Multi-turn QA: A RNN Contextual Approach to Intent Classification for Goal-oriented Systems

Martino Mensio
Politecnico di Torino
Torino, Italy
martino.mensio@studenti.polito.it

Giuseppe Rizzo
Istituto Superiore Mario Boella
Torino, Italy
giuseppe.rizzo@ismb.it

Maurizio Morisio
Politecnico di Torino
Torino, Italy
maurizio.morisio@polito.it

ABSTRACT

QA systems offer a human friendly interface to navigate through knowledge, which can range from encyclopedic to domain-specific. Generally, a QA system is designed to provide an answer to a specific question once (so-called single turn) and state-of-the-art systems reach nowadays robust performance in such a scenario. However, most of the interactions with QA systems are based on multiple handshakes of question/answer pairs, where the human being refines the questions further, while the system can collect the necessary information and generate a compelling final answer through multiple turns. In this paper, we investigate and experiment a multi-turn QA system that is suited to work given a particular domain of knowledge and configurable goals. Our approach models the entire dialogue as a sequence of turns, i.e. questions and answers, using a Recurrent Neural Network which is firstly trained to understand natural language, classifying entities and intents using prior knowledge of domain-specific interactions, and provide answers according to the domain used as background knowledge. We have compared our approach with state-of-the-art sequence-based intent classification using a well-known and standardized gold standard observing an increase of 17.16% of F1. Results show the robustness of the approach and the competitive results motivate the adoption in multi-turn QA scenarios.

CCS CONCEPTS

• **Human-centered computing** → **Natural language interfaces**;
• **Computing methodologies** → *Neural networks*;

KEYWORDS

Multi-turn question answering; Conversational agent; Goal-oriented conversational agent; Recurrent Neural Networks

ACM Reference Format:
Martino Mensio, Giuseppe Rizzo, and Maurizio Morisio. 2018. Multi-turn QA: A RNN Contextual Approach to Intent Classification for Goal-oriented Systems. In *The 2018 Web Conference Companion, April 23–27, 2018, Lyon, France.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3184558.3191539

WWW'18 Companion, April 23–27, 2018, Lyon, France

ACM ISBN 978-1-4503-5640-4/18/04.
https://doi.org/10.1145/3184558.3191539

1 INTRODUCTION

Question Answering (QA) systems [26] are defined as software components that provide natural language interface towards information that is usually stored in structured tables [13], graphs of data [6, 33], or in collections of documents. They are usually defined as search tools that first understand natural language and provide back a response in a single turn (i.e. one question and one answer). No context is usually carried over eventual next interrogations. The system has to understand them separately.

With the continuous transition towards *voice-first* technologies, we have observed a large adoption of goal-oriented agents. They are a particular case of domain-specific QA systems [13] that are not just limited to retrieve information but also to complete some actions [7], and thus they provide a conversational interface to complete transactions. A transaction is defined as a set of interactions towards a specific goal, e.g. buying tickets, where the user has its own goal (called intent) and talks to the system in order to achieve it. The transaction can have different turns because the system may require some information from the user and the user can refine his initial goal. To keep track of all the possible paths of a dialogue, usually a Finite State Machine [2] is used. On those systems it is usually difficult for a user to interrupt an intent and to take back the dialog initiative without using special commands (like "stop" or "restart"). In addition, QA systems lack in exploiting the surrounding context when processing the single question, while goal-oriented agents offer interfaces that answer generally simple questions over a predetermined number of things. Having this landscape, this study proposes a system able to manage the interaction context in a multi-turn fashion. The presented work has been researched and applied for a goal-oriented agent, and it can be applied to multi-turn QA systems in a similar way by considering the coarse-grained sentence classification.

Our approach aims to provide interaction context, i.e. a surrounding environment for the present sentence, by taking into account previous user's intents and previous agent's sentences in order to have a better understanding of the current sentence sent by the user. Being able to dynamically contextualize the sentences, by understanding when to keep the context and when to discard it following some signals in both user and agent turns, can be an important feature in agents that are interrogated in highly asynchronous scenarios (in particular in mobile devices when users are using numerous widgets in parallel). In those situations, using a time-based session split can be misleading and a working solution should rely on the contents only. Some studies have been already done on this problem of contextual understanding [5, 9, 10, 25, 37] with different variations at different levels of complexity: from simply feeding back some features of the previous turn, up to memory

networks [28] that may require a lot of training samples. Inspired by these, we want to understand how relevant is the interaction context in the process of understanding the sentences.

The remainder of this paper is structured as follows: in Section 2 we set the scene of our approach and experimentation while listing the peculiarities of goal-oriented QA agents and the differences with respect to domain-specific QA systems. Section 3 presents our approach and in Section 4 we first describe the experimental setup and then discuss the performance of the approach. Section 5 concludes the paper and outlines future research activities.

2 BACKGROUND

The origins of conversational agents are rooted in chit-chat agents. These systems were born to overcome the challenge initially proposed by the Turing Test [32] and aimed to emulate a natural language conversation between humans. The approach used in [35] implements a set of handcrafted rules: patterns executed on the input sentences are used to find a match over a rule-set, and then the responses are generated from a set of templates. Also, the memory of the agent (required to answer to next questions in a coherent manner) is managed by those rules, setting state variables and retrieving them when necessary. These agents are able to answer back only to the questions they are designed for and, furthermore, the interaction does not carry meaningful contents from the informational point of view because the conversation goal is only to make the interlocutor thinking to be speaking with something that understands the conversation. This section studies and elaborates how there has been a split of conversational systems, one towards rich interrogation and others towards more structured dialogues, emphasizing the differences and common points in the understanding part. We conclude outlining the point of contact between the two and how they can be merged to provide a more intelligent interaction.

2.1 Beyond rule-based QA systems

On the side of QA systems, there has been a lot of research towards the goal of building systems that can answer questions by providing rich answers with an easy to use conversational interface. For this type of systems the information is usually stored in Knowledge Bases (KBs), which can be in the form of structured tables or graphs or textual documents written in natural language. Two major paradigms of QA systems have arisen: knowledge-based and information retrieval-based for the unstructured one. For both the goal is to provide answers by performing three steps: *i)* understanding the question (extracting all the useful parameters), *ii)* retrieving the information (against the KB via queries or some more unstructured forms), and *iii)* combining the results and presenting them in textual form.

The approaches that use structured knowledge range from handcrafted or dynamically built templates [1], or using an intermediate logical form [4, 14, 30], or using some measures of entity proximity and relations with respect to a question in an appositely generated embeddings space [22]. A key role in those approaches is played by natural language processing methods such as named entity recognition and dependency parsing, that enable a fine-grained classification of the sentences. For unstructured information retrieval, where the information is not in the form of structured tables or graphs, but simply a collection of documents in natural language, instead of building the queries, the search is done in unstructured form, usually with passage retrieval [34]. This is the kind of interrogation that normally happens with search engines, where the documents are sorted by their relatedness to the question (presence of keywords or synonyms). A second stage is applied to process the search candidates in order to build an answer. This requires a deeper understanding both of the question (what is the desired response) and of the retrieved fragments [31]. Those systems, however, have a great limitation on the sentence contextualization: each question is answered independently from the others. This makes the interaction very rigid, as the user cannot refer easily to previously mentioned entities, nor refine a question without the need to reformulate again the whole set of constraints. It is a problem affecting QA systems that obtain the information in both ways. Even in bAbI[1], when multiple questions are asked to the system, each one of them is independent. The memory and context of the system only works for the sentences that contain facts. No follow-up questions are possible.

To the best of our knowledge, the only studies that have been made towards a contextualized understanding were presented in the TREC[2] "contextual suggestion" track. This track, over the years, used two types of context: discourse and user. The first, introduced in TREC10,[3] has been used with the goal to perform reference resolution [16, 29]: some indicators in the sentence (such as pronouns, definite nominals or ellipsis) are used to find in previous turns the ones that contain the referenced entity and build a model that selectively retains query terms, following the Centering Theory [15]. The user context instead turns the information retrieval into a recommendation problem, where the items are places (presented under the form of natural language text) that should be suggested to the user having as features his location and his preferences.

2.2 Goal-oriented agents

Goal-oriented agents come into the landscape of Question Answering as domain-specific systems that enable users to interact with some services to perform different tasks, instead of being a natural language interface to perform advanced queries. Being a QA branch, it lets users obtain domain-specific information. But it can also be used to perform some actions, under the form of digital personal assistant, in different specific domains: booking and travel services are just examples.

The word "goal" characterizes dialogues where each party is aware of the objective of their communication, from one side to use some services and receive some information, and on the other to provide them. The dialogue is not an end in itself like in chit-chat systems, as explained in [19]. These systems are similar to structured and unstructured QA in the language understanding part: categorize the sentences and extract some parameters from it. But there are some key differences:

main focus: it is very important to provide access to the available operations and manage interactive guided procedures,

[1] https://research.fb.com/downloads/babi
[2] http://trec.nist.gov/
[3] http://trec.nist.gov/pubs/trec10/t10_proceedings.html

that require handling the conversation state. The understanding of complex interrogations is not focal;

limited search capabilities: unlike QA systems, the access to a knowledge base may be limited by a specific set of available operations, caused by limited remote APIs or by a pre-existing application logic. Those can correspond to a finite set of question types and a finite set of parameters;

bidirectional QA: the system may require some missing parameters to the user making the interaction more complex;

interaction with dynamic data: the information stored can change frequently over time due to resource availability (for example when providing information about travel means or other dynamic domain). Furthermore some actions can actually modify the stored information (consider booking at a restaurant, occupying a table).

The usually chosen strategy to build this kind of systems, excluding button-based flows where the initiative of dialog is completely owned by the agent, is to map sentences expressed by a user onto a fixed set of intents (the sentence types) and slots (entities mentioned that together with a role are used as parameters). The approaches for those two tasks, namely intent classification and slot filling [23], usually make use of Recurrent Neural Networks (RNN) [12] that are able to work on sequence of words. The state-of-the-art condition is currently achieved with a joint approach [21], that makes use of encoder-decoder structure originally born for translations [11] in order to perform both a sequence tagging (slots tagging) and sentence categorization (intent). This approach is very good at working on single-turn cases: each sentence is processed autonomously.

When trying to deal with multi-turn interactions, a lot of problems arise. First of all, the presence of follow-up questions: resolving the references to entities is not trivial when done without apparent evidences. Then, the agent may ask some questions to the user to elicit some missing required parameters: the answers from the user can be fully structured, with signs that underline the entities that are there, or can contain only text that has to be further processed. Goal-oriented agents usually have a dialogue state-tracker component to manage multi-turn interactions, but their dynamicity understanding when to keep the interaction context and when to discard it, receiving signals from the current sentence, is a very critical point. The most common and easy solution is to have universal commands that can be used to stop the current transaction and begin a new one. Some recent approaches [7, 9, 10] to multi-turn problem use memory networks, applied to track the user's goal over the conversation as in the Dialog State Tracking Challenge [36].

2.3 The point of contact

As it has been noticed, those two macro categories of conversational systems would benefit a mutual integration. From the side of goal-oriented, the missing point is being able to answer to complex queries. For this problem a possible solution is to have hierarchical sentence classification: the coarse-grained types can keep corresponding to intents as they convey trait values on the whole sentence, while a fine-grained analysis can be done using the techniques from QA systems in order to build detailed structured queries. For this second step, a detailed parsing has to be done on the sentences to capture all the entities mentioned and their relations. A statistical parsing of such nature, to be fully exploited, needs an ontology whose entities and relations can be explored dynamically instead of being visible only through a finite set of APIs.

Instead QA systems need a more natural and conversational interaction, enabled by some context. The context can be of three types:

domain: a specialized knowledge, including domain-specific datasets for understanding better how to turn the natural language sentences into interrogations. This is already included in domain-specific QA;

interaction: going beyond the fixed form of atomic question-answer pairs. Human to human conversations rely a lot on the interaction context, referring explicitly or implicitly to things that have been previously said. A multi-turn environment can allow users to do questions and later refining them to find what they were searching for, by simply adding new parameters instead of rebuilding a complete independent interrogation, or doing follow-up questions on the previous results;

user: knowing better the user who is asking the questions can be advantageous to find results that are more relevant to him.

An interesting approach that combines information retrieval methodologies using structured knowledge with goal-oriented systems is proposed in [13], which aims to overcome the main issue of goal-oriented systems: learn how to extract information from a KB without the need of handcrafted dialogue state tracking. This is possible thanks to the KB structure that enables a good exchange of data between the conversation and the KB without the need of intent tracker and relying only on latent neural embeddings. This approach mainly is a sequence-to-sequence generator enriched by a KB that provides triples of *(subject, relation, value)*. The values are available to the output generation thanks to some placeholders in the output dictionary in the form of `subject_relation`, that is later replaced by its value after decoding. As can be derived, this approach really combines the techniques of QA over structured knowledge with goal-oriented conversations. But it has two main limitations. The first one is that the KB entries cannot be searched by value, so a question like "Which appointments I have at 9pm?" cannot generate a response targeted to the value "9pm" because the value is not considered. The second limitation is that this approach has no way to perform actions and therefore can act as a readonly service. Removing the intermediate handcrafted level and letting the conversation flow in encoder-decoder fashion, actually makes impossible to call some actions, unless other techniques are found to semantically correlate them.

3 APPROACH

In this section we first give an overview of our approach with its features and then we describe the novelties with respect to the state-of-the-art system [21].

3.1 High level overview

Our approach uses a bidirectional sentence encoding that summarizes the values of the embeddings at word-level into one low-dimensional array, taking the outputs after providing all the words embeddings as inputs. Not only the words of the user, but also the

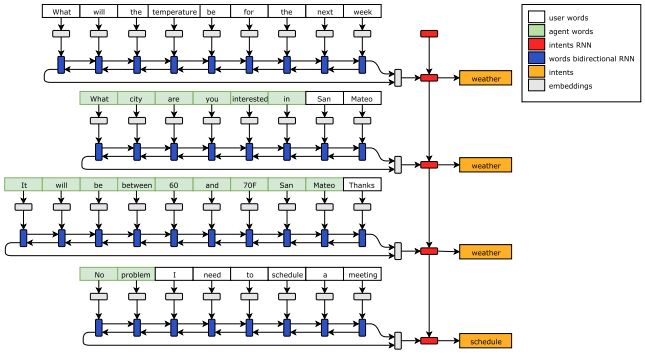

Figure 1: Sentences are encoded in fixed-length vectors using the word-level bidirectional RNN (blue in the figure). The outputs of this first RNN are passed to a second RNN (red in the figure) which models the intent propagation of the user sentences over the different timesteps. This last RNN (the red one) for each user sentence produces the contextualized intent value. The agent words are in green, while the words generated by the human are in white boxes.

ones of the agent are processed. This sentence representation in single-turn NLU approaches is used to categorize the sentence over a set of intents [21]. Instead our representation is fed to the top-level RNN to provide outputs that depend from both current sentence and the surrounding interaction context. Three main challenges are addressed:

- **detect the change of intent in a multi-turn environment:** in other words, to understand dynamically when a certain session (sequence of messages related to a single intent) ends in favour of a new one. This corresponds to choose for each input sentence whether to keep the value of the previous intent or to consider some evidence on the current input. The first case happens when the input sentence is part of a preceding session, and the user is simply continuing the interaction with the same initial intent. The second case instead is when a new intent is expressed in the current sentence, signalling an intent change;
- **capture intent dependencies using the RNN:** capturing the sequences of intent values, a better prediction of the sentence can be done knowing the proceeding intents. This can be quite useful with sentences that are not so expressive because they are referring implicitly to some context of the interaction;
- **consider the current agent turn words:** having a knowledge about what has been replied to the user can help contextualize the new sentence that may not have evident indicators of the intent;

Figure 1 illustrates our approach. In literature also other studies have been done on the problem of sentence classification inside an interaction context. The approach proposed in [37] on the classification of the domain, that is like the high level class in a hierarchical intent classification, uses the previous model prediction at word-level, concatenating it with each word vector. The approach proposed in this work is different both in the specific point where the previous classification is used (not together with the input words but on the sentence level, using the high-level RNN) and also in the way the word-level features are summarized in sentence-level features and considered for next interactions by the learning network.

3.2 Novelty

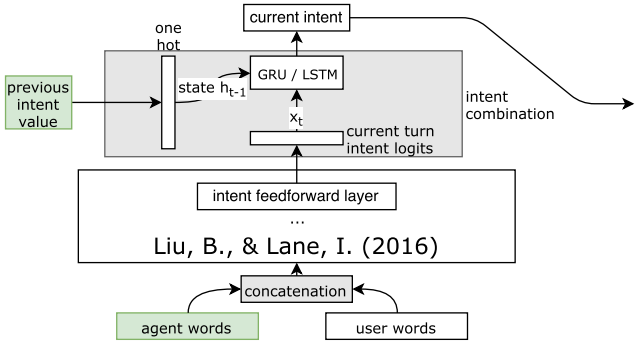

Figure 2: The modifications to [21].

Our approach introduces two main novelties: first, as we can observe in Figure 2, we consider the previous intent value on top of the intent logits[4] that come out from the original network [21]. In particular, the previous intent value is turned into a one-hot vector that is passed as previous internal state h_{t-1} to a RNN cell. Two types of cells have been studied: the GRU cell [37] and the LSTM cell [18] that is commonly used and has two vectors that are carried over timesteps: the cell state c_{t-1} and the output vector h_{t-1}. The GRU cell has less parameters than the LSTM and has shown on different fields to have quite the same performance. For this reason both LSTM and GRU cells are considered as valid alternatives in Section 4.2. The RNN cell, used with all inputs and outputs of size $n\_intents$, takes as input at the current timestep x_t the logits on the current turn. It combines them with the previous intent and thanks to the internal gates it outputs values that represent the contextualized value of the current intent. The key elements are the reset and update gates that are internal to the cell itself and they allow to keep the previous intent value, consider the features of the current sentence, and to learn the intent dependencies. This hierarchical use of RNN (one for encoding sentences and the other one for considering the chain on sentence-level features) is also proposed in [27] but in the domain of query-suggestions: the difference with the proposed approach is that the top level RNN outputs an encoded value that is used for decoding, while here the intent RNN works in the intent space, having as hidden dimension the size of the intent vocabulary and therefore outputs directly the intent logits that passing through a softmax produce the categorization label. In addition, our approach concatenates the previous agent turn together with the current user turn, following the idea that having a knowledge of what a user is replying to can help understanding better his request.

[4] A *logit* is simply an output of a prediction with values that are not yet normalized to a probability distribution by a function like *softmax*.

For having better performance also on smaller datasets whose dictionary can be small, and with the need to be ready for an online setting where words out-of-dictionary may be received, the choice on the input embedding layer has been to use GloVe [24] word vectors pretrained on the CommonCrawl corpus [8]. In this way the embedding values, covering a dictionary of 1.1 million different English words, are set to be non-trainable.

4 EXPERIMENTATION

We first describe the rationale of using the Key-Value Retrieval dataset, then we present the experimentation protocol and, finally, the evaluation results.

4.1 Dataset

The setup of the experiment required a search of a standardized benchmark dataset. The ATIS dataset [17], historically used on the two tasks of intent classification and slot filling, does not fit the experimental setup because of its single-turn orientation.

The problem of multi-turn has been analyzed mostly on proprietary datasets [5, 25, 37], or requiring a participation in a challenge (like the Dialogue State Tracking Challenge [36]). Others, focusing mainly on memory networks and simple questions, are not very relevant for the multi-turn goal-oriented problem (bAbI). The Frames dataset [3] is publicly available but it only contains a single intent ("book"), and focuses more on the tracking of user and machine actions. The only one that has been found is the Key-Value Retrieval published in [13].

This dataset contains multi-turn sessions corresponding to dialogues between a driver and his car assistant. Each dialogue begins with a user sentence that establishes one out of three intents, and the following turns are used by both parties to reach the goal of the driver. Some slots are annotated after each sentence to indicate which entities the system should keep considering the context. About the annotation of the slots, they are available but not annotated in a straightforward way: each slot (15 types available) is stored with its value, but there are some problems in identifying their displacement in the sentences. The selected approach, while providing also outputs for the slot labels being trained jointly, is here analyzed only under the point of view of the intents, so this is actually not a problem. This dataset, therefore, satisfies our needs: each sentence is annotated with its speaker and the intent values are available. The preprocessing is composed of three steps: *i)* annotation of the intent from session-level to sentence-level by copying the values; *ii)* concatenation of all the sentences, removing the concept of session that remains only on the intent values; *iii)* consider as samples only the driver sentences, each one stored together with the current and previous intent value and with the previous sentence of the agent.

With this setup of the samples, on the train set there are 1583 intent changes over 6429 samples, while on the test set 189 changes over 820 samples.

4.2 Evaluation protocol and results

The goal is to measure how the system models the intents. We evaluate the transitions of their values. So the most appropriate measure is the F1 over the intent changes. Being the previous state fixed both in true conditions and on the expected conditions considered for the F1 measure, evaluating the state transition or the destination intent leads to the same values. For this reason, the F1 measure is evaluated on the current sentence intent.[5]

Table 1: F1 over the test test. F1 scores represent the max values reached at the given epoch indicated in the table.

Row	Approach	F1(intent)	epoch
1	our approach with LSTM	**0.9987**	**7**
2	our approach without agent words LSTM	**0.9987**	8
3	our approach with GRU	0.9975	14
4	[21] with the extension of agent words	0.9951	5
5	our approach without agent words GRU	0.9585	9
6	[21]	0.8524	8
7	CRF on pretrained word embeddings	0.7049	100
8	CRF on words	0.4976	100

We compared our approach with the state-of-the-art approach for single-turn [21].[6] To measure separately the effects of the two modifications that have been described in Section 3.2, two more approaches have been considered: the first one considers the original single-turn network with the only addition of the agent words, while the second one considers the proposed multi-turn without the agents words (resulting in the only addition of the top-level RNN working on the intent values). We extended the comparison to a CRF [20] simply applied at word-level with words as inputs and intent labels as outputs. In this case two different configurations have been used: in the first one, the lower cased words are used as input features, while in the second one the pre-trained word embeddings [8].

Table 1 reports the results of the F1 measure on the selected approaches. From the results obtained, we can observe that the role of the interaction context is crucial to perform a better understanding. Natural Language dialogues have great dependencies between the sentences used by both parties. The experimental results show also that the intent changes are correctly detected on the sequence of input samples. We can observe that, considering only the previous value of the intent without concatenating the agent words, gives also an increase with respect to the single-turn model. Then, by looking at the F1 measure change between the couples of rows (1,2), (3,5) and (4,6), we can notice that the agent words on their own give an important contribution in terms of both score and epoch number. Combining both modifications helps going a little bit higher with the score achieving the top score faster. The comparison with the simple CRF approach highlights how important is to work on a properly encoded sentence using RNN.

We acknowledge that the top scores are really close each other's, actually changing the output on only one or two samples from 100%. For this reason a similar work on other datasets may show which one of the two novelties is more important. But, being the distance from the single-turn approach [21] more consistent (delta difference of 0.1463 in F1), we can be sure that the multi-turn classification is important.

[5] micro F1 is used: globally counting the total TP, FN and FP over the single sentences
[6] the approach was reimplemented using https://github.com/HadoopIt/rnn-nlu

5 CONCLUSIONS

The initial research question about the importance of the interaction context for better understanding the requests has been analyzed and the results achieved in a controlled experiment using a standard benchmark dataset showed that the interaction context is very crucial in multi-turn interactions.

The work focused on the first step required for QA, understanding the sentences, that is very important for doing the next steps of knowledge base interrogation and response generation.

The analysis has been done on the intents only. In order to have a rule-free context management it is necessary to perform a similar work also on the entities to know which ones (implicitly or explicitly referenced in the current sentence) have to be kept into consideration into the current context.

Future works may thus include a focus on the entities: both for having a correctly preprocessed corpus, both for including their propagation across turns inside the model. We will obtain a contextualized representation of the current sentence not only in terms of intent, but also with respect to the entities. In this way, no more manual tracking of dialogue components will be necessary and the agent will be able to understand multi-turn interactions seamlessly.

REFERENCES

[1] Abdalghani Abujabal, Mohamed Yahya, Mirek Riedewald, and Gerhard Weikum. 2017. Automated template generation for question answering over knowledge graphs. In *Proceedings of the 26th international conference on world wide web*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 1191–1200.

[2] James F Allen, Donna K Byron, Myroslava Dzikovska, George Ferguson, Lucian Galescu, and Amanda Stent. 2001. Toward conversational human-computer interaction. *AI magazine* 22, 4 (2001), 27.

[3] Layla El Asri, Hannes Schulz, Shikhar Sharma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman. 2017. Frames: A corpus for adding memory to goal-oriented dialogue systems. In *Proceedings of the SIGDIAL 2017 Conference*. Association for Computational Linguistics, Saarbrücken, Germany, 207–219.

[4] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. 1533–1544.

[5] Aditya Bhargava, Asli Celikyilmaz, Dilek Hakkani-Tür, and Ruhi Sarikaya. 2013. Easy contextual intent prediction and slot detection. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 8337–8341.

[6] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. ACM, 1247–1250.

[7] Antoine Bordes and Jason Weston. 2017. Learning end-to-end goal-oriented dialog. *the International Conference on Learning Representations (ICLR)* (2017).

[8] Christian Buck, Kenneth Heafield, and Bas van Ooyen. 2014. N-gram Counts and Language Models from the Common Crawl. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, Vol. 2. Citeseer, Reykjavik, Iceland, 4.

[9] Po-Chun Chen, Ta-Chung Chi, Shang-Yu Su, and Yun-Nung Chen. 2017. Dynamic Time-Aware Attention to Speaker Roles and Contexts for Spoken Language Understanding. *The 2017 IEEE Automatic Speech Recognition and Understanding Workshop* (2017).

[10] Yun-Nung Chen, Dilek Hakkani-Tür, Gokhan Tur, Jianfeng Gao, and Li Deng. 2016. End-to-End Memory Networks with Knowledge Carryover for Multi-Turn Spoken Language Understanding. *Interspeech* (2016), 3245–3249.

[11] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. (October 2014), 1724–1734. http://www.aclweb.org/anthology/D14-1179

[12] Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science* 14, 2 (1990), 179–211.

[13] Mihail Eric and Christopher D Manning. 2017. Key-Value Retrieval Networks for Task-Oriented Dialogue. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*. Association for Computational Linguistics, Saarbrücken, Germany, 37–49.

[14] Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2014. Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1156–1165.

[15] Barbara J Grosz, Scott Weinstein, and Aravind K Joshi. 1995. Centering: A framework for modeling the local coherence of discourse. *Computational linguistics* 21, 2 (1995), 203–225.

[16] Sanda M Harabagiu, Dan I Moldovan, Marius Pasca, Mihai Surdeanu, Rada Mihalcea, Roxana Girju, Vasile Rus, V Finley Lacatusu, Paul Morarescu, and Razvan C Bunescu. 2001. Answering Complex, List and Context Questions with LCC's Question-Answering Server.. In *TREC*.

[17] Charles T Hemphill, John J Godfrey, and George R Doddington. 1990. The ATIS spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*.

[18] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[19] Brendan Juba and Madhu Sudan. 2008. Universal semantic communication ii: A theory of goal-oriented communication. In *Electronic Colloquium on Computational Complexity (ECCC)*, Vol. 15.

[20] John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. (2001).

[21] Bing Liu and Ian Lane. 2016. Attention-Based Recurrent Neural Network Models for Joint Intent Detection and Slot Filling. *Interspeech* (2016), 685–689.

[22] Denis Lukovnikov, Asja Fischer, Jens Lehmann, and Sören Auer. 2017. Neural network-based question answering over knowledge graphs on word and character level. In *Proceedings of the 26th international conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1211–1220.

[23] Grégoire Mesnil, Xiaodong He, Li Deng, and Yoshua Bengio. 2013. Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding.. In *Interspeech*. 3771–3775.

[24] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[25] Yangyang Shi, Kaisheng Yao, Hu Chen, Yi-Cheng Pan, Mei-Yuh Hwang, and Baolin Peng. 2015. Contextual spoken language understanding using recurrent neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 5271–5275.

[26] Robert F Simmons. 1970. Natural language question-answering systems: 1969. *Commun. ACM* 13, 1 (1970), 15–30.

[27] Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 553–562.

[28] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*. 2440–2448.

[29] Mingyu Sun and Joyce Y Chai. 2007. Discourse processing for context question answering based on linguistic knowledge. *Knowledge-Based Systems* 20, 6 (2007), 511–526.

[30] Valentin Tablan, Danica Damljanovic, and Kalina Bontcheva. 2008. A natural language query interface to structured information. In *European Semantic Web Conference*. Springer, 361–375.

[31] Stefanie Tellex, Boris Katz, Jimmy Lin, Aaron Fernandes, and Gregory Marton. 2003. Quantitative evaluation of passage retrieval algorithms for question answering. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. ACM, 41–47.

[32] Alan M Turing. 1950. Computing machinery and intelligence. *Mind* 59, 236 (1950), 433–460.

[33] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85.

[34] Courtney Wade and James Allan. 2005. *Passage retrieval and evaluation*. Technical Report. University of Massachusetts, Amherst Center for Intelligent Information Retrieval.

[35] Joseph Weizenbaum. 1966. ELIZA - a computer program for the study of natural language communication between man and machine. *Commun. ACM* 9, 1 (1966), 36–45.

[36] Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. 2013. The dialog state tracking challenge. In *Proceedings of the SIGDIAL 2013 Conference*. 404–413.

[37] Puyang Xu and Ruhi Sarikaya. 2014. Contextual domain classification in spoken language understanding systems using recurrent neural network. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 136–140.