Paper	Notes
Paper 8	1. Table 1 has an example of a conversation along with a description. 2. Evaluated on one conversation dataset, six intent datasets and one short text clustering dataset. Datasets and code are released by authors. 3. Table 2 has the datasets which were used for evaluation. It shows the number of utterances, number of intents and max and min number of utterances for any intent. 4. The different types of datasets used are 1) Conversation datasets and 2) Intent dataset and 3) Short text dataset (consisting of short text snippets like a tweet) 5. Evaluation using precision, recall, F1 score and unsupervised clustering accuracy (ACC) https://deepnotes.io/deep-clustering. 6. Evaluation of short text dataset done using 1) Normalised mutual information and 2) Adjusted Rand Index 7. Results presented in tables 3 and 4.
Paper 10	1. ER features used for reranking are 1) ER success 2) No match 3) Lexical similarity of slots. 2. Annotated internal dataset is used with 1 million annotated utterances. Domain, intent and slots are annotated. 3. Synthetic data is generated for domain specific evaluation. The different types of generated data are 1) "play X" for the music and video domains. X can be artist name, song name, album name, video name, etc. X is sampled from the ER catalogs of artist, songs, albums, videos, etc 2) "give me the direction to X" for local search domain. X can be place name, destination name, etc. 4. Semantic Error Rate (SemER) is used as the evaluation metric. It is E/T where E is the total number of substitution, insertion and deletion errors of the slots and T is the total number of slots. 5. Baseline reranking model used for comparison is a linear function that predicts ranking score based on DC, IC and NER confidence scores. 6. Results are in Table 1. 7. The 2nd, 3rd and 4th columns are the same for in-domain reranking as they add score matching which is only useful for cross domain reranking. 8. Examples of utterances where the proposed architecture performs better are given on the right of Table 1.

Paper	Notes
Paper 11	1. Figure 1 has an example of a baseline model, a model which introduces new knowledge and a model which introduces knowledge smoothly with dialog acts (here it is feedback then a statement then a question and finally a statement). 2. Figure 2 shows the architecture. Knowledge selection occurs in which a new knowledge statement is selected from the knowledge corpus based on the user utterance. The the dialog act planner uses the user utterance and the selected knowledge statements to produce an action plan. It consists of the sentences in the response represented by (d, k, t, h) tuples. 1) Dialog act (d) - the type of sentence it is (or what purpose it serves in the response). 2) Knowledge (k) - the knowledge sentence selected. 3) Topic (t) - The domain of the sentence. 4) Use - knowledge flag (h) - Flag used to signal whether to use k. 3. The tuples are fed into a response generator which generates the natural language output. 4. Table 1 shows the possible dialog acts. 5. Publicly available Topical Chat dataset is used for experiments. It is a large open-domain dataset. 6. Simple models are used to automatically annotate the Topical Chat dataset for attributes such as dialog acts, fine grained association between knowledge sentences and dialog turns, etc. 7. Table 3 has the results of evaluation. The metrics used are perplexity, BLEU-1, ROUGE-L, unigram F1-score, n-gram diversity and human evaluation. For human evaluation they generated 200 snippets each with 5 dialog turns. Then the two models were used to generate responses and 3 crowd workers were asked which response was more appropriate. BLEU-1 means 1-grams were considered. Perplexity is 2 to the power cross entropy loss. ROUGE-L measures the length of the Longest Common Subsequence between the predicted sequence and the ground truth sequence. 8. F1, BLEU, ROUGE scores for the proposed model are lower than the baseline as the model decodes shorter sentences resulting in lower recall. 9. Table 4 shows how well each model followed the action plan. Figure 5 gives an example of the model which best adheres to action plans. 10. There are two types of dialog act planning 1) Knowledge dependent and 2) Knowledge independent. There are 4 ways to implement knowledge independent dialog act planning but the evaluation only uses 2, simple and Seq2Seq. Simple has a set of predefined transitions which determine the next dialog act given the previous. Seq2Seq uses a BiLSTM to estimate dialog act given the dialog context. The automated metrics are given in Table 5. Human evaluation was also done in the same manner as before. This is given in Table 6.
Paper 12	1. Figure 2 shows an example of the hierarchical frame based representation. 2. Figure 3 shows the actual methods of representation. 3. Figure 4 shows the actual representation learnt by the model. NT represents Non Terminal which leads the model to predict the token by decoding the NT hidden vector. 4. The evaluation is done on the ATIS public dataset. It has a shallow hierarchy. ATIS has audio recordings of people requesting flight reservations. There are 20 intents, 120 slots and 5871 utterances. A simulated dataset is also used for evaluation. It has 2 intents, 7 slots and 6810 utterances. Both modified datasets are available. 5. There are two types of forms generated by the model which are shown in Figure 5. The model is evaluated for both. The metric used is micro-f1 scores. Micro-f1 scores are used for multi-label problems. It puts more emphasis on the common labels. 6. Table 2 summarises the results. 7. Baseline model used for comparison extends flat frame structures by predicting a group number along with slots. An example of this is given in Figure 6 and the comparison results are in Table 3.

Paper	Notes
Paper 13	1. Figure 1 gives the architecture. 2. 2 experiments are performed. 1) Attention module is tested with different methods of computing attention weights. 2) SBL is compared against upsampling. 3. The evaluation metric used is Replication Accuracy (RA) which is a measure of how well the model replicates skill routing behaviour. 4. The user queries were split into 21 slices. 1 base slice and 20 tail intent slices. 5. First experiment 6. The baseline model used for comparison is based on attention based BiLSTMs with fully connected layers on top of it. The comparison results between baseline and slice-based architectures are given in Table 2. 7. Second experiment 8. SBL improves upon upsampling and the results are given in Table 3. Table 4 gives detailed results for each tail intent. 9. Discussion paragraph says "Although the overall performance gain of slice-aware approach compared to the upsampling was marginal, it is worthwhile to note that the slice-aware approach was able to lift up the replication accuracy for more number of tail intents while minimising unexpected performance degradation that was more noticeable in the upsampling approach. This result implies that the slice-aware approach has more potential in stably and evenly supporting tail intents."
Paper 14	1. Figure 1 shows the rejection modules within a system. 2. Figure 3 has the architecture for a hypothesis rejection module. 3. 2 experiments are conducted 1) Comparison of the two types of hypothesis rejection modules on human generated transcripts. 2) Performance of rejection modules in a real life scenario is evaluated. 4. The metrics used for evaluation are False Acceptance Rate (FAR) and False Rejection Rate (FRR). FAR/FRR are the number of false accepts/rejects divided by the test set size. 5. First experiment 6. Table 1 has the results. The authors conclude that there is no significant difference between the R1 and R2 schemes in terms of the results of the evaluation. 7. Second experiment 8. The rejection hypothesis are trained by also including ASR features. The test set is only 60% of the original test set. 9. Table 3 gives the results. Factoring in ASR features improves the rejection module. Table 2 gives 4 types of ASR errors and the corresponding action taken by the hypothesis rejection module.

Paper	Notes
Paper 15	1. Figure 2 shows the parse of a complex query. 2. To the right of Figure 2 are the 3 types of queries. 3. Figure 3 gives the architecture. 4. Approach is tested on 5 different dataset, 3 public and 2 internal. Public datasets are 1) Facebook TOP - Contains complex hierarchical and nested queries. Has 45k annotations with 25 intents and 36 slots. 2) SNIPS - Contains simple queries. Has 7 intents with about 2000 training examples and 100 test examples for each intent. 3) ATIS. Internal datasets consisted of millions of examples from datasets used to train and test Alexa. 5. Baseline model used for internal datasets performed joint intent and slot tagging using BiLSTMs and Conditional Random Fields (CRF). CRF is a statistical learning method which models the input as a graph and takes into account an object's neighbours when predicting its class. The embedding and encoder layers are a language model pre-trained on Alexa data. 6. Baseline model used for ATIS and SNIPS are the top performing models from another paper. 7. Baseline model for Facebook TOP is a Shift-Reduce Parser. 8. The evaluation metric used is Exact Match Accuracy. EM accuracy is total number of examples with exact label match/total number of examples. An example has multiple samples. 9. Results are given in Tables 1 to 5. Section 6.1 gives an interesting example from Facebook TOP in which the proposed model generates a better parse than the annotation.
Paper 16	1. Figure 1 shows the three training steps proposed by the authors. 1) Transformer is pre-trained by masking words in the input sentences. 2) The pre-trained model is transferred to a general model. QA input pairs are given and the model has to decide whether the given answer answers the given question. 3) The transferred model is now adapted to the target domain in the same as 2). 2. A new dataset is developed by the authors by using the NQ dataset. Table 1 contains the types of samples in NQ and what type they are in ASNQ. 3. 3 AS2 datasets are used for evaluating the model. 1) WikiQA 2) TREC-QA 4. The evaluation metrics used are Mean Average Precision (MAP) and Mean Reciprocal Recall (MRR) on the test set. 5. Results 1) WikiQA - Table 3, 2) TREC-QA - Table 4 6. Figure 4 shows the stability of TANDA as number of training epochs are varied. Table 5 shows the robustness of TANDA to noise. 7. The effectiveness of the ASNQ dataset is also given. 8. The TANDA technique was also tested on Alexa data to show that the results generalise well.

Paper	Notes
Paper 17	1. Figure 1 shows a multi-domain dialog with dialog state shown at each step. 2. Figure 2 shows the model architecture. The decoder generates 2 probability distributions, One over the vocabulary and the other over all the words in the conversation history. This allows the decoder to generate words which are not defined in the vocabulary. This removes the requirement that all the possible slot values must be predefined in an ontology. This also helps the model in zero shot cases. 3. Approach proposed is evaluated on MultiWOZ 2.1 which is a multi-domain Wizard-of-Oz dataset. It has 10k dialogs each with an average of 13.67 turns. There are 7 domains and 37 slots. Only 5 out of 7 domains are used in the experiments. 4. The metrics used are 1) Average slot accuracy, which, for a turn, is the fraction of slot for which the model predicts the correct value. 2) Joint goal accuracy, which is the fraction of turns for which all slot values were predicted correctly. 5. Baseline model used for comparison is the TRADE model. Table 2 gives comparison of TRADE and proposed model on single domain data. Table 3 gives comparison of TRADE and proposed model on multi-domain data. It also includes several ablation variants of the model. Table 4 contains comparison of TRADE and the proposed architecture in a zero shot setting. 6. Shortcomings - The model makes the most errors with open ended slots such as restaurant-name. The Average slot accuracy and Joint goal accuracy both decline as the number of turns in a conversation increase. The Joint goal accuracy decreases to a much greater extent as shown in Figure 4. When some randomly selected dialogs were manually examined, it was seen that the model often correctly predicts the slot value but for the wrong slot (this happened in 36% of the errors). The model may also differ from the ground truth value by a single character or word, but this may completely change its meaning.
Paper 9	1. Figure 2 shows CoBot, which is the conversational bot toolkit provided to the teams. It can perform natural language understanding and dialog management. 2. Table 1 shows the improvement in topic and intent classification accuracy of the new system compared to last years. 3. Table 2 shows topic and intent classification accuracy for HRNNs compared with BERT and WLM (a BERT model trained on warped sentences from Wikipedia, there is more noise in its training data so it is supposed to be more robust than BERT).
Paper 7	1. Figure 1 shows hows modulation of voice, body movement and facial expressions can simulate 2 different personalities. The figure shows the agent interacting with a passport officer who asks the agent where he is from and the asks for his passport. The left side shows an extraverted agent and the right side shows an introverted agent. 2. Figure 2 shows the Scenario Handler, which is implemented as a state machine. It determines the agent's response to a user utterance. The paper uses the Watson API to convert Speech to Text, find intent, select an appropriate response and convert Text to Speech. The base animation for the CA is determined by the Scenario Handler based on the intents and entities. However the personality specific modifications to the base are made by the Animation Modifier. The Watson Text to Speech API is also used to tweak vocal features based on the OCEAN parameters. 3. Figure 3 shows the Animation Modifier. It modifies the base animation based on the agent personality. It maps the OCEAN values into 3 group of animation modification parameters 1) Laban Shape Quality (LSQ) 2) Laban Effort (LE) 3) Facial Expression. LSQ parameters determine the inverse kinematic targets' positions and weights (the way and speed in which joints should move to reach the target position). LE parameters are used to refine the animation. Facial Expression parameters regulate interpolation of face shape keys. Lipsync (bu Oculus) is used to extract visemes (the movements made by the mouth for different sounds) from speech and blend them into face shape keys to animate the mouth. 4. Table 1 has dialogs which show the application of different OCEAN parameters.

Paper	Notes
Conversational Agents: Theory and Applications	1. Types of conversational agents 1) Chatbot - Intended for casual conversation with a user 2) Task-oriented agents - Engage in discussion with a user to reach one or several specific goals often (but not always) in a specific domain. 2. Types of conversational agents based on dialog representation 1) Interpretable 2) Black Box (involving deep neural networks) 3. Types of chatbots 1) Pattern-based 2) Information-retrieval 3) Generative. The first two are interpretable and the third may be an interpretable or black box system. 4. Pattern-based - Usually based on AIML (Artificial Intelligence Markup Language). AIML is an XML-like language which is used to define template matching rules for chatbots. Each user input (pattern) is associated with an output (template). An example AIML specification is given on page 9. AIML also has the ability to make redirections from one pattern to another. So various patterns can be mapped to the same template. An example is given on page 10. 5. Information-retrieval - Match the user input to the most similar sentence in a dialog corpus (dataset of conversations) and return the corresponding response. Similarity between sentences is judged with sentence embeddings. A common embedding is TF-IDF (term frequency - inverse document frequency). TF vector is a vector of same length as vocabulary with the number of occurrences of each word in the sentence normalised by the number of words in the sentence. TF gives too much emphasis to common words like "we", "the", "a", etc. The IDF for a word "w" is defined in equation 2 on page 10. The IDF for less frequent words will be more than the IDF of more frequent words. Document and sentence mean the same. We place the IDF of each word in a vector of the same length of the vocabulary. Now given a sentence "u" we can find "q(u)" which is the elements wise product of the sentence vector with the IDF vector. The sentence can then be compared with all other sentences using the normalized dot product (cosine similarity). The most similar sentence is found in this way. TF-IDF has the disadvantages that it does not consider order of words, synonyms and context. 6. The disadvantages of TF-IDF are solved by word embeddings. DNNs may be used to learn word embeddings. 7. Generative - They generate their responses by modelling a probability distribution over language (language models). This is currently done by training DNNs on large amounts of data. Seq2Seq models are used in generative chatbots. Transformers are another model used which use self-attention to consider context from other positions of the sentence. 8. Task-Oriented Agents - The classification into interpretable and black box is especially important for task-oriented agents as they are usually used for reaching concrete goals and may be deployed in high-stakes settings. 9. Figure 4 on page 15 shows the pipeline model for task-oriented agents. When using black box systems, the lines between the modules in the pipeline model are blurred. This is because there is a push towards requiring less and less human specification of how the processing should take place. 10. Interpretable task-oriented agents - 1) Finite state systems - The conversation follows a rigid and predefined structure and sequence. Dialog initiative is with the CA and the user can only answer the questions asked. 2) Frame-Based/Slot-Filling systems - There is mixed initiative. The system attempts to fill the slots in a frame through its conversation. The domain (for multi-domain systems) and intent also must be determined. Intent detection is implemented in the form of explicit, hand written rules or with semantic grammars, which define a set of rules which form a compact representation of many different variations of a sentence. This representation is usually insufficient and we require determination of dialog acts and dialog state tracking (ability to keep track of context). The methods used can maintain only a single, deterministic description of the dialog state. 11. Methods to model dialog state in a probabilistic and statistical way - 1) Markov Decision Process (MDP) - There are a set of states (S), a set of actions (A) and a matrix of transition probabilities. The matrix has the probability of moving from a particular state to another when performing a specific action. Any action and corresponding resulting state have an immediate reward associated with them. The dialog policy decides what action the agent takes in a given situation. The dialog policy is optimized using reinforcement learning to maximize reward.

Paper	Notes
Conversational Agents: Theory and Applications - Continued	1. 2) Partially Observable MDP (POMDP) - They take into account the inherent uncertainty of being in a state. It defines a belief state which defines a probability distribution over all possible dialog states. 2. Black box task oriented agents - They are similar to black box chatbots. They also use word embeddings and rely on recurrent, convolutional or transformer-based architectures. Memory networks or graph CNNs are also used. Black box systems can also be framed using the pipeline model with a different network for each module. But they are still connected and are dependent on each other so they are not completely independent. There are a methods to improve the interface between DNNs and the knowledge base. Normally the DNN for dialog policy choses the most appropriate information retrieval query. 1) Soft queries with multiple degrees of truth can be used to obtain richer results. 2) Querying of large databases can be made faster by assimilating the knowledge base into network parameters during training. 3. Evaluation of CAs - Low level language processing metrics - precision, recall, GLUE (general language understanding evaluation)/SuperGLUE (they consist of different natural language understanding tasks and corresponding benchmarking datasets), BLEU, ROUGE 4. Interaction quality evaluation - 1) Conversational interaction - Human evaluation can be categorized into sensibleness (agent's answers make sense in the given context) and specificity (ability to provide specific and informative answers). Measuring these 2 parameters correlates well with measuring perplexity (perplexity is the exponentiated negative log likelihood of a sequence. Likelihood is the probability of the model predicting the correct i^{th} token given the previous tokens). 5. Other metrics like conversational user experience (overall user rating), coherence (number of sensible response over total number of responses), engagement (number of utterances), topical breadth (number of topics touched), topical depth (number of consecutive utterances of a topic) were used in the Alexa prize.
Paper 19	1. Figure 1 shows the correlation between SSA and perplexity as well as the performance of various chatbots. 2. Evolved transformer is an architecture obtained by applying evolutionary neural architecture search to a transformer model in order to find a better version of the transformer. Evolutionary Neural Architecture Search defines a search space (with the constraints such as maximum number of parameters, which blocks to use, maximum number of each block) and find the best model by finding better models and getting rid of the worst models. 3. Figure 2 shows the correlation between SSA and human likeness. 4. Tables 2 and 3 show the possible output produced by Meena for a prompt. Table 2 picks the best 10 by sampling with temperature. Temperature sampling is done by dividing the log likelihood by number of tokens in the response. Beam search is done by considering N possible sequences with best probability at every token prediction step. Temperature sampling and ranking produces better responses than beam search in terms of SSA. 5. Section 3.5 gives some sample conversations. 6. Figure 5 and Table 4 show the comparison of Meena with other chatbots. 7. Tables 5 and 6 show cross turn repetitions (repetitions of sentences in two turns).

Paper	Notes
Paper 20	1. Table 1 shows the logical representation given to annotators and the corresponding natural language templates produced. 2. Table 2 gives an example conversation. To the left of table 2 are the possible intents in the banking domain. 3. User and system bots are finite state machines. 4. Table 3 shows an example of turn compression, Table 4 shows new API, Table 5 shows reordering, Table 6 shows Another Slot and Table 7 shows Audit More. 5. The papers uses 2 types of end-to-end Memory Networks to test the data collection technique. Memory networks have a component used to store data. They use current inputs and stored data to make inferences. Memory networks have been shown to be able to do non-trivial tasks such as issuing API calls to a knowledge base. 2 types of memory networks are used for evaluation. 1) Single End-to-End Memory Network - It is trained with dialogs from all intents. 2) Multiple End-to-End Memory Networks - It has 6 memory networks, each trained for 1 intent and a seventh memory network to select one of the six. Both networks are implemented to see if sharing information improves or hinders the performance of memory networks. 6. For evaluation, 200 dialogs per intent per test case are generated. 1/3 is used for training and 1/3 is used for validation (development set). In-template test cases are randomly sampled from training and validation sets. Out of Template test cases use the remaining 1/3 of the dataset. The model is evaluated in ranking candidate answers/API calls instead of generation. 7. The metric used for evaluation is per-response accuracy, which is the number of correct system responses predicted in all the dialog turns of the test set. Table 8 shows the per-response accuracy of single and multi memory networks on In Template and Out of Template test cases. 8. OOP has a large impact on the performance of both models. In general MMNs outperform SMNs. This may be due to the fact that MMNs have specialized parts for each intent which can perform better than SMNs which combine all intents. In Template or Out of Template does not make much of a difference. This may be because Out of Vocabulary test case are not included so the model may learn to directly recognise entities and their corresponding slots instead of learning to recognize them with linguistic patterns. Multiple Out of Pattern test cases in the same dialog significantly decrease the performance of both SMNs and MNMs. This is shown in Tables 9 and 10. The model predicts responses based on the past context and current user utterance. When we give Multiple Out of Pattern turns, the context will also contain a previously unseen dialog template which makes it significantly harder for the model. 9. Turn compression is found to be the easiest for networks and new APIs are seen to be the most difficult. This is mostly because new APIs are usually called at the beginning of the conversation and there is very little context whereas turn compression is during slot reconfirmation which comes later at which point there is a lot of context.