of individual binary classifiers resulted in better performance than treating each combination as an atomic class. This implies that we trained and saved individual classifiers for each of the 13 communicative functions as binary classifiers. We experimented with different learning rates and found best results for 1e-5 with batch size 16 and max epochs 20. Also, to account for varied distribution of labels, we computed class weights for each class label as $size(label\_i)/max$, where max is over all sizes.

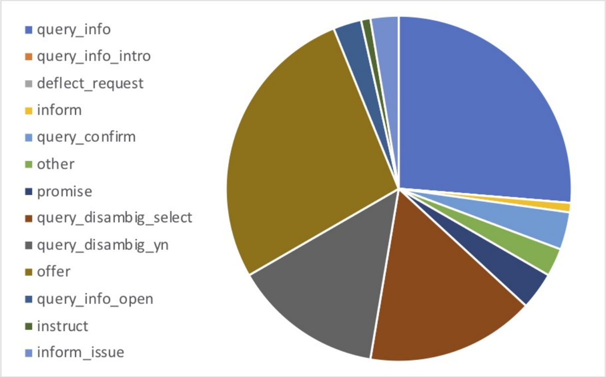


Figure 2: Different variations of asking the end-user customer about their name.

5 Experiments

We conducted experiments in two phases. The first phase is to train multi-label classifier for classifying system responses into space of communicative functions with a reasonable degree of accuracy. In the second phase, we analysed the quality of responses retrieved by the system for both simple queries given only CF or textual description and complex queries defined using a combination of other querying dimensions.

5.1 Multi-label Classification

We begin the training process by manually annotating a subset of about 200 responses. The training set had responses with the number of CFs varying from 1 to 3, with the ratios 0.60, 0.30 and 0.10 respectively. We then adopted an active learning approach to train and validate batches from the un-annotated corpus and adding them to the labelled data set. The distribution of communicative function over first 200 samples is presented in Fig. 2. The initial distribution clearly indicates that more than 50% of the prompts were information seeking prompts with labels ”*query_info*” and ”query_disambig”. Also note that the initial

cf	F1-score
$deflect\_request$	1.0
$query\_info$	0.875
$query\_confirm$	0.833
$query\_info\_open$	1.0
$query\_info\_intro$	0.875
$query\_disambig\_select$	1.0
$query\_disambig\_yn$	0.698
$inform$	0.95
$inform\_issue$	1.0
$instruct$	1.0
$promise$	0.91
$offer$	0.96
$report$	0.5
$other$	0.615

Table 3: F1-scores for communicative functions at the end of training process.

distribution had no training utterances for ”deflect_request”, ”query_info_intro” and some other relatively occuring combinations such as ”*offer, query_info*.

These classifiers were used to predict next batch of 200 prompts that were manually verified by the authors and language experts from the design team. With each iteration of training, predictions and evaluation, the classification accuracy improved finally leading to state-of-the-art performance score of 85% at the end of our training process. Though not directly comparable with joint intent and slot prediction models (Qin et al., 2020)(He et al., 2021) due to difference in the training objective, we observed that the accuracy scores are in comparable range of current literary works. We repeated our training-prediction-manual evaluation cycle four times increasing the test set from 100 to 500 samples. Each time, the predictions were manually verified with a team of designers discussing and agreeing to the final set of CF label. We adopted manual verification process because the communicative intent can not always be explicitly inferred from the wording of the response.

Table 3 reports the F1-scores at the end of round four with a training dataset of 1100 system responses. We found the F1-scores reaching to their maximum performance scores for most of our class labels and felt that the current classification model can be used to annotate our repository with reasonable accuracy. One of the reason for lower accuracy but higher F1-score was that certain percentage of responses were not predicted any CF label. Overall this percentage was close to 1%, where none of the classifiers were confident to assign the CF label. On inspection, we found that this was for those cases where either the system response has been incomplete or the wording of the prompt was such that the classifier could not predict any class label with a higher degree of confidence. Also, for the response labels $query\_confirm$ and $query\_disambig\_yn$, the responses were difficult to annotate clearly even by human experts. We are extending our training dataset with more such examples and hope to increase the accuracy level while keeping F1-scores at the maximum.

Once we could annotate the collection of system responses with a reasonable degree of accuracy, we created a repository that can be used by the design team to retrieve prompts by either specifying the communicative functions or by providing an abstract description of the current dialog state.

5.2 Response Generation using ES

It is an unrealistic assumption for the design team to learn a new technology (Elasticsearch) and a new language of communicative functions to specifying the desired information state. We therefore created a GUI based user interface for designers. to enter their requirements. The interface internally converts the search filters and their values into the query language executed by the Elasticsearch. We experimented with different search filters (query combinations) and found that given sufficient information, the system generated response variations consistent with the specifications mentioned by the designer.

Generating responses for a specific communicative function simply transitioned to executing a boolean query over the Elasticsearch. Table 4 provides subset of sample responses generated for the criteria ($cf = inform$). From the perspective of Elasticsearch, the results were 100% accurate but from the perspective of using these prompts for the current context we found them not directly usable. As no other semantic information was available about the dialog state, the responses retrieved by the system are coming form various domains and dialog state level.

From Table4 we can observe that there is one response informing customer about payments made, another response mentions mailing address while another is for street address and so on. In the ab-

response
Thank you, I have successfully submitted your payment.
I see that you have a repair issue that is scheduled to be resolved on DATE.
The mailing address we have on file for you is WORD.

Table 4: Variations for communicative function:$inform$

response
I see your street address is.
I have your street address as.
The street address I have on file for you is.

Table 5: A subset of 3 response variations generated for prompt:”street address” and communicative function:”inform”

sence of context specific information, the design choice is left to the designers to select which variation is more appropriate for the current context. As searching only by CFs would lead to data abundance problem, there are two different ways to specify context specific information; by selecting entities or by providing an abstract description of the content. One such example is presented in Table 5 where the designer filters the responses by including street address in search criteria. As we can see, almost all the responses are semantically similar to each other and can be adapted by the designers for the current conversation state.

As another example, Table 6 presents the scenario where the user provides a text based description and does not specify any communicative function explicitly. The system returns three different kinds of responses that look very similar but have different communicative intents behind each. The first response informs the customer about longer wait times and offers to help fulfill the desired task. The second response on the other hand provides the reason for longer wait times whereas the third response only informs the customer about the current situation. By providing three different variations, the system can reveal how these cases have been previously handled and provides an option to reuse any one of these realizations as per the current context.

Using a combination of both communicative function and text description provides the most appropriate means to specify the search requirements. We tested 25 different queries specifying both the text specification of the content and the context appropriate communicative function and observed the quality of system responses returned. We used Mean Reciprocal Rank to evaluate the set of responses generated given only text based specifications. We executed 30 different queries using a mixture of simple text based descriptions and complex queries with both components textual description and communicative functions. We found average MRR scores of 0.6, 0.71 and 0.72 for Top-1, Top-3 and Top-5 respectively with Universal Sentence Embedding (USE) for computing semantic similarity. The MRR scores for ELMO and SBERT were much lower for our datasets.

6 Literature Review

Accurately predicting the speakers communicative intent is extremely important for a successful communication and thus intent detection has always been widely pursued research thread. As virtual assistants are becoming a part of daily life, it has been acknowledged that most a times speaker is communicating multiple aspects with in a single utterance. There is an increasing trend towards training joint models for intent detection as Multi-Label Classification (MLC) and entity detection(also called slot filling (Hou et al., 2021) (Qin et al., 2020) (He et al., 2021). These systems compute relevance score for each label and utterance combination and then select the labels with maximum similarity score. Some of these approaches are few shot learning approaches proposing techniques to perform MLC with fewer examples, but they all pretrain on domain specific data and then extend this to out-domain dataset. In contrast, our work aims to annotate with domain-independent dialog act labels and only focuses on predicting communicative functions, hence we adopted conventional machine learning approaches for classifying communicative functions.

The concept of representing dialog acts using domain independent general purpose schemas has been studied multiple times as Dialog Act Markup in Several Layers (DAMSL) by Allen et. al (Allen and Core, 1997b) and as ISO standard by Bunt et.al. (Bunt et al., 2012). The ISO taxonomy pro-

cf	response
$offer$	Due to heavy call volume at this time it could take over 90 minutes to talk to a representative, lets see if I can help you.
$inform\_issue$	We are sorry for your inconvenience, however, we are experiencing extremely high call volume due to the recall and this has caused extremely long wait times to connect with an agent
$inform$	Wait a moment while this call is being transferred to our system.Wait times are longer due to heavy call volumes.

Table 6: Variations of the system response informing customer that there are excessive wait times.

vided generic representations of a speakers intent by defining 9 core dimensions and around 60 different communicative functions using domain independent and task independent labels.

7 Conclusion

In this paper, we proposed a taxonomy of communicative functions that effectively captures the communicative intent of a dialog turn using domain independent labels providing means for flexible and generic dialog modelling. The taxonomy was used to annotate a subset of user responses from human-machine conversations used by our real-life applications on day-to-day basis. We experimented with this annotated dataset to generate different linguistic variations of the system responses for given communicative function and desired keywords indicating the essence of the current dialog turn. Our experiments indicated that the proposed taxonomy can successfully learn representations that capture what the dialog is written to accomplish across different applications and verticals. We experimented with these annotations in a dialog generation settings and found that we are able to generate system responses given desired specifications from the existing data itself.

References

Jan Alexandersson, Bianka Buschbeck-Wolf, Tsutomu Fujinami, Elisabeth Maier, Norbert Reithinger, Birte Schmitz, and Melanie Siegel. 1997. Dialogue acts in VERBMOBIL-2. Technical Report 226.

James Allen and Mark Core. 1997a. DAMSL: Dialog act markup in several layers (Draft 2.1). Technical report, University of Rochester, Rochester, N.Y.

James Allen and Mark Core. 1997b. Draft of DAMSL: Dialog act markup in several layers. Unpublished manuscript.

James F Allen, Lenhart K Schubert, George Ferguson, Peter Heeman, Chung Hee Hwang, Tsuneaki Kato, Marc Light, Nathaniel G Martin, Bradford W Miller, Massimo Poesio, and David Traum. 1994. The TRAINS project: A case study in building a conversational planning agent. Technical Report 532, Computer Science Department, University of Rochester, Rochester, N.Y.

Anne H Anderson, Miles Bader, Ellen Gurman Bard, Elizabeth Boyle, Gwyneth Doherty, Simon Garrod, Stephen Isard, Jacqueline Kowtko, Jan McAllister, Jim Miller, Catherine Sotillo, Henry Thompson, and Regina Weinert. 1991. The HCRC map task corpus. *Language and speech*, 34(4):351–366.

John L. Austin. 1962. *How to do things with words*. Cambridge: Oxford University Press.

Harry Bunt. 2000. Dialogue pragmatics and context specification. *Abduction, Belief and Context in Dialogue. Studies in Computational Pragmatics. Amsterdam: Benjamins*, pages 81–150.

Harry Bunt. 2006. Dimensions in dialogue act annotation. In *LREC*, pages 919–924.

Harry Bunt, Jan Alexandersson, Jae-Woong Choe, Alex Chengyu Fang, Koiti Hasida, Volha Petukhova, Andrei Popescu-Belis, and David Traum. 2012. ISO 24617-2: A semantically-based standard for dialogue annotation. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 430–437.

Harry Bunt, Volha Petukhova, Andrei Malchanau, Kars Wijnhoven, and Alex Chengyu Fang. 2016. The dialogbank. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*. European Language Resources Association (ELRA).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ting He, Xiaohong Xu, Yating Wu, Huazhen Wang, and Jian Chen. 2021. Multitask learning with knowledge base for joint intent detection and slot filling. *Applied Sciences*, 11(11).

Yutai Hou, Yongkui Lai, Yushan Wu, Wanxiang Che, and Ting Liu. 2021. Few-shot learning for multi-label intent detection. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13036–13044. AAAI Press.

Libo Qin, Xiao Xu, Wanxiang Che, and Ting Liu. 2020. TD-GIN: token-level dynamic graph-interactive network for joint multiple intent detection and slot filling. *CoRR*, abs/2004.10087.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

John R. Searle. 1969. *Speech acts: An essay in the philosophy of language*, volume 626. Cambridge university press.

Amanda Stent, Matthew Marge, and Mohit Singhai. 2005. Evaluating evaluation methods for generation in the presence of variation. In *Proceedings of CICLing*.

David R Traum and Staffan Larsson. 2003. The information state approach to dialogue management. In *Current and new directions in discourse and dialogue*, pages 325–353. Springer.