Preprocessing Techniques for End-to-End Trainable RNN-based Conversational System

Hussein Maziad[1], Julie-Ann Rammouz[1], Boulos El Asmar[2], and Joe Tekli[1*]

[1] E.C.E. Department, Lebanese American University, 36 Byblos, Lebanon
{hussein.maziad, julieann.rammouz}@lau.edu, joe.tekli@lau.edu.lb

[2] Logistics Robotics, BMW Group, 80788 Munich, Germany
boulos.el-asmar@bmw.de

Abstract. Spoken dialogue system interfaces are gaining increasing attention, with examples including Apple's Siri, Amazon's Alexa, and numerous other products. Yet most existing solutions remain heavily data-driven, and face limitations in integrating and handling data semantics. They mainly rely on statistical co-occurrences in the training dataset and lack a more profound knowledge integration model with semantically structured information such as knowledge graphs. This paper evaluates the impact of performing knowledge base integration (KBI) to regulate the dialogue output of a deep learning conversational system. More specifically, it evaluates whether integrating dependencies between the data, obtained from the semantic linking of an external knowledge base (KB), would help improve conversational quality. To do so, we compare three approaches of conversation preprocessing methods: i) no KBI: considering conversational data with no external knowledge integration, ii) All Predicates KBI: considering conversational data where all dialogue pairs are augmented with their linked predicates from the domain KB, and iii) Intersecting Predicates KBI: considering conversational data where dialogue pairs are augmented only with their intersecting predicates (to filter-out potentially useless or redundant knowledge). We vary the amount of history considered in the conversational data, ranging from 0% (considering the last dialogue pair only) to 100% (considering all dialogue pairs, from the beginning of the dialogue). To our knowledge, this is the first study to evaluate knowledge integration in the preprocessing phase of conversational systems. Results are promising and show that knowledge integration – with an amount of history ranging between 10% and 75%, generally improves conversational quality.

Keywords. Conversational dialogue systems, data semantics, knowledge base, knowledge integration, conversational data preprocessing.

1 Introduction

Spoken dialogue system interfaces are gaining increasing attention, with examples including Apple's Siri, Google Assistant, Microsoft's Cortana, Amazon's Alexa, and numerous other products. Most existing solutions utilize deep learning, where recurrent neural networks (RNNs) have been successfully adapted to dialogue systems through encoder-decoder architectures [31]. While the main advantage of deep (RNN) learning is its reduced feature engineering, it often requires large amounts of labeled data (which are not always available), and purely data-driven learning can lead to unexpected results (depending on the quality of the training

[*] Corresponding author

data) [20, 29]. In this context, recent works in language representation and processing, e.g., [1, 5, 12, 16], have investigated the integration of external domain knowledge to augment the training of deep learners. Yet their applications and related data preprocessing do not target conversational dialogue systems.

The development of an RNN-based dialogue system consists of four main steps: i) preprocessing the conversational dataset at hand to use as the training data, ii) building the RNN responsible for inferring dialogue policies from the conversational data, iii) training the model with and without testing and eventually preprocessing external knowledge, and iv) representing the external knowledge alongside the conversational data to compare the obtained results. In this context, data preprocessing techniques for end-to-end RNN-based conversational systems seem to lack common grounds and comparative evaluations. Results in [24] show that data representation plays a crucial role in the performance of a neural network. In other words, the initial preprocessing step, including input data and context representation, is of central importance in building an end-to-end RNN-based dialogue system, and needs to be properly designed and fine-tuned before diving deeper into external knowledge integration and processing.

This paper evaluates the impact of performing knowledge base integration (KBI) to regulate the dialogue output of a deep learning conversational system. More specifically, it evaluates whether integrating dependencies between the data, obtained from the semantic linking of an external knowledge base (KB), would help improve conversational quality. In contrast with most existing solutions (cf. Section 2), where the authors rely solely on the quality of the training data to improve conversational quality, this study aims at evaluating whether integrating additional dependencies between the data, obtained from the semantic linking of an external KB, would help improve conversational quality. To do so, we evaluate and compare three approaches of conversation preprocessing methods: i) *No KBI*: considering conversational data with no external knowledge integration, ii) *All Predicates KBI*: considering conversational data where all dialogue pairs are augmented with their linked predicates from the domain KB, and iii) *Intersecting Predicates KBI*: considering conversational data where dialogue pairs are augmented only with their intersecting (common) predicates (in order to reduce and filter-out potentially useless or redundant knowledge). For each of the mentioned approaches, we vary the amount of history considered in the conversational data, ranging from 0% (considering the last dialogue pair only) to 100% (considering all dialogue pairs, from the beginning of the dialogue). To our knowledge, this is the first study to evaluate knowledge integration in the preprocessing phase of conversational systems. Results are promising and show that knowledge integration – with an amount of history ranging between 10% and 75%, generally improves conversational quality.

The remainder of this paper is organized as follows. Section 2 briefly reviews the related works. Section 3 describes our proposal and the suggested conversation preprocessing methods. Section 4 describes our experimental evaluation and results, before concluding with future work in Section 5.

2 Related Works

2.1. Conversational Systems

The main functionality of a spoken dialogue system consists in decoding text utterances to extract semantic information through spoken language understanding techniques [34]. The semantic representation of every utterance is then processed by a dialogue state tracker, which estimates the dialogue state in order to decide what action to take, according to a pre-defined dialogue policy. Such modular architectures depend to a large extent on a series of hierarchical handcrafted rules to adapt the dialogue policy according to the detected entities and to the utterance intent. This may work, with much effort, for restricted domains where the number of intents is generally limited. However, the extraction of semantic information becomes much more intricate when shifting to a more general domain environment, or when the dialogue is required to cover more features during the conversation. Providing an exhaustive list of references for such traditional dialogue systems is out of the scope of the current work. However, recent advances in end-to-end training of neural networks, along with the availability of large-scale conversation datasets [23] has permitted to directly infer dialogue policy from conversational data. Notably, recurrent neural networks (RNNs) have been successfully adapted to dialogue systems through encoder-decoder architectures [31]. While the main advantage of (deep) RNN learning is its reduced feature engineering, yet it often requires large amounts of labeled data (which are not always available), and the purely data-driven learning can lead to unexpected results (depending on the quality of the training data) [20, 29]. However, integrating domain knowledge, in the form of an external knowledge base (KB) semantic linking approach – which we refer to as KB Integration (KBI) – has many advantages, including: i) resolving ambiguity in language, ii) performing semantic-aware data integration, and iii) linking conversations with relevant documents and meta-data through semantic search and semantic similarity evaluation. KBI does introduce an increase in training time and computation, which might not be a major concern since the training is done offline, prior to system run-time.

2.2. Generative Sequence-to-Sequence Deep Learning Models

Generative sequence-to-sequence (seq2seq) models follow the line initiated by Ritter et al. [22] who treats the generation of conversational dialogue as a statistical machine translation problem. Seq2seq models have recently shown promising results, mapping complicated structures together. This has direct applications in natural language understanding [28], and in dialogue response generation by mapping queries with responses [26], such as in recent works [25, 26] where RNNs have been used to model dialogue in short conversations. Seq2seq models have also been used for neural machine translation [2, 15, 28], and have achieved remarkable results in syntactic constituency parsing [30], and in image captioning [32]. As it is the case for most deep learning models, seq2seq requires little feature engineering and domain specificity whilst matching or surpassing state-of-the-art results. However, these models, being based on recurrent neural networks (RNNs), suffer from the vanishing gradient problem, that's why variants of Long Short-Term Memory (LSTM) RNNs [11] are mostly used. Yet it is often very hard to control the output of such models, primarily determined by statistical co-occurrences in the used training data with limited synthesis of additional external knowledge (cf. Section 2.3). Furthermore, such approaches are still unable to generate coherent responses [17] which remains a major drawback for conversational dialogue applications.

2.3. Deep Neural Models with External Knowledge

Incorporating external knowledge within deep neural models has been of increasing interest recently, promising to enhance generalization, increase interpretability, and control network output. Recent works in [12, 13], have focused on transferring logical knowledge into diverse neural network architectures by imposing posterior constraints on the network. Also, the authors in [5] have used a structured label relation graph to improve object classification. Other approaches integrate domain knowledge on training time, consist in integrating first order logic with Bayesian models [6], or deriving probabilistic graphical models for Markov logic networks from a set of rules. Also, in [1] a novel neural knowledge language model was developed, bringing symbolic knowledge from a knowledge graph into the expressive power of RNN language models. In a related study, the authors in [17] improve generative models by learning external knowledge, represented as distributed embeddings, and refined during training time to increase model consistency and infer speaker-specific characteristics. External knowledge integration approaches have also been investigated with neural networks using external memory [4, 14, 27], where a long-term memory structure acts as a (dynamic) knowledge base. However, the latter approaches focus on learning attention models over unstructured data, whereas we aim to link to entities using a structured knowledge base (KB).

3 Proposal: Preprocessing Techniques for Conversational RNN

We process the dialogue as a seq2seq learning problem within a neural encoder-decoder architecture. The overall architecture of our approach is shown in Fig. 1.

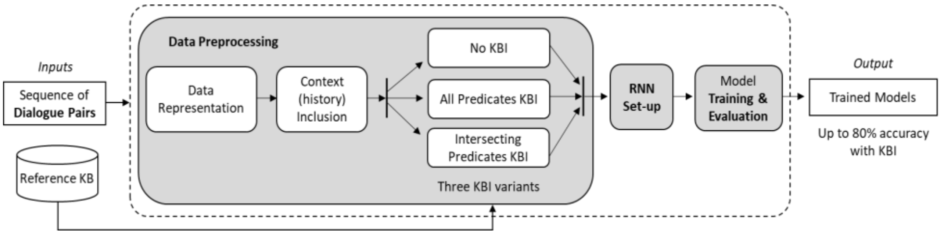

Inputs
Sequence of Dialogue Pairs
Reference KB
Data Preprocessing
Data Representation
Context (history) Inclusion
No KBI
All Predicates KBI
Intersecting Predicates KBI
Three KBI variants
RNN Set-up
Model Training & Evaluation
Output
Trained Models
Up to 80% accuracy with KBI

Fig. 1. Simplified activity diagram describing our approach

3.1. Data Representation

We adopt a typical data representation model where dialogue utterances are represented as a sequence of user requests *u* and system responses *s*: (u_1, s_1), (u_2, s_2), …, (u_k, s_k), and *k* represents the number of turns in the dialogue [33]. More specifically, we first parse the raw text, split it into conversations, and then split every conversation into turns, where every turn consists of a pair of user utterance and the corresponding system response. In order to easily process the dialogue within a seq2seq learning problem, we represent each word in every utterance and reply as a one-hot vector[1]. Additionally, we use a unique index per word to represent the input and targets of the network.

Note that the different preprocessing techniques considered in this study will produce different variations of the above-mentioned data representation, which we present and discussion in the following sections.

3.2. Context and History

We define the context of a dialogue training pair (u_i, s_i) as ([u_1, u_2,…, u_{i-1}, u_i], s_i), where the network is fed: a concatenation of all previous user utterances (i.e., the user request history) up until the current one u_i, as well as the current system response s_i. Accordingly, we define the history of a dialogue as the context of the dialogue starting from the present utterance, by specifying the percentage of the previous utterance that will be included in the following one. For instance, ([$u_1^{50\%}$, $u_2^{50\%}$,…, $u_{i-1}^{50\%}$, u_i], s_i) represents 50% of (u_i, s_i)'s dialogue history where $u_{i-1}^{50\%}$ represents half of u_{i-1}'s textual tokens, and so forth. Similarly, ([$u_1^{100\%}$, $u_2^{100\%}$,…, $u_{i-1}^{100\%}$, u_i], s_i) represents 100% of the dialogue history, and is equivalent to the complete context of training pair (u_i, s_i), i.e., ([u_1, u_2,…, u_{i-1}, u_i], s_i).

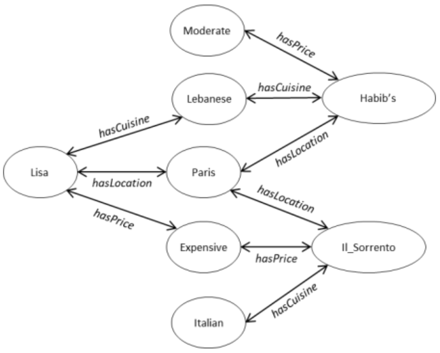
3.3. RNN Set-up

We utilize a typical seq2seq network where two RNNs work together in order to transform one sequence into another. The first network, i.e., the encoder, reads the input sequence and condenses it into a vector using typical one-hot-encoding. The decoder network reads the vector and its context and transforms it into an output sequence. More specifically, the decoder network accepts as input the context vector which includes the history of the entire sequence. At every decoding stage, the decoder is given an input token and a hidden state where the context vector serves as an initial hidden state. A problem with typical decoders is that they process the complete context vectors which carry the dialogue's entire sequence (100% history). For this reason, and in order to improve our model, we add an attention mechanism which teaches the decoder to focus on a particular part of the input sequence. To do so, we compute a set of attention weights, and multiply them by the encoder output vectors to create a weighted combination which contains information about the specific part of the input sequence that helps the decoder produce the right output sequence. The attention weights are calculated using an additional feed-forward layer, which accepts as input the decoder's input and hidden states and produces the weights accordingly.

3.4. Knowledge Representation

We represent domain knowledge in the form of a machine readable knowledge base (KB), consisting of nodes and edges, where nodes represent groups of words/expressions and edges represent the semantic links connecting the nodes (synonymy, hyponymy (Is-A), meronymy (Part-Of), etc. [19]). The latter can also be represented as sets of triplets: *concept$_1$-relationship-concept$_2$*, or as more commonly known: *subject-predicate-object* triplets [10] (cf. Fig. 2).

[1] It is called one-hot because only one bit is "hot" or TRUE at any time. For example, a 3-bit one-hot encoding would have three states: 001, 010, and 100, compared with 2^3 binary combinations obtained with binary encoding. Note that other encodings such as word2vec and GloVe vetor representations can be used.



Subject	Predicate	Object
Liza	*hasPrice*	Expensive
Il_Sorrentino	*hasPrice*	Expensive
Chez_Pippo	*hasPrice*	Expensive
Roadsters	*hasPrice*	Moderate
Il_Sorrentino	*hasLocation*	Paris
Chez_Pippo	*hasLocation*	Paris
Liza	*hasLocation*	Paris
Roadsters	*hasLocation*	Beirut
Il_Sorrentino	*hasCuisine*	Italian
Napoli	*hasCuisine*	Italian
Chez_Pippo	*hasCuisine*	French
Liza	*hasCuisine*	Lebanese
Roadsters	*hasCuisine*	Lebanese

Fig. 2. Sample KB tailored base on DSRC2 [7]

In the following, we evaluate the impact of performing KB integration (KBI) at the data preprocessing stage, to regulate the dialogue output of a conversational system.

3.5. Data Preprocessing Techniques

As motivated previously, data preprocessing in conversational systems is of central importance in building an end-to-end RNN approach. For this reason, we propose and evaluate three methods for KBI at the input and output of the encoder and decoder networks: i) *No KBI*, ii) *All Predicates KBI*, and iii) *Intersecting Predicates KBI*. These methods can be applied to preprocess the training, the validation, and the testing data sets.

3.5.1. No Knowledge Base Integration (No KBI)

This is the elementary approach where the dialogue exchange is represented in its most basic form: (u_1, s_1), (u_2, s_2), . . . (u_k, s_k), i.e., as a sequence of pairs of user utterance u and system reply s tokens. We do not consider any external knowledge here and vary dialogue history to include more or less of the previous user utterances following our context model: $([u_1, u_2,, \ldots, u_i], s_i)$ where the system is fed an amount of previous user utterances following user-specified history percentage (cf. Section 3.2).

Consider for instance the examples in Table 1: 0% history in considered in Table 1.a where no previous user utterances are included in the training pairs, 50% history is considered in Table 1.b where the first half of the previous user utterances is included in the training pair, and 100% history is considered in Table 1.c where all the previous user utterances are included in the training pair. By observing this representation, one can notice that the size of the training data increases significantly with the increase in the percentage of history. This will impact both training speed and quality as we will observe in the experimental evaluation section.

Table 1. Sample examples for *No KBI* preprocessing approach[1]

Text in red represents the additions from previous user utterance history

a. 0% history

Pair	User Utterance	System Response
1	Hello	Hi, how can I help you?
2	I would like to book a reservation at an expensive restaurant in Paris	Sure, I will look one up
3	<Silence>	Do you have a cuisine in mind?
4	Italian would be great	Okay, how about Il Sorrentino?

b. 50% history

Pair	User Utterance	System Response
1	Hello	Hi, how can I help you?
2	I would like to book a reservation at an expensive restaurant in Paris	Sure, I will look one up
3	<Silence> I would like to book	Do you have a cuisine in mind?
4	Italian would be great <Silence> I would	Okay, how about Il Sorrentino?

c. 100% history No KBI

Pair	User Utterance	System Response
1	Hello	Hi, how can I help you?
2	I would like to book a reservation at an expensive restaurant in Paris Hello	Sure, I will look one up
3	<Silence> I would like to book a reservation at an expensive restaurant in Paris Hello	Do you have a cuisine in mind?
4	Italian would be great <Silence> I would like to book a reservation at an expensive restaurant in Paris Hello	Okay, how about Il Sorrentino?

3.5.2. All Predicates KBI

Our ambition is to integrate knowledge at the level of the preprocessing step. As such, we introduce the *All predicates KBI* representation, which scans each word token in the current user utterance to identify matching objects in the reference KB. Then, it intersects the subjects connected with the matching object predicates and appends them to the user utterance. It also allows appending a varying amount of dialogue history to include more or less of the previous user utterances with their augmented KB subjects.

Consider in Table 2 an extension of the examples from Table 1: 10% history in Table 1.a where no previous user utterances are included in the training pairs, 50% history in Table 1.b where the first half of the previous user utterances is included in the training pair, and 100% history in Table 1.c where all the previous user utterances are included in the training pair.

[1] All examples presented in the paper are prepared based on the real dialogues from the restaurant reservation system dataset: DSTC2 [9], using the dataset's reference KB (cf. Fig. 2).

Table 2. Sample examples extended from Table 1, to represent *All Predicates KBI* preprocessing approach

Text in red represents the additions from previous user utterance history
Text in blue represents the additions of subject tokens from the knowledge base
Test underlined represents a matching user utterance with an object from the knowledge base

a. 0% history

Pair	User Utterance	System Response
1	Hello	Hi, how can I help you?
2	I would like to book a reservation at an expensive restaurant in Paris Liza Chez_Pippo il_Sorrentino	Sure, I will look one up
3	<Silence>	Do you have a cuisine in mind?
4	Italian would be great Il_Sorrentino Napoli	Okay, how about Il_Sorrentino?

b. 50% history

Pair	User Utterance	System Response
1	Hello	Hi, how can I help you?
2	I would like to book a reservation at an expensive restaurant in Paris Liza Chez_Pippo il_Sorrentino	Sure, I will look one up
3	<Silence> I would like to book a reservation at an	Do you have a cuisine in mind?
4	Italian would be great Il_Sorrentino Napoli <Silence> I would like to	Okay, how about Il_Sorrentino?

c. 100% history

Pair	User Utterance	System Response
1	Hello	Hi, how can I help you?
2	I would like to book a reservation at an expensive restaurant in Paris Liza Chez_Pippo il_Sorrentino Hello	Sure, I will look one up
3	<Silence> I would like to book a reservation at an expensive restaurant in Paris Liza Chez_Pippo il_Sorrentino Hello	Do you have a cuisine in mind?
4	Italian would be great Il_Sorrentino Napoli <Silence> I would like to book a reservation at an expensive restaurant in Paris Liza Chez_Pippo il_Sorrentino Hello	Okay, how about Il_Sorrentino?

While the *All Predicates KBI* approach seems promising, it presents a major drawback: for every single matching object token between a training pair and the reference KB, this approach will retrieve all matching subjects whose number might be significantly large, depending on the size of the KB. Consider for instance the example in Table 3, where certain pairs contain hundreds of matching entries, while other pairs contain only a few. This makes the dialogue unbalanced in terms of the number of terms per sequence. While Table 3 presents a case for 0% history, the problem will be further exacerbated when including more of the conversation history, as shown in Table 4.

Note that in the last pair of Table 4, we have thousands of entries, which will probably make the training model overfitted, and will thus reflect badly on the results. In other words, a single training pair with only one matching object token might end up encompassing a huge number of subject tokens, resulting in crowded (oversized) training pairs which would negatively affect both training time and quality (including potentially noisy data entries).

Table 3. Crowded data example for *All Predicates KBI* with 0% history

Pair	User Utterance	System Response
1	Hello	Hi, how can I help you?
2	I would like to book a reservation at an expensive restaurant in Paris Liza Chez_Pippo il_Sorrentino	Sure, I will look one up
3	<Silence>	Do you have a cuisine in mind?
4	Italian would be great (250 italian restaurants augmented here…)	Okay, how about Il_Sorrentino?
5	I think I will go for Lebanese instead (250 Lebanese restaurants augmented here…)	Sure, I have found ten in Paris
6	<Silence>	Anything else?
7	I would rather have them in madrid (300 restaurants in madrid augmented here…)	I will look for restaurants in madrid
8	Please make sure the restaurants are in a moderate price range (1500 moderately priced restaurants augmented here…)	I will have them ready in no time

Table 4. Crowded data example for *All Predicates KBI* with 100% history

Pair	User Utterance	System Response
1	Hello	Hi, how can I help you?
2	I would like to book a reservation at an expensive restaurant in Paris Liza Chez_Pippo il_Sorrentino Hello	Sure, I will look one up
3	<Silence> I would like to book a reservation at an expensive restaurant in Paris Liza Chez_Pippo il_Sorrentino Hello	Do you have a cuisine in mind?
4	Italian would be great (250 Italian restaurants augmented here…) + (18 terms from pair 3)	Okay, how about Il_Sorrentino?
5	I think I will go for Lebanese instead (250 Lebanese restaurants augmented here…) + (~250 from pair 4)	Sure, I have found ten in Paris
6	<Silence> + (~500 from pair 5)	Anything else?
7	I would rather have them in madrid (300 restaurants in madrid augmented here…) + (~500 from pair 6)	I will look for restaurants in madrid
8	Please make sure they are in moderate price range (1500 moderately priced restaurants augmented here…) + (~800 from pair 5)	I will have them ready in no time

3.5.3. Intersecting Predicates KBI

By closely analyzing the example in Table 4, one can realize that the *All Predicates KBI* approach does not accumulate the user's choices on every new request. For instance, if the user asks for "Italian" in pair #3, and then "expensive" in pair #4, we would expect to have in pair #4 restaurants that intersect both descriptions, i.e., "expensive Italian" restaurants, instead of retrieving restaurants of each description separately. To solve this problem, as well as the data crowding issue discussed in the previous sub-section, we introduce a dedicated data structure that maps two entries: one storing keys as the KB predicates, and the other storing values as the KB objects. The contents of this map are updated with new values as the system iterates over each user-system pair. At each pair, we return the intersection of the subjects corresponding to these predicates if the pair contains a predicate itself. We refer to this enhanced approach as the *Intersection Predicated KBI* solution.

Using the same examples from Tables 3 and 4, we showcase the following three samples using the *Intersecting Predicates KBI* approach. In Table 5.b (50% history), we update the map at each pair with the new objects, and then insert the intersection of the corresponding subjects at the end of the same pair. If the pair does not contain an object, then we insert 50% of the previous objects. The same goes for Table 5.c (100% history).

Table 5. Applying *Intersecting Predicates KBI* on the example from Tables 3 and 4.

a. 0% history

Pair	User Utterance	Map	System Response
1	Hello	{ **hasPrice**: “”, **hasLocation**: “”, **hasCuisine** : “” }	Hi, how can I help you?
2	I would like to book a reservation at an expensive restaurant in Paris Liza Chez_Pippo il_Sorrentino	{ **hasPrice**: “expensive”, **hasLocation**: “Paris”, **hasCuisine** : “” }	Sure, I will look one up
3	<Silence>	{ **Price**: “expensive”, **hasLocation**: “Paris”, **hasCuisine** : “” }	Do you have a cuisine in mind?
4	Italian would be great Il_Sorrentino	{ **hasPrice**: “expensive”, **hasLocation**: “Paris”, **hasCuisine** : “Italian” }	Okay, how about Il_Sorrentino?

b. 50% history

Pair	User Utterance	Map	System Response
1	Hello	{ **hasPrice**: “”, **hasLocation**: “”, **hasCuisine** : “” }	Hi, how can I help you?
2	I would like to book a reservation at an expensive restaurant in Paris Liza Chez_Pippo il_Sorrentino	{ **hasPrice**: “expensive”, **hasLocation**: “Paris”, **hasCuisine** : “” }	Sure, I will look one up
3	<Silence> Liza Chez_Pippo	{ **hasPrice**: “expensive”, **hasLocation**: “Paris”, **hasCuisine** : “” }	Do you have a cuisine in mind?
4	Italian would be great Il_Sorrentino	{ **hasPrice**: “expensive”, **hasLocation**: “Paris”, **hasCuisine** : “Italian” }	Okay, how about Il_Sorrentino?

c. 100% history

Pair	User Utterance	Map	System Utterance
1	Hello	{ **hasPrice**: “”, **hasLocation**: “”, **hasCuisine** : “” }	Hi, how can I help you?
2	I would like to book a reservation at an expensive restaurant in Paris Liza Chez_Pippo il_Sorrentino	{ **hasPrice**: “expensive”, **hasLocation**: “Paris”, **hasCuisine** : “” }	Sure, I will look one up
3	<Silence> Liza Chez_Pippo Sorrentino	{ **hasPrice**: “expensive”, **hasLocation**: “Paris”, **hasCuisine** : “” }	Do you have a cuisine in mind?
4	Italian would be great Il_Sorrentino	{ **hasPrice**: “expensive”, **hasLocation**: “Paris”, **hasCuisine** : “Italian” }	Okay, how about Il_Sorrentino?

One can realize that *Intersecting Predicates KBI* allows to gradually converge toward the subject tokens that match the user’s evolving requests, and thus significantly reduces the amount of knowledge added to the individual training pairs, compared with *All Predicates KBI* described previously.

4 Experimental Evaluation

4.1. Experimental Data

To evaluate our approach, we utilize the Dialogue State Tracking Challenge 2 (DSTC2) dataset [9] consisting of restaurant reservation user-system conversation pairs. These dialogues are derived from a real-world system rendering the data raw and real, while training a task-oriented dialogue system. The dialogs come from 6 conditions consisting of the combinations of 3 dialog managers and 2 speech recognizers. There are roughly 500 dialogs in each condition, of average length 7.88 turns from 184 unique users. In our current study, we use the raw version of dataset from [3] which only includes user and system utterances[1]. We also utilize DSTC2's underlying KB[2] (cf. Fig. 2) as the reference source of knowledge when performing KBI. It consists of 8400 subject-predicate-object triplets where subjects represent restaurant names, predicates represent semantic relationships, e.g., *hasPrice*, *hasLocation*, or *hasCuisine*, and objects represent relationship properties, e.g., price could be *cheap*, *moderate*, or *expensive*. For better visualization and understanding of the results, data is pre-processed and cleaned such that all API calls are removed before using the data in any further steps.

4.2. Experimental Results

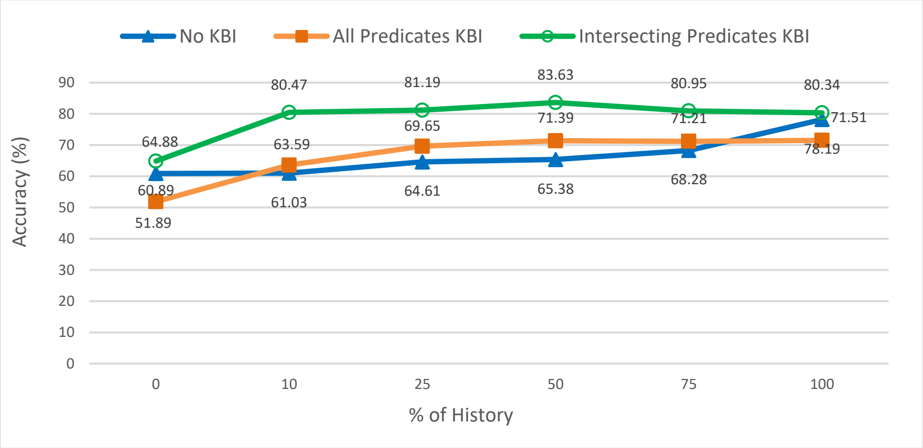
The evaluation of conversational dialogue systems remains an open problem. With the lack of structure in the dialogues, it remains unclear which attributes of the conversation are relevant to measure the response's quality. Evaluations can be of two types: i) coarse-grained, which focus on the appropriateness (accuracy) of a response, and ii) fine-grained, which focus on the specific behaviors a dialogue system should manifest (such as perceived human likeness) [6]. In this study, we adopt the former approach (coarse-grained) and utilize k-fold cross validation applied on each of the three preprocessing variations: *No KBI*, *All Predicates KBI*, and *Intersecting Predicates KBI*. For each variation, we vary the amount of conversational history from 0%, 10%, 25%, 50%, 75%, to 100%. Also, for each amount of history, we perform two degrees of *k*-fold: k= 5 and k=10. This brings the total number of trained models to 3*6*(5+10) = 270, requiring a total number of 2,845 hours to train. For every trained model, we compare the generated system response with the expected response obtained from the reference dataset, and then compute the number of matching responses (i.e., hits). We then evaluate accuracy as the sum of all the matching responses (hits) over the total number of compared responses. Fig. 3 shows the average accuracy levels for the different iterations of each k-fold degree.

Comparing *No KBI* with *All Predicates KBI*: we notice that performing KBI generally improves overall accuracy, except at the boundaries: with 0% and 100% history. This is probably due to the following: i) at 100% history, many subject tokens are added to every training pair which leads to overfitting; ii) at 0% history, some user-system pairs contain thousands of subjects from the KB while other pairs contain only a few or none at all, which renders the training data unbalanced and unpredictable; hence iii) an amount of history between the boundaries allows the training data to become more balanced which generally produces better results.

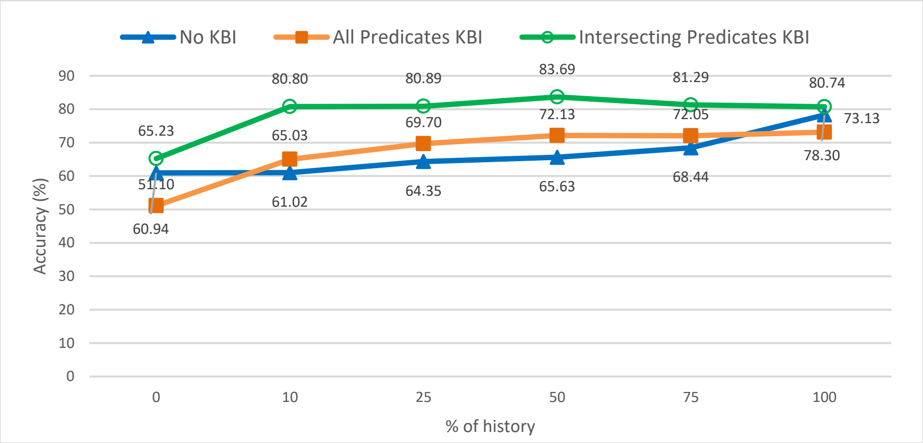
[1] Available online at: https://github.com/HLTCHKUST/Mem2Seq/tree/master/data/dialog-bAbI-tasks

[2] https://github.com/HLTCHKUST/Mem2Seq/blob/master/data/dialog-bAbI-tasks/dialog-babi-kb-all.txt

Comparing *Intersecting Predicates KBI* with alternatives: This approach yields the highest accuracy levels for every history %. This is because it keeps the dataset balanced while making most of the KB by converging to a handful of useful subject tokens as the dialogue evolves. One important observation is that the accuracy of *Intersecting Predicates KBI* peaks almost in the middle of the history % (at around 50%). This concurs with the observations made in the previous paragraph regarding the need for a balanced training set to improve training quality.



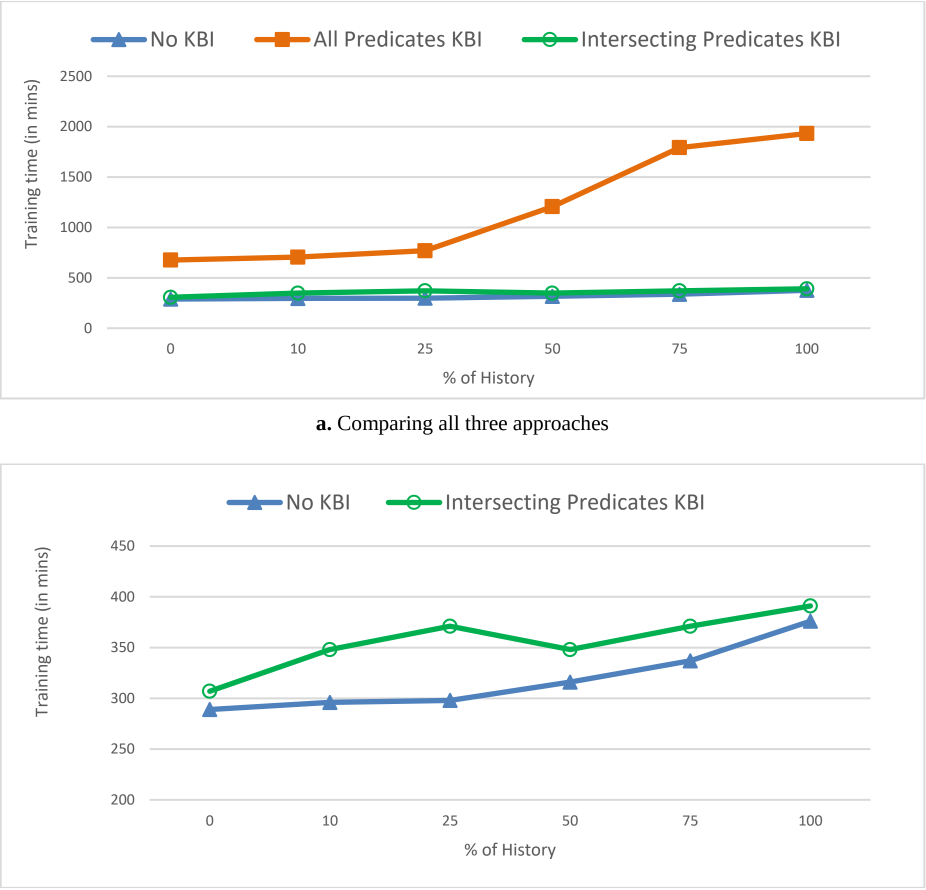
a. Results for k=5



b. Results for k=10

Fig. 3. Accuracy of conversation results applied on DSTC2 dataset, using k-fold cross validation with k=5 and k=10

Concerning training time, results in Fig. 4 show that the *All Predicates KBI* approach introduces a significant increase in training time compared with its counterparts. This is due to the substantial increase in training data with the inclusion of all matching predicates in every training pair, resulting in oversized training pairs which require more time to process and train. While the *Intersection Predicates KBI* approach requires more training time than its *No KBI* counterpart (cf. Fig. 4.b), yet the two approaches are almost undiscernible compared with *All Predicated KBI* (cf. Fig. 4.a).



a. Comparing all three approaches

b. Comparing *No KBI* and *Interescting Predicates KBI* only

Fig. 4. Training time of conversation results applied on DSTC2 dataset, using k-fold cross validation with k=5 (similar results are obtained for k=10, with an average 5% increate in time)

5 Conclusion

Knowledge base integration (KBI) in the training of sequence-to-sequence (seq2seq) generative conversational approaches has not been widely explored so far. In this work, we evaluate and compare three approaches of conversational data preprocessing that involve knowledge integration: i) *No KBI*: considering dialogue pairs with no external knowledge integration, ii) *All Predicates KBI*: where all dialogue pairs are augmented with their linked predicates from the KB, and iii) *Intersecting Predicates KBI*: where dialogue pairs are augmented only with their intersecting predicates from the KB. The latter are prerequisites to generating semantically structured text integrated at training time. Results show that KBI generally improves overall accuracy, except at the boundaries: with 0% and 100% history where the models tend to become either unbalanced due to discrepancies in the sizes of the training pairs (with 0%), or overfitted (at 100%). *Intersecting Predicates KBI* produces the best accuracy levels since it tends to keep the training dataset balanced, compared with its *All Predicates KBI* alternative.

Future work includes evaluating KBI with different conversational models such as BERT [21], XLNet [35], and RoBERTa [18], and comparing them with our seq2seq RNN-based solution. This also requires combining multiple conversational datasets from different domains, along with their reference KBs, to analyze how different models react accordingly. The latter is an important step towards creating a general purpose spoken dialogue system. Considering KBI with multilingual solutions, e.g., [7, 8], is another future direction.

References

[1] Ahn S., et al., *A Neural Knowledge Language Model.* CoRR abs/1608.00318, 2016.

[2] Bahdanau D., Cho K., and Bengio Y., *Neural Machine Translation by Jointly Learning to Align and Translate.* International Conference on Learning Representations (ICLR), 2015.

[3] Bordes A., Boureau Y., and Weston J., *Learning End-to-End Goal-Oriented Dialog.* International Conference on Learning Representations (ICLR), 2017.

[4] Collier M., B.J., *Implementing Neural Turing Machines.* International Conference on Artificial Neural Networks and Machine Learning (ICANN), 2018. pp. 94-104.

[5] Deng J., et al., *Large-Scale Object Classification Using Label Relation Graphs.* European Conference on Computer Vision (ECCV), 2014. pp. 48-64.

[6] Deriu J., et al., *Survey on Evaluation Methods for Dialogue Systems.* CoRR abs/1905.04071, 2019.

[7] Haraty R. and El Ariss O., *Lebanese Colloquial Arabic Speech Recognition.* ISCA International Conference on Computer Applications in Industry and Engineering (CAINE), 2005. pp. 285-291.

[8] Haraty R. and Nasrallah R., *Indexing Arabic Texts using Association Rule Data Mining.* Library Hi Tech, 2019. 37(1): 101-117.

[9] Henderson M. and W.J. Thomson B., *The Second Dialog State Tracking Challenge.* SIGDIAL Conference, 2014. pp. 263-272.

[10] Herrera R. T., et al., *Toward RDF Normalization.* 34th International Conference on Conceptual Modeling (ER'15), , 2015. pp. 261-275, Stockholm, Sweden.

[11] Hochreiter S. and Schmidhuber J., *Long Short-Term Memory.* Neural Computing, 1997. 9(8): 1735-1780.

[12] Hu Z., et al., *Harnessing Deep Neural Networks with Logic Rules.* Annual Meeting of the Association for Computational Linguistics (ACL), 2016.

[13] Hu Z., et al., *Deep Neural Networks with Massive Learned Knowledge.* Conference on Empirical Methods in Natural Language Processing (EMNLP) 2016. pp. 1670-1679.

[14] Jafari R., Razvarz S., and Gegov A., *End-to-End Memory Networks: A Survey.* Science and Information Conference (SAI), 2020. pp. 291-300.

[15] Kalchbrenner N., B.P., *Recurrent Continuous Translation Models.* Conference on Empirical Methods in Natural Language Processing (EMNLP), 2013. pp. 1700-1709.

[16] Karaletsos T., Belongie S. J., and Rätsch G., *When Crowds Hold Privileges: Bayesian Unsupervised Representation Learning with Oracle Constraints.* International Conference on Learning Representations (ICLR), 2016.

[17] Li J., et al., *A Persona-Based Neural Conversation Model.* Annual Meeting of the Association for Computational Linguistics (ACL), 2016.

[18] Liu Y., et al., *RoBERTa: A Robustly Optimized BERT Pretraining Approach.* CoRR abs/1907.11692, 2019.

[19] Miller G.A. and Fellbaum C., *WordNet Then and Now.* Language Resources and Evaluation, 2007. 41(2): 209-214.

[20] Nguyen A., Yosinski J., and Clune J., *Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images.* Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, 2015. pp. 427–436.

[21] Qu C., et al., *BERT with History Answer Embedding for Conversational Question Answering.* Proc. of the 42nd Inter. ACM SIGIR Conference on Research and Development in Information Retrieval, 2019. pp. 1133-1136.

[22] Ritter A., Cherry C., and Dolan W., *Data-Driven Response Generation in Social Media.* Conf. on Empirical Methods in Natural Language Processing (EMNLP), 2011, 583-593.

[23] Serban J., L.R., et al., *A Survey of Available Corpora For Building Data-Driven Dialogue Systems: The Journal Version.* Dialogue Discourse, 2018. 9(1): 1-49.

[24] Shaik R., et al., *The Analysis of Data Representation Techniques for Early Prediction of Breast Cancer.* International Journal of Pure and Applied Mathematics, 2017, 1311–8080.

[25] Shang L., Lu Z., and Li H., *Neural Responding Machine for Short-Text Conversation.* Annual Meeting of the Association for Comput. Linguistics (ACL), 2015, 1577-1586.

[26] Sordoni A., et al., *A Neural Network Approach to Context-Sensitive Generation of Conversational Responses.* North American Chapter of the Association for Computational Linguistics (NAACL), 2015. pp. 196-205.

[27] Sukhbaatar S., et al., *End-To-End Memory Networks.* Neural Information Processing Systems (NeurIPS), 2015. pp. 2440-2448.

[28] Sutskever I., Vinyals O., and V. Le Q., *Sequence to Sequence Learning with Neural Networks.* Neural Information Processing Systems (NeurIPS), 2014. pp. 3104-3112.

[29] Szegedy C., et al., *Intriguing Properties of Neural Networks.* International Conference on Learning Representations (ICLR), 2013.

[30] Vinyals O., et al., *Grammar as a Foreign Language.* Neural Information Processing Systems (NeurIPS), 2015. pp. 2773-2781.

[31] Vinyals O. and Le Q., *A Neural Conversational Model.* CoRR abs/1506.05869, 2015.

[32] Vinyals O., et al., *Show and Tell: A Neural Image Caption Generator.* Computer Vision and Pattern Recognition (CVPR), 2015. pp. 3156-3164.

[33] Wen T., et al., *A Network-based End-to-End Trainable Task-oriented Dialogue System.* Conference of the European Chapter of the Association for Computational Linguistics (EACL), 2017. pp. 438-449.

[34] Williams J., R.A. and Henderson M., *The Dialog State Tracking Challenge Series: A Review.* Dialogue Discourse, 2016. 7(3): 4-33.

[35] Yang Z., et al., *XLNET: Generalized Autoregressive Pretraining for Language Understanding.* In Advances in Neural Information Processing Systems, 2019, 5753-5763.