Semi-Supervised Neural Text Generation by Joint Learning of Natural Language Generation and Natural Language Understanding Models

Raheel Qader[1] **François Portet**[2] **Cyril Labbé**[2]
Univ. Grenoble Alpes, LIG
38000 Grenoble, France
[1]raheel.qader@univ-grenoble-alpes.fr
[2]{francois.portet, cyril.labbe}@imag.fr

Abstract

In Natural Language Generation (NLG), End-to-End (E2E) systems trained through deep learning have recently gained a strong interest. Such deep models need a large amount of carefully annotated data to reach satisfactory performance. However, acquiring such datasets for every new NLG application is a tedious and time-consuming task. In this paper, we propose a semi-supervised deep learning scheme that can learn from non-annotated data and annotated data when available. It uses an NLG and a Natural Language Understanding (NLU) sequence-to-sequence models which are learned jointly to compensate for the lack of annotation. Experiments on two benchmark datasets show that, with limited amount of annotated data, the method can achieve very competitive results while not using any pre-processing or re-scoring tricks. These findings open the way to the exploitation of non-annotated datasets which is the current bottleneck for the E2E NLG system development to new applications.

1 Introduction

Natural Language Generation (NLG) is an NLP task that consists in generating a sequence of natural language sentences from non-linguistic data. Traditional approaches of NLG consist in creating specific algorithms in the consensual NLG pipeline (Gatt and Krahmer, 2018), but there has been recently a strong interest in End-to-End (E2E) NLG systems which are able to jointly learn sentence planning and surface realization (Dušek and Jurcícek, 2016; Agarwal et al., 2018; Juraska et al., 2018; Gehrmann et al., 2018). Probably the most well known effort of this trend is the E2E NLG challenge (Novikova et al., 2017b) whose task was to perform sentence planing and realization from dialogue act-based Meaning Representation (MR) on *unaligned* data. For instance, Figure 1 presents, on the upper part, a meaning representation and on the lower part, one possible textual realization to convey this meaning. Although the challenge was a great success, the data used in the challenge contained a lot of redundancy of structure and a limited amount of concepts and several reference texts per MR input (8.1 in average). This is an ideal case for machine learning but is it the one that is encountered in all E2E NLG real-world applications?

Source sequence (MR):
name[The Eagle], eatType[coffee shop], food[French], priceRange[moderate], customerRating[3/5], area[riverside], kidsFriendly[yes], near[Burger King]

Target sequence (natural language):
The three star coffee shop, The Eagle, gives families a mid-priced dining experience featuring a variety of wines and cheeses. Find The Eagle near Burger King.

Figure 1: Example of Meaning Representation (MR) and one of its paired possible text realizations. This is a excerpt of the E2E NLG challenge dataset.

In this work, we are interested in learning E2E models for real world applications in which there is a low amount of annotated data. Indeed, it is well known that neural approaches need a large amount of carefully annotated data to be able to induce NLP models. For the NLG task, that means that MR and (possibly many) reference texts must be *paired* together so that supervised learning is made possible. In NLG, such paired datasets are rare and remains tedious to acquire (Novikova et al., 2017b; Gardent et al., 2017; Qader et al., 2018). On the contrary, large amount of *unpaired* meaning representations and texts can be available but cannot be exploited for supervised learning.

In order to tackle this problem, we propose a semi-supervised learning approach which is able to benefit from unpaired (non-annotated) dataset which are much easier to acquire in real life applications. In an unpaired dataset, only the input data

Proceedings of The 12th International Conference on Natural Language Generation, pages 552–562,
Tokyo, Japan, 28 Oct - 1 Nov, 2019. ©2019 Association for Computational Linguistics

is assumed to be representative of the task. In such case, autoencoders can be used to learn an (often more compact) internal representation of the data. Monolingual word embeddings learning also benefit from unpaired data. However, none of these techniques are fit for the task of generating from a constrained MR representation. Hence, we extend the idea of autoencoder which is to regenerate the input sequence by using an NLG and an NLU models. To learn the NLG model, the input text is fed to the NLU model which in turn feeds the NLG model. The output of the NLG model is compared to the input and a loss can be computed. A similar strategy is applied for NLU. This approach brings several advantages: 1) the learning is performed from a large unpaired (non-annotated) dataset and a small amount of paired data to constrain the inner representation of the models to respect the format of the task (here MR and abstract text); 2) the architecture is completely differentiable which enables a fully joint learning; and 3) the two NLG and NLU models remain independent and can thus be applied to different tasks separately.

The remaining of this paper gives some background about seq2seq models (Sec 2) before introducing the joint learning approach (Sec 3). Two benchmarks, described in Sec 4, have been used to evaluate the method and whose results are presented in Sec 5. The method is then positioned with respect to the state-of-the-art in Sec 6 before providing some concluding remarks in Sec 7.

2 Background: E2E systems

E2E Natural Language Generation systems are typically based on the Recurrent Neural Network (RNN) architecture consisting of an encoder and a decoder also known as seq2seq (Sutskever et al., 2014). The encoder takes a sequence of source words $\mathbf{x} = \{x_1, x_2, ..., x_{T_x}\}$ and encodes it to a fixed length vector. The decoder then decodes this vector into a sequence of target words $\mathbf{y} = \{y_1, y_2, ..., y_{T_y}\}$. Seq2seq models are able to treat variable sized source and target sequences making them a great choice for NLG and NLU tasks.

More formally, in a seq2seq model, the recurrent unit of the encoder, at each time step t receives an input word x_t (in practice the embedding vector of the word) and a previous hidden state $h_t - 1$ then generates a new hidden state h_t using:

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

where the function f is an RNN unit such as Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) or Gated Recurrent Unit (GRU) (Cho et al., 2014). Once the encoder has treated the entire source sequence, the last hidden state h_{T_x} is passed to the decoder. To generate the sequence of target words, the decoder also uses an RNN and computes, at each time step, a new hidden state s_t from its previous hidden state s_{t-1} and the previously generated word y_{t-1}. At training time, y_{t-1} is the previous word in the target sequence (teacher-forcing). Lastly, the conditional probability of each target word y_t is computed as follows:

$$P(y_t|\mathbf{y}_{<t}, \mathbf{x}) = softmax(W[s_t, c_t] + b), \tag{2}$$

where W and b are a trainable parameters used to map the output to the same size as the target vocabulary and c_t is the context vector obtained using the sum of hidden states in the encoder, weighted by its attention (Bahdanau et al., 2014; Luong et al., 2015). The context is computed as follow:

$$c_t = \sum_{i=1}^{T_x} \alpha_i^t h_i \tag{3}$$

Attention weights α_i^t are computed by applying a softmax function over a score calculated using the encoder and decoder hidden states:

$$\alpha_i^t = softmax(e_i^t) \tag{4}$$

$$e_i^t = score(s_t, h_i) \tag{5}$$

The choice of the score adopted in this papers is based on the *dot attention* mechanism introduced in (Luong et al., 2015). The attention mechanism helps the decoder to find relevant information on the encoder side based on the current decoder hidden state.

3 Joint NLG/NLU learning scheme

The joint NLG/NLU learning scheme is shown in Figure 2. It consists of two seq2seq models for NLG and NLU tasks. Both models can be trained separately on paired data. In that case, the NLG task is to predict the text $\hat{y}$ from the input MR x

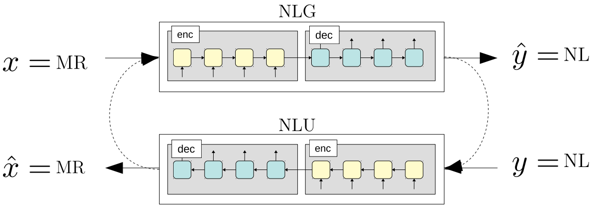

Figure 2: The joint NLG/NLU learning scheme. Dashed arrows between NLG and NLU models show data flow in the case of learning with unpaired data.

while the NLU task is to predict the MR $\hat{x}$ from the input text y. On unpaired data, the two models are connected through two different loops. In the first case, when the unpaired input source is text, y is provided to the NLU models which feeds the NLG model to produce $\hat{y}$. A loss is computed between y and $\hat{y}$ (but not between $\hat{x}$ and x since x is unknown). In the second case, when the input is only MR, x is provided to the NLG model which then feeds the NLU model and finally predicts $\hat{x}$. Similarly, a loss is computed between x and $\hat{x}$ (but not between $\hat{y}$ and y since y is unknown). This section details these four steps and how the loss is backpropagated through the loops.

Learning with Paired Data:

The NLG model is a seq2seq model with attention as described in section 2. It takes as input a MR and generates a natural language text. The objective is to find the model parameters θ^{nlg} such that they minimize the loss which is defined as follows:

$$\mathcal{L}_p^{nlg} = -\frac{1}{T_y}\sum_{t=1}^{T_y} logP(y_t|\mathbf{x};\theta^{nlg}) \tag{6}$$

The NLU model is based on the same architecture but takes a natural language text and outputs a MR and its loss can be formulated as:

$$\mathcal{L}_p^{nlu} = -\frac{1}{T_x}\sum_{t=1}^{T_x} logP(x_t|\mathbf{y};\theta^{nlu}) \tag{7}$$

Learning with Unpaired Data:

When data are unpaired, there is also a loop connection between the two seq2seq models. This is achieved by feeding MR to the NLG model in order to generate a sequence of natural language text $\hat{y}$ by applying an argmax over the probability distribution at each time step ($\hat{y}_t =$ argmax$P(y_t|\mathbf{x};\theta^{nlg})$). This text is then fed back into the NLU model which in turn generates an MR. Finally, we compute the loss between the original MR and the reconstructed MR:

$$\mathcal{L}_u^{nlu} = -\frac{1}{T_x}\sum_{t=1}^{T_x} logP(x_t|\mathbf{x};\theta^{nlg},\theta^{nlu}) \tag{8}$$

The same can be applied in the opposite direction where we feed text to the NLU model and then the NLG model reconstructs back the text. This loss is given by:

$$\mathcal{L}_u^{nlg} = -\frac{1}{T_y}\sum_{t=1}^{T_y} logP(y_t|\mathbf{y};\theta^{nlg},\theta^{nlu}) \tag{9}$$

To perform joint learning, all four losses are summed together to provide the uniq loss $\mathcal{L}$ as follows:

$$\mathcal{L} = \alpha{\cdot}\mathcal{L}_p^{nlg} + \beta{\cdot}\mathcal{L}_p^{nlu} + \gamma{\cdot}\mathcal{L}_u^{nlg} + \delta{\cdot}\mathcal{L}_u^{nlu} \tag{10}$$

The weights α, β, δ and $\gamma \in [0,1]$ are defined to fine tune the contribution of each task and data to the learning or to bias the learning towards one specific task. We show in the experiment section the impact of different settings.

Since the loss functions in Equation 6 and 7 force the model to generate a sequence of words based on the target and the losses in Equation 9 and 8 force the model to reconstruct back the input sequence, this way the model is encouraged to generate text that is supported by the facts found in the input sequence. It is important to note that the gradients based on $\mathcal{L}_p^{nlg}$ and $\mathcal{L}_p^{nlu}$ can only backpropagate through their respective model (i.e., NLG and NLU), while $\mathcal{L}_u^{nlg}$ and $\mathcal{L}_u^{nlu}$ gradients should backpropagate through both models.

Straight-Through Gumbel-Softmax:

A major problem with the proposed joint learning architecture in the unpaired case is that the model is not fully differentiable. Indeed, given the input x and the intermediate output $\hat{y}$, the $\mathcal{L}_u^{nlu}$ and the NLG parameter θ_{nlg}, the gradient is computed as:

$$\frac{\partial\mathcal{L}_u^{nlu}}{\partial\theta_{nlg}} = \sum_t^T \left(\frac{\partial\mathcal{L}_u^{nlu}}{\partial\hat{y}_t} + \frac{\partial\hat{y}_t}{\partial p_{y_t}} + \frac{\partial p_{y_t}}{\partial\theta_{nlg}}\right) \tag{11}$$

At each time step t, the output probability p_{y_t} is computed trough the softmax layer and $\hat{y}_t$ is obtained using $\hat{y}_t = onehot(argmax_w p_{y_t}[w])$ that is the word index w with maximum probability at time step t. To address this problem, one solution is to replace this operation by the identity matrix $\frac{\partial \hat{y}_t}{\partial p_{y_t}} \approx \mathbb{1}$. This approach is called the Straight-Through (ST) estimator, which simply consists of backpropagating through the argmax function as if it had been the identity function (Bengio et al., 2013; Yin et al., 2019).

A more principled way of dealing with the non-differential nature of argmax, is to use the Gumbel-Softmax which proposes a continuous approximation to sampling from a categorical distribution (Jang et al., 2017). Hence, the discontinuous argmax is replaced by a differentiable and smooth function. More formally, consider a k-dimensional categorical distribution u with probabilities $\pi_1, \pi_2, ..., \pi_k$. Samples from u can be approximated using:

$$y_i = \frac{\exp((\log(\pi_i) + g_i)/\tau)}{\sum_{j=1}^{k} \exp((\log(\pi_j) + g_j)/\tau)} \tag{12}$$

$$gi = -\log(-\log(u_i)) \tag{13}$$

$$u_i \sim \text{Uniform}(0, 1), \tag{14}$$

where g_i is the Gumbel noise drawn from a uniform distribution and τ is a temperature parameter. The sample distribution from the Gumbel-Softmax resembles the argmax operation as $\tau \to 0$, and it becomes uniform when $\tau \to \infty$.

Although Gumbel-Softmax is differentiable, the samples drawn from it are not adequate input to the subsequent models which expect a discrete values in order to retrieve the embedding matrix of the input words. So, instead, we use the Straight-Through (ST) Gumbel-Softmax which is basically the discrete version of the Gumbel-Softmax. During the forward phase, ST Gumbel-Softmax discretizes y in Equation 12 but it uses the continuous approximation in the backward pass. Although the Gumbel-Softmax estimator is biased due to the sample mismatch between the backward and forward phases, many studies have shown that ST Gumbel-Softmax can lead to significant improvements in several tasks (Choi et al., 2018; Gu et al., 2018; Tjandra et al., 2018).

4 Dataset

The models developed were evaluated on two datasets. The first one is the E2E NLG challenge dataset (Novikova et al., 2017b) which contains 51k of annotated samples. The second one is the Wikipedia Company Dataset (Qader et al., 2018) which consists of around 51K of noisy MR-abstract pairs of company descriptions.

4.1 E2E NLG challenge Dataset

The E2E NLG challenge Dataset has become one of the benchmarks of reference for end-to-end sentence-planning NLG systems. It is still one of the largest dataset available for this task. The dataset was collected via crowd-sourcing using pictorial representations in the domain of restaurant recommendation.

Although the E2E challenge dataset contains more than 50k samples, each MR is associated on average with 8.1 different reference utterances leading to around 6K unique MRs. Each MR consists of 3 to 8 slots, such as *name*, *food* or *area*, and their values and slot types are fairly equally distributed. The majority of MRs consist of 5 or 6 slots while human utterances consist mainly of one or two sentences only. The vocabulary size of the dataset is of 2780 distinct tokens.

4.2 The Wikipedia Company Dataset

The wikipedia company dataset (Qader et al., 2018), is composed of a set of company data from English Wikipedia. The dataset contains 51k samples where each sample is composed of up to 3 components: the Wikipedia article abstract, the Wikipedia article body, and the infobox which is a set of attribute–value pairs containing primary information about the company (*founder*, *creation date* etc.). The infobox part was taken as MR where each attribute–value pair was represented as a sequence of string `attribute [value]`. The MR representation is composed of 41 attributes with 4.5 attributes per article and 2 words per value in average. The abstract length is between 1 to 5 sentences. The vocabulary size is of 158464 words.

The Wikipedia company dataset contains much more lexical variation and semantic information than the E2E challenge dataset. Furthermore, company texts have been written by humans within the Wikipedia ecosystem and not during a controlled experiment whose human engagement was unknown. Hence, the Wikipedia dataset seems an ecological target for research in NLG. However, as pointed out by the authors, the Wikipedia dataset is not ideal for machine learning. First, the data is not controlled and each article contains only one reference (vs. 8.1 for the E2E challenge dataset). Second the abstract, the body and the infobox are only loosely correlated. Indeed, the meaning representation coverage is poor since, for some MR, none of the information is found in the text and vice-versa. To give a rough estimate of this coverage, we performed an analysis of 100 articles randomly selected in the test set. Over 868 total slot instances, 28% of the slots in the infobox cannot be found in their respective abstract text, while 13% are missing in the infobox.

Despite these problems, we believe the E2E and the Wikipedia company datasets can provide contrasted evaluation, the first being well controlled and lexically focused, the latter representing the kind of data that can be found in real situations and that E2E systems must deal with in order to percolate in the society.

5 Experiments

The performance of the joint learning architecture was evaluated on the two datasets described in the previous section. The joint learning model requires a paired and an unpaired dataset, so each of the two datasets was split into several parts.

E2E NLG challenge Dataset: The training set of the E2E challenge dataset which consists of 42K samples was partitioned into a 10K paired and 32K unpaired datasets by a random process. The unpaired database was composed of two sets, one containing MRs only and the other containing natural texts only. This process resulted in 3 training sets: paired set, unpaired text set and unpaired MR set. The original development set (4.7K) and test set (4.7K) of the E2E dataset have been kept.

The Wikipedia Company Dataset: The Wikipedia company dataset presented in Section 4.2 was filtered to contain only companies having abstracts of at least 7 words and at most 105 words. As a result of this process, 43K companies were retained. The dataset was then divided into: a training set (35K), a development set (4.3K) and a test set (4.3K). Of course, there was no intersection between these sets.

The training set was also partitioned in order to obtain the paired and unpaired datasets. Because of the loose correlation between the MRs and their corresponding text, the paired dataset was selected such that it contained the infobox values with the highest similarity with its reference text. The similarity was computed using "difflib" library[1], which is an extension of the Ratcliff and Obershelp algorithm (Ratcliff and Metzener, 1988). The paired set was selected in this way (rather than randomly) to get samples as close as possible to a carefully annotated set. At the end of partitioning, the following training sets were obtained: paired set (10.5K), unpaired text set (24.5K) and unpaired MR set (24.5K).

The way the datasets are split into paired and unpaired sets is artificial and might be biased particularly for the E2E dataset as it is a rather easy dataset. This is why we included the Wikipedia dataset in our study since the possibility of having such bias is low because 1) each company summary/infobox was written by different authors at different time within the wikipedia eco-system making this data far more natural than in the E2E challenge case, 2) there is a large amount of variation in the dataset, and 3) the dataset was split in such a way that the paired set contains perfect matches between the MR and the text, while reserving the least matching samples for the the unpaired set (i.e., the more representative of real-life Wikipedia articles). As a result, the paired and unpaired sets of the Wikipedia dataset are different from each other and the text and MR unpaired samples are only loosely correlated.

5.1 Evaluation with Automatic Metrics

For the experiments, each seq2seq model was composed of 2 layers of Bi-LSTM in the encoder and two layers of LSTM in the decoder with 256 hidden units and *dot attention* trained using Adam optimization with learning rate of 0.001. The embeddings had 500 dimensions and the vocabulary was limited to 50K words. The Gumbel-Softmax temperature τ was set to 1. Hyper-parameters tuning was performed on the development set and models were trained until the loss on the development set stops decreasing for several consecutive iterations. All models were implemented with PyTorch library.

[1] https://docs.python.org/2/library/difflib.html#difflib.SequenceMatcher

					NLG			NLU		
System	α	β	γ	δ	BLEU	Rouge-L	Meteor	Precision	Recall	F-score
Paired	-	-	-	-	0.60	0.64	0.42	0.74	**0.83**	0.78
Paired + Unpaired	0.25	0.25	1	1	**0.64**†	0.66†	0.43	0.73	0.78	0.76
	0.1	0.1	1	1	**0.64**†	**0.67**†	0.42	0.73	0.74	0.74
	1	0.1	1	1	0.63†	**0.67**†	0.43†	0.72	0.78	0.75
	1	0.1	1	0.1	**0.64**†	**0.67**†	**0.45**†	**0.77**	**0.83**	**0.80**

Table 1: Results on the test set of E2E dataset. † indicates t-test $p < 0.001$ against the paired NLG results.

					NLG			NLU		
System	α	β	γ	δ	BLEU	Rouge-L	Meteor	Precision	Recall	F-score
Paired	-	-	-	-	0.08	0.24	0.11	**0.20**	0.33	0.25
Paired + Unpaired	0.25	0.25	1	1	0.02†	0.15†	0.07†	**0.20**	**0.43**	**0.27**
	0.1	0.1	1	1	0.04†	0.18†	0.08†	0.08	0.22	0.12
	1	0.1	1	1	0.08	**0.26**†	**0.12**†	0.18	0.42	0.25
	1	0.1	1	0.1	**0.09**†	**0.26**†	**0.12**†	**0.20**	0.35	0.26

Table 2: Results on the test set of Wikipedia company dataset. † indicates t-test $p < 0.001$ against the Paired NLG results.

Results of the experiment on the E2E challenge data are summarized Table 1 for both the NLG and the NLU tasks. BLEU, Rouge-L and Meteor were computed using the E2E challenge metrics script[2] with default settings. NLU performances were computed at the slot level. The model learned using paired+unpaired methods shows significant superior performances than the paired version. Among the paired+unpaired methods, the one of last row exhibits the highest balanced score between NLG and NLU. This is achieved when the weights α and γ favor the NLG task against NLU ($\beta = \delta = 0.1$). This setting has been chosen since the NLU task converged much quicker than the NLG task. Hence lower weight for NLU during the learning avoided over-fitting. This best system exhibits similar performances than the E2E challenge winner for ROUGE-L and METEOR whereas it did not use any pre-processing (delexicalisation, slot alignment, data augmentation) or re-scoring and was trained on far less annotated data.

Results of the experiment on Wikipedia company dataset are summarized Table 2 for both the NLG and the NLU tasks. Due to noise in the dataset and the fact that only one reference is available for each sample, the automatic metrics show very low scores. This is in line with (Qader et al., 2018) for which the best system obtained BLEU$= 0.0413$, ROUGE-L$= 0.266$ and METEOR$= 0.1076$. Contrary to the previous results, the paired method brings one of the best performance. However, the best performing system is the one of the last row which again put more emphasis on the NLG task than on the NLU one. Once again, this system obtained performances comparable to the best system of (Qader et al., 2018) but without using any pointer generator or coverage mechanisms.

In order to further analyze the results, in Table 3 we show samples of the generated text by different models alongside the reference texts. The first two examples are from the model trained on the E2E NLG dataset and the last two are from the Wikipedia dataset. Although on the E2E dataset the outputs of paired and paired+unpaired models seem very similar, the latter resembles the reference slightly more and because of this it achieves a higher score in the automatic metrics. This resemblance to the reference could be attributed to the fact that we use a reconstruction loss which forces the model to generate text that is only supported by facts found in the input. As for the Wikipedia dataset examples, we can see that the model with paired+unpaired data is less noisy and the outputs are generally shorter. The model with only paired data generates unnecessarily longer text with lots of unsupported facts and repetitions. Needless to say that both models are doing lots of mistakes and this is because of all the noise contained in

[2] https://github.com/tuetschek/e2e-metrics

Input	name[the punter], eattype[restaurant], food[indian], pricerange[moderate], customer_rating[1 out of 5], area[city centre], familyfriendly[no], near[express by holiday inn]
Reference	the punter is a restaurant providing indian food in the moderate price range. it is located in the city centre. it is near express by holiday inn. its customer rating is 1 out of 5.
Paired model	the punter is a moderately priced indian restaurant in the city centre near express by holiday inn. it has a customer rating of 1 out of 5.
Paired+unpaired model	the punter is a restaurant providing indian food in the moderate price range. it is located in the city centre. it is near express by holiday inn. its customer rating is 1 out of 5.
Input	name[the cricketers], eattype[restaurant], food[chinese], pricerange[less than 20], customer_rating[low], area[city centre], familyfriendly[no], near[all bar one]
Reference	the cricketers is a restaurant providing chinese for under 20. it has a low customer rating. it is located in the city center. it is not family friendly. it is near all bar one.
Paired model	the cricketers is a chinese restaurant located in the city centre near all bar one that is not family friendly and has a low customer rating and a price range of less than 20.
Paired+unpaired model	the cricketers is a restaurant providing chinese food in the less than 20 price range. it is located in the city centre. it is near all bar one. its customer rating is low.
Input	name1[sedgwick group], headquarters1[london], founded1[1998], industry1[insurance], key_people1[sax riley (chairman)], key_people2[rob whitecooper (ceo)], defunct1[1998]
Reference	sedgwick group plc was a very large british insurance broker . it was listed on the london stock exchange and was a constituent of the ftse 100 index . however it was acquired by marsh & mclennan in 1998 .
Paired model	the porton group was a british insurance company . it was founded in 1998 by former mckinsey and entrepreneur erik muir . it was the first property manager until it was acquired by the saatchi group in 1998 .
Paired+unpaired model	sedgwick group plc was a british insurance company based in london . it was listed on the london stock exchange and was once a constituent of the ftse 100 index .
Input	name1[federal realty investment trust], founded1[1962], industry1[real estate investment trust], key_people1[joseph s vassalluzzo chairman], key_people2[donald c wood ceo president], key_people3[daniel guglielmone cfo], products1[shopping centers], founder1[samuel j gorlitz]
Reference	federal realty investment trust is a real estate investment trust that invests in shopping centers in the northeastern united states , the mid-atlantic states , california , and south florida .
Paired model	city capital trust trust , inc. is a real estate investment trust and investment trust . it was founded in 1962 by robert c. smith , and is based in new york city , and is headquartered in cleveland , connecticut . the company is headquartered in cleveland , florida , and has offices in new york city , new york , and geneva .
Paired+unpaired model	the federal realty investment trust , is a real estate investment trust that invests in shopping centers in the united states . it was founded in 1962 by robert duncan , jr. and john epstein .

Table 3: Sample of generated text from the E2E and Wikipedia test sets using our systems along with the reference text.

	cover.	non-redun.	semant.	gramm.
reference	3.42	4.25	4.19	4.13
paired	2.26	**3.67**	3.28	**4.11**
unpaired	**2.87**†	3.63	**3.67**	3.96

Table 4: Results of the human evaluation per system on the Wikipedia corpus using the best unpaired system. † indicates wilcoxon $p < 0.05$ against the paired results.

the training data.

5.2 Human Evaluation

It is well know that automatic metrics in NLG are poorly predictive of human ratings although they are useful for system analysis and development (Novikova et al., 2017a; Gatt and Krahmer, 2018). Hence, to gain more insight about the generation properties of each model, a human evaluation with 16 human subjects was performed on the Wikipedia dataset models. We set up a web-based experiment and used the same 4 questions as in (Qader et al., 2018) which were asked on a 5-point Lickert scale: How do you judge the Information Coverage of the company summary? How do you judge the Non-Redundancy of Information in the company summary? How do you judge the Semantic Adequacy of the company summary? How do you judge the Grammatical Correctness of the company summary?

For this experiment, 40 company summaries were selected randomly from the test set. Each participant had to treat 10 summaries by first read-

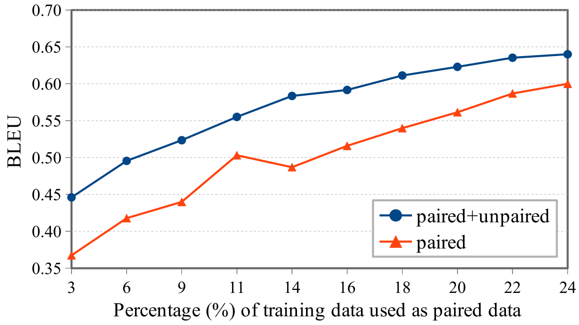

Figure 3: BLEU score as a function of percentage of paired data in the training set on the E2E dataset.

ing the summary and the infobox, then answering the aforementioned four questions.

Results of the human experiment are reported in Table 4. The first line reports the results of the reference (i.e., the Wikipedia abstract) for comparison, while the second line is the model with paired data, and the last line is the model trained on paired+unpaired data with parameters reported in the last row of Table 2, i.e., $\alpha = \gamma = 1$ and $\beta = \delta = 0.1$. It is clear from the coverage metric that no system nor the reference was seen as doing a good job at conveying the information present in the infobox. This is in line with the corpus analysis of section 4. However, between the automatic methods, the unpaired models exhibit a clear superiority in coverage and in semantic adequacy, two measures that are linked. On the other side, the model learned with paired data is slightly more performing in term of non-redundancy and grammaticality. The results of the unpaired model with coverage and grammaticality are equivalent to best models of Qader et al. (2018) but for non-redundancy and semantic adequacy the result are slightly below. This is probably because the authors have used a pointer generator mechanism (See et al., 2017), a trick we avoided and which is subject of further work.

These results express the difference between the learning methods: on the one hand, the unpaired learning relaxes the intermediate labels which are noisy so that the model learns to express what is really in the input (this explain the higher result for coverage) while, on the other hand, the paired learning is only constrained by the output text (not also with the NLU loss as in the unpaired case) which results in slightly more grammatical sentence to the expense of semantic coverage.

α	β	γ	δ	BLEU	Rouge-L	Meteor
1	0.1	1	0.1	**0.64**	**0.67**	**0.45**
0	0.1	1	0.1	0.62	0.66	0.42
1	0	1	0.1	0.63	0.67	0.42
1	0.1	0	0.1	0.50	0.58	0.36
1	0.1	1	0	0.63	0.66	0.44

Table 5: Effect of loss weights on the performance of the NLG model on the E2E dataset.

α	β	γ	δ	Precision	Recall	F-score
1	0.1	1	0.1	**0.77**	**0.83**	**0.80**
0	0.1	1	0.1	0.74	0.79	0.76
1	0	1	0.1	0.74	0.71	0.73
1	0.1	0	0.1	0.68	0.73	0.70
1	0.1	1	0	0.75	0.73	0.74

Table 6: Effect of loss weights on the performance of the NLU model on the E2E dataset.

5.3 Ablation Study

In this section, we further discuss different aspects of the proposed joint learning approach. In particular we are interested in studying the impact of: 1) having different amounts of paired data and 2) the weight of each loss function on the overall performance. Since only the E2E dataset is non-noisy and hence provide meaningful automatic metrics, the ablation study was performed only on this dataset.

To evaluate the dependence on the amount of paired data, the best model was re-trained by changing the size of the paired data ranging from 3% of the training data (i.e., 1K) up to 24% (i.e., 10K). The results are shown in Figure 3. The figure reveals that regardless of the amount of paired data, the joint learning approach: 1) always improves over the model with only paired data and 2) is always able to benefit from supplementary paired data. This is particularly true when the amount of paired data is very small and the difference seems to get smaller as the percentage of the paired data increases.

Next, to evaluate which of the four losses contribute most to the overall performance, the best model was re-trained in different settings. In short, in each setting, one of the weights was set to zero while the others three weights were kept similar as in the best case. The results are presented in Table 5 and Table 6 for NLG and NLU tasks respectively. In these table the first line if the best model as reported in Table 1. It can be seen that all the four losses are important since setting any of the weights to zero leads to a decrease in performance. However, the results of both tables show that the most important loss is the NLG unpaired loss $\mathcal{L}_u^{nlg}$ since setting γ to zeros leads to a significant reduction in the performance for both NLU and NLG.

6 Related Work

The approach of joint learning has been tested in the literature in other domains than NLG/NLU for tasks such machine translation (Cheng et al., 2016; He et al., 2016; Tu et al., 2017) and speech processing (Tjandra et al., 2017, 2018; Liu et al., 2018). In (Tu et al., 2017) an encoder-decoder-reconstructor for MT is proposed. The reconstructor, integrated to the NMT model, rebuilds the source sentence from the hidden layer of the output target sentence, to ensure that the information in the source side is transformed to the target side as much as possible. In (Tjandra et al., 2018), a joint learning architecture of Automatic Speech Recognition (ASR) and Text-To-Speech (TTS) is proposed which leverages unannotated data. In the unannotated case, during the learning, ASR output is fed to the TTS and the TTS output is compared with the original ASR signal input to compute a loss which is back-propagated through both modules. Regarding NLU, joint learning of NLU with other tasks remain scarce. In (Yang et al., 2017), an NLU model is jointly learned with a system action prediction (SAP) model on supervised dialogue data. The NLU model is integrated into the sequence-to-sequence SAP model so that three losses (intent prediction, slot prediction and action prediction) are used to backpropagate through both models. The paper shows that this approach is competitive against the baselines.

To the best of our knowledge, the idea of joint NLG/NLU learning has not been tested previously in NLG. In NLG E2E models (Dušek and Jurcícek, 2016; Juraska et al., 2018), some approaches have learned a concept extractor (which is close to but simpler than an NLU model), but this was not integrated in the NLG learning scheme and only used for output re-scoring. Probably the closest work to our is (Chisholm et al., 2017) in which a seq2seq auto-encoder was used to generate biographies from MR. In this work, the generated text of the 'forward' seq2seq model was constrained by a 'backward' seq2seq model, which shared parameters. However, this works differs from ours since their model was not completely differentiable. Furthermore, their NLU backward model was only used as a support for the forward NLG. Finally, the shared parameters, although in line with the definition of an auto-encoder, make each model impossible to specialize.

7 Conclusion and Further Work

In this paper, we describe a learning scheme which provides the ability to jointly learn two models for NLG and for NLU using large amount of unannotated data and small amount of annotated data. The results obtained with this method on the E2E challenge benchmark, show that the method can achieve a similar score of the winner of the challenge (Juraska et al., 2018) but with far less annotated data and without using any pre-processing (delexicalisation, data augmentation) or re-scoring tricks. Results on the challenging Wikipedia company dataset shows that highest score can be achieve by mixing paired and unpaired datasets. These results are at the state-of-the-art level (Qader et al., 2018) but without using any pointer generator or coverage mechanisms. These findings open the way to the exploitation of unannotated data since the lack of large annotated data source is the current bottleneck of E2E NLG systems development for new applications.

Next steps of the research include, replacing the ST Gumbel-Softmax with reinforcement learning techniques such as policy gradient. This is particularly interesting as with policy gradient we will be able do design reward functions that better suit the problem we are trying to solve. Furthermore, it would be interesting to evaluate how pointer generator mechanism (See et al., 2017) and coverage mechanism (Tu et al., 2016) can be integrated in the learning scheme to increase the non-redundancy and coverage performance of the generation.

Acknowledgments

This project was partly funded by the IDEX Université Grenoble Alpes innovation grant (AI4I-2018-2019) and the Région Auvergne-Rhône-Alpes (AISUA-2018-2019).

References

Shubham Agarwal, Marc Dymetman, and Eric Gaussier. 2018. Char2char generation with reranking for the e2e nlg challenge. In *Proceedings of INLG*, pages 451–456.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR*.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.

Yong Cheng, Wei Xu, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Semi-supervised learning for neural machine translation. In *Proceedings of ACL*, pages 1965–1974.

Andrew Chisholm, Will Radford, and Ben Hachey. 2017. Learning to generate one-sentence biographies from wikidata. In *Proceedings of EACL*, pages 633–642.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of EMNLP*, pages 1724–1734.

Jihun Choi, Kang Min Yoo, and Sang-goo Lee. 2018. Learning to compose task-specific tree structures. In *Proceedings of AAAI*.

Ondřej Dušek and Filip Jurcícek. 2016. Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. In *Proceedings of ACL*, pages 45–51.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for micro-planners. In *Proceedings of ACL*.

Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of AI Research*, pages 65–170.

Sebastian Gehrmann, Falcon Dai, Henry Elder, and Alexander Rush. 2018. End-to-end content and plan selection for data-to-text generation. In *Proceedings of INLG*, pages 46–56.

Jiatao Gu, Daniel Jiwoong Im, and Victor OK Li. 2018. Neural machine translation with gumbel-greedy decoding. In *Proceedings of AAAI*.

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. 2016. Dual learning for machine translation. In *Proceedings of NIPS*, pages 820–828.

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

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *Proceedings of ICLR*.

Juraj Juraska, Panagiotis Karagiannis, Kevin Bowden, and Marilyn A. Walker. 2018. A deep ensemble model with slot alignment for sequence-to-sequence natural language generation. In *Proceedings of NAACL-HLT*, pages 152–162.

Da-Rong Liu, Chi-Yu Yang, Szu-Lin Wu, and Hung-Yi Lee. 2018. Improving unsupervised style transfer in end-to-end speech synthesis with end-to-end speech recognition. In *Proceedings of SLT*, pages 640–647.

Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of EMNLP*, pages 1412–1421.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017a. Why we need new evaluation metrics for nlg. In *Proceedings of EMNLP*, pages 2241–2252.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017b. The E2E dataset: New challenges for end-to-end generation. In *Proceedings of SIGDIAL*, pages 201–206.

Raheel Qader, Khoder Jneid, François Portet, and Cyril Labbé. 2018. Generation of Company descriptions using concept-to-text and text-to-text deep models: dataset collection and systems evaluation. In *Proceedings of INLG*.

John W Ratcliff and David E Metzener. 1988. Pattern-matching-the gestalt approach. *Dr Dobbs Journal*, pages 46–51.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of ACL*, pages 1073–1083.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of NIPS*, pages 3104–3112.

Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura. 2017. Listening while speaking: Speech chain by deep learning. In *Proceedings of ASRU*, pages 301–308.

Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura. 2018. End-to-end feedback loss in speech chain framework via straight-through estimator. *arXiv preprint arXiv:1810.13107*.

Zhaopeng Tu, Yang Liu, Lifeng Shang, Xiaohua Liu, and Hang Li. 2017. Neural machine translation with reconstruction. In *Proceedings of AAAI*.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *Proceedings of ACL*, pages 76–85.

X. Yang, Y. Chen, D. Hakkani-Tr, P. Crook, X. Li, J. Gao, and L. Deng. 2017. End-to-end joint learning of natural language understanding and dialogue manager. In *Proceedings of ICASSP*, pages 5690–5694.

Penghang Yin, Jiancheng Lyu, Shuai Zhang, Stanley J. Osher, Yingyong Qi, and Jack Xin. 2019. Understanding straight-through estimator in training activation quantized neural nets. In *Proceedings of ICLR*.