Natural Language Generation for Open Domain Human-Robot Interaction

António Lopes
L2F/INESC-ID, Rua Alves Redol 9, 1000-029 Lisboa, Portugal
IST/Universidade de Lisboa, Av. Rovisco Pais, 1, 1049-001 Lisboa, Portugal
antonio.vilarinho.lopes@tecnico.ulisboa.pt

***Abstract*—In this work, we approach one of the key components of dialogue systems, natural language generation, to study how this component is affected by open domain communication. We rely on statistical methods, namely topic models and deep learning, and approach the traditional generation architecture by optimising the sentence planning and surface realisation, as different tasks. We use documentaries' subtitles to model domain-specific aspects and a large vocabulary dataset to account for domain-independent linguistic concerns. Latent Dirichlet Allocation is used for describing the fine-grained relationships in the domain-specific corpus, while word embeddings (providing geometric semantic relations) are used to represent the large vocabulary. Natural language generation tasks are modelled as deep learning problems. Specifically, sentence planning is implemented with feedforward and convolutional neural networks. Microplanning and surface realisation are implemented with recurrent neural networks, to account for sequential aspects of language. We evaluate our corpus construction method by analysing different time boundaries in the scene detection algorithm and how those parameters affects the topic models. We evaluate sentence planning using cosine similarity and surface realisation with subjective analysis. Our results suggest that the sentence planner can learn a mapping from the generic domain-independent space into the domain-specific space. The quality of surface realisation results must be considered preliminary.**

Keywords: Natural Language Generation, Open Domain, Deep Learning, Recurrent Neural Networks, Topic Models.

I. Introduction

Dialogue Systems (DSs) allow human and machines to interact via verbal, or written, communication. Thus, DSs have been developed to interact with humans to accomplish a certain task witch is often associated with a domain [10], [11], [19], i.e., DSs are, usually, developed to perform a well-defined task on a well-defined domain. However, this implies that these systems are not developed for addressing open domain communication. Therefore, traditional approaches to DS, and its components, focus on accomplish a specific task, covering only the vocabulary of the domain in question, and consequently its semantic interpretation, and not generalising for new domains, in fact, usually, these systems require hand-crafted rules for the specific domain. Nevertheless, to consider an open domain communication the system must be able to generalise to new domains and talk about new concepts.

The Natural Language Generation (NLG) is the component in a DS responsible for determining "how" should the communication goal be presented, this is, this component is responsible for choosing which are the utterances that best represent the system's goal. Moreover, the NLG component is responsible for mapping the system's intentions and goals into natural language. Thus, this module is a key component of the DS as it is the responsible for deciding what should be presented to the end user and has a very important role when considering an open domain communication.

Usual approaches to generation approach the problem in a domain-specific context, where hand-crafted rules must be defined to the domain in question. These approaches usually are known as template-based approaches [30], [31], where the method is filling in empty slots from lexicalised sentences. Although this method is robust and can be applied to different domains, it lacks the fluency, flexibility, and naturalness required for an interaction: the templates are very repetitive. Therefore, these approaches do not provide the required flexibility to approach generation in open domain.

To address some of the limitations previously mentioned, the trainable generation approach proposes using statistical methods in the generation problem, for instance HALLOGEN [16]. Thus, this approach regards the modules within the generation architecture as "trainable", i.e., the modules can be trained from data, for instance to adapt to different domains [32]. However, these approaches still rely on a number of different hand-crafted rules which are then used for optimisation. More recently, data-driven approaches [22], [33], [34] have modelled the generation process by learning directly from data, while using an overgeneration and reranking approach. These approaches propose learning directly from data due to the flexibility using statistical data-driven methods provide, as well as removing the handcrafted rules.

We study how this can be accomplished in an open domain set. Furthermore, we study statistical approaches to generation by approaching both sentence planning and surface realisation in a statistical manner, using deep learning in both modules. Therefore, we approach the conventional approach [24] in a statistical way to reduce the domain-dependent hand-crafted rules.

The document structure is as follows: in section II we provide the background in deep learning and topic modelling and the state of the art in NLG. In section VII we describe our approach and the experimental setup. Next, in section VIII, we present and discuss the results of all experiments. Finally, in section IX we present our conclusions and directions for

future work.

II. Related Word

Regardless of the approach, the process of language generation is often viewed as goal-driven communication [24]. Consequently, a *communicative goal* or *communicative intention* is attempted to be satisfied in order to produce an utterance. This communicative goal or intention is abstract, its interpretation is not fixed and is often to inform the listener, request or persuade the listener to do something or obtain information from the listener. To satisfy those goals, a *communicative act* is decided and performed, so that the listener understands the speaker's intention. The conventional approach [24] divides the problem into two different tasks: sentence planning and surface realisation. The first is responsible for mapping semantic symbols into an intermediary representation for the utterance, while the second module is responsible for given the intermediary representation decide which words best fit the intended utterance. Both modules have been studied before using statistical methods, whether individually or by jointly optimising both modules.

For the first module, sentence planning, the approaches rely on statistical methods, such as Reinforcement Learning (RL) [25], yet still rely on hand-crafted rules for the domain. For the second module, approaches are either template based or statistical, for instance using Hierarchical Reinforcement Learning (HRL) with a Bayesian Network (BN) for surface realisation [9], yet also still rely on prior knowledge and rules.

Data-driven approaches promise to learn the generation process from data without handcrafted rules. Therefore, Oh and Rudnicky [22] were one of the first to propose this method using an n-gram language model, while Angeli et al. [1] proposed using an hybrid approach between template and statistical by learning which template best realises the system's goal. More recently, approaches using deep learning have been proposed [17], [26], [33], [34], as this type of learning allows building powerful generators without relying on hand-crafted rules by learning directly from data. Moreover, approaches using deep learning rely on the power of neural networks to map arbitrary spaces and the power of Recurrent Neural Network (RNN) to model arbitrarily long sequences. Wen et al. [33], [34] uses the paradigm of overgeneration and reranking using Convolutional Neural Networks (CNNs) and RNNs to to validate the semantic consistency of candidates during re-ranking, where the final response derives from re-ranking a set of candidates created by a stochastic generator, and using Long-Short Term Memorys (LSTMs) by jointly optimising the sentence planning and surface realisation in a semantically conditioned unit. Serban et al. [26] also propose learning a variational latent model in an hierarchical way by learning the next utterance in a dialogue. Li et al. [17] use generative models instead of discriminative models and proposes conditioning the generation problem with a topic model, first by using a Markov topic model, and last proposing a variational latent model, as Serban et al. already proposed [26], that learns the context directly from data.

A. Recurrent Neural Network

An RNN is a type of neural network that has the ability to model and learn arbitrarily long sequences of data, this is, instead of considering a simple data point and predicting its output (like a regular Deep Neural Network (DNN)), this type of networks can model the sequential dependencies between inputs. These dependencies are often associated with a temporal dependency as the previous data point influences directly the current. In contrast with feedforward neural networks, which predict each word without the context, an RNN has the ability to model the sequence by maintaining the context in a persistent state, instead of always forgetting. Moreover, this persistence can be regarded as applying the same transformation at each step of the sequence, while considering the previous transformation.

Let us define the input sequence as $x = (x_1, \cdots, x_T)$, the hidden state vector as $h = (h_1, \cdots, h_T)$, and the predicted output vector as $y = (y_1, \cdots, y_T)$, where $t = 1$ to T. The RNN is defined by the following equations:

$$h_t = \sigma(W_h \cdot x_t + U_h \cdot h_{t-1} + b_h) \tag{1}$$

$$y_t = W_y \cdot x_t + U_y \cdot h_{t-1} + b_y \tag{2}$$

where h_t is the network's hidden state, x_t is the input at step t, y_t is the output at step t, σ to a non-linear activation function and $W_{h,y}$, $U_{h,y}$ and b are weight matrices and biases, corresponding to the network's parameters.

Although in theory RNN can learn arbitrarily long sequences, in practice these networks have shown difficulties learning dependencies that have a very large span of information [4]. Another problem with these networks is how to train efficiently, as training with traditional Back Propagation Through Time (BPTT) algorithm has been proved to be extremely difficult due to the exploding and vanishing gradient problems [4]. Thus, to address this problem two different RNN-based methods were proposed, LSTM and Gated Recurrent Unit (GRU).

1) Long-Short Term Memory: To address the limitations of vanilla RNNs, the LSTM was proposed by [14]. In contrast with RNNs, which possess cell structure consisting mainly of a single neural network, LSTMs' cell structure is more complex, as it uses different gates to control how the internal morphing of the information is processed. Thus, these gates interact with each other to control how the information is morphed in the cell's internal state, this is accomplished via "selective writing, reading, and forgetting" [23].

Therefore, an LSTM is, usually, defined as follows:

$$i_t = \sigma(W_i \cdot x_t + U_i \cdot h_{t-1} + b_i) \tag{3}$$

$$f_t = \sigma(W_f \cdot x_t + U_f \cdot h_{t-1} + b_f) \tag{4}$$

$$\tilde{C_t} = \tanh(W_c \cdot x_t + U_c \cdot h_{t-1} + b_c) \tag{5}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C_t} \tag{6}$$

$$o_t = \sigma(W_o \cdot x_t + U_o \cdot h_{t-1} + b_o) \tag{7}$$

$$h_t = o_t * \tanh(C_t) \tag{8}$$

where σ is an activation function (usually the sigmoid function, as it is differentiable and produces continues values between 0 and 1), x_t is the input at step t, h_{t-1} is the previous hidden state, C_{t-1} is the previous cell state, f_t is the forget gate, i_t is the input (write) gate, $\tilde{C}_t$ is the candidate state value, C_t is the current cell state, o_t is the output (read) gate, h_t is the LSTM current hidden state, and $W_{i,f,c,o}$, $U_{i,f,c,o}$ and b are weight matrices and bias vectors, respectively. Although there are other variants of LSTM, we only consider this one as it is the most used one. More variants can be found in [13].

2) Gated Recurrent Unit: The GRU was proposed in [7] as a variation of the LSTM, where instead of coordinating the writes and forgets, the GRU links them explicitly into one gate, called the update gate (which acts as a "do-not-update"). Furthermore, the GRU replaces the "selective writes" and "selective forgets" by a single "selective overwrites", this is accomplished by setting the forget gate to 1 minus the input (write) gate (which is in fact specifying how much of the previous state the cell should not overwrite).

The fundamental equations of the GRU are the following:

$$z_t = \sigma(W_z \cdot x_t + U_z \cdot h_{t-1} + b_z) \tag{9}$$

$$r_t = \sigma(W_r \cdot x_t + U_r \cdot h_{t-1} + b_r) \tag{10}$$

$$\tilde{h_t} = \tanh(W \cdot (r_t * h_{t-1}) + U \cdot x_t + b) \tag{11}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h_t} \tag{12}$$

where x_t is input at step t, h_{t-1} is the previous hidden state, z_t is the update gate, r_t is the reset gate, $\tilde{h_t}$ is the shadow gate, h_t is the current hidden state, and $W, W_{z,r}$, $U, U_{z,r}$, and b are the weight matrices and bias vectors respectively. Note that the original formulation defined r_t as the reset gate, however this gate functions more as a read gate.

III. Topic Models

Topic models are unsupervised models that aim to model words relationships inside a collection and group them by their corresponding topic, i.e., topic modelling, usually, does not require any prior annotations and aims at given a collection of documents find the group of words that best represent the collection, find the topics which best describe the collection. Moreover, the number of topics is, usually, one of the parameters these algorithms require to be determined manually. These algorithms enable discovering topic patterns in the collection as they are based on the assumption that documents are mixtures of topics, where each topic is a probability distribution over the words [28].

Furthermore, topic models are, usually, generative and only consider the number of times the word is seen in a document (although there are extensions which preserve word ordering). Statistical topic models require the number of topics to be defined manually and specify a multinomial distribution over words for topics and over topics for documents. Topic models do not make any assumption regarding the meaning of the words in a document, instead they make the bag-of-words assumption and aim at fitting the word in a topic.

A. Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is a well known topic model that assumes that each document in a collection is made of multiple topics, where each topic is a distribution over the a fixed vocabulary of terms [28]. Moreover, the model assumes there are K topics associated with the collection and each document is composed by different percentages of each topic. As a probabilistic topic model, LDA represent their hidden variables via topics, where the documents' hidden variables are representative of the subjects of the collection.

The iterative process for document generation is as follows, for each document d in the collection:

(1) Choose θ_i, modelled by a Dirichlet distribution with the parameter α (Dir(α)).
(2) For each word n in document d:
   (a) Retrieve topic z_N form the multinomial distribution Multinomial(θ_d);
   (b) Given topic z_n, retrieve a word from the multinomial distribution Multinomial(φ), the probability over the vocabulary, where φ is modelled through a Dirichlet distribution with the parameter β (Dir(β)).

IV. Dataset Requirements

In order to study open domain generation, especially using a data-driven method, our dataset must meet the following requirements: coherent discourse structure, domain-dependent discourse, and a large domain-independent vocabulary. Therefore, to meet these requirements we have two different corpus in our dataset. The first corpus, documentaries subtitles, provides the first and second requirement, while the second corpus, Google n-gram [6], provides the last requirement. Furthermore, our documentaries subtitles focus on a specific domain, physics, and their nature provide a domain-dependent discourse, as the subtitles address the domain in question, thus having a more domain dependent and limited vocabulary. In addition, the subtitles scenes also provide a coherent discourse structure, a narrator is describing the different aspects of the documentary. Finally, the Google n-gram corpus provides a large domain-independent vocabulary. To be able to generate utterances in open domain we require a large vocabulary, as the vocabulary addresses most of the language.

In this section, we describe our large domain-independent vocabulary and representation, using word representations. We describe our corpora and our approach to build a corpus from subtitles. Our original corpus is an extension of the documentaries subtitles set described by [2]. The corpus was gathered for abstract summarisation tasks and was also used for topic modelling, which is a relevant part of this work (even if it is not the focus). Moreover, this corpus consists of 265 subtitles from documentaries of the physics domain, where almost all the documentaries are monologues and the narrator presents the different aspects of the documentaries' subject. Therefore, this corpus is, potentially, adequate for the focus of this thesis: on the one hand, the corpus can be used for to topic modelling and, thus, is suitable for the domain part of the work, and, on the other hand, it enables the narrator/explainer to demonstrate the purpose of this work.

Furthermore, documentaries provide domain-dependent structure and coherent discourse implicitly in the scenes that compose the documentary. Therefore, we consider that each scene can be regarded as an individual document, as each scene not only compresses enough information regarding the subject of the documentary and can be more detailed about a particular context of the documentary's subject, but also, provides a coherent domain-dependent discourse structuring: each scene is coherent with respect to itself and to the documentary's subject. To achieve this, we propose an approach for detecting scenes in subtitles and extracting them from the original documentary.

A. Large Vocabulary Corpus

Having a large vocabulary is a very important requirement to address generation in an open domain context, as the vocabulary is not limited and covers most of the words in the language. This way, when generating an utterance, it is possible to cover, potentially, most of the words from any specific domain.

Thus, we use the Google's ngram dataset [6] as our domain independent corpus, as its vocabulary covers must of the English language. Moreover, how we represent the words is also a relevant aspect of the large vocabulary, as we want to have a generic space where words preserve the relations between one another. Thus, we represent the words using word2vec [20], i.e., we transform the discrete word space into a continuous dense space that preserves words relations.

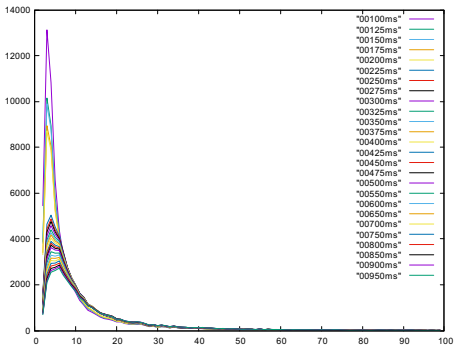
We train word embeddings using the Skip-gram model with negative sampling and use Google's ngram dataset [6] to train the word2vec model [12], where the size of an embedding is a 200 dimensional vector, the text is normalised using the default normalisation provided by the word2vec toolkit, and we use a window of 5 context words for the Skip-gram.

B. Tuning the Corpus

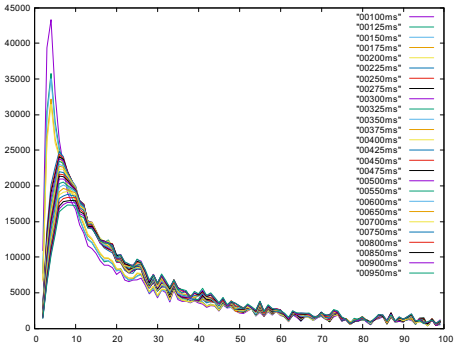
Our approach uses a parameter, in ms, to determine the boundary of the scene, i.e., we exploit the temporal nature of a documentary to detect the boundary between scenes. We perform this by collapsing alignment items whose distance is smaller than the parameter and stop when we can not perform any more collapsing. After the scene detection, we extract each scene using parameters to regulate the minimum and maximum durations of scenes, as well minimum number of words.

With the variation of the time parameter to detect scene's boundaries, we expect that for small values less scenes are grouped, which leads to having a greater number of small scenes (figure 1 (a)). However, there is a pattern in the frequency: from $100ms$ to $200ms$ the gap is not stable and the number of small scenes tend to oscillate from one value to the next, then from $225ms$ until $950ms$ the decay is smoother, which leads to conclude that the gap stabilises and the decay of frequencies is just a natural consequence of the gradual increment of less distant scenes (more scene collapse).

Furthermore, although the frequency analysis leads us to reason that this parameter should be at least $200ms$, we look



(a) Frequencies



(b) Mass of Frequencies

Fig. 1: Corpus Analysis. Figure (a) represents the frequencies and figure (b) represents the mass of frequencies.

at the mass of the curve (the integral) to further understand how the parameters influence the boundary detection. In figure 1 (b), not only, the range from $100ms$ to $200ms$ presents the same result, very distant from the other intervals, but also, the mass of the curve concentrates in shorter scenes and decays very fast. In contrast with this range, the $225ms$ to $950ms$ range proves to concentrate the mass in a very similar way, concentrating with short to medium size scenes, while decaying smoother than the previous range.

C. Topic Modelling

Using a topic model provides a natural method for grouping concepts in a collection. This way, we can model the fine-grained relations of our domain-dependent corpus. From the different topic models we choose LDA due to its wide application in the literature and the previous application of this model in the subtitles domain [2]. Furthermore, we use the LDA implementation provided by [5].

To perform an LDA estimation over our collection we need to create the vocabulary of the collection, ignoring stop words, and, then, create for each document of the collection the bag-of-words of the document's (scene) vocabulary. After creating the vocabulary, we estimate the model using a random topic initialisation, an α of 0.3, and vary the number of topics the model should estimate. We perform the same conditions for 50, 100, and 200 topic models.

A good topic model is essential for the domain-dependent part of this work. Therefore, we study how the corpus con-

Scenes Gap (ms)	Vocabulary Size	Number of scenes
100	9574	62035
100 *	9052	61635
200	9635	52654
200 *	8993	52597
300	9660	44326
300 *	8987	44058
400	9655	42571
400 *	8984	42107
500	9658	40832
500 *	8965	40357
600	9660	39510
600 *	8945	39025
700	9660	38389
700 *	8938	37895
800	9661	37178
800 *	8919	36665
900	9662	36087
900 *	8896	35563

TABLE I: Merged Scenes Statistics (* cutoff of $100000ms$).

Gap (ms)	50 topics	100 topics	200 topics
100	2	2	2
100 *	2	2	2
200	4	2	2
200 *	3	2	2
300	12	3	**4**
300 *	5	4	2
400	13	3	3
400 *	9	2	2
500	10	3	2
500 *	**14**	2	3
600	10	3	2
600 *	5	2	2
700	12	**6**	2
700 *	12	2	3
800	13	4	2
800 *	10	2	2
900	**14**	**6**	3
900 *	13	3	2

TABLE II: Word topic co-occurrence (* cutoff of $100000ms$).

struction affects the LDA models. In table I we depict how the vocabulary size and number of scenes differ when the boundary is incrementally crisper. Moreover, the number of scenes decreases with the increment of the merging parameters, even more accentuated when the $100s$ cutoff is applied, this is due to the boundaries being incrementally crisper, which leads to shorter scenes being merged more frequently. Furthermore, the vocabulary size oscillates, which can be explained by the extracting parameters: softer boundaries will have shorter scenes that when extract may be discarded; while crisper boundaries will have short to medium scenes preserving more vocabulary. In addition, when the cutoff is applied the number of scenes discarded increases but not significantly, in contrast with the vocabulary which decreases significantly when the cutoff is applied.

Furthermore, we evaluate how many overlapping words there are between topics, so as to understand whether the topic model is performing a softer or crisp boundary between topics. In table II we present max number of word co-occurrence in different topic models. We we can conclude that each of the 100 and 200 topic models have a crisp boundary, as the maximum co-occurrence is 6 and 4, respectively. However for 50 topics models the maximum word topic co-occurrence is 14, which is a significant co-occurrence and allows us to understand that these models are not adequate for our problem. Although each model can be used for further experiences with sentence planning and surface realisation, we only consider the $300ms$ gap merged scenes with 100 topics, as its behaviour is regular both in the corpus analysis and in the LDA topic model analysis.

V. Sentence Planning

Sentence planning is, usually, responsible for determining the content and structure of the response. We approach the content determination problem and structure of the response in different modules, not exclusive to the sentence planner, i.e., we use the sentence planner to determine the content of the response and, implicitly, the structure. However, we explicitly determine the real structure in a micro-content planner which is described in section VI.

Gap (ms)	Topic 1	Topic 2	Topic 3
100	galaxy	world	result
	milky	science	detect
	astronauts	true	nebula
	changed	lead	suddenly
	group	decades	spinning
	starting	range	died
	andromeda	carry	occurred
	proposed	straight	fine
	powered	fiction	dying
	copernicus	war	device
100 *	universe	kind	process
	place	complex	days
	call	molecules	order
	beginning	university	crew
	existence	chemistry	weather
	birth	organic	absolutely
	expanding	nucleus	difference
	giving	failed	changing
	creation	grand	desert
	imagined	europe	engine
200	measure	looked	water
	camera	control	scale
	minute	top	volcanoes
	path	radio	active
	straight	room	volcanic
	familiar	send	lava
	cut	hear	craters
	tons	bottom	molten
	trees	signal	volcano
	lower	signals	flows
200 *	asteroid	spacecraft	fact
	happened	rocket	created
	event	seconds	interesting
	land	launch	direction
	dinosaurs	shuttle	gave
	named	active	possibility
	brought	program	satellite
	including	engine	mountain
	belt	meant	paper
	extinction	color	fraction

TABLE III: Top 10 words of Latent Dirichlet Allocation 100 topic models (first five topics) for different gaps (* cutoff of $100000ms$).

We approach sentence planning using a statistical way, by learning a mapping from a generic word embeddings (synthesised questions) into a topic space, this can be regarded as the planning the context of the utterance in a lower dimensionality space. Moreover, our approach is determining the domain content by refining from a question, in a large vocabulary, into a topic specific space.

We achieve this by using word2vec [20] for the embedding space (question) and LDA for the topic space (content determination). This way, the sentence planner is responsible for refining the generic domain into a more specific domain, in our case a physics domain (see section IV). Figure 2 depicts the sentence planner, while Figure 3 depicts the topic model.

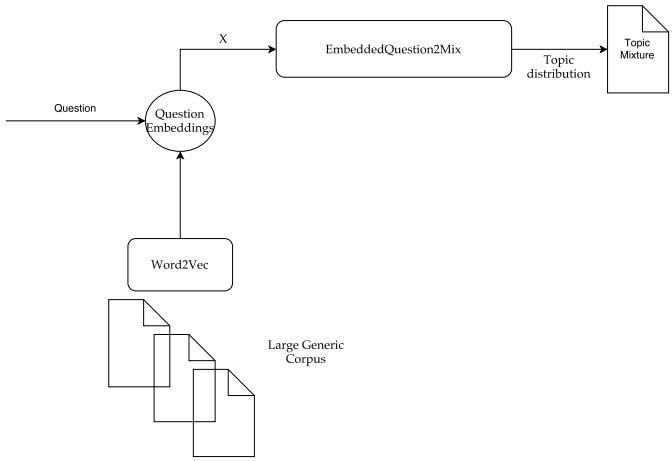


Fig. 2: Sentence Planner architecture.

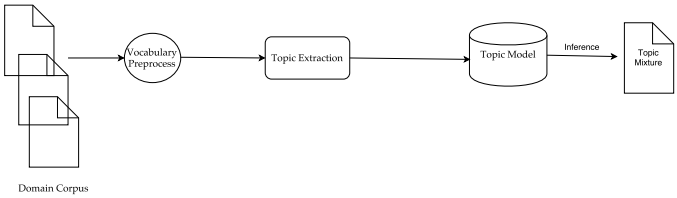


Fig. 3: Topic model. Domain-dependent part of the sentence planning.

Moreover, the sentence planner is a generic mapping from word representations in a generic space into topic distributions, i.e., the sentence planner learns how words should be encoded into a topic distribution (analogously to what a topic model does). Therefore, our focus on the sentence planner is how to map generic word representations into a topic distribution, i.e., how to perform a mapping from an already dense space which preserves relationships between words into an even denser space which models the distribution of words in a topic space. Thus, determining the content of the response in the domain space.

In Figure 2 the question is transformed from the discrete space of words into the domain-independent space, which is achieved by transforming each word using word2vec trained over a large generic corpus. Next, the embedded question models, implicitly, the communicative goal and the EmbeddedQuestion2Mix is our sentence planner, which maps from the question space (word embeddings space) into the topic, domain-specific, space.

VI. Microplanning and Surface Realisation

In this section, we describe our statistical approach to microplanning and surface realisation. The micro-content planner is, usually, responsible for refining the response structure of the sentence planner, while the surface realisation is responsible for given the abstract representation from the previous steps, transform that representation into words. Thus, the realisation is responsible for deciding which lexical items should be chosen to map from the abstract representation to the words.

In our approach, the micro planner and surface realisation modules are responsible for determining the structure of the response and which words best suit the response, respectively. Furthermore, the planner must be able to determine the structure of the response from the response content, determined by the sentence planner. This can be regarded as the micro planner unfolding the response content, global content, into a refined sequence of sentence representations that satisfy the global response content, via a sequence of local content representations. In turn, the micro planner determines a sequence of representations of sentences which are structure of the explanation.

We use two methods for microplanning and surface realisation, which perform the mapping at document and word level: jointly optimising the process; and dividing the problem into two sub-problems. This way, these approaches learn how to map the context into a sequence of sentences which in turn maps into a sequence of words. To address the inter-sentence relations of a document our approach learns the mapping between the topic distribution and the sequence of local topic distributions that defines the structure of the response and the mapping from this structure into a sequence of words. This way, not only, learning the structure of the response by determining which topic distributions best represent the sentences that should be produced, but also, learning which words best suit the topic distributions. The first approach learns how to map the context into a sequence of words, while the second approach learns how to map the context into a sequence of sentences which in turn maps into a sequence of words.

To address the inter-sentence relations of a document we learn the mapping between the topic distribution the sentence planner should produce and model, not only, learning the structure of the response, by determining which topic distributions best represent the sentences that should be produced, but also, learning which words best suit the topic distributions. Therefore, we use two approaches: jointly optimise both document and sentence structure using a hierarchical approach and split the problem into two parts, first, optimising the content planning, by learning a sequence of topic distributions, and, second, we optimise the words in each of the previous sequence.

A. Microplanning and Realisation

In this approach, we divide the realisation into two steps: first, decide the structure of the final answer, addressing the

inter-sentence relations; and, second, given the structure, i.e., the content, perform the realisation into a linguistic structure.

Figure 4 depicts the first step, while Figure 5 depicts the second step. Dividing the problem into two steps has as its main advantage allowing to scrutinise the result of the content planning before performing the word realisation. This way, we can understand if the inter-sentence relations are addressing the content selection as they should and only then perform the word realisation.

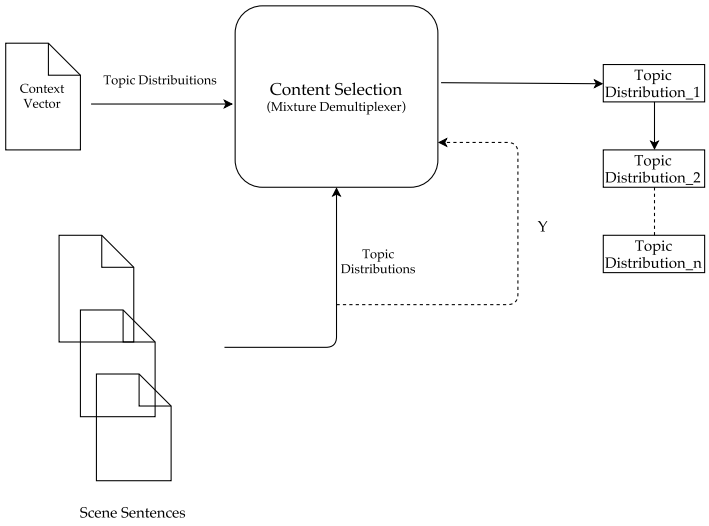


Fig. 4: Content planning. Determining the structure of the explanation (Mixture Demultiplexer).

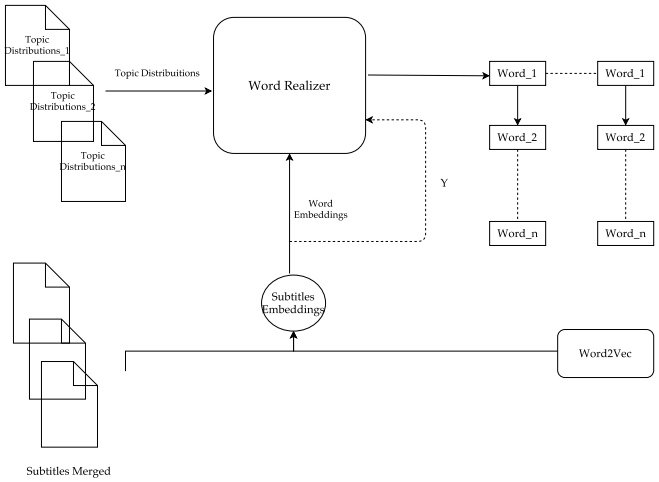


Fig. 5: Word realisation. Realise the non linguistic content with linguistic content.

B. Hierarchical Naive Approach

This approach models the microplanning and realisation problem joint problem by jointly learning the mapping between the topic distribution and the response structure, as well as the mapping from the response structure into the words. Thus, the hierarchical naive approach models the microplanner and realisation by determining which sentences should be in the response and which words should be realised in each sentence. Moreover, this approach is the approach in Figures 4 and 5, where both steps are performed jointly, conditioned by the sentence representation.

C. Naive Surface Realisation

We also use a naive approach which does not perform microplanning and tries to decode on word level, not taking into consideration the document structure. Therefore, the model learns how to generate arbitrarily long sentences by learning how to map from the topic mixture into a sequence of words. This approach main advantage is its simplicity, approaching the realisation in a naive way by simply optimising the sequence of words with respect to a context vector. However, this approach lacks the ability to model the document structure and the inter-sentence relations, which is relevant for generating better utterances. Furthermore, this approach tries to learn arbitrarily long sequences, which can lead to worse results, as there are practical limitations in the statistical learning approach. The surface realiser module for the naive approach is depicted in Figure 6.

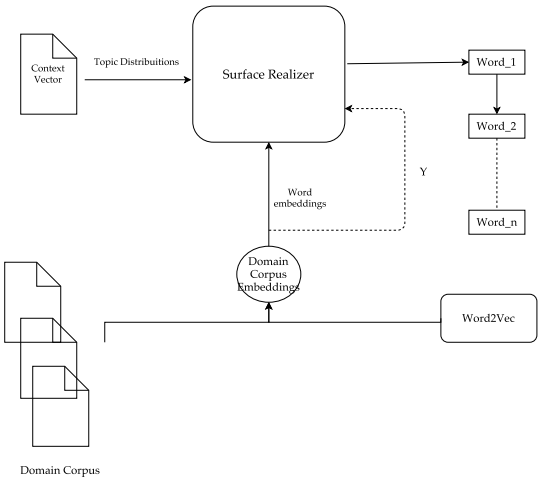


Fig. 6: Surface Realiser.

VII. Experimental Setup

In this section we describe our approach to the generation problem, as well as the experimental setup for all the experiments.

A. Sentence Planning

DNNs can learn arbitrary mappings between two different spaces [8]. Thus, they fit perfectly in this problem, where we want to map from a generic word embedding space into the topic latent space. Thus, we approach the sentence planning using deep learning, namely using feedforward DNNs and CNNs, to learn a one-to-one mapping, more concretely, to perform the mapping between the generic space into the topic space. In addition, we use word2vec, for the generic space, and LDA for the topic space. We evaluate three different approaches to perform the mapping from the word embeddings

(question) into the topic distribution (content determination): a deep feedforward network and two convolutional networks (both 1D ConvNets).

Our approach to sentence planning relies on both topic modelling and deep learning, the first one was already described in III, while the second consists of using DNN to mimic the behaviour of a topic model by mapping the generic space to the topic space. Thus, we conducted two experiments, which are influenced directly by the LDA results. First, we divide the document collection into train and test and perform the model estimate over the training set exclusively, which is then used on the sentence planner as the training set (10% of the training set is used as validation), and evaluate with respect to the LDA inference, comparing this way previously unseen document for both models. Second, we use LDA to infer the topic distributions of the previous estimate model and train the planner with the inferred topic distributions (10% of the training set is used as validation), i.e., instead of using the internal LDA model directly we train with an approximation given by the inference, which is closer to what the planner should predict. Furthermore, we evaluate using the test set which both LDA and the network have never seen. The collection division into train and test is performed randomly.

To study how well the planner maps the embeddings space to the topic space, we use three networks: a feedforward DNN and two CNN. The feedforward DNN and one of the CNN map from a sentence representation (sum of the embeddings of all words) into the topic space, while the other CNN looks at the bag-of-words of the synthesised question. In addition, we study how do the hyper-parameters affect the networks' performance.

Finally, all models were trained with minibatching and Adaptive Moment Estimation (ADAM) [15] as the optimiser. To prevent overfit we apply the dropout technique [27]. All models run on a GeForce GTX TITAN X [21].

B. Microplanning and Surface Realisation

Our micro-planner is responsible for determining the structure of the response explicitly, while modelling fine-grained relationships by learning a mapping from the document topic distribution into its sentences topic distributions: the planner unrolls a topic mixture into a sequence of topic mixtures. Surface realisation determines linguistic content from non-linguistic content, as such we propose an approach which relies on deep learning, with emphasis on recurrent neural networks, to decode the representation of the sentence planner into words.

We approach the problem as a sequence-to-sequence problem, namely a one-to-many and a one-to-many-to-many problem – naive realisation and micro content and realisation, respectively. The reason to address the problem as formulated is to follow a purely statistical approach, namely deep learning, which provides the ability to model arbitrarily long sequences (RNN) while providing a method to increase the flexibility and, arguably, naturalness of the generated utterance. The decoders considered in this work are similar to the one proposed in [3], where the embedding of target word $e(y_t)$ at step t is a peek of the output, and the additional weight matrices $C, C_{r,z}$ compute the context vector c_t at each step:

$$z_t = \sigma(W_z \cdot e(y_t) + U_z \cdot h_{t-1} + C_z \cdot c_t + b_z) \quad (13)$$

$$r_t = \sigma(W_r \cdot e(y_t) + U_r \cdot h_{t-1} + C_r \cdot c_t + b_r) \quad (14)$$

$$\tilde{h_t} = \tanh(W \cdot e(y_t) + U \cdot (r_t * h_{t-1}) + C \cdot c_t + b) \quad (15)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h_t} \quad (16)$$

Furthermore, the hierarchical naive approach is similar to the one used by [18], with one decoder decoding on document level (sentences) and another decoder conditioned by the first one decoding on sentence level (words), using a "static" attention mechanism [35]. Therefore, this decoder operates on inter-sentence and intra-sentence level, preserving the document (response) structure and the sentence structure of each sentence. The micro planning and realisation divide this approach into two different steps, the first models a one-to-many sequence, unrolling a topic distribution into its composing topic distributions, while the second is for each topic distribution realise a sequence of words.

Both the naive realisation and microplanning and realisation models were develop using Theano [29] and the framework developed during this thesis. Both use the same corpus and the same normalisation (the same normalisation used for the word embeddings). In addition, both models run on a GeForce GTX TITAN X [21].

VIII. Experimental Results and Discussion

In this section we present the results from all the experiments, namely the statistical approaches to sentence planning and surface realisation.

A. Sentence Planning

We evaluate our sentence planner with respect to the LDA model, as the planner is responsible for determining the context of the utterance to be generated, this context is provided by the topic model. We evaluate how well the planner learns the topic model, estimate and inference. In Figure 7, we depict a comparison between the best approaches for the first scenario, where the networks learn the model estimate. In addition, in Figure 8, we depict a comparison between the best approaches for the second scenario, where the networks learn the inference directly. Finally, in figure 9 we depict a comparison between the best approach from the first scenario and the worst from the second, to show that learning from the inference yields better results than from the estimation.

From figures 7 and 8, we can conclude that the sentence planner learns to map embeddings to both LDA estimate and inference model. Moreover, learning to map from the embeddings space to the topic space does not yield worse results than performing directly the LDA inference. This way, the method we use, not only, is more flexible, as the word embeddings are a continuous dense space and the vocabulary in the LDA model is a discrete word space, but also, provides a way for mapping a generic embedding space into a lower dimension topic space. Finally, when learning the inference

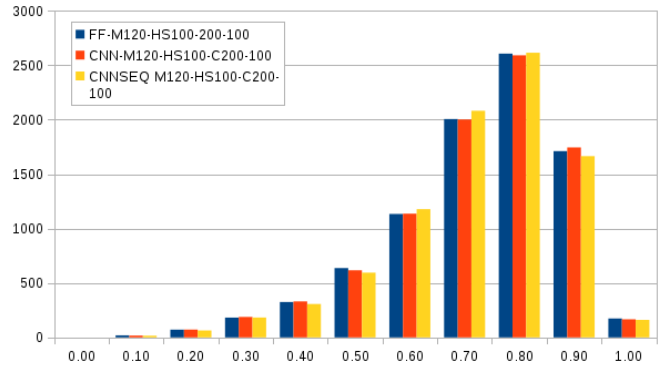


Fig. 7: Sentence Planner vs LDA Inference, cosine similarity frequency. Best networks results for estimate learning.

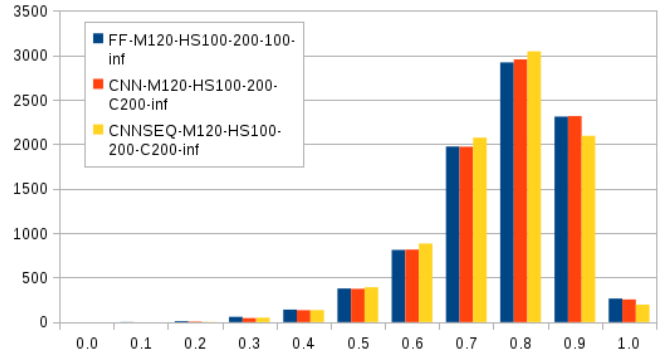


Fig. 8: Sentence Planner vs LDA Inference, cosine similarity frequency. Best networks results for inference learning.

directly, instead of the estimate, the networks, as expected, have a better performance, as the network learns to map from the embeddings directly into the inference space. Furthermore, the better performance of inference training is depicted in figure 9, where the worst approach using the inference training is better than the best using estimate training.

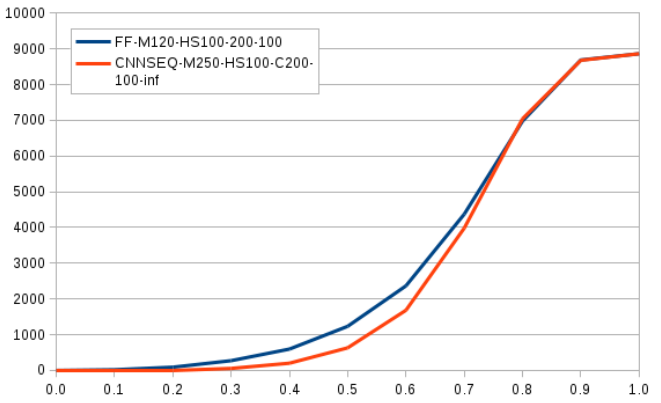


Fig. 9: Comparison between the best estimation and worst inference learning.

B. Surface Realiser

Evaluating a natural language generator is a hard task, as different utterances can encode the same meaning. In fact, the community is often divided when the subject is evaluating a generation system, as the existent objective measures only take into consideration the frequency of ngrams in the generated utterance with respect to the expected utterance, the same applies for fields such as machine translation. Thus, subjective measures where independent judges classify the system's response are one of the methods to evaluate the system's performance, for instance considering the fluency and naturalness of the generated utterance.

The results from the surface realisation are preliminary and require further study to conclude whether the approach is feasible, thus they are not depicted here. Moreover, the micro-planner could not map the topic distribution to a sequence of topic distributions properly. Furthermore, the network's behaviour was predicting identical sequence of topic mixtures for different initial mixtures. This can be explained by the loss function not being the most appropriated one, the architecture lacking the deep nature required for this type of applications, among others.

All the experiments performed suffer from the same defect, which could be improved: the number of layers in the architecture is not sufficient to model from the domain-specific content into the word generic representation. This requires further study by studying deeper architectures.

While conducting the experiments, we faced an insuperable obstacle: the Graphics Processing Unit (GPU) memory limitations. The desired approach would be performing a softmax, or an approximation, at the end of the network and use as loss function such as categorical cross entropy. However, due to the vocabulary size and the GPU memory limitations, we used an approximation by using as loss function the mean squared error for the flat model and hierarchical model, predicting directly the embeddings, and the cosine distance (and mean squared error) for the content planning.

C. Discussion

Approaching the generation problem as a statistical optimisation task provides a flexible method for generating utterances. Furthermore, to address the generation process using statistical methods, both components of the pipeline conventional architecture must be able to generalise to different concepts and domains. We study how conditioning generation with a domain-dependent aspect performs, as scaling to different domains is feasible.

The main task of the sentence planner is to determine the response content and the structure of the response. However, the explicit structure of the response is addressed by the micro-content planner. Therefore, our approach to sentence planning has as its main task synthesising the communicative objective and determining the content of the response in a domain-specific space, by mapping a (synthesised) question from a domain-independent question into a topic distribution (domain-dependent), i.e., we approach the planning as determining the domain content by mapping from a question

in a embedding space into a response in the topic model space. Therefore, the sentence planner maps from a generic embedding space into a more fine-grained domain space by mapping word representations into topic distributions. We showed that performing this mapping does not yield significant worse results than using the LDA model. Thus, the main advantage of performing the inference via DNN instead of LDA is the mapping from the generic word representations into a more fined-grained representation which the LDA can not provide as it is constrained on the collection's vocabulary.

The results of the surface realisation are still preliminary and are far from the ones we expected. All the experiments performed suffer from the same defect, which could be improved: the number of layers in the architecture is not sufficient to model from the domain-specific content into the word generic representation. This requires further study by increasing the number of layers.

There are different limitations in our work: for the sentence planning, the best scenario was not addressed, partitioning the dataset into three parts so that the networks are trained with inference of never seen documents exclusively and not and approximation; for the surface realisation we faced a constraint in the GPU memory. For the first, using an approximation proved to be feasible, while for the second instead of optimising a cross entropy, we had to approximate and learn the embeddings directly.

IX. Conclusions

We approached the generation problem in a statistical way to study the problem under an open domain environment, in the context of a narrator. Furthermore, we address the generation problem by considering both sentence planning and surface realisation a statistical problem.

We propose a method for creating a corpus based on documentaries subtitles which exploits the temporal nature of subtitles, while preserving the original time alignment. The reason to extract scenes is that we need a domain-dependent part to refine a domain-independent vocabulary into a domain-dependent vocabulary, as the scenes from subtitles provide a domain-dependent discourse structure. These scenes also provide a coherent discourse structure by definition, as they are coherent with respect to themselves and the documentary.

This method was further tested with topic models and showed that the models estimated had a crisp boundary as the word co-occurrence was not significant, specially for 100 and 200 topic models.

We approach sentence planning as a mapping from a generic word space into a topic space by learning the mapping between a (synthesised) question in a embedding space into a topic model inference space. This way, we showed that the planner can learn how to lower the dimensionality of the word space, by learning the mapping from a dense space (word embeddings) with a large vocabulary into a lower denser space (topic inference) with a small vocabulary.

We approach surface realisation with a naive approach and with microplanning and surface realisation, with the main task of mapping the representation (topic distribution) in the domain-specific space back into the domain-independent space, while conditioning the generation with the domain aspects. Our results are preliminary as all approaches yield results distant from the ones we were expecting. Thus, this approach requires further study.

We plan using deeper architectures for the surface realisation and evaluate if we can improve our results. Furthermore, we plan using a native method for mapping embeddings into topic mixtures, Gaussian Latent Dirichlet Allocation, and study if we can replace our approach to sentence planning with the Gaussian Latent Dirichlet Allocation. Finally, we plan using machine learning techniques, such as scheduled sampling, to improve the microplanning and surface realisation steps and dividing the realisation into two steps: predicting content words and sentence reconstruction.

References

[1] G. Angeli, P. Liang, and D. Klein. A simple domain-independent probabilistic approach to generation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, 9-11 October 2010, MIT Stata Center, Massachusetts, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 502–512, 2010.

[2] M. Aparício, P. Figueiredo, F. Raposo, D. Martins de Matos, R. Ribeiro, and L. Marujo. Summarization of films and documentaries based on subtitles and scripts. *Pattern Recogn. Lett.*, 73(C):7–12, Apr. 2016.

[3] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[4] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *Trans. Neur. Netw.*, 5(2):157–166, Mar. 1994.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[6] T. Brants and A. Franz. Web 1t 5-gram version 1. 2006.

[7] K. Cho, B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014.

[8] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)*, 2(4):303–314, 1989.

[9] N. Dethlefs and H. Cuayáhuitl. Combining hierarchical reinforcement learning and bayesian networks for natural language generation in situated dialogue. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 110–120. Association for Computational Linguistics, 2011.

[10] G. M. Ferguson and J. F. Allen. TRIPS: the rochester interactive planning system. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence and Eleventh Conference on Innovative Applications of Artificial Intelligence, July 18-22, 1999, Orlando, Florida, USA.*, pages 906–907, 1999.

[11] G. M. Ferguson, J. F. Allen, and B. W. Miller. TRAINS-95: towards a mixed-initiative planning assistant. In *Proceedings of the Third International Conference on Artificial Intelligence Planning Systems, Edinburgh, Scotland, May 29-31, 1996*, pages 70–77, 1996.

[12] F. Ginter and J. Kanerva. Fast training of word2vec representations using n-gram corpora, 2014.

[13] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber. LSTM: A search space odyssey. *CoRR*, abs/1503.04069, 2015.

[14] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, Nov. 1997.

[15] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[16] I. Langkilde and K. Knight. Generation that exploits corpus-based statistical knowledge. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL '98, pages 704–710, Stroudsburg, PA, USA, 1998. Association for Computational Linguistics.

[17] J. Li and D. Jurafsky. Neural net models for open-domain discourse coherence. *CoRR*, abs/1606.01545, 2016.

[18] J. Li, M. Luong, and D. Jurafsky. A hierarchical neural autoencoder for paragraphs and documents. *CoRR*, abs/1506.01057, 2015.

[19] D. J. Litman and S. Silliman. ITSPOKE: An intelligent tutoring spoken dialogue system. In *Demonstration Papers at HLT-NAACL 2004*, HLT-NAACL–Demonstrations 2004, pages 5–8, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[20] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[21] Nvidia Corporation. Nvidia GeForce Titan X, 2015. http://www.geforce.com/hardware/desktop-gpus/geforce-gtx-titan-x.

[22] A. Oh and A. I. Rudnicky. Stochastic natural language generation for spoken dialog systems. *Computer Speech & Language*, 16(3-4):387–407, 2002.

[23] R2RT. Written memories: Understanding, deriving and extending the LSTM, 2016. http://r2rt.com/written-memories-understanding-deriving-and-extending-the-lstm.html.

[24] E. Reiter and R. Dale. *Building Natural Language Generation Systems*. Cambridge University Press, New York, NY, USA, 2000.

[25] V. Rieser and O. Lemon. *Reinforcement Learning for Adaptive Dialogue Systems - A Data-driven Methodology for Dialogue Management and Natural Language Generation*. Theory and Applications of Natural Language Processing. Springer, 2011.

[26] I. V. Serban, A. Sordoni, Y. Bengio, A. C. Courville, and J. Pineau. Hierarchical neural network generative models for movie dialogues. *CoRR*, abs/1507.04808, 2015.

[27] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[28] M. Steyvers and T. Griffiths. Probabilistic topic models. *Handbook of latent semantic analysis*, 427(7):424–440, 2007.

[29] Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016.

[30] M. Theune, E. Klabbers, J. R. De Pijper, E. Krahmer, and J. Odijk. From data to speech: A general approach. *Nat. Lang. Eng.*, 7(1):47–86, Mar. 2001.

[31] K. Van Deemter, E. Krahmer, and M. Theune. Real versus template-based natural language generation: A false opposition? *Computational Linguistics*, 31(1):15–24, Mar. 2005.

[32] M. A. Walker, A. Stent, F. Mairesse, and R. Prasad. Individual and domain adaptation in sentence planning for dialogue. *J. Artif. Intell. Res. (JAIR)*, 30:413–456, 2007.

[33] T. Wen, M. Gašić, D. Kim, N. Mrksic, P. Su, D. Vandyke, and S. J. Young. Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking. *CoRR*, abs/1508.01755, 2015.

[34] T. Wen, M. Gašić, N. Mrksic, P. Su, D. Vandyke, and S. J. Young. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1711–1721, 2015.

[35] R. Yan. I, poet: Automatic poetry composition through recurrent neural networks with iterative polishing schema. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 2238–2244. AAAI Press, 2016.