A Survey of Natural Language Generation Techniques with a Focus on Dialogue Systems - Past, Present and Future Directions

Sashank Santhanam SSANTHA1@UNCC.EDU
Department of Computer Science
University of North Carolina at Charlotte

Samira Shaikh SAMIRASHAIKH@UNCC.EDU
Department of Computer Science
University of North Carolina at Charlotte

Editor: Name Surname

Submitted MM/YYYY; Accepted MM/YYYY; Published online MM/YYYY

Abstract

One of the hardest problems in the area of Natural Language Processing and Artificial Intelligence is automatically generating language that is coherent and understandable to humans. Teaching machines how to converse as humans do falls under the broad umbrella of Natural Language Generation. Recent years have seen an unprecedented growth in the number of research articles published on this subject in conferences and journals both by academic and industry researchers. There have also been several workshops organized alongside top-tier NLP conferences dedicated specifically to this problem. All this activity makes it hard to clearly define the state of the field and reason about its future directions. In this work, we provide an overview of this important and thriving area, covering traditional approaches, statistical approaches and also approaches that use deep neural networks. We provide a comprehensive review towards building open domain dialogue systems, an important application of natural language generation. We find that, predominantly, the approaches for building dialogue systems use *seq2seq* or *language models* architecture. Notably, we identify three important areas of further research towards building more effective dialogue systems: 1) incorporating larger context, including conversation context and world knowledge; 2) adding personae or personality in the NLG system; and 3) overcoming dull and generic responses that affect the quality of system-produced responses. We provide pointers on how to tackle these open problems through the use of cognitive architectures that mimic human language understanding and generation capabilities.

Keywords: deep learning, language generation, dialog systems

1. Introduction

Language Generation is a sub-field of the field of Natural Language Processing (NLP), Artificial Intelligence (AI) and Cognitive Science (CS) that has been studied since the 1960s. NLG entails not only incorporating fundamental aspects of *artificial intelligence* but also *cognitive science* (Reiter and Dale, 2000). Yet, it is still one of the major challenges towards achieving Artificial General Intelligence (AGI).

In Figure 1, we provide an overview of the applications that can be categorized under the umbrella of language generation. In this work, we focus on the domain of dialogue systems that is fundamental to Natural User Interfaces (Gao et al., 2019).

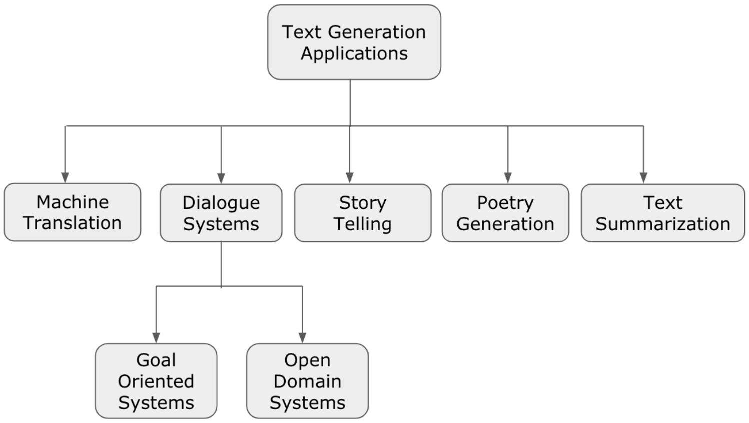

Figure 1: Categories in the sub-field of language generation.

Some of the early success in the field of language generation was building systems like Eliza (Weizenbaum, 1966) and PARRY (Colby, 1975). These systems generated language through a set of rules. However, such rule based systems were too constrained and brittle and could not be easily generalized to produce diverse set of responses. Other traditional NLG techniques generated text from structured data or from knowledge bases. Some examples are domain-based systems that produce weather reports (Angeli et al., 2010) and sports reports (Barzilay and Lee, 2004).

The field of text generation systems shifted from traditional approaches to statistical approaches where the focus was on exploiting patterns in text data and building models to make a prediction based on the text it has seen. Mikolov *et al.* (2010) argued that there had not been any significant progress in using statistical approaches to model language. This observation led to his experimentation on using recurrent neural networks (Mikolov et al., 2010) and achieved state-of-the-art results which set the wheels in motion for neural networks becoming a model of choice for modeling sequential data like text. Neural Networks belong to a class of machine learning models that are capable of identifying patterns in text and identify features that help solve different problems related to computer vision, object recognition, image captioning and speech recognition (Sutskever et al., 2014). Another phenomenon that suited the rise of neural networks is the large amount of corpora and significant computational resources that became available. In the applications of language generation, neural networks have helped achieve state-of-the-art results in problems related to machine translation (Bahdanau et al., 2014), story telling Holtzman et al. (2018), dialogue systems (Wolf et al., 2019; Xing et al., 2017; Dinan et al., 2018) and poetry generation (Zhang and Lapata, 2014).

However, even with the powerful performance of neural networks for developing dialogue systems, current systems still suffer from problems like dull and generic responses (Li et al., 2015), lack of encoding context (Serban et al., 2016; Sordoni et al., 2015) and lack of consistent persona (Li et al., 2016a). Most current dialogue systems and conversational models lack style, which can be an issue as users may not be entirely satisfied with the interaction. Generating personalized dialogues is another substantially difficult task as the generated response has to be contextually-relevant to the conversation, while also conveying accurate paralinguistic features (Niu and Bansal, 2018).

To make clear the directions towards which the field is heading, we produce a comprehensive overview of the field of open domain dialogue systems. Our primary goal is to identify the research gaps in the field and identify clear avenues for future research. While a few recent survey papers on this topic exist (Gatt and Krahmer, 2017; Gao et al., 2019), these do not identify the clear research gaps and also do not provide a comprehensive review of the field of open domain dialogue systems.

In summary, the purpose of this paper is to: a) provide an overview of the research in the field of natural language generation from traditional approaches to deep learning based approaches (which we cover in Section 2 and Section 3); b) to give a comprehensive overview of the field of open domain dialogue systems (which we summarize in Table 1 in Section 4,); and c) to propose avenues for future research for tackling these open problems (in Section 5).

2. Traditional Approaches to Language Generation

Reiter and Dale (2000) defined Natural Language Generation (NLG) as "*the sub-field of artificial intelligence and computational linguistics that is concerned with the construction of computer systems than can produce understandable texts in English or other human languages from some underlying non-linguistic representation of information*". They also presented a standard architecture for the developing NLG systems (Figure 2[1]) that comprised of six components each performing an important task to generate a coherent output.

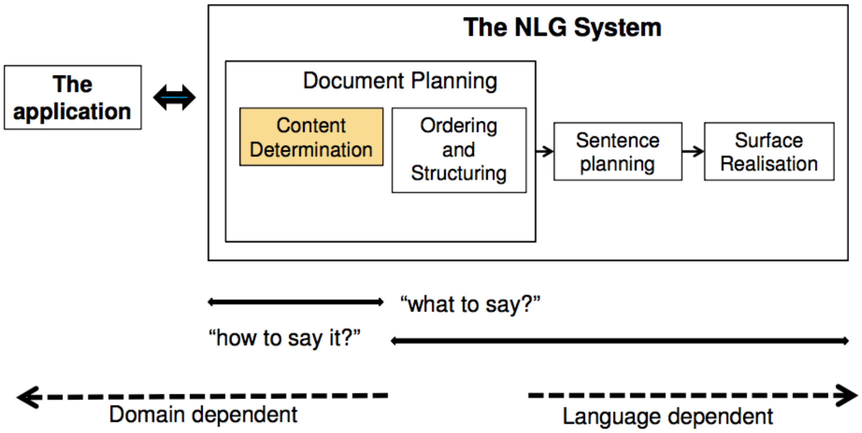

Figure 2: Three stage pipeline architecture proposed by Reiter and Dale focusing on the aspects of document planning, sentence planning and surface realization.

Their architecture was motivated by the fact that there were many NLG systems at the time for different applications but no well-defined, comprehensive architecture. Before the six stage pipeline, Reiter introduced a simple three stage pipeline of 1) content determination; 2) sentence planning and 3) surface realisation and named it the "consensus" architecture (Reiter, 1994). Cahill *et al.* (1999) conducted experiments and argued that the pipeline process was not detailed and the architecture was too constrained. In order to overcome the issues, the authors suggested a finer architecture based on the linguistic operations such as 1) lexicalisation; 2) referring expression generation and 3) aggregation (Cahill et al., 1999). One drawback of the architecture suggested by Cahill *et al.* (1999) was that no details were provided about how the systems get input and in what form. Reiter

[1] https://tinyurl.com/ydgyawvw

and Dale (Reiter and Dale, 2000) iterated on their initial architecture and suggested a new standard architecture for the NLG systems comprising of 4-tuple $\langle k, c, u, d \rangle$ where k is the knowledge source, c is the communicative model, u is the user model and d is the discourse theory (Evans et al., 2002) and the iterated model also implemented some of the aspects of Cahill *et al.* (1999) work into the architecture.

In the following sub-sections, we explain the functionality of the six components and extensive research work that has been carried out to address that component.

2.1 Content Determination

Content Determination is the problem of deciding the domain that is needed to generate text for a given input. Content determination is affected by communicative goals i.e., different communicative goals from different kinds of people may require different contents to be expressed by the system that satisfy the parties involved. Content determination is affected by the expertise of the end user and also by the content of the information source present in the system (Reiter and Dale, 2000).

The problem of content determination has been approached from two different perspectives: 1) Schemas or templates; and 2) Statistical data driven approaches.

Schema- or Template-based content determination methods focused on generating content by an analysis of the corpora and they are prominent in tasks while are standardised like weather forecast systems like FOG (Goldberg et al., 1994) where rhetorical relations can be encoded as schemas or schemata (McKeown, 1985). The schemata is made up of *identification, constituency, attributive and contrastive*. Each component of the schemata is used to describe different predicate patterns (McKeown, 1985). Schemas or Templates can be improved upon by using rule based approaches. Rule based approaches for the task of content determination have been used for domain specific systems where the implicit knowledge of the domain expert is used for more knowledge acquisition (Reiter et al., 2000; Elhadad and Robin, 1996). Reiter *et al.* (2000) list the different techniques such as sorting, thinking aloud, expert revision for knowledge acquisition in the STOP system that generated personalized smoking-cessation leaflets.

With the availability of more data, the process of **content determination became data-driven**. Duboue and McKeown (2003) developed a system that automated the process of producing constraints on every input and deciding if it should appear as a part of the output with the help of a two-stage process of exact matching and statistical selection, where the semantic data is clustered and text corresponding to each cluster is used to measure its degree of influence with regards to the other clusters. An alternative method was suggested by Barzilay and Lee (2004), so that content selection can be applied to domains where the knowledge base has not been provided by using a novel adaptation of Hidden Markov Models. In their method, the states of the Hidden Markov models correspond to the type of information characteristic to the domain of interest. Barzilay and Lapata (2005) suggested another method along similar lines, in which the content selection is treated as a collective classification problem by capturing the contextual dependencies between the input items.

Liang *et al.* (2009) extended the work done by Barzilay and Lapata, by describing a probabilistic generative model that combines text segmentation and fact identification in a single unified framework using Hidden Markov Models. They proposed a generative model consisting of three stages of selecting a set of records, identifying the fields from the records and choosing a sequence of words from the fields and each stage, optimized using Expectation Maximization (EM). The work done by Liang *et al.* (2009) proved instrumental in combining the process of content determination and linguistic realization into a unified framework. Another example is the work done by Angeli *et al.* (2010), where the process of generation is broken down into a sequence of local decisions and using a classifier on decisions that include choosing records from the database, choosing a subset of fields from records and choosing a template to render the generated text . However, Kim and Mooney (2010) identified a drawback with the method suggested by Liang *et al.* (2009) of just using bag of words and a simple Hidden Markov Model and not considering the context-free linguistic syntax. To address this issue, Kim and Mooney used a generative model with hybrid trees which expresses correspondence between the word in natural language and grammatical structure (meaning representation) and iterative generation strategy learning (a method similar to EM that iteratively improves probability to determine which event likely to be received as input from the human). Another example of content determination (in an end-to-end system) is the work done by Konstas and Lapata (2012), where a set of records are converted into probabilistic context free grammar that describes the structure of the database and the grammar is encoded as a weighted hypergraph. The generation process is based upon finding the best derivation of the hypergraph. In the next section, we will cover document structuring, the next sub-problem of language generation.

2.2 Document Structuring

The second sub-problem specified by Reiter and Dale (2000) is document or text structuring. This is the process of determining the order in which the text is to be conveyed back to the user once the content is determined. Document Structuring and Content Determination are closely linked.

A method which had a significant impact on addressing this problem was the understanding of discourse relations with the help of Rhetorical Structure Theory (RST) (Mann and Thompson, 1986). RST has four elements consisting of “relations” which identifies relationships between different parts of the text in the form of satellite and nuclei. Nuclei represents the important part of the text and satellite represents the supplementary part of the text, “schemas” defines patterns in a part of text can be analyzed with regards to other spans (nodes of a tree), “schemas application” and “structures” and help in creating coherent texts (Mann and Thompson, 1987).

Moore and Paris (1993) found issues with using RST when they tried to use the individual segments and rhetorical relations between segments to construct a text plan for their dialogue system. RST were not able to generate proper responses for follow up questions. Due to these problems with RST, Moore and Pollack (1992) suggested a two-level discourse analysis process. The first level is called “information level” which involves the relation conveyed between two sentences in a discourse and second level is called “intentional level” which deals with the discourse produced to effect change in the mental state of the participants (Moore and Pollack, 1992). Dimitromanolaki and Androutsopoulos (2003) used supervised machine learning to learn a new representation of document structuring task and applied this approach to for the task of document structuring for a specific domain. A lot of other researchers have interlinked the process of the text structuring and content determination into a single one which has been described in the previous subsection.

2.3 Lexicalization

Lexicalization or the task of choosing the right words to express the contents of the message is the third sub-problem defined by Reiter and Dale (2000). They broke down the task of lexicalization into two categories, namely, Conceptual Lexicalization and Expressive Lexicalization. Conceptual Lexicalization is defined as converting data into linguistically expressible concepts and Expressive

Lexicalization is how lexemes available in a language can be used to represent a conceptual meaning (Reiter and Dale, 2000). In order to solve the problem choosing the best lexeme to realize the meaning, Bangalore and Rambow (2000) suggested using a tree representation of the syntactic structure and an independently hand-crafted grammar. One of the drawbacks of this method was not using a part-of-speech tagger and using a mechanism of making a union of all the synonyms from the synset. While the traditional approaches to NLG view the process of lexicalization as belonging to the sentence planning phase along with the process of sentence aggregation and referring expression generation, however, recent research in NLG views lexicalization as the part of the linguistic realization phase (Gatt and Krahmer, 2017).

2.4 Referring Expression Generation

Referring Expression Generation (REG) is the fourth sub-problem defined by Reiter and Dale (2000) and it is aggregated with the sentence planning phase of the architecture. REG is the ability to produce a description of an entity and distinguish it from the other domain entities (Reiter and Dale, 2000). An entity might be referred to in many different ways. For example, consider the following sentence, `Adrian arrived late to an event and he missed a majority of it`. There can be two ways in which an entity can be referred to. The first is the *initial reference* (`Adrian`) in the example) when the entity is brought into the discourse and the other is *subsequent reference* (`he`) in the example) which refers to entity after it has been introduced in discourse (Reiter and Dale, 2000). The first step of the solution suggested by Reiter and Dale (2000) is to identify the type of reference for the target, such as pronoun or description or proper name. The identification of proper names is the easiest, while identification of pronouns can be based on a rules such as “the target is referred to in the previous sentence and if the sentence contained no other entity of the same gender” (Krahmer and Van Deemter, 2012).

There are multiple existing algorithms for the task of REG. Dale (1989) created the **Full Brevity** algorithm that generates very short descriptions referring expression by the identification of target and distractors. However, this algorithm suffered from major drawbacks such as being able to only generate short referring expressions and computing these short expressions had a high complexity (NP-hard)(Krahmer and Van Deemter, 2012). An improvement over the Full Brevity was the **Greedy Heuristic** algorithm, which picks a property of target that rules out most of the distractors (words that do not co-reference with the target) and adding that property to the description (Dale, 1992). Greedy algorithm was later eclipsed in terms of performance by the **Incremental Algorithm** (IA). The Incremental Algorithm sequentially picks the properties and then rules out the distractors until a distinguishable expression is generated (Dale and Reiter, 1995). However, these description generated may contain redundant properties which becomes a drawback of incremental algorithm.

To address these drawbacks, Kees and Van Deemter (2002) explored how the incompleteness of IA could be overcome with the help of a two stage algorithm to generate boolean descriptions. The first stage is the process of generalization of the IA by taking a union of the properties that help in singling out the target set and the next stage was to optimize the expressions produced (Van Deemter, 2002; Krahmer and Van Deemter, 2012). One of the issues this work failed to address the notion of vagueness which was addressed in the work done by Horacek (2005). Horacek (2005) introduced measures including the following to represent the uncertainties: p_k - the user is acquainted with the terms mentioned, p_p- the user can perceive the properties uttered, p_A - the user agrees with the applicability of the terms used. With the help of these three probabilities, the probability of recognition

p is calculated as the product of the three probabilities and this helps in distinguishing vagueness along with misinterpretation and ambiguity (Horacek, 2005). Later, Khan *et al.* (2008) addressed the issue of structural ambiguity in coordinated phrases in the form of "Adjective Noun1 Noun1" to determine if the Adjective was associated with Noun1 or Noun2. Khan *et al.* (2008) conducted user studies and suggested how the generator can avoid these issues. However, Engonopoulos and Koller (2014) argued that the listeners might misunderstand the generated expression. To address these concerns, Engonopoulos and Koller (2014) proposed an algorithm to maximize the likelihood that a referring expression is understood by the user with the help of a probabilistic referring expression model $P(a|t)$, where t refers to the expression and a to the object in the domain. In the next subsection, we will cover the aspects of sentence aggregation which is dependent on the capability of REG algorithms.

2.5 Sentence Aggregation

Sentence aggregation is characterized as the process of removing redundant information during the generation of discourse without losing any information and to produce text in a concise, fluid and readable manner (Dalianis, 1999). Dalianis, in his survey suggested that aggregation can be done in all the stages of the NLG process except during content determination and surface realization. Reiter and Dale marked this subproblem as belonging to the sentence planning or microplanning phase (Reiter and Dale, 2000). Reiter and Dale characterized the problem of aggregation to be closely interlinked with lexicalization as both deal with understanding the knowledge source and linguistic elements of words, phrases and sentences (Reiter and Dale, 2000).

One of the initial approaches to tackle the problem of sentence aggregation was put forward by Cheng and Mellish (2000) by using Genetic Algorithms, where they used a constraint-based program with a preference function to evaluate the coherence of a text. Walker *et al.*. (2001) used a data-driven approach to overcome the issue of using a hand-crafted preference function used by Cheng *et al.* (2000). In their work, they used two phases; the first phase generated a large sample of sentences for an input and the next phase ranked the sentences with the help of rules generated from training data. Barzilay and Lapata (2006) presented an automatic method to learn the grouping constraints with the help of a parallel corpus of sentences and their corresponding database entries by looking at the number of attributes shared by the entries. In the next section, we cover the aspect of linguistic realization that is the final stage of the pipeline and the different mechanisms that operate on the work done win earlier stages of the pipeline.

2.6 Linguistic Realization

Linguistic Realization was characterized by Reiter and Dale as the task of ordering different parts of a sentence and using the right morphology along with punctuation marks which is governed by rules of grammar to produce a syntactically and orthographically correct text (Reiter and Dale, 2000). In this section, we will cover three approaches for linguistic realization.

2.6.1 HAND-CODED GRAMMAR-BASED SYSTEMS

Grammar-based NLG systems are systems that make their choice depending on the grammar of the language, which can be manually written with the help of multilingual realizers. An example of multilingual realizer is KPML, developed by Bateman (Gatt and Krahmer, 2017; Bateman, 1997), that depended on the systemic grammar to help understand the syntactic characteristic of a sentence.

Another popular realizer, SURGE, was developed by Elhadad and Robin (1996), based on functional unification formalism. Another popular realizer was called Halogen, which was introduced by Langkilde (2002). This system uses a small set of hand-crafted grammar rules as features to generate alternative representations. A downside of using these realizers is that they are complicated to use and have a steep learning curve for the users, which made the NLG community move towards simple realization engines.

2.6.2 TEMPLATES

Templates are often used in systems which require limited syntactic variability in their output (Reiter and Dale, 1997). Consider the template [*person*] `is leaving` [*country*] and in this scenario `person` and `country` values can be replaced by the system during output phase. One of the issues with template-based NLG systems is lack of flexibility of the templates to produce a diverse set of generated texts. McRoy *et al.* (2003) suggested a method to overcome these issues with the help of declarative control expressions to augment traditional templates. Van Deemter *et al.* (2005) argued that as new NLG systems have been developed, the differences between standard NLG systems and template-based systems have blurred as the modern systems use handcrafted grammars to help with realization. Another disadvantage of using templates is the need for knowledge expertise to construct templates for the system (McRoy et al., 2003; Gatt and Krahmer, 2017). Angeli *et al.* (2012) used a probabilistic approach and compositional grammar to learn the rules for parsing time expressions. Kondadadi *et al.* (2013) used *k*-means clustering to create template banks derived using named entity tagging and semantic analysis. Despite the advantage of using template based methods, most of the recent NLG systems have moved to a statistical-based approach.

2.6.3 STATISTICAL APPROACHES

Statistical approaches have been used in NLG systems in order to reduce the manual effort of using hand written grammar rules and to deal with large corpora to acquire probabilistic grammar to get better realizations of text. The work by Langkilde (2000) was one of the seminal works in using statistical approaches towards linguistic realization. In this approach, Langkilde (2000) used corpus based statistical knowledge and a small hand crafted grammar to generate many different representations of a sentences that were packed in the form of forest of trees. Langkilde (2000) ranked each phrase by calculating a score which was decomposed into a internal and external score, former known to be context independent and latter was context dependent. This method introduced by Langkilde served as the base for subsequent research in this field.

Another important work was carried out by Langkilde and Knight (1998), to build a generator by computing word lattices from meaning representations by introducing new grammar formalisms. Bangalore and Rambow (2000) suggested improvements by introducing a tree-based model of syntactic representation along with independently hand-crafted grammar rules to improve to performance of the syntactic choice module. Cahill *et al.* (2007) presented a different method to rank and suggested using a log-linear ranking system, and they show that log-linear ranking obtained better performance than existing systems.

One major downside of these approaches listed above is that the they are computationally expensive, as they generate a lot of possible sentence and then do the filtering with the help of the ranking mechanism. To overcome this drawback, Belz and Anja (2008) introduced the Probabilistic

Context-free Representationally Underspecified (pCRU) which uses probabilistic choice to inform generation instead of listing all the choices and then selecting a phrase.

The approaches described above all use a set of hand-crafted rules as the base generation and only use statistical method for the filtering the output. An alternative would be to apply statistical approaches on the base-generation systems. There have been approaches where grammatical rules have been derived from treebanks (Gatt and Krahmer, 2017). Hockenmaier and Steedman (2007) presented a method to extract dependencies and combinatory categorical grammar(CCG) from the Penn Treebank corpus.

Having given an overview of the traditional methods used for NLG and also the methods to address the subcomponents of the language generation process, in the next subsection we cover deep neural networks and the recent surge in these architectures towards solving natural language generation problem.

3. Deep Learning approaches for Language Generation

Applying deep neural networks to Natural Language Processing has helped achieve state-of-the-art performance across different tasks, including the task of language generation due to the capability of neural networks to learn representations with different levels of abstraction (LeCun et al., 2015; Goldberg, 2016). The simplest and most widely used type of neural network is the feed forward neural network or multilayer perceptron (Rosenblatt, 1958) in which the data flow is in one direction and feed forward neural networks are acyclic graph structures. Bengio *et al.* (2003) demonstrated the ability of feed forward neural networks on language modeling tasks.

Another type of neural network architecture that is more suited for dealing with sequential data is the Recurrent Neural Network (RNN) architecture. RNNs are used for the processing of sequential data with the help of recurrent connections that perform the same task over every sequence (Goodfellow et al., 2016). RNNs have the capability to handle long sequences using the knowledge gained (**"memory"**) from previous sequence computations unlike networks without sequence-based specialization. Application of memory to neural networks was demonstrated as early as 1982, through the Hopfield Network that was used to store and retrieve memory from a pre-trained set of patterns or memories, similar to the human brain. The network relied on neurons each producing a value of +1 or -1 depending on the input from the previous layer (Hopfield, 1982).

Hopfield's network was the inspiration behind Jordan's network (Jordan, 1986) (represented in Figure 3A), for doing supervised learning on sequences with the help of a single hidden layer and special units which receive input from the output unit which then forwards the values to the hidden nodes (Lipton et al., 2015a). Elman simplified the Jordan's architecture (represented in Figure 3B), by adding a context unit with each hidden unit receiving its input from the units at the previous time step. Elman showed that network can learn dependencies by training the network on sequence of 3000 bits. The model achieved an accuracy rate of 66.7% on predicting the third bit in the sequence (Elman, 1990; Lipton et al., 2015a).

The Elman architecture played a substantial role in the discovery of long short term memory networks (LSTM) (Hochreiter and Schmidhuber, 1997). LSTMs helped in tackling the important problems of vanishing and exploding gradients caused by backpropagation while training the neural networks (Schmidhuber, 2015). During backpropagation, the neural network weights receive an update proportional to the gradient of the error function. These gradients are multiplied across layers and sometimes the gradients become too small or vanish and in certain cases the gradients

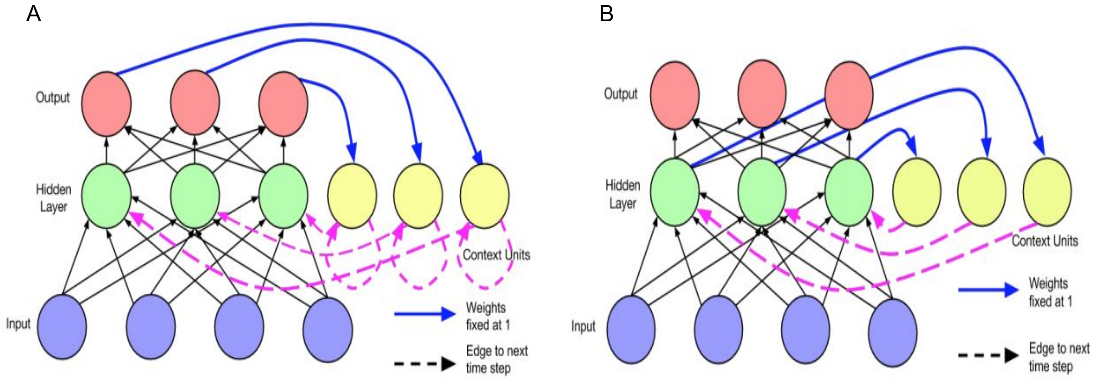

Figure 3: A. Represents the Jordan Architecture. B. Represents the Elman Architecture (Figure credit: Lipton *et al.*, (2015a))

grow exponentially and explode. LSTM replaced the hidden units of the neural networks with a new concept called memory cell, which is built around a central linear unit (internal state) with a fixed self connection to ensure that the gradients can pass without exploding or vanishing. The memory cell also contains an input and output gate; later the forget gate was added to the structure of the memory cell by Gers *et al.* (1999). Gates are regulating structures that carefully allow the information to the internal state to be added or removed.

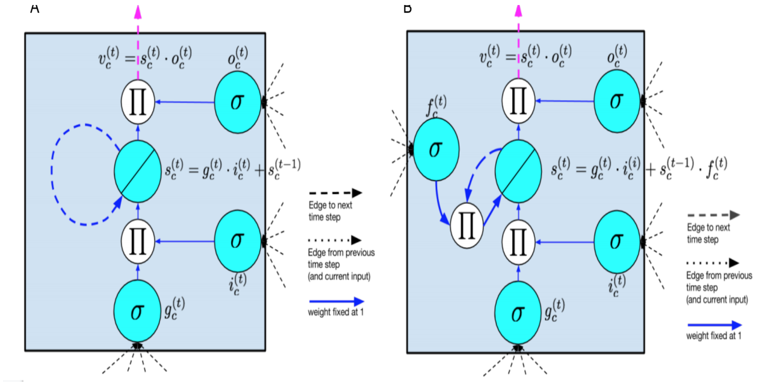
A
$v_c^{(t)} = s_c^{(t)} \cdot o_c^{(t)}$
$o_c^{(t)}$
$\prod$
σ
$s_c^{(t)} = g_c^{(t)} \cdot i_c^{(t)} + s_c^{(t-1)}$
$i_c^{(t)}$
$g_c^{(t)}$
Edge to next time step
Edge from previous time step (and current input)
weight fixed at 1
B
$v_c^{(t)} = s_c^{(t)} \cdot o_c^{(t)}$
$o_c^{(t)}$
$f_c^{(t)}$
$s_c^{(t)} = g_c^{(t)} \cdot i_c^{(i)} + s_c^{(t-1)} \cdot f_c^{(t)}$
$i_c^{(t)}$
$g_c^{(t)}$
Edge to next time step
Edge from previous time step (and current input)
weight fixed at 1

Figure 4: A. Represents the LSTM Architecture by Horcheiter *et al.* (1997). B. Represents the updated LSTM Architecture by Gers *et al.*(1999). (1) is the input node, (2) is the input gate, (3) is the output gate, (4) is the cell state, (5) is the forget gate. (Figure credit: Lipton *et al.*, (2015a)

We describe the various components shown in Figure 4 next:

- Input node –The input nodes takes in the input from current layer $x^{(t)}$, t represents the current time step and also takes in the value from the hidden layer at the previous time step $h_{(t-1)}$ and the weighted sum input is taken is passed through an activation function "sigmoid", which was replaced by "tanh" as the LSTM architecture was improved.

$$g_c^{(t)} = \sigma(W_g.[x^{(t)}, h_{(t-1)}] + b_g) \tag{1}$$

- Input gate –The input gate takes the input from the current layer $x^{(t)}$ and value from the hidden layer at the previous time step $h_{(t-1)}$ and applies a "sigmoid" activation function to the weighted sum. A sigmoid is used as the gate in this situation to make sure that any value that is a 0 then the corresponding value from the input gate is also cut off and cannot affect the internal state update.

$$i_c^{(t)} = \sigma(W_i.[x^{(t)}, h_{(t-1)}] + b_i) \tag{2}$$

- Forget gate –The forget gate was added to the LSTM architecture to overcome a limitation of the the cell state growing linearly and when presented with a continuous stream the cell state might grow in an unbounded station (Gers et al., 1999). The main job of the forget gate is to provide with a way to reset the contents of the cell state.

$$f_c^{(t)} = \sigma(W_f.[x^{(t)}, h_{(t-1)}] + b_f) \tag{3}$$

- Cell State –The cell state is the heart of the memory cell and carries information that it has maintained until the current time step t so that the loss function is not only dependent on the data from the current time step.

$$s_c^{(t)} = s_c^{(t-1)} \times f_c^{(t)} + g_c^{(t)} \times i_c^{(t)} \tag{4}$$

- Output Gate –The output gate takes the input from the current layer $x^{(t)}$ and value from the hidden layer at the previous time step $h_{(t-1)}$ and applies a sigmoid activation function to the weighted sum. A sigmoid is used as the gate in this situation to determine what values of the cell part of the cell state is to output.

$$o_c^{(t)} = \sigma(W_o.[x^{(t)}, h_{(t-1)}] + b_o) \tag{5}$$

- Output Node –The final output of the LSTM cell is obtained after passing the cell state through a tanh activation function and multiple it with the contents of the output gate.

$$v_c^{(t)} = o_c^{(t)} \times \tanh(s_c^{(t)}) \tag{6}$$

Another variant of RNN model called **GRU** (See Fig 5 was introduced Cho *et al.*(2014a), inspired by the functionality of the LSTM. A performance comparison between the LSTM and GRU were conducted by Chung *et al.* (2014) who found the performance between the two to be comparable. The GRU consists of two gates, reset (Eq 7) and update gates (Eq 8) and exposes the whole state each time without having a mechanism to control it. The reset helps the hidden unit forget

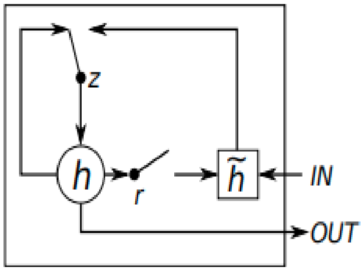

Figure 5: GRU architecture by Cho *et al.*

information not needed and the update gate controls how much information is carried forward from the previous hidden state. The actual activation is computed as a linear interpolation of the previous activation and candidate activation (Eq 9).

$$r_j = \sigma([W_r.x]_j + [U_r.h_{(t-1)}]_j \tag{7}$$

$$z_j = \sigma([W_z.x]_j + [U_z.h_{(t-1)}]_j \tag{8}$$

$$\begin{aligned} h_t^j &= (1 - z_t^j).h_{t-1}^j + z_t^j.\tilde{h}_t^j \\ \tilde{h}_t^j &= tanh(W.x_t + U(r_t \odot h_{t-1}))^j \end{aligned} \tag{9}$$

In the next four subsections, we list the different approaches such as language modeling, encoder-decoder, memory networks and transformer models based approaches that have been applied to the task of language generation.

3.1 Language Models

Language models are probabilistic models that are capable of predicting the next word given the preceding words in a sequence. Language models are widely used in the generative modeling tasks. The ability of language models to model sequential data of fixed length context using feed forward neural networks was first demonstrated in the work done by Bengio *et al.* (2003). However, a major drawback of the approach, which is the usage of fixed length context, was overcome in the seminal work done by Mikolov *et al.* (2010) demonstrating the efficiency of RNN based language models. Similarly, another seminal work in the area of language models is the work done by Sutskever *et al.* (2011) demonstrating the effectiveness of LSTM in predicting the next character of a sequence. Conditional language models are also used as a variant of language models where the language model is conditioned on variables other than the preceding words, like the work done by generating product reviews based on sentiment, author, item or category (Lipton et al., 2015b) or generating text with emotional context (Ghosh et al., 2017).

3.2 Encoder-Decoder Architecture

Another important architecture that enhanced the task of language generation was the usage of two RNNs in an end-to-end model (Figure 6) (Cho et al., 2014b) that overcame a significant limitation where the neural networks could only be applied to problems where input and target can be encoded with fixed dimensionality. The **encoder** converts the input sequence into a fixed vector representation c by Eq 10 where h_t refers to hidden state at time step t, f represents any non-linear function and x represents the input sequence. The **decoder** tries to predict sequence of symbols with the help of the context vector c. The hidden state of the **decoder** depends on the context vector c and is represented by Eq 11 and next symbol to be predicted is based on a condition probability 12 where g is a softmax function.

$$h_{(t)} = f(h_{(t-1)}, x_t) \tag{10}$$

$$s_i = f(s_{i-1}, y_{i-1}, c) \tag{11}$$

$$P(y_t | y_{t-1}, y_{t-2}, ..., y_1, c) = g(s_{(t)}, y_{t-1}, c) \tag{12}$$

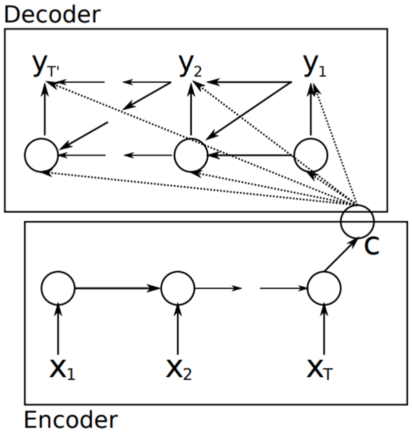

Figure 6: Encoder-Decoder architecture proposed by Cho *et al.* (Figure credit: Cho *et al.*(2014b).)

Along similar lines to the work done by Cho *et al.* (2014b), ***seq2seq*** was introduced by Sutskever *et al.* (2014) which uses two LSTMs, one to map the input sequence to a fixed vector and the other RNN to decode the fixed vector into a sequence of target symbols of varying lengths. A key difference between the work done by Cho *et al.* (2014b) and Sutskever *et al.* (2014) was the discovery that reversing the order of the input sequence improves the performance of the model and also helps with creating short term dependencies between input and target sequence. Bahdanau *et al.* (2014) identified the bottleneck caused by encoding the entire sequence into a fixed vector in the simple encoder-decoder architecture and proposed a modification which allows the decoder to *attend* to different parts of the source sentence that are relevant for predicting the next word or character of the sequence. In the attention mechanism, the context vector c_i is calculated as the weighted combination of all the encoder hidden states (see Eq.13) and α refers to how much importance should be given to respective input states.

$$\begin{aligned} c_i &= \sum_{j=1}^{T_x} \alpha_{ij} h_j \\ \alpha_{ij} &= \frac{exp(e_{ij})}{\sum_{k=1}^{T_x} exp(e_{ik})} \\ e_{ij} &= a(s_{i-1}, h_j) \end{aligned} \tag{13}$$

A majority of the work done for the task of language generation was done using the encoder-decoder architecture. Zhang and Lapata (2014) proposed a model for Chinese poetry generation with the help of RNN. In their work, they combined the process of content determination and realization was jointly into one joint process.

Another example was the NLG system developed by Wen *et al.* (2015), who modified the architecture of the LSTM to constrain it semantically and be able to predict the next utterance in a dialogue context. The architecture of the modified LSTM cell was used for surface realization and the dialogue act cell which acts similar to the memory cell was used for the sentence planning phase. Along similar lines was the work by Goyal *et al.* (2016), who presented a character-level RNN for dialogue generation and addressed the issue of delexicalization and sparsity. Mei *et al.* (2015) used the encoder-decoder aligner architecture to perform the task of content selection and realization on a set of weather database event records as a joint task. The aligner is based on the attention mechanism (Xu et al., 2015; Bahdanau et al., 2014). The encoder-decoder architecture was also used to generate emotional text as demonstrated by the work done by Asghar *et al.*, (2017), Zhou *et al.*, (2018) and Ke *et al.*, (2018).

3.3 Memory Networks

Memory networks, a type of learning model, were introduced by Weston *et al.* (2014) to overcome to short memory encoded in the hidden states. These networks were used for a variety of question-answering tasks where the answer is generated from a set of facts fed into the model. The answer generated by the model can be a one-word answer or paragraph of text. The memory networks introduced by Weston *et al.* (2014) had four major components: input feature map, generalisation, output feature map and a response. Kumar *et al.* (2016) introduced a different type of memory network, based on episodic memory and were able to solve a wider range of question answering tasks and also on questions related to part of speech and sentiment analysis. The work done by Kumar *et al.* (2016) was extended for visual question answering by Xiong *et al.* (2016). Other works on visual question answering included the work done on using hierarchical attention on question-image pairs (Lu et al., 2016)), using relational networks for generating answers for visual question answering (Santoro et al., 2017) and using facts for visual question answering (Wang et al., 2018).

3.4 Transformer Models

The Transformer models (Figure 7) introduced by Vaswani *et al.* (2017) have helped achieve improvements over a wide range of NLP tasks. Transformer models are based on attention mechanisms, that draw global dependencies between the input and output. The transformer is a made of the encoder-decoder architecture but each encoder is a stack of six encoders with each encoder containing a self-attention and point-wise fully connected feed forward neural networks. The decoder

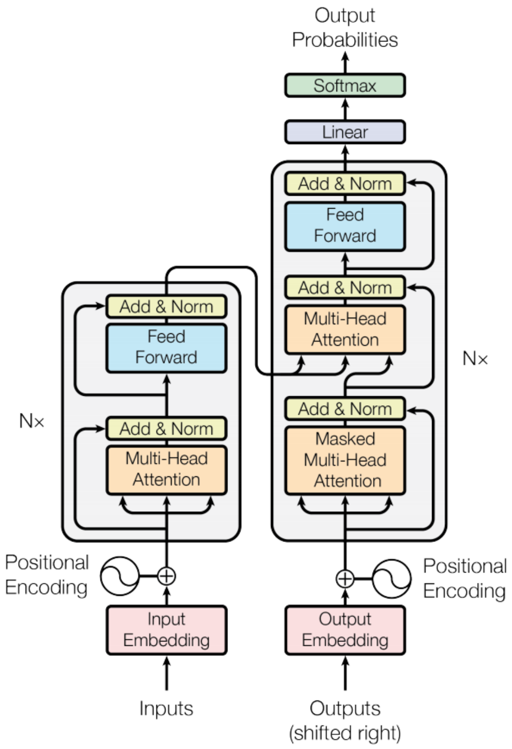

Figure 7: Transformer architecture as represented in Vaswani *et al.* (Figure credit: Vaswani *et al.* (2017)

is also a stack of six decoders with each decocder containing the same components as the encoder, but with an additional attention layer that helps the decoder focus on relevant parts of the input sentence. Work using transformer models is still in its infancy. Radford *et al.*, (2018) and Devlin *et al.* (2018) showed impressive results on several NLP tasks. Their work improved the existing state-of-the-art across a wide range of tasks such as language modeling, children's book test, reading comprehension, machine translation, question answering, modeling long range dependencies (LAMBADA), Winograd Schema challenge and summarization.

4. Open Domain Dialogue Systems using Deep Learning

Dialogue systems or conversational agents (CA) are designed with the intention of generating meaningful and coherent responses that are easy to respond to and informative when the system is engaged in a conversation with humans. A good dialogue model incorporated in conversational agents should be able to generate dialogues with high similarity to how humans converse (Li et al., 2017). Conversational agents are of great importance to a large variety of applications and can be grouped under two major categories, namely, (1) Closed Domain goal-oriented systems that help users achieve a particular goal, (2) Open Domain conversational agents engaging in a conversation with a human –also referred to as chit-chat models. Work on building end-to-end systems using neural networks (Vinyals and Le, 2015; Shang et al., 2015) has been increasingly published in recent years, and is the primary focus of this section.

With the fast paced advancement of research in this area, we find there is a lack of a comprehensive survey particularly in the area of the **open domain dialogue systems**. To address this gap in research, we summarize research done in this field by analyzing all the papers published in top conferences from 2015. We focus on the key aspects of these papers to summarize current trends: **corpora used, architecture implemented, optimization strategy used, evaluation metrics to evaluate efficacy**.

These are summarized in the columns in Table 1 and we observe the following trends:

Corpora refers to the language data that has been used in the paper. The most commonly used corpora are *Open Subtitles, Twitter Conversation Dialogues, Movie Triples, Cornell Movie Dialogues*. More recently, new datasets such as *PERSONA chat dataset, Reddit dataset* have been made available to the community.

Architecture gives an overview of the type of architecture used in the paper. Most of the research done in this field, have used variation of *seq2seq* models with attention mechanism. More recently, with the creation of the transformer models, researchers have started using this architecture for the open domain dialogue systems but the work is still in its infancy.

Evaluation is one of the most important aspect of open-domain dialogue systems. This is still an open research problem as there exist no appropriate or standardized metrics for evaluating performance. Researchers have primarily relied on adapting automated metrics such as BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005) and embedding-based metrics Shen et al. (2017); Lowe et al. (2017) for validating the performance. However, research has shown that these metrics show little to no correlation with evaluation from humans (Novikova et al., 2017; Lowe et al., 2017). We find that human evaluation is another primary evaluation metric that exists in this field and researchers use different criteria such as *Semantic Relevance, Appropriateness, Interestingness, Fluency, Grammar*. These are listed in Table 1. When there is no criteria presented in the paper, we simply list Human Evaluation. This refers to the cases where the researchers asked humans to simply judge which response is better between the generated and the ground-truth response.

Table 1: Summary of deep learning-based open-domain dialogue systems (from 2015 to present) providing an overview of the corpus used, architecture and optimization strategy implemented and evaluation metrics used in the paper.

Authors	Corpora	Architecture	Optimization	Evaluation Metrics
Vinyals and Le (2015)	• Open Subtitles • IT Help Desk	• Seq2Seq	• Cross Entropy	• Human Evaluation
Sordoni et al. (2015)	• Twitter Conversation Dialogue	• Language Model	• Adgrad	• BLEU • METEOR • Human Evaluation
Li et al. (2015)	• Twitter Conversation Dialogues • Open Subtitles	• Seq2Seq	• SGD	• BLEU • Distinct-1 • Distinct-2 • Human Evaluation

Shang et al. (2015)	• Weibo Conversation	• Seq2Sseq + Attention	N/A	• Human Evaluation – Grammar & Fluency – Logic Consistency – Semantic Relevance – Scenario Dependance – Generality
Yao et al. (2015)	• Helpdesk chat service	• Seq2Seq + Attention + Intention Network	N/A	• Perplexity
Serban et al. (2016)	• Movie Triples	• Hierarchical Encoder Decoder	• Adam	• Perplexity
Li et al. (2016a)	• Twitter Persona dataset • Twitter Conversation Dialogues • TV Series Transcripts	• Seq2Seq + Persona Embeddings	• MMI	• Perplexity • BLEU • Human Evaluation
Luan et al. (2016)	• Ubuntu Dialogues	• Language Model + LDA	• SGD	• Perplexity • Response Ranking
Li et al. (2016b)	• Open Subtitles	• Seq2Seq + RL	• MMI + Policy Gradient	• BLEU • Dialogue Length • Diversity • Human Evaluation
Dušek and Jurčíček (2016)	• Public Transport Information	• Seq2Seq + Attention + Context Encoder	• Cross Entropy	• BLEU • NIST • Human Evaluation
Mou et al. (2016)	• Baidu Teiba Forum	• Seq2Seq	• SGD	• Human Evaluation • Length • Entropy
Asghar et al. (2016)	• Cornell Movie Dialogues	• Seq2Seq + Online Active Learning	• Cross Entropy	• Human Evaluation • Syntactic Coherence • Relevance • Interestingness

Serban et al. (2017b)	• Twitter Conversation Dialogues • Ubuntu Dialogues	• Latent Variable Hierarchical Encoder Decoder	• Adam	• Human Evaluation • Length • Entropy
Mei et al. (2017)	• Movie Triples • Ubuntu Dialogues	• Language Models + Attention + LDA Reranking	• Adam	• Perplexity • Word Error Rate • Recall • Distinct-1 • Human Evaluation
Xing et al. (2017)	• Baidu Teiba Forum	• Seq2Seq + LDA + Joint Attention	• Adadelta	• Perplexity • Distinct-1 • Distinct-2 • Human Evaluation
Cao and Clark (2017)	• Open Subtitles	• Variational Autoencoder	• MMI	• Human Evaluation
Lewis et al. (2017)	• Negotiation dataset	• Seq2Seq + self play + RL	• SGD	• Human Evaluation – Score – Agreement – Pareto Optimality • Perplexity
Li et al. (2017)	• Open Subtitles	• GAN	N/A	• Human Evaluation • Adversarial Evaluation
Qian et al. (2017)	• Weibo Dataset • Profile Binary Subset • Profile Related Subset • Manual Dataset	• Encoder Decoder + Profile Detector	• SGD	• Human Evaluation – Naturalness – Logic – Semantic Relevance – Correctness – Consistency – Variety • Profile Detection • Position Detection

Qiu et al. (2017)	• Chatlog Online Customer Service	• Attentive Seq2Seq + IR + Rerank	N/A	• Precision • Recall • F1 score • Human Evaluation
Serban et al. (2017a)	• Ubuntu Dialogues • Twitter Conversation Dialogues	• MrRNN	• Adam	• Human Evaluation
Shen et al. (2017)	• Ubuntu Dialogues	• Hierarchical Encoder Decoder	• KL Divergence	• Human Evaluation – Grammaticality – Coherence – Diversity • Embedding Evaluation – Greedy – Average – Extrema
Tian et al. (2017)	• Baidu Teiba Forum	• Hierarchical Encoder Decoder	• AdaDelta	• BLEU • Length • Entropy • Diversity
Bhatia et al. (2017)	• Yik Yak Dataset	• Seq2Seq + Locations • Seq2Seq + User model	N/A	• Perplexity • ROUGE
Ghosh et al. (2017)	• Fisher English Training Speech Parts • Distress Assessment Interview • SEMAINE Dataset • CMU-MOSI Dataset	• Language Model	N/A	• Perplexity • Human Evaluation
Kottur et al. (2017)	• Movies-DiC Dataset • TV Series Transcripts • Open Subtitles	• Context-aware Persona based Hierarchical Encoder Decoder	• Adam	• Perplexity • Recall@1 • Recall@5
Xing et al. (2018)	• Douban Group Dataset	• Hierarchical Recurrent Attention Network	N/A	• Perplexity • Human Evaluation

Zhou et al. (2018)	• NLPCC Dataset • STC Dataset • Weibo Emotion Dataset	• Encoder Decoder + External Memory + Internal Memory + Emotion Embedding	• Cross Entropy	• Human Evaluation – Content – Emotion • Perplexity • Accuracy
Asghar et al. (2018)	• Cornell Movie Dialogues	• Seq2Seq + Affective Embeddings	• Cross Entropy • Min Affective Dissonance • Max Affective Dissonance • Max Affective Content	• Human Evaluation – Syntactic Coherence – Naturalness – Emotional Appropriateness
Zhang et al. (2018a)	• STC Dataset	• Specificity Controlled Seq2Seq	• Adam	• BLEU-1 • BLEU-2 • Distinct-1 • Distinct-2 • Average Embedding • Extrema Embedding
Mazaré et al. (2018)	• Reddit Dataset	• Transformer + Persona + Context + Response Encoder	• Adamax	• Hits@1

Zhang et al. (2018b)	• PERSONA Chat Dataset	• Baseline Ranking Models • Ranking Profile Memory Network • Key-Value Memory Network • Seq2Seq • Generative Profile Memory Network	N/A	• Human Evaluation – Fluency – Engagingness – Consistency – Persona Detection • Perplexity • Hits@1
Rashkin et al. (2018)	• Empathetic Dialogues	• Transformer Model	• Adamax	• Perplexity • Avg BLEU • P@1 • Human Evaluation – Empathy – Relevance – Fluency
Huang et al. (2018)	• Open Subtitles • CBET	• Seq2Seq	• Adam	• Accuracy
Niu and Bansal (2018)	• Stanford Politeness Corpus • Stack Exchange	• Seq2Seq • Fusion model (Seq2Seq + polite-LM) • Label fine tune Model • Polite-RL	• Adam	• Perplexity • Perplexity@L • Word Error Rate • Word Error Rate@L • BLEU-4 • Human Evaluation – Politeness – Quality
Chen et al. (2018)	• Ubuntu Dialogues • Douban Conversation • JD Customer Service	• Hierarchical Variational Memory Network	• Adam	• Human Evaluation – Appropriateness – Informativeness • Embedding Evaluation – Average – Greedy – Extrema

Ghazvininejad et al. (2018)	• Twitter Conversation Dialogues • Four Square	• Seq2Seq + World Facts + Contextual Facts	• Adam	• Perplexity • BLEU • Diversity • Human Evaluation – Informativeness – Appropriateness
Young et al. (2018)	• Twitter Conversation Dialogues	• Tri-LSTM Encoder	• SGD	• Recall@k
Dinan et al. (2018)	• Wizards of Wikipedia	• Retrieval Transformer Memory Network • Generative Transformer Memory Network	• NLL	• Recall@1 • Perplexity • Human Evaluation – Engagingness
Wolf et al. (2019)	• PERSONA Chat Dataset	• Transformer Model	• Adam	• Perplexity • Hits@1 • F1 Score
Zheng et al. (2019)	• PERSONALDIALOG Dataset	• Seq2Seq + Personality Fusion	• Adam	• Perplexity • Distinct-1 • Distinct-2 • Accuracy • Human Evaluation – Fluency – Appropriateness

5. Conclusion

In this work, we summarized the work done in the area of language generation, starting from traditional approaches through recent work using deep learning approaches. Even with the rapid advancement in this sub-field of natural language generation for open domain dialogue systems, many of the approaches were based on the historical findings and prior research. We provided a summary of the important contributions by the standard Reiter and Dale (Reiter and Dale, 2000) architecture and provided explanations and body of research conducted to address the six different components of the architecture.

Another important aspect of this summarized work is identifying potential research gaps that persist in the field of conversational agents. We propose that tackling them would advance the field further.

5.1 Open Challenges for Open-Domain Dialogue Systems

Even though prior work done in the area of the open domain dialogue systems have helped advance the field, there are specific issues that affect their quality. We identify three main issues:

1. **Encoding Context** - Encoding contextual information such as world facts from knowledge bases or previous turns of the conversation are important issues to ensure that the conversational agent has enough information to produce a coherent, informative and novel response that is in tune with the context of the conversation. From Table 1, we find that a lot of prior research used a one-to-one mapping between a single input utterance and the generated response. This makes it hard to judge the quality of the response generated or the performance of the model with regards to the context of the conversation or how the model would perform when it comes to multi-turn conversations.

   To overcome this issue, researchers have focused on including the previous turns of the conversation as contextual information to the model. This has been accomplished in two different ways: through sequential models (Sordoni et al., 2015) and through hierarchical models (Serban et al., 2016). In sequential encoding of the context, the previous turn of the conversation is concatenated to the current input utterance. In hierarchical encoding of the context, a two-step approach is followed by performing an utterance-level encoding followed by an inter-utterance encoding. Tian *et al.*, (2017) conducted an empirical study that evaluated the advantages and disadvantages of sequential and hierarchical models and show that the hierarchical models outperform sequential models when encoding contextual information.

   Encoding factual knowledge to augment the model was demonstrated by Dinan *et al.* (2018) and Young *et al.* (2018) using the transformer models and Tri-LSTM encoder approach respectively.

2. **Incorporating Personality** - Endowing conversational agents with a coherent persona is key to building a engaging and convincing conversational agent (Niu and Bansal, 2018). The concept of personality has been well studied in the psychology. Traditionally, research on using personality traits has been based on the standard ***Big Five*** model (extraversion, neuroticism, agreeableness, conscientiousness, and openness to experience) and some of the early works on building personalized dialogue systems have been based on the Big Five model (Mairesse and Walker, 2007).

   However, identifying personality traits through ***Big Five*** model is difficult and expensive to obtain (Zheng et al., 2019; Zhang et al., 2018b). Alternative approaches that take advantage of the psycholinguistics are still in their infancy. Some of the proposed approaches to solve this problem have been through explicit or implicit modeling of personality (Zheng et al., 2019). Explicit modeling involves creating profiles of users with features such as age, gender (Zheng et al., 2019) or assigning artificial persona to users and asking them to interact (Zhang et al., 2018b). Implicit Modeling of persona involves creating vectors about the users based on similar features such as age, gender and other personal information (Li et al., 2016a; Kottur et al., 2017).

   More recently, the transformer models have been used for conversational agents (Wolf et al., 2019; Dinan et al., 2018; Rashkin et al., 2018). Wolf *et al.* (2019) demonstrated the usage of transformer model for personalized response generation on the PERSONA-CHAT dataset where the model concatenates each artificial persona provided along with the utterances of the conversation.

3. **Dull and generic responses** - One problem with building end-to-end conversational agents based on vanilla seq2seq is that they are prone to generating dull and generic responses such as `I don't know, I am not sure.` etc. (Vinyals and Le, 2015; Li et al., 2015). These trivial responses make the conversational agents unable to sustain longer conversations with a human. Li *et al.* (2015) suggested a mechanism to overcome this issue with an optimization function (see equation 14, where T is target and S is source sentence). The authors only considered likelihood of the responses when given an input and proposed using Maximum Mutual Information (MMI) as the optimization objective function (see equation 15) where λ is a hyperparameter to penalize generic responses .

$$\hat{T} = \arg\max_{T}\{logp(T|S)\} \tag{14}$$

$$\hat{T} = \arg\max_{T}\{logp(T|S) - \lambda logp(T)\} \tag{15}$$

Recent approaches toward conversational modeling have all tackled the issue of dull and generic response through the use of previous utterance as contextual information or with the help of attention mechanism that focuses on a particular part of the input utterance or using reinforcement learning that penalizes the agent when it produces trivial or repetitive utterance (Li et al., 2016b, 2017; Liu et al., 2018).

5.2 Future Directions

Having identified three open challenges in the subsection above, we now propose two promising future directions on how to tackle these open challenges.

1. **Cognitive Architectures** –We argue that natural language generation entails not only incorporating fundamental aspects of *artificial intelligence* but also *cognitive science* (Reiter and Dale, 2000; Sun, 2007). The role of cognitive architectures(CA) that offers a different perspective has not been explored for deep learning architectures. Cognitive architectures provides a blueprint for building intelligent agents by modelling human behavior. One prominent model in cognitive architecture is the Standard Model (Figure 8) (Norris, 2017; Laird et al., 2017). This model provides the framework with which to conceptually and practically address both long-term memory and short-term memory (also known as working memory), along with an action-selection mechanism acting as a bridge between them. According to this model of human cognition, given an input (for example, through perception), an output is generated by taking into account elements stored in the working memory as well as long-term storage.

2. **Encoding Emotional Content** –Emotions are recognized as functional in decision-making by influencing motivation and action selection (**?**). Therefore, computational emotion models should be grounded in the agent‘s decision making architecture. For example, Badoy *et al.*, (2014) proposed using four basic emotions: joy, sadness, fear, and anger to influence a Qlearning agent. Simulations show that the proposed affective agent required fewer steps to

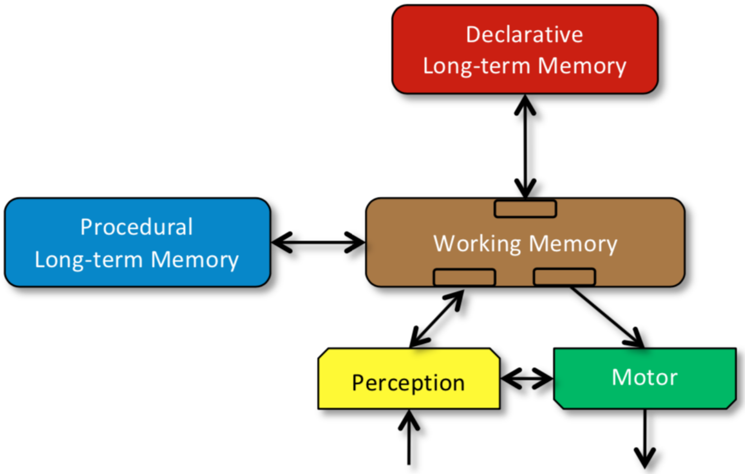

Figure 8: Standard Model of Cognitive Architecture containing two forms of long-term memory (Procedural and Declarative) and Working Memory to address given input. Figure credit: Laird *et al.* (2017)

find the optimal path. In language generation work, Zhuo *et al.*, (2018) have proposed Emotional Chatting Machine (ECM) that can generate appropriate responses not only in content (relevant and grammatical) but also in emotion (emotionally consistent). ECM addresses the factor using three new mechanisms that respectively (1) models the high-level abstraction of emotion expressions by embedding emotion categories, (2) captures the change of implicit internal emotion states, and (3) uses explicit emotion expressions with an external emotion vocabulary. Experiments show that the proposed model can generate responses appropriate not only in content but also in emotion. However, the problem of generating emotionally appropriate responses in longer conversations is still to be explored.

We hypothesize that by incorporating elements of cognitive architectures and adding emotional content, researchers can address the open challenges that we have identified in the prior section. By adapting deep learning approaches to closely mirror the the memory mechanisms as postulated by the Standard Model (Figure 8), dialogue systems can take advantage of longer conversational context as well as world and domain knowledge from databases. By incorporating emotional content, the challenge of having conversational agents mirror a personality can be addressed. In future work, we aim to address the open challenges through these two directions.

Acknowledgements

This work was supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No FA8650-18-C-7881. All statements of fact, opinion or conclusions contained herein are those of the authors and should not be construed as representing the official views or policies of AFRL, DARPA, or the U.S. Government.

References

Gabor Angeli, Percy Liang, and Dan Klein. A simple domain-independent probabilistic approach to generation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 502–512. Association for Computational Linguistics, 2010.

Gabor Angeli, Christopher D Manning, and Daniel Jurafsky. Parsing time: Learning to interpret time expressions. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 446–455. Association for Computational Linguistics, 2012.

Nabiha Asghar, Pascal Poupart, Xin Jiang, and Hang Li. Deep active learning for dialogue generation. *arXiv preprint arXiv:1612.03929*, 2016.

Nabiha Asghar, Pascal Poupart, Jesse Hoey, Xin Jiang, and Lili Mou. Affective neural response generation. *CoRR*, abs/1709.03968, 2017. URL `http://arxiv.org/abs/1709.03968`.

Nabiha Asghar, Pascal Poupart, Jesse Hoey, Xin Jiang, and Lili Mou. Affective neural response generation. In *European Conference on Information Retrieval*, pages 154–166. Springer, 2018.

Wilfredo Badoy and Kardi Teknomo. Q-learning with basic emotions. *CoRR*, abs/1609.01468, 2014.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Satanjeev Banerjee and Alon Lavie. 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*, volume 29, pages 65–72, 2005.

Srinivas Bangalore and Owen Rambow. Corpus-based lexical choice in natural language generation. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, ACL ’00, pages 464–471, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics. doi: 10.3115/1075218.1075277. URL `https://doi.org/10.3115/1075218.1075277`.

Regina Barzilay and Mirella Lapata. Collective content selection for concept-to-text generation. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 331–338. Association for Computational Linguistics, 2005.

Regina Barzilay and Mirella Lapata. Aggregation via set partitioning for natural language generation. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 359–366. Association for Computational Linguistics, 2006.

Regina Barzilay and Lillian Lee. Catching the drift: Probabilistic content models, with applications to generation and summarization. *arXiv preprint cs/0405039*, 2004.

John A Bateman. Enabling technology for multilingual natural language generation: the kpml development environment. *Natural Language Engineering*, 3(1):15–55, 1997.

Anja Belz. Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. *Natural Language Engineering*, 14(4):431–455, 2008.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

Parminder Bhatia, Marsal Gavalda, and Arash Einolghozati. soc2seq: Social embedding meets conversation model. *arXiv preprint arXiv:1702.05512*, 2017.

Aoife Cahill, Martin Forst, and Christian Rohrer. Stochastic realisation ranking for a free word order language. In *Proceedings of the Eleventh European Workshop on Natural Language Generation*, pages 17–24. Association for Computational Linguistics, 2007.

Lynne Cahill, Christy Doran, Roger Evans, Chris Mellish, Daniel Paiva, Mike Reape, Donia Scott, and Neil Tipper. In search of a reference architecture for nlg systems. In *Proceedings of the 7th European Workshop on Natural Language Generation*, pages 77–85, 1999.

Kris Cao and Stephen Clark. Latent variable dialogue models and their diversity. *arXiv preprint arXiv:1702.05962*, 2017.

Hongshen Chen, Zhaochun Ren, Jiliang Tang, Yihong Eric Zhao, and Dawei Yin. Hierarchical variational memory network for dialogue generation. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 1653–1662. International World Wide Web Conferences Steering Committee, 2018.

Hua Cheng and Chris Mellish. Capturing the interaction between aggregation and text planning in two generation systems. In *Proceedings of the first international conference on Natural language generation-Volume 14*, pages 186–193. Association for Computational Linguistics, 2000.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014a.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014b. URL `http://arxiv.org/abs/1406.1078`.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Kenneth Mark Colby. *Artificial paranoia: a computer simulation of paranoid process*. Pergamon Press, 1975.

Robert Dale. Cooking up referring expressions. In *27th Annual Meeting of the association for Computational Linguistics*, 1989.

Robert Dale. *Generating referring expressions: Constructing descriptions in a domain of objects and processes.* The MIT Press, 1992.

Robert Dale and Ehud Reiter. Computational interpretations of the gricean maxims in the generation of referring expressions. *Cognitive science*, 19(2):233–263, 1995.

Hercules Dalianis. Aggregation in natural language generation. *Computational Intelligence*, 15(4): 384–414, 1999.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Aggeliki Dimitromanolaki and Ion Androutsopoulos. Learning to order facts for discourse planning in natural language generation. *arXiv preprint cs/0306062*, 2003.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. Wizard of wikipedia: Knowledge-powered conversational agents. *arXiv preprint arXiv:1811.01241*, 2018.

Pablo A Duboue and Kathleen R McKeown. Statistical acquisition of content selection rules for natural language generation. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 121–128. Association for Computational Linguistics, 2003.

Ondřej Dušek and Filip Jurčíček. A context-aware natural language generator for dialogue systems. *arXiv preprint arXiv:1608.07076*, 2016.

Michael Elhadad and Jacques Robin. An overview of surge: A reusable comprehensive syntactic realization component. In *Eighth International Natural Language Generation Workshop (Posters and Demonstrations)*, 1996.

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

Nikolaos Engonopoulos and Alexander Koller. Generating effective referring expressions using charts. In *INLG*, pages 6–15, 2014.

Roger Evans, Paul Piwek, and Lynne Cahill. What is nlg? Association for Computational Linguistics, 2002.

Jianfeng Gao, Michel Galley, Lihong Li, et al. Neural approaches to conversational ai. *Foundations and Trends® in Information Retrieval*, 13(2-3):127–298, 2019.

Albert Gatt and Emiel Krahmer. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *arXiv preprint arXiv:1703.09902*, 2017.

Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. 1999.

Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. A knowledge-grounded neural conversation model. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Sayan Ghosh, Mathieu Chollet, Eugene Laksana, Louis-Philippe Morency, and Stefan Scherer. Affect-lm: A neural language model for customizable affective text generation. *arXiv preprint arXiv:1704.06851*, 2017.

Eli Goldberg, Norbert Driedger, and Richard I Kittredge. Using natural-language processing to produce weather forecasts. *IEEE Intelligent Systems*, (2):45–53, 1994.

Yoav Goldberg. A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57:345–420, 2016.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

Raghav Goyal, Marc Dymetman, Eric Gaussier, and Uni LIG. Natural language generation through character-based rnns with finite-state prior knowledge. In *COLING*, pages 1083–1092, 2016.

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

Julia Hockenmaier and Mark Steedman. Ccgbank: a corpus of ccg derivations and dependency structures extracted from the penn treebank. *Computational Linguistics*, 33(3):355–396, 2007.

Ari Holtzman, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub, and Yejin Choi. Learning to write with cooperative discriminators. *arXiv preprint arXiv:1805.06087*, 2018.

John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

Helmut Horacek. Generating referential descriptions under conditions of uncertainty. In *Proceedings of the 10th European Workshop on Natural Language Generation (ENLG)*, pages 58–67, 2005.

Chenyang Huang, Osmar Zaiane, Amine Trabelsi, and Nouha Dziri. Automatic dialogue generation with expressed emotions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, volume 2, pages 49–54, 2018.

MI Jordan. Serial order: a parallel distributed processing approach. technical report, june 1985-march 1986. Technical report, California Univ., San Diego, La Jolla (USA). Inst. for Cognitive Science, 1986.

Pei Ke, Jian Guan, Minlie Huang, and Xiaoyan Zhu. Generating informative responses with controlled sentence function. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1499–1508, 2018.

Imtiaz Hussain Khan, Kees Van Deemter, and Graeme Ritchie. Generation of referring expressions: Managing structural ambiguities. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 433–440. Association for Computational Linguistics, 2008.

Joohyun Kim and Raymond J Mooney. Generative alignment and semantic parsing for learning from ambiguous supervision. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 543–551. Association for Computational Linguistics, 2010.

Ravi Kondadadi, Blake Howald, and Frank Schilder. A statistical nlg framework for aggregated planning and realization. In *ACL (1)*, pages 1406–1415, 2013.

Ioannis Konstas and Mirella Lapata. Unsupervised concept-to-text generation with hypergraphs. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 752–761. Association for Computational Linguistics, 2012.

Satwik Kottur, Xiaoyu Wang, and Vítor Carvalho. Exploring personalized neural conversational models. In *IJCAI*, pages 3728–3734, 2017.

Emiel Krahmer and Kees Van Deemter. Computational generation of referring expressions: A survey. *Computational Linguistics*, 38(1):173–218, 2012.

Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. In *International Conference on Machine Learning*, pages 1378–1387, 2016.

John E Laird, Christian Lebiere, and Paul S Rosenbloom. A standard model of the mind: Toward a common computational framework across artificial intelligence, cognitive science, neuroscience, and robotics. *Ai Magazine*, 38(4), 2017.

Irene Langkilde. Forest-based statistical sentence generation. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 170–177. Association for Computational Linguistics, 2000.

Irene Langkilde and Kevin Knight. Generation that exploits corpus-based statistical knowledge. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, pages 704–710. Association for Computational Linguistics, 1998.

Irene Langkilde-Geary and Kevin Knight. Halogen statistical sentence generator. In *Proceedings of the ACL-02 Demonstrations Session*, pages 102–103, 2002.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

Mike Lewis, Denis Yarats, Yann N Dauphin, Devi Parikh, and Dhruv Batra. Deal or no deal? end-to-end learning for negotiation dialogues. *arXiv preprint arXiv:1706.05125*, 2017.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*, 2015.

Jiwei Li, Michel Galley, Chris Brockett, Georgios P Spithourakis, Jianfeng Gao, and Bill Dolan. A persona-based neural conversation model. *arXiv preprint arXiv:1603.06155*, 2016a.

Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541*, 2016b.

Jiwei Li, Will Monroe, Tianlin Shi, Alan Ritter, and Dan Jurafsky. Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547*, 2017.

Percy Liang, Michael I Jordan, and Dan Klein. Learning semantic correspondences with less supervision. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 91–99. Association for Computational Linguistics, 2009.

Zachary C Lipton, John Berkowitz, and Charles Elkan. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*, 2015a.

Zachary C Lipton, Sharad Vikram, and Julian McAuley. Generative concatenative nets jointly learn to write and classify reviews. *arXiv preprint arXiv:1511.03683*, 2015b.

Yahui Liu, Wei Bi, Jun Gao, Xiaojiang Liu, Jian Yao, and Shuming Shi. Towards less generic responses in neural conversation models: A statistical re-weighting method. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2769–2774, 2018.

Ryan Lowe, Michael Noseworthy, Iulian V Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. Towards an automatic turing test: Learning to evaluate dialogue responses. *arXiv preprint arXiv:1708.07149*, 2017.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Hierarchical question-image co-attention for visual question answering. In *Advances In Neural Information Processing Systems*, pages 289–297, 2016.

Yi Luan, Yangfeng Ji, and Mari Ostendorf. Lstm based conversation models. *arXiv preprint arXiv:1603.09457*, 2016.

François Mairesse and Marilyn Walker. Personage: Personality generation for dialogue. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 496–503, 2007.

William C Mann and Sandra A Thompson. Relational propositions in discourse. *Discourse processes*, 9(1):57–90, 1986.

William C Mann and Sandra A Thompson. *Rhetorical structure theory: A theory of text organization*. University of Southern California, Information Sciences Institute, 1987.

Pierre-Emmanuel Mazaré, Samuel Humeau, Martin Raison, and Antoine Bordes. Training millions of personalized dialogue agents. *arXiv preprint arXiv:1809.01984*, 2018.

Kathleen R McKeown. Discourse strategies for generating natural-language text. *Artificial Intelligence*, 27(1):1–41, 1985.

Susan W McRoy, Songsak Channarukul, and Syed S Ali. An augmented template-based approach to text realization. *Natural Language Engineering*, 9(4):381–420, 2003.

Hongyuan Mei, Mohit Bansal, and Matthew R Walter. What to talk about and how? selective generation using lstms with coarse-to-fine alignment. *arXiv preprint arXiv:1509.00838*, 2015.

Hongyuan Mei, Mohit Bansal, and Matthew R Walter. Coherent dialogue with attention-based language models. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, page 3, 2010.

Johanna D Moore and Cécile L Paris. Planning text for advisory dialogues: Capturing intentional and rhetorical information. *Computational linguistics*, 19(4):651–694, 1993.

Johanna D Moore and Martha E Pollack. A problem for rst: The need for multi-level discourse analysis. *Computational linguistics*, 18(4):537–544, 1992.

Lili Mou, Yiping Song, Rui Yan, Ge Li, Lu Zhang, and Zhi Jin. Sequence to backward and forward sequences: A content-introducing approach to generative short-text conversation. *arXiv preprint arXiv:1607.00970*, 2016.

Tong Niu and Mohit Bansal. Polite dialogue generation without parallel data. *Transactions of the Association of Computational Linguistics*, 6:373–389, 2018.

Dennis Norris. Short-term memory and long-term memory are still different. *Psychological bulletin*, 143(9):992, 2017.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. Why we need new evaluation metrics for nlg. *arXiv preprint arXiv:1707.06875*, 2017.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

Qiao Qian, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. Assigning personality/identity to a chatting machine for coherent conversation generation. *arXiv preprint arXiv:1706.02861*, 2017.

Minghui Qiu, Feng-Lin Li, Siyu Wang, Xing Gao, Yan Chen, Weipeng Zhao, Haiqing Chen, Jun Huang, and Wei Chu. Alime chat: A sequence to sequence and rerank based chatbot engine. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 498–503, 2017.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf*, 2018.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. I know the feeling: Learning to converse with empathy. *arXiv preprint arXiv:1811.00207*, 2018.

Ehud Reiter. Has a consensus nl generation architecture appeared, and is it psycholinguistically plausible? In *Proceedings of the Seventh International Workshop on Natural Language Generation*, pages 163–170. Association for Computational Linguistics, 1994.

Ehud Reiter and Robert Dale. Building applied natural language generation systems. *Natural Language Engineering*, 3(1):57–87, 1997.

Ehud Reiter and Robert Dale. *Building natural language generation systems*. Cambridge university press, 2000.

Ehud Reiter, Roma Robertson, and Liesl Osman. Knowledge acquisition for natural language generation. In *Proceedings of the first international conference on Natural language generation-Volume 14*, pages 217–224. Association for Computational Linguistics, 2000.

Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. In *Advances in neural information processing systems*, pages 4967–4976, 2017.

Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Iulian Vlad Serban, Tim Klinger, Gerald Tesauro, Kartik Talamadupula, Bowen Zhou, Yoshua Bengio, and Aaron Courville. Multiresolution recurrent neural networks: An application to dialogue response generation. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017a.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017b.

Lifeng Shang, Zhengdong Lu, and Hang Li. Neural responding machine for short-text conversation. *arXiv preprint arXiv:1503.02364*, 2015.

Xiaoyu Shen, Hui Su, Yanran Li, Wenjie Li, Shuzi Niu, Yang Zhao, Akiko Aizawa, and Guoping Long. A conditional variational framework for dialog generation. *arXiv preprint arXiv:1705.00316*, 2017.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. A neural network approach to context-sensitive generation of conversational responses. *arXiv preprint arXiv:1506.06714*, 2015.

Ron Sun. The importance of cognitive architectures: An analysis based on clarion. *Journal of Experimental & Theoretical Artificial Intelligence*, 19(2):159–193, 2007.

Ilya Sutskever, James Martens, and Geoffrey E Hinton. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1017–1024, 2011.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

Zhiliang Tian, Rui Yan, Lili Mou, Yiping Song, Yansong Feng, and Dongyan Zhao. How to make context more useful? an empirical study on context-aware neural conversational models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 231–236, 2017.

Kees Van Deemter. Generating referring expressions: Boolean extensions of the incremental algorithm. *Computational Linguistics*, 28(1):37–52, 2002.

Kees van Deemter, Mariët Theune, and Emiel Krahmer. Real vs. template-based natural language generation: a false opposition. *Computational Linguistics*, 31(1):15–24, 2005.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

Oriol Vinyals and Quoc Le. A neural conversational model. *arXiv preprint arXiv:1506.05869*, 2015.

Marilyn A Walker, Owen Rambow, and Monica Rogati. Spot: A trainable sentence planner. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–8. Association for Computational Linguistics, 2001.

Peng Wang, Qi Wu, Chunhua Shen, Anthony Dick, and Anton van den Hengel. Fvqa: Fact-based visual question answering. *IEEE transactions on pattern analysis and machine intelligence*, 40 (10):2413–2427, 2018.

Joseph Weizenbaum. Elizaa computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45, 1966.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745*, 2015.

Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *CoRR*, abs/1410.3916, 2014. URL `http://arxiv.org/abs/1410.3916`.

Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. Transfertransfo: A transfer learning approach for neural network based conversational agents. *arXiv preprint arXiv:1901.08149*, 2019.

Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. Topic aware neural response generation. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Chen Xing, Yu Wu, Wei Wu, Yalou Huang, and Ming Zhou. Hierarchical recurrent attention network for response generation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Caiming Xiong, Victor Zhong, and Richard Socher. Dynamic coattention networks for question answering. *arXiv preprint arXiv:1611.01604*, 2016.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057, 2015.

Kaisheng Yao, Geoffrey Zweig, and Baolin Peng. Attention with intention for a neural network conversation model. *arXiv preprint arXiv:1510.08565*, 2015.

Tom Young, Erik Cambria, Iti Chaturvedi, Hao Zhou, Subham Biswas, and Minlie Huang. Augmenting end-to-end dialogue systems with commonsense knowledge. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Ruqing Zhang, Jiafeng Guo, Yixing Fan, Yanyan Lan, Jun Xu, and Xueqi Cheng. Learning to control the specificity in neural response generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1108–1117, 2018a.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. Personalizing dialogue agents: I have a dog, do you have pets too? *arXiv preprint arXiv:1801.07243*, 2018b.

Xingxing Zhang and Mirella Lapata. Chinese poetry generation with recurrent neural networks. In *EMNLP*, pages 670–680, 2014.

Yinhe Zheng, Guanyi Chen, Minlie Huang, Song Liu, and Xuan Zhu. Personalized dialogue generation with diversified traits. *arXiv preprint arXiv:1901.09672*, 2019.

Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. Emotional chatting machine: Emotional conversation generation with internal and external memory. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.