University of Turin
Computer Science Department

Ph.D. Program in Computer Science
Cycle XXXIV

Deep Learning Methods for Faithful Data-To-Text Generation

Marco ROBERTI

Supervisor: Prof. Rossella CANCELLIERE

Ph.D. Program Coordinator: Prof. Viviana PATTI

Academic years of enrollment: 2018/2019
2019/2020
2020/2021

Code of scientific discipline: INF/01

UNIVERSITY OF TURIN

Abstract

School of Science of Nature

Computer Science Department

Doctor of Philosophy

Deep Learning Methods for Faithful Data-To-Text Generation

by Marco ROBERTI

Data-To-Text Generation (DTT) is a subfield of Natural Language Generation aiming at transcribing structured data in natural language descriptions. The field has been recently boosted by the use of neural-based generators, which exhibit great syntactic skills without the need of hand-crafted pipelines, but do not currently guarantee a fully faithful natural language transduction. Two different approaches can help to reach this goal: the direct copy of some input content into the generated output, and the ability to avoid the generation of information which is not included in the input data – usually called *hallucinations*. In this thesis, both issues are analyzed, and new methods are introduced to deal with them.

We present an end-to-end sequence-to-sequence model with attention, enriched by a copy mechanism which reads and generates at a character level. Such architecture includes two major features: (i) the possibility to alternate between the standard generation mechanism and a copy one, which allows to directly copy input facts to produce outputs, and (ii) the use of an original training pipeline that further improves the quality of the generated texts.

In order to treat hallucinations, we introduce a Multi-Branch Decoder which is able to leverage word-level labels to learn the relevant parts of each training instance. These labels are obtained following a simple and efficient scoring procedure based on co-occurrence analysis and dependency parsing. The results obtained demonstrate a greater faithfulness of the generated text to input data.

Contents

List of Figures

List of Tables

List of Abbreviations

ASCII	**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange
BLEU	**Bi**Lingual **E**valuation **U**nderstudy
CIDEr	**C**onsensus-based **I**mage **D**escription **E**valuation
CNN	**C**onvolutional **N**eural **N**etwork
CTG	**C**ontrolled-**T**ext-**G**eneration
DTT	**D**ata-**T**o-**T**ext Generation
E2E	**E**nd-to-**E**nd
GPU	**G**raphical **P**rocessing **U**nit
GRU	**G**ated **R**ecurrent **U**nit
HSMM	**H**idden **S**emi **M**arkov **M**odel
LM	**L**anguage **M**odel
LSTM	**L**ong **S**hort-**T**erm **M**emory
MBD	**M**ulti **B**ranch **D**ecoder
METEOR	**M**etric for **E**valuation of **T**ranslation with **E**xplicit **OR**dering
MR	**M**eaning **R**epresentation
NIST	**N**ational **I**nstitute of **S**tandards and **T**echnology
NL	**N**atural **L**anguage
NLG	**N**atural **L**anguage **G**eneration
OOV	**O**ut-**O**f-**V**ocabulary
PARENT	**P**recision **A**nd **R**ecall of **E**ntailed **N**grams from the **T**able
REG	**R**eferring **E**xpression **G**eneration
ReLU	**Re**ctified **L**inear **U**nit
RL	**R**einforcement **L**earning
RNN	**R**ecurrent **N**eural **N**etwork
ROUGE	**R**ecall-**O**riented **U**nderstudy for **G**isting **E**valuation
TL	**T**ransfer **L**earning

Chapter 1

Introduction

On March 17, 2014, a magnitude 4.4 earthquake[1] struck the Los Angeles zone, in the south-western USA, at 13:25:36 UTC. Three minutes later, the *Los Angeles Times* published the following news article:

> A shallow magnitude 4.7 earthquake was reported Monday morning five miles from Westwood, California, according to the U.S. Geological Survey. The temblor occurred at 6:25 a.m. Pacific time at a depth of 5.0 miles.
>
> According to the USGS, the epicenter was six miles from Beverly Hills, California, seven miles from Universal City, California, seven miles from Santa Monica, California and 348 miles from Sacramento, California. In the past ten days, there have been no earthquakes magnitude 3.0 and greater centered nearby.
>
> This information comes from the USGS Earthquake Notification Service and this post was created by an algorithm written by the author.[2]

As suggested by the last sentence, Ken Schwencke, formal author of the article, just had to hit a "Publish" button[3], as the actual content was written by *Quakebot*, an automatic Natural Language Generation system that reads United States Geological Survey's earthquake reports and extracts from them the relevant information. Quakebot is a simple real-world example of Data-To-Text generation.

Data-To-Text generation (DTT) is an instance of Natural Language Generation (NLG). Reiter and Dale (1997) define NLG as "the subfield 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". Data-To-Text generation has been later characterized, in a more fine-grained way, as "the problem of generating descriptive text from database records" (Wiseman et al., 2017). This definition still includes a broad range of applications, including:

- summary of hospital patients conditions (Banaee et al., 2013; Gatt et al., 2009);

[1] https://earthquake.usgs.gov/earthquakes/eventpage/ci15476961
[2] https://www.latimes.com
[3] https://slate.com

- current news reports (Leppänen et al., 2017);
- weather forecasts (Goldberg et al., 1994; Ramos-Soto et al., 2015; Reiter et al., 2005; Turner et al., 2007);
- financial reports (Plachouras et al., 2016);
- soccer matches chronicle (D. L. Chen & Mooney, 2008; Theune et al., 2001);
- museum-specific interactive information about archaeological finds, paintings, or sculptures (O'Donnell et al., 2001; Stock et al., 2007);
- persuading and motivating content (Carenini & Moore, 2006; Reiter et al., 2003).

Traditionally, the DTT problem has been faced via pipeline models, in which each submodule addresses a specific sub-task (detailed in Subsection 3.1). Symbolic systems were the *de facto* standard, even if they typically require handcrafted rules heavily exploiting domain experts' work, and are very far from being generalizable. The rise of data-driven architectures, and particularly of Deep Learning-based ones in the last decade, reverberated on the Natural Language Processing field, including DTT. Modern architectures have typically an end-to-end architecture: their data-driven design allows the development of general models for DTT, but they require a significant amount of data to reach satisfactory performances. The need for larger and more complex datasets led the research community to the publication of new datasets (Gardent et al., 2017a; Lebret et al., 2016; Novikova et al., 2017b; Parikh et al., 2020; Wiseman et al., 2017), still used nowadays.

Interesting challenges arose from the joint use of Deep Learning architectures and massive datasets. What is the most effective way to encode database records? How do differences between DTT and NLG can be exploited to build better architectures? How datasets sizes affect the way they are built, and what are the consequences? How can we build models which are robust to noisy datasets?

The above challenges can be summarized by a single question: how can we ensure faithful generation using data-driven models? This thesis focuses on **faithful Data-To-Text Generation**. In particular, it (i) investigates the effectiveness of a copy-based model specifically designed for DTT, whose generality is enforced by a character-level tokenization, and (ii) it addresses the hallucination problem via a specifically built architecture, which minimizes the negative consequences of noisy, big-sized datasets.

In Chapter 2 the Data-To-Text task is formalized. Classical architectures are presented in Chapter 3, together with a brief overview of related work. Chapter 4 presents a copy mechanism for Data-To-Text Generation, applied in a character-wise fashion; in Chapter 5 the *hallucination problem* is presented and faced with a specially designed neural framework. Chapter 6 draws the conclusions of this thesis and suggests some research lines for future work.

Chapter 2

Data-To-Text Generation

2.1 Generating text from structured data

Data-to-Text Generation (DTT) is the subfield of Computational Linguistics and Natural Language Generation (NLG) that is concerned with transcribing structured data into natural language descriptions, or, said otherwise, transcribing machine understandable information into a human understandable description (Gatt & Krahmer, 2018). DTT objectives includes *coverage*, i.e. all the required information should be present in the text, and *adequacy*, i.e. the text should not contain information that is not covered by the input data.

DTT is a domain distinct from other NLG task (e.g. machine translation (Wiseman et al., 2017), text summarization (Kryscinski et al., 2019)) with its own challenges (Wiseman et al., 2017), starting with the nature of inputs (Narayan & Gardent, 2020; Reiter & Dale, 1997). Such inputs include and are not limited to databases of records, spreadsheets, knowledge bases, sensor readings, and they can effectively be expressed through the idea of *tables*.

In the DTT context, input tables are defined as variable-sized sets of key-value pairs, in which the key consist of a single vocabulary token, and the value is a sequence of words. Every table defines an *entity*: an example of a single-entity DTT dataset is the WikiBio dataset. Recent developments of DTT also involves multiple-entity inputs (Wiseman et al., 2017), as shown in Table 2.1. Fig. 5.1 shows a a WikiBio instance, where a data table containing information about Kian Emadi is paired with the corresponding natural language description found on Wikipedia.

Early approaches to DTT relied on static rules hand-crafted by experts, in which content selection (what to say) and surface realization (how to say it)

Dataset	Domain	Cont. selection	Noisy	Entity
WeatherGov (Liang et al., 2009)	Weather forecast	✓	✗	Single
WikiBio (Lebret et al., 2016)	Biographies	✓	✓	Single
WebNLG (Gardent et al., 2017a)	Various	✗	✗	Single
E2E (Novikova et al., 2017b)	Restaurants	✗	✗	Single
RotoWire (Wiseman et al., 2017)	Sportscast	✓	✓	Multiple
SBNation (Wiseman et al., 2017)	Sportscast	✓	✓	Multiple
ToTTo (Parikh et al., 2020)	Various	✓	✗	Single

TABLE 2.1: Most commonly used Data-To-Text Generation datasets and their main features.

Dataset	No. of instances			Vocabulary size	Avg. target length
	Train	Valid.	Test		
WeatherGov (Liang et al., 2009)	29 528 (total)			345	30.6
WikiBio (Lebret et al., 2016)	582 659	72 831	72 831	$\sim$ 400 000	26.1
WebNLG (Gardent et al., 2017b)	25 298 (total)			8077	22.7
E2E (Novikova et al., 2017b)	42 061	4672	4693	$\sim$ 3000	14.3
RotoWire (Wiseman et al., 2017)	3398	727	728	$\sim$ 11 300	337.1
SBNation (Wiseman et al., 2017)	7633	1635	1635	$\sim$ 68 600	805.4
ToTTo (Parikh et al., 2020)	120 761	7700	7700	136 777	17.4

TABLE 2.2: Size of the Data-To-Text Generation datasets included in Table 2.1

are typically two separate tasks (Ferreira et al., 2019; Reiter & Dale, 1997). In recent years, neural models have blurred this distinction: various approaches showed that both content selection and surface realization can be learned in an end-to-end, data-driven fashion (Liu et al., 2019a; Mei et al., 2016; Puduppully et al., 2019a). Based on the now-standard encoder-decoder architecture, with attention and copy mechanisms (Bahdanau et al., 2015; Bonetta et al., 2021; Roberti et al., 2019; See et al., 2017), neural methods for DTT are able to produce fluent text conditioned on structured data in a number of domains (Lebret et al., 2016; Puduppully et al., 2019c; Wiseman et al., 2017), without relying on heavy manual work from field experts.

Such advances have gone hand in hand with the introduction of larger and more complex benchmarks. In particular, surface-realization abilities have been well studied on hand-crafted datasets such as E2E (Novikova et al., 2017c) and WebNLG (Gardent et al., 2017a), while content-selection has been addressed by automatically constructed datasets such as WikiBio (Lebret et al., 2016) or RotoWire (Wiseman et al., 2017). These large corpora are often constructed from internet sources, which, while easy to access and aggregate, do not consist of perfectly aligned source-target pairs (Dhingra et al., 2019; Perez-Beltrachini & Gardent, 2017). Consequently, model outputs are often subject to over-generation: misaligned fragments from training instances, namely *divergences*, can induce similarly misaligned outputs during inference, the so-called *hallucinations*.

2.2 Data-To-Text tasks

Traditionally, the DTT problem has been faced by addressing a number of subtasks, simplifying the global objective of generating natural language from structured data. Reiter and Dale (1997) identify the following six sub-tasks:

Content selection: determining the subset of the input information to include in the generated text;

Text structuring: ordering the information in an accessible way;

Sentence aggregation: splitting the information in sentences, both at the semantic and at the syntactic level;

Lexicalization: deciding the words and phrases that verbalize information

*Referring Expression Generation (*REG*):* choosing the words and phrases that unambiguously identify domain objects. The essential difference with lexicalization is that REG, consists in a discrimination task. This problem is in turn split in the determination of *referential forms* (pronoun, proper name, definite or indefinite description) and referential contents (set of properties that identify the target entity);

Surface realization: generating the final well-formed, syntactically correct sentences. This involves ordering the sentence components, ensuring morphological correctness, producing function words and punctuation. Surface generation typically faces the generation gap and can be interpreted as a mapping between non-isomorphic structures (Ballesteros et al., 2015).

Splitting the DTT task expose generation systems to the *generation gap* (Meteer, 1991), defined as the presence of mismatches between early and later components, so that antecedent decisions in the pipeline have unexpected, and possibly unfavorable, consequences on the later ones (Gatt & Krahmer, 2018). The problem can be attenuated by merging two or more tasks, such as content selection and text structuring (Duboué & McKeown, 2003), lexicalization and surface realization (Elhadad et al., 1997), or content selection and REG (Engonopoulos & Koller, 2014). Data-driven end-to-end systems take this idea to an extreme, as they solve the DTT generation problem as a whole, without the need for splitting it in sub-tasks and therefore avoiding the consequences of the generation gap.

Chapter 3

Data-To-Text Models

Recent Data-To-Text Generation literature highlights two parallel global trends in research (Gatt & Krahmer, 2018). On the one hand, architectures formerly composed of a pipeline of self-contained, subsequent modules are now blurring the boundaries between their sub-tasks, turning into integrated end-to-end systems. On the other hand, symbolic or knowledge-based systems are giving way to domain-independent data-driven methods, whose behavior is strongly defined by the examples they are fed.

These trends are embodied by neural generation systems, that are nowadays the state-of-the-art in DTT (Clive et al., 2021; Rebuffel et al., 2020a), as well as in other NLP tasks (Shoeybi et al., 2019; Yang et al., 2019). Neural methods for DTT are typically borrowed from Neural Machine Translation and Neural Summarization ones (Dusek & Jurcícek, 2016; Mei et al., 2016), but they can be further adapted to the task at hand.

Following the historical development of DTT methods, we briefly describe modular architectures in Section 3.1. We then present integrated data-driven models in Section 3.2, focusing on neural-based models, which range from Recurrent Neural Networks (and their improvements given by the attention mechanism) to the Transformer architectures.

3.1 Modular architectures

The classical three-stage pipeline architecture, originally introduced by Reiter (1994), consist of a *Text Planner*, a *Sentence Planner*, and a *Linguistic Realizer*. This abstract model, outlined by Figure 3.1, has been described as the "de facto standard" (Reiter, 2010; Reiter & Dale, 1997), even if a fair number of systems relax or violate it (Gatt & Krahmer, 2018).

The Text Planner is in charge of content selection and text structuring. It is often referred to as *Macroplanner*, and it determines "what to say". On the other side, the Sentence Planner incorporates sentence aggregation, lexicalization and referring expression generation. In opposite with the Text Planner, it

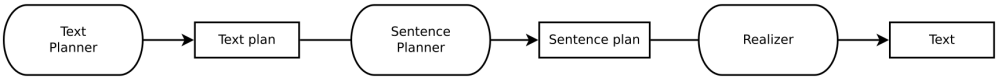


FIGURE 3.1: The classical three-stage pipeline architecture (Reiter, 1994)

Token	Description
<s>	Start of sequence
</s>	End of sequence
<unk>	Unknown, out-of-vocabulary word
<pad>	Padding (used in mini-batching)

TABLE 3.1: Special tokens included in the vocabulary $\mathcal{V}$, and their respective roles.

is often referred to as *Microplanner*, and it determines "how to say". Finally, the Linguistic Realizer carries out the surface realization task alone.

All the above sub-models have shared the historical path from earlier domain-dependent, rule-based methods (Dale, 1989; Hovy, 1987; McKeown, 1985; Reiter et al., 1995; Reiter et al., 2005; Scott et al., 1991) to more recent data-driven ones (Althaus et al., 2004; Belz, 2008; Venigalla & Eugenio, 2013; Viethen & Dale, 2008; White & Howcroft, 2015). Architectures which are simultaneously modular and data-driven are less common, as the shift to data-driven techniques occurred in parallel with the movement of the research interest towards end-to-end architectures, described in the following Section.

3.2 Integrated architectures

3.2.1 Notation

Neural sequence-to-sequence architectures take a sequence $\{x_1, \ldots, x_{T_x}\}$ as input, and output another sequence $\{y_1, \ldots, y_{T_y}\}$. Both input and output sequences consist of lists of embedded tokens. Their lengths are T_x and T_y, respectively. Input sub-sequences ranging from 1 to j are referred to as $x_{1:j}$, defining in this way $x = x_{1:T_x}$. Similarly, output sub-sequences ranging from 1 to t are referred to as $y_{1:t}$, defining $y = y_{1:T_y}$.

More specifically, in Data-To-Text Generation inputs are variable-sized sets of key-value pairs $\langle k; w_{1:T_k} \rangle$, in which the key k consist of a single vocabulary token, and the value is a sequence of T_k words. Input and output sequences share the same *vocabulary* $\mathcal{V}$, defined as the set of all possible $V = |\mathcal{V}|$ tokens, including the special ones shown in Table 3.1.

Weight matrices and vectors, whose values are learned via back-propagation, are referred to as W_* and b_* respectively; subscripts are used to distinguish them from each other. Hidden states sizes (or *model sizes*) will be generically referred to as $emb \in \mathbb{N}^+$, regardless of their possible variations between layers.

3.2.2 RNN Encoder-Decoder

The Recurrent Neural Network Encoder-Decoder architecture (Cho et al., 2014b; Sutskever et al., 2014) consist of two separate RNNs, the *encoder* and the *decoder*, that play different roles.

Name	Reference	Formula	Notes
Additive	Bahdanau et al. (2015)	$b^\intercal \cdot \tanh(W \cdot [d; h])$	$b \in \mathbb{R}^{emb}$, $W \in \mathbb{R}^{emb \times 2 \cdot emb}$
General	T. Luong et al. (2015)	$d \cdot W \cdot h$	$W \in \mathbb{R}^{emb \times emb}$
Dot-product	T. Luong et al. (2015)	$d \cdot h$	
Scaled dot-product	Vaswani et al. (2017)	$\frac{d \cdot h}{\sqrt{emb}}$	

TABLE 3.2: Different approaches for computing score(d, h) in the attention mechanism.

The encoder is in charge of reading the input sequence token by token, updating its hidden state vector $h_j \in \mathbb{R}^{emb}$:

$$h_j = \text{RNN}_{\text{enc}}(x_j, h_{j-1}), \quad j = 1, \ldots, T_x. \tag{3.1}$$

The decoder updates its hidden state conditioned by the previous generated token and by the encoder's final hidden state, used as its initial one (i.e. $d_0 = h_{T_x}$):

$$d_t = \text{RNN}_{\text{dec}}(y_{t-1}, d_{t-1}), \quad t = 1, \ldots, T_y. \tag{3.2}$$

At each time step t, the corresponding decoder's hidden state d_t is projected to a vocabulary-sized vector o_t, which is in turn converted to a categorical probability distribution:

$$o_t = W_{do}^\intercal \cdot d_t \tag{3.3}$$

$$P(y_t | y_{1:t-1}, x) = \text{softmax}(o_t), \tag{3.4}$$

where $W_{do} \in \mathbb{R}^{emb \times V}$. The probability $P(y_t | y_{1:t-1}, x)$ is used to generate the output token y_t.

A typical RNN Encoder-Decoder shows the following features:

- the RNN variants used for both the encoder and the decoder are Long Short-Term Memory (Gers et al., 2000; Hochreiter & Schmidhuber, 1997) or, less frequently, Gated Recurrent Units (Cho et al., 2014b). Those architectures reduce the exploding or vanishing gradient problems (Bengio et al., 1993; Bengio et al., 1994) and better deal with long-term dependencies inside sequences;
- the encoder is bidirectional (Schuster & Paliwal, 1997), as the whole input sequence is typically available and information from both left and right tokens can be informative;
- the decoder uses input feeding, i.e. "attentional vectors are fed as inputs to the next time steps to inform the model about past alignment decisions" (T. Luong et al., 2015);
- the whole architecture is trained end-to-end, using Back-Propagation Through Time (Rumelhart et al., 1986; Williams & Zipser, 1989) and Teacher Forcing (Williams & Zipser, 1989).

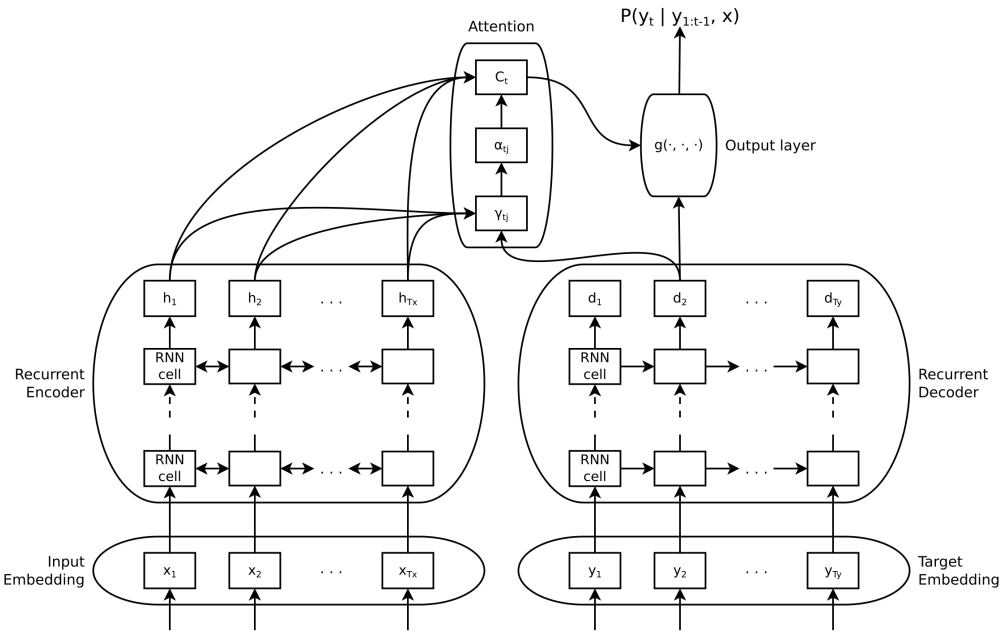


FIGURE 3.2: The Encoder-Decoder with attention architecture (Bahdanau et al., 2015; T. Luong et al., 2015)

3.2.3 RNN Encoder-Decoder with Attention

The Recurrent Encoder-Decoder with attention (Bahdanau et al., 2015; T. Luong et al., 2015) is an improvement of the aforementioned RNN Encoder-Decoder architecture. The latter requires that the input information read by the decoder is contained in a fixed-sized vector $d_0 = h_{T_x}$ produced by the encoder. This fixed size is problematic, as it may lead to either overfitting or information loss, depending on the ratio between the amount of information to store and the RNNs' hidden size. Indeed, Cho et al. (2014a) observed that "the performance of the neural machine translation suffers significantly from the length of sentences". The attention mechanism overcomes this problem.

The attention mechanism is a neural network technique that consists in performing a weighted sum over a list of values, whose weights depend on their comparison with a query vector. The result of the weighted sum is called the *context vector*. Given the last decoder's hidden state d_{t-1} and the sequence of the encoder's hidden states h_j $(j = 1, \ldots, T_x)$, the main components of the attention mechanism are:

(i) the alignment model γ_{tj}

$$\gamma_{tj} = \text{score}(d_{t-1}, h_j), \qquad 1 \leq j \leq T_x, \quad 1 \leq t \leq T_y, \tag{3.5}$$

which scores how well input in position j-th and output observed in the t-th time instant match. It can be computed in several ways, as shown in Table 3.2.

(ii) the attention probability distribution α_{tj}

$$\begin{aligned}\alpha_{tj} &= \frac{\exp(\gamma_{tj})}{\sum_{k=1}^{T_x}\exp(\gamma_{tk})} \\ &= [\text{softmax}(\gamma_t)]_j, \qquad 1 \le j \le T_x, \quad 1 \le t \le T_y,\end{aligned} \tag{3.6}$$

where $\gamma_t \in \mathbb{R}^{T_x}$ is the vector whose j-th element is γ_{tj}, i.e. $\gamma_t \coloneqq \gamma_{t,1:T_x}$.

(iii) the context vector C_t, weighted sum of the encoder annotations h_j

$$C_t = \sum_{j=1}^{T_x} \alpha_{tj} h_j, \qquad 1 \le t \le T_y. \tag{3.7}$$

In the RNN Encoder-Decoder with Attention, the update of the decoder's hidden state only includes the currently useful input information, via the context vector C_t computed at each time step:

$$\tilde{d}_{t-1} = \tanh(W_{dc}^{\intercal} \cdot [d_{t-1}; C_t]), \quad t = 1, \dots, T_y \tag{3.8}$$

$$d_t = \text{RNN}_{\text{dec}}(y_{t-1}, \tilde{d}_{t-1}), \quad t = 1, \dots, T_y, \tag{3.9}$$

where $\tilde{d}_{t-1}$ brings the information from both the decoder's last hidden and the context vector, and $W_{dc} \in \mathbb{R}^{2 \cdot emb \times emb}$ is a trainable parameter.

According to Bahdanau et al. (2015), the context vector C_t is the key element for evaluating the final conditional probability $P(y_t|y_{1:t-1}, x)$ to output a target token y_t, given the previously outputted tokens $y_{1:t-1}$ and the input x. In fact, they express this probability generalizing Eq. 3.3 as:

$$P(y_t|y_{1:t-1}, x) = g(y_{t-1}, d_t, C_t), \tag{3.10}$$

where g is a non-linear, potentially multi-layered, function. So doing, the explicit information about $y_{1:t-1}$ and x is replaced with the knowledge of the context C_t and the decoder's state d_t.

3.2.4 Transformer

Recurrent architectures are inherently sequential, which precludes parallelization and negatively affect computation times, especially for longer sequences. The Transformer architecture (Vaswani et al., 2017) overcomes these limitations, eschewing recurrence and relying entirely on the attention mechanism.

Similarly to the architectures presented earlier, it is composed by an encoder and a decoder, which mainly rely on improvements of the attention mechanism. *Multi-head attention* is a linear combination of independently computed *scaled dot-product* attentions (see Table 3.2); *self-attention* aims at finding relationships within a sequence's tokens.

The Transformer encoder is a stack of identical modules, each one consisting of a multi-head self-attention followed by a feed-forward layer. The Transformer decoder differs from the encoder in the facts that multi-head self-attention is followed by multi-head input-output attention, and it includes a linear projection and a softmax activation as final layers, determining the categorical

distribution from which the t-th output token is sampled, similarly to Equations 3.3 and 3.10.

3.2.5 Data-To-Text adaptations

The main difference between Data-To-Text and other text-to-text tasks, such as Machine Translation, Automatic Summarization, and Dialogue Response Generation, is the non-linear structure of the input. Despite that, all the neural models described above assume the data they are fed with to be a sequence. Data tables can have either a key-value structure (Lebret et al., 2016; Novikova et al., 2017b), or a more complex table form (Wiseman et al., 2017): they need a pre-processing step called *linearization*, which makes them compatible with neural sequence-to-sequence systems. Linearization of the input involves (i) creating embedding vectors that encode it in a convenient way, and (ii) determining an arbitrary order for inherently unordered data.

Key-value pairs can be treated as independent subsequent embedded tokens, delegating to the network the task of distinguishing tokens belonging to data keys, from those belonging to the corresponding values (Dusek & Jurcícek, 2016). A more refined approach consists in concatenating the field embedding and each value token's one (Sha et al., 2018), possibly adding a linear projection and a non-linear activation, such as the hyperbolic tangent (Wiseman et al., 2017; Yang et al., 2017). The representation of the field relative of a given token can be enriched by such token's position, counted from both the start and the end of the sequence (Lebret et al., 2016; Liu et al., 2018).

Different ordering of key-value pairs during training impacts the resulting generation systems' performance (Kedzie & McKeown, 2020). In particular, when the order of the pairs matches the output sentence's realization order, models tend to be more controllable. Transformer-based models' faithfulness is less affected by the ordering of the pairs, as their architecture is less influenced by input tokens' positions. The positional encoding of the Transformer encoder can be simply removed, preserving the unorderedness of the input (Rebuffel et al., 2020a).

3.3 Previous work

Neural models for Data-To-Text generation gained popularity increasingly during the last decade.

Conditioned Neural Language Models Wen et al. (2015a) proposed the first neural system specifically designed for this task, which consists in a RNN Language Model (Mikolov et al., 2010), conditioned by a one-hot representation of the input data structure. The model is enriched by a CNN sentence model, which checks the generated sentence for semantic consistency, and by a backward RNN-based reranker. The architecture has been later improved by extending the recurrent LSTM architecture with a gating "sentence planning cell", yielding better results (Wen et al., 2015b).

Standard RNN Encoder-Decoder An RNN Encoder-Decoder with attention (see Section 3.2.3) for DTT has been developed by Dusek and Jurcícek (2016),

aiming to generate either a deep syntax dependency tree, realized by an external module, or the final sentence, in an end-to-end fashion. Again, a reranking strategy is applied, penalizing the absence of required information and the addition of irrelevant one. As an alternative to reranking, Chisholm et al. (2017) propose an auto-encoding strategy: an RNN Encoder-Decoder with attention is used to generate the natural language description of the input table, and then to get such table from the generated utterance. This constrains output sequences to only express the facts that are present in the data.

Improved Attention Mechanisms The encoder-aligner-decoder architecture (Mei et al., 2016) is an RNN Encoder-Decoder with a "coarse-to-fine alignment" mechanism. Standard attention weights are re-weighted by the probability of each input token of being selected, computed by a *pre-selector* solely on the basis of the input. This allows a more picky content selection phase. Differently from the previous architectures, the encoder-aligner-decoder takes rid of beam search, reranking and auto-encoding, simply relying on greedy generation. Sha et al. (2018) replace the conventional attention mechanism with a *dispatcher*, that uses a soft switch to choose between the standard content-based attention and a link-based attention, that learns the transition between table fields during decoding, explicitly modeling the generation order of the input fields. A more complex architecture is proposed by Puduppully et al. (2019b), as they interpose a content selection gate and a neural planner between the encoder and the decoder's attention mechanism, that uses the generated plan as the attention keys.

Encoding Structured Data The main difference between DTT and Machine Translation, i.e. the structured form of the data, has led to work on the encoding side. Lebret et al. (2016) use a novel table encoding and embedding strategy to condition a neural Manguage Model, both locally and globally. They also include copy actions, taking into account that input tables often contain output tokens. This encoding strategy has been included in an Encoder-Decoder architecture by Liu et al. (2018). Their encoding RNN is a modification of the LSTM cell, in which the cell state is updated using also the field information. Their *dual attention* mechanism uses the product of independent word-based and field-based attention weights to compute the final context vector. Differently, Puduppully et al. (2019d)'s model creates entity representations which are dynamically updated. Their attention mechanism has a hierarchical structure, and it takes into account both the input data and the entity representations. Hierarchical encoders are proposed by Liu et al. (2019b) and Rebuffel et al. (2020a) as well. The former use a word-level and an attribute-level LSTM, and the respective attention weights are combined via an element-wise product. The latter encodes entities from records, and data-structures from entities, taking advantage of Transformer-based architectures. This allows to encode multiple-entity data structures such as the ones included in the RotoWire (Wiseman et al., 2017) dataset.

The Decoding Side Compared to the encoder, relatively little work has been focusing on the decoder. In fact, generating human-like sentences is not an exclusive property of DTT, unlike encoding data tables. Wiseman et al. (2018) propose a neural reinterpretation of template-based models: their Hidden

Semi-Markov Model (HSMM) decoder architecture "learns latent, discrete templates jointly with learning to generate". Such templates are easily interpretable and facilitate the controllability of the generation, even if the quality of the resulting sentences is quite far from state-of-the-art models.

3.4 Perspectives

As seen in this Chapter, DTT is nowadays an ever-growing field of research. New models are still being proposed in the literature, mainly focusing on finding more convenient ways to encode structured data. However, sticking to the information provided by the input tables remains an open problem in this domain, as current systems do not guarantee a fully faithful natural language transduction. Faithfulness can be obtained by two main features. In the one hand, the ability to directly transcript input content to the generated output – in short, to copy; on the other hand, the ability to avoid the generation of information which is not included in the table, still allowing to produce content that can be inferred from it. In this thesis both components are analyzed, and new methods are subsequently introduced: Chapter 4 describes a character-level system which integrates a copy mechanism, while Chapter 5 presents a framework for reducing hallucinations, made of a word-level labeling procedure and a multi-branch deep neural model.

Chapter 4

A Copy Mechanism for Data-To-Text Generation

4.1 Introduction and related work

Recurrent Encoder-Decoder models with Attention have proved to be very effective in Data-To-Text Generation (DTT) and Natural Language Generation (NLG) tasks (Karpathy & Li, 2015; Mei et al., 2016; Wen et al., 2015b), as well as in machine translation (Bahdanau et al., 2015; Cho et al., 2014c; Sennrich et al., 2016b; Sutskever et al., 2014) and in language modeling (Al-Rfou et al., 2019).

In this Chapter we present a character-level model that results in a completely neural end-to-end architecture for DTT. When compared to traditional word-based approaches, character-based ones entail several benefits:

- word-based models involve the non-trivial choice of a tokenization algorithm, which implies constraints on the vocabulary size, the presence of Out-Of-Vocabulary (OOV) words, finding (often non-optimal) subwords. The character-based approach implies, by definition, a predetermined and straightforward way of splitting inputs;
- OOV words are often *delexicalized* in data-to-text generation, i.e., table values are replaced by a key-dependent placeholder that needs to be post-processed when generated by the system. Character-based models do not include any OOV token, eliminating the delexicalization - relexicalization procedure;
- lowercasing words is a common strategy to reduce the vocabulary size in word-based models, leading to a loss of useful information and to the need for a post-processing phase called truecasing. Character-based approaches bypass the issue, as they have a natively small vocabulary;
- differently from word-based paradigm, the character-based vocabulary does not depend on a specific domain's set of terms, but rather on a general, small-sized alphabet.

As we will see, our an approach achieves rather interesting performance results and produces a vocabulary-free model that is inherently more general. According to our experiments, it never hallucinates words, nor duplicates them. Because of this, it opens up the possibility to adapt already trained networks to deal with different datasets.

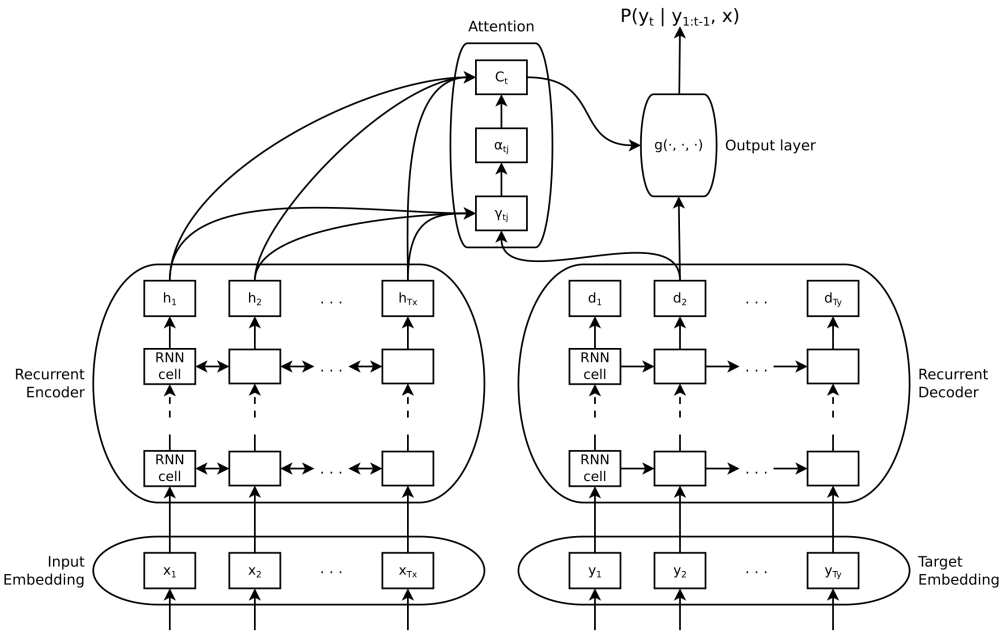


FIGURE 4.1: Encoder-Decoder with Attention model, described in Section 3.2.3

In sequence-to-sequence frameworks (Aharoni et al., 2016; Cho et al., 2014c; Sutskever et al., 2014), data are usually represented word-by-word both in input and output sequences; anyway, such schemes can't be effective without a special, non-neural delexicalization phase that handles unknown words, such as proper names or foreign words (see Wen et al. (2015b)). The delexicalization step has the benefit of reducing the dictionary size and, consequently, the data sparsity, but it is affected by various shortcomings. In particular, according to Goyal et al. (2016) (i) it needs some reliable mechanism for entity identification, i.e. the recognition of named entities inside text; (ii) it requires a subsequent "re-lexicalization" phase, where the original named entities take back placeholders' place; (iii) it cannot account for lexical or morphological variations due to the specific entity, such as gender and number agreements, that can't be achieved without a clear context awareness.

Some recent works tried to solve this problem: Gu et al. (2016) describe *CopyNet*, a word-based technique that can integrate output generation with a copying mechanism which can choose portions of the input sequence and include them in the final sentence. Similarly, in the word-based *Pointer-Generator Network* (See et al., 2017), a soft switch determines whether the next output token is generated or copied from the input, re-using the attention distribution. Such techniques, albeit conceived for words, can be adapted to the character copying task, leading to more robust and effective models. T. Luong et al. (2015) tries to extend neural networks with a post-processing phase that copies words as indicated by the model's output sequence. Some character-level aspects appear as a solution of the issue as well, either as a fallback for rare words (M.-T. Luong & Manning, 2016), or as subword units (Sennrich et al., 2016b).

A significantly different approach consists in employing characters instead of words, for input slot-value pairs tokenization as well as for the generation of the final utterances, as done for instance in Agarwal and Dymetman

(2017) and Al-Rfou et al. (2019). One of the very first attempts to model natural language via a character-level mechanism is described by Sutskever et al. (2011). According to this paper, a simple variant of the vanilla recurrent neural network can generate well-formed sentences after being trained based on sequences of characters. A representative case of character-based systems is Goyal et al. (2016), but this model incorporates prior knowledge in the form of a finite-state automaton to prevent "the generation of non-words and the hallucination of named entities".

In this Chapter we present a character-level sequence-to-sequence model with attention mechanism that results in a completely neural end-to-end architecture. More specifically, our model shows two important features, with respect to the architecture proposed by Bahdanau et al. (2015): (i) a character-wise copy mechanism, consisting in a soft switch between generation and copy mode, that disengages the model to learn rare and unhelpful self-correspondences, and (ii) a peculiar training procedure, which improves the internal representation capabilities, enhancing recall; it consists in the exchange of encoder and decoder RNNs, – GRUs (Cho et al., 2014c) in our specific case – , depending on whether the input is a tabular Meaning Representation (MR) or a natural language sentence.

We also introduce a new dataset, described in Section 4.3.1, whose particular structure allows to better highlight improvements in copying/recalling abilities with respect to character-based state-of-art approaches.

In Section 4.2 we detail our model: Section 4.2.1 is devoted to explaining the copy mechanism while in Section 4.2.2 our peculiar training procedure is presented. Section 4.3 includes the datasets descriptions, some implementation specifications, the experimental framework and the analysis and evaluation of the achieved results.

4.2 Model Description

4.2.1 Learning to Copy

We build a character-based copy mechanism, depicted in Figure 4.2, inspired by the Pointer-Generator Network (See et al., 2017), a word-based model that hybridizes the Bahdanau traditional model and a Pointer Network (Vinyals et al., 2015). Basing on these ideas, in our model we identify two probability distributions that, differently from what done by See et al. (2017) and Wiseman et al. (2017), *act now on characters* rather than on words: the alphabet distribution P_{alph} and the attention distribution P_{att}.

The former is the network's generative probability of sampling a given character at time t, similarly to eq. (3.10):

$$P^t_{alph} = \text{softmax}(W[d_t; C_t] + b), \tag{4.1}$$

where W and b are trainable parameters.

The latter is the distribution reminded in eq. (3.6), created by the attention mechanism over the input tokens, i.e. in our case, over input characters:

$$P^{tj}_{att} \equiv \alpha_{tj} \tag{4.2}$$

In our method this distribution is used for directly copying characters from the input to the output, pointing their input positions, while in Bahdanau et al. (2015) P_{att} is used only internally to weigh the input annotations and create the context vector C_t.

The final probability of outputting a specific character c is obtained combining P_{alph} and P_{att} through the quantity p^t_{gen}, defined later, which acts as a soft switch between generating c or copying it:

$$P^t(c) = p^t_{gen} \cdot P^t_{alph}(c) + (1 - p^t_{gen}) \sum_{j|x_j=c} P^{tj}_{att}(c), \tag{4.3}$$

where $P^t_{alph}(c)$ is the component of P^t_{alph} corresponding to that character c.

The backpropagation training algorithm, therefore, brings p^t_{gen} close to 1 when it is necessary to generate the output as in a standard Encoder-Decoder with Attention ($P^t(c) \simeq P^t_{alph}(c)$); conversely, p^t_{gen} will be close to 0 (i.e. $P^t(c) \simeq \sum_{j|x_i=c} P^j_{att}(c)$) when a copying step is needed.

The model we propose therefore learns when to sample from P_{alph} for selecting the character to be generated, and when to sample from P_{att} for selecting the character that has to be copied directly from the input.

This copy mechanism is fundamental to output all the unknown words present in the input, i.e. words which never occur in the training set. In fact, generating characters in the right order to reproduce unknown words is a sub-task not "solvable" by a naive sequence-to-sequence model, which learns to output only known words.

The generation probability $p^t_{gen} \in [0, 1]$ is computed as follows:

$$p^t_{gen} = \sigma(W_y \cdot \tilde{y}_{t-1} + W_s \cdot d_t + W_p \cdot p^{t-1}_{gen} + W_c \cdot C_t) \tag{4.4}$$

where σ is the sigmoid function, $\tilde{y}_{t-1}$ is the last output character's embedding, d_t is the current decoder's cell state and C_t is the current context vector. W_y, W_s, W_c and W_p are the parameters whose training allows p^t_{gen} to have the convenient value.

We highlight that in our formulation p^{t-1}_{gen}, i.e. the value of p^t_{gen} at time $t-1$, contributes to the determination of p^t_{gen}. In fact, in a character-based model it is desirable that this probability remains unchanged for a fair number of time steps, to correctly complete the word; knowing its last value helps this behavior. Conversely, in word-based models (such as See et al. (2017)), copying for a single time step, when required, is typically enough.

We also help the model to learn when it is necessary to start a copying phase, using the following formulation of $P(c)$ (Bonetta et al., 2021):

$$P^t(c) = p^t_{gen} \cdot P^t_{alph}(c) + (1 - p^t_{gen}) \sum_{j|x_j=c} P^{t,j-1}_{att}(c) \tag{4.5}$$

Sometimes, our model has difficulty in focusing on the first letter it has to copy. This may be caused by the variety of characters it could be attending on; instead, it seems easier to learn to focus on the most largely seen characters,

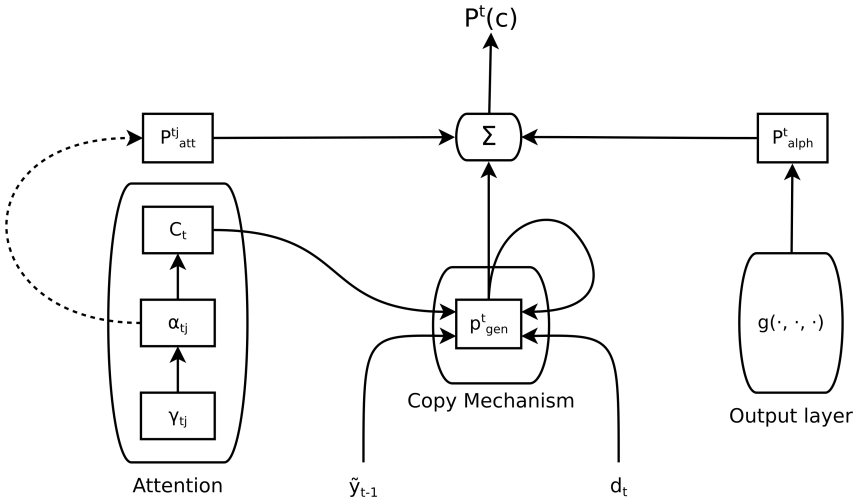


FIGURE 4.2: The copy mechanism included in our model. The final output $P^t(c)$ is the sum of P^t_{alph} and P^{tj}_{att}, weighted by p^t_{gen}.

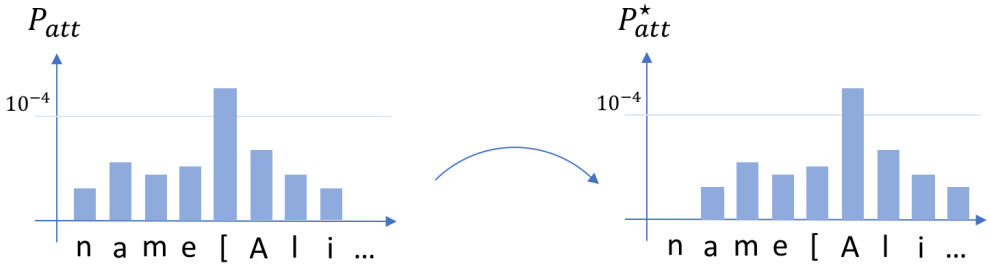


FIGURE 4.3: An example of shifting the attention distribution

as for instance ' ' and '['. As these special characters are very often the prefix of the words we need to copy, when this focus is achieved, we would like the attention distribution to be translated one step to the right, over the first letter that must be copied. Therefore, the final probability of outputting a specific character c, introduced in eq. (4.3), is modified to $P^{t,j-1}_{att}$, i.e. the attention distribution shifted one step to the right and normalized. Figure 4.3 shows the convenience of this approach.

Notice that $P^{t,j-1}_{att}$ is the only shifted probability, while P^t_{alph} remains unchanged. Therefore, if the network is generating the next token (i.e. $p^t_{gen} \simeq 1$), the shift trick does not involve $P^t(c)$ and the network samples the next character from P^t_{alph}, as usual. This means that the shift operation is not degrading the generation ability of the model, whilst improving the copying one.

4.2.2 Switching GRUs

Aiming at improving performance, we enrich our model's training pipeline with an additional phase, which forces an appropriate language representation inside the recurrent components of the model. In order to achieve this goal, the encoder and the decoder *do not own a fixed GRU*, differently from what happens in classical end-to-end approaches. The recurrent module is

passed each time as a parameter, depending on which one of the two training phases is actually performed.

Three main reasons make this switching procedure possible: (i) the character-based architecture, that leads the encoding and decoding RNNs to share the same vocabulary; (ii) the neural networks' effectiveness in Multi-Task Learning, (Abu-Mostafa, 1990; Caruana, 1997; Raffel et al., 2020) – both encoding and generating characters, in this case; (iii) the fact that both RNNs are bidirectional, and the decoder ignores the backward part of the recurrent output, as reported in Section 4.3.4.

In the first phase, similar to the usual one, the GRU assigned to the encoder deals with a tabular representation x as input, the GRU assigned to the decoder has to cope with natural language, and the model generates an output utterance $\tilde{y} = \Phi(x)$. Conversely, in the second phase, GRUs are switched, and we use as input the just obtained natural language utterance $\tilde{y}$ to generate a new table $\tilde{x} = \Gamma(\tilde{y}) = \Gamma(\Phi(x))$. Therefore, the same model can build both Φ and Γ, thanks to the switch of GRUs, as shown by Figure 4.4.

In other words, the learning iteration is performed as follows.

- A dataset example (x, y) is given. x is a tabular meaning representation and y is the corresponding reference sentence.
- We generate an output utterance $\tilde{y} = \Phi(x)$
- We perform an optimization step on the model's parameters, aiming at minimizing $L_{forward} = \text{loss}(\tilde{y}, y)$
- We reconstruct the meaning representation $\tilde{x}$ back from the previously generated output: $\tilde{x} = \Gamma(\tilde{y}) = \Gamma(\Phi(x))$
- We perform a further optimization step on the model's parameters, this time aiming at minimizing $L_{backward} = \text{loss}(\tilde{x}, x)$

The higher training time, direct consequence of the just described technique, is a convenient investment, as it brings an appreciable improvement of the model's performance (see Section 4.3.5).

4.3 Experiments

4.3.1 Datasets

We tested our model on four datasets, whose main descriptive statistics are given in Table 4.1: among them, the most known and frequently used in literature is the E2E dataset (Novikova et al., 2017b), used as benchmark for the E2E Challenge organized by the Heriot-Watt University in 2017. It is a crowdsourced collection of roughly 50,000 instances, in which every input is a list of slot-value pairs and every expected output is the corresponding natural language sentence. The dataset has been partitioned by the challenge organizers in predefined training, validation and test sets, conceived for training data-driven, end-to-end Natural Language Generation models in the restaurant domain. Table 4.2 shows a typical E2E data instance: in this dataset every Meaning Representation (MR) has 8.1 reference sentences on average; in turn,

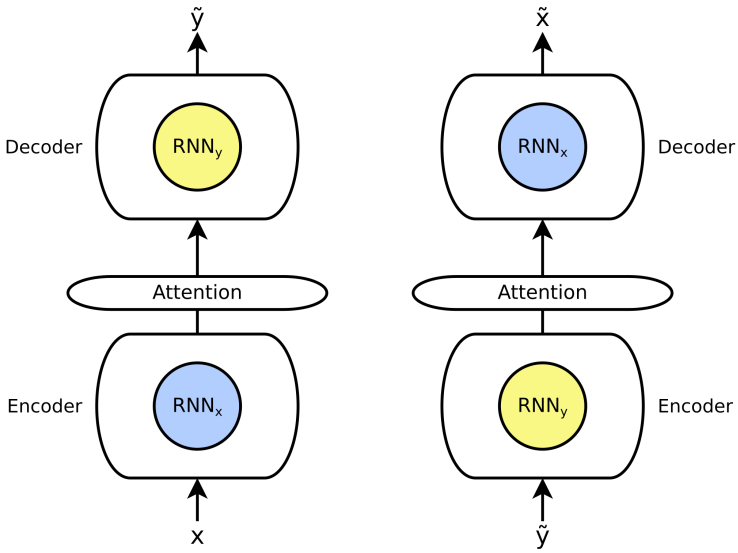


FIGURE 4.4: In the training procedure detailed in Section 4.2.2, the encoder and the decoder do not own a fixed GRU. The configuration on the left models Φ; the one on the right models Γ.

Dataset	Number of instances			Avg. number of characters	
	training	validation	test	MRs	NL sentences
E2E	42,061	4,672	4,693	112.11	115.07
E2E+	42,061	4,672	4,693	112.91	115.65
Hotel	2,210	275	275	52.74	61.31
Restaurant	2,874	358	358	53.89	63.22

TABLE 4.1: Descriptive statistics: on the left, sizes of training, validation and test sets are shown. On the right, the average number of characters, respectively for Meaning Representations and natural language sentences, are presented

each MR is composed of a set of key-value pairs. The ontology consists of 8 attributes of different types.

However, during our experiments, we noticed that the values contained in the E2E dataset are a little naive in terms of variability. In other words, a slot like *name*, that could virtually contain a very broad range of different values, is filled alternating between 19 fixed possibilities. Moreover, values are partitioned among training, validation and test set, in such a way that test set always contains values that are also present in the training set. Consequently, we created a modified version of the E2E dataset, called E2E+, as follows: we selected the slots that represent more copy-susceptible attributes, i.e. *name*, *near* and *food*, and conveniently replaced their values, in both meaning representations and reference sentences. New values for *food* are picked from Wikipedia's list of adjectival forms of countries and nations[1], while both *name* and *near* are filled with New York restaurants' names contained in the Entree dataset presented in Burke et al. (1997). It is worth noting that none of the values of *name* are found in *near*; likewise, values that belong to the training set are not found in the validation set nor in the test one, and vice versa. This value partitioning

[1]https://en.wikipedia.org/wiki/List_of_adjectival_and_demonymic_forms_for_countries_and_nations, consulted on August 30, 2018

Meaning Representation	References
name[The Wrestlers], eatType[coffe shop], food[Indian] priceRange[less than L20] area[city centre] familyFriendly[yes] near[Raja Indian Cuisine]	Indian food meets coffee shop at The Wrestlers located in the city centre near Raja Indian Cuisine. This shop is family friendly and priced at less than 20 pounds. Near Raja Indian Cuisine, The Wrestlers provides the atmosphere of a coffee shop with Indian food. At less than 20 pounds, it provides a family friendly setting for its customers right in the city centre. The Wrestlers is a coffee shop providing Indian food in the less than L20 price range. It is located in the city centre. It is near Raja Indian Cuisine.

TABLE 4.2: An E2E data instance. The Meaning Representation appears in the dataset once for each reference sentence.

shall ensure the absence of generation bias in the copy mechanism, stimulating the models to copy attribute values, regardless of their presence in the training set. The *MR* and *1st reference* fields in Table 4.6 are instances of this new dataset.

Finally, we decided to test our model also on two datasets, Hotel and Restaurant, frequently used in literature (for instance in Wen et al. (2015b) and Goyal et al. (2016)). They are built on a 12 attributes ontology: some attributes are common to both domains, while others are domain specific. Every MR is a list of key-value pairs enclosed in a dialogue act type, such as *inform*, used to present information about restaurants, *confirm*, to check that a slot value has been recognized correctly, and *reject*, to advise that the user's constraints cannot be met. For the sake of compatibility, we filtered out from Hotel and Restaurant all inputs whose dialogue act type was not *inform*, and removed the dialogue act type. Besides, we changed the format of the key-value pairs to E2E-like ones.

Tables are encoded simply converting all characters to ASCII and feeding every corresponding index to the encoder, sequentially. The resulting model's vocabulary is independent of the input, allowing the application of the transfer learning procedure.

4.3.2 Metrics

We evaluated the models' performance on test sets' output utterances using the Evaluation metrics script[2] provided by the E2E NLG Challenge organizers. It rates quality according to five different metrics:

- BLEU (Papineni et al., 2002), a length-penalized precision score over n-grams, $n \in [\![1, 4]\!]$, optionally improved with a smoothing technique (B. Chen & Cherry, 2014).
- NIST (Doddington, 2002), a variant of BLEU which gives more credit to rare n-gram and less credit to common ones.

[2]https://github.com/tuetschek/E2E-metrics

Hyperparameter	Value
Embedding size	16
GRU hidden size	256
N. of recurrent layers	4
Attention size	128
Learning rate	10^{-4}
$\beta_1; \beta_2$ for Adam (Kingma & Ba, 2015a)	0.9; 0.999
Max gradient norm (Pascanu et al., 2013)	1
Batch size	16
Max no. of epochs	30

TABLE 4.3: Model hyperparameters and training settings used in our experiments.

- METEOR (Banerjee & Lavie, 2005), that tries to overcome the fact that BLEU does not take recall into account, and it only allows exact n-gram matching. Hence, METEOR uses the F-measure and a relaxed matching criterion.
- ROUGE_L (C.-Y. Lin, 2004), based on a variation of the F-measure where the precision and recall are computed using the length of the longest common subsequence between hypothesis and reference.
- CIDER (Vedantam et al., 2015), that weighs each hypothesis's n-gram based on its frequency in the reference set and in the entire corpus. The underlying idea is that frequent dataset's n-grams are less likely to be informative/relevant.

4.3.3 Baselines

In order to show the effectiveness of our proposed Encoder-Decoder model with Attention, Copy and Switch (hereafter EDA_CS), we compare it with the following models:

- EDA, a character-based Encoder-Decoder model with Attention (Bahdanau et al., 2015), a standard baseline in literature (Agarwal & Dymetman, 2017; Goyal et al., 2016; See et al., 2017).
- TGen (Dusek & Jurcícek, 2016), the strong word-based baseline of the E2E challenge (Dusek et al., 2018). Its pipeline consists of a delexicalizer, a neural Encoder-Decoder system which outputs a syntax tree using beam search with reranking, a surface realizer, and a relexicalizer.

4.3.4 Implementation Details

We developed EDA and EDA_CS using the PyTorch framework[3], release 0.4.1[4]. The training has been carried out as described in Subsection 4.2.2: this training procedure needs the two GRUs to have the same dimensions, in terms of input size, hidden size, number of layers and presence of a bias term. Moreover, they

[3]Code and datasets are publicly available at https://github.com/marco-roberti/char-dtt-tailored

[4]https://pytorch.org/

EDA	BLEU	0.4999	EDA_S	BLEU	0.6538
	NIST	7.1146		NIST	8.4601
	METEOR	0.3369		METEOR	0.4337
	ROUGE_L	0.5634		ROUGE_L	0.6646
	CIDER	1.3176		CIDER	1.9944
EDA_C	BLEU	0.6255	EDA_CS	BLEU	**0.6705**
	NIST	7.7934		NIST	**8.5150**
	METEOR	0.4401		METEOR	**0.4449**
	ROUGE_L	0.6582		ROUGE_L	**0.6894**
	CIDER	1.7286		CIDER	**2.2355**

TABLE 4.4: The ablation study on the E2E dataset. All five metrics considered in Section 4.3.2 are reported next to each model. The study evidences the final performance improvement reached by our model. Best values for each metric are highlighted (the higher the better)

both have to be bidirectional, even if the decoder ignores the backward part of its current GRU. We minimize the negative log-likelihood loss using teacher forcing (Williams & Zipser, 1989) and Adam (Kingma & Ba, 2015a), the latter being an optimizer that computes individual adaptive learning rates. As a consequence of the length of the input sequences, a character-based model is often subject to the exploding gradient problem, that we solved via the well-known technique of gradient norm clipping (Pascanu et al., 2013). The training stopping criterion was based on the absence of models' performance improvements (Dusek & Jurcícek, 2016).

Three-fold cross-validation was used to find the optimal hyperparameters and training settings values, using the BLEU metric (Papineni et al., 2002) for evaluating each model. In the resulting configuration shown in Table 4.3, our model has $9,719,920$ trainable parameters.

Training and inference have been performed on 24GB NVIDIA GPUs (TITAN RTX and Quadro P6000). The training time is in the order of magnitude of $\sim$10 hours, depending on the hardware. Generation at inference occurs in real time, i.e. roughly 2 minutes for the 4,693 test instances.

As for the TGen baseline, we used the code originally provided by Dusek and Jurcícek (2016)[5].

4.3.5 Results and Discussion

In order to show that our model represents an effective and relevant improvement, we carry out two different experimentations: an ablation study and a quantitative and qualitative analysis, in comparison with the baselines described in Section 4.3.3

Our first experimentation, the **ablation study**, refers to the E2E dataset because of its wide diffusion, and is shown in Table 4.4; "EDA_CS" identifies our model, and 'C' and 'S' stand for "Copy" and "Switch", the two major improvements presented in this work. It is evident that the partially-improved

[5]https://github.com/UFAL-DSG/tgen

networks are able to provide independent benefits to the performance. Those components cooperate positively, as EDA_CS further enhances those results. Furthermore, the obtained BLEU metric value on the E2E test set would allow our model to be ranked fourth in the E2E NLG Challenge, while its baseline TGen was ranked tenth.

Our second experimentation, the **comparison study**, is shown in Table 4.5. The character-based design of EDA_CS led us to explore in this context also a possible behavior as a transfer learning capable model: in order to test this hypothesis, we used the weights learned during training on the E2E+ dataset as the starting point for a fine-tuning phase on all the other datasets. We chose E2E+ because it reduces the generation bias, as discussed in Subsection 4.3.1. We named this approach EDA_CS^{TL}.

A first interesting result is that our model EDA_CS always obtains higher metric values with respect to TGen on the Hotel and Restaurant datasets, and three out of five higher metrics values on the E2E dataset. However, in the case of E2E+, TGen achieves three out of five higher metrics values. These results suggest that EDA_CS and TGen are comparable, at least from the point of view of automatic metrics' evaluation.

A more surprising result is that the approach EDA_CS^{TL} allows to obtain better performance with respect to training EDA_CS in the standard way on the Hotel and Restaurant datasets (for the majority of metrics); on E2E, EDA_CS^{TL} outperforms EDA_CS only in one case (i.e. METEOR metric).

Moreover, EDA_CS^{TL} shows a BLEU increment of at least 14% with respect to TGen's score when compared to both Hotel and Restaurant datasets.

Finally, the baseline model, EDA, is largely outperformed by all other examined methods. Notice that its scores do not drop below a certain threshold because, even if new names are not correctly reproduced, values occurring in other fields of the generated sentences are generally still correct. We hypothesize that their performances would be even worse on datasets containing unseen values on other fields as well (e.g. *food*, *near*).

Therefore, we can claim that our model exploits its transfer learning capabilities effectively, showing very good performances in a context like data-to-text generation in which the portability of features learned from different datasets, in the extent of our knowledge, has not yet been explored.

We highlight that EDA_CS's model's good results are achieved even if it consists in a fully end-to-end model which does not benefit from the delexicalization-relexicalization procedure, differently from TGen. Most importantly, the latter represents a word-based system: as such, it is bound to a specific, limited vocabulary, in contrast to the general-purpose character one used in our work.

Table 4.6 reports the output of the analyzed models for a couple of MR, taken from the E2E+ test set. The EDA's inability to copy is clear, as it tends, in its output, to substitute those values of *name*, *food* and *near* that do not appear in the training set with known ones, guided by the first few characters of the input slot's content. Besides, it shows serious coverage issues, frequently 'forgetting' to report information, and/or repeating more times the same ones.

		E2E+	E2E	Hotel	Restaurant
EDA	BLEU	0.3773	0.4999	0.4316	0.3599
	NIST	5.7835	7.1146	5.9708	5.5104
	METEOR	0.2672	0.3369	0.3552	0.3367
	ROUGE_L	0.4638	0.5634	0.6609	0.5892
	CIDER	0.2689	1.3176	3.9213	3.3792
TGen	BLEU	**0.6292**	0.6593	0.5059	0.4074
	NIST	**9.4070**	**8.6094**	7.0913	6.4304
	METEOR	0.4367	0.4483	0.4246	0.3760
	ROUGE_L	**0.6724**	0.6850	0.7277	0.6395
	CIDER	2.8004	2.2338	5.0404	4.1650
EDA_CS	BLEU	0.6197	**0.6705**	0.5515	0.4925
	NIST	9.2103	8.5150	**7.4447**	6.9813
	METEOR	**0.4428**	0.4449	0.4379	0.4191
	ROUGE_L	0.6610	**0.6894**	0.7499	0.7002
	CIDER	**2.8118**	**2.2355**	5.1376	4.7821
EDA_CS^{TL}	BLEU	-	0.6580	**0.5769**	**0.5099**
	NIST	-	8.5615	7.4286	**7.3359**
	METEOR	-	**0.4516**	**0.4439**	**0.4340**
	ROUGE_L	-	0.6740	**0.7616**	**0.7131**
	CIDER	-	2.1803	**5.3456**	**4.9915**

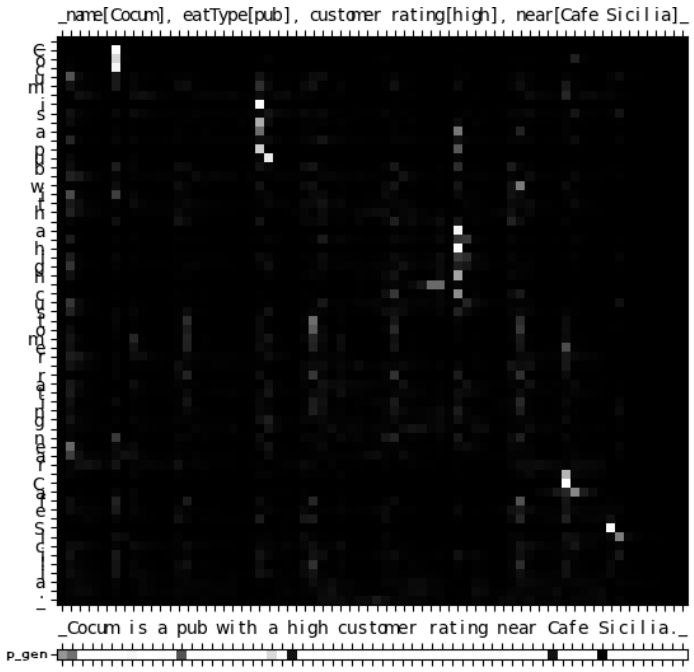
TABLE 4.5: Performance comparison, according to the five metrics considered in Section 4.3.2, reported next to each model. Note the absence of transfer learning on the E2E dataset, as in this case the training and fine-tuning datasets are the same. Best values for each metric are highlighted (the higher the better)

These troubles are not present in EDA_CS output utterances: the model nearly always renders all of the input slots, still without duplicating any of them. This goal is achieved even in absence of explicit coverage techniques thanks to our peculiar training procedure, detailed in Section 4.2.2, that for each input sample minimizes also the loss on the reconstructed tabular input. It is worth noting that the performance of TGen and EDA_CS are overall comparable, especially when they deal with names or other expressions not present in training.

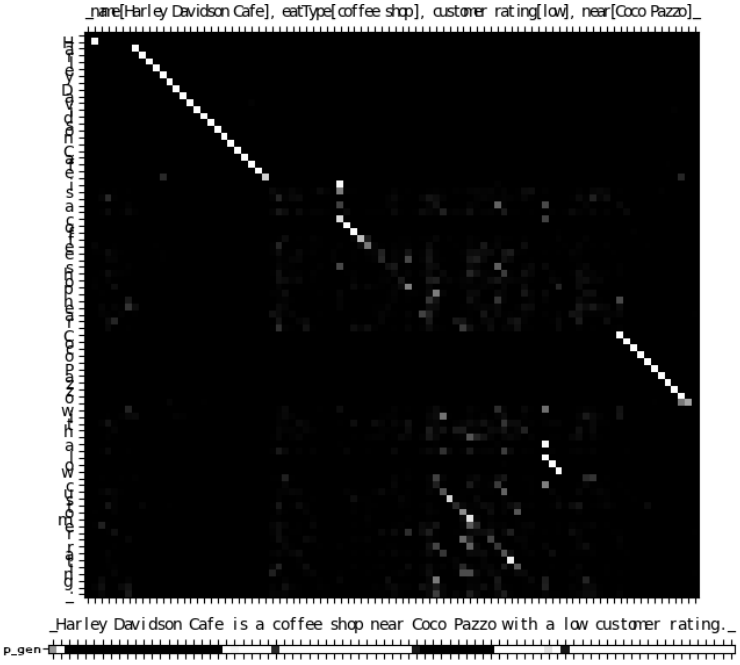
The joint analysis of the matrix of the attention distribution P_{att}^{tj} and the vector p_{gen}^{t} allows a deeper understanding of how our model works.

In Figure 4.5 every row shows the attention probability distribution "seen" when an output character is produced at the t-th time instant (i.e. the vector $P_{att}^{tj}, 1 \leq j \leq T_x$), while every column shows values of the attention distribution corresponding to a specific input position j (i.e. the vector $P_{att}^{tj}, 1 \leq t \leq T_y$). We can therefore follow the white spots, corresponding to higher values of attention, to understand the flow of the model's attention during the generation of the output utterance.

Moreover, p_{gen}^{t} values, which lie in the numeric interval $[0, 1]$, help us in the



(A) *On an E2E instance.*



(B) *On an E2E+ instance.*

FIGURE 4.5: Attention matrix and vector p_{gen}, as calculated by the model. The former is a matrix of values between 0 (black, less attention) and 1 (white, more attention) with one column for each input character and one row for each generated one. The i-th attention row is associated to the i-th value of p_{gen}, which ranges from 0 (black, copy) to 1 (white, generation)

MR	`name[New Viet Huong], eatType[pub], customer rating[1 out of 5], near[Ecco]`
1st reference	The New Viet Huong is a pub near Ecco that has a customer rating of 1 out of 5.
EDA_CS	New Viet Huong is a pub near Ecco with a customer rating of 1 out of 5.
TGen	New Viet Huong is a pub near Ecco with a customer rating of 1 out of 5.
EDA	Near the riverside near the ERNick Restaurant is a pub near the ER-Nicker's.
MR	`name[La Mirabelle], eatType[restaurant], food[Iraqi], priceRange[high], area[riverside], familyFriendly[yes], near[Mi Cocina]`
1st reference	La Mirabelle is a children friendly restaurant located in the Riverside area near to the Mi Cocina. It serves Iraqi food and is in the high price range.
EDA_CS	La Mirabelle is a high priced Iraqi restaurant located in the riverside area near Mi Cocina. It is children friendly.
TGen	La Mirabelle is a high priced Iraqi restaurant in the riverside area near Mi Cocina. It is child friendly.
EDA	La Memaini is a high priced restaurant that serves Iranian food in the high price range. It is located in the riverside area near Manganaro's Restaurant.

TABLE 4.6: A comparison of the three models' output on some MR of the E2E+ test set. The first reference utterance is reported for convenience

interpretation of the attention: they are represented as a grayscale vector from zero (black) to one (white) under the matrices. Values close to 0 mean copying and those near 1 mean generating.

We can note that our model's behavior varies significantly depending on the dataset it has been trained on. Figure 4.5a shows the attention probability distribution matrix of EDA_CS (together with p^t_{gen} vector) trained on the E2E dataset: as observed before, attribute values in this dataset have a very low variability (and are already present in the training set), so that they can be individually represented and easily generated by the decoder. In this case, a typical pattern is the copy of only the first, discriminating character, clearly noticeable in the graphical representation of the p^t_{gen} vector, and the subsequent generation of the others. Notice that the attention tends to remain improperly focused on the same character for more than one output time step, as in the first letter of "high".

On the other hand, the copy mechanism shows its full potential when the system must learn to copy attribute values, as in the E2E+ dataset. In Figure 4.5b the diagonal attention pattern is pervasive: (i) it occurs when the model actually copies, as in "Harley Davidson" and "Coco Pazzo", and (ii) as a *soft track* for the generation, as in "customer rating", where the copy-first-generate-rest behavior emerges again.

A surprising effect is shown in Figure 4.6, when the model is expected to copy words that, instead, are usually generated: an initial difficulty in copying the word "The", that is usually a substring of a slot value, is ingeniously overcome as follows. The first character is purely generated, as shown by the white color in the underlying vector, and the sequence of the following characters,

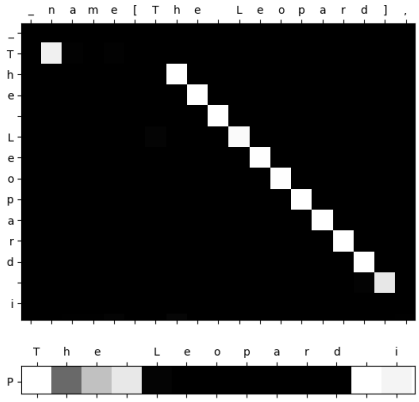


FIGURE 4.6: Copying common words leads the model to "uncertain" values of p^t_{gen}

"he_", is half-generated and half-copied. Then, the value of p^t_{gen} gets suddenly but correctly close to 0 (black) until the closing square bracket is met. The network's output is not affected negatively by this confusion and the attention matrix remains quite well-formed.

As a final remark, the metrics used, while being useful, well-known, and broadly accepted, do not reflect the ability to directly copy input facts to produce outputs, so settling the rare word problem.

4.4 Limitations and future work

The major drawback of relying on a character-based model is the increased length of the sequences it deals with. As an example, the average English word length is 4.7 characters (Mayzner & Tresselt, 1965); therefore, one may expect a roughly $5\times$ increase when switching from words to characters.

Recent developments in the field, however, may mitigate this issue in various ways: (i) the non-autoregressive nature of the more recent Transformer model (Vaswani et al., 2017) allows for a more aggressive parallelization of the computation flow than RNN-based methods such as EDA_CS. The quadratic space complexity of the former, that is generally considered as its major performance bottleneck, has already been faced in various ways (Child et al., 2019; Choromanski et al., 2021; Dai et al., 2019; Kitaev et al., 2020); (ii) the introduction of pre-trained character-based models, such as (Ma et al., 2020), should sharply reduce the training time requirements; and (iii) the hardware improvements in GPU and TPU's design and parallelization techniques remain a strong ongoing trend nowadays. Future work should include the incorporation and adaptation of such novel techniques to the model presented in this Chapter.

Chapter 5

Controlling Hallucinations at Word Level

5.1 Introduction

In this Chapter, we specifically address the issue of hallucinations, which is currently regarded as a major issue in DTT (Narayan & Gardent, 2020). Indeed, experimental surveys show that real-life end-users of DTT systems care more about reliability than about readability (Reiter & Belz, 2009), as unfaithful texts can potentially mislead decision makers, with dire consequences.

Even if the concept of hallucination is intuitive, a formal and universally accepted definition has not been stated yet. In this work, we stick to the one given by Dhingra et al. (2019): "Hallucination (K. Lee et al., 2019; Rohrbach et al., 2018) refers to when an NLG system generates text which mentions extra information than what is present in the source from which it is generated.". Dhingra et al. (2019) also state that the phenomenon occurs when "the reference contains extra information which no system can be expected to produce given only the associated table. We call such reference texts divergent from the table."

Hallucinations-reduction methods such as the one presented here have applications in a broad range of tasks requiring high reliability, like news reports (Leppänen et al., 2017), in which hallucinations may give rise to *fake news*, or summaries of patient information in clinical contexts (Banaee et al., 2013; Portet et al., 2009). When corpora include a mild amount of noise, as in handcrafted ones (e.g. E2E, WebNLG), dataset regularization techniques (Dusek et al., 2019; Nie et al., 2019) or hand crafted rules (Juraska et al., 2018) can help to reduce hallucinations. Unfortunately, these techniques are not suited to more realistic and noisier datasets, as for instance WikiBio (Lebret et al., 2016) or RotoWire (Wiseman et al., 2017). On these benchmarks, several techniques have been proposed, such as reconstruction loss terms (S. Lin et al., 2020; H. Wang, 2019; Wiseman et al., 2017) or Reinforcement Learning (RL) based methods (Liu et al., 2019c; Perez-Beltrachini & Lapata, 2018; Rebuffel et al., 2020b). These approaches suffer however from different issues: (1) the reconstruction loss relies on the hypothesis of one-to-one alignment between source and target which does not fit with content selection in DTT; (2) RL–trained models are based on instance-level rewards (e.g. BLEU (Papineni et al., 2002), PARENT (Dhingra et al., 2019)) which can lead to a loss of signal because divergences occur at the word level. In practice, parts of the target sentence

KEY	VALUE
name	kian emadi
fullname	kian emadi-coffin
currentteam	retired
discipline	track
role	rider
ridertype	sprinter
proyears	2012-present
proteams	sky track cycling

Ref.: kian emadi (born 29 july 1992) is a british track cyclist .

FIGURE 5.1: An example of a WikiBio instance, composed by an input table and its (partially aligned) description.

express source attributes (in Fig. 5.1 name and occupation fields are correctly realized), while others diverge (the birthday and nationality of Kian Emadi are not supported by the source table).

Interestingly, one can view DTT models as Controlled Text Generation (CTG) ones focused on controlling content, as most CTG techniques condition the generation on several key-value pairs of *control factors* (e.g. tone, tense, length) (Dong et al., 2017; Ficler & Goldberg, 2017; Hu et al., 2017). Recently, Filippova (2020) explicitly introduced CTG to DTT by leveraging an *hallucination score* simply attached as an additional attribute which reflects the amount of noise in the instance. As an example, the table from Fig 5.1 can be augmented with an additional line (`hallucination_score`, `80%`)[1]. However, this approach requires a strict alignment at the instance-level, namely between control factors and target text. A first attempt towards word-level approaches is proposed by Perez-Beltrachini and Lapata (2018) (also *PB&L* in the following). They design word-level alignment labels, denoting the correspondence between the text and the input table, to bootstrap DTT systems. However, they incorporate these labels into a sentence-level RL-reward, which ultimately leads to a loss of this finer-grained signal.

In this Chapter, we go further in this direction with a DTT model by fully leveraging word-level alignment labels with a CTG perspective. We propose an original approach in which the word-level is integrated at all phases:

- we propose a **word-level labeling procedure** (Section 5.3), based on co-occurrences and sentence structure through dependency parsing. This mitigates the failure of strict word-matching procedure, while still producing relevant labels in complex settings.
- we introduce a **weighted multi-branch neural decoder** (Section 5.4), guided by the proposed alignment labels, acting as word-level control

[1] The reader may disagree with such a strong hallucination score. Indeed, while the birthdate and nationality are clearly divergences, the rest of the sentence is correct. This illustrates the complexity of handling divergences in complex datasets, where alignment cannot be framed as a simple word-matching task.

factors. During training, the model is able to distinguish between aligned and unaligned words and learns to generate accurate descriptions without being misled by un-factual reference information. Furthermore, our multi-branch weighting approach enables control at inference time.

We carry out extensive experiments on WikiBio, to evaluate both our labeling procedure and our decoder (Section 5.6). We also test our framework on ToTTo (Parikh et al., 2020), in which models are trained with noisy reference texts, and evaluated on references reviewed and cleaned by human annotators to ensure accuracy. Evaluations are based on a range of automated metrics as well as human judgments, and show increased performances regarding hallucinations reduction, while preserving fluency.

Moreover, our approach makes training neural models on noisy datasets possible, without the need to handcraft instances. This work shows the benefit of word-level techniques, which leverage the entire training set, instead of removing problematic training samples, which may form the great majority of the available data.

5.2 Related work

Handling hallucinations in noisy datasets. The use of Deep Learning based methods to solve DTT tasks has led to sudden improvements in state of the art performances (Lebret et al., 2016; Liu et al., 2018; Puduppully et al., 2019a; Wiseman et al., 2017). As a key aspect in determining a model's performance is the quality of training data, several large corpora have been introduced to train and evaluate models' abilities on diverse tasks. E2E (Novikova et al., 2017c) evaluates surface realization, i.e. the strict transcription of input attributes into natural language; RotoWire (Wiseman et al., 2017) pairs statistics of basketball games with their journalistic descriptions, while WikiBio (Lebret et al., 2016) maps a Wikipedia info-box with the first paragraph of its associated article. Contrary to E2E, the latter datasets are not limited to surface realization. They were not constructed by human annotators, but rather created from Internet sources, and consist of loosely aligned table-reference pairs: in WikiBio, almost two thirds of the training instances contain divergences (Dhingra et al., 2019), and no instance has a 1-to-1 source-target alignment (Perez-Beltrachini & Gardent, 2017).

On datasets with a moderate amount of noise, such as E2E, data pre-processing has proven effective for reducing hallucinations. Indeed, rule-based (Dusek et al., 2019) or neural-based methods (Nie et al., 2019) have been proposed, specifically with table regularization techniques, where attributes are added or removed to re-align table and target description. Several successful attempts have also been made in automatically learning alignments between the source tables and reference texts, benefiting from the regularity of the examples (Gehrmann et al., 2018; Juraska et al., 2018; Shen et al., 2020). For instance, Juraska et al. (2018) leverage templating and hand-crafted rules to re-rank the top outputs of a model decoding via beam search; Gehrmann et al. (2018) also leverage the possible templating formats of E2E's reference texts, and train an ensemble of decoders where each decoder is associated to one template; and Kasner and Dusek (2020) produce template-based lexicalizations and improve them via a *sentence fusion* model. The previous techniques are not applicable

in more complex, general settings. The work of Dusek et al. (2019) hints at this direction, as authors found that neural models trained on E2E were principally prone to omissions rather than hallucinations. In this direction, Shen et al. (2020) were able to obtain good results at increasing the coverage of neural outputs, by constraining the decoder to focus its attention exclusively on each table cell sequentially until the whole table was realized. On more complex datasets (e.g. WikiBio), a wide range of methods has been explored to deal with factualness such as loss design, either with a reconstruction term (H. Wang, 2019; Wiseman et al., 2017) or with RL-based methods (Liu et al., 2019c; Perez-Beltrachini & Lapata, 2018; Rebuffel et al., 2020b). Similarly to the coverage constraints, a reconstruction loss has proven only marginally efficient in these settings, as it contradicts the content selection task (H. Wang, 2019), and needs to be well calibrated using expert insight in order to bring improvements. Regarding RL, Perez-Beltrachini and Lapata (2018) build an instance-level reward which sums up word-level scores; Liu et al. (2019c) propose a reward based on document frequency to favor words from the source table more than rare words; and Rebuffel et al. (2020b) train a network with a variant of PARENT (Dhingra et al., 2019) using self-critical RL. Note that data regularization techniques have also been proposed (Thomson et al., 2020; H. Wang, 2019), but these methods require heavy manual work and expert insights, and are not readily transposable from one domain to another.

From CTG to controlling hallucinations. Controlled Text Generation (CTG) is concerned with constraining a Language Model's output during inference on a number of desired attributes, or *control factors*, such as the identity of the speaker in a dialog setting (Li et al., 2016), the politeness of the generated text or the text length in machine-translation (Kikuchi et al., 2016; Sennrich et al., 2016a), or the tense in generated movie reviews (Hu et al., 2017). Earlier attempts at neural CTG can even be seen as direct instances of DTT as it is currently defined: models are trained to generate text conditioned on attributes of interest, where attributes are key-value pairs. For instance, in the movie review domain, Ficler and Goldberg (2017) proposed an expertly crafted dataset, where sentences are strictly aligned with control factors, being either content or linguistic style aspects (e.g. tone, length).
In the context of dealing with hallucinations in DTT, Filippova (2020) recently proposed a similar framework, by augmenting source tables with an additional attribute that reflects the degree of hallucinated content in the associated target description. During inference, this attribute acts as an *hallucination handle* used to produce more or less factual text. As mentioned in Section 5.1, we argue that a unique value can not accurately represent the correspondence between a table and its description, due to the phrase-based nature of divergences.

Based on the literature review, the lack of model control can be evidenced when loss modification methods are used (Liu et al., 2019a; Rebuffel et al., 2020b; H. Wang, 2019), although these approaches can be efficient and transposed from one domain to another. On the other hand, while CTG deals with control and enables choosing the defining features of generated texts (Filippova, 2020), standard approaches rely on instance-level control factors that do not fit with hallucinations, which rather appear due to divergences at the word

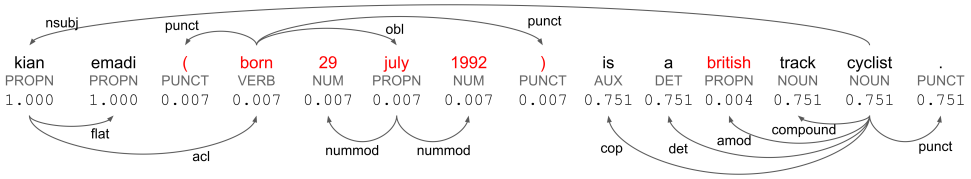


FIGURE 5.2: The reference sentence of the example shown in Fig. 5.1. Every token is associated to its Part-of-Speech tag and hallucination score s_t. Words in red denote $s_t < \tau$. The dependency parsing is represented by labeled arrows that flow from parents to children. Important words are *kian*, *emadi*, *29*, *july*, *1992*, *british*, *track*, and *cyclist*.

level. Our approach aims at gathering the merits of both trends of models and is guided by previous statements highlighting that word-level is primary in hallucination control. More particularly, our model differs from previous ones in several aspects:

(1) Contrasting with data-driven approaches (i.e. dataset regularization) which are costly in expert time, and loss-driven approaches (i.e. reconstruction or RL losses) which often do not take into account key subtasks of DTT (content-selection, world-level correspondences), we propose a multi-branch modeling procedure which allows the controllability of the hallucination factor in DTT. This multi-branch model can be integrated seamlessly in current approaches, allowing to keep peculiarities of existing DTT models, while deferring hallucination management to a parallel decoding branch.

(2) Unlike previous CTG approaches (Ficler & Goldberg, 2017; Filippova, 2020; Li et al., 2016; Sennrich et al., 2016a) which propose instance-level control factors, the control of the hallucination factor is performed at the word-level to enable finer-grained signal to be sent to the model.

Our model is composed of two main components: (1) a word-level alignment labeling mechanism, which makes the correspondence between the input table and the text explicit, and (2) a multi-branch decoder guided by these alignment labels. The branches separately integrate co-dependent control factors (namely content, hallucination and fluency). We describe these components in Sections 5.3 and 5.4, respectively.

5.3 Word-level Alignment Labels

We consider a DTT task, in which the corpus $\mathcal{C}$ is composed of a set of entity-description pairs, (e, y). A *single-entity table* e is a variable-sized set of T_e key-value pairs $x_j := (k_j, v_j)$, $j = 1, \ldots, T_e$. A *description* $y := y_{1:T_y}$ is a sequence of T_y tokens representing the natural language description of the entity; we refer to the tokens spanning from indices t to t' of a description y as $y_{t:t'}$. A description is made of *statements*, defined as text spans expressing one single idea (Appendix A.1 presents in detail the statement partitioning procedure). We refer to the first index of a statement as t_i, so that $y_{t_i:t_{i+1}-1}$ is the i^{th} statement itself. Fig. 5.1 shows a WikiBio entity made by 8 key-value pairs together with its associated description.

First, we aim at labeling each word from a description, depending on the presence of a correspondence with its associated table. We call such labels *alignment labels*. We drive the word-level labeling procedure on two intuitive constraints: (1) important words (names, adjectives and numbers) should be labeled depending on their alignment with the table, and (2) words from the same statement should have the same label.

With this in mind, the *alignment label* for the t^{th} token y_t is a binary label: $l_t \coloneqq \mathbb{1}_{\{s_t > \tau\}}$ where s_t refers to the *alignment score* between y_t and the table, and τ is set experimentally (see Sec. 5.5.3). The *alignment score* s_t acts as a normalized measure of correspondence between a token y_t and the table e:

$$s_t \coloneqq \text{norm}(\max_{x \in e} \text{align}(y_t, x),\ \ y) \tag{5.1}$$

where the function *align* estimates the alignment between token y_t and a key-value pair x from the input table e, and *norm* is a normalization function based on the dependency structure of the description y. Fig. 5.2 illustrates our approach: under each word we show its word alignment score, and words are colored in red if this score is lower than τ, denoting an alignment label equal to 0. Below, we describe these functions (Appendix A.1 contains reproducibility details).

Co-occurrence-based alignment function (align$(\cdot, \mathbf{x})$). This function assigns to important words a score in the interval $[0, 1]$ proportional to their co-occurrence count (a proxy for alignment) with the key-value pair from the input table. If the word y_t appears in the key-value pair $x \coloneqq (k, v)$, $align(y_t, x)$ outputs 1; otherwise, the output is obtained scaling the number of occurrences $co_{y_t,x}$ between y_t and x through the dataset:

$$\text{align}(y_t, x) \coloneqq \begin{cases} 1 & \text{if } \ y_t \in x \\ a \cdot (co_{y_t,x} - m)^2 & \text{if } \ m \leq co_{y_t,x} \leq M \\ 0 & \text{if } \ 0 \leq co_{y_t,x} \leq m \end{cases} \tag{5.2}$$

where M is the maximum number of word co-occurrences in the dataset vocabulary and the row x, m is a threshold value, and $a \coloneqq \frac{1}{(M-m)^2}$.

Score normalization (norm$(\cdot, \mathbf{y})$). According to the already stated assumption (2) – words inside the same statement should have the same score – , we first split the sentence y into statements $y_{t_i:t_{i+1}-1}$, via dependency parsing and its rule-based conversion to constituency trees (Borensztajn et al., 2009; Han et al., 2000; Hwa et al., 2005; Xia & Palmer, 2001). Given a word y_t associated to the score s_t and belonging to statement $y_{t_i:t_{i+1}-1}$, its normalized score corresponds to the average score of all important words in this statement:

$$\text{norm}(s_t, y) = \frac{1}{t_{i+1} - t_i} \sum_{j=t_i}^{t_{i+1}-1} s_j \tag{5.3}$$

This in-statement average depends on both the specific word and its context, leading to coherent hallucination scores which can be thresholded without affecting the syntactical sentence structure, as shown in Fig. 5.2.

5.4 Multi-Branch Architecture

The proposed Multi-Branch Decoder (MBD) architecture aims at separating targeted co-dependent factors during generation. We build upon the standard DTT architecture, an encoder-decoder with attention and copy mechanism, which we modify by duplicating the decoder module into three distinct parallel modules. Each control factor (i.e. content, hallucination or fluency) is modeled via a single decoding module, also called branch, whose output representation can be weighted according to its desired importance. At training time, weights change depending on the word currently being decoded, inducing the desired specialization of each branch. During inference, weights are manually set, according to the desired trade-off between information reliability, sentence diversity and global fluency. Text generation is thus controllable, and consistent with the control factors.

Figure 5.3 illustrates a training step over the sentence "*Giuseppe Mariani was an Italian art director*", in which *Italian* is a divergent statement (i.e. is not supported by the source table). While decoding factual words, the weight associated to the content (resp. hallucination) branch is set to 0.5 (resp. 0) while during the decoding of *Italian*, the weight associated to the content (resp. hallucination) branch is set to 0 (resp. 0.5). Note that the weight associated to the fluency branch is always set to 0.5, as fluency does not depend on factualness.

The decoding modules' actual architecture may vary, as we framed the MBD model from a high-level perspective. Therefore, all types of decoder can be used, such as Recurrent Neural Networks (RNNs) (Rumelhart et al., 1986), Transformers (Vaswani et al., 2017), and Convolutional Neural Networks (Gehring et al., 2017). The framework can be generalized to different merging strategies as well, such as late fusion, in which the final distributions are merged, instead of the presented early fusion, which works at the decoder states level.

In this Chapter, experiments are carried out on RNN-based decoders, weighting their hidden states. As stated above, the Transformer architecture (Vaswani et al., 2017) is perfectly compatible with our framework; however, its decoding module is slower than RNNs at inference (Zhang et al., 2018), as it recomputes the attentions over the whole sequence at every step. This drawback would be worsened by the multi-branch architecture, hence the choice of sticking to a more agile recurrent decoder.

The model works at the word level: character-based approaches, such as the one presented in Chapter 4, are not appropriate, as they do not preserve the one-to-one correspondence between word-level hallucination labels and the neural model's tokens. The need for word-level hallucination labels has been discussed in Section 5.2.

Section 5.4.1 presents the standard DTT encoder-decoder architecture; Section 5.4.2 shows how it can be extended to MBD, together with its peculiarities and the underlying objectives and assumptions.

5.4.1 Standard DTT architecture

Neural DTT approaches typically use an encoder-decoder architecture (Wiseman et al., 2017) in which (1) the encoder relies on a RNN to encode each element of the source table into a fixed-size latent representation h_j (elements

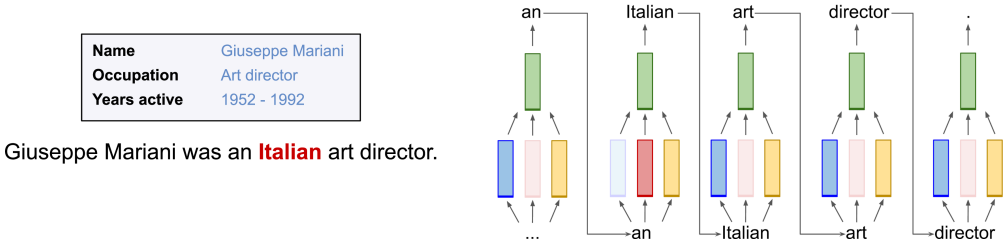


FIGURE 5.3: Our proposed decoder with three branches associated to content (in blue – left), hallucination (in red – middle) and fluency (in yellow – right). Semi-transparent branches are assigned the weight 0.

of the input table are first embedded into T_e N-dimensional vectors, and then fed sequentially to the RNN (Wiseman et al., 2017)), and (2) the decoder generates a textual description y using a RNN augmented with attention and copy mechanisms (See et al., 2017). Words are generated in an auto-regressive way. The decoder's RNN updates its hidden state d_t as:

$$d_t := \text{RNN}(d_{t-1}, [y_{t-1}, C_t]) \tag{5.4}$$

where y_{t-1} is the previous word and C_t is the context vector obtained through the attention mechanism. Finally, a word is drawn from the distribution computed via a copy mechanism (See et al., 2017).

5.4.2 Controlling Hallucinations via a Multi-Branch Model

Our objective is to enrich the decoder in order to be able to tune the content/hallucination ratio during generation, aiming at enabling generation of hallucination-free text when needed. Our key assumption is that the decoder's generation is conditioned by three co-dependent factors:

- *Content factor* constrains the generation to realize only the information included in the input;
- *Hallucinating factor* favors lexically richer and more diverse text, but may lead to hallucinations not grounded by the input;
- *Fluency factor*[2] conditions the generated sentences toward global syntactic correctness, regardless of the relevance.

Based on this assumption, we propose a multi-branch encoder-decoder network, whose branches are constrained on the above factors at word-level, as illustrated in Fig. 5.3. Our network has a single encoder and $F = 3$ distinct decoding RNNs, noted RNN^f respectively, one for each factor. During each decoding step, the previously decoded word y_{t-1} is fed to all RNNs, and a

[2]Wiseman et al. (2018) showed that the explicit modeling of a fluency latent factor improves performance.

final decoder state d_t is computed using a weighted sum of all the corresponding hidden states,

$$d_t^f := \text{RNN}^f(d_{t-1}^f, [y_{t-1}, C_t]) \tag{5.5}$$

$$d_t := \sum_{f=1}^{F} \omega_t^f d_t^f \tag{5.6}$$

where d_t^f and ω_t^f are respectively the hidden state and the weight of the f^{th} RNN at time t. Weights are used to constrain the decoder branches to the desired control factors ($\omega_t^0, \omega_t^1, \omega_t^2$ for the content, hallucination and fluency factors respectively) and sum to one.

During training, their values are dynamically set depending on the *alignment label* $l_t \in \{0, 1\}$ of the target token y_t (see Sec. 5.5.3). While a number of mappings can be used to set the weights given the alignment label, early experiments have shown that better results were achieved when using a binary switch for each factor, i.e. activating/deactivating each branch, as shown in Fig. 5.3 (note that fluency should not depend on content and therefore its associated branch is always active).

During inference, the weights of the decoder's branches are set manually by a user, according to the desired trade-off between information reliability, sentence diversity and global fluency. Text generation is then controllable and consistent with the control factors.

5.5 Experimental setup

5.5.1 Datasets

We evaluated the model on two representative large size datasets, which have been collected automatically and present a significant amount of table-text divergences for training. Both datasets involve content selection and surface realization, and represent a relatively realistic setting.

WikiBio (Lebret et al., 2016) contains 728,321 tables, automatically paired with the first sentence of the corresponding Wikipedia English article. Reference text's average length is 26 words, and tables have on average 12 key-value pairs. We use the original data partition: 80% for the train set, and 10% for validation and test sets. This dataset has been automatically built from the Internet; concerning divergences, 62% of the references mention extra information not grounded by the table (Dhingra et al., 2019).

ToTTo (Parikh et al., 2020) contains 120,761 training examples, and 7,700 validation and test examples. For a given Wikipedia page, an example is built up by pairing its summary table and a candidate sentence, selected across the whole page via simple similarity heuristics. Such a sentence may accordingly realize whichever table cells, making content selection arbitrary; furthermore, its lexical form may strongly depend on the original context, because of pronouns or anaphoras. Divergences are of course present as well. Those issues have been addressed by Parikh et al. (2020) by (1) *highlighting* the input cells realized by the output, and (2) removing divergences and making the sentence self-contained (e.g. replacing pronouns with their invoked noun or noun

phrase). Fig. 5.6 exemplifies the difference between noisy and clean ToTTo sentences. In our experiments, we limit the input to the highlighted cells and use the original, noisy sentence as output. Noisy texts' average length is 17.4 words, and 3.55 table cells are highlighted, on average.

5.5.2 Baselines

We assess the accuracy and relevance of our alignment labels against the ones proposed by Perez-Beltrachini and Lapata (2018), which is, to the best of our knowledge, the only work proposing such a fine-grained alignment labeling.

To evaluate our Multi-Branch Decoder (*MBD*), we consider five baselines:

- *stnd* (See et al., 2017), a LSTM-based encoder-decoder model with attention and copy mechanisms. This is the standard sequence-to-sequence recurrent architecture.
- *stnd_filtered*, the previous model trained on a filtered version of the training set: tokens deemed hallucinated according to their hallucination scores, are removed from target sentences.
- *hsmm* (Wiseman et al., 2018), an encoder-decoder model with a multi-branch decoder. The branches are not constrained by explicit control factors, but they are rather a neural transposition of the HSMM theoretical model (Yu, 2010). This model is used as a baseline to show that the multi-branch architecture by itself does not guarantee the absence of hallucinations.
- *hier* (Liu et al., 2019a), a hierarchical sequence-to-sequence model, with a coarse-to-fine attention mechanism to better fit the *attribute-value* structure of the tables. This model is trained with three auxiliary tasks to capture more accurate semantic representations of the tables: auxiliary sequence labeling, text auto-encoder and multi-label classification.
- hal_{WO} (Filippova, 2020), a *stnd*-like model trained by augmenting each source table with an additional attribute (*hallucination ratio*, *value*).

We ran our own implementations of *stnd*, *stnd_filtered* and hal_{WO}. Authors of *hier* and *hsmm* models kindly provided us their WikiBio's test set outputs. The metrics described in Sec. 5.5.4 were directly applied on them.

5.5.3 Implementation Details

During training of our multi-branch decoder the fluency branch is always active ($\omega_t^2 = 0.5$) while the content and hallucination branches are alternatively activated, depending on the alignment label l_t: $\omega_t^0 = 0.5$ (content factor) and $\omega_t^1 = 0$ (hallucination factor) when $l_t = 1$, and conversely. The threshold τ used to obtain l_t is set to 0.4 using human tuning to optimize for highest accuracy[3]. All hyperparameters were tuned in order to optimize the validation PARENT F-measure (Dhingra et al., 2019). In particular, we use the [0.4 0.1 0.5]

[3]Note that accuracy is not heavily impacted by different choices of τ. We report in Appendix A.2 the respective accuracy scores of our proposed automated labels for different values of τ.

weight combination during inference. See Sec. 5.6.2 for a discussion about weight combinations and Appendix A.2 for other implementation details.[4]

5.5.4 Metrics

To evaluate our model, we carried out (1) an automatic analysis and (2) a human evaluation for a qualitative analysis of generated sentences.

For the automatic analysis, we use five metrics:

- BLEU (Papineni et al., 2002), introduced in Section 4.3.2. Despite being the standard choice, recent findings show that it correlates poorly with human evaluation, especially on the sentence level (Novikova et al., 2017a; Reiter, 2018), and that it is a proxy for sentence grammar and fluency aspects rather than semantics (Dhingra et al., 2019).
- PARENT (Dhingra et al., 2019) computes smoothed *n*-gram precision and recall over both the reference and the input table. It is explicitly designed for DTT tasks, and its F-measure shows "the highest correlation with humans across a range of settings with divergent references in WikiBio." (Dhingra et al., 2019)
- The *hallucination rate* computes the percentage of tokens labeled as hallucinations (Sec. 5.3).
- The average generated sentence length in number of words.
- The classic readability Flesch index (Flesch, 1962), which is based on words per sentence and syllables per word, and is still used as a standard benchmark (Kosmajac & Keselj, 2019; Smeuninx et al., 2020; Stajner & Hulpus, 2020; Stajner et al., 2020).

Finally, we perform qualitative evaluations of the results obtained on WikiBIO and ToTTo, following the best practices outlined by van der Lee et al. (2019) for intrinsic evaluation, including multiple annotators, well-defined ranking criteria, Likert-scaled or continuous ranking, report of Inter-Annotator Agreement, random ordering of instances. We selected $\sim$20 human annotators from several countries across Europe, between 20 and 55 years old and proficient in English. They have been assigned two different tasks: (i) hallucination labeling, i.e. the selection of sentence pieces which include incorrect information, and (ii) sentence analysis, i.e. evaluating different realizations of the same table according to their fluency, factualness and coverage. Scores are presented as a 3-level Likert scale for Fluency (*Fluent*, *Mostly fluent*, or *Not fluent*) and Factualness (likewise), while coverage is the number of cells from the table that have been realized in the description. To avoid all bias, annotators are shown a randomly selected table at a time, together with its corresponding descriptions, both from the dataset and the models that are being evaluated. Sentences are presented each time in a different order. Following Tian et al. (2019), we first tasked three expert annotators to annotate a pilot batch of 50 sentences. Once confirmed that Inter-Annotator Agreement was approx. 75% (a similar finding to Tian et al. (2019)), we asked 16 annotators to annotate a

[4]Code is given to reviewers and will be available upon acceptance.

Labels	Accuracy	Precision	Recall	F-measure
PB&L	46.9%	21.3%	49.2%	29.7%
ours	**87.5%**	**80.6%**	**59.8%**	**68.7%**

Labels	BLEU	PARENT		
		Precision	Recall	F-measure
PB&L	32.15%	76.91%	39.28%	48.75%
ours	**40.51%**	**77.71%**	**45.01%**	**54.57%**

TABLE 5.1: Performances of hallucination scores on the WikiBio test set, w.r.t. human-designated labels (upper table) and *MBD* trained with different labeling procedures (lower table). Our model always significantly overpasses *PB&L* (T-test with $p < 0.005$).

bigger sample of 300 instances (where each instance consists of one table and four associated outputs), as Liu et al. (2019a).[5]

5.6 Results

We perform an extensive evaluation of our scoring procedure and multi-branch architecture on the WikiBio dataset: we evaluate - the quality of the proposed alignment labels, both intrinsically using human judgment and extrinsically by means of the DTT downstream task and - the performance of our model with respect to the baselines. Additionally, we assess the applicability of our framework on the more noisy ToTTo benchmark, which represents a harder challenge for today's DTT models.

5.6.1 Validation of Alignment Labels.

To assess the effectiveness of our alignment labels (Sec. 5.3), we first compare the alignment labels against human judgment, and then explore their impact on a DTT task. As a baseline for comparison we report performances of *PB&L*.

Intrinsic performance. Tab. 5.1 (top) compares the labeling performance of our method and *PB&L* against human judgment. Our scoring procedure significantly improves over *PB&L*: the latter only achieves 46.9% accuracy and 29.7% F-measure, against 87.5% and 68.7% respectively for our proposed procedure. Perez-Beltrachini and Lapata (2018) report a F-measure of 36%, a discrepancy that can be explained by the difference between the evaluation procedures: *PB&L* evaluate on 132 sentences, several of which can be tied to the same table, whereas we explicitly chose to evaluate on 300 sentences all from different tables in order to minimize correlation.

We remark that beyond F-measure, the precision of *PB&L*'s scoring procedure is at 21.3% compared to 80.6% for ours, and recall stands at 49.2% against 59.8%. We argue that selecting a negative instance at random for training their classifier leads the network to incoherently label words, without apparent justification. See Figure 5.4 for two examples of this phenomenon; and

[5] An eyesight of our platform is available in Appendix A.3.

KEY	VALUE
name	patricia flores fuentes
birth_date	25 july 1977
birth_place	state of mexico , mexico
occupation	politician
nationality	mexican
article_title	patricia flores fuentes

Ref.: patricia flores fuentes -lrb- born 25 july 1977 -rrb- is a mexican politician affiliated to the national action party .
PB&L: patricia flores fuentes -lrb- born 25 july 1977 -rrb- is a mexican politician affiliated to the national action party .
Ours: patricia flores fuentes -lrb- born 25 july 1977 -rrb- is a mexican politician affiliated to the national action party .

(A)

KEY	VALUE
name	ryan moore
spouse	nichole olson -lrb- m. 2011 -rrb-
children	tucker
college	unlv
yearpro	2005
tour	pga tour
prowins	4
pgawins	4
masters	t12 2015
usopen	t10 2009
open	t10 2009
pga	t9 2006
article_title	ryan moore -lrb- golfer -rrb-

Ref.: ryan david moore -lrb- born december 5 , 1982 -rrb- is an american professional golfer , currently playing on the pga tour .
PB&L: ryan david moore -lrb- born december 5 , 1982 -rrb- is an american professional golfer , currently playing on the pga tour .
Ours: ryan david moore -lrb- born december 5 , 1982 -rrb- is an american professional golfer , currently playing on the pga tour .

(B)

FIGURE 5.4: WikiBio instances' hallucinated words according either to our scoring procedure or to the method proposed by Perez-Beltrachini and Lapata (2018). *PB&L* labels word incoherently (a), and sometimes the whole reference text (b). In comparison, our approach leads to a fluent breakdown of the sentences in hallucinated/factual statements.

Model	BLEU↑	PARENT↑			Halluc. rate↓	Mean sent. length	Flesch↓
		Precision	Recall	F-measure			
Gold	-	-	-	-	23.82%	19.20	**53.80%**
`stnd`	41.77%	79.75%	45.02%	55.28%	4.20%	13.80	58.90%
`stnd_filtered`	34.66%	**80.90%**	42.48%	53.27%	**0.74%**	12.00	62.10%
`hsmm`	35.17%	71.72%	39.84%	48.32%	7.98%	14.80	58.60%
`hier`	**45.14%**	75.09%	46.02%	54.65%	10.10%	16.80	56.20%
$\texttt{hal}_{\texttt{WO}}$	36.50%	79.50%	40.50%	51.70%	-	-	-
`MBD`	41.56%	79.00%	**46.40%**	**56.16%**	1.43%	14.60	58.80%

TABLE 5.2: Comparison results on WikiBio. ↑ (resp. ↓) means higher (resp. lower) is better. "Gold" refers to the gold standard, i.e. the reference texts included in the dataset.

Appendix A.4 for other comparisons. In contrast, our method is able to detect hallucinated statements inside a sentence, without incorrectly labeling the whole sentence as hallucinated.

Impact on a DTT downstream task. Additionally, we assess the difference of both scoring procedures using their impact on the WikiBio DTT task. Specifically, Tab. 5.1 (bottom) shows the results of training *MBD* using either *PB&L*'s or our labels. We observe significant improvements, especially in BLEU and PARENT-recall (40.5% vs 32.2% and 45% vs 39.3%), showing that our labeling procedure is more helpful at retaining information from training instances (the system better picks up what humans picked-up, ultimately resulting in better BLEU and recall).

5.6.2 Automatic System Evaluation

Comparison with SOTA systems. Tab. 5.2 shows the performances of our model and all baselines according to the metrics of Sec. 5.5.4. Two qualitative examples are presented in Figure 5.5 and more are available in Appendix A.4.

First of all, reducing hallucinations is reached with success, as highlighted by the hallucination rate (1.43% vs. 4.20% for a standard encoder-decoder and 10.10% for the best SOTA model on BLEU). The only model which gets a lower hallucination rate (0.74%, corroborated by its PARENT-precision of 80.9%), *stnd_filtered*, achieves such a result at a high cost. As can be seen in Figure 5.5 where its output is factual but cut short, its sentences are the shortest and the most naive in terms of the Flesch readability index, which is also reflected by a lower BLEU score. The high PARENT precision – mostly due to the shortness of the outputs – is counterbalanced by a low recall: the F-measure indicates the overall lack of competitiveness of this trade-off. This shows that the naive approach of simply filtering training instances is not the appropriate solution for hallucination reduction. This echoes (Filippova, 2020) who trained a vanilla network on the cleanest 20% of the data and found that predictions are more precise than those of a model trained on 100% but that PARENT-recall and BLEU scores are low.

At the other extreme, the best model in terms of BLEU, *hier*, falls short regarding precision, suggesting that often the generated text is not matched in the input table; this issue is also reflected by the highest hallucination rate of all models (10.10%). A reason could be the introduction of their auxiliary training tasks which often drive the decoder to excess in mimicking human behavior.

Weights			BLEU	PARENT		
ω^0	ω^1	ω^2		Precision	Recall	F-measure
0.5	0.0	0.5	38.90%	80.37%	44.96%	55.29%
0.4	0.1	0.5	41.56%	79.00%	46.40%	56.16%
0.3	0.2	0.5	42.68%	72.99%	45.81%	53.74%
0.2	0.3	0.5	22.64%	53.92%	32.96%	36.55%
0.1	0.4	0.5	2.03%	57.88%	4.82%	6.79%
0.0	0.5	0.5	0.32%	85.01%	1.02%	1.78%
0.0	0.4	0.6	1.07%	62.71%	2.47%	3.66%
0.0	0.3	0.7	2.81%	42.86%	6.15%	7.94%
0.0	0.2	0.8	7.30%	41.78%	16.58%	18.68%
0.0	0.1	0.9	15.51%	56.93%	32.85%	36.88%

TABLE 5.3: Performances of *MBD* on WikiBio validation set, with various weight settings. Weights' order is (ω^0 – *content*, ω^1 – *hallucination*, ω^2 – *fluency*).

While BLEU score improves, overall factualness of outputs decreases, showing that the model picks up domain lingo (how to formulate ideas) but not domain insight (which ideas to formulate) (see Figure 5.5). This is in line with (Filippova, 2020; Reiter, 2018) who argue that BLEU is an inappropriate metric for generation tasks other than machine translation.

The analysis of *hsmm*, and especially of its relatively weak performance both in terms of BLEU and PARENT, highlights the insufficiency of the multi-branch architecture by itself. This reinforces the need of the additional hallucinations supervision provided by our labeling procedure.

Finally, in the comparisons with hal_{WO}, we can see that while it achieves one of the highest performances in term of precision (79.5%), this comes at the cost of the lowest recall (40.5%) of all models and thus poor F-measure. This confirms our hypothesis that, while effective at producing mostly factual content, modeling hallucination only as a fixed value for a whole instance is detrimental to the content generation procedure. Finer-grain annotations are required, as shown by our model recall (46.4%), coupled with a robust precision (79.0%).

Weight impact on decoding. As we deal with a CTG system, we can guide our network at inference to generate sentences following desired attributes. The impact of different weight combinations is explored in Tab. 5.3. In particular, we can see that changing weights in favor of the hallucination factor (top five lines) leads to decreases in both precision and recall (from 80.37% to 57.88% and 44.96% 4.82% respectively). We also observe that strongly relying on the hallucinating branch dramatically impacts performances ([0.0 0.5 0.5] obtains near 0 BLEU and F-measure), as it is never fed with complete, coherent sentences during training. However, some performance can still be restored via the fluency branch: [0.0 0.1 0.9] performs at 15.51% BLEU and 36.88% F-measure.

It is interesting to note that the relaxation of the strict constraint on the content factor in favor of the hallucination factor, ([0.4 0.1 0.5] $\rightarrow$ [0.5 0.0 0.5]) obtains better performances (56.16% vs 55.29% F-measure). This highlights that strictly constraining on content yields sensibly more factual outputs (79% vs

name	zack lee
birth_name	zack lee jowono
nationality	indonesian
occupation	actor , boxer , model
birth_date	15 august 1984
birth_place	liverpool , merseyside , england , uk
years_active	2003 - present
parents	hendra and ayu jowono
spouse	nafa urbach (2007 - present)
article_title	zack lee

Gold	zack lee (born 15 august 1984) is an indonesian actor , model and boxer of british descent .
stnd	zack lee jowono (born 15 august 1984) is an indonesian actor and model .
stnd_filtered	zack lee (born zack lee jowono ; 15 august 1984) is an indonesian actor .
hsmm	zack lee jowono (born 15 august 1984) is an indonesian actor who has appeared in tamil films .
hier	zack lee jowono (born 15 august 1984) , better known by his stage name zack lee , is an indonesian actor , model and model .
MBD[.4, .1, .5]	zack lee (born zack lee jowono ; 15 august 1984) is an indonesian actor , boxer and model .

(A)

name	wayne r. dynes
birth_date	23 august 1934
occupation	professor , historian , and encyclopedist
article_title	wayne r. dynes

Gold	wayne r. dynes (born august 23 , 1934) is an american art historian , encyclopedist , and bibliographer .
stnd	wayne r. dynes (born august 23 , 1934) is an american historian and encyclopedist .
stnd_filtered	wayne r. dynes is a professor .
hsmm	wayne r. dynes (born august 23 , 1934) is an american historian , historian and encyclopedist .
hier	wayne r. dynes (born august 23 , 1934) is an american professor of history at the university of texas at austin .
MBD[.4, .1, .5]	wayne r. dynes (born august 23 , 1934) is an american professor , historian , and encyclopedist .

(B)

FIGURE 5.5: Qualitative examples of our model and baselines on the WikiBio test set. Note that: (1) *gold* references may contain divergences; (2) *stnd* and *hsmm* seem to perform well superficially, but often hallucinate; (3) *stnd_filtered* doesn't hallucinate but struggles with fluency; (4) *hier* overgenerate "human-sounding" statements, that lacks facutalness; (5) *MBD* sticks to the fact contained by the table, in concise and fluent sentences.

Model	Fluency	Factualness	Coverage
Gold	98.7%	32.0%	**4.47**
`stnd_filtered`	93.5%	**86.1%**	4.07
`hier`	97.4%	55.0%	4.45
`MBD`	**99.6%**	76.6%	4.46

TABLE 5.4: Results of the human evaluation on WikiBio[6].

Model	BLEU$^{\uparrow}$	PARENT$^{\uparrow}$			Human evaluation		
		Precision	Recall	F-measure	Fluency$^{\uparrow}$	Factualness$^{\uparrow}$	Coverage
Gold(noisy)	-	-	-	-	97.1% (97.1)	91.2% (79.4)	3.618
`stnd`	**21.27%**	56.60%	25.16%	**29.71%**	55.9% (26.5)	53.0% (20.6)	2.824
`stnd_filtered`	19.48%	56.69%	22.31%	27.18%	29.4% (8.8)	70.6% (50.0)	2.706
`hal`$_{\text{WO}}$	17.06%	**77.64%**	22.65%	29.38%	61.7% (38.2)	61.8% (32.4)	2.725
`MBD`	18.35%	50.44%	**25.25%**	28.25%	**91.2% (50.0)**	**85.3% (55.9)**	**3.613**

TABLE 5.5: Comparison results on ToTTo. $^{\uparrow}$ (resp. $^{\downarrow}$) means higher (resp. lower) is better. In human evaluation for Fluency, reported are for "Fluent" and "Mostly Fluent", with only "Fluent" in parentheses. Same for Factualness.

80.37% precision), at the cost of constraining the model's generation creativity (46.40% vs 44.96% recall). The [0.4 0.1 0.5] variant has more "freedom of speech" and sticks more faithfully to domain lingo (recall and BLEU), without compromising too much in terms of content.

5.6.3 Human evaluation

To measure subtleties which are not captured by automatic metrics, we report in Tab. 5.4 human ratings of our model, two baselines and the gold. These baselines have been selected because they showcase interesting behaviors on automatic metrics: *hier* obtains the best BLEU score but a poor precision, and *stnd_filtered* gets the best precision but poor BLEU, length and Flesch index.

First, coherently with (Dhingra et al., 2019), we found that around two thirds of gold references contain divergences from their associated tables. Such data also confirm our analysis on the *stnd_filtered* baseline: it's training on truncated sentences lead to an unquestionable ability to avoid hallucinations, while dramatically impacting both its fluency and coverage, leading to less desired outputs overall, despite the high PARENT-precision score.

The comparison between *hier* and *MBD* shows that both approaches lead to similar coverage, with *MBD* obtaining significantly better performances in terms of factualness. We also highlight that *MBD* is evaluated as being the most fluent one, even better than the reference (which can be explained by the imperfect pre-processing done by Lebret et al. (2016)).

`page_title`	`Huge (TV series)`
`section_title`	`Episodes`
`Original_air_date`	`June 28 2010`
`U.S._viewers_(millions)`	`2.53`

Gold (clean)	The TV series , Huge , premiered on June 28 , 2010 with 2.53 million viewers.
Gold (noisy)	The series premiered on June 28 , 2010 at 9 p.m. with 2.53 million viewers .
`stnd`	On June 28 , 2010 , it was watched by 2.53 million viewers .
`stnd_filtered`	was watched by 2.53 on June 28 , 2010 .
$\texttt{hal}_{\texttt{WO}}$	June 28 , 2010 : Huge million viewers .
`MBD[.4, .1, .5]`	Huge 's first episode , aired on June 28 , 2010 , was watched by 2.53 million .

(A)

`page_title`	`LM317`
`section_title`	`Specification`
`Parameter`	`Output voltage range`
`Value`	`1.25 - 37`

Gold (clean)	LM317 produces a voltage of 1.25 V .
Gold (noisy)	Internally the device has a bandgap voltage reference which produces a stable reference voltage of Vref= 1.25 V followed by a feedback-stabilized amplifier with a relatively high output current capacity .
`stnd`	The Output is a Output range of 1.25 – 37 .
`stnd_filtered`	range from 1.25 to 37 .
$\texttt{hal}_{\texttt{WO}}$	Output voltage range 1.25 – 37 – 37 .
`MBD[.4, .1, .5]`	The Output 's range is approximately 1.25 .

(B)

FIGURE 5.6: Qualitative examples of *MBD* and hal_{WO} on ToTTo. hal_{WO}'s poor generation quality is not detected by discrete metrics. In contrast, *MBD* generates fluent and naively factual sentences. Note that *stnd* and *stnd_filtered* have the same behavior as on WikiBio: the former produces fluent but nonsensical text; the latter generates very un-fluent, but factual, text.

5.6.4 ToTTo: a considerably noisy setting

The ToTTo dataset is used in the following experiments to explore models' robustness to the impact of extreme noise during training. As stated in Section 5.5.1, we use as inputs only the *highlighted* cells, as content selection is arbitrary (i.e. the cells were chosen depending on the target sentence, and not vice versa). On the other hand, we use as targets the noisy references, which may contain both divergences and lexical issues. This setting is particularly challenging and is more effective in recreating a representational, hallucination-prone real-life context than WikiBio. Other datasets (Gardent et al., 2017a; Novikova et al., 2017c; Wen et al., 2015a) available in literature are too similar to WikiBio concerning their goals and challenges, and are therefore less interesting in this context.

Table 5.5 reports the performances of *stnd*, *stnd_filtered*, hal_{WO} and *MBD* with regards to automatic metrics and human evaluation. Compared to their respective performances on WikiBio, all models show significantly decreased scores. They struggle at generating syntactically correct sentences but, at the same time, they have still learned to leverage their copy mechanism and to stick to the input. This behavior is illustrated in both examples of Fig. 5.6. In particular, hal_{WO}'s high PARENT-precision score (77.64%) seems to be due to its tendency to blindly copy input data without framing them in a sentence structure, as its low BLEU and PARENT-recall scores suggests (17.06% and 22.65%). These lower scores are good indicators that the ToTTo task, as framed in this Chapter, is difficult. Following the same evaluation protocol than for WikiBio, we report human ratings of different models, also included in Table 5.5.

MBD's factualness is judged favorably, with 55.9% hallucination-free texts, and up to 85.3% texts with a single error at most. In contrast, hal_{WO} stands at 32.4% and 61.8% for error-free texts and single-error texts respectively. Interestingly, *stnd_filtered* obtains the second best performance (70.6% texts with a single error).

Fluency scores are also meaningful: hal_{WO} and *MBD* respectively obtain 61.7% and 91.2%. Word-based filtering is not suitable for noisy datasets, as shown by *stnd_filtered*'s worse fluency score, 29.4%.

As for coverage performances, our model *MBD* obtains the maximum coverage score 3.613, surpassing all baselines by at least 0.789 slots (the second best coverage score is obtained by *stnd* at 2.824), and getting very close to the Gold value (which stands at 3.618). These performances, and qualitative examples of Figure 5.6, suggest that *stnd_filtered* and hal_{WO} try to reduce hallucinations at the cost of missing some input slot, while *MBD* effectively balances both goals.

The analysis of Factualness, Fluency and Coverage can be enhanced using qualitative error analysis on randomly sampled generated texts (we report two such examples in Figure 5.6). In particular, we want to highlight the following considerations:

[6]Fluency reports the sum of "fluent" and "mostly fluent", as "mostly fluent" often comes from misplaced punctuation and doesn't really impact readability. However, Factualness reports only the count of "factual", as "mostly factual" sentences contain hallucinations and cannot be considered "factual".

- As most training examples are very noisy, sentence-level models fail at learning from them. *stnd_filtered* has been trained on factual statements only, at the cost of using mostly incomplete sentences during training. On both examples of Figure 5.6, it generated truncated sentences, missing their subjects. Its relatively high Factualness and low Fluency scores indicate that it did not learn to produce diverging outputs, nor complete sentences. Differently, hal_{WO} generates incorrectly ordered sequences of words extracted from the table (Fig. 5.6a), or repetitions (Fig. 5.6b). The low number of training instances containing the input pair *(hallucination ratio, 0)* does not allow to learn what a non-hallucinated sentence actually consists in.
- In contrast, our proposed finer-grained approach proves helpful in this setting, as shown by the human evaluation: sentences generated by *MBD* are more fluent and more factual. The multi-branch design enables the model to leverage the most of each training instance, leading to better performances overall.
- Finally, we acknowledge that despite over-performing other models, *MBD* obtains only 55.9% of *factual* sentences. For instance, in Figure 5.6b, our model does not understand that a range consists of two numbers. The difficulty of current models to learn on very noisy and diverse datasets shows that there is still room for improvement in hallucination reduction in DTT.

5.7 Limitations and future work

We designed our alignment procedure to be general and easily reproducible on any DTG dataset. One strength of our approach is that co-occurrences and dependency parsing can be used intuitively to extract more information from the tables than a naive word matching procedure. However, in the context of tables mainly including numbers (e.g., RotoWire), the effectiveness of the co-occurrence analysis is not guaranteed. A future work will be to improve upon the co-occurrence analysis to generalize to tables which contain less semantic inputs. For instance, the labeling procedure of Perez-Beltrachini and Lapata (2018) might be revised so that adverse instances are not selected randomly, which we hypothesize would result in more relevant labels.

Finally, experiments on ToTTo outline the narrow exposure to language of current models when used on very noisy datasets. Our model has shown interesting properties through the human evaluation, but it still shows itself to be perfectible. Recently introduced large pre-trained Language Models, which have seen significantly more varied texts, may attenuate this problem. In this direction, adapting the work of (Z. Chen et al., 2020; Kale & Rastogi, 2020) to our model could bring improvements to the results presented in this Chapter.

Chapter 6

Conclusions and future work

This thesis focuses on the Data-To-Text generation task, characterized as the development of systems that generate meaningful text from structured table records. Most recent research has moved away from rule-based models, acknowledging the better performance of end-to-end and data-driven methods, in particular following the outbreak of deep learning. New challenges arise from this shift, that are being faced by current research. Wiseman et al. (2017) outlines some of them, such as the difficulty in performing content selection, in keeping inter-sentence coherence, in avoiding redundance, and in being faithful to the input. This thesis elaborates on this last issue.

In particular, we inquire on the properties a neural Data-To-Text Generation system should own to ensure faithfulness of the generated outputs. The difficulties deep learning models encounter on this matter arise from the fact that they are data-driven to the core. The quality of the training data reflects itself on the quality of the deployed model. Nevertheless, more modern deep learning systems are composed of a huge number of trainable parameters, which results in the need for huge amounts of training data (LeCun et al., 2015). Reasonably sized DTT corpora can only be constructed from internet sources, resulting in roughly aligned source-target pairs (Dhingra et al., 2019). Non-perfect alignment degrades the models' performance. Manual data cleaning is prohibitive for such amount of data: the development of faithful-ensuring systems is therefore crucial (Filippova, 2020).

This work addresses the issue, arguing that faithful models for DTT must include two main features. First, the ability to copy input content to the output should be able to establish strong links between the best-aligning parts of a table and a corresponding utterance. Second, the ability to avoid the generation of information which is not included in the table weakens the alignment between such table and the divergent part of the reference. Meanwhile, the faculty of producing content that can be inferred from the table should not be affected. Both desired features are taken into consideration: Chapter 4 introduces a copy-enabled system working on characters, while Chapter 5 proposes a framework for reducing hallucinations, composed of a labelling procedure and a Multi-Branch Decoder. Experimental results validate the effectiveness of the proposed frameworks, pushing the DTT field towards designing more reliable and robust Natural Language Generation systems.

A qualitative analysis of the outputs generated by faithful-oriented systems (Appendix A) draws attention to a tradeoff between sticking to the input facts

and allowing for more complex (and risky) inferences. Once obtained sufficiently precise models, their constraints should be relaxed to allow the inclusion of some implicit information. In the biography domain, for example, the age can be inferred from the year of birth, while the nationality should not be deduced from the first name. Specialized submodules for logic reasoning (Shi et al., 2020; P.-W. Wang et al., 2019) or mathematical deduction (J. Lee et al., 2019; Schlör et al., 2020) may be integrated in neural DTT architectures. Besides, handling numeric values can be included via a generalization of the copy mechanism. As seen in Chapter 4, the binary soft switch determines the current reasoning mode between generation and copy. An n-ary switch, in contrast, would allow additional modes such as the numeric one. Another research direction involves the development of neural submodules explicitly tailored for macro and macro-planning, following the traditional pipeline architecture described in Section 3.1 (Puduppully et al., 2019a). Such components can naturally account for numeric and logic operations in the form of actions.

As the proposed systems integrate a computational and memory overhead, either in the training phase (such as the RNN switch described in Section 4.2.2) or at inference (as in the Multi-Branch Decoder structure detailed by Section 5.4), future work should reduce such added costs. The introduction of the Transformer (Vaswani et al., 2017) and, consequently, of pre-trained Language Models (Devlin et al., 2019; Raffel et al., 2020), opens new possibilities in this direction. The former's architecture allows for more aggressive parallelization strategies, resulting in faster optimization steps. The latter reduces the number of steps required to obtain a well-working model, as the initial weights configuration already contains a general knowledge of natural language's structure and semantics. Recent developments in attention-based architectures, outlined in Sections 4.4 and 5.7, can be merged with the techniques proposed and analyzed in this thesis, tracing a good path for bridging the gap between research and production systems.

Appendix A

Controlling Hallucinations at Word Level

A.1 Alignment labels reproducibility

We consider as *important words*, i.e. nouns, adjectives or numbers, those which are Part-of-Speech tagged as `NUM`, `ADJ`, `NOUN` and `PROPN`.

In order to apply the score normalization function $norm(\cdot, y)$, we separate sentences y into statements $y_{t_i:t_{i+1}-1}$. To do so, we identify the set of introductory dependency relation labels[1], following previous work on rule-based systems for the conversion of dependency relations trees to constituency trees (Borensztajn et al., 2009; Han et al., 2000; Hwa et al., 2005; Xia & Palmer, 2001). Our segmentation algorithm considers every leaf token in the dependency tree, and seeks its nearest ancestor which is the root of a statement.

Two heuristics enforce the score normalization: (i) conjunctions and commas next to hallucinated tokens acquires these lasts' hallucination scores, and (ii) paired parentheses and quotes acquire the minimum inner tokens' hallucination score.

Part-of-Speech tagging has been done using the HuggingFace's Transformers library (Wolf et al., 2019) to fine-tune a BERT model (Devlin et al., 2019) on the UD English ParTUT dataset (Sanguinetti & Bosco, 2015); Stanza (Qi et al., 2020) has been exploited for dependency parsing.

A.2 Implementation details

Our system is implemented in Python 3.8[2] and PyTorch 1.4.0[3]. In particular, our multi-branch architecture is developed, trained and tested as an OpenNMT (Klein et al., 2017) model. Sentence lengths and Flesch index (Flesch, 1962) are computed using the standard `style` Unix command.

Differently to Perez-Beltrachini and Lapata (2018), we did not adapt the original WikiBio dataset[4] in any manner: as we work on the model side, we fairly preserve the dataset's noisiness.

[1] `acl`, `advcl`, `amod`, `appos`, `ccomp`, `conj`, `csubj`, `iobj`, `list`, `nmod`, `nsubj`, `obj`, `orphan`, `parataxis`, `reparandum`, `vocative`, `xcomp`; every dependency relation is documented in the Universal Dependencies website.

[2] http://www.python.org

[3] http://www.pytorch.org

[4] https://github.com/DavidGrangier/wikipedia-biography-dataset

Threshold	Accuracy	F-measure	Precision	Recall
0.0	70.2%	56.8%	42.2%	86.7%
0.4	**86.0%**	**70.6%**	67.0%	74.6%
0.8	85.8%	62.8%	77.3%	52.9%

TABLE A.1.1: Accuracy scores of our proposed word-level automated labels for different values of the threshold τ.

Model	BLEU	PARENT		
		Precision	Recall	F-measure
`MBD[.4, .1, .5]`	42.50%	79.26%	46.09%	55.95%

TABLE A.1.2: The performances of our model on the WikiBio validation set.

Word-level alignment labels are computed setting $m = 5$, following Mikolov et al. (2013). As stated in Sec. 5.5.3, the threshold τ's value is optimized for highest accuracy via human tuning: Table A.1.1 shows accuracy scores of our proposed automated labels for different values of τ.

We share the vocabulary between input and output, limiting its size to 20000 tokens. Hyperparameters were tuned using performances on the development set: Tab. A.1.2 reports the performances of our best performing *MBD* on the development set. Our encoder consist of a 600-dimensional embedding layer followed by a 2-layered bidirectional LSTM network with hidden states sized 600. We use the *general* attention mechanism with input feeding (T. Luong et al., 2015) and the same copy mechanism as See et al. (2017). Each branch of the multi-branch decoder is a 2-layered LSTM network with hidden states sized 600 as well.

Training is performed using the Adam algorithm (Kingma & Ba, 2015b) with learning rate $\eta = 10^{-3}$, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate is decayed with a factor of 0.5 every 10000 steps, starting from the 5000th one. We used minibatches of size 64 and regularized via clipping the gradient norm to 5 and using a dropout rate of 0.3. We used beam search during inference, with a beam size of 10.

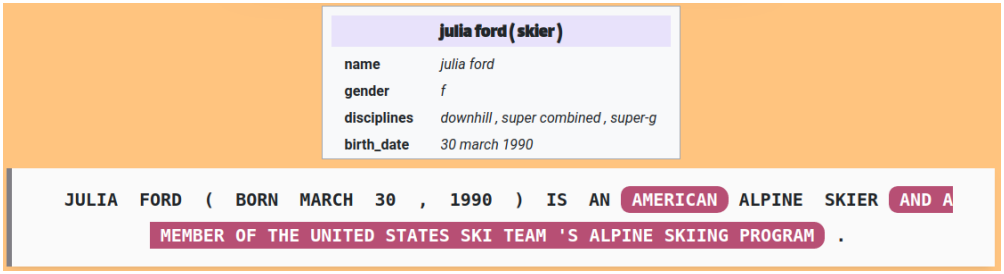
All experiments were performed on a single NVIDIA Titan XP GPU. Number of parameters and training times are shown in Table A.1.3. Same model's differences between WikiBio and ToTTo are justified by the different datasets' number of instances and input vocabulary sizes.

A.3 Annotation interface

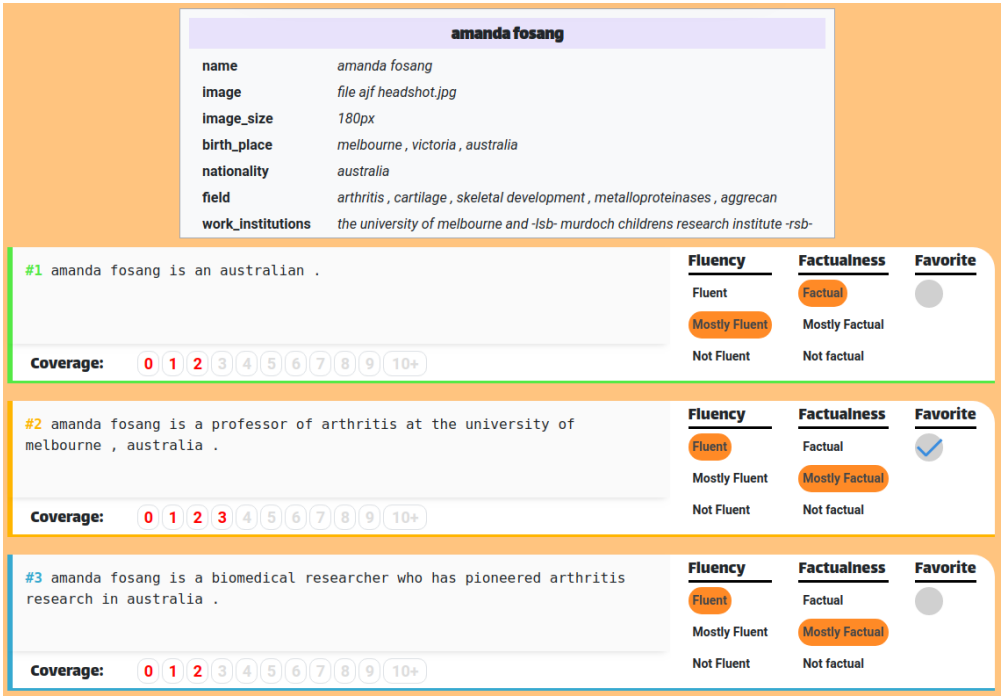
The human annotation procedure is done via a web application specifically developed for this research. Fig. A.2.1a shows how the hallucination tagging user interface looked like in practice, while in Fig. A.2.1b a typical sentence analysis page is shown.

Dataset	Model	Size [M]	Training time [h]
WikiBio	`stnd`	41	5
	`stnd_filtered`	41	5
	`hal`$_{\texttt{WO}}$	41	5
	`MBD`	55	10
ToTTo	`stnd`	62	4
	`stnd_filtered`	62	4
	`hal`$_{\texttt{WO}}$	62	4
	`MBD`	76	8

TABLE A.1.3: Sizes and training times of the implemented models.



(A) Hallucination tagging



(B) Sentence analysis

FIGURE A.2.1: The human annotation tasks, as presented to the annotators.

A.4 Qualitative examples

Tables A.3.1 to A.3.5 show word-level labeling of WikiBio training examples. Underlined, red words are hallucinated according either to our scoring procedure or to the method proposed by Perez-Beltrachini and Lapata (2018).

In the subsequent tables, some WikiBio (A.3.6 to A.3.15) and ToTTo (A.3.16 to A.3.18) inputs are shown, coupled with the corresponding sentences, either as found in the dataset, or as generated by our models and baselines.

KEY	VALUE
name	susan blu
birth_name	susan maria blupka
birth_date	12 july 1948
birth_place	st paul , minnesota , u.s.
occupation	actress , director , casting director
yearsactive	1968 - present
article_title	susan blu

Ref.: susan maria blu -lrb- born july 12 , 1948 -rrb- , sometimes credited as sue blu , is an american voice actress , voice director and casting director in american and canadian cinema and television .
PB&L: susan maria blu -lrb- born july 12 , 1948 -rrb- , sometimes credited as sue blu , is an american voice actress , voice director and casting director in american and canadian cinema and television .
Ours: susan maria blu -lrb- born july 12 , 1948 -rrb- , sometimes credited as sue blu , is an american voice actress , voice director and casting director in american and canadian cinema and television .

TABLE A.3.1: Hallucinated words according either to our scoring procedure or to the method proposed by Perez-Beltrachini and Lapata (2018).

KEY	VALUE
name	patricia flores fuentes
birth_date	25 july 1977
birth_place	state of mexico , mexico
occupation	politician
nationality	mexican
article_title	patricia flores fuentes

Ref.: patricia flores fuentes -lrb- born 25 july 1977 -rrb- is a mexican politician affiliated to the national action party .
PB&L: patricia flores fuentes -lrb- born 25 july 1977 -rrb- is a mexican politician affiliated to the national action party .
Ours: patricia flores fuentes -lrb- born 25 july 1977 -rrb- is a mexican politician affiliated to the national action party .

TABLE A.3.2: Hallucinated words according either to our scoring procedure or to the method proposed by Perez-Beltrachini and Lapata (2018).

KEY	VALUE
name	ate faber
birth_date	19 march 1894
birth_place	leeuwarden , netherlands
death_date	19 march 1962
death_place	zutphen , netherlands
sport	fencing
article_title	ate faber

Ref.: ate faber -lrb- 19 march 1894 - 19 march 1962 -rrb- was a dutch fencer .
PB&L: ate faber -lrb- 19 march 1894 - 19 march 1962 -rrb- was a dutch fencer .
Ours: ate faber -lrb- 19 march 1894 - 19 march 1962 -rrb- was a dutch fencer .

TABLE A.3.3: Hallucinated words according either to our scoring procedure or to the method proposed by Perez-Beltrachini and Lapata (2018).

KEY	VALUE
name	alex wilmot sitwell
birth_date	16 march 1961
birth_place	uk
occupation	president , europe and emerging markets -lrb- ex-asia -rrb- of bank of america merrill lynch
article_title	alex wilmot-sitwell

Ref.: alex wilmot-sitwell heads bank of america merrill lynch 's businesses across europe and emerging markets excluding asia .
PB&L: alex wilmot-sitwell heads bank of america merrill lynch 's businesses across europe and emerging markets excluding asia .
Ours: alex wilmot-sitwell heads bank of america merrill lynch 's businesses across europe and emerging markets excluding asia .

TABLE A.3.4: Hallucinated words according either to our scoring procedure or to the method proposed by Perez-Beltrachini and Lapata (2018).

KEY	VALUE
name	ryan moore
spouse	nichole olson -lrb- m. 2011 -rrb-
children	tucker
college	unlv
yearpro	2005
tour	pga tour
prowins	4
pgawins	4
masters	t12 2015
usopen	t10 2009
open	t10 2009
pga	t9 2006
article_title	ryan moore -lrb- golfer -rrb-

Ref.: ryan david moore -lrb- born december 5 , 1982 -rrb- is an american professional golfer , currently playing on the pga tour .
PB&L: ryan david moore -lrb- born december 5 , 1982 -rrb- is an american professional golfer , currently playing on the pga tour .
Ours: ryan david moore -lrb- born december 5 , 1982 -rrb- is an american professional golfer , currently playing on the pga tour .

TABLE A.3.5: Hallucinated words according either to our scoring procedure or to the method proposed by Perez-Beltrachini and Lapata (2018).

title	prince of noër
name	prince frederick
image	prinsen af noer.jpg
image_size	200px
spouse	countess henriette of danneskjold-samsøe mary esther lee
issue	prince frederick , count of noer prince christian louise , princess michael vlangali-handjeri princess marie
house	house ofschleswig-holstein-sonderburg-augustenburg
father	frederick christian ii , duke of schleswig-holstein-sonderburg-augustenburg
mother	princess louise auguste of denmark
birth_date	23 august 1800
birth_place	kiel
death_date	2 july 1865
death_place	beirut
article_title	prince frederick of schleswig-holstein-sonderburg-augustenburg
Gold	prince frederick emil august of schleswig-holstein-sonderburg-augustenburg (kiel , 23 august 1800 – beirut , 2 july 1865) , usually simply known by just his first name , frederick , " prince of noër " , was a prince of the house of schleswig-holstein-sonderburg-augustenburg and a cadet-line descendant of the danish royal house .
stnd	prince frederick of schleswig-holstein-sonderburg-augustenburg (23 august 1800 – 2 july 1865) was a member of the house of schleswig-holstein-sonderburg-augustenburg .
stnd_filtered	prince frederick of schleswig-holstein-sonderburg-augustenburg (23 august 1800 – 2 july 1865) was a german .
hsmm	prince frederick of schleswig-holstein-sonderburg-augustenburg (23 august 1800 – 2 july 1865) was a danish noblewoman .
hier	prince frederick of schleswig-holstein-sonderburg-augustenburg ( ) (23 august 1800 – 2 july 1865) was a german prince of the house of schleswig-holstein-sonderburg-augustenburg .
MBD[.4, .1, .5]	prince frederick of schleswig-holstein-sonderburg-augustenburg (; 23 august 1800 – 2 july 1865) was the son of frederick christian ii , duke of schleswig-holstein-sonderburg-augustenburg and princess louise auguste of denmark .

TABLE A.3.6: A WikiBio input table, coupled with the corresponding sentence and the models-generated outputs.

name	godgory
background	group_or_band
origin	karlstad , sweden
genre	melodic death metal doom metal gothic metal
years_active	1992 - 2004
label	nuclear blast
current_members	matte andersson erik andersson
past_members	mikael dahlqvist stefan grundel henrik lindström fredric danielsson thomas heder
article_title	godgory
Gold	godgory was a swedish melodic death metal or death doom metal band that was formed august 1992 by erik andersson (drums) and matte andersson (vocals) of no relation .
stnd	godgory was a melodic death metal band from karlstad , sweden .
stnd_filtered	godgory was a swedish metal band .
hsmm	godgory (10 august 2004 – 4 january 2010) was a melodic death metal band from karlstad , sweden .
hier	godgory was a melodic death metal band from sweden , sweden .
MBD[.4, .1, .5]	godgory was a swedish melodic death metal band from karlstad .

TABLE A.3.7: A WikiBio input table, coupled with the corresponding sentence and the models-generated outputs.

name	christy mihos
image	christy2006wp.jpeg
px|office	massachusetts turnpike commission member
term_start	july 1 , 1999
term_end	july 1 , 2004
successor	daniel grabauskas
birth_date	13 june 1949
birth_place	brockton , massachusetts
nationality	american
occupation	convenience store franchise owner
spouse	andrea mihos (1974-2013 ; divorce)
religion	greek orthodox
alma_mater	stonehill college
party	republican
article_title	christy mihos

Gold	christy peter mihos is an american politician and businessman from massachusetts .
stnd	christy mihos (born june 13 , 1949 in brockton , massachusetts) is a former republican member of the massachusetts house of representatives .
stnd_filtered	christy mihos (born june 13 , 1949 in brockton , massachusetts) is an american politician .
hsmm	christy mihos (born june 13 , 1949) is an american politician from the state of massachusetts .
hier	christy mihos (born june 13 , 1949) is an american politician who served as the massachusetts state representative from 1999 to 2004 .
MBD[.4, .1, .5]	christy mihos (born june 13 , 1949 , in brockton , massachusetts) is a former member of the massachusetts turnpike commission .

TABLE A.3.8: A WikiBio input table, coupled with the corresponding sentence and the models-generated outputs.

name	duncan eagleson
nationality	american
area	painter , writer , penciller , colorist , sculptor , designer
article_title	duncan eagleson

Gold	duncan eagleson is an american self-trained painter and former graffiti artist .
stnd	duncan eagleson is an american painter , writer , and penciller .
stnd_filtered	duncan eagleson is an american painter .
hsmm	duncan eagleson (born 1975) is an american comic book painter and writer .
hier	duncan eagleson is an american painter , illustrator , and designer .
MBD[.4, .1, .5]	duncan eagleson is an american painter , writer , and sculptor .

TABLE A.3.9: A WikiBio input table, coupled with the corresponding sentence and the models-generated outputs.

name	gerald warner brace
imagesize	208px
birth_date	24 september 1901
birth_place	islip , long island , suffolk county , new york
death_date	20 july 1978
death_place	blue hill , maine
occupation	novelist , writer , educator , sailor , boat builder
nationality	american
genre	fiction , non-fiction
article_title	gerald warner brace
Gold	gerald warner brace (september 24 , 1901 – july 20 , 1978) was an american novelist , writer , educator , sailor and boat builder .
stnd	gerald warner brace (september 24 , 1901 – july 20 , 1978) was an american novelist , writer , and boat builder .
stnd_filtered	gerald warner brace (september 24 , 1901 – july 20 , 1978) was an american novelist .
hsmm	gerald warner brace (september 24 , 1901 – july 20 , 1978) was an american novelist and writer .
hier	gerald warner brace (september 24 , 1901 – july 20 , 1978) was an american novelist , short story writer , educator , and sailor .
MBD[.4, .1, .5]	gerald warner brace (september 24 , 1901 – july 20 , 1978) was an american author , educator , sailor , and boat builder .

TABLE A.3.10: A WikiBio input table, coupled with the corresponding sentence and the models-generated outputs.

name	robert b. murrett
image	robertbmurrett.jpg
office	4th director of the national geospatial-intelligence agency director of the office of naval intelligence
president	george w. bush barack obama george w. bush
term_start	2006 2005
term_end	2010 2006
predecessor	james r. clapper richard b. porterfield
successor	letitia long tony l. cothron
alma_mater	university at buffalo georgetown university joint military intelligence college
branch	united states navy
rank	vice admiral 20px
article_title	robert b. murrett

Gold	vice admiral robert b. murrett was the fourth director of the national geospatial-intelligence agency , from 7 july 2006 through july 2010 .
stnd	robert b. murrett is a retired vice admiral of the united states navy .
stnd_filtered	robert b. murrett is the director of the national geospatial-intelligence agency .
hsmm	robert b. " bob " murrett (born 1956) is an american naval officer and the director .
hier	robert b. murrett is a retired vice admiral in the united states navy .
MBD[.4, .1, .5]	robert b. murrett is a vice admiral in the united states navy .

TABLE A.3.11: A WikiBio input table, coupled with the corresponding sentence and the models-generated outputs.

name	rosane ferreira
image	deputada federal rosane ferreira.jpg
office	federal deputy for state of parná
term_start	1 february 2011
term_end	actual
president	dilma rousseff
order	federal deputy for the state of roraima
birth_date	31 july 1963
birth_place	clevelândia , parná , brazil
dead	alive
nationality	brazilian
party	green party (brazil)
article_title	rosane ferreira

Gold	rosane ferreira (cleusa rosane ribas ferreira , born clevelândia , paraná , july 31 , 1963) , is a nurse and a brazilian politician .
stnd	rosane ferreira (born 31 july 1963 in clevelândia , parná) is a brazilian politician .
stnd_filtered	rosane ferreira (born 31 july 1963) is a brazilian politician .
hsmm	rosane ferreira (born july 31 , 1963) is a brazilian politician and the federal deputy .
hier	rosane ferreira (born 31 july 1963) is a brazilian politician and the current federal deputy for the state of roraima .
MBD[.4, .1, .5]	rosane ferreira (born 31 july 1963 in clevelândia , parná , brazil) is a brazilian politician .

TABLE A.3.12: A WikiBio input table, coupled with the corresponding sentence and the models-generated outputs.

name	polina miller
gender	f
birth_date	23 november 1988
birth_place	saint petersburg , russian sfsr , soviet union
discipline	wag
article_title	polina miller

Gold	polina miller (, born november 23 , 1988 in saint petersburg) is a russian gymnast .
stnd	polina miller (born november 23 , 1988) is a russian artistic gymnast .
stnd_filtered	polina miller (born november 23 , 1988) is a .
hsmm	polina miller (born 23 november 1988 in saint petersburg) is a russian artistic gymnast .
hier	polina miller (born 23 november 1988) is a russian rhythmic gymnast .
MBD[.4, .1, .5]	polina miller (born 23 november 1988 in saint petersburg , russian sfsr , soviet union) is a russian gymnast .

TABLE A.3.13: A WikiBio input table, coupled with the corresponding sentence and the models-generated outputs.

name	pat burke
irish	pádraig de búrca
sport	gaelic football
code	football
county	dublin
province	leinster
clposition	corner forward
club	kilmacud crokes
clubs	kilmacud crokes
counties	dublin
icprovince	1
birth_place	dublin , ireland
article_title	pat burke (gaelic footballer)

Gold	pat burke is an irish gaelic footballer who plays for dublin and kilmacud crokes .
stnd	pat burke is a gaelic footballer from dublin , ireland .
stnd_filtered	pat burke is a gaelic footballer for dublin .
hsmm	pat burke (born in dublin) is a former irish gaelic footballer who played as a gaelic footballer .
hier	pat burke is a former gaelic footballer for dublin .
MBD[.4, .1, .5]	pat burke is a gaelic footballer from county dublin .

TABLE A.3.14: A WikiBio input table, coupled with the corresponding sentence and the models-generated outputs.

name	odiakes
background	non_vocal_instrumentalist
birth_date	march 22
origin	tokyo , japan
instrument	keyboard , synthesizer
genre	j-pop , techno
occupation	composer
years_active	1998 - present
article_title	odiakes

Gold	odiakes (born march 22) is a japanese composer from tokyo , japan who has worked for a variety of visual novel companies .
stnd	, better known by his stage name odiakes , is a japanese composer .
stnd_filtered	odiakes is a japanese composer .
hsmm	odiakes “ odiakes ” (born march 22) is a japanese composer .
hier	composer (born march 22) is a japanese j-pop player .
MBD[.4, .1, .5]	odiakes (born march 22 in tokyo , japan) is a japanese composer .

TABLE A.3.15: A WikiBio input table, coupled with the corresponding sentence and the models-generated outputs.

Club	Istiklol
Season	2015
League	Tajik League
page_title	Parvizdzhon Umarbayev
section_title	Club
section_text	As of match played 29 July 2018

Gold	In 2015 , Umarbayev signed for Tajik League FC Istiklol .
stnd	Umarbayev joined Tajik League side Istiklol in 2015 .
stnd_filtered	2015 , he joined Istiklol in the Tajik League .
hal_WO	Parvizdzhon joined Tajik League club Istiklol in 2015 .
MBD[.4, .1, .5]	Umarbayev signed with Istiklol ahead of the 2015 Tajik League season .

TABLE A.3.16: A ToTTo input table, coupled with the corresponding sentence and the models-generated outputs.

Rank	5
Island	Hulhumeedhoo
page_title	List of islands of the Maldives
section_title	Islands by area size
section_text	This list ranks the top 10 islands of the Maldives by area . Some islands in the Maldives , although geographically one island , are divided into two administrative islands (for example , Gan and Maandhoo in Laamu Atoll) .

Gold	Hulhumeedhoo is the 5th largest island in Maldives .
stnd	It has a area of Hulhumeedhoo km² (Islands sq mi) .
stnd_filtered	is the fourth of the Maldives in Maldives .
hal_WO	Hulhumeedhoo is the largest islands of the Maldives by area size .
MBD[.4, .1, .5]	Hulhumeedhoo is the fifth largest island by area size .

TABLE A.3.17: A ToTTo input table, coupled with the corresponding sentence and the models-generated outputs.

Single	24.7 (Twenty-Four Seven)
page_title	Singular (band)
section_title	2010

Gold	In 2010 , Singular released its first single , " 24.7 (Twenty-Four Seven) " .
stnd	The first single , 24.7 (Twenty-Four Seven) , was released in 2010 .
stnd_filtered	The band won the 24.7 (Twenty-Four Seven) .
hal_WO	24.7 (Twenty-Four Seven) .
MBD[.4, .1, .5]	Singular released their first album , 24.7 (Twenty-Four Seven) .

TABLE A.3.18: A ToTTo input table, coupled with the corresponding sentence and the models-generated outputs.

Bibliography

Abu-Mostafa, Y. S. (1990). Learning from hints in neural networks. *J. Complex.*, *6*(2), 192–198. https://doi.org/10.1016/0885-064X(90)90006-Y

Agarwal, S., & Dymetman, M. (2017). A surprisingly effective out-of-the-box char2char model on the E2E NLG challenge dataset (K. Jokinen, M. Stede, D. DeVault, & A. Louis, Eds.). *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, Saarbrücken, Germany, August 15-17, 2017*, 158–163. https://doi.org/10.18653/v1/w17-5519

Aharoni, R., Goldberg, Y., & Belinkov, Y. (2016). Improving sequence to sequence learning for morphological inflection generation: The BIU-MIT systems for the SIGMORPHON 2016 shared task for morphological reinflection. *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, 41–48. https://doi.org/10.18653/v1/W16-2007

Al-Rfou, R., Choe, D., Constant, N., Guo, M., & Jones, L. (2019). Character-level language modeling with deeper self-attention. *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, 3159–3166. https://doi.org/10.1609/aaai.v33i01.33013159

Althaus, E., Karamanis, N., & Koller, A. (2004). Computing locally coherent discourses (D. Scott, W. Daelemans, & M. A. Walker, Eds.). *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, 21-26 July, 2004, Barcelona, Spain*, 399–406. https://doi.org/10.3115/1218955.1219006

Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate (Y. Bengio & Y. LeCun, Eds.). *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. http://arxiv.org/abs/1409.0473

Ballesteros, M., Bohnet, B., Mille, S., & Wanner, L. (2015). Data-driven sentence generation with non-isomorphic trees (R. Mihalcea, J. Y. Chai, & A. Sarkar, Eds.). *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, 387–397. https://doi.org/10.3115/v1/n15-1042

Banaee, H., Ahmed, M. U., & Loutfi, A. (2013). Towards NLG for physiological data monitoring with body area networks (A. Gatt & H. Saggion, Eds.). *ENLG 2013 - Proceedings of the 14th European Workshop on Natural Language Generation, August 8-9, 2013, Sofia, Bulgaria*, 193–197. https://aclanthology.org/W13-2127/

Banerjee, S., & Lavie, A. (2005). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, 65–72. https://www.aclweb.org/anthology/W05-0909

Belz, A. (2008). Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. *Nat. Lang. Eng.*, *14*(4), 431–455. https://doi.org/10.1017/S1351324907004664

Bengio, Y., Frasconi, P., & Simard, P. Y. (1993). The problem of learning long-term dependencies in recurrent networks. *Proceedings of International Conference on Neural Networks (ICNN'88), San Francisco, CA, USA, March 28 - April 1, 1993*, 1183–1188. https://doi.org/10.1109/ICNN.1993.298725

Bengio, Y., Simard, P. Y., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Networks*, *5*(2), 157–166. https://doi.org/10.1109/72.279181

Bonetta, G., Roberti, M., Cancelliere, R., & Gallinari, P. (2021). The rare word issue in natural language generation: A character-based solution. *Informatics*, *8*(1), 20. https://doi.org/10.3390/informatics8010020

Borensztajn, G., Zuidema, W. H., & Bod, R. (2009). Children's grammars grow more abstract with age - evidence from an automatic procedure for identifying the productive units of language. *topiCS*. https://doi.org/10.1111/j.1756-8765.2008.01009.x

Burke, R. D., Hammond, K. J., & Young, B. C. (1997). The findme approach to assisted browsing. *IEEE Expert*, *12*(4), 32–40. https://doi.org/10.1109/64.608186

Carenini, G., & Moore, J. D. (2006). Generating and evaluating evaluative arguments. *Artif. Intell.*, *170*(11), 925–952. https://doi.org/10.1016/j.artint.2006.05.003

Caruana, R. (1997). Multitask learning. *Mach. Learn.*, *28*(1), 41–75. https://doi.org/10.1023/A:1007379606734

Chen, B., & Cherry, C. (2014). A systematic comparison of smoothing techniques for sentence-level BLEU. *WMT@ACL*. https://www.aclweb.org/anthology/W14-3346

Chen, D. L., & Mooney, R. J. (2008). Learning to sportscast: A test of grounded language acquisition (W. W. Cohen, A. McCallum, & S. T. Roweis, Eds.). *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, *307*, 128–135. https://doi.org/10.1145/1390156.1390173

Chen, Z., Eavani, H., Chen, W., Liu, Y., & Wang, W. Y. (2020). Few-shot NLG with pre-trained language model (D. Jurafsky, J. Chai, N. Schluter, & J. R. Tetreault, Eds.). *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 183–190. https://doi.org/10.18653/v1/2020.acl-main.18

Child, R., Gray, S., Radford, A., & Sutskever, I. (2019). Generating long sequences with sparse transformers. *CoRR*, *abs/1904.10509*. http://arxiv.org/abs/1904.10509

Chisholm, A., Radford, W., & Hachey, B. (2017). Learning to generate one-sentence biographies from wikidata (M. Lapata, P. Blunsom, & A. Koller, Eds.). *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April*

3-7, 2017, Volume 1: Long Papers, 633–642. https://doi.org/10.18653/v1/e17-1060

Cho, K., van Merrienboer, B., Bahdanau, D., & Bengio, Y. (2014a). On the properties of neural machine translation: Encoder-decoder approaches (D. Wu, M. Carpuat, X. Carreras, & E. M. Vecchi, Eds.). *Proceedings of SSST@EMNLP 2014, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, 25 October 2014*, 103–111. https://doi.org/10.3115/v1/W14-4012

Cho, K., van Merrienboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014b). Learning phrase representations using RNN encoder-decoder for statistical machine translation (A. Moschitti, B. Pang, & W. Daelemans, Eds.). *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, 1724–1734. https://doi.org/10.3115/v1/d14-1179

Cho, K., van Merrienboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014c). Learning phrase representations using RNN encoder-decoder for statistical machine translation (A. Moschitti, B. Pang, & W. Daelemans, Eds.). *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, 1724–1734. http://aclweb.org/anthology/D/D14/D14-1179.pdf

Choromanski, K. M., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlós, T., Hawkins, P., Davis, J. Q., Mohiuddin, A., Kaiser, L., Belanger, D. B., Colwell, L. J., & Weller, A. (2021). Rethinking attention with performers. *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. https://openreview.net/forum?id=Ua6zuk0WRH

Clive, J., Cao, K., & Rei, M. (2021). Control prefixes for text generation. *CoRR, abs/2110.08329*. https://arxiv.org/abs/2110.08329

Dai, Z., Yang, Z., Yang, Y., Carbonell, J. G., Le, Q. V., & Salakhutdinov, R. (2019). Transformer-xl: Attentive language models beyond a fixed-length context (A. Korhonen, D. R. Traum, & L. Màrquez, Eds.). *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, 2978–2988. https://doi.org/10.18653/v1/p19-1285

Dale, R. (1989). Cooking up referring expressions (J. Hirschberg, Ed.). *27th Annual Meeting of the Association for Computational Linguistics, 26-29 June 1989, University of British Columbia, Vancouver, BC, Canada, Proceedings*, 68–75. https://doi.org/10.3115/981623.981632

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. *NAACL-HLT*. https://doi.org/10.18653/v1/n19-1423

Dhingra, B., Faruqui, M., Parikh, A., Chang, M.-W., Das, D., & Cohen, W. (2019). Handling divergent reference texts when evaluating table-to-text generation. *ACL*. https://www.aclweb.org/anthology/P19-1483

Doddington, G. (2002). Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. *Proceedings of the Second International Conference on Human Language Technology Research*, 138–145. http://dl.acm.org/citation.cfm?id=1289189.1289273

Dong, L., Huang, S., Wei, F., Lapata, M., Zhou, M., & Xu, K. (2017). Learning to generate product reviews from attributes. *EACL*. https://www.aclweb.org/anthology/E17-1059.pdf

Duboué, P. A., & McKeown, K. R. (2003). Statistical acquisition of content selection rules for natural language generation. *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2003, Sapporo, Japan, July 11-12, 2003*. https://aclanthology.org/W03-1016/

Dusek, O., Howcroft, D. M., & Rieser, V. (2019). Semantic noise matters for neural natural language generation. *INLG*. https://aclweb.org/anthology/papers/W/W19/W19-8652/

Dusek, O., & Jurcícek, F. (2016). Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*. https://doi.org/10.18653/v1/p16-2008

Dusek, O., Novikova, J., & Rieser, V. (2018). Findings of the E2E NLG challenge (E. Krahmer, A. Gatt, & M. Goudbeek, Eds.). *Proceedings of the 11th International Conference on Natural Language Generation, Tilburg University, The Netherlands, November 5-8, 2018*, 322–328. https://doi.org/10.18653/v1/w18-6539

Elhadad, M., McKeown, K. R., & Robin, J. (1997). Floating constraints in lexical choice. *Comput. Linguistics*, *23*(2), 195–239.

Engonopoulos, N., & Koller, A. (2014). Generating effective referring expressions using charts (M. Mitchell, K. F. McCoy, D. McDonald, & A. Cahill, Eds.). *INLG 2014 - Proceedings of the Eighth International Natural Language Generation Conference, Including Proceedings of the INLG and SIGDIAL 2014 Joint Session, 19-21 June 2014, Philadelphia, PA, USA*, 6–15. https://doi.org/10.3115/v1/w14-5002

Ferreira, T. C., van der Lee, C., van Miltenburg, E., & Krahmer, E. (2019). Neural data-to-text generation: A comparison between pipeline and end-to-end architectures (K. Inui, J. Jiang, V. Ng, & X. Wan, Eds.). *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, 552–562. https://doi.org/10.18653/v1/D19-1052

Ficler, J., & Goldberg, Y. (2017). Controlling linguistic style aspects in neural language generation. *Workshop on Stylistic Variation @ ACL*. https://www.aclweb.org/anthology/W17-4912

Filippova, K. (2020). Controlled hallucinations: Learning to generate faithfully from noisy data. *Findings of EMNLP*. https://www.aclweb.org/anthology/2020.findings-emnlp.76

Flesch, R. (1962). *The art of readable writing*. Wiley. https://books.google.it/books?id=4JMB1WybUvYC

Gardent, C., Shimorina, A., Narayan, S., & Perez-Beltrachini, L. (2017a). Creating training corpora for NLG micro-planners (R. Barzilay & M.-Y. Kan, Eds.). *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, 179–188. https://doi.org/10.18653/v1/P17-1017

Gardent, C., Shimorina, A., Narayan, S., & Perez-Beltrachini, L. (2017b). The webnlg challenge: Generating text from RDF data (J. M. Alonso, A.

Bugarín, & E. Reiter, Eds.). *Proceedings of the 10th International Conference on Natural Language Generation, INLG 2017, Santiago de Compostela, Spain, September 4-7, 2017*, 124–133. https://doi.org/10.18653/v1/w17-3518

Gatt, A., & Krahmer, E. (2018). Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *J. Artif. Intell. Res.*, *61*, 65–170. https://doi.org/10.1613/jair.5477

Gatt, A., Portet, F., Reiter, E., Hunter, J., Mahamood, S., Moncur, W., & Sripada, S. (2009). From data to text in the neonatal intensive care unit: Using NLG technology for decision support and information management. *AI Commun.*, *22*(3), 153–186. https://doi.org/10.3233/AIC-2009-0453

Gehring, J., Auli, M., Grangier, D., Yarats, D., & Dauphin, Y. N. (2017). Convolutional sequence to sequence learning (D. Precup & Y. W. Teh, Eds.). *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, *70*, 1243–1252. http://proceedings.mlr.press/v70/gehring17a.html

Gehrmann, S., Dai, F., Elder, H., & Rush, A. (2018). End-to-end content and plan selection for data-to-text generation. *INLG*. https://www.aclweb.org/anthology/W18-6505

Gers, F. A., Schmidhuber, J., & Cummins, F. A. (2000). Learning to forget: Continual prediction with LSTM. *Neural Comput.*, *12*(10), 2451–2471. https://doi.org/10.1162/089976600300015015

Goldberg, E., Driedger, N., & Kittredge, R. I. (1994). Using natural-language processing to produce weather forecasts. *IEEE Expert*, *9*(2), 45–53. https://doi.org/10.1109/64.294135

Goyal, R., Dymetman, M., & Gaussier, É. (2016). Natural language generation through character-based rnns with finite-state prior knowledge (N. Calzolari, Y. Matsumoto, & R. Prasad, Eds.). *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, 1083–1092. http://aclweb.org/anthology/C/C16/C16-1103.pdf

Gu, J., Lu, Z., Li, H., & Li, V. O. K. (2016). Incorporating copying mechanism in sequence-to-sequence learning (K. Erj & N. A. Smith, Eds.). *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. http://aclweb.org/anthology/P/P16/P16-1154.pdf

Han, C.-h., Lavoie, B., Palmer, M. S., Rambow, O., Kittredge, R. I., Korelsky, T., Kim, N., & Kim, M. (2000). Handling stuctural divergences and recovering dropped arguments in a korean/english machine translation system. *AMTA*. https://doi.org/10.1007/3-540-39965-8%5C_5

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hovy, E. (1987). Generating natural language under pragmatic constraints. *Journal of Pragmatics*, *11*(6), 689–719. https://doi.org/https://doi.org/10.1016/0378-2166(87)90109-3

Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R., & Xing, E. P. (2017). Toward controlled generation of text. *ICML*. http://proceedings.mlr.press/v70/hu17e.html

Hwa, R., Resnik, P., Weinberg, A., Cabezas, C. I., & Kolak, O. (2005). Bootstrapping parsers via syntactic projection across parallel texts. *Nat. Lang. Eng.*

Juraska, J., Karagiannis, P., Bowden, K. K., & Walker, M. A. (2018). A deep ensemble model with slot alignment for sequence-to-sequence natural language generation. *NAACL-HLT*. https://www.aclweb.org/anthology/N18-1014

Kale, M., & Rastogi, A. (2020). Text-to-text pre-training for data-to-text tasks (B. Davis, Y. Graham, J. D. Kelleher, & Y. Sripada, Eds.). *Proceedings of the 13th International Conference on Natural Language Generation, INLG 2020, Dublin, Ireland, December 15-18, 2020*, 97–102. https://aclanthology.org/2020.inlg-1.14/

Karpathy, A., & Li, F.-F. (2015). Deep visual-semantic alignments for generating image descriptions. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, 3128–3137. https://doi.org/10.1109/CVPR.2015.7298932

Kasner, Z., & Dusek, O. (2020). Data-to-text generation with iterative text editing (B. Davis, Y. Graham, J. D. Kelleher, & Y. Sripada, Eds.). *Proceedings of the 13th International Conference on Natural Language Generation, INLG 2020, Dublin, Ireland, December 15-18, 2020*, 60–67. https://aclanthology.org/2020.inlg-1.9/

Kedzie, C., & McKeown, K. R. (2020). Controllable meaning representation to text generation: Linearization and data augmentation strategies (B. Webber, T. Cohn, Y. He, & Y. Liu, Eds.). *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, 5160–5185. https://doi.org/10.18653/v1/2020.emnlp-main.419

Kikuchi, Y., Neubig, G., Sasano, R., Takamura, H., & Okumura, M. (2016). Controlling output length in neural encoder-decoders. *EMNLP*. https://www.aclweb.org/anthology/D16-1140.pdf

Kingma, D. P., & Ba, J. (2015a). Adam: A method for stochastic optimization (Y. Bengio & Y. LeCun, Eds.). *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. http://arxiv.org/abs/1412.6980

Kingma, D. P., & Ba, J. (2015b). Adam: A method for stochastic optimization. *ICLR*. http://arxiv.org/abs/1412.6980

Kitaev, N., Kaiser, L., & Levskaya, A. (2020). Reformer: The efficient transformer. *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. https://openreview.net/forum?id=rkgNKkHtvB

Klein, G., Kim, Y., Deng, Y., Senellart, J., & Rush, A. M. (2017). OpenNMT: Open-source toolkit for neural machine translation. *Proc. ACL*. https://doi.org/10.18653/v1/P17-4012

Kosmajac, D., & Keselj, V. (2019). Twitter user profiling: Bot and gender identification. *CLEF*. http://ceur-ws.org/Vol-2380/paper%5C_208.pdf

Kryscinski, W., McCann, B., Xiong, C., & Socher, R. (2019). Evaluating the factual consistency of abstractive text summarization. *CoRR, abs/1910.12840*. http://arxiv.org/abs/1910.12840

Lebret, R., Grangier, D., & Auli, M. (2016). Neural text generation from structured data with application to the biography domain (J. Su, X. Carreras, & K. Duh, Eds.). *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, 1203–1213. https://doi.org/10.18653/v1/d16-1128

LeCun, Y., Bengio, Y., & Hinton, G. E. (2015). Deep learning. *Nat.*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Lee, J., Lee, Y., Kim, J., Kosiorek, A. R., Choi, S., & Teh, Y. W. (2019). Set transformer: A framework for attention-based permutation-invariant neural networks (K. Chaudhuri & R. Salakhutdinov, Eds.). *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, *97*, 3744–3753. http://proceedings.mlr.press/v97/lee19d.html

Lee, K., Firat, O., Agarwal, A., Fannjiang, C., & Sussillo, D. (2019). Hallucinations in neural machine translation. https://openreview.net/forum?id=SkxJ-309FQ

Leppänen, L., Munezero, M., Granroth-Wilding, M., & Toivonen, H. (2017). Data-driven news generation for automated journalism (J. M. Alonso, A. Bugarín, & E. Reiter, Eds.). *Proceedings of the 10th International Conference on Natural Language Generation, INLG 2017, Santiago de Compostela, Spain, September 4-7, 2017*, 188–197. https://doi.org/10.18653/v1/w17-3528

Li, J., Galley, M., Brockett, C., Spithourakis, G. P., Gao, J., & Dolan, B. (2016). A persona-based neural conversation model. *ACL*. https://www.aclweb.org/anthology/P16-1094

Liang, P., Jordan, M. I., & Klein, D. (2009). Learning semantic correspondences with less supervision (K.-Y. Su, J. Su, & J. Wiebe, Eds.). *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore*, 91–99. https://aclanthology.org/P09-1011/

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, 74–81. https://www.aclweb.org/anthology/W04-1013

Lin, S., Wang, W., Yang, Z., Liang, X., Xu, F. F., Xing, E. P., & Hu, Z. (2020). Record-to-text generation with style imitation (T. Cohn, Y. He, & Y. Liu, Eds.). *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, *EMNLP 2020*, 1589–1598. https://doi.org/10.18653/v1/2020.findings-emnlp.144

Liu, T., Luo, F., Xia, Q., Ma, S., Chang, B., & Sui, Z. (2019a). Hierarchical encoder with auxiliary supervision for neural table-to-text generation: Learning better representation for tables. *AAAI*. https://doi.org/10.1609/aaai.v33i01.33016786

Liu, T., Luo, F., Xia, Q., Ma, S., Chang, B., & Sui, Z. (2019b). Hierarchical encoder with auxiliary supervision for neural table-to-text generation: Learning better representation for tables. *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, 6786–6793. https://doi.org/10.1609/aaai.v33i01.33016786

Liu, T., Luo, F., Yang, P., Wu, W., Chang, B., & Sui, Z. (2019c). Towards comprehensive description generation from factual attribute-value tables. *ACLs*. https://doi.org/10.18653/v1/p19-1600

Liu, T., Wang, K., Sha, L., Chang, B., & Sui, Z. (2018). Table-to-text generation by structure-aware seq2seq learning (S. A. McIlraith & K. Q. Weinberger, Eds.). *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, 4881–4888. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16599

Luong, M.-T., & Manning, C. D. (2016). Achieving open vocabulary neural machine translation with hybrid word-character models (K. Erj & N. A. Smith, Eds.). *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. http://aclweb.org/anthology/P/P16/P16-1100.pdf

Luong, T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation (L. Màrquez, C. Callison-Burch, J. Su, D. Pighin, & Y. Marton, Eds.). *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, 1412–1421. https://doi.org/10.18653/v1/d15-1166

Ma, W., Cui, Y., Si, C., Liu, T., Wang, S., & Hu, G. (2020). Charbert: Character-aware pre-trained language model (D. Scott, N. Bel, & C. Zong, Eds.). *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, 39–50. https://doi.org/10.18653/v1/2020.coling-main.4

Mayzner, M., & Tresselt, M. (1965). *Tables of single-letter and digram frequency counts for various word-length and letter-position combinations*. Psychonomic Press. https://books.google.it/books?id=FI7BHgAACAAJ

McKeown, K. R. (1985). Discourse strategies for generating natural-language text. *Artif. Intell.*, *27*(1), 1–41. https://doi.org/10.1016/0004-3702(85)90082-7

Mei, H., Bansal, M., & Walter, M. R. (2016). What to talk about and how? selective generation using lstms with coarse-to-fine alignment (K. Knight, A. Nenkova, & O. Rambow, Eds.). *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, 720–730. https://doi.org/10.18653/v1/n16-1086

Meteer, M. (1991). Bridging the generation gap between text planning and linguistic realization. *Computational Intelligence*, *7*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 26: 27th annual conference on neural information processing systems 2013. proceedings of a meeting held december 5-8, 2013, lake tahoe, nevada, united states* (pp. 3111–3119). http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality

Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., & Khudanpur, S. (2010). Recurrent neural network based language model (T. Kobayashi, K. Hirose, & S. Nakamura, Eds.). *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari,*

Chiba, Japan, September 26-30, 2010, 1045–1048. http://www.isca-speech.org/archive/interspeech%5C_2010/i10%5C_1045.html

Narayan, S., & Gardent, C. (2020). *Deep learning approaches to text production*.

Nie, F., Wang, J., Pan, R., & Lin, C.-Y. (2019). An encoder with non-sequential dependency for neural data-to-text generation. *INLG*. https://aclweb.org/anthology/papers/W/W19/W19-8619/

Novikova, J., Dusek, O., Curry, A. C., & Rieser, V. (2017a). Why we need new evaluation metrics for NLG. *EMNLP*. https://doi.org/10.18653/v1/d17-1238

Novikova, J., Dusek, O., & Rieser, V. (2017b). The E2E dataset: New challenges for end-to-end generation (K. Jokinen, M. Stede, D. DeVault, & A. Louis, Eds.). *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, Saarbrücken, Germany, August 15-17, 2017*, 201–206. https://doi.org/10.18653/v1/w17-5525

Novikova, J., Dusek, O., & Rieser, V. (2017c). The E2E dataset: New challenges for end-to-end generation. *SIGdial Meeting on Discourse and Dialogue*. https://doi.org/10.18653/v1/w17-5525

O'Donnell, M., Mellish, C., Oberlander, J., & Knott, A. (2001). ILEX: an architecture for a dynamic hypertext generation system. *Nat. Lang. Eng.*, *7*(3), 225–250. https://doi.org/10.1017/S1351324901002698

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). Bleu: A method for automatic evaluation of machine translation. *ACL*. https://www.aclweb.org/anthology/P02-1040/

Parikh, A. P., Wang, X., Gehrmann, S., Faruqui, M., Dhingra, B., Yang, D., & Das, D. (2020). Totto: A controlled table-to-text generation dataset (B. Webber, T. Cohn, Y. He, & Y. Liu, Eds.). *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, 1173–1186. https://doi.org/10.18653/v1/2020.emnlp-main.89

Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, *28*, 1310–1318. http://jmlr.org/proceedings/papers/v28/pascanu13.html

Perez-Beltrachini, L., & Gardent, C. (2017). Analysing data-to-text generation benchmarks. *INLG*. https://www.aclweb.org/anthology/W17-3537.pdf

Perez-Beltrachini, L., & Lapata, M. (2018). Bootstrapping generators from noisy data. *NAACL-HLT*. https://doi.org/10.18653/v1/n18-1137

Plachouras, V., Smiley, C., Bretz, H., Taylor, O., Leidner, J. L., Song, D., & Schilder, F. (2016). Interacting with financial data using natural language (R. Perego, F. Sebastiani, J. A. Aslam, I. Ruthven, & J. Zobel, Eds.). *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016*, 1121–1124. https://doi.org/10.1145/2911451.2911457

Portet, F., Reiter, E., Gatt, A., Hunter, J., Sripada, S., Freer, Y., & Sykes, C. (2009). Automatic generation of textual summaries from neonatal intensive care data. *Artif. Intell.*, *173*(7-8), 789–816. https://doi.org/10.1016/j.artint.2008.12.002

Puduppully, R., Dong, L., & Lapata, M. (2019a). Data-to-text generation with content selection and planning. *AAAI*. https://doi.org/10.1609/aaai.v33i01.33016908

Puduppully, R., Dong, L., & Lapata, M. (2019b). Data-to-text generation with content selection and planning. *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, 6908–6915. https://doi.org/10.1609/aaai.v33i01.33016908

Puduppully, R., Dong, L., & Lapata, M. (2019c). Data-to-text generation with entity modeling. *ACL*. https://doi.org/10.18653/v1/p19-1195

Puduppully, R., Dong, L., & Lapata, M. (2019d). Data-to-text generation with entity modeling (A. Korhonen, D. R. Traum, & L. Màrquez, Eds.). *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, 2023–2035. https://doi.org/10.18653/v1/p19-1195

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., & Manning, C. D. (2020). Stanza: A Python natural language processing toolkit for many human languages. *System Demonstrations @ ACL*.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, *21*, 140:1–140:67. http://jmlr.org/papers/v21/20-074.html

Ramos-Soto, A., Diz, A. J. B., Barro, S., & Taboada, J. (2015). Linguistic descriptions for automatic generation of textual short-term weather forecasts on real prediction data. *IEEE Trans. Fuzzy Syst.*, *23*(1), 44–57. https://doi.org/10.1109/TFUZZ.2014.2328011

Rebuffel, C., Soulier, L., Scoutheeten, G., & Gallinari, P. (2020a). A hierarchical model for data-to-text generation (J. M. Jose, E. Yilmaz, J. Magalhães, P. Castells, N. Ferro, M. J. Silva, & F. Martins, Eds.). *Advances in Information Retrieval - 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14-17, 2020, Proceedings, Part I*, *12035*, 65–80. https://doi.org/10.1007/978-3-030-45439-5_5

Rebuffel, C., Soulier, L., Scoutheeten, G., & Gallinari, P. (2020b). Parenting via model-agnostic reinforcement learning to correct pathological behaviors in data-to-text generation. *INLG*. https://www.aclweb.org/anthology/2020.inlg-1.18

Reiter, E. (1994). Has a consensus NL generation architecture appeared, and is it psycholinguistically plausible? *Proceedings of the Seventh International Workshop on Natural Language Generation, INLG 1994, Kennebunkport, Maine, USA, June 21-24, 1994*. https://aclanthology.org/W94-0319/

Reiter, E. (2010). Natural language generation. *The handbook of computational linguistics and natural language processing* (pp. 574–598). John Wiley & Sons, Ltd. https://doi.org/https://doi.org/10.1002/9781444324044.ch20

Reiter, E. (2018). A structured review of the validity of BLEU. *Computational Linguistics*. https://doi.org/10.1162/coli%5C_a%5C_00322

Reiter, E., & Belz, A. (2009). An investigation into the validity of some metrics for automatically evaluating natural language generation systems. *Computational Linguistics*.

Reiter, E., & Dale, R. (1997). Building applied natural language generation systems. *Nat. Lang. Eng.*, *3*(1), 57–87. https://doi.org/10.1017/S1351324997001502

Reiter, E., Mellish, C., & Levine, J. (1995). Automatic generation of technical documentation. *Appl. Artif. Intell.*, *9*(3), 259–287. https://doi.org/10.1080/08839519508945476

Reiter, E., Robertson, R., & Osman, L. (2003). Lessons from a failure: Generating tailored smoking cessation letters. *Artif. Intell.*, *144*(1-2), 41–58. https://doi.org/10.1016/S0004-3702(02)00370-3

Reiter, E., Sripada, S., Hunter, J., Yu, J., & Davy, I. (2005). Choosing words in computer-generated weather forecasts. *Artif. Intell.*, *167*(1-2), 137–169. https://doi.org/10.1016/j.artint.2005.06.006

Roberti, M., Bonetta, G., Cancelliere, R., & Gallinari, P. (2019). Copy mechanism and tailored training for character-based data-to-text generation (U. Brefeld, É. Fromont, A. Hotho, A. J. Knobbe, M. H. Maathuis, & C. Robardet, Eds.). *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2019, Würzburg, Germany, September 16-20, 2019, Proceedings, Part II*, *11907*, 648–664. https://doi.org/10.1007/978-3-030-46147-8_39

Rohrbach, A., Hendricks, L. A., Burns, K., Darrell, T., & Saenko, K. (2018). Object hallucination in image captioning (E. Riloff, D. Chiang, J. Hockenmaier, & J. Tsujii, Eds.). *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, 4035–4045. https://doi.org/10.18653/v1/d18-1437

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. *Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1: Foundations* (pp. 318–362). MIT Press.

Sanguinetti, M., & Bosco, C. (2015). Parttut: The turin university parallel treebank. In R. Basili, C. Bosco, R. Delmonte, A. Moschitti, & M. Simi (Eds.), *Parli*. https://doi.org/10.1007/978-3-319-14206-7%5C_3

Schlör, D., Ring, M., & Hotho, A. (2020). Inalu: Improved neural arithmetic logic unit. *Frontiers Artif. Intell.*, *3*, 71. https://doi.org/10.3389/frai.2020.00071

Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.*, *45*(11), 2673–2681. https://doi.org/10.1109/78.650093

Scott, A. C., Clayton, J. E., & Gibson, E. L. (1991). *Practical guide to knowledge acquisition*. Addison-Wesley.

See, A., Liu, P. J., & Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. *ACL*. https://doi.org/10.18653/v1/P17-1099

Sennrich, R., Haddow, B., & Birch, A. (2016a). Controlling politeness in neural machine translation via side constraints. *NAACL-HLT*. https://www.aclweb.org/anthology/N16-1005.pdf

Sennrich, R., Haddow, B., & Birch, A. (2016b). Neural machine translation of rare words with subword units (K. Erj & N. A. Smith, Eds.). *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. http://aclweb.org/anthology/P/P16/P16-1162.pdf

Sha, L., Mou, L., Liu, T., Poupart, P., Li, S., Chang, B., & Sui, Z. (2018). Order-planning neural text generation from structured data (S. A. McIlraith & K. Q. Weinberger, Eds.). *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, 5414–5421. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16203

Shen, X., Chang, E., Su, H., Zhou, J., & Klakow, D. (2020). Neural Data-to-Text Generation via Jointly Learning the Segmentation and Correspondence. *ACL*. https://www.aclweb.org/anthology/2020.acl-main.641

Shi, S., Chen, H., Ma, W., Mao, J., Zhang, M., & Zhang, Y. (2020). Neural logic reasoning (M. d'Aquin, S. Dietze, C. Hauff, E. Curry, & P. Cudré-Mauroux, Eds.). *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, 1365–1374. https://doi.org/10.1145/3340531.3411949

Shoeybi, M., Patwary, M., Puri, R., LeGresley, P., Casper, J., & Catanzaro, B. (2019). Megatron-lm: Training multi-billion parameter language models using model parallelism. *CoRR*, *abs/1909.08053*. http://arxiv.org/abs/1909.08053

Smeuninx, N., Clerck, B. D., & Aerts, W. (2020). Measuring the readability of sustainability reports: A corpus-based analysis through standard formulae and nlp. *International Journal of Business Communication*. https://doi.org/10.1177/2329488416675456

Stajner, S., & Hulpus, I. (2020). When shallow is good enough: Automatic assessment of conceptual text complexity using shallow semantic features. *LREC*. https://www.aclweb.org/anthology/2020.lrec-1.177/

Stajner, S., Nisioi, S., & Hulpus, I. (2020). Coco: A tool for automatically assessing conceptual complexity of texts. *LREC*. https://www.aclweb.org/anthology/2020.lrec-1.887/

Stock, O., Zancanaro, M., Busetta, P., Callaway, C. B., Krüger, A., Kruppa, M., Kuflik, T., Not, E., & Rocchi, C. (2007). Adaptive, intelligent presentation of information for the museum visitor in PEACH. *User Model. User Adapt. Interact.*, *17*(3), 257–304. https://doi.org/10.1007/s11257-007-9029-6

Sutskever, I., Martens, J., & Hinton, G. E. (2011). Generating text with recurrent neural networks (L. Getoor & T. Scheffer, Eds.). *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, 1017–1024.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, & K. Q. Weinberger, Eds.). *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, 3104–3112. https://proceedings.neurips.cc/paper/2014

Theune, M., Klabbers, E., de Pijper, J.-R., Krahmer, E., & Odijk, J. (2001). From data to speech: A general approach. *Nat. Lang. Eng.*, *7*(1), 47–86. http://journals.cambridge.org/action/displayAbstract?aid=73673

Thomson, C., Zhao, Z., & Sripada, S. (2020). Studying the Impact of Filling Information Gaps on the Output Quality of Neural Data-to-Text. *INLG*. https://www.aclweb.org/anthology/2020.inlg-1.6

Tian, R., Narayan, S., Sellam, T., & Parikh, A. P. (2019). Sticking to the facts: Confident decoding for faithful data-to-text generation. http://arxiv.org/abs/1910.08684

Turner, R., Sripada, S., Reiter, E., & Davy, I. P. (2007). Selecting the content of textual descriptions of geographically located events in spatio-temporal weather data (R. Ellis, T. Allen, & M. Petridis, Eds.). *Applications and Innovations in Intelligent Systems XV - Proceedings of AI-2007, the Twenty-seventh SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, Cambridge, UK, 10-12 December 2007*, 75–88. https://doi.org/10.1007/978-1-84800-086-5_6

van der Lee, C., Gatt, A., van Miltenburg, E., Wubben, S., & Krahmer, E. (2019). Best practices for the human evaluation of automatically generated text. *INLG*. https://aclweb.org/anthology/papers/W/W19/W19-8643/

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need (I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, & R. Garnett, Eds.). *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 5998–6008. https://proceedings.neurips.cc/paper/2017

Vedantam, R., Zitnick, C. L., & Parikh, D. (2015). Cider: Consensus-based image description evaluation. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, 4566–4575. https://doi.org/10.1109/CVPR.2015.7299087

Venigalla, H., & Eugenio, B. D. (2013). UIC-CSC: the content selection challenge entry from the university of illinois at chicago (A. Gatt & H. Saggion, Eds.). *ENLG 2013 - Proceedings of the 14th European Workshop on Natural Language Generation, August 8-9, 2013, Sofia, Bulgaria*, 210–211. https://aclanthology.org/W13-2134/

Viethen, J., & Dale, R. (2008). The use of spatial relations in referring expression generation (M. White, C. Nakatsu, & D. McDonald, Eds.). *INLG 2008 - Proceedings of the Fifth International Natural Language Generation Conference, June 12-14, 2008, Salt Fork, Ohio, USA*. https://aclanthology.org/W08-1109/

Vinyals, O., Fortunato, M., & Jaitly, N. (2015). Pointer networks (C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, & R. Garnett, Eds.). *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, 2692–2700. http://papers.nips.cc/paper/5866-pointer-networks

Wang, H. (2019). Revisiting challenges in data-to-text generation with fact grounding. *INLG*. https://aclweb.org/anthology/papers/W/W19/W19-8639/

Wang, P.-W., Donti, P. L., Wilder, B., & Kolter, J. Z. (2019). Satnet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver (K. Chaudhuri & R. Salakhutdinov, Eds.). *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, *97*, 6545–6554. http://proceedings.mlr.press/v97/wang19e.html

Wen, T.-H., Gasic, M., Kim, D., Mrksic, N., Su, P.-H., Vandyke, D., & Young, S. J. (2015a). Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking. *Proceedings of the SIGDIAL 2015 Conference, The 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 2-4 September 2015, Prague, Czech Republic*, 275–284. https://doi.org/10.18653/v1/w15-4639

Wen, T.-H., Gasic, M., Mrksic, N., Su, P.-h., Vandyke, D., & Young, S. J. (2015b). Semantically conditioned lstm-based natural language generation for spoken dialogue systems (L. Màrquez, C. Callison-Burch, J. Su, D. Pighin, & Y. Marton, Eds.). *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, 1711–1721. https://doi.org/10.18653/v1/d15-1199

White, M., & Howcroft, D. M. (2015). Inducing clause-combining rules: A case study with the sparky restaurant corpus (A. Belz, A. Gatt, F. Portet, & M. Purver, Eds.). *ENLG 2015 - Proceedings of the 15th European Workshop on Natural Language Generation, 10-11 September 2015, University of Brighton, Brighton, UK*, 28–37. https://doi.org/10.18653/v1/w15-4704

Williams, R. J., & Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Comput.*, *1*(2), 270–280. https://doi.org/10.1162/neco.1989.1.2.270

Wiseman, S., Shieber, S. M., & Rush, A. M. (2017). Challenges in data-to-document generation (M. Palmer, R. Hwa, & S. Riedel, Eds.). *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, 2253–2263. https://doi.org/10.18653/v1/d17-1239

Wiseman, S., Shieber, S. M., & Rush, A. M. (2018). Learning neural templates for text generation (E. Riloff, D. Chiang, J. Hockenmaier, & J. Tsujii, Eds.). *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, 3174–3187. https://doi.org/10.18653/v1/d18-1356

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., & Brew, J. (2019). Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, *abs/1910.03771*.

Xia, F., & Palmer, M. (2001). Converting dependency structures to phrase structures. *HLT*. https://www.aclweb.org/anthology/H01-1014/

Yang, Z., Dai, Z., Yang, Y., Carbonell, J. G., Salakhutdinov, R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding (H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, & R. Garnett, Eds.). *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 5754–5764. https://proceedings.neurips.cc/paper/2019/hash/dc6a7e655d7e5840e66733e9ee67cc69-Abstract.html

Yang, Z., Blunsom, P., Dyer, C., & Ling, W. (2017). Reference-aware language models (M. Palmer, R. Hwa, & S. Riedel, Eds.). *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, 1850–1859. https://doi.org/10.18653/v1/d17-1197

Yu, S.-Z. (2010). Hidden semi-markov models. *Artif. Intell.*, *174*(2), 215–243. https://doi.org/10.1016/j.artint.2009.11.011

Zhang, B., Xiong, D., & Su, J. (2018). Accelerating neural transformer via an average attention network (I. Gurevych & Y. Miyao, Eds.). *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, 1789–1798. https://doi.org/10.18653/v1/P18-1166