Dealing with hallucination and omission in neural Natural Language Generation: A use case on meteorology

Anonymous ACL submission

Abstract

Hallucinations and omissions need to be carefully handled when using neural models for performing Natural Language Generation tasks. In the particular case of data to text applications, neural models are usually trained on large-scale datasets and sometimes generate text with divergences in respect to the data input. In this paper, we show the impact of the lack of domain knowledge in the generation of texts containing input-output divergences through a use case on meteorology. We propose a novel approach for the detection of hallucinations and omissions when using neural models for automatic generation of meteorological descriptions from tabular data. Main contributions are: (i) we provide the research community in Natural Language Generation with new resources (dataset and corpus curated by meteorologists); (ii) we explain how to adapt a Transformer-based model to generate meteorological texts from tabular data; and (iii) we explain how to detect divergences (i.e., hallucinations and omissions) between the output texts and the input data, regarding common sense knowledge.

1 Introduction

Since the emergence of Natural Language Generation (NLG), this subfield of Natural Language Processing (NLP) has not stopped to evolve. However, the fastest evolution has occurred in the last years, due to the advances made in Deep Learning models. With the arrival of the attention mechanism and the Transformer-based models (e.g., BERT [Devlin et al., 2019], GPT-2 [Radford et al., 2019], GPT-3 [Brown et al., 2020]), the way in which NLG tasks such as text summarization, question answering, or data to text (D2T) are approached has changed drastically. Before the appearance of these end-to-end neural models, the generation of NLG models had at least two main tasks to accomplish (content selection and surface realization) (Reiter and Dale, 1997). Now, with end-to-end models, the whole generation process is made in a single step. Furthermore, neural models allow us to obtain natural, diverse, and fluent texts. Of course, these models also have their drawbacks, such as the necessity of a large corpus or enough computational resources to train the model for a given task.

In addition, in the context of D2T, texts generated by neural models are sometimes affected by divergences with the input data (Dusek et al., 2019). On the one hand, neural models may generate texts that are incoherent or unrelated with the input of the D2T system, i.e., hallucinations. On the other hand, generated texts may not mention some (relevant) information from the input data, i.e., omissions. Despite recent efforts for minimizing the appearance of these undesired divergences (Nie et al., 2019; Dušek and Kasner, 2020), further research is needed to deal properly with them when building neural models for D2T systems.

The first step to minimize hallucination and omission on neural models is to detect them. The task of detection requires checking for each generated text if its content matches with the input provided to the model. But it depends on the task for which the model has been designed. The input of an NLG system can be provided in different forms (e.g., structured meaning representation, images, tabular data, or text). In this paper, we focus our research in the detection of hallucinations and omissions when performing a D2T task in which the input is tabular data. Accordingly, we must analyze the content of a generated text, extract its meaning, and then check the consistency or divergence with respect to the data table that was provided as input to the generation system.

Performing this task manually is tough and costly, in terms of time and human resources. Nevertheless, due to the variety and diversity of the texts generated by neural models, sometimes a
**Input data:** Table-AXYZ

<table>
<thead>
<tr>
<th>Zone</th>
<th>Morning</th>
<th>Afternoon</th>
<th>Night</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z01</td>
<td>weak showers</td>
<td>showers</td>
<td>weak showers</td>
</tr>
<tr>
<td>Z02</td>
<td>weak showers</td>
<td>showers</td>
<td>weak showers</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Z03</td>
<td>weak showers</td>
<td>showers</td>
<td>clouds and clearings</td>
</tr>
</tbody>
</table>

**Generated text:** The skies are expected to be cloudy with intermittent showers, occasionally stormy and accompanied by hail, more frequent in the morning.

**Reference text:** Skies will remain partly cloudy with showers, more frequent in the west.

Figure 1: Illustrative example of divergence between input data and output text of a neural D2T system. The hallucinated content is highlighted in red.

Fully automatic detection does not work properly because of context dependencies, ambiguity, or domain-specific language that only humans can understand.

Our focus is on an end-to-end D2T system for meteorology. Let us introduce an example in the context of a use case on meteorology. We can see in Fig. 1 a case of hallucinated content. The generated text refers to “hail” although there is no evidence of hailstone anywhere in the data. Thus, the generated text includes content which is not present in the input data. However, when we asked a meteorologist to rate the severity of this hallucination, he rated it as acceptable because “when there is rainy weather in the whole region there is a chance for occasional hail in some locations”. This type of cases highlights the importance of considering explicit domain knowledge, something that neural models are not able to achieve by themselves, since they only operate with the provided data. The main contributions in this work are:

1. A dataset which includes a clean Spanish corpus of meteorological texts for D2T. It is made up of 3,033 state of the sky descriptions made by meteorologists, along with the corresponding tabular data for each described situation.

2. An adaption of a Transformer-based model to generate weather descriptions in Spanish from tabular data.

3. A novel approach for detecting divergences (i.e., hallucinated and omitted content) between input and output of end-to-end neural D2T systems. All detected hallucinations in our use case have been subsequently validated by a meteorologist.

The rest of the manuscript is organized as follows. Section 2 introduces related work. Section 3 presents the new dataset. Section 4 presents our proposal of neural D2T system. Section 5 presents our approach for detecting hallucinations and omissions. Finally, Section 6 concludes the paper with some final remarks and points out future work.

## 2 Background

### 2.1 Data-to-text

One of the most popular and complete books on NLG, centered on D2T, was published by Reiter and Dale (1997). But, since the publication of this pioneering book, new methods have been developed in the field of NLG and, in particular, in the D2T subfield. Nowadays, rule-based or template-based systems tend to be replaced by deep learning models derived from the Machine Learning field, as described by Gatt and Krahmer (2018). Traditionally, NLG had to address, at least, two main tasks (usually addressed independently): the content selection, i.e., selecting the appropriate pieces of information to include in the final narrative; and the surface realization, i.e., communicating the selected information in the right format. However, end-to-end models are capable of addressing the whole generation pipeline at once, thus generating more complex outputs than traditional models while learning lexical and syntactic richness from large corpus and associated datasets.

Moreover, the appearance of the attention mechanism and the Transformer architecture (Vaswani et al., 2017) revolutionized both NLP and NLG fields. Even though, initially, Transformer models were used mainly for NLP tasks (e.g., question-answering or summarization) and text-to-text generation, their use in the context of D2T has also increased during last years (Devlin et al., 2019; Radford et al., 2019; Raffel et al., 2019).

In this paper, we focus on a subtask of D2T, named table-to-text, which aims to produce textual descriptions from an input in the form of structured tabular data. Recently, some end-to-end models were proposed to accomplish this task. For example, Puduppully et al. (2019a) designed and developed a neural model which creates entity-specific representations, avoiding treating entities as simple vocabulary tokens. In addition, Gong et al.
Although end-to-end NLG models usually produce texts which is characterized by fluency and naturalness, the fidelity to data of such text is sometimes arguable. Some generated texts mention false information, information that is not in the data, or simply ignore some relevant data. These phenomena, in many cases, are not acceptable (e.g., generation of medical or financial reports) and in many others become simply unpleasant or useless for the user (e.g., a virtual hotel advisor that gives you false information or omits good deals) what jeopardize trust and credibility.

Accordingly, there has been an effort to propose novel methods to detect and minimize negative effects associated to hallucinations and/or omissions, and that way contributing to a more responsible NLP. Some studies showed how semantic noise in training data may lead neural models to divergence between input and output, either in the form of omissions or hallucinations (Dusek et al., 2019). Thus, some authors (Wang, 2019; Nie et al., 2019) proposed to reduce noise in training data with the aim of producing more consistent texts, while maintaining good fluency. In addition, Rebuffel et al. (2021) opted for enhancing the neural models instead of cleaning the datasets: they proposed the use of a decoder to leverage word-level labels and to learn relevant parts of each data instance. In the context of text-to-text summarization, Feijo and Moreira (2021) proposed first the creation of different “views” of the source text and then the selection of those candidate summaries which were more faithful to the source.

Notice that, all the proposals mentioned above are aimed to reducing the apparition of divergence between input and output for a given dataset and specific model. Nevertheless, if we want to address the problem in a general way, we must address first the detection and classification of divergences. Then, we may select the right way to deal properly with each case of hallucination or omission. Accordingly, Maynez et al. (2020) carried out a thorough analysis on different types of hallucination in the context of summarization. Human annotators read multiple summaries and identified both intrinsic (i.e., manipulating the information obtained from the input) and extrinsic (i.e., adding information beyond the one directly inferred from the input) hallucinations. This analysis reveals the dimension of the problem which affects not only to summarization but also to all tasks related to end-to-end NLG neural models.

In addition, Dušek and Kasner (2020) presented a metric for evaluating D2T semantic accuracy based on Natural Language Inference. This metric detects both hallucinations and omissions automatically, but it is only for tasks where no content selection is required. Furthermore, there are some cases in which an automatic metric is not faithful enough to analyze the goodness of texts (e.g., context dependencies or domain-specific vocabulary) and complementary human evaluation is required.

In this paper, we show how to automatically detect both hallucination and omission in texts generated by an end-to-end D2T Transformer-based neural model in the context of meteorology. It is worth noting that detected divergences between texts and data were afterwards validated by a meteorologist.

3 The WEATHER-ES Dataset

Weather forecasting is a popular topic in the D2T research field. There are some well-known datasets. For example, SUMTIME (Sripada et al., 2008) and WEATHERGOV (Liang et al., 2009). Here, we introduce a new dataset (WEATHER-ES) which is
made up of 3,033 records of meteorological tabular data along with hand-written textual descriptions in Spanish. Notice that, the dataset comprises real data and texts written by meteorologists. It was provided by METEO, the Official Meteorological Agency of X₁.

### 3.1 Data tables

The data contained in WEATHER-ES represent the state of the sky by categorical values (e.g., “sunny”, “clouds”, “rain”, “fog”, etc.). The data is saved in a database and organized in the form of different instances, each one composed by a table divided into 4 columns and 32 rows. The first column indicates the geographical zone of interest in X, which covers a group of councils. The other columns contain a value for each period of the day (morning, afternoon and night). This way, we have 3 state of the sky values for each of the different 32 zones in X, i.e., 96 (3 × 32) values per table.

All in all, in agreement with METEO’s Style Guide, there are 19 different possible values for the state of the sky, such as “rainy”, “high clouds”, “clear”, etc. Unfortunately, being real data, the distribution of these data values is not homogeneous in the entire region. Therefore, in order to provide readers with useful and meaningful statistics, we have grouped the 19 possible values into 6 main categories regarding similar weather events, which are ranked in terms of their coverage of the dataset. We considered only those events which are in METEO’s Style Guide. Each one of these events is represented in maps by a single specific icon, while textual descriptions admit some variety in the form of a list of admitted synonyms.

1. **Cloud**: it contains the four events that involve any type of clouds: (1.1) “clouds and clearings”, (1.2) “clouds”, (1.3) “high clouds” and (1.4) “covered”. This is by far the main category which covers a 47.3% of the data values, i.e., nearly the half of the cases in the dataset are related with events including clouds.

2. **Rain**: it contains the six events that involve water dropping: (2.1) “weak rains”, (2.2) “showers”, (2.3) “rain”, (2.4) “weak showers”, (2.5) “drizzle” and (2.6) “cloudy with showers”. This category is associated with the 27.6% of cases in the dataset.

3. **Clear**: it contains only the value (3.1) “clear”, i.e., which applies when there is no more than sun in the sky. This category represents the 21.5% of cases in the dataset.

4. **Snow**: it contains four events which involve frozen water: (4.1) “snow showers”, (4.2) “snow”, (4.3) “hail” and (4.4) “sleet”. This category only covers the 1.7% of cases in the dataset.

5. **Fog**: it contains three events which involve visibility reduction: (5.1) “fog”, (5.2) “fog banks” and (5.3) “mist”. Only 1.6% of cases are in this category.

6. **Storm**: it contains only the value (6.1) “storm”, i.e., which applies when electrical events (thunder and lightning) appear in the sky. This is by far the more underrepresented category, with only 0.3% of cases.

It is also worth noting that some state of the sky values do not appear repeatedly in the same data instance. For example, the “snow” value appears only in specific zones of the region, i.e., in a particular cell of the data table. In addition, if we only take into account the single apparitions of the values in each data table (i.e., if a value appears more than once in an instance, it counts only as one) the computed statistics are quite different from the introduced above. In the 93.74% of data tables, there is at least one reference to the **Cloud** category. This means that almost all the meteorological situations from the dataset include weather phenomena involving clouds. The second most common category is **Rain**, with the 68.84% of the records referring to some rain phenomena. In addition, the **Clear** category covers nearly the half of the tables (47.25%) and the **Fog** category covers the 40.45% of tables. **Snow** and **Storm** are the most underrepresented categories, covering 14.41% and 8.41% of tables, respectively.
As we can see, the weather categories in WEATHER-ES are unbalanced, some categories are overrepresented (e.g., Cloud and Rain) while others (e.g., Snow and Storm) are underrepresented. This is due to the fact that we are dealing with real data which were collected from 2010 to 2020, so they provide us with a complete picture of the weather in the region under consideration.

### 3.2 Texts

Associated to each data table, there is a textual description written by a meteorologist. All in all, there are 3,033 short meteorological descriptions of the state of the sky made by experts in the field. Each description was cleaned and cured, correcting common punctuation or spelling typos. The length of the texts is variable, from a minimum of 25 characters until a maximum of 557 characters. The average length of the descriptions is 186 characters, while the standard deviation is 71.

We also made a deeper analysis of the collected texts, taking into account the type of textual references that they include. We considered both value references and spatial references. Value references match a state of the sky value (e.g., “fogs”, “rain”, “hail”, etc.), while spatial references determine where a weather phenomenon takes place (e.g., “coast” vs “inland”, or “north” vs “south”).

In order to detect these two types of reference, we performed different searching methods based on the METEO’s Style Guide. This guide contains the vocabulary which is allowed to refer to each weather phenomenon, and also the correct spatial references to name each zone in the map. This way, we created a dictionary with all potential expressions used by meteorologists when referring to zones and state of the sky values. As a result of our analysis, we found out that in each text there are in average 2.53 value references and 1.66 spatial references. As expected, since texts in the corpus describe the state of the sky of a day in X, we have more value references than spatial ones. Having between two and three value references per text means that data tables and descriptions are well aligned. It must be also highlighted the presence of above 1.5 spatial references in each text.

Additionally, we performed an analysis over temporal references, i.e., expressions that determine when a phenomenon occurs. In this case, we could not trust on the vocabulary established by the METEO’s Style Guide because it does not say anything about temporal references. Therefore, we performed a preliminary ad-hoc search of simple expressions (e.g., morning, afternoon, night). Following this naive approach, we discovered in average about 1.07 temporal references in each text. Taking into account that we have probably overlooked some temporal expressions and therefore underestimated their presence in the dataset, we think they are likely to play a relevant role in the detection of hallucinations and/or omissions, and we will address this important issue in future work.

### 4 Data-to-text generation

This section describes an end-to-end D2T neural model which is trained with the WEATHER-ES dataset previously introduced. Instead of designing a D2T system from scratch, we have reused the architecture of an existing Transformer-based model (Obeid and Hoque, 2020) which is carefully modified to be effective in our use case: generation of textual descriptions from tabular meteorological data. Nevertheless, it is worth noting that for the purpose of this paper we do not need building the best (or a very good) D2T system for the given use case. Since our ultimate goal, which will be carefully addressed in the next section, is testing a novel approach for automated detection of hallucinations and omissions, having a D2T which performs perfectly free of divergences between inputs and outputs would make our detector useless.

In the rest of this section, we first describe the Transformer-based architecture that is taken as base model. Then, we go in detail about how it has been reused, enhanced, trained and tested with the WEATHER-ES dataset for generating weather forecasts in Spanish.

#### 4.1 Base model

We took as starting point the Chart-to-text model (Obeid and Hoque, 2020). Given a chart and its title, this model describes the data embedded and depicted in the chart. Chart-to-text extends another previous Transformer-based D2T model (Gong et al., 2019b) in the following way: (i) Chart-to-text passes from input tuples to input records, which facilitate the addition of contextual information to the D2T system; (ii) Chart-to-text reintroduces positional embeddings as defined in the pioneering Transformer-based models for machine translation (Vaswani et al., 2017); and (iii) Chart-to-text can be fed with both numerical and
categorical data values. These extensions are well aligned with our purposes because (i) we deal with more than the four values per tuple which were allowed by the original model; (ii) weather forecasting requires dealing with ordered/temporal relationships; and (iii) we have categorical values, such as the state of the sky for each zone in X (see the categories that we introduced in Section 3).

Additionally, the Chart-to-text base model includes a pre-processing stage initially thought for minimizing overfitting of the model but which can be seen as a very naive way for minimizing hallucinations, as we will see in the next section. More precisely, before training the model, the gold summaries in the corpus are pre-processed as follows: each token that refers to a value included either in the data table or in the chart title is replaced by a predefined label. This way, the model learns to generate more generic template-based summaries.

4.2 Our approach

Due to the nature of the data in WEATHER-ES, we had to carry out several modifications on the base model with the aim of making it operative. First, our corpus is in Spanish while the base model was thought for being trained with a corpus in English. Second, the WEATHER-ES dataset comes from the specific field of meteorology, while the base model was aimed for describing generic charts from any field. In the rest of this section we explain in detail, step by step, how we have recycled and extended the base model.

4.2.1 Input data and pre-processing

Since we are dealing with tabular data, we maintain the base format. In the base model, each chart came with a data table and a brief title, which was taken into account when generating the related texts. In our case, each table comprises all available meteorological data for one given day, i.e., it includes categorical values associated to the state of the sky for each zone in X and period of the day (morning, afternoon, night). We also added a generic title (“Weather forecast of a day in X, by period of the day”) to each data table. This way, the D2T system can extract relevant tokens from the title during text generation. Notice that, each data table and title have attached the textual description in Spanish, which was handwritten by a meteorologist.

In the data pre-processing stage, our model had to be pre-trained to identify relevant tokens in Spanish before being ready to use them properly in the text generation stage. Similarly to Chart-to-text, we applied named entity recognition (Manning et al., 2014) to WEATHER-ES with the aim of extracting important information from the given descriptions and titles associated to each data table.

4.2.2 Training and validation

Regarding the training and validation stages, we reused the architecture of the base neural model (Obeid and Hoque, 2020) with some variations in the parameters. We first randomized all the WEATHER-ES instances and then used the 70% of them for training, 15% for validation and 15% for testing. The model was trained for 10 epochs with an epoch size of 1000, a dropout rate of 0.1, using 1 encoder layer, 6 decoder layers, embedding size of 512, batch size of 6, and beam size of 4. We used the hyperparameter values recommended by Chart-to-text without additional hyperparameter search. The model was trained on a GeForce RTX-2080 machine. The whole training took around 30 minutes. Once the model was trained, we obtained the so-called gold templates, which are to be filled in on the fly during the testing stage.

4.2.3 Testing and post-processing

In the testing stage, the pre-trained model was provided only with the testing tabular data, and it was able to generate the final texts by filling in the previously generated templates. In the base model, each label in a gold template referred directly to a single value in the data table or to a single word in the title. Accordingly, filling in the given templates was straightforward. In our case, labels in templates are directly replaced by the given values only if the labels refer to values in the title. Otherwise, the BETO model (Cañete et al., 2020) is applied to fill in the gap. This model is a Spanish version of the BERT model (Devlin et al., 2019) which replaces each label referring to tabular data with the best word from a set of candidates which includes the values in the corresponding category of data values. This way we improve naturalness while ensuring that gaps in templates are filled only with words that match the context of the sentence, thus minimizing typos as well as syntactic errors in surface realization. Finally, we run a post-processing step for polishing the generated texts and fixing some writing and/or concordance errors (e.g., fixing the use of capital letters after a full stop, verifying concordance of words in gender and number, removing repetitions of words, etc.).
5 Hallucination and Omission detector

This section first introduces and then validates our proposal for detecting hallucinations and omissions in texts generated by neural D2T systems. While the proposed approach is generic, it is validated in the meteorology use case we are considering.

The divergence detector is a software application composed of two independent parts, one for detecting each type of divergence. On the one hand, the omission detection part works as follows: it looks first at the table with input data values (i.e., identifies all state of the sky values which apply to the case under consideration) and then checks if all these values are mentioned in the generated text. The detector counts as omission each value which is in the input data but is not explicitly referred to in the output text. On the other hand, the hallucination detection part follows the other way around. It looks first to the output text, identifies all data values which are mentioned in the text, and then checks if they are also included in the related input data. The detector counts as hallucination each value which is mentioned in the output text but is not present in the input data.

It is worth noting that the current detector only looks for exact values, i.e., synonyms are not taken into consideration during the detection stage, what we are aware is a limitation of the present proposal to be addressed as future work. With the aim of evaluating the goodness of our proposal, we have validated the divergence detector with all the 272 unseen cases in the test set which was introduced in the previous section.

5.1 Reported omissions

Making use of our detector, we found that the number of omissions detected was very high. We identified omissions in 160 out of 272 texts (58%). This result shows how frequent omissions are in texts generated by neural models. However, further research is needed to assess how many of those omissions are admissible, and then refining accordingly our detector with the aim of reporting only those omissions that are more likely to be negatively perceived by humans. Indeed, omissions are naturally used by humans (as well as by traditional non-neural NLG systems) and they may be sometimes well appreciated because of producing shorter texts which only mention the most relevant pieces of information (as traditional NLG systems do thanks to the explicit stage of content determination).

For example, the data table associated to a given case includes the value “high clouds” while the output text refers to “open skies will prevail”. Formally speaking, this case counts as an omission because “high clouds” are not explicitly mentioned in the text. However, it should not be because the generated text is considered valid by the meteorologist since it makes sense and conveys the correct information. This kind of cases are easily evaluated by humans, but really hard to be identified correctly by an automatic detector.

In order to identify which omissions could be admissible for humans and therefore should not be reported as unacceptable by our detector, we asked a meteorologist to analyze in detail a group of randomly selected cases among the detected omissions. He confirmed that many of them were admissible because in the context of meteorology missing some information is not so severe as it may be in other application domains. In fact, in some cases, the meteorologist preferred certain omission to the exhaustive verbalization of all the values in the data table what could lead to a long, verbose, repetitive and less natural text.

It is worth noting that our results are in agreement with those reported by related work in which a similar analysis was done. For example, Dusek et al. (2019) and Nie et al. (2019) also reported many omissions when analyzing the content coverage of texts generated by neural models. They also noted that forcing the model to verbalize all slots during training leads to fewer omissions but at the cost of producing longer texts.

5.2 Reported hallucinations

The texts generated by our model describe meteorological situations in a geographical region, but the handwritten reference texts sometimes describe the state of the sky of specific zones inside the whole region X, e.g., “Skies will be cloudy in the Atlantic coast”. Considering this, if our model generates a text in which a state of the sky value is associated to a wrong zone (e.g., following the previous example, there are no clouds in the Atlantic coast), it must be considered also a case of hallucination.

Accordingly, we analyzed two different levels of hallucination: basic hallucinations and spatial hallucinations. The former do not take into account the geographical location of the values in the map, while the latter do. Once again, we followed the METEO’s Style Guide, with the aim of identifying
the list of admissible spatial references along with their related locations in the map. As a result, we identified 48 different reference expressions that meteorologist may use to refer to different geographical zones in X.

The detector identified 35 basic hallucinations and 11 spatial hallucinations out of all the 272 texts under study. In order to assess the goodness of our detector and determine if these hallucinations were really worthy to note, we asked once again the assistance of a meteorologist. He rated the degree of relevance of each detected hallucination in a 3-points Likert scale (admissible, partially admissible, inadmissible). Surprisingly, 12 (10 basic and 2 spatial hallucinations) out of all the 46 detected hallucinations were deemed admissible. Formally speaking, all these 12 cases were strict hallucinations (i.e., the state of the sky values mentioned in the output text were not present in the input data) but, according to the meteorologist’s background and in agreement with contextual information and common-sense reasoning, they were admissible.

Figure 1 shows an example of admissible hallucination. Even if according to the strict data checking done by our detector this is a case of hallucination, the meteorologist rated it as admissible due to the observed situation in the whole region, which according to his experience justifies a very high possibility of hail. It is also worthy to note that in four of the hallucinations rated as admissible by the meteorologist, the reference texts in the corpus also mentioned some values which were not in the data. For example, in one of the cases, both reference and model texts mention “storm with hail” while in the associated data there are only “storm” values. This suggests that it may be a good thing to use our detector as part of the pre-processing stage for automatically identifying and fixing similar cases that are likely to be included in the training set. We will address this challenging task in the near future.

6 Final Remarks and Future Work

In this paper, we first introduced a new dataset (WEATHER-ES) for D2T in the application domain of meteorology. Then, we reused and adapted a neural D2T system to generate weather descriptions from WEATHER-ES. Finally, we described how to automatically detect and validate hallucinations and/or omissions in the texts generated by the D2T system previously trained.

In the light of the reported results, we can draw the following important remarks. First of all, neural D2T systems, after being trained with large-scale datasets, can generate natural and fluid texts, but more often than not the generated texts provide unfaithful information or inconsistencies with respect to the input data, mainly in the form of omissions and/or hallucinations. In our specific use case, we detected more omissions than hallucinations, but in general hallucinations were more negatively perceived and deemed as misleading by the meteorologist who assisted us in the validation stage. Notice that the observed divergence between input and output in some controversial cases is likely to be due to the lack of ability of the designed D2T system to deal with contextual information and common-sense reasoning as humans naturally do.

In addition, we must take into account that in practice, meteorologists rely on contextual information and common-sense reasoning beyond input data when writing weather forecasts. Current neural D2T systems can not capture such a general knowledge because they are only guided by the given training data. This means that for truly complex tasks, where either omissions or hallucinations may be critical, neural models have to be endowed and integrated with other knowledge sources different from data, if we want them to achieve high quality automatically generated texts which are as correct as expert-made ones.

Last but not least, the high level of naturalness and fluidity that neural D2T systems usually achieve may raise too high expectations in end users, who may be frustrated when discovering some misleading pieces of information. We claim that providing users with the generated texts and the findings of our detector contributes to lower expectations in the sense that we make explicit limitations and undesired behaviors of the underlying D2T system. This way, we contribute to a more responsible NLP.

As future work, we plan in the midterm to enrich our neural D2T system with a knowledge base including meteorological facts (regarding both spatial and temporal references) but also in the long-term with temporal knowledge. As a result, we expect to improve both text generation and hallucination/omission detection. Moreover, we will go deeper with understanding how classical NLG approaches for content determination can help to identify relevant omissions.
References


Heng Gong, Xiaocheng Feng, Bing Qin, and Ting Liu. 2019a. Table-to-text generation with effective hierarchical encoder on three dimensions (row, column and time). CoRR, abs/1909.02304.


