The Rare Word Issue in Natural Language Generation: A Character-Based Solution

Giovanni Bonetta, Marco Roberti, Rossella Cancelliere, Patrick Gallinari

informatics

MDPI

Article

The Rare Word Issue in Natural Language Generation: A Character-Based Solution

Giovanni Bonetta [1], Marco Roberti [1], Rossella Cancelliere [1,*] and Patrick Gallinari [2,3]

[1] Department of Computer Science, University of Turin, 10149 Turin, Italy; giovanni.bonetta@unito.it (G.B.); m.roberti@unito.it (M.R.)
[2] Laboratoire d'Informatique de Paris 6, Sorbonne University, CNRS, 75005 Paris, France; patrick.gallinari@lip6.fr
[3] Criteo AI Lab, 75009 Paris, France
* Correspondence: rossella.cancelliere@unito.it; Tel.: +39-011-6706737

Abstract: In this paper, we analyze the problem of generating fluent English utterances from tabular data, focusing on the development of a sequence-to-sequence neural model which shows two major features: the ability to read and generate character-wise, and the ability to switch between generating and copying characters from the input: an essential feature when inputs contain rare words like proper names, telephone numbers, or foreign words. Working with characters instead of words is a challenge that can bring problems such as increasing the difficulty of the training phase and a bigger error probability during inference. Nevertheless, our work shows that these issues can be solved and efforts are repaid by the creation of a fully end-to-end system, whose inputs and outputs are not constrained to be part of a predefined vocabulary, like in word-based models. Furthermore, our copying technique is integrated with an innovative shift mechanism, which enhances the ability to produce outputs directly from inputs. We assess performance on the E2E dataset, the benchmark used for the E2E NLG challenge, and on a modified version of it, created to highlight the rare word copying capabilities of our model. The results demonstrate clear improvements over the baseline and promising performance compared to recent techniques in the literature.

Keywords: data-to-text generation; deep learning; sequence-to-sequence models; natural language processing

Citation: Bonetta, G.; Roberti, M.; Cancelliere, R.; Gallinari, P. The Rare Word Issue in Natural Language Generation: A Character-Based Solution. *Informatics* **2021**, *8*, 20. https://doi.org/10.3390/informatics8010020

Academic Editor: Antony Bryant

Received: 19 January 2021
Accepted: 17 March 2021
Published: 23 March 2021

1. Introduction

Natural Language Generation (NLG) is the research domain that focuses on automatically generating narratives and reports in fluent, well-structured, and rich natural language text, in order to describe, summarize, or explain input data [1,2]. It includes, among others, the fields of machine translation, automatic summarization, simplification of complex texts, generation of paraphrases, and data journalism.

In the last decade, a paradigm shift occurred: neural networks and deep learning-based methods have increasingly been used as building blocks in NLG algorithms, to obtain completely end-to-end outcomes, i.e., outputs generated without non-neural pre-processing or post-processing [3,4]. As deep learning is data-driven by definition, and it is typically used in an end-to-end mode, the availability of big data makes it possible to obtain new systems which shift from symbolic to data-driven methods, and from modular to comprehensive design. This shift has the major benefit of obtaining architectures that are intrinsically more general and directly applicable to very different domains [5].

This paper involves the application of deep Recurrent Neural Networks (RNNs) in particular to Data-to-Text (DTT) generation, a subfield of computational linguistics and natural language generation which aims at transcribing *structured data* into natural language descriptions [6]. RNN architectures are among the most used architectures for text processing because the input can be treated sequentially by reading inputs token

by token while updating network recurrent states. Some of the most interesting DTT applications are soccer and weather reports, summaries of patient information in clinical contexts, and robo-journalism.

Neural methods for DTT have shown the ability to produce fluent texts conditioned on input data in several domains [7,8], without relying on heavy manual work from field experts. They successfully combined ideas from machine translation, such as encoder–decoder architectures [9] and attention mechanisms [10,11], and from extractive summarization, such as the copy mechanism [12,13], often explicitly modeling the key-value structure of the input table [14–19]. However a common issue to deal with is that most of the commonly used models share the word-by-word representation of data in both input sequences and generated utterances; such schemes cannot be effective without a special, non-neural delexicalization phase that handles rare or unknown words, such as proper names, telephone numbers, or foreign words [20].

Various recent studies tried to solve this problem: Gu et al. [12] present *CopyNet*, a word-based technique that can integrate output generation with a copying mechanism that can choose portions of the input sequence and include them in the final sentence. Similarly, in the word-based *Pointer-Generator Network* [13], 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.

One of the very first attempts to model natural language via a character-level mechanism is described by Sutskever et al. [21]. 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. [22], 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".

The character-based approach developed in our work is characterized by the ability to *switch between generating a character and copying it* when required. This makes it possible to naturally handle rare words using the copy mechanism for carrying data from the original raw input directly to output. This feature is very hardly obtained by word-based architectures, which have to learn every single word-to-word self-mapping from scratch. Moreover, during the training phase, we also exploit an innovative shift mechanism that helps the model focusing on the word to be copied, character by character.

Our technique allows us to deal with a significantly smaller vocabulary, reduced to ∼100 symbols, i.e., the printable ASCII characters, obtaining the desirable side effect of reducing the computational effort, something that remains a bottleneck of deep learning systems; furthermore, a more general, domain-free technique is obtained this way.

In Section 2, we briefly review the background and present our model. Section 3 describes the datasets and the metrics chosen to evaluate performance, together with some implementation details. Section 4 is devoted to the achieved results, subsequently discussed in Section 5. Finally, in Section 6, we draw some conclusions and outline future work.

2. Materials and Methods

Figure 1 shows the encoder–decoder architecture with an attention mechanism introduced by Bahdanau et al. [10]. It builds upon the original encoder–decoder model, introduced by Cho et al. [9], and based on two RNNs. One RNN encodes a sequence of symbols into a representation vector (encoder RNN, left part of Figure 1), and the other one decodes the representation into another sequence of symbols (decoder RNN, right part of Figure 1). The sequence of symbols can be a phrase, part of a sentence, a sentence, or a longer linguistic unit [23].

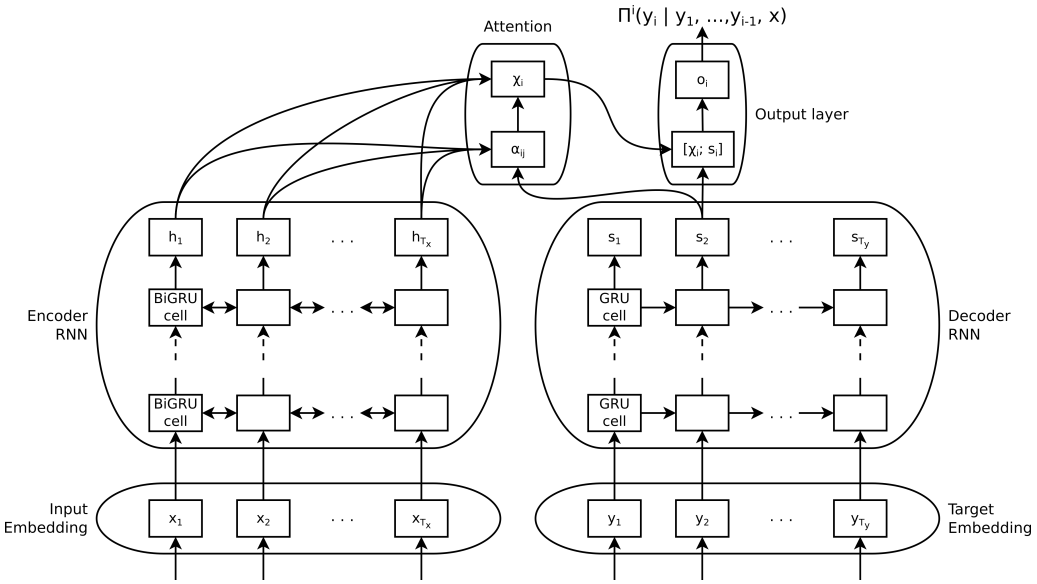


Figure 1. Encoder–decoder with attention model.

Usually, the encoder is made of one or more stacked bi-directional RNNs (represented by double arrows on the left side of Figure 1) while the decoder is made of single-directional RNNs. Both are formed by Gated Recurrent Units (GRUs), the RNNs introduced by Cho et al. [9], to alleviate the well-known vanishing (or exploding) gradient problem. In fact, when dealing with longer texts, RNNs could run into several gradient issues, as the gradient is calculated through a long chain of recurrence.

Bahdanau et al. [10] successively added the now standard attention mechanism to this model, shown in the upper side of Figure 1. As can be seen, the encoder outputs a list of annotations h_j, one for each input embedding x_j; the decoder reads a context vector

$$\chi_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j, \tag{1}$$

built from these T_x annotations.

The sum weights α_{ij} are computed via the attention mechanism:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} \tag{2}$$

where e_{ij} is an *alignment model*, and $e_{ij} = att(s_{i-1}, h_j)$, which evaluates the matching between the jth input and the output produced in the ith time instant. It is based on the decoder's previous output s_{i-1} and on the jth encoder's annotation, and is parameterized as a single hidden layer feed-forward neural network, jointly trained with the whole model. We can also reformulate α_{ij} as

$$\alpha_{ij} = [softmax(e_i)]_j, \qquad j = 1, \ldots, T_x \tag{3}$$

where e_i is the vector whose elements are the alignment model e_{ij}, which evaluates, in a given temporal instant i, the matching with every character j of the input sequence.

In our work, we build upon this model, including the major feature of a character-based copy mechanism. This improvement results in a model that never produces non-words or hallucinations.

2.1. Copy Mechanism

The copy mechanism included in our system is inspired by the pointer-generator network [13], a word-based model that hybridizes Bahdanau et al. [10]'s model and a pointer network [24]. This combination allows for both word generation from a fixed

vocabulary and words copying via directly pointing to an input position, taking advantage of a soft switch between those two operating modes. To adapt this framework to our character-based model, we define two important probability distributions: the *character distribution* Π_{char} and the *attention distribution* Π_{att}. Π_{char} represents the probability of sampling a given token at time i, and is computed as

$$\Pi^i_{char} = softmax(o_i), \qquad o_i = V \cdot [\chi_i, s_i] + b \tag{4}$$

where o_i contains the unnormalized log probabilities over the alphabet at time i, computed from the decoder's output state s_i and the context vector χ_i; V and b are parameters learned during the training phase. That is, o_i is just the result of a linear projection from $[\chi_i, s_i]$ to a vector with the same dimension as the vocabulary, and each vector's element is the GRU cell's estimate of how likely is to pick the corresponding character as the next output token.

Π_{att} is the distribution created by the attention mechanism over the input tokens: we define it as coinciding with α_{ij}, introduced in Equation (2):

$$\Pi^{ij}_{att} \equiv \alpha_{ij}. \tag{5}$$

Π^{ij}_{att} can be represented in a graphical form, known as the attention matrix: this meaningful diagram makes it easier to understand how the model is handling a certain input sequence. Some examples of attention matrices are shown in Section 5.

In Bahdanau et al. [10]'s model, the next output token is sampled just from the vocabulary distribution Π^i_{char}, while Π^{ij}_{att} is only used to weigh the input annotations and create the context vector χ_i.

We built upon this model by adding the capability to learn when to sample from Π^i_{char}, for generating a text item, or from Π^{ij}_{att}, for copying the actual input item.

This is made possible by the definition of a generation probability $p^i_{gen} \in [0, 1]$ computed as:

$$p^i_{gen} = \sigma(W_y \cdot y_{i-1} + W_s \cdot s_i + W_p \cdot p^{i-1}_{gen} + W_c \cdot \chi_i), \tag{6}$$

where y_{i-1} is the last output character's embedding, s_i is the current decoder cell state, p^{i-1}_{gen} is the last computed value of p^i_{gen}, and χ_i is the current context vector. W_y, W_s, W_c, and W_p are parameters learned during training and σ is the sigmoid function.

The resulting final probability distribution is the sum of Π_{char} and Π_{att}, weighted by p^i_{gen}:

$$\Pi^i(c) = p^i_{gen} \cdot \Pi^i_{char}(c) + (1 - p^i_{gen}) \sum_{j|x_j=c} \Pi^{ij}_{att}(c), \tag{7}$$

where c is a specific character and the j index scrolls through all the occurrences of c in the input sequence. p^i_{gen} acts as a soft switch because when it is close to 1, $\Pi^i(c) \simeq \Pi^i_{char}(c)$: our model generates the output as if it was a standard encoder–decoder with attention. Conversely, when p^i_{gen} is close to 0, $\Pi^i(c) \simeq \sum_{j|x_j=c} \Pi^{ij}_{att}(c)$, and a copying step is carried out.

The key difference from the word-based *pointer-generator network* [13] is that in our model, p^{i-1}_{gen} contributes to the determination of p^i_{gen} (see equation). We introduced this dependency because in a character-based model, it is desirable that, at inference time, the switch maintains the same value—predictably near 1 or 0—for a fair number of time steps, to correctly complete the word; knowledge of the last choice helps this behavior. Conversely, in a word-based model, a single copying time step, when required, is typically enough.

We also propose a new formulation of $\Pi^i(c)$ which helps the model to learn how to copy characters:

$$\Pi^i(c) = p^i_{gen} \cdot \Pi^i_{char}(c) + (1 - p^i_{gen}) \sum_{j|x_j=c} \Pi^*_{att}(c). \tag{8}$$

Here, $\Pi^*_{att} = \Pi^{i,j-1}_{att}$, i.e., the attention distribution shifted one step on the right with respect to the considered input position j. This is done because for the model, it is easier

to learn how to center the attention distribution on just a few characters; for instance, the space (' ') and the opening square bracket ('['), which appear as word separators in the record strings, are frequently the prefix of a word that has to be copied. Once this is achieved, the attention distribution is translated one step on the right, over the first letter to copy. Figure 2 shows the convenience of this approach.

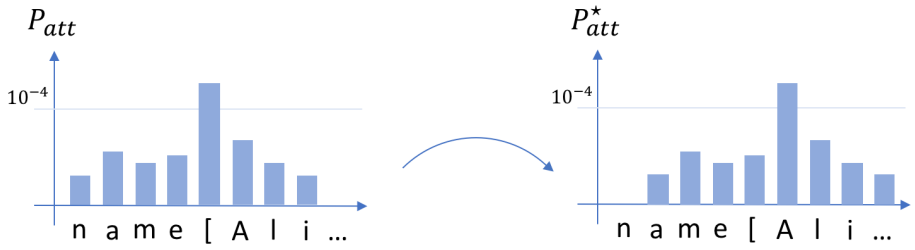


Figure 2. An example of the shift trick.

3. Experimental Setup

3.1. Datasets

We assess our model's performance using two datasets: E2E, which is well-known and frequently used in the literature [25–30], and E2E-NY, which we specifically created to highlight the copying capabilities of our model.

The E2E dataset [31] was used as a benchmark for the E2E Challenge, organized by the Heriot-Watt University in 2017. It is a crowdsourced collection with 42,061/4672/4693 instances for train/dev/test, respectively. It is conceived for training data-driven, end-to-end NLG models in the restaurant domain. Table 1 shows a typical E2E data sample.

E2E has 8.1 reference sentences on average, for every Meaning Representation (MR); each MR is in turn composed of a list of key-value pairs. The ontology consists of eight attributes of different types.

Table 1. An E2E data instance. The meaning representation appears in the dataset once for each reference sentence.

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.

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*, which could virtually include a very broad range of possible values, contains only 19 different restaurant names. Moreover, the test set always contains values which are also present in the training set.

Consequently, we created E2E-NY, an augmented version of the E2E dataset, replacing all the values in the *name* slot with New York restaurant names, as contained in the Entree dataset presented in Burke et al. [32]. Consequently, values that belong to the training set are not found in the development set or in the test set, ensuring the absence of generation bias in the copy mechanism.

All the data were converted to ASCII prior to use, so that the model's vocabulary is independent of the input.

3.2. Metrics

Performance was assessed according to five different metrics [6]: BLEU, NIST, METEOR, ROUGE_L, and CIDEr. These are the same metrics used in the E2E Challenge. Here is a brief description of each one:

- BLEU [33]: It is a precision-based metric that computes the n-gram overlap between the reference and the hypothesis. In particular, BLUE is the ratio of the number of overlapping n-grams over the total number of n-grams in the hypothesis;
- NIST [34]: It is a variant of BLEU which gives more credit to rare n-gram and less credit to common ones;
- METEOR [35]: It 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 criteria;
- ROUGE_L [36]: It is 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 [37]: It weighs each hypothesis' 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.

3.3. Baseline and Competitors

In order to show that our proposed Encoder–Decoder model with Attention, Copy, and Shift (hereafter, ED+ACS) represents an effective improvement with respect to previous work, we compare it to the following models:

- ED+A: a character-based Encoder–Decoder model with Attention [10]; a standard baseline in the literature [13,22,38];
- Qader et al. [25]'s model: a word-based encoder–decoder with attention;
- Puzikov and Gurevych [29]'s model: a non-neural system that was ranked as the best template-based model in the E2E Challenge [39];
- Dusek and Jurcícek [40]'s model: the strong word-based baseline of the E2E challenge [39]. Its pipeline consists of a delexicalizer, a neural encoder–decoder system which outputs a syntax tree using beam search with re-ranking, a surface realizer, and a relexicalizer.

3.4. Implementation Details

We developed our system using the PyTorch framework, release 0.4 (https://pytorch.org/, accessed on 22 March 2021).

The training was performed via Teacher Forcing [41], aiming at maximizing the log-likelihood between output utterances and the target ones. We optimized this loss using the Adam [42] algorithm with learning rate cosine annealing [43].

As RNNs are typically subject to the exploding gradient problem, gradient norm clipping [44] was applied. Three-fold cross-validation was used to find the optimal hyperparameters and training settings values, as shown in Table 2. In this configuration, EDA+ACS has 5,940,960 trainable parameters.

Table 2. Model hyperparameters and training settings used in our experiments.

Hyperparameter	Value
Embedding size	32
GRU hidden size	300
No. of recurrent layers	3
Attention size	128
Learning rate	10^{-3}
$\beta_1; \beta_2$ (Adam [42])	0.9; 0.999
$T_{max}; \eta_{min}$ (annealing [43])	50,000; 0
Max gradient norm [44]	5
Batch size	32
No. of epochs	32

We use the evaluation script (https://github.com/tuetschek/e2e-metrics, accessed on 22 March 2021) provided by the E2E NLG Challenge organizers, which compares the models' output utterances with target ones. We provide results on E2E and E2ENY test sets. Code and data are available at https://github.com/marco-roberti/char-dtt-rareword (accessed on 22 March 2021).

Training and inference were performed on 24GB NVIDIA GPUs (TITAN RTX and Quadro P6000). The training time was in the order of magnitude of ∼10 h, depending on the hardware. Generation at inference occurred in real time, i.e., roughly 2 minutes for the 4693 test instances.

4. Results

Table 3 compares the performance from baselines and our model on the E2E dataset (results come from the proposed methods' original papers, with the exception of ED+A, which we implemented).

In order to assess the statistical significance of our results, we trained and tested our models five times and validated them via Student's *t*-test with a significance level of 99.5%. This showed that ED+ACS was consistently better than ED+A by at least 5%. Our method also performed better than Puzikov and Gurevych [29] BLEU, NIST, and CIDEr, while the values for METEOR and ROUGE_L were statistically equal. This fact can be explained by the fact that METEOR and ROUGE_L are based on the F-measure, which takes into account both the precision and the recall, and high recall is easily achieved by template-based systems as the work of Puzikov and Gurevych [29]. Statistical equality also applies with respect to the metrics of Qader et al. [25].

Dusek and Jurcícek [40] obtained better results for three metrics out of five, i.e., NIST, ROUGE_L, and CIDEr; however, this is not a neural end-to-end method, and needs the additional handcrafted pre- and post-processing phases described in Section 3.3.

Table 3. Performance comparison on the E2E test set.

Model	Metric				
	BLEU	NIST	METEOR	ROUGE_L	CIDEr
ED+A	0.5704	7.8060	0.3895	0.6283	1.5877
Qader et al. [25]	0.655	—	0.450	0.673	—
Puzikov and Gurevych [29]	0.5657	7.4544	0.4529	0.6614	1.8206
Dusek and Jurcícek [40]	0.6593	8.6094	0.4483	0.6850	2.2338
ED+ACS (our model)	0.6400	8.3467	0.4463	0.6680	2.0264

Differences among the various models were even more interesting when the E2E-NY dataset was used for comparison, as shown in Table 4. The Student's *t*-test results at a significance level of 99.5% demonstrated that:

1. ED+ACS was always better than ED+A by more than 17% according to all metrics, achieving as much as a 190% improvement on the CIDEr value;
2. ED+ACS obtained better results than those presented by Qader et al. [25] and Puzikov and Gurevych [29] with respect to all metrics, except METEOR in the latter method;
3. ED+ACS demonstrated statistically equivalent results to those presented by Dusek and Jurcícek [40].

Table 4. Performance comparison on the E2E-NY test set.

Model	Metric				
	BLEU	**NIST**	**METEOR**	**ROUGE_L**	**CIDEr**
ED+A	0.5151	7.1702	0.3617	0.5608	0.8505
Qader et al. [25]	0.5092	7.2798	0.3756	0.5413	0.8768
Puzikov and Gurevych [29]	0.5606	7.7671	0.4535	0.6608	2.3787
Dusek and Jurcícek [40]	0.6517	8.8043	0.4421	0.6749	2.7136
ED+ACS (our model)	0.6482	8.6563	0.4521	0.6770	2.7346

Notice that the scores of the competing models did not drop below a certain threshold because, even if new names were not correctly reproduced, values occurring in other fields of the generated sentences were generally still correct. We hypothesize that their performances would be even worse on datasets containing unseen values in other fields as well (e.g., *food*, *near*).

5. Discussion

Table 5 shows three examples of output sentences generated when the same MR (from the E2E-NY test set) was presented to all models.

The first example consists of a short MR including three key-value pairs. Our model is not the only one that successfully verbalized the whole information; however, ED+ACS tried to express it in a lexically richer way than the models of Puzikov and Gurevych [29] and Dusek and Jurcícek [40]. This came at the cost of the repetition of the word "pub". In contrast, both ED+A and Qader et al. [25]'s model failed to express the pub's name, falling back to similar training names.

In the second example, we noted that the ED+A and Qader et al. [25]'s model were not able to reproduce the restaurant name "Brother's Bar-B-Q" in the output, while all other models did so. In addition, in Dusek and Jurcícek [40]'s model, the price range was completely skipped. Only Puzikov and Gurevych [29]' model and our model generated a complete and credible sentence relying on all key-value pairs.

The third table row shows an example in which our model generated poorly with respect to Dusek and Jurcícek [40] and Puzikov and Gurevych [29]'s models. It repeated "city centre" twice, and did not consider the "family friendly" value. The issue of repeating part of sentences is very common among neural recurrent models, and is also present in the ED+A output. Again, both ED+A and Qader et al. [25]'s model failed to reproduce the *name*.

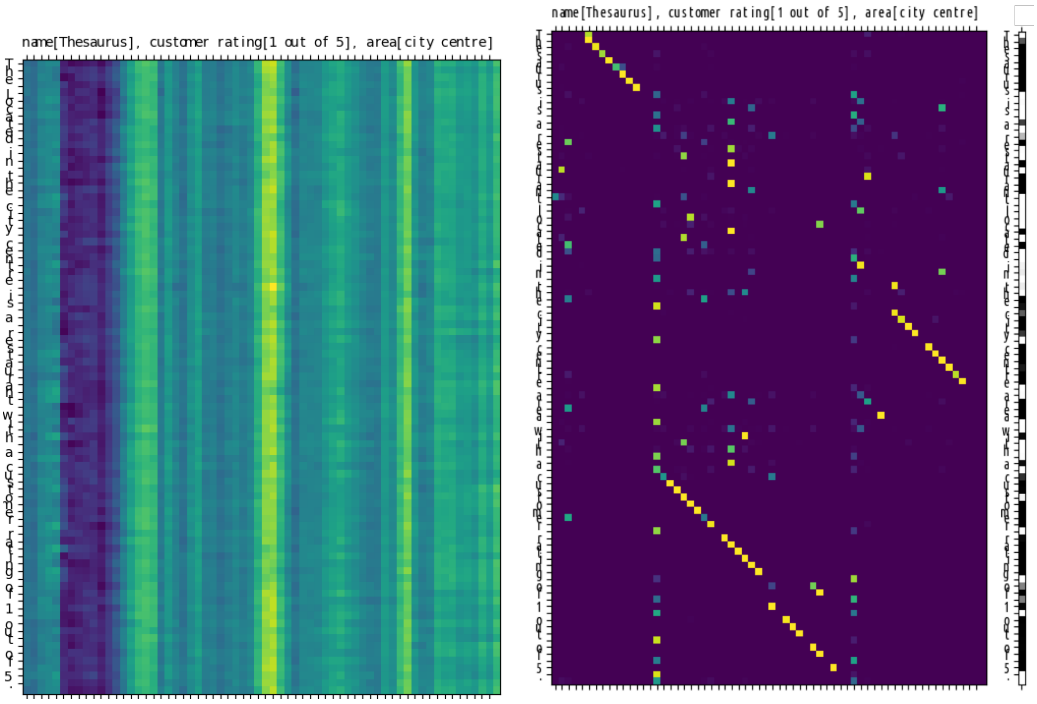
In summary, ED+A and Qader et al. [25]'s approaches did not seem to be able to deal with the rare words in the E2E-NY test set. They continuously failed in realizing the *name* field, which contains values that were not seen in the training phase. In practice, both models could "borrow" values from other fields, use similar values from the training set, or simply avoid any reference to the restaurant names. In all cases, this results in weird, or even syntactically incorrect, sentences. The template-based model proposed by Puzikov and Gurevych [29] always leads to the realization of the whole input by design. However, its sentences always follow the same structure, resulting in more boring and less human-like sentences. The qualitative analysis confirmed what the metrics suggest: the performances of ED+ACS and Dusek and Jurcícek [40]'s were substantially equivalent from the point of view of generation capabilities. Nevertheless, we stress the fact that ED+ACS, differently from Dusek and Jurcícek [40]'s model, is a fully end-to-end system

that does not need any pre-processing or post-processing phase, and that it is inherently more general thanks to its character-based design.

Table 5. Meaning representations, references, and model outputs on various E2E-NY test set examples.

MR	Model	Output
name[Zoe], eatType[pub], near[The Rice Boat]	Reference	Near The Rice Boat you can visit pub called Zoe.
	ED+A	Zona Rosa is a pub near The Rice Boat.
	Qader et al. [25]	Near The Rice Boat is a pub called the The Black Sheep which is a great place to eat
	Puzikov and Gurevych [29]	Zoe is a pub located near The Rice Boat.
	Dusek and Jurcícek [40]	Zoe is a pub near The Rice Boat .
	ED+ACS (our model)	Zoe is a pub that provides pub food items and is located near The Rice Boat.
name[Brother's Bar-B-Q], eatType[pub], food[French], priceRange[moderate], customer rating[1 out of 5], area[riverside], familyFriendly[yes], near[Crowne Plaza Hotel]	Reference	A family friendly pub, Brother's Bar-B-Q, has a moderately expensive menu that is located next to Crowne Plaza Hotel on the river.
	ED+A	Brother Jimmy's BBQ is a pub located in the riverside area. It serves French food in the moderate price range.
	Qader et al. [25]	The French pub called Burger King is located near Crown Plaza Hotel in the riverside area. It is kid friendly and has a moderate price range.
	Puzikov and Gurevych [29]	Brother's Bar-B-Q is a family-friendly pub which serves French food in the moderate price range. It has a customer rating of 1 out of 5 and is located in the riverside area, near Crowne Plaza Hotel.
	Dusek and Jurcícek [40]	Brother's Bar-B-Q is a pub near Crowne Plaza Hotel in the riverside area . It serves french food and has a customer rating of 1 out of 5 and is kids friendly .
	ED+ACS (our model)	Brother's Bar-B-Q is a pub located near the Crowne Plaza Hotel in the riverside area. It is kids friendly and serves French food with a moderate price range and a customer rating of 1 out of 5.
name[Afghan Kebab House], eatType[restaurant], food[Italian], priceRange[high], area[city centre], familyFriendly[no], near[Raja Indian Cuisine]	Reference	Located in the city centre near Raja Indian Cuisine is an adult restaurant called Afghan Kebab House. This high priced restaurant serves Italian food.
	ED+A	Antico Caffee is a not family-friendly Italian restaurant which is not family friendly.
	Qader et al. [25]	The restaurant near Raja Indian Cuisine in the city centre is called The Leopard, which is not child friendly and serves Italian food but is highly priced.
	Puzikov and Gurevych [29]	Afghan Kebab House is a restaurant which serves Italian food in the high price range. It is located in the city centre area, near Raja Indian Cuisine. It is not family friendly.
	Dusek and Jurcícek [40]	Afghan Kebab House is an italian restaurant in the city centre near Raja Indian Cuisine . It is not children friendly and has a high price range.
	ED+ACS (our model)	Afghan Kebab House is a high priced Italian restaurant located in the city centre near Raja Indian Cuisine in the city centre.

We can also inspect the attention distribution for every time step in the decoding phase. In the attention matrices, every ith row is the attention distribution that was used to output the ith token, i.e., the Π_{att} distribution described in Section 2.1. Figure 3a shows the attention matrix generated by ED+A when presented with the input "name[Thesaurus], customer rating[1 out of 5], area[city centre]". Each row is similar. This means that the context vector χ_i (see Equation (1)) is always almost the same: the decoder sees similar input information at every time step. The dark column in the "Thesaurus" word position indicates that the restaurant's name is the less weighted part of the input, another indication that the ED+A model is just skipping this field's value. The decoder is prevented from the realization of the "name" field because the context is "unaware" of those data.



(**a**) (**b**)

Figure 3. Attention matrix of the two models when presented with the input "name[Thesaurus], customer rating[1 out of 5], area[city centre]". In the x axes, there is the input sequence. In the y axes, there is the output sequence. Every ith row is the attention distribution that was used to output the ith token. Therefore, the value in position (i, j) is the "focus" value over the jth character when the ith letter was given in the output. Light dots mean higher attention than dark ones. (**a**) ED+A model. (**b**) Encoder–Decoder model with Attention, Copy, and Shift (ED+ACS). The column on the right shows the p^i_{gen} values for every time step, black being near 0 and white near 1.

Figure 3b shows the attention matrix generated by the ED+ACS model when presented with the same sample as in Figure 3a. On the right, a column with the p^i_{gen} values for every time step is shown. Colors darkening to black mean that the model is weighting the attention distribution more than the alphabet one; i.e., a copying step.

Notice that each ith row is shifted one time step to the right before the decoder uses it to generate the ith output letter, as shown in Equation (7). It is clearly noticeable that the network learned to precisely focus on specific letters, and all attention distributions were more peaked with respect to what happened using the ED+A model (Figure 3a). For example, in the upper left corner, we see that the network almost perfectly chose to focus on every letter in "Thesaurus" while reporting it to the output. For the same time steps, the p^i_{gen} values are close to 0, confirming that the model was copying. We can therefore state that the ED+ACS model was more effective than the ED+A model.

6. Conclusions

In this paper, we showed the advantages of using a character-based sequence-to-sequence model to perform DTT tasks. Using a reduced vocabulary lets the model circumvent the rare word issue, as every word is in fact a sequence of characters. Moreover, the copying and shifting mechanisms introduced in this work give the network the ability to bring input portions directly to the output. This eliminates the need to leverage non-neural scripts to pre-process and post-process data, as usually happens when dealing with lexicalization and tokenization steps. Therefore, our work demonstrates that a fully end-to-end system can achieve results that are comparable to top-performing systems. In addition, it is not constrained to a specific domain, allowing for the creation of more general models.

Author Contributions: Conceptualization, G.B., and P.G.; methodology, G.B., and R.C. and P.G. and M.R.; software, G.B. and M.R.; validation, G.B., and M.R.; formal analysis, R.C. and P.G.; investigation, G.B., and M.R.; resources, R.C. and P.G.; data curation, G.B., and M.R.; writing—original draft preparation, G.B., M.R. and R.C.; writing—review and editing, G.B., M.R., R.C., and P.G.; visualization, G.B., and M.R.; supervision, R.C. and P.G.; project administration, R.C. and P.G. All authors have read and agreed to the published version of the manuscript.

Acknowledgments: We thank the anonymous reviewers for helping us to improve our paper. This research has been partially carried on in the context of the Visiting Professor Program of the Italian Istituto Nazionale di Alta Matematica (INdAM).

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: https://github.com/tuetschek/e2e-dataset (accessed on 22 March 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Puduppully, R.; Dong, L.; Lapata, M. Data-to-Text Generation with Content Selection and Planning. In Proceedings of 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, HI, USA, 27 January–1 February 2019; AAAI Press: Palo Alto, CA, USA, 2019; pp. 6908–6915. [CrossRef]
2. Dusek, O.; Novikova, J.; Rieser, V. Evaluating the state-of-the-art of End-to-End Natural Language Generation: The E2E NLG challenge. *Comput. Speech Lang.* **2020**, *59*, 123–156. [CrossRef]
3. Otter, D.W.; Medina, J.R.; Kalita, J.K. A Survey of the Usages of Deep Learning in Natural Language Processing. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 604–624. [CrossRef] [PubMed]
4. Moryossef, A.; Goldberg, Y.; Dagan, I. Step-by-Step: Separating Planning from Realization in Neural Data-to-Text Generation. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019; Volume 1 (Long and Short Papers); Burstein, J., Doran, C., Solorio, T., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 2267–2277. [CrossRef]
5. Mei, H.; Bansal, M.; Walter, M.R. What to talk about and how? Selective Generation using LSTMs with Coarse-to-Fine Alignment. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2016), San Diego, CA, USA, 12–17 June 2016; Knight, K., Nenkova, A., Rambow, O., Eds.; The Association for Computational Linguistics: Stroudsburg, PA, USA, 2016; pp. 720–730.
6. Gatt, A.; Krahmer, E. Survey of the State of the Art in Natural Language Generation: Core tasks, applications and evaluation. *J. Artif. Intell. Res.* **2018**, *61*, 65–170. [CrossRef]
7. Lebret, R.; Grangier, D.; Auli, M. Neural Text Generation from Structured Data with Application to the Biography Domain. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, TX, USA, 1–4 November 2016; pp. 1203–1213. [CrossRef]
8. Wiseman, S.; Shieber, S.M.; Rush, A.M. Challenges in Data-to-Document Generation. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, 9–11 September 2017; Palmer, M., Hwa, R., Riedel, S., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2017; pp. 2253–2263. [CrossRef]
9. Cho, K.; van Merrienboer, B.; Gülçehre, Ç.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv* **2014**, arXiv:1406.1078.
10. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015.
11. Luong, T.; Pham, H.; Manning, C.D. Effective Approaches to Attention-based Neural Machine Translation. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, 17–21 September 2015; Màrquez, L., Callison-Burch, C., Su, J., Pighin, D., Marton, Y., Eds.; The Association for Computational Linguistics: Stroudsburg, PA, USA, 2015; pp. 1412–1421. [CrossRef]
12. Gu, J.; Lu, Z.; Li, H.; Li, V.O.K. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, (ACL 2016), Berlin, Germany, 7–12 August 2016; Volume 1: Long Papers; Erj, K., Smith, N.A., Eds.; The Association for Computer Linguistics: Stroudsburg, PA, USA, 2016.

13. See, A.; Liu, P.J.; Manning, C.D. Get To The Point: Summarization with Pointer-Generator Networks. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017), Vancouver, BC, Canada, 30 July–4 August 2017; Volume 1: Long Papers; The Association for Computer Linguistics: Stroudsburg, PA, USA, 2017; pp. 1073–1083. [CrossRef]
14. Chan, Z.; Chen, X.; Wang, Y.; Li, J.; Zhang, Z.; Gai, K.; Zhao, D.; Yan, R. Stick to the Facts: Learning towards a Fidelity-oriented E-Commerce Product Description Generation. In 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, 3–7 November 2019; Inui, K., Jiang, J., Ng, V., Wan, X., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 4958–4967. [CrossRef]
15. Elder, H.; Foster, J.; Barry, J.; O'Connor, A. Designing a Symbolic Intermediate Representation for Neural Surface Realization. *arXiv* **2019**, arXiv:1905.10486. [CrossRef]
16. Nie, F.; Wang, J.; Yao, J.; Pan, R.; Lin, C. Operation-guided Neural Networks for High Fidelity Data-To-Text Generation. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; Riloff, E., Chiang, D., Hockenmaier, J., Tsujii, J., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2018; pp. 3879–3889. [CrossRef]
17. Liu, T.; Wang, K.; Sha, L.; Chang, B.; Sui, Z. Table-to-Text Generation by Structure-Aware Seq2seq Learning. In 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, LA, USA, 2–7 February 2018; pp. 4881–4888.
18. Liu, T.; Luo, F.; Xia, Q.; Ma, S.; Chang, B.; Sui, Z. Hierarchical Encoder with Auxiliary Supervision for Neural Table-to-Text Generation: Learning Better Representation for Tables. In Proceedings of 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, HI, USA, 27 January–1 February 2019; AAAI Press: Palo Alto, CA, USA, 2019; pp. 6786–6793. [CrossRef]
19. Rebuffel, C.; Soulier, L.; Scoutheeten, G.; Gallinari, P. A Hierarchical Model for Data-to-Text Generation. In Proceedings of the Advances in Information Retrieval—42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, 14–17 April 2020; Proceedings, Part I; Jose, J.M.; Yilmaz, E.; Magalhães, J.; Castells, P.; Ferro, N.; Silva, M.J.; Martins, F., Eds.; Springer: Berlin/Heidelberg, Germany, 2020; Lecture Notes in Computer Science (LNCS), Volume 12035, pp. 65–80. [CrossRef]
20. Wen, T.; Gasic, M.; Mrksic, N.; Su, P.; Vandyke, D.; Young, S.J. Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, 17–21 September 2015; Màrquez, L., Callison-Burch, C., Su, J., Pighin, D., Marton, Y., Eds.; The Association for Computational Linguistics: Stroudsburg, PA, USA, 2015; pp. 1711–1721.
21. Sutskever, I.; Martens, J.; Hinton, G.E. Generating Text with Recurrent Neural Networks. In Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, DC, USA, 28 June–2 July 2011; Getoor, L., Scheffer, T., Eds.; Omnipress: Madison, WI, USA, 2011; pp. 1017–1024.
22. Goyal, R.; Dymetman, M.; Gaussier, É. Natural Language Generation through Character-based RNNs with Finite-state Prior Knowledge. In Proceedings of the COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, Osaka, Japan, 11–16 December 2016; Calzolari, N., Matsumoto, Y., Prasad, R., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2016; pp. 1083–1092.
23. Babić, K.; Martinčić-Ipšić, S.; Meštrović, A. Survey of Neural Text Representation Models. *Information* **2020**, *11*, 511. [CrossRef]
24. Vinyals, O.; Fortunato, M.; Jaitly, N. Pointer Networks. In Proceedings of the Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, Montreal, QC, Canada, 7–12 December 2015; Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R., Eds.; Curran Associates: San Diego, CA, USA, 2015; pp. 2692–2700.
25. Qader, R.; Portet, F.; Labbé, C. Seq2SeqPy: A Lightweight and Customizable Toolkit for Neural Sequence-to-Sequence Modeling. In Proceedings of the 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, 11–16 May 2020; Calzolari, N., Béchet, F., Blache, P., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Isahara, H., Maegaard, B., Mariani, J., et al., Eds.; European Language Resources Association: Paris, France, 2020; pp. 7140–7144.
26. Wiseman, S.; Shieber, S.M.; Rush, A.M. Learning Neural Templates for Text Generation. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; Riloff, E., Chiang, D., Hockenmaier, J., Tsujii, J., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2018; pp. 3174–3187. [CrossRef]
27. Su, S.; Huang, C.; Chen, Y. Dual Supervised Learning for Natural Language Understanding and Generation. In Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, 28 July–2 August 2019; Volume 1: Long Papers; Korhonen, A., Traum, D.R., Màrquez, L., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 5472–5477. [CrossRef]
28. Su, S.; Chen, Y. Investigating Linguistic Pattern Ordering In Hierarchical Natural Language Generation. In Proceedings of the 2018 IEEE Spoken Language Technology Workshop, SLT 2018, Athens, Greece, 18–21 December 2018; pp. 779–786. [CrossRef]
29. Puzikov, Y.; Gurevych, I. E2E NLG Challenge: Neural Models vs. Templates. In Proceedings of the 11th International Conference on Natural Language Generation, Tilburg University, Tilburg, The Netherlands, 5–8 November 2018; Krahmer, E., Gatt, A., Goudbeek, M., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2018; pp. 463–471. [CrossRef]

30. Smiley, C.; Davoodi, E.; Song, D.; Schilder, F. The E2E NLG Challenge: A Tale of Two Systems. In Proceedings of the 11th International Conference on Natural Language Generation, Tilburg University, The Netherlands, 5–8 November 2018; Krahmer, E., Gatt, A., Goudbeek, M., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2018; pp. 472–477. [CrossRef]
31. Novikova, J.; Dusek, O.; Rieser, V. The E2E Dataset: New Challenges For End-to-End Generation. In Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, Saarbrücken, Germany, 15–17 August 2017; Jokinen, K., Stede, M., DeVault, D., Louis, A., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2017; pp. 201–206.
32. Burke, R.D.; Hammond, K.J.; Young, B.C. The FindMe Approach to Assisted Browsing. *IEEE Expert* **1997**, *12*, 32–40. [CrossRef]
33. Papinemi, K.; Roukos, S.; Ward, T.; Zhu, W. Bleu: a Method for Automatic Evaluation of Machine Translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, USA, 6–12 July 2002; Association for Computational Linguistics: Stroudsburg, PA, USA, 2002; pp. 311–318.
34. Doddington, G. Automatic Evaluation of Machine Translation Quality Using N-gram Co-occurrence Statistics. In Proceedings of the Second International Conference on Human Language Technology Research (HLT '02), San Diego, CA, USA, 24–27 March 2002; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2002; pp. 138–145.
35. Banerjee, S.; Lavie, A. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, Ann Arbor, MI, USA, 29 June 2005; Association for Computational Linguistics: Ann Arbor, MI, USA, 2005; pp. 65–72.
36. Lin, C.Y. ROUGE: A Package for Automatic Evaluation of summaries. In Proceedings of the ACL workshop on Text Summarization Branches Out, Barcelona, Spain, 25–26 July 2004, p. 10.
37. Vedantam, R.; Zitnick, C.L.; Parikh, D. CIDEr: Consensus-based image description evaluation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, 7–12 June 2015; IEEE Computer Society: Washington, DC, USA, 2015; pp. 4566–4575. [CrossRef]
38. Agarwal, S.; Dymetman, M. A surprisingly effective out-of-the-box char2char model on the E2E NLG Challenge dataset. In Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, Saarbrücken, Germany, 15–17 August 2017; Jokinen, K., Stede, M., DeVault, D., Louis, A., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2017; pp. 158–163.
39. Dusek, O.; Novikova, J.; Rieser, V. Findings of the E2E NLG Challenge. In Proceedings of the 11th International Conference on Natural Language Generation, Tilburg University, The Netherlands, 5–8 November 2018; Krahmer, E., Gatt, A., Goudbeek, M., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2018; pp. 322–328. [CrossRef]
40. Dusek, O.; Jurcícek, F. Sequence-to-Sequence Generation for Spoken Dialogue via Deep Syntax Trees and Strings. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, Berlin, Germany, 7–12 August 2016; Volume 2: Short Papers; The Association for Computer Linguistics: Stroudsburg, PA, USA, 2016. [CrossRef]
41. Williams, R.J.; Zipser, D. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Comput.* 1989, *1*, 270–280. [CrossRef]
42. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015.
43. Loshchilov, I.; Hutter, F. SGDR: Stochastic Gradient Descent with Warm Restarts. In Proceedings of the 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April 2017.
44. Pascanu, R.; Mikolov, T.; Bengio, Y. On the difficulty of training recurrent neural networks. In Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16–21 June 2013.