KATHOLIEKE UNIVERSITEIT
LEUVEN

Sequence-to-Sequence Learning for End-to-End Dialogue Systems

Jordy Van Landeghem

Thesis submitted for the degree of
Master of Science in Artificial Intelligence

Thesis supervisor:
Prof. dr. Marie-Francine Moens

Assessors:
dr.Vincent Vandeghinste

Mentors:
dr. Fabrice Nauze
Sergiu Nisioi

Academic year 2016 – 2017

Table of Contents

1. Introduction

The following document reports on the internship project that I completed while part of Oracle's Service Cloud development team working on Virtual Assistant technology. The original goals projected in the job description have shifted towards being more research-oriented.
The initial goals were the following:

- a survey of recent literature within the field of end-to-end dialogue systems
- collection and preprocessing of relevant social interaction data
- evaluation of methods from the reviewed literature with the collected data
  – Suggesting improvements to current dialogue systems
  – Assessing the scalability of selected methods

The first goal is addressed in the first part of this report. In order to restrict coverage of this vast domain, the main focus is on the characterization of Deep Learning models for unsupervised dialogue response generation. About the second goal, although no adequate "chitchat" dataset has been readily found, we resorted to a collecting a dataset within the troubleshooting domain. Arguably, the language used in the troubleshooting domain is very difficult to model and is in nature closest to social chitchat. With respect to the larger third goal, due to time and resources restraints not all reviewed methods have been evaluated on the collected data, it is doubtful this would have been very insightful. Instead, as agreed amongst the collaborators, we would take a step back from more recently introduced, more complex models to probe the generation capacity of the core Deep Learning (where appropriate abbreviated as "DL") algorithm underlying the recent surge in academic and industrial interest. The main idea of the project is to get a deeper understanding of the base algorithm, encoder-decoder – sequence to sequence, henceforth Seq2Seq – architecture, by inspecting its mechanisms on an original dataset, and to identify dangerous pitfalls and promising directions to take for the future of this ever-increasingly popular and propitious technology.

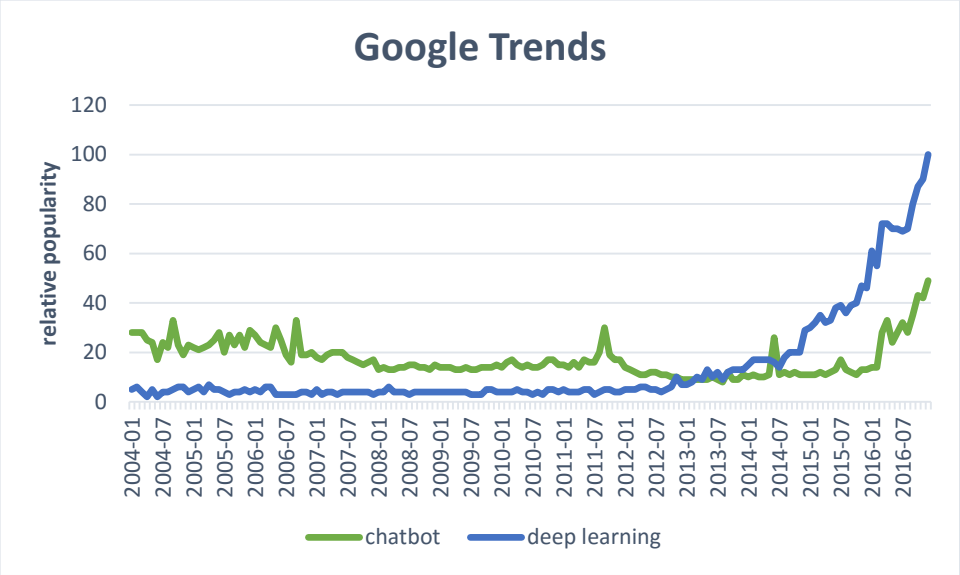
The research in building and improving dialogue systems shows that it is very difficult to get a grasp on how the DL algorithm effectually maps input and output, how to efficiently improve results, how to evaluate the quality for the generated responses, and most importantly how to perform all of this in a principled manner. Nevertheless, valuable insights have been made during the experiments which in themselves deserve attention and warning for future endeavours with this technology and models which build upon it.

The report will be outlined as follows:

1. Contextualization of the-to-be-reviewed recent trends from within the industrial landscape
2. Survey of the recent academic-industrial contributions on automatically building dialogue systems
3. Experiments with the *seq2seq* framework and custom-defined methods
4. Discussion of promising (from the reviewed literature) and novel (experimented) methods to solve identified or encountered problems
5. Conclusion summarizing the insights made in the project

2. Contextualization

Before starting with discussing the project and its results, it would be appropriate to place the aforementioned academic and industrial interest into dialogue systems within context. First of all, let it be clear that the surge of interest is rather evidence of a renewal of interest correlated with the promise of Deep Learning:



Graph 1: Relative popularity score of search terms/topics and associated buzzwords "chatbot" and "deep learning". It can be observed that searches for Deep Learning steadily increase starting 2012, whereas searches for chatbot only really recently in 2016 have spiked.

Dialogue systems (also chatterbot, conversational agent, …) have been around for several decades already (for an overview: Jurafsky and Martin 2009, Jokinen and McTear 2009, Markus Berg 2014: Chapter 3) with as one of the earliest and well known example the ELIZA system (Weizenbaum 1966), a rule/script-based computer program emulating a psychotherapist, which was even mistaken for a human by some of its patients. Like ELIZA, the more recent A.L.I.C.E (Wallace 2008), using AIML (Artificial Intelligence Markup Language), is an example of a rule-based dialogue system. Traditional - and in practice most of current - dialogue systems employ scripted dialogue management as exemplified below by a typical goal-oriented dialogue system which can be used for booking airline tickets.

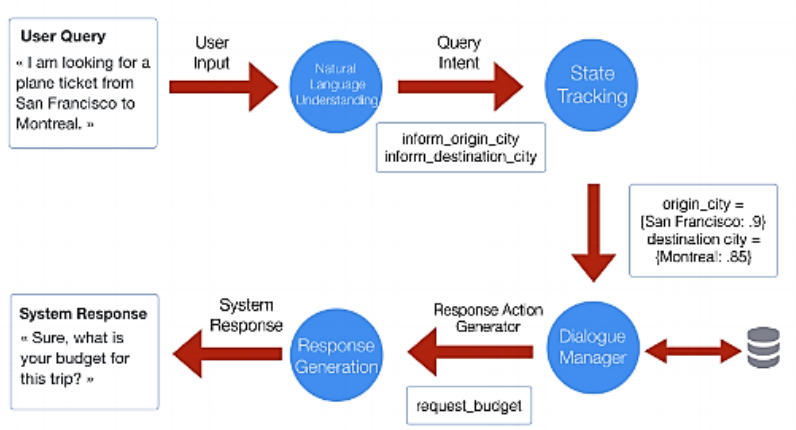


Figure 1: A graphical representation of a traditional dialogue system pipeline architecture exemplified in the domain of air travel bookings. A user's natural language input query is translated to specific intents, which are given to the dialogue manager, who gives an API call to search for the queried information in a database, retrieving possible responses. However, the system does not want to present all answers, a possibly large list, and prompts a question response via the system. Image credit: Maluuba

A modular dialogue system consists of different key components (Jurafsky & Martin 2009: 865) each optimizing separately their specific objective function. Typically, the first component of a dialogue system is a *natural language understanding* module, which extracts meaning from the input, and conversely at the end a *natural language generation* module, which translates meaning to an output response. Given the nature of the input optionally preceding and succeeding modules can be *speech recognition* and *text-to-speech synthesis*. Plausibly the most important building block in the pipeline is the *dialogue manager*. This module is connected to the other two basic components. The central position allows the dialogue manager to control the full process by tracking and updating dialogue states, additionally by sending calls to a knowledge base or task manager which contain information on the dialogue task domain. A point of criticism frequently uttered against rule/script-based dialogue management is that they are expensive to build and fragile in operation (Young et al. 2013). Alternatively, one other suggested approach to dialogue management is based on a combination of uncertainty and reinforcement learning (Young et al. 2010).

Dialogue systems have been employed in various applications with a wide range of domains, including car navigation, customer support, medical diagnosis, tourism, wealth management, language learning and amusement. Building dialogue systems for entertainment purposes[1], the so called social "chatbots", has sparked interest from the larger tech industrials and bot start-ups are sprouting in recent years. In the strive for the best intelligent assistant - Apple's *Siri*, Amazon *Alexa*, *Google Now*, Microsoft's *Cortana*, IBM's *Watson*, Nuance's *Dragon Go!* – the biggest companies in tech are leapfrogging to integrate the best natural conversational speech interaction system into their products, applications, and services. Other large tech companies take different approaches: Facebook Messenger allows implementations for bot functionality (integration with the Heroku platform), so does Microsoft's Skype, and at the

[1] Dialogue system and chatbot are sometimes used interchangeably. In this paper, we will use dialogue system as the general overarching term and chatbot/bot to refer to respectively a dialogue system exhibiting more open-domain, leaning towards social chitchat, natural conversation.

2016 Oracle OpenWorld conference Larry Ellison himself has showcased Oracle's chatbot development platform. More recently, Amazon has announced the "Alexa Prize", a $2.5 million university competition towards advancing conversational A.I.. A noteworthy example of a "successful at entertaining" chatbot is Mitsuku, which, although employing highly scripted conversation, won the Loebner prize in 2013 and 2016. Another interesting case is Microsoft's Xiaoice (Markoff & Mozur 2015) which is an engaging chatbot which converses in Chinese and has over 20 million registered users.
The increased media coverage (contextualised A.I. in the top 10 emerging technologies of 2016 by the report of the World Economic Forum) and economical investments (e.g. Google acquiring api.ai) are clear evidence of the promising future use for this technology in sectors such as healthcare (as an interactive companion) and ICT (bots replacing apps for providing information, supporting decision making and facilitating actions).

> *How does the "revival of Deep Learning" influence the tumultuous industrial landscape sketched above?*

To be clear on the matter and to appreciate the historical foundations, the first Deep Learning system dates back more than half a century (Ivakhnenko and Lapa 1965) and advances in the field have never really halted (Schmidhüber 2015 for a valuable overview). Admittedly, most of the advances were theoretical in nature and the ongoing promise of theory failed to meet any large practical applications for decades – dubbed the "A.I. Winter". Nevertheless, the "rediscovery" of Deep Learning has been attributed to the pioneers Geoffrey Hinton, Yoshua Bengio and Yann Lecun (see their 2015 paper in Nature, but also Schmidhüber's critique on the "Deep Learning Conspiracy"). The most clear, evidence-based breakthrough is presented in Hinton et al. (2012) where they used DL models to significantly outperform the state-of-the-art on five long-standing benchmarks (Reuters, TIMIT, MNIST, CIFAR and ImageNet), each representing a key Machine Learning task to be solved.
In general, finally disposing of a training algorithm that converges to a global optimum, the computational power (of GPUs [Graphical Processing Unit] from the gaming industry driving progress even further) and large amounts of data ("scaling") to put large multi-layered feedforwardnets to practical usage, Deep Learning has matured into a promising field and is seen as a prime candidate towards building "real A.I." systems.
Although most practical dialogue systems are still built through significant engineering and expert knowledge (Serban et al. 2015), recently "re-"(?)invented DL models hold promise to reduce not only the amount of production time and manual labour needed to create high-quality dialogue systems, but also to increase overall performance and domain coverage. Concerning the earlier mentioned intelligent assistants, large modules of their traditional pipelines are being (fully) replaced for DL architectures. As a preliminary conclusion to the introductory contextualisation, it suffices to state that neither dialogue systems, nor Deep Learning are new endeavours, rather that the now hot-topic DL technology backed by large tech companies is giving impetus to research into automatically building dialogue systems.

What promise holds Deep Learning more specifically for dialogue systems research?

The unit in question for building dialogue systems is "natural conversation", which due to its intrinsically difficult nature has escaped appropriate modelling by purely linguistic features, nor combinations of those with handcrafted features defined in scripted mark-up languages. A major driving force behind the adoption of deep learning technology in favour of shallow, mostly supervised methods making use of the aforementioned time and effort-taking to build feature sets, is the ability of deep nets to learn "multiple levels of representation or abstraction" (Hinton, Bengio, Lecun 2015).

More specifically, the following definition elaborates on this ability:

> "Learning multiple levels of representation to help a learner accomplish a task of interest, with higher levels capturing more abstract concepts through a deeper composition of computations" (Quora, Yoshua Bengio 2016)

This points towards the fact that Deep Learning is more concerned with discovering the features that best represent the problem or task, rather than solely finding a way to combine them (after typically being created by domain experts). In general, a representation in deep learning offers a way to embed whatever data manifold in n^n dimensions, i.e. transformed versions of data which are moulded into a form with which neural networks make their task more easy to complete.

In language modelling, continuous representations of words estimated with neural network inspired architectures have presented a way to compactly encode linguistic context, i.e. syntactic and semantic similarity. According to Baroni et al. (2014) embeddings, statistical representations of structural and contextual patterns, have been found superior on Natural Language Processing (NLP) tasks concerning semantic relatedness, synonymy, concept categorisation and analogy reasoning. They have been successfully applied in a variety of NLP tasks (non-exhaustive list), Named Entity Recognition (Passos et al. 2014), Unsupervised POS-tagging (Lin et al. 2015), Dependency Parsing (Komatsu et al. 2015), Social Media Sentiment Analysis (Wang and Yang 2015) and Machine Comprehension (Trischler et al. 2016). In the Information Retrieval scene, text embeddings are being explored in the context of very innovative themes such as induction from clickthrough data to improve web search (Shen et al. 2014), cross-lingual retrieval models (Vulic & Moens 2015) and query expansion with locally-trained word embeddings (Diaz et al. 2016).

For conversational modelling, representation learning with deep neural networks presents an unprecedented possibility to start from raw text containing dialogues and induce "dialogue representations" which could in theory be used to build dialogue systems generating appropriate contextual responses. In the next section, we will review the academic literature for recent endeavours at learning end-to-end dialogue systems directly from substantially large amounts of conversational data.

3. Literature review

A modern challenge in building dialogue systems is that traditional modular architectures do not scale well to open-domain conversation (Lowe 2016). In this regard we stick to the definition that *"[a] Natural Dialogue System* (…) *tries to improve usability and user satisfaction by imitating human behaviour*" (Berg, 2014: 44). Conversational interfaces are increasingly more expected to offer "smooth and natural exchanges" (Farrell et al. 2016) between humans and devices. Recent advances in neural language modelling show promise to make dialogue systems to communicate conversationally. More specifically, the rise of (deep) recurrent neural network (RNN) methods for flexible modelling of conversational text makes it theoretically feasible to take on building end-to-end systems at a large scale. Instead of relying on different modules with separate objectives in a pipeline architecture, learning a single model trained directly in its entirety on conversational data can broaden the scope in domains and improve general performance of dialogue systems. However, significant open challenges remain to be tackled before end-to-end trained neural networks are "able to converse naturally with humans about any subject" (Ryan Lowe 2016).

The goal of the literature review is to introduce the significant progress in learning end-to-end dialogue systems, the neural network architecture on which the advancement is fundamentally based, an overview of modern challenges – available datasets/resources, response generation issues and reliable evaluation -, and the newest models and academic contributions will be reviewed. Given that an exhaustive survey is outside of the scope of this review, references will be made to literature going deeper on specific topics.

3.1. Data-driven dialogue systems

Dialogue systems research has a long-standing tradition and a large body of work to boast for it. Data-driven machine learning methods have been advancing progress and performance in a wide range of tasks relevant for dialogue such as generation of user-tailored responses (Walker 2004), user modelling (Georgila et al. 2006), dialogue state tracking (Young et al. 2010, Williams et al. 2013, Henderson et al. 2013, Kim et al. 2015), dialogue policy learning (Young et al. 2013) and natural language generation (Wen et al. 2015a). With the advent of large dialogue datasets (Serban et al. 2015), dialogue systems research is being pushed towards data-driven approaches.

> "There is strong evidence that over the next few years, dialogue research will quickly move **towards large-scale data-driven model approaches***,* in particular in the form of **end-to-end trainable systems***"* (Serban et al. 2015: 39, my emphasis).

Typically, three main types of data-driven systems are discerned (Hastie 2016):

1) the *statistical machine translation (SMT) based method* to exploit high-frequency patterns through a combination of aligned phrase pairs constrained for lexical overlap (e.g. Ritter et al. 2011)
2) *retrieval-based systems* that seek to select (by ranking) a response from a possibly large list of utterances (e.g. Bachs & Li 2012, Lowe et al. 2016b)
3) (*neural*) *response generation systems* that seek to produce a most likely response given a conversational context by sampling word-by-word from their model's probability distribution (e.g. Vinyals & Le 2015)

The reality is however not so black-and-white, for example Sordoni et al. (2015) have built a hybrid system based on both (1) and (3).
The below graph illustrates a more high-level categorization of data-driven dialogue systems in the spirit of Serban et al. (2015: 4-5):

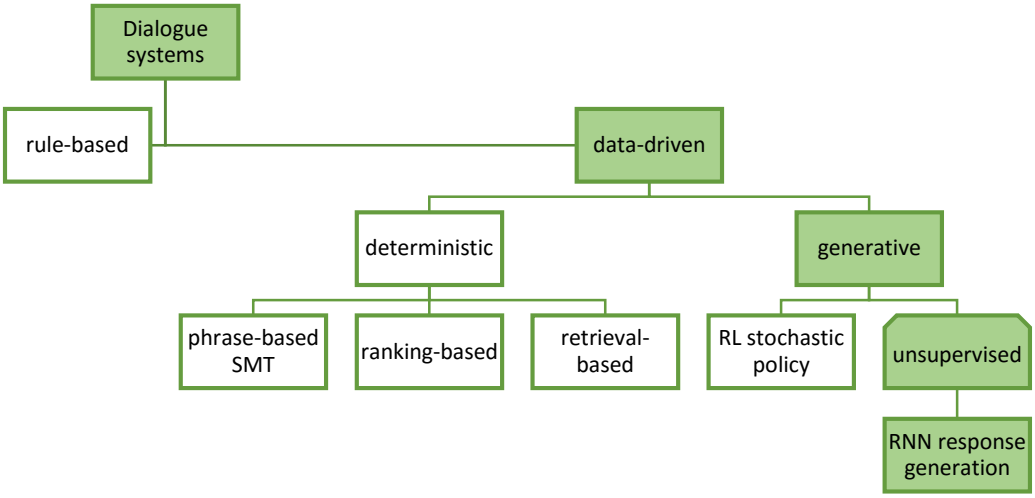


Figure 2: From left to right: rule-based systems are posited as the anti-thesis to data-driven systems. The latter type can be subdivided into deterministic, where the system's responses present a fixed set, and generative, where the system samples a likely response given conversational input, response generation systems. Each of these have representative subtypes.

This categorisation is neither perfect (see Serban et al. 2015 for categorization with different dimensions), but suits to illustrate how the diverse models approach dialogue modelling from the perspective of data. With regard to neural response generation systems, they are considered as *unsupervised* learning models due to their aim of reproducing data distributions, i.e. by maximizing likelihood with respect to words in the training corpus. In alternative fashion, the absence of any "explicit supervised signal such as task completion or user satisfaction" (Lowe et al. 2016b: 1) gives a strong point from an evaluation perspective towards classifying both retrieval-based and end-to-end response generation systems as unsupervised models.
In the discussion we will limit ourselves to a particular type of dialogue systems, the recently proposed, ***end-to-end trainable neural* response generation systems** (Vinyals & Le 2015, Sordoni et al. 2015, Shang et al. 2015, Wen et al. 2015b, Li et al. 2016a, Serban et al. 2016a, Serban et al. 2017) and a specific modality, text-based (written) dialogues. This type of models

aim to generate a highly probable textual response to a given conversation context or dialogue history. Specifically for dialogue systems, end-to-end training means training a single model both in its entirety, from one end – the input – to the other – the output –, and directly on conversational data. This implies that not only understanding and generation components are to be learned from the data, but that the model should also perform the functions of dialogue management, primarily (but not limited to) state-tracking. End-to-end dialogue systems hold many advantages to traditional modular architectures (Lowe et al. 2016a): it does not require any feature engineering, only architecture engineering and design choices; given adjusted training data a system is easily transferrable to different domains; and no supervised data is required for discriminatively learning and improving each separate module. On the contrary, one major necessity to make end-to-end trainable dialogue systems perform well is a large amount of training data. In order to understand the end-to-end training inspiration for the response generation models and the principles underlying these models, a small high-level introduction to the neural network-based *seq2seq* framework is needed.

3.2. Sequence-to-Sequence Learning

3.2.1. Encoder-Decoder Architecture

The sequence-to-sequence learning framework is essentially a family of approaches where encoder-decoder models are being used to look for an intermediate representation of meaning when mapping one complicated structure to another (Sutskever et al. 2014). Recurrent neural networks lend themselves naturally for modelling sequential data (the unreasonable effectiveness of RNNs, Karpathy 2015), although training and optimizing them is remarkably difficult (Bengio et al. 2013). However, natural language sentences typically exhibit long-distance dependencies for which a long short-term memory (LSTM, Hochreiter & Schmidhuber 1997) architecture or a gated recurrent unit (GRU, Cho et al. 2014) are more suited. Respectively, LSTMs deal with vanishing gradients by using a gating mechanism and GRUs also employ gating components to control the flow of information.

Next to the canonical encoder-decoder architecture (for a more mathematical description see Li et al. 2016a: 995), a special *soft attention mechanism* (Bahdanau et al. 2014) with inspiration from the human perceptual system has been introduced into *seq2seq* models to allow for more efficient encoding of long sequences which should be encoded into a fixed length context vector by letting the decoder to look selectively at the input sequence at every decoding step.

The generative encoder-decoder model is based on its application in the context of Machine Translation, where an input needs to be translated from a source into an output in a target language. A typical loss function for *seq2seq* models demonstrating this MT inspiration is negative loglikelihood, $-\log(\mathrm{P}(\mathrm{target}|\mathrm{source}))$. For example, Sutskever et al. (2014) make use of a "multilayer LSTM to encode the input sequence to a vector of a fixed dimensionality, and then another deep LSTM to decode the target sequence from the vector". A general trick

(amongst others such as padding and bucketing vectors) they propose is to reverse the input sequence to introduce more short-term dependencies to be encoded.

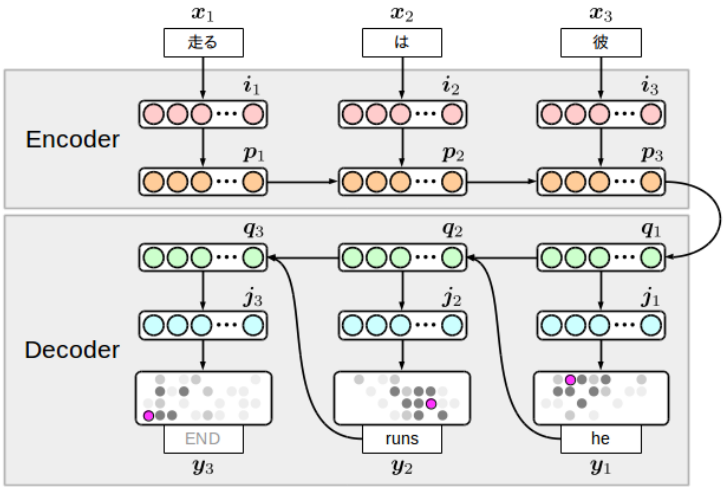


Figure 3: Here we see a typical encoder-decoder model used in translation from the source language (Japanese) to the target language (English). Layers *i and j* are used to build a distributed vector representation for each input and output word. The final hidden state of p contains all the previously encoded information in a summarizing context vector, which is then decoded one word at a time to generate the full output sequence. Image credit: https://goo.gl/XDf66h

The advantages of *seq2seq* models are that they can capture contextual dependencies, both short and long-term, can implicitly learn semantic and syntactic relations between sequence pairs, and most importantly offer the promise of scalability, given enough training data, and language independence (Li et al. 2016b: 1).

3.2.2. *Seq2seq* applications

Significant advances in end-to-end training with these models has made it possible to build successful systems for various complex natural language tasks. Machine Translation, being the inspirational task (Kalchbrenner & Blunsom 2013, Sutskever et al. 2014, Cho et al. 2014, Cho et al. 2015, Luong et al. 2014, Luong et al. 2015) for the construction of these models, has certainly benefited from these advances. Quite recently, March 2016, Google has announced their Neural Machine Translation system based on *seq2seq* models, which significantly outperformed their original phrase-based SMT system. Interestingly, the use of these models has made even zero-shot translation possible (Johnson et al. 2016). In a related note, the HarvardNLP group have launched OpenNMT, an open-source neural machine translation system. Other notable NLP tasks in which *seq2seq* models have been successfully applied are syntactic parsing (Vinyals et al. 2015), generative question answering (Yin et al. 2015), abstractive text summarization (Rush et al. 2015) and neural conversation agents (Vinyals & Le 2015).

Given the exposition above, it should be clear that in the *seq2seq* approach as a simple extension dialogue is cast as a "source to target sequence transduction problem" (Wen et al. 2016: 1), where an encoder is used to map a user query into a distributed representation representing its semantics, which is then used to condition a decoder to generate a system

response. In fact, the strength of such a model lies in its simplicity, generalization capacity and generative properties. However, the obvious simplifications in the base *seq2seq* model with respect to the nature and actual objective of dialogue translate themselves into various problems. Notwithstanding the sometimes impressive sample conversations from end-to-end dialogue systems, a lot of challenges are to be tackled in order to make these systems ready for real-life applications. Given remarkable success in other NLP tasks yet nuanced by the previous argument, dialogue systems with neural generation models as their backbone present both a great opportunity and a grand challenge. In the next sections, the challenges underlying data-driven dialogue modelling and generation with *seq2seq* models will be presented from the work addressing these.

3.3. Dialogue datasets and tasks

A major prerequisite towards building robust general-purpose dialogue systems, which can perform well given any variation of input, is a sufficiently large amount of data for training. In order to advance progress in the field and foster innovation, large (>500k dialogues) *open-source* datasets are needed (Lowe et al. 2016a). Given that we are dealing with essentially highly data-driven models, dataset construction presents an important challenge. In order to make progress towards general-purpose dialogue systems, there is a need for both open and closed domain datasets.

3.3.1. Available resources

An important resource on existing dialogue datasets, is the survey by Serban et al. (2015), where a classification (modality; constrained, unconstrained; human-human, human-machine) and an individual characterization (content, size, url) is given of each dataset.
We will shortly introduce some of the most used human-human written dialogue datasets.
The *Ubuntu Dialogue Corpus* (Lowe et al. 2015) is one of the largest currently available dialogue datasets consisting of around 1 million tech support dialogues. More specifically, 2-person dialogues have been extracted from archived chatlogs hosted on Ubuntu's public Internet Relay Chat technical support channels. One drawback to this dataset is that the conversation is focused on very technical topics and is thus as training data not very useful for evaluating progress in open-domain end-to-end dialogue systems.
The *Twitter Corpus* (Ritter et al. 2011) is almost equally large as the former dataset, collecting over 850.000 tweets and replies extracted with an API from the social media platform Twitter. It has mostly been used to model open-domain short conversation, given that any tweet is limited to 140 characters.
Another very frequently used open-source dataset is the *Cornell Movie-Dialogs Corpus* (Danescu-Niculescu-Mizil & Lee 2011), a large metadata-rich (IMDB rating, genre, character gender etc.) collection of "fictional" conversations that were extracted from raw movie scripts. In total, it contains around 300.000 utterances from 10.000 pairs of movie characters. One

point of warning, although the conversations are varying in length and presenting open-domain conversation, the movie-scripted nature should be taken into account during evaluation.
Lowe et al. (2016) makes an interesting case that there are a lot of very good existing datasets that are however proprietary. If a company would decide to make a dataset available for the research community, it should always take into account that no public or sensitive data are present, which calls for time and labour intensive data screening and possibly anonymizing the data to prevent any information leakage. This remark does not only apply to the datasets, but also the models trained on such proprietary datasets, and with special regard to the "generative" neural models in discussion. A cautionary tale: anything which features in the training data can possibly be generated as output by sampling from the learned probability distribution. For example see the below sample conversation of a sample model trained on the Cornell Movie Dialogues Corpus:

> Human: what can you talk about?
> Robot: I'm not a ah, f*ck it, just stay away from my f*cking lady friend, man

As can be observed, these models sometimes return unpredictable answers and might not be suited to handle the full conversation without some additional (human or automatic) filtering supervision.

3.3.2. Towards open-domain dialogue data collection

Essentially, end-to-end dialogue systems use computational methods based on next utterance prediction for learning dialogue strategies offline from large amounts of text data. So as to learn relevant responses to any input, you will need a lot of different datasets exhibiting large variation in the mappings between queries and responses. With regard to the variation in mappings, question answering presents the first step in difficulty, followed by recommendation, and finally social chitchat presents the holy grail in interactional capacity for any end-to-end trained dialogue system. Serban et al. (2015: 34) suggest transfer learning between datasets as a feasible approach towards building open-domain dialogue systems. Similar to humans, dialogue systems have to be trained on multiple different data sources for solving multiple different tasks. Certainly for chat-oriented dialogue systems, it is necessary to leverage external information from sources such as news articles, movie reviews or online encyclopaedia to make the models converse in an entertaining fashion on real-world events with a user.

3.3.3. Synthetic data and tasks

Moreover, for end-to-end trained dialogue systems there is no "intermediate task or state that can be used for training and evaluating" (Bordes NIPS 2016). Beside natural conversation data, synthetic data can be used to evaluate the aptitude of end-to-end dialogue systems in

different conversational situations, or in the case of goal-oriented systems different stages of a (most possibly) multi-turn transaction. *Maluuba Frames* (2016) is a prime example of a synthetic dataset made available to the research community to drive progress towards the goal of automatic text understanding and reasoning. The synthetic dataset is the result of applying a crowdsourcing version of the Wizard-of-Oz paradigm (Kelley 1984) to create dialogues spanning various topics of increasing conversational complexity. A similar tactic has been employed in Wen et al. (2016) to collect domain-specific corpora. Given the success of the synthetic bAbi tasks for question answering research (Weston et al., 2016), a set of 20 basic reasoning tasks generated from simulated stories, Facebook AI Research have now also created bAbI tasks for goal-oriented dialog (Bordes and Weston 2016).
The collection of synthetic datasets and tasks is a good incentive and can certainly help to identify both pitfalls and new possibilities towards advancing the state-of-the-art in end-to-end dialogue systems. Nevertheless, collecting larger and better-quality training datasets representing various domains should be the first priority.

3.4. Evaluation of neural generation models

3.4.1. What to evaluate?

The evaluation of dialogue systems can be considered one of the hardest challenges in constructing dialogue systems. With respect to the long-term goal of building intelligent conversational agents advanced evaluation of all aspects in a dialogue system is required. Below figures an evaluation hierarchy (adapted from Hastie NIPS 2016) that can represent the complexity of fully evaluating a model from the bottom-up:

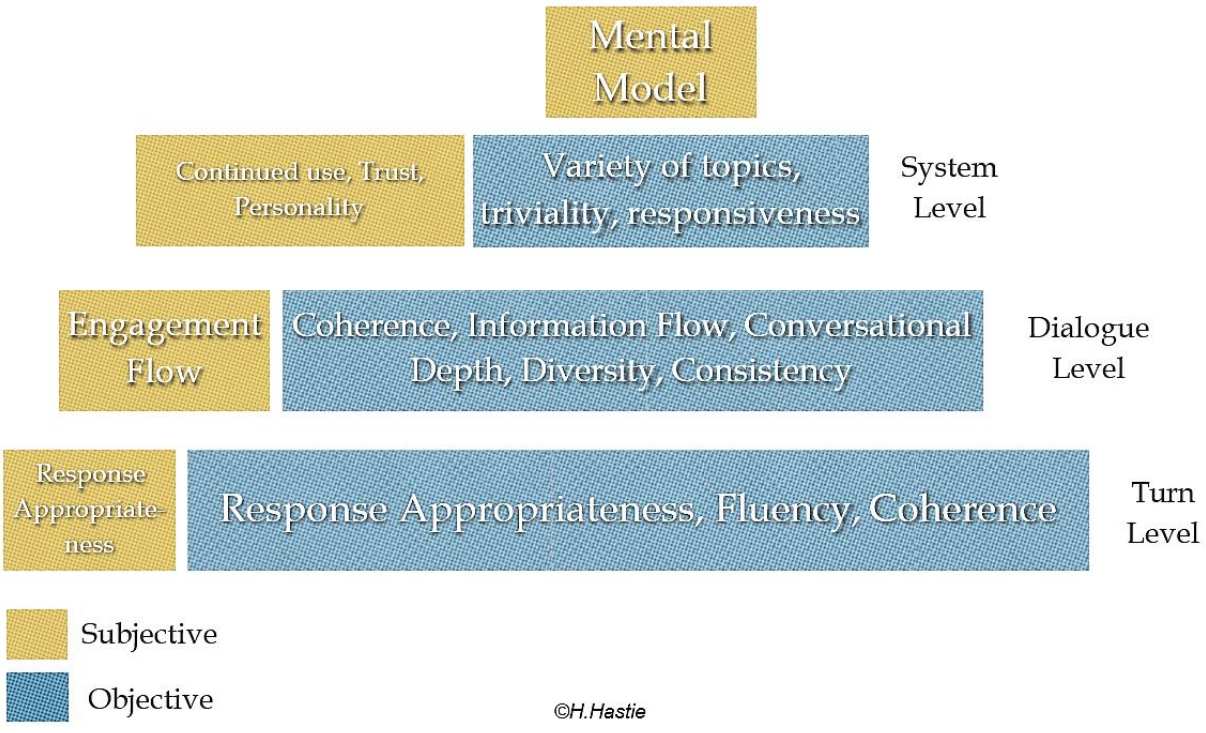


Figure 4: The evaluation hierarchy as defined by Helen Hastie (NIPS 2016 presentation). It gives a structured overview of what subjective and objective measures should be taken into account at the different levels of evaluation. Furthermore, the hierarchical composition implies that even in the instance where the three lowest levels have individually near-perfect scores, if the mental model of the user does not agree with the system, changes need to be made in order for it to be successful.

When constructing a dialogue system, one needs access to some ideally automated pseudo-performance metric to measure progress and to make a decision as to when a system is ready to be released in any given (even beta-testing) application to the real world (Serban et al. 2015: 37). It is desirable to have some reasonable degree of confidence that the built model is optimized before submitting it to human evaluation.
With regard to evaluation, two types of assessment can be discerned: ***extrinsic* evaluations** which make an assessment of the "effect of a system" from an external perspective such as task completion (e.g. successfully helping a user book a flight of his choosing) and ***intrinsic* evaluations** which take only the properties of the system in itself into consideration. In the discussion, we will focus mainly on the latter type, and more specifically on the evaluation of non-goal oriented, "unsupervised" generative dialogue systems. It should be noted that the evaluation of any unsupervised system presents a general problem for machine learning research. The evaluation of non-goal oriented models is most problematic, which is clear from its roots in as early as the Turing Test (Turing 1950). Conversations related to chitchat have no clear objective, making evaluation very difficult. "One of the challenges of this line of work is to develop methods to automatically evaluate, either directly [with a metric (Galley et al. 2015)] or indirectly [through user simulation (Schatzmann et al. 2005)]" (Lowe et al. 2016b) the quality of responses generated by an end-to-end unsupervised dialogue system.

3.4.2. How to evaluate?

The current methods for evaluation focus on comparing the system's generated response to a ground truth next response. An important resource on these methods is "How NOT To Evaluate Your Dialogue System" (Liu et al. 2016). It offers an overview of which automatic metrics for evaluating generated responses have been used in other publications and give critical advice on what methods (not) to use. Below a more up-to-date overview is given of which metrics have been used in the literature:

Evaluation of generative approaches	
Automatic	**Manual**
N-gram diversity (Li et al. 2016b); **BLEU** (Li et al. 2016a, Sordoni et al. 2015), **DeltaBLEU** (Galley et al. 2015); **Length metrics** (Mou et al. 2016, Li et al. 2016b); **Perplexity** (Vinyals & Le, 2015); **ROUGE** (Gu et al., 2016); **METEOR** (Sordoni et al., 2015); **Embedding-based metrics** (Serban et al. 2016b, Serban et al. 2017)	**Pairwise comparison with rule-based system** (Vinyals & Le, 2015); - **between models** (Li et al. 2016b, Wen et al. 2016, Serban et al. 2016b); **next utterance rating** (Sordoni et al. 2015) ; **5 turn 3rd party rating** (Li et al., 2016b)
+++ **fast, uncostly, scalable, easily reproducible** --- **non-correlated with human evaluation**	+++ **test specific quality**, **representative** --- **costly**, **non-reproducible**, **possibly biased**

Table 1: This table offers an overview on what automatic and human measures have been used for the quality evaluation of response generation by unsupervised dialogue systems. Expanded version of Helen Hastie (NIPS 2016) with evaluation of evaluation by Antoine Bordes (NIPS 2016).

3.4.2.1. Automatic metrics

What can be observed is that the in practice used automatic metrics have almost all been inspired by metrics which are successfully used in Machine Translation (BLEU (Papeneni et al. 2002), METEOR (Banerjee & Lavie 2005)) and Text Summarization (ROUGE (Lin 2004)). However in the comparative study of (Liu et al. 2016: 9), it has been proven that "many metrics commonly used in the literature for evaluating unsupervised dialogue systems do not correlate strongly with human judgment." It is widely debated how well the "borrowed" automatic metrics correlate with true response quality, such as what do the absolute numbers mean and what aspects of quality do they measure? One family of metrics shows possible promise: the embedding-based metrics, which do not rely on any word-based overlap for evaluating relevancy of responses (overview and description: Liu et al. 2016: 3). However, the community should keep on the look-out for model-independent automated metrics which capture the quality of the dialogue system response generation well and can accelerate the deployment of these systems.

3.4.2.2. Manual & third-party human

As a final point of advice, Liu et al. (2016: 10) posit that for the meanwhile human evaluations, although costly and non-reproducible, should best be used for quality assessment, possibly accompanied by evaluation with multiple of the automatic metrics. Initiatives such as the crowdsourcing platform Amazon Mechanical Turk can help in making human evaluation scalable, although the non-expert status of the paid subjects can possibly lead to biased results (Young et al. 2013). A more tangible advantage of evaluation by (third-party) human evaluators is the possibility to devise small dialogue tasks that can measure a specific criterion in the loop, such as coherency, relevancy, non-repetitiveness or robustness. With respect to this, both fixed datasets for offline evaluation and online evaluation benchmarks should be made available to the general research community. The dialogue tasks (cf. 3.3.3) for

measuring the quality of goal-oriented dialogue, although easier than for non-goal oriented, presents a good first step in this regard. In conclusion, automated tasks and metrics are necessary engineering requirements for scaling the development of small domain-specific, goal-oriented dialogue systems to conversation agents which perform well even in non-goal oriented settings (Pietquin & Hastie 2013).

3.5. Generally identified issues and contributions

In this section, we will review the literature for general issues that have been identified from the start, when the first neural generative models started generating output resembling humanlike conversation, until the present, and what different learning architectures have been experimented with to mediate inherent issues with neural response generation.
The discussion will inescapably touch on work addressing the even larger unsolved long-term challenge of intelligent machines learning to communicate in natural language. According to Jiwei Li (2016), some logically requisite capabilities for an successful conversational agent are that it needs to understand what a user asks and should be able to "generate coherent, meaningful sequences in response to what is asked, that require domain knowledge, discourse knowledge [and] world knowledge".
Generative neural network architectures offer promising methods towards progressively more complex interaction and intelligent behaviour in communication. In "A Roadmap towards Machine Intelligence" (Mikolov et al. 2016) a holistic view is taken on addressing all aspects of intelligence within a single system. Some desiderata have been identified with regard to Communication-based Machine Intelligence (Baroni NIPS 2016b):

- Ability to communicate through natural language
- Ability to learn about new tasks efficiently
- Ability to learn, efficiently, through language
- No explicit supervision, sparse reward, self-motivation

In related fashion, Dodge et al. (2015) discuss what prerequisite qualities should be evaluated ranging from question answering to chitchat for learning end-to-end dialogue systems with a given neural network model.
In the discussion, we hope to shed light on what deficiencies have been identified in the unsupervised generative architectures, both on the system and the dialogue output level, what solutions have been suggested to the learning architectures and finally where there is still room for improvement.

3.5.1. Local dialogue context

3.5.1.1. Genericity problem / universal reply

> "Sequence-to-sequence neural network models for generation of conversational responses tend to generate safe, commonplace responses (e.g., I don't know) **regardless of the input**." (Li et al. 2016a: 1, my emphasis)

A most discerning problem is encountered when neural conversation models are used in practice. The response generation system answers almost any input with a "universal reply", a non-content rich, universally relevant utterance such as "ok", "sure", "I don't know" (Serban et al. 2015, Sordoni et al. 2015). The phenomenon can be most probably ascribed to the relative high frequency of such generic responses in the conversational training datasets and the maximum likelihood training objective function. The latter seeks to optimize the likelihood of outputs given training data inputs. In fact, for generation this objective is less suited, seeing that it effectively optimizes compression, but thereby restricts intrinsically different outputs. Obviously, we would prefer output that is grammatical, shows a high grade of coherence and is diverse and interesting. Li et al. (2016a) propose to use another more suited objective function, Maximum Mutual Information (MMI), which takes its inspiration from information theory. This optimization objective is changed to at generation time to forcefully increase diversity in responses by maximizing the mutual information between the context (previous dialogue sentences) and the response. Intuitively, context-specific responses would be selected over the universal fall-back strategy with unconditionally probable responses (Farrell et al. 2016) and overall by adhering to MMI semantic coherence throughout the dialogue should also be increased. This strategy to condition responses on the context is a first good step towards making an end-to-end conversational agent provide interesting, diverse and informative responses. Alternative approaches are *seq2BF* (Mou et al. 2016), a sequence to backward and forward sequences model, which builds on the previous MMI-based strategy by generating a more content-rich reply containing a given predicted related content (as measured by PMI) keyword candidate at any position. A similar study is presented in Xing et al. (2016) where replies are augmented with topic information instead of a related keyword predicted to be rich in content.
The notorious issue of genericity is not yet "solved" and will require larger corpora to learn semantic representations of dialogue and to present a larger training context exhibiting longer conversations, more variation in users and different grades in domain-specificity.

3.5.1.2. Speaker consistency

The connotation of a dialogue system as a "conversational agent" is important with respect to response generation in applications such as personal assistants, non-playable game characters and automated customer support. The response generation by unsupervised dialogue systems implies simulation of human behaviour by different persons present within the training data. In more carefree conversational situations it is prone to deliver inconsistent responses such as exemplified below:

message	Where do you live now?
response	I live in Los Angeles.
message	In which city do you live now?
response	I live in Madrid.
message	In which country do you live now?
response	England, you?
message	Where were you born?
response	I was born in Canada.
message	Where are you from?
response	England, you?
message	Where did you grow up?
response	I grew up in Texas.
message	How old are you?
response	16 and you?
message	What's your age?
response	18.

Figure 5: Inconsistent responses generated by a 4-layer *seq2seq* model trained on 25 million Twitter conversations (Li et al. 2016c). In the first two conversations, the system changes its geographical identity and in the last it gives changes its profile from a minor to an adult.

"A Persona based Neural Model" (Li et al. 2016c: 1) shows that *seq2seq* models provide a "straightforward mechanism for incorporating personas as embeddings". A "persona dimension" permits the incorporation of background facts for user profiles, person-specific language behaviour and interaction style. As a result this allows the agent to establish common ground with the user and make use of recurrent conversation patterns. For example, the system will adapt its responses based on clusters of users, viz. Brits will be responded to differently than to Americans. User modelling is an important aspect which should be taken into account when relying on unsupervised dialogue generation systems. Additionally, user models together with dialogue history possibly spanning over multiple conversations (Al-Rfou et al. 2016) can be used to personalize ranked responses.

3.5.2. Global dialogue context

3.5.2.1. Multi-context response generation

End-to-end dialogue systems perform jointly natural language understanding and generation, which modelled in data-driven fashion is a complex feat worthy of recognition. Traditionally, a dialogue manager controls these modules, reasoning with current or new queries and updating dialogue states conditioning the response generation. Given the absence of dialogue management of any kind in the canonical *seq2seq* model which is being used for unsupervised conversational modelling, multi-context response generation presents a big challenge.

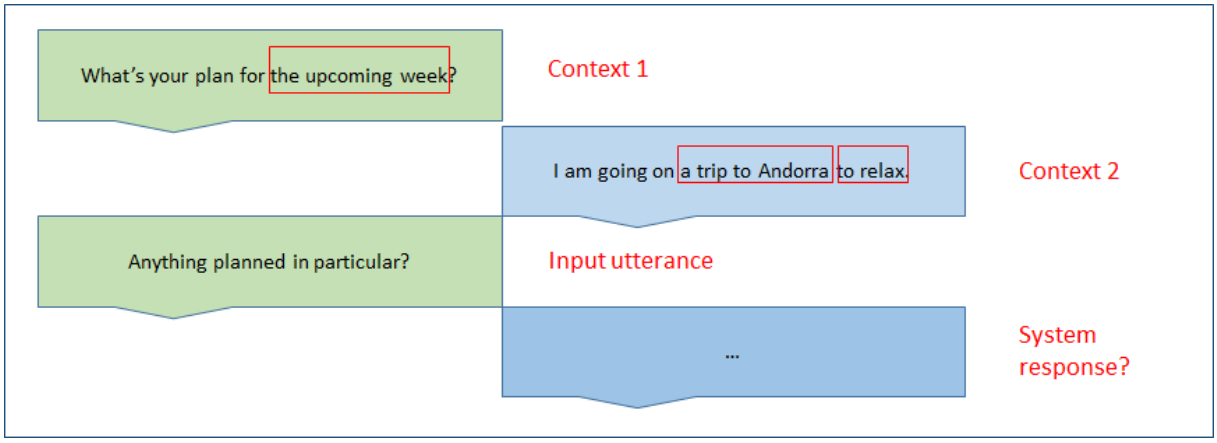


Figure 6: A sample conversation based on (Li 2016). The next to be predicted system response is conditioned by the input utterance and all preceding context sentences, where important informational parts have been marked with red boxes.

What can be discerned from the above sample is that there is a need for both a long-term memory module to take into account long-distance dependencies spanning possibly multiple context sentences, and a content-based attention mechanism, which can extract important information from the context sentences in memory, so that not everything has to be encoded.

> "It is clear that a full end-to-end goal-driven dialogue system should not only output a final sentence to respond an input sentence, but also **keep and update fruitful internal representations or memories for dialogues.** The internal memories can be either explicitly extracted and represented, or validated by some external tasks such as question answering." (Wang & Yuan 2017: 13, my emphasis)

To put the above observation into perspective, we need to address (resurged) research dealing with the bigger picture of reasoning, attention and memory (RAM):

> "Two closely related challenges in artificial intelligence are designing models which can **maintain an estimate of the state of a world with complex dynamics over long timescales [1]**, and models which can **predict the forward evolution of the state of the world from partial observation [2]**." (Henaff et al. 2017: 9, my emphasis)

[1]: **state-tracking of the world over time**

Efforts are being made to develop learning algorithms that can combine components of short-term, long-term memory and effective attention mechanisms that can access and reason with what is stored in memory. The interplay between long-term dependencies and short-term context presents itself in open-domain conversation and requires models with deeper reasoning capacities. A lot of research is being performed to implement stacks, addressable memory and sophisticated attention and pointer mechanisms to increase the ability of RNN-based models to model larger sequences and allowing reasoning with language. Much attention is given towards improving or even changing the learning architecture, i.e. by replacing RNNs, LSTMs and GRUs (e.g. Fast Weights, Ba et al. 2016) with more complex architectures that allow extrinsic memory representations (e.g. Memory Networks, Weston et al. 2014) and possibly the incorporation of – or rather achieve through attentional

processes interaction with - external knowledge into a network (e.g. Neural Turing Machine, Graves et al. 2014 ; Key-value networks, Miller et al. 2016).
One such representative model are ***End-to-End Memory Networks*** (Suhkbaatar 2015), where the explicit "memory" works analogous to the attention mechanism presented in (Bahdanau et al. 2014) - which finds the hidden states that are most useful for outputting the next to be generated word - but here attention is applied over multiple sequences (sentence-wise) instead of a single one (token-wise).

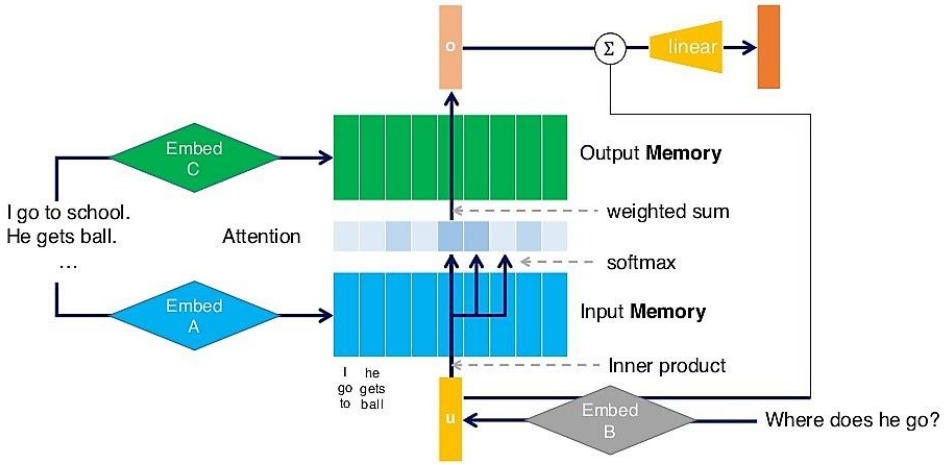


Figure 7: A high-level overview of end-to-end memory networks in the context of question answering. Embedding A contains the full set of input sequences to be embedded into input memory vectors, Embedding C contains the corresponding output vectors to be embedded in output memory vectors, and Embedding B contains the query which is also given a vector representation. The final predicted answer is given by passing a sum of the output vector (o) and the query input embedding (u) through a final weight matrix (linear).

Experiments presented in recent publications focus on question answering and reading comprehension evaluated with synthetic tasks [bAbi tasks (Weston et al. 2014)] or large datasets [e.g. Children's Book Test (Hill et al. 2016a) and SQuAD (Rajpurkar et al. 2016)] to show how well the suggested models can reason over what information it has stored. Respectively in the present, the state-of-the-art models for both tasks are *Recurrent Entity Networks* (Henaff et al. 2017), which successfully solve all 20 bAbI tasks, and the *Gated-attention Reader* (Dhingra et al. 2016, Yang et al. 2017), where multiplicative gating mechanisms combined with a multi-hop architecture give substantial improvements over competitive reading comprehension baselines.
For multi-context response generation end-to-end dialogue systems need to draw on the insights from the above introduced models by investigating smart methods to condition decoding and generation on the full dialogue context. Additionally, neural response generation can benefit also from more research into local (structure) attention mechanisms. For example, Gu et al. (2016) consider a ***Copying Mechanism***, which is an important mechanism in human language communication:

A: I went to the bakery this morning
B: Ah, did you bring the sandwiches I asked for?
A: Not really, the baker **forgot to bake them**…

B: What do you mean by "he **forgot to bake them**"?

The copying mechanism works by locating a certain segment of the input and putting it in the output sentence. In some way the copying mechanism tries to achieve the same result like transformations rules in a Machine Translation context, for example:

"Hello, my name is **X**" -> Nice to meet you, **X**.

Next to normal sequence generation, different mechanisms should be explored by drawing inspiration from how humans (or animals) use language (linguistics!) and use stored memories. In spite of the resurgence in interest for learning algorithms combining "RAM", the proposed algorithms are still in its infancy and can benefit from grounding in cognitive science.

Simple *seq2seq* models fail to capture the difficult, hierarchical flow of natural dialogue, showing little to no progression in natural conversation. One other suggested approach is imposing a hierarchical structure on dialogue, jointly modelling the utterances (comprised of tokens) and interactive structure (comprised of utterances) of the conversation. For this purpose, Serban et al. (2016b) extended the ***Hierarchical Recurrent Encoder-Decoder Neural Network*** (HRED) proposed in (Sordoni et al. 2015) originally for web query suggestion:

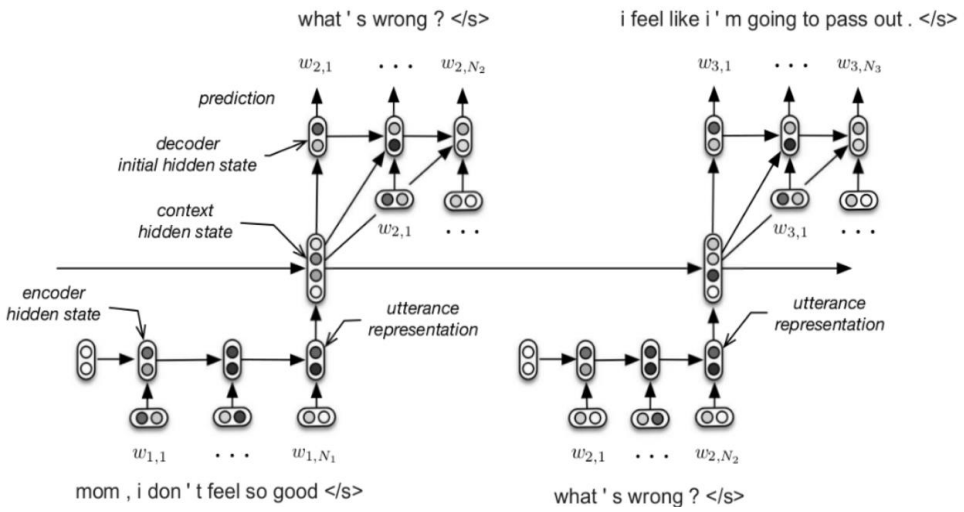


Figure 8: A computational graph representing the HRED architecture for dialogue over a span of three turns. The major addition to the architecture is a higher-level *context*-RNN keeping track of past utterances by progressively processing over time each utterance vector and conditioning the decoding on the last hidden state of the context vector (middle).

By making the architecture hierarchical with the addition of a context-RNN, the temporal long-term structure of dialogue is taken into consideration by the model, effectively mimicking a dialogue state tracker and allowing multi-turn conversation.

Another suggested complementary approach by the same research group from Montreal University, seeks to generalize the seq2seq architecture towards multiple input and output pairs, where each sequence represents its own stochastic process. Motivated by fact that existing models have been unable to generate meaningful responses taking dialogue context

into account and thereby failing at learning high-level abstractions of dialogue, they propose a novel neural network architecture, ***Multiresolution Recurrent Neural Networks*** (Serban et al. 2017), capable at modelling complex output sequences with long-term dependencies. More specifically, the twofold hierarchical architecture allows modelling multiple sequences in parallel, one being a simple sequence of natural language tokens (natural language) and the other a sequence representing high-level coarse tokens (inference) such as the suggested Noun Representation or Activity-Entity Representation (Serban et al. 2017: 5). The full architecture is trained by maximizing joint log-likelihood over both sequences, which during optimization should bias the models towards making high-level abstractions. The hierarchical generation model exhibits compositional learning skills, making it more adept at overcoming the sparsity of natural language and capturing the long-term flow of dialogue. Moreover, the suggested models follows the intuition that in order to generate structured output – coherent, relevant, informative and possibly long – you should also present to some degree structured input – here by adding high-level representations – to aid the RNN-based language generation.

[2]: **forward evolution prediction of the state of the world from partial observation**

One of the requirements for an intelligent conversational agent is that it can *learn while conducting conversations* (Weston 2016). It should be able to learn from feedback presented in the form of natural language when and how to adapt to the conversation.
With the goal of handling long open-ended conversation, end-to-end dialogue systems are expected to efficiently track many variables such as topic of the conversation, speaker goals, interaction style, sentiment etc. which are latent in nature, i.e. only inferable from observation. In natural human-human conversation these variables influence the dialogue greatly and people adjust their responses accordingly. An end-to-end dialogue system should also address these features by conditioning generation based on these latent variables and predicting forward how the next generation step will be influenced. Serban et al. (2016c) propose to represent this variability, uncertainty and ambiguity in the current and next states, by augmenting the HRED architecture with latent stochastic variables which possibly span over a variable number of turns. The hierarchical generation process represented by a probabilistic distribution is conditioned by latent variables which allow for modelling complex, highly variable intra-sequence dependencies.

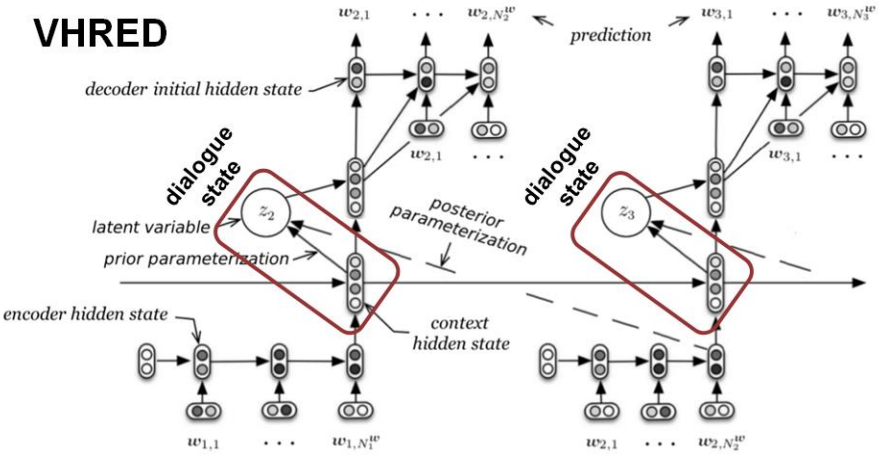


Figure 9: The context RNN is augmented with latent variables, which represent the uncertainty about the current dialogue state represented (progressively) by the hidden state of the context vector for each utterance pair. The output variation is both conditioned by the latent variable at the sequence level and by the input at the token level $w_{1...N}$.

The above architecture presents a possibility and the need for implementing a form of belief tracking into end-to-end, probabilistic generative dialogue systems. Incorporating an additional predictive look-ahead is another promising approach for optimizing the generation policy towards the best possible answers (Weston 2016). End-to-end dialogue systems learn characteristically in an unsupervised fashion from data. However, by purely relying on simulated human behaviour in conversational data, learning dialogue strategies off-policy (not related to the current (state of the) conversation (history)) might not be the best approach. Continuing training on-policy and learning from real user's feedback within the given conversation can increasingly improve the natural conversation abilities of dialogue agents. Weston (2016: 3) shows how dialogue learning can succeed in practice and suggests specific tasks and strategies for learning from teacher responses implying varying grades of feedback to be learned from that are more informative than "correct/incorrect". By incorporating the identified feedback, "linguistic cues of success", back into the system, the model will have a conditioned notion of what (not) to generate in further states, effectively using predicted knowledge to constrain dialogue response generation. The suggested on-policy learning from feedback (with an implicit reward) is an idea which will be reprised in the discussion of the next issue.

3.5.3. Deep Reinforcement Learning for dialogue generation

A strong claim postulates that "all communication is either task-driven or purpose-driven" (Baroni 2016a). However, for open-domain "chitchat" communication, defining exactly what is success or the immediate purpose is very hard. The conversation in itself is inherently rewarding and the longer, the more satisfied one might be, provided it is not too long and keeps the hearer engaged.

A big issue with neural response generation that has already been touched upon by the earlier discussed models, is the inability to account for repetition with dull, generic responses. Simple

neural models of dialogue generation tend to generate short-sighted responses, conditioned primarily (if not solely) on the next utterance in the training data.

To put it forward more concretely, conversations with end-to-end trained dialogue systems do not approach the human-level of lengthy and engaging conversation by far. Li et al. (2016b) propose combining the natural generation capacity of *seq2seq* models with the strengths of reinforcement learning, i.e. learning which action to take given a state, thereby maximizing the overall reward function (the span of a conversation). Below an example is given from (Li et al. 2016b: 2) where we clearly see the lack of any progression in the conversations, even with the MMI-based model as baseline (left):

Baseline mutual information model (Li et al. 2015)	**Proposed reinforcement learning model**
A: Where are you going ? (1)	A: Where are you going ? (1)
B: I 'm going to the restroom . (2)	B: I'm going to the police station . (2)
A: See you later . (3)	A: I'll come with you . (3)
B: See you later . (4)	B: no , no , no , no , you ?re not going anywhere . (4)
A: See you later . (5)	A: Why? (5)
B: See you later . (6)	B: I need you to stay here. (6)
...	A: I don't know what you are talking about. (7)
...	...
A: how old are you ? (1)	A How old are you ? (1)
B: I'm 16 . (2)	B I'm 16. Why are you asking ? (2)
A: 16 ? (3)	A I thought you were 12. (3)
B: I don't know what you are talking about . (4)	B What made you think so ? (4)
A: you don't know what you are saying. (5)	A I don't know what you are talking about. (5)
B: I don't know what you are talking about . (6)	B You don't know what you are saying. (6)
A: you don't know what you are saying. (7)	...
...	...

Figure 10: Left: results of dialogue simulation between two conversational agents trained on the Opensubtitle dataset. Right: results of using additional reinforcement learning to keep the dialogue flow going for as long as possible. The first turn is always inputted manually. It is clear that the conversations on the right side show more engagement and that the model with reinforcement learning achieves reasonably more "entertaining" dialogue.

More specifically, in the combined approach Li et al. (2016b) use heuristics to approximate "good" conversation implying a forward-looking perspective, interactive turn-taking, informative utterances and coherence in responses. Additionally, the reinforcement learning setting allows the definition of custom reward functions. Here the model is mostly encouraged to deliver interesting informative output (tfidf-measure), learn from user feedback and optimize for conversation length as long-term rewards. As the different samples show, conversational black holes are evaded (for some while) resulting in one step towards more sustained dialogue.

Simulation with reinforcement learning models has been called upon multiple times in the context of dialogue (e.g. policy learning for dialogue management, Young et al. 2013) and now again it can be useful to apply simulation to evaluate the generation capacities of the models (Bordes NIPS 2016) and to let the models continue learning on-policy, even in different domains by using Gaussian processes (Gasic et al. 2016). With respect to the former, it is argued that human-machine communication needs to be evaluated by a feedback loop of simulation and synthetic tasks, which can break existing models or get solved, either way pushing progress in the field towards promising directions.

3.5.4. Incorporation of external knowledge into end-to-end systems

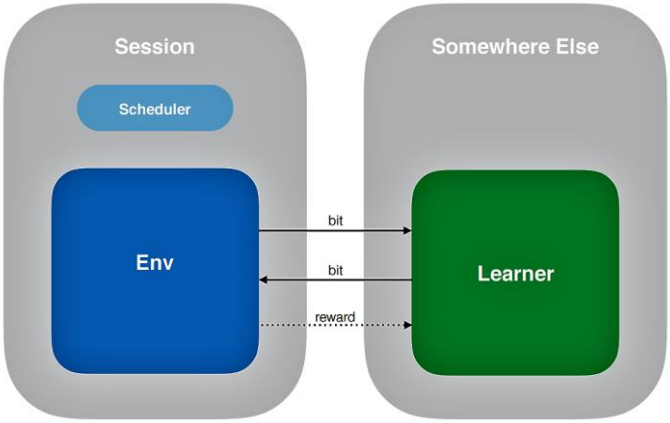
> "A core purpose of language is to **communicate knowledge**, and thus the ability to take advantage of knowledge (…) is important for [achieving] human-level language understanding [and generation]." (Ahn et al. 2016: 1, my emphasis)

With the exception of some in particular, all the formerly discussed models use no or weak supervision at best. Supervision in the form of symbolic rules, external knowledge and multimodal input can boost the performance and coverage of end-to-end dialogue systems. *Seq2seq* models have severe limitations in what they can encode or decode. For example, how to deal with out-of-vocabulary words in new input, which affect the robustness of the dialogue system. One opportunity that has been suggested already (cf. 3.5.2.1) is changing the architectures to allow for adding external memory resources (e.g. the introduced end-to-end MemNN, Suhkbataar 2015). By interacting with external knowledge sources domain-specific knowledge can be infused into neural networks, making them capable to support domain-specific tasks that require different vocabulary and dialogue structure. Another possibility to learn domain-specific knowledge or patterns is by transfer learning. For example, Yin et al. (2015) demonstrate the usefulness of external knowledge with a neural dialogue-like question-answering system which can query a knowledge base for answers by using a distributed representation of the query. Relatedly, Ahn et al. (2016) put forward a *Neural Knowledge Language Model* which is an RNN-based language model conditioned on knowledge graphs providing symbolic knowledge. Wen et al. (2016) propose a network-based approach to end-to-end trainable task-oriented dialogue systems, which seeks to aggregate useful information into the system's restaurant-suggestion responses by two supervised signals, dialogue history modelled by a belief tracker and the most probable search outcome within a database. Interaction with the real world via knowledge resources is a viable strategy towards building robust neural network dialogue systems which seek to understand language.

> Language is **grounded in sensory-motor experience**. Grounding connects concepts to the physical world enabling humans to acquire and use words and sentences in context. Currently most machines which process language are not grounded (…). [R]epresent[ing] words, utterances, and underlying concepts in terms of sensory-motor experiences [can] lead to richer levels of machine understanding. (Roy 2003:1, my emphasis)

End-to-end dialogue systems can undoubtedly benefit from richer input and grounding the conversational interactions in a multimodal environment. With respect to this, Das et al. (2016) present the task of *Visual Dialogue*, where a conversational agent must be able to hold meaningful conversation with humans about visual content. It is considered a test for measuring 'real' machine intelligence, provided that the system has to show that it can ground textual input in vision, infer context from dialogue history and generate appropriate responses. Grounding response generation in vision allows for objective evaluation with regard to individual system or model responses and provides a good competitive benchmark.

Very recently, virtual environments have been proposed with respect to the need for grounding communication-based AI systems (Mikolov et al. 2015). Douwe Kiela (2016) argues in favour of virtually embodying a conversational agent in an environment. Considering the current state of communication-based AI, a controllable virtual environment is more suited than the non-controllable real world for the moment to test the (language) reasoning capacities of various systems. Various open-source "Wizard-of-Oz" environments have been made available to the community (CommAI-env, Universe) to stimulate the development of communication-based AI and in which reinforcement learning strategies are used for learning from simulation. More specifically, agent "learn" by playing language-based reasoning games and interacting with the "complexly scripted" environment. Below an example is given of an agent in the CommAI-env environment successfully completing a "reasoning game" and receiving a bit+1-reward:



T: there is an apple in front of you. pick up the apple.
L: i pick up the apple.
T: there is no apple here.
L: i move forward.
T: you moved.
L: i pick up the apple.
T: you picked up the apple **[R+1].**

Figure 11: Left: high-level overview of the CommAI-env environment and how learning proceeds. Right: example in which a teacher 'T' plays a simple spatial reasoning game with the learner 'L'.

The proposed environments can certainly give good stimulus to open-source development of communication-based AI, because a lot of control is given to the designer of an algorithm/model on how to make it learn, i.e. the "learning curriculum". However, two points need to be made to nuance the laid-out advantages of these environments: first of all, it imposes a "heavy engineering burden on developers" (Lazaridou et al. 2016) and secondly, there is the question what the games or tasks really test. With regard to the latter, Gemma Boleda (NIPS 2016) offers critical remarks on the CommAI-env environment: On the one hand, the tasks are ill-defined (simple addition or subtraction operations) and are not appropriate measures of intelligence nor ability to do reasoning due to their simplicity in language, and on the other hand, who will teach the task definer (teacher) to come up with good tasks? Important care should be taken in the task definition and how to approach learning in a simulated environment.

4. Experiments

4.1. Motivation for dialogue representation

Conversational modelling is an indispensable task toward teaching machines how to make sense of and use language in the way humans do. It constitutes a major challenge, seeing that whatever constitutes a "valid" utterance/language exchange in a dialogue implies a very large search space, which at the same time is linguistically constrained on the semantic, syntactic and pragmatic level, and extra-linguistically on the social, cultural, conceptual and contextual level. The following conversations show that the search space of language is incredibly large:

1) A: Can you give me a hand with chopping the onions?
B: I could, but then again, you did not help me with the laundry
B: But alright, I will be the bigger man, I will help you cut the onions
A: Thanks...

2) A: Can you give me a hand?
B: I don't like giving handshakes...
A: You silly!

3) A: I need some help with preparing the food, would you mind?
B: I am rather busy at the moment, but maybe later...
A: Are you still mad for me not helping you out with the laundry?
B: No, no, never mind, I finished anyway... Shall I start with the onions then?
A: You bet ya

Highly related to the illustrated complexity of natural conversation are the questions we asked ourselves at the start of the project:

- What *existing language modelling techniques* could be leveraged to scale towards the conversational level?
- Does there exist a prototypical, universal "*dialogue unit*", comparable to syntactic units such as word, phrase, clause, sentence?
- Consequently, given a unit of dialogue, what *representation* is best suited and how to construct it automatically from conversational text data?
- Finally, given a chosen representation, how can we detect "*similarity*" between dialogues (e.g. examples 1 & 3)?

This three-month project will not claim to have found a "dialogue representation", which would constitute a fundamental finding - if even possible to acquire with more time and resources - but rather seeks to experiment with and review different 'unsupervised' machine learning methods which can give intuition as to what a "dialogue representation" could or should capture when applied on unstructured conversational text.

The first experiments conducted in the project have consisted of an exploration phase into existing unsupervised algorithms and what structure they model from existing dialogue files. With our collected dataset divided in different logfiles (cf. infra), we started using a naive approach of performing **document clustering** on a representative part of our dataset to see if we could find "similar" dialogues. Not only did this approach not scale well over the full set of dialogues, but clustering results were rather poor and did not really capture any semantic similarity between dialogues, rather only purely BOW-overlap.

This first exploratory experiment indicated that the used features could not capture the content of a conversation well enough, nor that a dialogue shares the same representation as a document.

On the word-level, neural network generated embeddings have been successful to capture semantic similarity in continuous vector space. Some work has been done on adapting existing embedding algorithms to capture larger textual units, **doc2vec** (Le, Mikolov 2014) and **skip-thought vectors** (Kiros et al. 2015). However, this approach has not been attempted (on its own!) due to the noisy character of the larger unit embeddings and no reported substantial breakthroughs in the literature (Hill et al. 2016b).

If the two former approaches are not very successful, what is being used in practice to model conversation and build dialogue systems?

An **intent** is a manually engineered descriptive script, which typically contains a question or a set of questions that should return a response (and/or action) by the dialogue system, which in turn is included in the intent.

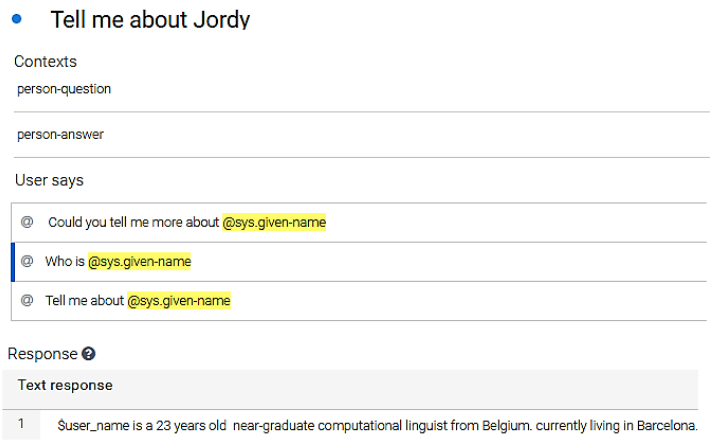


Figure 12: A very basic intent created on api.ai, an online service which allows one to build a text-based chatbot by manually constructing intents. In this case a set of possible questions is mapped to one deterministic response.

When considering the above intent template, the "dialogue unit" in question is more on the paired sentences level, and in the case of missing information - when the agent requires more information - interspersed by prompts and prompt answers.

What if we could train an unsupervised model on dialogue data to learn what input questions and output responses are frequently linked? This could in principle be leveraged to model intents in a principally data-driven fashion instead of manually engineering all contexts. For

example in a troubleshooting context, when a user asks for a password reset, most often a customer support agent will prompt for a user's e-mail or login information.
Companies hosting technical support platforms could certainly benefit from getting insight into their backlogged customer-agent data and frequently occurring issues ("FAQ" & popular answers). Whereas most attention is invested into open-domain intelligent assistants, building chatbots tailored to specific domains and enterprises (in-house data) can entail an even larger business opportunity.

For the dialogue modelling experiments we will draw on the recently proposed *seq2seq* framework. In line with what is discussed above, a *seq2seq* model can be used to map a sequence (input) to another sequence (output) and is end-to-end trainable on a dataset of conversations. A pair of sequences does not imply any limitation as to size or length, which for a "dialogue unit" is very appropriate. Moreover, it builds up a flexible, adaptable representation in the encoder's final hidden state, which abstracts over the domain knowledge present in the data.

The questions the experiments seek to answer are strongly tied to the questions asked in the beginning of this section. More specifically, the task consists of researching the potential of *seq2seq* learning for dialogue response generation and reviewing the ability to learn dialogue representations. By experimenting with a standard *seq2seq* architecture and an original dataset, we would like to analyse how the core algorithm behaves with respect to the unstructured data it is given, how the encoder responds to changes in the input and how this translates in decoding. With respect to the task at hand, the corresponding challenges have been identified:

- Data (pre)processing and encoding-decoding
- Network designer choices and empirical parameter tuning
- Evaluation of model performance
- Relevance in answer generation

The discussion of the experiments will proceed as follows:
First, we will introduce the dataset on which the *seq2seq* model will be trained. Consequently, we will clearly describe the neural network's design and characterize the methods used to build the different text-in, text-out end-to-end trained dialogue generation systems to make the experiments reproducible. The discussion proceeds with a quantitative and qualitative analysis of the results, the overall experimentation strategy (what went wrong/right along the way) will be reviewed in detail together with the degree as to which the challenges have been attempted. Finally, the conclusion will summarize the project's findings.

The set-out planning has been respected:

- ✓ Data collection
- ✓ Document clustering as an initial exploration of the data
- ✓ Data preprocessing
- ✓ GPU setup
- ✓ Construct three different dataset representations
- ✓ Encoded data as input for specific Seq2Seq implementation
- ✓ Trained three Seq2Seq models with varying input
- ✓ Build test set with representative dialogue situations
- ✓ Evaluate different system's generated responses on test set constrained input

4.2. Dataset: web-games troubleshooting

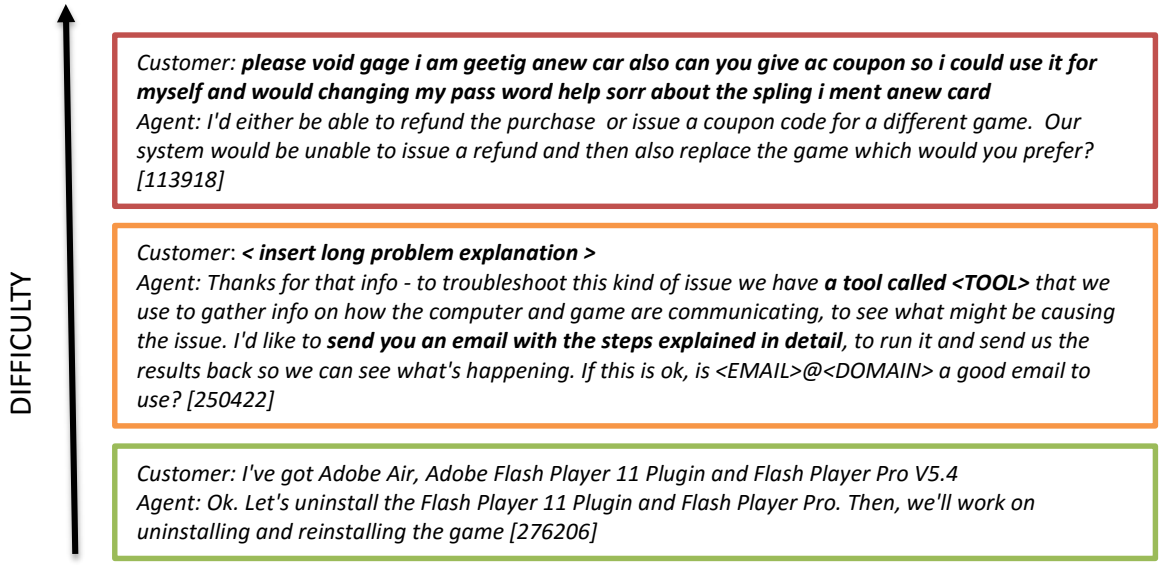
In our experiments we used a closed-domain, human-human conversational dataset from a troubleshooting customer support service hosted by a flash games website. The service reports on customers facing issues with quite generally their games, account, membership, monthly billing and promotional codes. An agent walks them through a solution by asking for more information related to the issue, asking to perform troubleshooting steps or passing on their company email for further correspondence. The full dataset consists of 21167 English chatlogs with 414205 utterances summing in total around 8M tokens. On average, a chatlog takes up 20 turns and is typically 370 words long. When comparing to other available conversational datasets (Serban et al. 2015) and those used in other *seq2seq* experiments (Vinyals & Le 2015, Shang et al 2015, Wen et al. 2015) the dataset used in our experiments is on the small-medium size scale.
The dataset is both representative for goal-oriented dialogue, the troubleshooting domain, and exhibits characteristics from non-goal-oriented dialogue, the chitchat domain. To support this claim, the following excerpts have been extracted from the data:

> *Customer: ok, now I understand.* ***I was confused because I've never done this before****. I thought you would give me a code number. Sorry for the confusion.* ***I had a blonde moment.*** *Thank you so much for your help. Have a great weekend*
> *Agent: Ok, no worries at all. Is there anything else I can do for you today in chat before you go?*
>
> *Customer: thank you <AGENT>. Have a great day.* ***It was actually nice to chat with someone instead of dealing with a computerized voice.***
> *Agent: It was my pleasure <CUSTOMER>. Is there anything else I can do for you today in chat?*

The first example shows that even human-human domain-specific instructed conversation is hard and "confusing", seeing that it requires some degree of domain knowledge from both participant sides. Next to that, the more in social context used figurative language, "a blonde moment", would be very difficult to model and be understood by a machine. In the second example, a customer utters his preference for human assistance (which he probably shares with other customers). This implies for human-machine interaction that the machine in question, a fully end-to-end trained dialogue system, must take the human's mental state into

account and not only try to imitate human, social-conversational behaviour from existing data, but try to emulate it without triggering the uncanny valley effect.

Moreover, the troubleshooting domain contains generally very difficult dialogue to model in an automated fashion (Williams 2007). Again, the dataset provides ample evidence to compose a modelling difficulty scale.



The top example presents ungrammatical, noisy language which without the full dialogue context is near impossible to understand even for humans. In the middle example, we illustrate that a lot of valuable information ideally to be learned by the end-to-end model is sourced out of the dataset, so there are also a lot of less informative, non-instructive mappings present. An easy mapping is exemplified in the lowest example, where there is a lot of shared content and even word overlap between the two utterances.

Additionally, an issue for *seq2seq* modelling is the presence of non-purely turn-taking interactions instead of a clear "duologue" structure.

> **"The longer the conversation the more difficult to automate it.** On one side of the spectrum are Short-Text Conversations (easier) where the goal is to create a single response to a single input. (…) Then there are long conversations (harder) where you **go through multiple turns and need to keep track of what has been said**. *Customer support conversations* are typically long conversational threads with multiple questions." (Britz WildML 2016, my emphasis)

The below example illustrates the complex non-sequential structure of the dialogues:

> Agent: Hi, my name is <NAME>. How may I help you?
> Customer: I have a new computer running Windows 8. **[**Download the last game I purchased and there is no sound.**]**^{Q1} **[**Are your games compatible with Windows 8?**]**^{Q2}
> Agent: **[**Hi <NAME>, I am really sorry to hear that.**]**^{PROMPT}
> Agent: **[**I can tell you that most of our games are definitely compatible with Win8**]**^{A2}

Agent: **[**Can you let me know the game <NAME>?**]**[A1]
Customer: <GAME-TITLE>

There is evidence of "intent" structure as identified by the question and answer tags, but it is also clear that these are not one-on-one sequential mappings. In addition to multiple questions within one customer turn, the agent takes several turns to respond to the different questions, with a non-purposeful prompt sabotaging any meaningful sequential mapping between the "real" questions and "real" answers.
All the modelling complexity and issues with sequentiality discussed above will be taken into account and have either been dealt with in the preprocessing stage or by defining auxiliary methods.

In order to have the dataset ready for processing, a severe clean-up and formatting was performed. All identification tags and utterance logging timestamps were removed first. We treated consecutive utterances belonging to the same participant to all be part of one turn. The model is trained to predict the next sentence given the previous one, so turn-taking should be strictly respected. A "brute-force merge" strategy has been adopted here, but research into more intelligent strategies for utterance splitting-combining is advised.
One unpreprocessed training set has been kept aside for the experiments. The other training set received a more specific preprocessing treatment. We applied html-tag removal, stripped punctuation except for sentence boundary markers, chose not to apply sentence tokenization as it would negate the utterance-merging, lowercased all words and introduced a meta-character '+++$+++' to identify the difference between speaker and utterance.

4.3. Methods

4.3.1. LSTM encoder-decoder architecture

For the experiments we consider the *seq2seq* framework originally described in (Sutskever et al. 2014). It should be underlined that we use a completely data-driven architecture, which can be easily trained end-to-end using unstructured data without requiring any human annotation, scripting, or automatic parsing (Sordoni et al. 2015: 197). The standard encoder-decoder architecture makes use of a recurrent neural network (RNN) for encoding an input sequence token-per-token into a context vector representation, and another recurrent neural network which functions as the decoder, mapping the context vector representation to a probability distribution over the next possible tokens. In general we follow the approach from the Neural Conversational Model (Vinyals & Le 2015), where an application of the Long Short-Term Memory architecture (LSTM, Hochreiter et al. 1997) is used instead of RNNs as encoder-decoder components. Arguably, an LSTM is more successful at learning on data with "long

range temporal dependencies" (Sutskever et al. 2014: 2) than a standard RNN (Graves 2013).

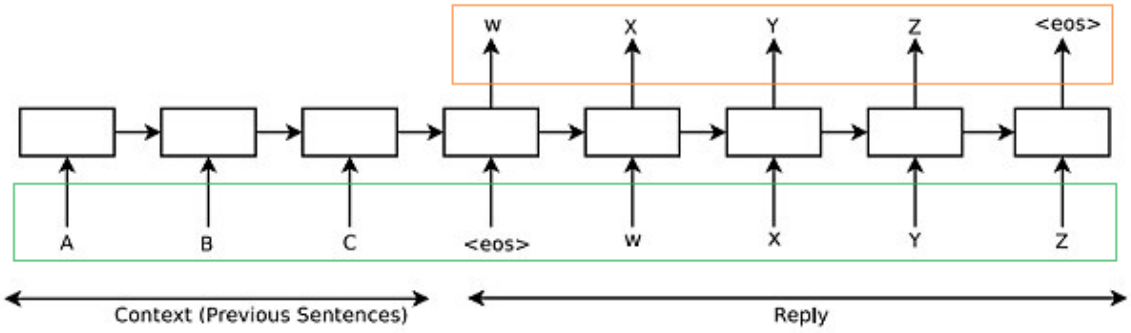


Figure 13: (in green below) the input sequence for the LSTM encoder, (in orange above) the output sequence from the LSTM decoder. The model needs to learn how to predict "WXYZ" from "ABC" context.

When applied to our dataset, the input sequence is a concatenation of agent-customer utterances ("abc<eos>wxyz") separated by an end-of-sequence (EOS) token. The hidden state of the model when receiving the EOS token constitutes the thought vector, which stores the information of the processed context sentence. Effectively, the model is encouraged to learn semantic representations of sequences. During training to maximize cross entropy, the model learns a probabilistic mapping, a ground root based on the training data, which it can use to produce answers. The model's ability to converse arises from its training by predicting the next dialogue turn in a given conversational context using the maximum-likelihood estimation (MLE) objective function. In the inference stage the model takes a "greedy" approach by simply feeding predicted output tokens as input to predict the next output token until an EOS token is generated (Vinyals & Le 2015: 2).

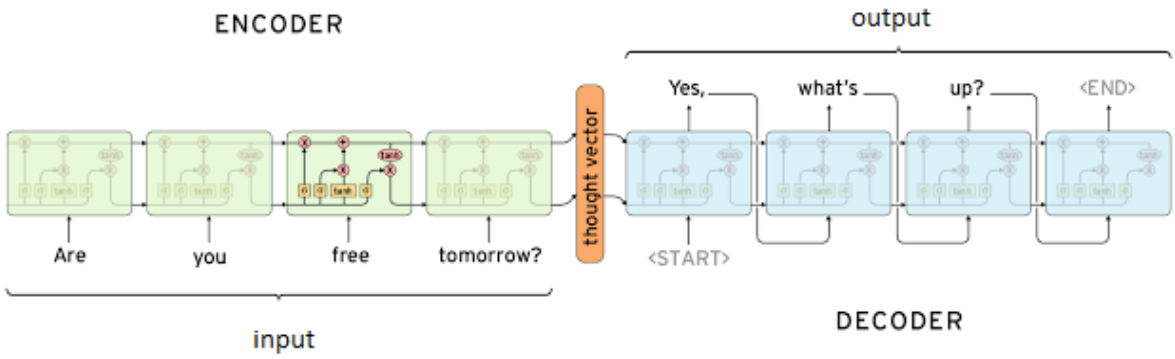


Figure 14: Here we see the greedy inference at work in the decoder component, where using a softmax function the output tokens are sequentially predicted until an EOS token is reached, thus allowing the model to define a distribution over sequences of any possible length. Original picture adapted slightly.

More specifically, the model we have built is based on an implementation of "A Neural Conversational Model" (Vinyals & Le 2015) in Torch. The implementation closely mimics the architecture employed in the IT helpdesk Troubleshooting experiments (Vinyals & Le 2015: 3), but applies it to the Cornell Movie-Dialogs Corpus (Danescu-Niculescu-Mizil & Lee 2011). Adaptations have been made where needed to account for the structure of the dataset used in our experiments. The main architectural features of the implementation are as follows:

Both the encoder and decoder are each a one-layer FastLSTM with the same parametrization and *hyperbolic tangent* transfer functions. Given the size of our dataset, adding more layers would only make learning more complex with marginal gains. In this regard, we stick to the one layer for encoder and one for the decoder as in (Vinyals & Le 2015)'s first experiment. The output layer of the decoder has a *logsoftmax* transfer function in order to generate a logged probability distribution. The designer choices are plenty as defined below and exemplified with the training setup and parameters of our choice (indicated with coloured box) for the experiments:

'dataset'	0	'approximate size of dataset to use (0 = all)'
'maxVocabSize'	0	'max number of words in the vocab (0 = no limit)'
'cuda'	True	'use CUDA toolkit for training'
'hiddenSize'	500	'number of hidden units in LSTM'
'learningRate'	0.001	'learning rate at t=0'
'gradientClipping'	5	'clip gradients at this value'
'momentum'	0.9	'momentum term'
'minLR'	0.00001	'minimum learning rate'
'saturateEpoch'	20	'epoch at which linear decayed LR will reach minLR'
'maxEpoch'	30	'maximum number of epochs to run'
'batchSize'	10	'mini-batch size'
'EncodeLength'	60	'Maximal Length of input sequence to Encode'
'DecodeLength'	60	'Maximal Length of output sequence to Decode'

The motivation for the design choices comes from experimentation with the parameters. For instance, given the small-medium size of the dataset, we have used the full dataset for training (more details cf. 4.4.2.). The setting of 500 hidden units for each LSTM has been deemed sufficient, increasing the size of the hidden layer did not render "noticeable" difference in generated responses. Two settings that have been importantly changed are the maximal length of the to be encoded and decoded sequences. Respectively, the standard settings were 25 and 20 tokens, which were fine-tuned to the dataset characteristics, roughly two times the average utterance length to allow for some margin.

In order to scale towards modelling sequences with the above setup and given the size of the dataset and the network, a Graphical Processing Unit is a favourable prerequisite. The machine used for the experiments has 2 Nvidia Titan Black GPU's with 6 GB RAM, 2880 cores and based on a Kepler microarchitecture.

4.3.2. *Custom seq2vec* Representation

As illustrated in section 4.2. natural conversation does not follow a one-on-one sequential mapping (also Baroni NIPS 2016). The dialogues within the data exhibit a high-grade of non-strictly purposeful "chitchat" which might confuse the *seq2seq* learner when presented with customer-agent utterance pairs. In order to account for the complexity in dialogue structure a method needed to be defined which would be able to identify what are the content questions and what are the contextual answers. The method should be capable of generating an "idealised dialogue representation", which is in nature closer to the in practice used

"intents", but also allows some degree of principled "similarity" comparison between dialogues.
This sparked the idea of a *custom sequence to continuous vector representation* (custom *seq2vec*). In short, we build a continuous space vector representation for all utterances, apply clustering on the utterances from the dialogue participant showing the most regularity, ground the clustering on that participant's utterance representation and use either the original dataset indexing with manual post-pruning or a "maximum cosine similarity dialogue-bounded objective" to recover sequence pairs, now collected together in clusters of approximately similar sequence pairs.
The strategy of building a "custom" sequence representation in continuous vector space requires elaboration, given that we formerly repudiated the use of any word2vec extension towards sentence and document representations (Le & Mikolov 2014).

4.3.2.1. Custom *word2vec* model

The first problem we addressed is how to have a representation for all the words in the dataset, while also drawing upon an already established, general-purpose word embedding model, the Google News Corpus (Mikolov et al. 2013). Eventually, we would like to combine the predictions from different word embedding models and use this effective mixture to build an utterance representation (i.e. in the sense of "an uninterrupted chain of [spoken or] written language", Oxford English Dictionary).
We created word embeddings using the "Gensim library" (Rehurek & Sojka 2010) for the dataset by preprocessing and tokenizing into words, which have been fed as input to a continuous skip-gram architecture to be trained with negative sampling method (Mikolov et al. 2013). Seeing that the results of word2vec training are strongly affected by the parametrization of the model, we have done extensive manual assessment of the quality of different parametrizations. As a proxy for model quality, we printed the 10 most similar words (as measured by cosine similarity) for the 10 most frequent 'meaningful' words in the dataset [*game, help, email, account, name, thanks, code, chat, click, coupon*] and additionally performed clustering on the embeddings for which the first 10 clusters containing first 10 tokens have also been printed. Alternatively, interactively visualising the embeddings with the TensorBoard Embedding Projector (TensorFlow Development team Dec 2016) could additionally have helped in indicating the general model's quality. The quality assessment is suited for the purpose at hand, but embeddings for general semantic modelling purpose should receive a more standardized evaluation treatment (Schnabel et al. 2015).

Setting the skip-gram model's parameters to a dimensionality of 1000, restricting the words in the vocabulary to having a minimum of 50 occurrences, with a context window of 25 and a standard frequent words' downsampling rate of 0.001 resulted in (arguably) the best overall model. The choice of the skip-gram architecture dubs from its better capacity at modelling infrequent words despite being generally slower than the continuous bag-of-words architecture. The dataset contains a large amount of specific 'local' entities such as web-game

titles that we would like to take into account by exploiting the skip-gram's ability. In line with the previous, we do restrict the vocabulary to only account for words with a minimum word count of 50, which most of the aforementioned local entities will have. However, to make the skip-gram model work a larger context window value is needed. Therefore, setting the context window to 25 forces the word2vec model to take a larger bounded focus, correlating with the average utterance length in the dataset. In general, increasing the dimensionality of a word vector should increase the overall quality, given enough data. The dimensionality of the vectors is admittedly very high, yet after manual inspection the quality of both the max-similarity lists and clusters, all be the latter the result of a non-deterministic process allowing a certain degree of randomness, were deemed the best in joint assessment. As per illustration of the former claim, a cluster and a most similar list from the model results are given below:

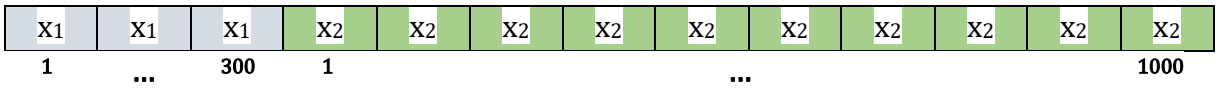
```
Cluster 3
[mouse, custom, area, windowed, moving, drag, move, cursor, capture]

code
[(coupon, 0.6856838464736938), (codes, 0.4115700125694275), (standard,
0.40007632970809937), (cart, 0.3469886779785156), (enter, 0.3424854874610901),
(apply, 0.3375224471092224), (paste, 0.3262205421924591), (expires,
0.3194829523563385), (replacement, 0.3133856952190399), (buy, 0.31181472539901733)]
```

4.3.2.2. Concatenated composition technique

Having built the word embedding model for the dataset (1000-dimensional vectors) and collected pre-trained (300-dimensional) vectors trained on part of the Google News corpus (3 billion running words), an effective "ensemble" approach or hybrid representation needs to be constructed. Since the vectors of both distributional models have different dimensionality, which means the original spaces cannot be combined, no further retraining on the vectors is attempted. In fact, we would like a distributional representation which takes both models contributions into account. We follow a similar approach to (Garten et al. 2015), where the *concatenated composition technique* is proposed and preferred over a normalized sum of component spaces. For each word the composition supplies a normalization into a new single vector:

$$1) \quad X_{new} = X1 \frown X2$$



Direct concatenation allows the advantage of "allowing vectors with different underlying ranks" (Garten et al. 2015: 97). This implies that a word does not necessarily have to have the same number of non-zero rows, and most importantly, that a word does not need to have a distributed representation in both models. Moreover, the produced hybrid representation allows easy usage of cosine similarity in the concatenated space, because it is "determined by a linear combination of the dot products of the component vectors" (Garten et al. 2015: 99).

4.3.2.3. Vector averaging method

In the next step the concatenated representation is extended towards the modelling of sequences of words. Here we opted for the *vector averaging method* (Socher et al. 2013), where an utterance vector is represented by the average of the values for the word vectors constituting the whole sequence. More advanced methods for building compositional embeddings (Socher et al. 2013, Le & Mikolov 2014, Hill et al. 2016b) could have been attempted, but as will be explained, the quality of the compositional embeddings is not the ultimate goal of the project, more a means to an end.
In continuation, a slight adaptation has been made with regards to out-of-vocabulary words for either of the models. In all cases, a word is represented by a concatenated vector with a dimensionality of 1300, of which the first 300 values correspond to the Google News corpus vectors and the remaining 1000 values to the custom-built model's vectors. Even so, when a token does not have a corresponding vector in one of the models, the part of the concatenated vector corresponding to the model with the OOV-token will be padded with zeroes. It is hypothesized that this will not affect the end result in averaging, nor have an effect on cosine similarity given the inherent normalisation of the dot products.

4.3.2.4. Mini-Batch K-Means Clustering

Finally, the *custom seq2vec* representation will be put to use by applying it on the utterances of the dataset. One particular characteristic of the dataset will be exploited, the *frequency bias* present in the agent's utterances (cf. section 4.2.). What has been observed, is that the agent uses consistently the same formulations when addressing similar problems or questions from the customer. The frequency bias cuts both ways, both positively by responding in a similar fashion to frequently-occurring issues, and negatively by answering content-rich customer questions with formulaic, non-content bearing prompts. By means of building clusters on the basis of the *custom seq2vec* representation of agent utterances, we would like to obtain clusters containing similar agent utterances, which paired with their respective customer utterance at the agent's index -1 will generate a dialogue-topical structure of the dataset.

For the clustering we applied MiniBatch K-Means to reduce training time and initialized *k* cluster centres with kmeans++ (Arthur & Vassilvistkii 2007), a careful seeding method which spreads out initial cluster centres yielding improvement in performance and overall convergence speed, although extra computation time is needed for the initial selection. Together with a colleague at Oracle we manually compared different settings of *k [50, 100, 200, 350, 500, 750, 1000, 1500, 2000, 2500]* clusters, although silhouette analysis (Rousseeuw 1987) might also have been useful, yet was not attempted. Eventually, a reasonable amount of 500 clusters has been selected, which showed overall the best balance between capturing topical context and more local structures.

4.3.2.5. Structural cluster post-pruning

However as conceded, not all clusters will contain content-rich utterance pairs, but it offers an ideal situation to perform “structured” manual post-pruning on the clustered dataset in order to purify the data of rubbish and generic utterance pairs. Below we present some of the resulting clusters exemplifying respectively bad and good examples (in preprocessed format):

As a guide the tags used in the utterances are collected together with their denotation and a fictional example:

<GAME>: title of a specific hosted web-game (+optional id), *PAC-MAN adventures (2)*

<EVENT>: a seasonal event causing promo codes to be shared, *Thanksgiving*

<NAME>: a first name, either from the agent or the customer, *John*

<AGENT>: a moderator name which the agent uses to refer to him/herself, *John_Doe*

<MEMBERSHIP>: various types of membership exist a with varying identifiers, *regular*

<TITLE>: a charge statement from the company hosting the web games, *Company #001*

<TAG>: a specific billing code for the company hosting the web games, *x035B8$6A*

<MAIL>: personal username for email, *john.doe@<DOMAIN>*

<DOMAIN>: custom domains can be used for email correspondence, *gmail*/*oracle.com*

<PROMO>: promotional item given away to loyal members, *game character toy*

Customer: i am trying to buy the <GAME> today and use the <EVENT> code . it is not allowing me to neither use the code or buy what is wrong ?
Agent: hi <NAME> **i ' m sorry to hear that** let me check this just a minute please .

Customer: im really annoyed
Agent: hi <NAME> **i ' m sorry to hear that** what happened?

Customer: I have purchased <GAME> after playing the hour trial i cannot get it to play now it freezes on the install screen
Agent: hi <NAME> **i am really sorry to hear that** .

Example 3: Cluster 227
One of the most frequent formulations by the agent as a generic response to most questions is "I am sorry to hear that". Clusters containing purely these as agent responses need to be structurally pruned.

Customer: hi i have a question about <MEMBERSHIP> .
Agent: **sure thing**

Customer: okay i will look said did not install correctly . had to click on something and it is trying again . so we wait .
Agent: **sure thing**

Customer: i have called the bank and they have posted correctly . i would like to go and purchase game manually
Agent: **sure thing**

Customer: i will try that in just a sec
Agent: **sure thing . goodbye**

Example 4: Cluster 38
The Agent's generic response "sure thing" does not make for valuable utterance pairs. Here the non-sequential nature of the content-bearing utterances is obvious by the lack thereof by relying purely on the indexing strategy.

Customer: you too **bye** .
Agent: **# 676747 hi** my name is <AGENT> . how may i help you ?

Customer: **thanks** for the nice words
Agent: **# 673753** hi my name is <AGENT> . how may i help you ?

Agent: take care and happy gaming .
Agent: **# 663474** hi my name is <AGENT> . how may i help you ?

Example 2: Cluster 36
Due to the indexing crossing chatlog file boundaries, utterances are mapped together which are from different conversations with possibly different participants. Another problem is the agent-agent pairing based on the indexing. However, both are now easily prunable.

Customer: yes
Agent: can you give me the charge details ? charges from our site should be something like <TITLE> # # # # # # # # or <TAG> # # # # # # # # #

Example 1: Cluster 188
A full cluster containing only one utterance pair. This is a sign of a very bad seed chosen at the initialization, probably caused by the presence of many hashtag signs, which in the preprocessing phase were not removed to allow for the chatlog- identification tag. <TITLE> is the name of the hosting web-game hosting platform and <TAG> is an abbreviation of the former name.

Customer: i have a few accounts with a lot of pre purchased games but i am unable to access the email accounts to reset the password for the game accounts .
Agent: i ' m sorry to hear that and i can certainly help you with this . which emails are those accounts under ?

Customer: i am having a problem trying to make a purchase it will not take my coupon code i have been trying since this morning
Agent: i ' m sorry to hear that . i will definitely help with this . which coupon code are you trying to use ?

Example 5: Cluster 143
The example illustrates that the agent starts the utterance with the generic formulation and then proceeds to prompt a meaningful question to the customer. Seeing that it does contain some information, clusters containing these have not been pruned if the prompts were generally content-rich enough.

Customer: the game <GAME> when i **try to play it** i have no mouse to search for the items do you know why
Agent: no i dont . have you **tried uninstalling the game and reinstalling** it ? Cluster 63

Customer: i did **download a trial** game this morning
Agent: does it let you **run a full game download** this time ? Cluster 173

Customer: thank you how will the **money be refunded**
Agent: it will be **refunded on your visa card** . Cluster 294

Customer: i keep getting an **error** 5001 9
Agent: that error usually occurs when there isn ' t enough **bandwidth** in order for games to be able to stream properly . do you have **any other devices or programs that are streaming or using bandwidth** ? Cluster 373

Customer: your online game is ready **play now** and a diiferent play now
Agent: ok . **when you click the play now** next to your online game is ready does it bring up the game ? Cluster 483

Example 6:
Illustrative examples of good clusters containing similarly formulated agent responses to related customer questions and/or problems.

Inevitably, the results of manually post-pruning unwanted clusters brings along a reduction of the dataset size. Before pruning, there were 483 clusters taking up total of 324371 utterances. Pruning resulted in a relative cluster set reduction of 18% with 397 remaining clusters and a relative utterance set reduction of 21% with 256963 remaining utterances. A possibly positive sign is that before pruning a cluster contained on average 336 utterance pairs, whereas after pruning only 324 paired utterance are recorded. This observation implies that the pruned-away clusters were more than average in size, which is plausible for clusters containing highly similar and frequent rubbish. The envisioned "positive" effect of the manual post-pruning, despite a dataset size decrease, needs to be verified when applied as training input to the *seq2seq* learner.

4.3.2.6. *Maximum Cosine Similarity* objective

Another, yet comparable, strategy is more experimental and seeks to identify the most semantically similar customer utterance to each agent utterance, instead of retrieving the customer utterance based on the original indexing of the data. Having built exactly the representation needed to do so, utterance pairs are formed by mapping for each agent utterance (a_j) the corresponding customer utterance(c_i) for which the *custom seq2vec* representation has the *maximum cosine similarity score* when comparing all the possible preceding, yet bounded to the same conversation, customer utterances (C).

$$\tilde{c} = arg\max_{c_i \in C} cos(a_j, c_i), \text{ where } C = \{c_i | 0 \leq i < j\}$$

The possible advantage of building utterance pairs based on *maximum cosine similarity* would be more utterance pairs that are more meaningful within the same clusters. If *customseq2vec* is a good representation, the similarity method should perform better than a naïve indexing-based retrieval of corresponding customer utterances. Initial inspection of the method's results shows that there are plenty of differences between utterance pairings from both approaches. For some formerly "bad" index-1_based mappings *maximum cosine similarity* results in remarkably better pairings which can jump over several possible closer candidates:

Max Cosine Similarity: 0.82289739
Best Idx: 238095 **Our Idx**: 238100
Most Similar: the message comes in the game itself . it then says press the ok button to start over
Previous: yes
Current: ok . so does the message come up before the timer actually runs out ?

Max Cosine Similarity: 0.86755512
Best Idx: 295143 **Our Idx**: 295150
Most Similar: yes i purchased a game on a cd an it doesn ' t work . can you tell me why ?
Previous: i just put it in an nothing shows up on the screen .
Current: ah right i see so when you put the disc into your pc the game simply doesn ' t open ?

The first example is a clear "win" for the less naïve approach, however for the second, it could be argued that the mapping skips a possibly good index-based candidate in favour of an earlier less-informed customer utterance.
Basing utterance pairings on *maximum cosine similarity* also produces different clusters. The

results of the latter approach translated on the cluster level is illustrated with a representative cluster exhibiting a promisingly high grade of coherence and reciprocal utterance pair similarity.

Cluster 56 [specific game orders]

Customer: jts the <GAME1> one
Agent: ok . the three games listed on your account all activated are <GAME1> <SEQUEL_GAME1> <GAME2> and <GAME3>

Customer: yes . are you able to see what the new games ordered are ?
Agent : i see that on your <MAIL>account you have 3 games <GAME1>, <GAME2> , <GAME3>

Customer : <AGENT> i donot see <GAME_misspelled> any more
Agent : are you refering to <GAME> ?

Customer: thats why i want you to look because i cant remember for sure because i also bought some for the pc yesterday
Agent: ok i ' m seeing the purchase of <GAME1> <GAME2> and <GAME3> for the mac does that sound right ?

Although we make use of an admittedly noisy representation, the *custom seq2vec* representation is extremely useful to identify similarity between sequences even when the word overlap is low. For example, the third utterance pair even makes the mapping between a misspelled game title and the title to which the agent believes the customer refers. The other examples demonstrate possibly the effect of the custom-built dataset-specific word2vec model on cosine similarity where 'local' entity embeddings such as the game titles are taken into account in the averaged sequence representation. The pre-trained general-purpose Google News corpus vectors will also influence cosine similarity for recognizing more globally topical-functional similarity. The latter claim is exemplified by the "orders", "listed" and "purchase-buy" relations in the utterance pairs.
Again, inputting the resulting data into the *seq2seq* learner might give a better impression of the advantage the *maximum cosine similarity* approach could entail.

4.4. Models

Two alternate, yet analogous strategies have been defined in the previous section to deal with and either eliminate or bypass the *frequency bias*. Both of these approaches have been translated into model inputs for the *seq2seq* architecture discussed in section 4.3.1. All three models share the same architecture and parametrization in order to ensure principled comparability on the side of encoding and decoding. Before starting the discussion, an overview and characterisation will be given of the three models which are to be compared and evaluated.

4.4.1. Model Descriptions

As a rough baseline model (**base seq2seq**), we consider a *seq2seq* architecture with the unpreprocessed raw dialogues as the training data divided into 21167 text files.

The second model (**index seq2seq**) takes a more complexly constructed training dataset, which has been built from applying the *custom seq2vec* representation on the agent's preprocessed utterances, grounding the clustering on these embeddings and using *the index-based pair retrieval* approach to create utterance pairs which will constitute these clusters that have finally been manually post-pruned to address the *frequency bias* issue.

For the third model (**cossim seq2seq**) the training set has been constructed in similar fashion to model 2 except for the last two steps, which in this case will re-use the first representational step from the second model on the customer utterances with the goal of creating most similar agent-customer utterances pairs following the *maximum cosine similarity* approach. With the motivation of comparing the advantages or disadvantages between the approaches in the two non-baseline models, no manual post-pruning of the clusters has been utilised.

4.4.2. Training setup

The below table gives an overview of the training setup deduced from each model's characteristic training dataset.

	training data size	# of logs / cluster	vocabulary	time per epoch
base seq2seq	292357	21167 logs	39089	6.5 hours
index seq2seq	256961	397 clusters	42844	6.5 hours
cossim seq2seq	282042	489 clusters	27499	6 hours

The *base seq2seq* model obviously has the highest number of utterances, whereas the *index seq2seq* has the least due to the manual post-pruning of the clusters. The irregular size of clusters and examples in the last two models proceeds from both the *custom seq2vec representation* filtering out utterances without a good vector representation and the clustering collapsing some clusters which have been initialised with bad seeds. What takes the attention is that the base model has a lower vocabulary size than the pruned model. This might be the result of the implementation's tokenizer on the unpreprocessed data. The lowest vocabulary size proceeds from the *cossim seq2seq* model's pairing approach, because of which some losing candidate customer utterances containing specific vocabulary might not be taken into the training set. Overall, training each models takes approximately as much time, around 8 days in total for completing the 30 epochs.

4.4.3. Hypotheses

The model *index seq2seq* is hypothesized to generate less generic responses, arising from the genericity-suppression and noise-reduction strategy implied by manually post-pruning clusters of *index-based* utterance pairs. A hypothesized disadvantage would be that a lot of useful to be encoded information from the customer utterances are lost due to too strong pruning or the clusters being based solely on the agent's utterances *custom seq2vec* representation.

The *cossim seq2seq* model offers a theorised improvement in that the training data will consist of many more content-bearing utterance pairs, which will aid the *seq2seq* decoder with generating content-rich answers to new unseen input, provided it is similar enough to patterns seen in the training data. The success of this approach depends highly on the *custom seq2vec* representation, which might be very noisy (discarding a lot of valuable syntactic information), and thus building utterance pairs on the basis of *maximum cosine similarity* might match semantically or pragmatically dissimilar utterances or take apart utterance pairs that were initially good when relying purely on indexing. We hope to see the third model's strategy be more effective in evading the frequency bias than the approach of manually pruning and simple index-retrieval.

4.4.4. Evaluation setup

The feasibility of each of the approaches and quality of the generated responses by the individual models will be evaluated and considered in unison in the discussion. Seeing that the experiments primarily focus on improving inter-sequence quality of utterance pairs, the evaluation will also be executed on the turn-level, i.e. the lowest level of the evaluation hierarchy introduced in section 2.4. The evaluation method of choice is ***ground truth response and pairwise model comparison*** in line with (Li et al. 2016a, Wen et al. 2016, Serban et al. 2016b). We have selected the option for testing with more constrained test input and consistently comparing learned mappings and generated responses between models. More specifically, we would like to experimentally observe if and how after training the individual *seq2seq* models build up meaningful mappings of known frequently occurring pairs. We hope that the learned models can abstract and generalize over the data well enough to offer at least some decent responses to test examples from highly frequent topics in the dataset. In the worst case, due to the frequency bias in the data, the maximal likelihood objective compresses towards generic answers too much and none of the approaches significantly affect the response generation. If there is enough variation between the models' test outputs, we will analyse where exactly the reason for the variation can be found.

An evaluation test set of 60 representative, content-rich customer utterances have been manually sampled from the original data with their corresponding ground root next agent utterance. The topics pertaining to these test samples vary widely from game manager issues,

reinstallation, save data fixes, purchase refunding, membership cancellation, thanking for help, specific game assistance, password reset, entering discount coupons, error codes, OS incompatibility, server disconnecting, audio/graphics issues to signing in problems. We would like to perform a targeted test on those topics, test the individual models and see what they have "learned" and if it is what was expected when comparing with the ground root.

The use of 'test' samples from the same distribution as which has been trained on could be regarded a methodological flaw, but it can be argued that in this case we are more interested in evaluating the training process itself (learned mappings) and not about testing the resulting models on unseen samples taken from another distribution. It is very probable that some test samples do not occur in the training sets for some of the individual models. For example, in *index seq2seq* the specific test instances might have been pruned away and in *cossim seq2seq* the test utterance might have been paired with another next agent utterance than the original ground root or not even appear in the training set at all.

The relevance of the generated responses to the specific test topics will be manually evaluated with a fuzzy binary scale (relevant, non-generic [1], possibly relevant or well-placed generic [2], irrelevant or generic misplaced [3]) and compared for each of the models. In order to ensure representativeness of the results, the human evaluators have been given the models' test output with context utterance and ground root, yet deliberately it was not identified which model had which output. Instead of measuring inter-rater agreement with Cohen's Kappa, the fuzzy binary scores have been averaged over the individual evaluators. In the discussion, specific test-output samples will be used to qualitatively illustrate possible divergences between the models which can be explained by the approaches taken in each of them. No automated metrics have been used here to compare model output with the ground root given that these do not correlate well with human judgement (Liu et al. 2016). Another idea (yet no time to implement and proprietary data issues) was to perform third party user simulation where the system would output responses for both *index seq2seq* and *cossim seq2seq* and the user could choose which model was best for each unconstrained input's response generation turn.

5. Discussion

5.1. Results of evaluation

In this part we consider the results from evaluating the models. First, we will have a look at quantitative results, compare and choose an overall best-performing model. Finally, we will zoom in on specific generated test model outputs to get a better grasp of what right/wrong in the individual models. Finally, the hypotheses for *index seq2seq* and *cossim seq2seq* will be discussed in the light of the evaluation results.

5.1.1. Quantitative results

Overall, the results over the three (time-restraint) individual evaluators were very consistent. The general trends identified were reflected in the individual, absolute scores as well as in the average score. Below, a table collects the fuzzy-binary scale scores which put numbers to the test input relevancy of each model's generated responses:

	relevant	quite relevant	irrelevant	
	1	**2**	**3**	**average score**
base seq2seq	23.33%	34.44%	42.22%	2.189
index seq2seq	**34.44%**	31.67%	33.89%	**1.994**
cossim seq2seq	8.89%	24.44%	67.22%	2.594
	22.18%	*30.13%*	*47.69%*	*2.1786*

Table 2: A table collecting the relative frequency of the fuzzy binary scores over the manual evaluations. The average score should be interpreted as performing better, the closer to 1.

The results show that over the 60 test inputs, the *index seq2seq* model generates the most relevant results, and most importantly, does it consistently better than the *base seq2seq* model. More relatively, for each model the point of mass is different, the *base seq2seq* has well-distributed scores that do learn towards more generic and irrelevant responses; the *index seq2seq* has quite evenly distributed scores which suggests that although genericity has been suppressed and more relevant [1] responses have been generated than the baseline, the genericity effect is not fully eliminated. The results for the *cossim seq2seq* model are not representative of what the model can answer. In this case, genericity is not the real problem, rather that in almost half (27) of the cases the model generates customer utterances to the customer test input instead of the to be expected agent answers. Possibly, the processing and tokenizing of the utterances in the training dataset has gone awry, which inevitably makes the quantitative results rather useless. In order to better analyse the effect of genericity on relevance, the three most common, similarly formulated generated responses over the models are collected.

	base seq2seq	*index seq2seq*	*cossim seq2seq*
I'm sorry to hear that	22	19	6
I can definitely help out with that.	11	12	12
Ok	6	8	9
	39	39	27 (+27)

Table 3: The three most common responses over all three models.
Additionally, *cossim seq2seq* has 27 non-relevant customer utterance responses which should be taken into account as well as to not consider it better at handling the genericity issue, on which no representative claims can be made here.

An important observation can be made here that *base seq2seq* and *index seq2seq* do no vary In absolute numbers of generic responses. However, given a more closer look at the test results, only in 18 out of 39 cases does *index seq2seq* generate a similarly bad generic response (analysed as bad if one of the evaluators has given a [3]-score to the *index seq2seq* response) like *base seq2seq*. In the remaining 11 cases, the generic response generated by the *index seq2seq* is deemed a relevant answering strategy. With regard to this, we can state that although generic formulations in the training data have not fully been removed by the manual post-pruning strategy, the *index seq2seq* model does improve in relevancy of generic answers over the baseline. The faulty processing of the *cossim seq2seq* training dataset leads to the insight that a hard constraint as to which participant (customer/agent) is linked to which utterances would be more favourable for learning the mappings. Another advantage of this would be keeping the persona of the *seq2seq* learner steady in the spirit of (Li et al. 2016c), whereas now changing the test input in the other models would generate agent responses possibly mixed with some customer utterances.
The quantitative results already demonstrate general trends in the results, but a qualitative discussion can offer a more fine-grained perspective as to what the individual models have learned and what (type of) test input presents issues for the *seq2seq* learners. As a final takeaway, a lot of research remains even on the local structure level to make *seq2seq* learners practically feasible response generation systems.

5.1.2. Qualitative results

In order to get a grasp on the test results representative samples and model output will be presented. The focus will be on understanding what the *seq2seq* models have learned in the mappings and how this is evident from the samples and generated responses.

We will begin with the ***base seq2seq*** model to state some basic example output and how (good/bad) the generated responses compare to the ground root. To be clear, each test-output sample starts with the context input utterance, followed by the model response in bold and the ground root response in italics.

[example 1: good]

> i ' ve done that several times but it doesn ' t install ? it takes me to my game manager
> it ' s not on the game manager .
> **Ok, so you're trying to download the game again?.**
> *hm that ' s strange . let me just have a closer look . so it just opens the game manager but you can only see other games here ?*

Example 1 illustrates a very good and relevant generated response, which stays in the game download/install topic. Compared to the ground root, the response still remains a viable communication strategy.

[example 2: generic and bad]

my email is <EMAIL>@<DOMAIN>.com
I'm sorry to hear that.
ok i ' ll be sure to get that email out to you as soon as i can . is there anything else i can help you with today in chat ?

This example illustrates exactly what the genericity issue and *frequency bias* constitute for *seq2seq* models. The generic agent-response is overused and in this context it is very misplaced.

[example 3: context discrepancy]

after i download the game manager pauses and tries to activate . i get the message right after that . when i ' ve tried to manually activate i get the message again
Ok, so you are unable to sign into your account, and sign into your account on our website.
alright i see . don ' t worry <NAME> . this seems to be a known issue with windows 8 but there are definitely some steps we can try and that worked for other gamers before

In this example the *base seq2seq* model does generate long, coherent output, yet it is unfit for the context at hand, which can be discerned when comparing with the ground truth next response.

[example 4: ungrammatical decoding]

hi . i have had a membership for over 3 years but have never received the <PROMO> . can you help me with that ?
Hi, it's definitely frustrating to just fine with that.
sure let me take a look and see what happened i ' ve taken a look at your account (…)

The customer complains that he/she has not received a promotional item yet, although being a loyal member for quite some time. The model generates a highly ungrammatical response, which starts off very well, acknowledging the customer's frustration, yet the word-per-word sampling from the learned probability distribution fails to generate a coherent response.

[example 5: conceptual problem]

that is fine i wanted the games but not the membership . thank you for your help and if there are any problems later on i will contact the company .
Ok, i'll be sure to get that email out to you as soon as i can.
absolutely and is there anything else i can help you with today ?

Here a conceptual problem is put forward: the customer says he will contact the company if future problems arise, whereas the model responds to this "contact proposal" by promising a non-referred to, possibly implicitly present in the conversation "email". The ground root gives no clue as to the relevance of the generated output, which points at the fact that high-level structure needs to be modelled and "remembered" to make a *seq2seq* model more capable at responding consistently.

The following samples have been drawn from the test-response generation of the ***index seq2seq*** model. A baseline has been established with the above samples, so where appropriate a comparison will be made with what was generated as a response by the *base seq2seq* model.

[Example 6: good, personal response]

sure email would be fine . . . thank you for helping a 70 yr old cranky lady . . everyone there is always so curteous and helpful
Great i'll have those steps sent out to you before the end of my work day today at the very latest.
ok and you ' re welcome we ' re definitely here to help out to the best of our ability . i ' ve gone ahead and sent you back (…) is <EMAIL>@<DOMAIN>.com a good place to send the coupon ?

Example 6 shows a very good response generated by the model which exhibits a high-grade of personality and "work commitment", which is very hard to manually script into a dialogue system. However, in a practical application these promises the model makes should obviously be connected to a task fulfilling the promise or appropriate conditioning on the response generation level as to what the model can promise. The next example (7) shows that human supervision is required to not let the model generate promises without looking up information in a database and following some scripted rules.

[Example 7: unsupervised, generic promise]

k when i signed up for the <MEMBERSHIP> i got charged for the regular for feb . and march i got refunded march but not feb so i need to get the 6 . 99 for that refunded if possible
Ok.
when a regular game club membership is started the first month of the membership is free so the only 6 . 99 charge that was put towards your card was from february 1st

[Example 8: legitimate I'm sorry]

i cancelled my membership 1 13 13 and was charged again this month .
index: **I'm sorry to hear that and i can certainly help clear this up for you now.**
base: **I'm sorry to hear that and i definitely want to help get this working for you.**
oh no . i ' m sorry about that <CUSTOMER> *. i can certainly help to get this sorted out . (…)*

Example 8 shows that *index seq2seq* can generate responses containing the highly frequent generic "I'm sorry" that are more appropriate than the fully generic formulations of *base seq2seq*. It presents but a small difference, but semantically "clear this up for you" is clearly more fitting than "get this working for you" which is more fitting in the context of a technical issue (very frequent in the data!).

[Example 9: generated chitchat]

i ' ll check my email and if i have any issue i ' ll get back yo you thank you
index: **Sounds like a plan.**
base: **Ok, absolutely.**
that ' s perfect now that i ' m here is there anything else i can do for you ?

Whereas some earlier examples take a hit at the potential use of seq2seq models, here we surprisingly observe the *index seq2seq* model generating output which is non-generic and learns toward more general open-domain chitchat. The response from *base seq2seq* can be considered in the same regard, although again a generic "ok" features, whereas the other response seems more natural.

[Example 10: topic-related, non-generic]

hi i bought a game yesterday and i tried to play it but it doesnt respond once i open it . i tried deleting it and re installing but it still wont work .

index: **I'm sorry to hear that it's possible that this issue might be related to a conflict between the game and computer.**

base: **I'm sorry to hear that.**

thanks for letting me know about the game issues you ' re seeing here and those are certainly never fun to have with your games . i would be glad to assist you today let me just get a minute to pull up your account .

As a final example in the comparison between the baseline and the *index seq2seq* model, it represents clear evidence of the noise-reduction strategy. Although the *index seq2seq* model still responds many customer utterances with generic responses, they are either longer and present a more topic-related response, or present a generic response which is still relevant given the input context.

The samples for *cossim seq2seq* are corrupted due to the model generating customer responses to customer input contexts. To illustrate the severity of the non-persona-bound responses, consider this model's generated response for examples 5 and 6 (cf. supra):

[Examples 11: persona-confusion ~5, 6]

that is fine i wanted the games but not the membership . thank you for your help and if there are any problems later on i will contact the company .

I have been trying to download a game for a game that i already owned it and it's a trial.

sure email would be fine . . . thank you for helping a 70 yr old cranky lady . . everyone there is always so curteous and helpful

Hi i have a problem with a game i purchased on the 26th may i purchased.

[Example 12: long and consistent response]

ok but how do i get back to you ? i had a time doing it before

It's my pleasure and i hope you have a good rest of the day thanks for using live chat today and please always feel free to come back with any other questions we're always happy to help.

i completely understand and if you would like i can send you an email with the steps to get this game installed . that was we can work on getting this fixed without worrying about becoming disconnected . you can reply to my email with any questions and it will go directly to my inbox .

A handful examples such as the above (12) do show that the *cossim seq2seq* model can generate good and surprisingly long responses in the case of keeping to the agent-persona.

5.1.3. End-(to-End) Results

Both the quantitative and qualitative results demonstrate that the *index seq2seq* model performs best. Over the 60 test inputs it shows a consistent drop in non-relevant generic responses, which are observable in abundance in the *base seq2seq* model. The more targeted manual pairwise comparisons between model outputs shows that although generic answers are not done away with completely, the manual post-pruning strategy does pay off to reduce compression caused by the interplay between the relative frequency of generic utterances and the MLE objective function.

Example 10 presents exactly the goal we had in mind with defining the dataset-massaging strategy. Instead of the simple universal reply, which in the case of the troubleshooting dataset is feeling sorry for whatever issue has occurred, the *seq2seq* model has not compressed its full generation space, yet has effectively learned more relevant content-context local mappings with high enough probability to be generated.

What concerns the evaluation of the noise-reduction strategy presented by *index seq2seq*, the hypotheses have been corroborated successfully, although the effect is not as (or too) strong as envisioned. With respect to the former, we acknowledge that the presented method relying on the *custom seq2vec* representation has proved its use, yet that more complex strategies need be researched, possibly fuelled by the insights provided here in the qualitative results. Moreover, two points have to be made with respect to the general approach of *index seq2seq*: First of all, the manual post-pruning of full clusters allows for structurally downsampling the relative frequency of universal replies, but it is only practically feasible for datasets exhibiting good inherent structure which can be captured by a reasonably large number of clusters. Secondly, the experiments here have focused on what a "vanilla" *seq2seq* learner captures on the local inter-sequence level and which remediation strategy helps in improving the learned mappings. However, by presenting the data as clusters which are subsequently pruned, the original high-level conversational structure is adapted. More intelligent dialogue-resampling strategies (for example by keeping an original utterance index store) are needed after the manual post-pruning step if the goal would be to also model high-level structure.

Due to problems with processing the training dataset for *cossim seq2seq* the model has not learned the expected mappings. The constructed clusters showed exceptional promise and this more experimental pairing method was devised, on the one hand supporting on the *custom seq2vec* representation to build the clusters as was the case for *index seq2seq*, and on the other hand resorting again to the representation and maximum cosine similarity within a dialogue's preceding index-range to retrieve the best to be paired customer utterance for a given agent sequence. In the evaluation and for practical usage, we were more interested in learning customer-agent mappings for which the *seq2seq* learner would act as the agent. Disputably, the model suffers from an identity crisis, which might be caused by incorrect processing of the pairs for training the model although exactly the same parametrization and training setup had been selected. Nevertheless, the results represented here are scarce, yet

should not scare away from the original idea behind experimenting with the representation. The intuition behind trying to find the real answer for a given question and making these sequential mappings more explicit to make the encoder-decoder model learn from these can debatably still be translated in a suitable method. Finally, in this regard we have proposed already some requirements as to the representation it could support on and what information can be leveraged to identify purposeful sequence pairs.

5.2. Relevance and impact

5.2.1. Adapting data to the algorithm

In the experiments we have revisited the base encoder-decoder algorithm and evaluated its core functionality on an original troubleshooting domain dataset with two newly proposed approaches, both relying on a custom-defined distributional utterance representation, which seek to remediate the commonly identified *genericity*-generation issue on the turn-level. Instead of adapting the algorithm's objective function, changing the architecture or implementing any attention mechanisms, the adopted strategy sought to change up the process from the data-side. The project has introduced a complimentary research direction, adapting the data to the algorithm, and has shown that this is a viable approach which can offer a good understanding of the inner workings of the algorithm. Again, these insights can be used to identify limitations in the algorithm and be addressed with methods on either side of the model creation. There exists a two-way relation between data and algorithms, which should be exploited accordingly. It is argued for that changing the data by using representations can help the algorithm and subsequently can improve model performance.
In the experiments we illustrated that there is benefit to be gained in better capturing dyadic conversation structure. By representation-conditioning and massaging the input data to the algorithm instead of always endeavouring the way around the experiments endeavoured to shed more light on the potential of *seq2seq* learning for dialogue response generation. Correspondingly, the presented approaches gently force meaningful sequentiality into the data by relying on a distributed compositional representation which allows to measure semantic similarity between utterances and subsequently are used for clustering similar utterances together. By performing continuous space clustering on averaged compositional two-model-concatenated word embeddings, we have built a representation which should capture topical conversation structure inherent in the dataset. Both presented models, *index seq2seq* and *cossim seq2seq*, (should – no hard claim for *cossim seq2seq* can be made) perform an encoder-relaxing strategy by respectively structurally reducing "generic noise" in the data and mapping more meaningful utterance pairs together.
Accordingly, it has to be admitted that *seq2seq* models do learn a representation for language unit mappings during training, that they can form grammatical output and learn a probabilistic mapping between sequences. However, in their current state they are not able to fully learn

natural conversation or any complete representation thereof. More research is needed into improving the quite simplistic probabilistically abstracting and generalizing algorithms, but also in how to adapt the data to make the algorithm perform better. The use of Deep Learning models does not signify that we can do away with all feature engineering or data preparation –whereas admittedly they do replace the process (fully/)partly by the ability to learn representations– nor does it mean that we should not help the algorithm find meaning or structure in what it is given as input.

5.2.2. Unsupervised methods for dialogue representation

One particular characteristic of the experiments that should be stressed is that only unsupervised methods have been used to generate dialogue in the complex troubleshooting domain. Perhaps the only "exception" is the manual post-pruning step which can be regarded as weak supervision at most.

Essentially, the experiments have used a manifold of available unsupervised methods to come as closest as possible to what on the turn-level a dialogue representation (c/sh)ould constitute. We have taken inspiration from every method explored, e.g. the in practice used **intents** which model manually built content-bearing utterance pairs, or employed in the experiments: (document) **clustering** to find the principal structure within the data, a **continuous space vector representation** to allow for semantic similarity reasoning over utterance pairs within dialogues and the ***seq2seq* architecture** for its practical ability to map sequence pairs together. We present our data in a format and a representation to which the *Seq2seq* learner is more able to deal with. Having pairs in clusters plays into the *Seq2seq* learner's limited capacity to model dialogue, but inherently 'reasonably good' ability to model input-output pairs. More high-level, we attempted to gently force the *seq2seq* learner to recognize the topical structure in the dialogues by presenting it a cluster-massaged QA-paired structure of our dataset. Correspondingly, the use of the various unsupervised methods has been an attempt at bridging simple pattern recognition towards intention/purpose recognition over the conversations. The approaches employed in the experiments might not be able to fully apprehend the purpose inherent in all the troubleshooting conversations, but they are theoretically and practically novel in their regard of accomplishing it to some significant degree in (almost) fully unsupervised fashion.

5.2.3. Intent extraction methodology

One of the main contributions of the project is the proposal of a semi-supervised intent extraction methodology. Businesses from different domains increasingly seek to automate their troubleshooting services with a virtual assistant (VA) which should be able to respond at least to the most frequently occurring issues of customers and possibly redirect to human service on a fall-back basis. In practice, VA systems are constructed by engineers with a lot of domain expertise who manually build intent templates and stores these in a database to which the VA system has access (e.g. Niculescu et al. 2014). For example, the human agents in the

troubleshooting dataset from the experiments show knowledge on specific game issues, common OS incompatibilities, error codes, query a user database to retrieve usernames or passwords etc. First of all, modelling the frequent question and answers requires a lot of manual construction time and domain expertise, even when possessing logs of various customer-agent interactions. Manually extracting intents out of backlogged data is a common practice, but in the experiments we argue that with the use of unsupervised methods, the engineers can be helped to construct intents more efficiently in a semi-supervised fashion. In working towards analysing the potential of the *seq2seq* architecture for end-to-end dialogue systems, we have defined a representation clustering method which can be used for extracting similar QA-style structure with the goal of constructing an intent database in a more principled fashion than pure manual engineering, which is time consuming, labour-intensive and expensive. For example, the in section 4.3.2.5. presented cluster 56 deals specifically with game orders by the user which the customer has to look up in the database and answers with a similar strategy every time.

The first exploratory experiment showed that clustering on the document level is too high-level, thereby being not able to give a clear view on what issues customers frequently report and which frequent response strategies human agents have to employ. What was more successful was performing clustering of utterance pairs deemed similar by using the *custom seq2vec* representation. Although it is not a perfect representation, it seems to capture to some degree both general semantic, structural similarity due to the pretrained Google News Corpus word vectors and similarity of local domain entities (e.g. specific game titles, error codes) or actions (e.g. cancel/renew/refund membership, list/order/purchase/try-out games) with the word2vec model for the dataset itself. Additionally, the concatenated composition technique allows the influence of a general word2vec model and adding domain/dataset-specific vocabulary into the mix for measuring similarity. Building a reasonable amount of manually to be inspected clusters based on the utterances of either the customer or the agent, given which one renders the best clustering results, pairing them with the "real" referring question or answer can help identify purposeful intents with varying formulations yet similar meaning.

In the regard for building a representation which leans more toward the practical intent-level, the project does not stand alone. In a very recent publication on *Multiresolution Recurrent Neural Networks* by Serban et al. (2017) a similar suggestion has been made with respect to modelling sub-sequence level structure. They demonstrate the extraction of an *activity-entity representation* which aims to exploit domain knowledge related to troubleshooting:

> "It is motivated by the observation that **most dialogues are centered around activities and entities**. For example, it is very common for users to state a **specific problem they want to resolve**, e.g. how do I install program X? or My driver X doesn't work, how do I fix it? In response to such questions, other users [in the case of the Ubuntu Chat Corpus or the human Agent in our case] **often respond with specific instructions**, e.g. Go to website X to download software Y or Try to execute command X. In such cases, it is clear that the **principal information resides in the technical entities and in the verbs** (e.g. install, fix, download), and therefore

that it will be advantageous to explicitly model this structure." Serban et al. (2017: 5, my emphasis)

Of course, their extracted representation will be used for a different purpose (cf. 3.5.2.1. for details) than what we have attempted, but both have the intention of extracting domain knowledge present in the data, which can be exploited for multiple other purposes.

As a preliminary conclusion to this subsection, we would like to underscore the usefulness of the presented approach for semi-automatic intent identification. First and foremost, it is very valuable for a business hosting a support service to get insight into their backlogged data and identify any frequently occurring problems or human-human chat situations. Furthermore, the presented representation and methods can be elaborated on by the community of researcher and business users working on dialogue systems or the self-styled "bots". Building a dialogue system is no easy matter, certainly if the goal is to provide natural, non-goal oriented conversation. Additionally, the conception of the dialogue system is but a small part in a natural cyclic process:

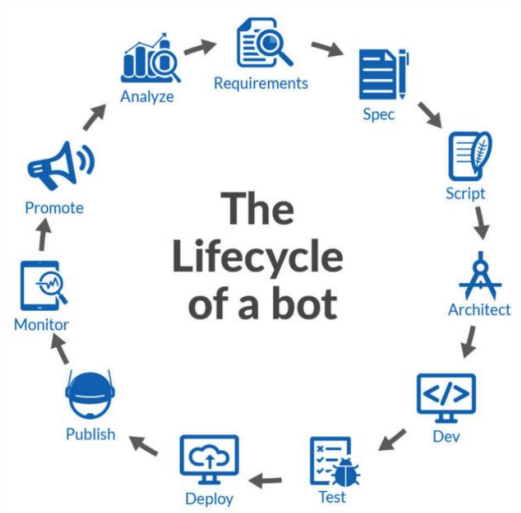


Figure 15: In order to appreciate the complexity and amount of work which is invested into what makes a dialogue system or "bot" successful, the various steps in the end-to-end building process must be taken into account. Image source and detailed description: https://goo.gl/jYptGK

With respect to the intent identification method, this but represents one part in the full building process, namely the scripting of the conversational interface, which should be scripted to be as representative as possible of actual user conversations.
Although the quality of conversation of communication-based AI is improving at a steady rate, they still require significant research and structural additions before fully replacing more conventional pipeline dialogue systems. As a tentative suggestion, for now the current models can be put to good use to exploit domain knowledge and help automate the engineering processes. As an extra takeaway, by working on "fake AI", insights can be made on the requisites for real communication-based AI (Arthur Szlam NIPS 2016).

5.3. Project challenges and identified pitfalls

Admittedly, the initial project goals were too ambitious and have been adjusted accordingly. Throughout the experiments it became clear that Deep Learning technology requires a high grade of know-how and that making it work for dialogue systems presents an even greater challenge. In order to give perspective to the project's results, we would like to point out what pitfalls have been identified throughout the work, which specific challenges have been contributed to and which will prove challenging for future research.

- Data (pre)processing and encoding–decoding

In the experiments a lot of time and attention has been paid to the preprocessing of the data to build up the *custom seq2vec* representation. While performing the former tasks, a large amount of "biases" have been discerned in the data. On the domain-level, the agents functions as the first point of contact and a lot of valuable troubleshooting information is outsourced to an installable external tool or via email correspondence. On the conversational level, the relative *frequency bias* of prompts and generic reply strategies is overly clear. Other difficult to model linguistic regularities are contextual (e.g. "my desktop - it") and conceptual (e.g. not explicitly mentioned video game NPC – "he") pronoun references. Although two approaches have been defined to mediate most of the identified data biases, they will inevitably have their effect on the final generated results. However noisy the dataset, it presents a valuable opportunity to test the limit(ation)s of Deep Learning models.
The introduced *seq2seq* models constitute a framework of relatively simple algorithms which are purely data-driven and abstract across domains, yet require a lot of data to be able to generalize properly. The latter requirement tells a cautionary tale which we have experienced first-hand.
Even with the data-massaging of the input conversations the *seq2seq* models are not able to produce relevant output. This mismatch between what the models should have learned and what is generated points either at the fact that the approaches were not successful at completely eliminating the too strong *frequency bias*, or that the dataset is too small and exhibits too little variation of the same dialogue situations. In the first case, it might be a good idea to make the approaches more complex and/or resort to an alternative objective function which can identify and down-sample the relative frequency of generic answers (cf. Li et al. 2016a). In the second case, more data needs to be collected within the same domain (possibly Ubuntu Chat Corpus, though too technical and not game-related issues) or a new model architecture has to be devised which can generalize with less data.
It is hypothesized however that, being the biggest factor, the dataset might in hindsight have been too small for experiments with a *seq2seq* architecture, certainly if the goal had been to model open-domain dialogue. To nuance this pitfall, unsupervised learning with Deep Learning models is still in its early research phase and the theoretical acumen will probably only be translated in real applications in an estimated 10 years. More in general, it can be

stated that a sufficiently large representative dataset represents the Achilles' heel of end-to-end training for complex output. Setting an experimental lower bound on the amount of data necessary for a "vanilla" *seq2seq* to learn and generalize from poses a valuable insight in itself. The conceptual bottleneck of more data – with additional caution for biases present in the data - and a more complex architecture serves as an important takeaway for future *seq2seq* experiments and applications.

- Network designer choices & empirical parameter tuning

Originally, the experiments were defined as a replication experiment of "a Neural Conversational Model" (Vinyals & Le 2015) on an original troubleshooting dataset. We have followed closely the implementation suggested in the paper, although given that the same settings might not apply to our dataset, some parameters have been adjusted accordingly. A primary example is the fine-tuning (2*average utterance length to allow for some margin) of the maximum encoder-decoder length parameter, which was originally at a setting too small to capture the full length of utterances present in our dataset. In the first attempts to build a *seq2seq* model with the implementation we have experimented with different settings (500 – 1000 – 1300) for the number of hidden neurons in the encoder layer and decoder layer to consider both possibly underfitting or overfitting the model on the data. Increasing this number did not significantly improve the quality of the generated responses, but it did increase training time drastically. Therefore, for constructing the final models the layers' hidden neuron size has been set to the lowest experimented setting of 500. Additionally, given that increasing the size of the hidden neurons did not bring about performance increase, adding more layers, i.e. making the architecture "deeper", was not attempted. An increase in number of layers also requires a potentially exponential increase in dataset size to make the model generalize well. We opted not to experiment too much with other parameters given that appropriately tuning each of them asks for a lot of time-consuming experimentation and testing.
In practice, optimization in Deep Learning entails choosing the most appropriate architecture for the task at hand, setting a cornucopia of parameters right and motivating designer choices. However, there is a need for a more theoretical basis on when and how to use a given Deep Learning model, what is the influence of the different parameters settings, how the performance will be affected by tuning them and how to make the in essence 'black box' models interpretable. Given the short-term usability of the constructed models, we did not endeavour too much in parameter optimization, although arguably it might have influenced the final results. In general, we have made use of open-source software, used mainly unsupervised methods with deliberate vanilla parametrization to allow for analysing the effect of switching it up on the input-side.

- Evaluation of model performance

Evaluation of unsupervised generative (not the retrieval-based) dialogue systems presents a general problem. There has been no reported objective automatic evaluation metric for which human correlation is significantly high enough. Therefore, as suggested by Liu et al (2016) we opted for human, manual evaluation which is the safest and most representative. The method of evaluation focuses on the relevancy of generated responses on the turn-level. Even for humans it is hard to evaluate what constitutes good conversation, so here we opted for restricting evaluation to the turn-level, also given that we were interested in the individual models' learned mappings between utterances. In that respect, the evaluation reports effectively on the bias (here as in bias-variance tradeoff) of the individual models by comparing generated responses with the ground root response.

- Relevance in answer generation

As has been explained in 4.2.A-C, we have motivated and experimented with different approaches to improve relevance in answer generation. Each approach has its pros and cons, but both draw on the *custom seq2vec* representation in an effort to identify and mediate the genericity issue in the dataset which is being reinforced by compression due to the MLE objective function. Whereas the experiments have focused on relevance in utterance pairs by presenting more structured input to the encoder, this "structured input" argument should also be taken into account on the conversation level. With the goal of generating complex structure with an encoder-decoder model, the input should also be presented with diversity yet reasonable regularity in structure. Short-term and long-term dependencies require joint modelling so that the simple *seq2seq* can generalize over them together, hopefully ensuring a higher grade of relevance on the conversational level. One key requirement for an extended dialog-based conversational system is the ability to maintain context over a period of time across multiple Q/A sequences (Vinyals & Le 2015).
The challenge of ensuring good, meaningful "local structure" mappings has been endeavoured, yet it is clear that more research is needed towards better "dialogue representations" which capture the inherent structure in data more efficiently. With respect to this, learning a simple unconditioned probability distribution to represent and generate from a corpus of conversations is not a viable long-term conversation strategy, nor does it not approach the real fine-grained structure of natural conversation.

6. Conclusion

In the final section we will give an overview of the major insights from each individual part of the project. The contextualisation of the recent trends in the industrial setting has shown that there is an increasing interest in open-domain, intelligent dialogue systems correlating with the breakthrough of Deep Learning. Traditional modular dialogue systems prove insufficiently capable of scaling towards natural conversation and rely too much on manual engineering within specific domains, which makes the hard efforts non-transferable. End-to-end dialogue

systems show promise due to the joint modelling of all components, natural language understanding-generation and dialogue management, which offers the possibility of jointly optimizing the flexible interactions between them. Significant advances have already been made in end-to-end training with (deep) recurrent neural networks, the *seq2seq* framework, yet it needs to be underlined that for practical dialogue system applications it is too early, currently the field is still in its research phase:

> "Most of the value of Deep Learning today is in narrow domains where you can get a lot of data. Here's one example of something it cannot do: have a meaningful conversation. There are demos, and if you cherry-pick the conversation, it looks like it's having a meaningful conversation, but if you actually try it yourself, it quickly goes off the rails." (Interview with Andrew Ng 2016)

The literature review section offers perspective to the above claim by providing a survey of the recent academic-industrial contributions and the present state-of-the-art in end-to-end generative dialogue systems. Essentially, various types of data-driven dialogue systems exist, which each differ in the approach of generating output. Most of interest and arguably holding the greatest potential are "unsupervised" dialogue response generation systems, which hold the promise of flexible, adaptable generation of conversation, which is sampled word-per-word from a probability distribution learned in data-driven fashion.

Notwithstanding the simple ingenuity of this approach, the data-driven and unsupervised properties present themselves as a conceptual bottleneck which makes representative data collection a primary requisite for driving progress and appropriate evaluation a major challenge. More specifically, when designing a non-goal oriented dialogue system one should evaluate over all levels of the system hierarchy. Moreover, the whole research community can benefit from the definition of an automated metric with high human correlation and meaningful (synthetic) tasks or compelling experiments which can help in the evaluation of specific requisite qualities of end-to-end dialogue systems.

However large the potential of unsupervised generative dialogue systems, a large amount of issues have been and remain to be addressed both on the local and the global dialogue context level. A most pervasive problem for end-to-end generation is the *genericity* issue, which is the result of the relative frequency of universal replies and the simplistic MLE objective function compressing the generation output space. Alternatively, an MMI objective function has been suggested (Li et al. 2016a) and Mou et al. (2016) & Xing et al. (2016) provide methods to augment replies with respectively content noun terms or topic information. In the experiments we propose a noise-reduction strategy which comes closer to meaningful responses, yet the issue is not remediated completely. Nevertheless, this delivers some insights as to how to influence the objective function and what mappings are learned or compressed to. Ensuring good local mappings requires not only more data, but also depends heavily on representative diversity in the training data. More research and experimentation are needed to approach a solution. Another identified issue is the lack of speaker consistency of the end-to-end dialogue models, which is another requirement to be met for generating natural and consistent conversation. Li et al. (2016c) demonstrate that the *seq2seq*

architecture allows a straightforward mechanism to do exactly this. More problems remain to be addressed on the subsequence level, where natural language phenomena dominate such as pronoun correference, temporal structure and ambiguity. Linguistics has a large tradition of addressing the features of human-to-human dialogue (Clark & Brennan 1991) and the insights from the field can be translated into new methods that integrate these (symbolic) features with the flexibility in learning of neural network architectures (e.g. Boleda et al. 2016).

A more high-level discussion has proceeded of research contributions in resolving issues with global dialogue context. The conversational level has not been the focus of the experiments, but the innovative ideas presented in this section can trickle down to improve the lower levels as well. Main suggestions have been changing the architecture to allow for memory-based reasoning, attention for short and long-term dependencies and access to external resources to condition what the model should learn and generate. With the goal of constructing communication-based AI - an open-domain, intelligent, response, interactive dialogue system being an instrumental part – the *seq2seq* architectures need to be improved for more explicit state-tracking required in multi-context generation and should be able to predict forward states to constrain and pinpoint the best possible generation strategy. In this regard simulation with (deep) reinforcement learning, in present systems still the main tool for dialogue management, is gaining increased consideration. It can offer the advantage of adding an online-policy learning component to the end-to-end dialogue systems, seeing that the conversational model they will have learned is essentially an off-policy data-driven distribution. Submitting an end-to-end trained dialogue system to simulation with new input can identify deficits in the offline-learned mappings and making it learn in a reward-based fashion can prove very valuable to increase its on-policy capacity to natural conversation. Very recently, simulated environments have been made available to test a system's conversational capacities and let it acquire new facts and skills through communication. More work is needed to increase the complexity and impact of both the tasks and the environments meant to improve the intelligence of systems reasoning with natural language.

Finally, the major issues and challenges have not been solved yet, but the models and architectures presented as the state-of-the-art in end-to-end dialogue generation show great potential and as the research community continues to grow stepwise improvement and rigour will make the resulting systems evolve gradually towards communication-based AI.

The experiments conducted in the project aim at arriving at a deeper understanding of the encoder-decoder algorithm – *seq2seq* architecture –, what it is expected to learn from the data versus what it actually learns and generates. More specifically, we will focus on the local dialogue context and seek to improve what mappings can be learned there by applying a basic implementation of an LSTM encoder-decoder model on an original dataset. The sequence pairs in this troubleshooting domain dataset are characterized by the *frequency bias* of the human agent's responses to customer reported questions. We would like to see how the standard *seq2seq* architecture responds, i.e. is affected by, with regard to the biases in the

data and how the negative effect can possibly be mediated. Moreover, the troubleshooting domain presents an excellent testing ground seeing that the conversations are mixed goal and non-goal oriented and ideally a system backing a personal assistant service should be able to do both.
Consequently, we have described our implementation of a LSTM-encoder-decoder architecture, motivated the parametrization and designer choices adjusted to the data. Additionally we have proposed new methods based on a custom-built, distributional representation, *custom seq2vec*. The representation is based on employing the concatenated composition technique on a pre-trained word2vec model and one model specifically built for the dataset, thus allowing to measure similarity between utterances and building clusters for a dialogue participant. For this purpose, we have applied MiniBatch K-Means clustering and motivated the advantage it holds with regard to scalability.
Two approaches have been defined to retrieve sequence pairs and massage the data: The first approach being simple index-based retrieval of corresponding other participant's utterances and structural post-pruning of "bad" sequence pair clusters. The second approach re-used the *custom seq2vec* representation together with a *maximum cosine similarity* objective to retrieve the corresponding utterance that is most meaningful with the goal of creating more purposeful sequence pairs. Two models have been proposed with their respective approach, *index seq2seq* and *cossim seq2seq*, and a baseline model, *base seq2seq*, has been established to evaluate the effect of each approach.

The evaluation has focused on pairwise comparing the relevancy of individual model generated responses to a representative test set. The quantitative results show that in no means the *genericity* issue has been solved completely, seeing that for any model more than 50% of generated responses feature a generic reply or a misplaced answer given the input test context. However, the *index seq2seq* model does show improvement over the baseline in downsampling generic responses which are not appropriate in the context. The qualitative results elaborate on this insight by illustrating representative test-model samples and corroborating the hypotheses for *index seq2seq*, proving the usefulness of the noise-reduction strategy. In the discussion the insights from the project are bundled together and tentative suggestions have been made towards further research and current usage of the models. Whereas the usefulness of the approach from *cossim seq2seq* has not been delivered in evaluation due to corrupted model output, its use for clustering similar utterance pairs in order to more structurally identify purposeful customer-agent mappings has been motivated.

Finally, there are lots of problems waiting to be solved in order to have a much-desired communication-based AI system. Some problems pose theoretical interests, other present a more practical appeal. Although hard predictions are being made as to when we will reach this point, we have stressed both the potential of the researched models and the currently identified limitations. More time and research is necessary to make fundamental changes to the simple *seq2seq* model architecture. One of many options that show long-term promise

although non-trivial is ‘hybridization’ (e.g. Wen et al. 2016) by adding supervision to the otherwise fully unsupervised end-to-end models, which can help them induce general knowledge and provide them the elusive natural human touch. We hope to have given a clear and relevant overview of the current trends to which we have sought to contribute to and offered perspective on where to the dialogue systems research field can evolve.

7. References

"utterance", *OED Online*. Oxford University Press, January 2017. Web. 2 Jan 2017.

Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José M.F. Moura, Devi Parikh, Dhruv Batra. Visual Dialog. *ArXiv 2016*

Antoine Bordes and Jason Weston Learning End-to-End Goal-Oriented Dialog 2016 arxiv:1605.07683.

Arthur Szlam: In praise of fake AI. 2016. NIPS: Machine Intelligence Workshop. (https://mainatnips.github.io/)

Arthur, D., & Vassilvitskii, S. (2007, January). k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms* (pp. 1027-1035). Society for Industrial and Applied Mathematics.

Ba, J., Hinton, G.E., Mnih, V., Leibo, J.Z. and Ionescu, C., 2016. Using Fast Weights to Attend to the Recent Past. In *Advances In Neural Information Processing Systems* (pp. 4331-4339).

Bahdanau, D., Cho, K. and Bengio, Y., 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bangalore, S., and A. Stent, *Natural Language Generation in Interactive Systems,* : Cambridge University Press, 2014.

Baroni, M., Dinu, G. and Kruszewski, G. 2014. Don't count, predict! a systematic comparison of context counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics* (Volume 1: Long Papers), pages 238{247, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

Baroni, Marco. 2016. Nursing Turing's Child Machine: Towards Communication-based Artificial Intelligence. NIPS Machine Intelligence workshop.

Bengio, Y., Boulanger-Lewandowski, N. and Pascanu, R., 2013, May. Advances in optimizing recurrent networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 8624-8628). IEEE.

Bengio, Y., Boulanger-Lewandowski, N. and Pascanu, R., 2013, May. Advances in optimizing recurrent networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 8624-8628). IEEE.

Bhuwan Dhingra, Hanxiao Liu, Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. Gated-attention

Boleda, G., S. Padó and M. Baroni. 2016. Show me the cup: Reference with continuous representations. arXiv e-print 1606.08777.

Border and Weston. 2016. Learning end-to-end goal-oriented dialog. arXiv. • Clark. 1996. Using language. CUP.

Britz, Denny. 01/04/16. “DEEP LEARNING FOR CHATBOTS, PART 1 – INTRODUCTION” Retrieved on 23/12/16. http://www.wildml.com/2016/04/deep-learning-for-chatbots-part-1-introduction/

C.-Y. Lin. 2004. Rouge: A package for automatic evaluation of summaries. In Text summarization branches out: Proceedings of the ACL-04 workshop, volume 8.

Carey, Pete. 27/03/2016. ‘Baidu research chief Andrew Ng fixed on self-taught computers, self-driving cars’ Retrieved on 22/12/2016. http://www.seattletimes.com/business/baidu-research-chief-andrew-ng-fixed-on-self-taught-computers-self-driving-cars/

Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2016. Topic augmented neural response generation with a joint attention mechanism. arXiv preprint arXiv:1606.08340.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Cho, S.J.K., Memisevic, R. and Bengio, Y., 2015. On Using Very Large Target Vocabulary for Neural Machine Translation.

Chung, J., Gulcehre, C., and Cho, K. et al. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv:1412.3555, 2014.

Clark and Brennan. 1991. Grounding in communication. In Perspectives on socially shared cognition.

Cristian Danescu-Niculescu-Mizil and Lillian Lee "Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs" in Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, ACL 2011.

Das et al. 2016. Visual dialog. arXiv.

Diaz, F., Mitra, B. and Craswell, N., 2016. Query Expansion with Locally-Trained Word Embeddings. *arXiv preprint arXiv:1605.07891*.

Encoder-decoder architecture image. http://googleresearch.blogspot.ca/2015/11/computer-respond-to-this-email.html image Retrieved 26/12/16.

Farrell, Michael et al. 2016. “End-To-End Generative Dialogue” https://github.com/michaelfarrell76/End-To-End-Generative-Dialogue

Garrod and Pickering. 2004. Why is conversation so easy. TICS.

Garten, J., Sagae, K., Ustun, V. and Dehghani, M., 2015, June. Combining Distributed Vector Representations for Words. In *Proceedings of NAACL-HLT* (pp. 95-101).

Gašić, M., Mrkšić, N., Rojas-Barahona, L.M., Su, P.H., Ultes, S., Vandyke, D., Wen, T.H. and Young, S., 2016. Dialogue manager domain adaptation using Gaussian process reinforcement learning. *Computer Speech & Language*.

Georgila, K., Henderson, J. and Lemon, O., 2006, September. User simulation for spoken dialogue systems: learning and evaluation. In *INTERSPEECH*.

Graves, A. Generating sequences with recurrent neural networks. arXiv:1308.0850, 2013.

Graves, A., Mohamed, A., and Hinton, G. E. Speech recognition with deep recurrent neural networks. In Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6645– 6649, 2013.

Graves, A., Wayne, G., and Danihelka, I. Neural turing machines. arXiv:1412.3555, 2014.

Gu, J., Lu, Z., Li, H. and Li, V.O., 2016. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*.

Gu, J., Lu, Z., Li, H., & Li, V. O. (2016). Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*.

Hasson et al. 2012. Brain-to-brain coupling: A mechanism for creating and sharing a social world. Trends Cog Sci.

Hastie, Helen, "Awkward Silence? The Evaluation of Social Dialogue Systems". 2016. NIPS invited talk in Dialogue Workshop.

Hill, F., Cho, K., & Korhonen, A. (2016b). Learning distributed representations of sentences from unlabelled data. *arXiv preprint arXiv:1602.03483*.

Hill, Felix Antoine Bordes, Sumit Chopra, and Jason Weston.(2016a) The goldilocks principle: Reading children's books with explicit memory representations. In ICLR, 2016.

Hinton, Bengio, Lecun. 2015. "Deep Learning tutorial @ NIPS 2015" https://www.iro.umontreal.ca/~bengioy/talks/DL-Tutorial-NIPS2015.pdf

Hochreiter, S. and Schmidhuber, J. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.

J. F. Kelley. An iterative design methodology for user-friendly natural language office information applications. ACM Transaction on Information Systems, 1984

J. Schmidhuber (2015). Deep learning in neural networks: An overview. Neural Networks, 61, 85-117.

Johnson, M., Schuster, M., Le, Q.V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G. and Hughes, M., 2016. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *arXiv preprint arXiv:1611.04558*.

Jokinen, K. and McTear, M., 2009. Spoken dialogue systems. *Synthesis Lectures on Human Language Technologies*, *2*(1), pp.1-151.

Kalchbrenner, N. and Blunsom, P., 2013. Recurrent Continuous Translation Models. In *EMNLP* (Vol. 3, No. 39, p. 413).

Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Skip-thought vectors. In *Advances in neural information processing systems* (pp. 3294-3302).

Komatsu, H., Tian, R., Okazaki, N. and Inui, K., 2015. Reducing Lexical Features in Parsing by Word Embeddings.

Lazaridou, A., Peysakhovich, A. and Baroni, M., 2016. Multi-Agent Cooperation and the Emergence of (Natural) Language. *arXiv preprint arXiv:1612.07182*.

Le, Q. V., & Mikolov, T. (2014, June). Distributed Representations of Sentences and Documents. In *ICML* (Vol. 14, pp. 1188-1196).

Let's Discuss: Learning Methods For Dialogue NIPS 2016 Workshop http://letsdiscussnips2016.weebly.com/

Li, J., Galley, M., Brockett, C., Gao, J., & Dolan, B. (2016a). A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

Li, J., Galley, M., Brockett, C., Gao, J., & Dolan, B. (2016b). A Persona-Based Neural Conversation Model. *arXiv preprint arXiv:1603.06155*.

Li, J., Monroe, W., Ritter, A., & Jurafsky, D. (2016c). Deep Reinforcement Learning for Dialogue Generation. *arXiv preprint arXiv:1606.01541*.

Li, Jiwei. 2016. Neural Dialogue Generation. Stanford University: CS class.

Lin, Chu-Cheng, Waleed Ammar, Chris Dyer, and Lori Levin. 2015. Unsupervised POS induction with word embeddings. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*. Denver, CO, pages 1311–1316.

Liu, C. W., Lowe, R., Serban, I. V., Noseworthy, M., Charlin, L., & Pineau, J. (2016). How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. arXiv preprint arXiv:1603.08023.

Lowe, R., Pow, N., Serban, I., & Pineau, J. (2015). The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. *arXiv preprint arXiv:1506.08909*.

Lowe, Ryan. 2016. Workshop: Modern Challenges in learning end-to-end dialogue systems, Sept 2016 http://ttic.uchicago.edu/~klivescu/MLSLP2016/lowe.htm

Luong, M.T., Pham, H. and Manning, C.D., 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Luong, M.T., Sutskever, I., Le, Q.V., Vinyals, O. and Zaremba, W., 2014. Addressing the rare word problem in neural machine translation. *arXiv preprint arXiv:1410.8206*.

M. Galley, C. Brockett, A. Sordoni, Y. Ji, M. Auli, C. Quirk, M. Mitchell, J. Gao, and B. Dolan. 2015. deltableu: A discriminative metric for generation tasks with intrinsically diverse targets. arXiv preprint arXiv:1506.06863.

M. Henderson, B. Thomson, and S. Young. Deep neural network approach for the dialog state tracking challenge. In Special Interest Group on Discourse and Dialogue (SIGDIAL), 2013.

Machine Intelligence Workshop @ NIPS 2016: https://mainatnips.github.io/

Markoff, John; Mozur, Paul 2015. "For Sympathetic Ear, More Chinese Turn to Smartphone Program". *The New York Times*. ISSN 0362-4331. Retrieved 12/01/17

Michael Denkowski and Alon Lavie, "Meteor Universal: Language Specific Translation Evaluation for Any Target Language",*Proceedings of the EACL 2014 Workshop on Statistical Machine Translation, 2014*

Mikolov, Tomas; et al. 2013. Efficient Estimation of Word Representations in Vector Space. arXiv:1301.3781

Mou, L., Song, Y., Yan, R., Li, G., Zhang, L., & Jin, Z. (2016). Sequence to backward and forward sequences: A content-introducing approach to generative short-text conversation. *arXiv preprint arXiv:1607.00970*.

Niculescu, A.I., Yeo, K.H., D'Haro, L.F., Kim, S., Jiang, R. and Banchs, R.E., 2014, December. Design and evaluation of a conversational agent for the touristic domain. In *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific* (pp. 1-10). IEEE.

Novet, Jordan. 18/09/16. "Oracle launches a chatbot development platform" Retrieved 22/12/16 http://venturebeat.com/2016/09/18/oracle-launches-a-chatbot-development-platform/.

Olah, Christopher 2015.Deep Learning blog. http://colah.github.io/posts/2015-09-NN-Types-FP/ Retrieved 11/11/2016.

Passos, A., Kumar, V. and McCallum, A., 2014. Lexicon infused phrase embeddings for named entity resolution. *arXiv preprint arXiv:1404.5367*.

Pietquin, O. & Hastie, H. (2013), *"A survey on metrics for the evaluation of user simulations"*, Knowledge Engineering Review., February, 2013, Vol. 28(01), pp. 59-73 *first published as FirstView*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In EMNLP, 2016.

Rehurek, R. and Sojka, P., 2010. Software framework for topic modelling with large corpora. In *In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, *20*, 53-65.

Roy, D., 2003. Grounded spoken language acquisition: Experiments in word learning. *IEEE transactions on multimedia*, *5*(2), pp.197-209.

Rush, A.M., Chopra, S. and Weston, J., 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*.

Rush, A.M., Chopra, S. and Weston, J., 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*.

S. Banerjee and A. Lavie. 2005. METEOR: An automatic metric for mt evaluation with improved correlation with human judgments. In Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization.

S. Kim, L. F. DHaro, R. E. Banchs, J. Williams, and M. Henderson. Dialog state tracking challenge 4. 2015.

Schatzmann, J., Georgila, K. and Young, S., 2005. Quantitative evaluation of user simulation techniques for spoken dialogue systems. In *6th SIGdial Workshop on DISCOURSE and DIALOGUE*.

Schmidhuber, Jürgen. 2015. "The Deep Learning Conspiracy". http://people.idsia.ch/~juergen/deep-learning-conspiracy.html Retrieved 22/12/16.

Serban, I. V., Lowe, R., Charlin, L., & Pineau, J. (2015). A survey of available corpora for building data-driven dialogue systems. *arXiv preprint arXiv:1512.05742*.

Serban, I. V., Sordoni, A., Lowe, R., Charlin, L., Pineau, J., Courville, A., & Bengio, Y. (2016b). A Hierarchical Latent Variable Encoder-Decoder Model for Generating Dialogues. *arXiv preprint arXiv:1605.06069*.

Serban, I. V., Sordoni, A., Bengio, Y., Courville, A., & Pineau, J. (2016a). Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI-16)*.

Serban, I. V., Klinger, T., Tesauro, G., Talamadupula, K., Zhou, B., Bengio, Y., & Courville, A. (2017). Multiresolution Recurrent Neural Networks: An Application to Dialogue Response Generation. arXiv preprint arXiv:1606.00776.

Shang, L., Lu, Z., & Li, H. (2015). Neural responding machine for short-text conversation. *arXiv preprint arXiv:1503.02364*.

Shen, Y., He, X., Gao, J., Deng, L., and Mesnil G., 2014. Learning semantic representations using convolutional neural networks for web search. In Proc. WWW, pages 373–374.

Socher, Richard, Perelygin, Alex, Wu, Jean Y., Chuang, Jason, Manning, Christopher D., Ng, Andrew Y., and Potts, Christopher. Recursive deep models for semantic compositionality over a sentiment treebank. In Conference on Empirical Methods in Natural Language Processing, 2013

Sordoni, A., Galley, M., Auli, M., Brockett, C., Ji, Y., Mitchell, M., ... & Dolan, B. (2015). A neural network approach to context-sensitive generation of conversational responses. *arXiv preprint arXiv:1506.06714*.

Speer, R. and Chin, J., 2016. An Ensemble Method to Produce High-Quality Word Embeddings. *arXiv preprint arXiv:1604.01692*.

Sutskever et al. 2014. Sequence to sequence learning with neural networks. NIPS.

T.Wen, M. Gaˇsic, D. Kim, N. Mrkˇsic, P. Su, D. Vandyke, and S. Young. Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking. Special Interest Group on Discourse and Dialogue (SIGDIAL), 2015a.

TensorFlow Development team. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of NIPS, 2013.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In Proceedings of Workshop at ICLR, 2013.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic Regularities in Continuous Space Word Representations. In Proceedings of NAACL HLT, 2013.

Trischler, A., Ye, Z., Yuan, X., He, J., Bachman, P., Suleman, K., 2016. A Parallel-Hierarchical Model for Machine Comprehension on Sparse Data. *ArXiv e-prints* 1603, arXiv:1603.08884.

Uthus, D. C., & Aha, D. W. (2013, March). The Ubuntu Chat Corpus for Multiparticipant Chat Analysis. In *AAAI Spring Symposium: Analyzing Microtext* (Vol. 13, p. 01).

Vinyals, O., & Le, Q. (2015). A neural conversational model. *arXiv preprint arXiv:1506.05869*.

Vinyals, O., Kaiser, Ł., Koo, T., Petrov, S., Sutskever, I. and Hinton, G., 2015. Grammar as a foreign language. In *Advances in Neural Information Processing Systems* (pp. 2773-2781).

Vulic, I. and Moens, M.-F. 2015. Monolingual and cross-lingual information retrieval models based on (bilingual) word embeddings. In *Proc. SIGIR*, pages 363–372. ACM.

Walker, M.A., Whittaker, S.J., Stent, A., Maloor, P., Moore, J., Johnston, M. and Vasireddy, G., 2004. Generation and evaluation of user tailored responses in multimodal dialogue. *Cognitive Science*, *28*(5), pp.811-840.

Wang, W.Y. and Yang, D., 2015(Sept). That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015), Lisbon, Portugal*.

Weizenbaum, Joseph (January 1966). "ELIZA – A Computer Program For the Study of Natural Language Communication Between Man and Machine" (PDF).

Wen, T. H., Gasic, M., Mrksic, N., Rojas-Barahona, L. M., Su, P. H., Ultes, S., ... & Young, S. 2016a. A Network-based End-to-End Trainable Task-oriented Dialogue System. *arXiv preprint arXiv:1604.04562*.

Wen, T.H., Gasic, M., Mrksic, N., Rojas-Barahona, L.M., Su, P.H., Ultes, S., Vandyke, D. and Young, S., 2016b. Conditional generation and snapshot learning in neural dialogue systems. *arXiv preprint arXiv:1606.03352*.

Wen, T.H., Gasic, M., Mrksic, N., Su, P.H., Vandyke, D. and Young, S., 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745*.

Williams, J. D., & Zweig, G. (2016). End-to-end LSTM-based dialog control optimized with supervised and reinforcement learning. *arXiv preprint arXiv:1606.01269*.

Williams, Jason D. 2007. "Applying POMDPs to dialog systems in the troubleshooting domain." *Proceedings of the Workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technologies*. Association for Computational Linguistics.

X. Wang, C. Yuan, Recent Advances on Human-Computer Dialogues, CAAI Transactions on Intelligence Technology (2017), doi: 10.1016/j.trit.2016.12.004.

Xiong, K., Cui, A., Zhang, Z., & Li, M. (2016). Neural Contextual Conversation Learning with Labeled Question-Answering Pairs. *arXiv preprint arXiv:1607.05809*.

Y. LeCun, Y. Bengio, G. Hinton (2015). Deep Learning. Nature 521, 436-444.

Yin, J., Jiang, X., Lu, Z., Shang, L., Li, H. and Li, X., 2015. Neural generative question answering. *arXiv preprint arXiv:1512.01337*.

Yin, J., Jiang, X., Lu, Z., Shang, L., Li, H. and Li, X., 2015. Neural generative question answering. *arXiv preprint arXiv:1512.01337*.

Yin, P., Lu, Z., Li, H. and Kao, B., 2015. Neural Enquirer: Learning to Query Tables. *arXiv preprint arXiv:1512.00965*.

Young et al. 2013. POMDP-based statistical spoken dialogue systems: A review. Proc IEEE.

Young, S., Gašić, M., Keizer, S., Mairesse, F., Schatzmann, J., Thomson, B. and Yu, K., 2010. The hidden information state model: A practical framework for POMDP-based spoken dialogue management. *Computer Speech & Language*, *24*(2), pp.150-174.

Yu, Z., Nicolich-Henkin, L., Black, A. W., & Rudnicky, A. I. (2016b). A Wizard-of-Oz Study on A Non-Task-Oriented Dialog Systems That Reacts to User Engagement. In Proceedings of SIGDIAL.

Yu, Z., Xu, Z., Black, A. W., & Rudnicky, A. I. (2016a). Strategy and Policy Learning for Non-Task-Oriented Conversational Systems. In Proceedings of SIGDIAL .

Zhilin Yang, Bhuwan Dhingra, Ye Yuan, Junjie Hu, William W. Cohen, Ruslan Salakhutdinov SOTA CBT: WORDS OR CHARACTERS? FINE-GRAINED GATING FOR READING COMPREHENSION 2017.

Appropriate modelling of dialogue or conversation presents an imperative task in order to achieve natural language understanding and 'real' machine intelligence. Whereas foregoing approaches focused on handcrafting rules for principally specific domains (e.g. restaurant ordering), recent initiatives have been proposing adaptations of deep recurrent neural network architectures to build unsupervised dialogue response generation systems. In this paper, we will review these innovative contributions for the larger task of dialogue generation & modelling and provide an overview of the state-of-the-art.

By means of an experiment with a new and original closed-domain noisy dataset we will test the potential of the recently proposed *sequence-to-sequence* framework for the task at hand. Hereby we hope to shed light on what is currently possible with the newly devised models and identify which directions are generally promising for the future.

The results demonstrate that the model and the ensuing generated responses are determined by many factors, most importantly the nature, size and preprocessing of the data on which it is trained and of which it seeks to model the inherent conversational patterns. As expected, the currently proposed *sequence-to-sequence* inspired models are not yet able to function as stand-alone dialogue systems, but can find their usage in unsupervised intent extraction or as a fall-back basis for an existing virtual assistant.

Keywords| Generative Dialogue modelling, End-to-End Training, *seq2seq* architecture, Unsupervised Intent Extraction