NABU – Multilingual Graph-Based Neural RDF Verbalizer

Diego Moussallem[1]([✉]) [iD], Dwaraknath Gnaneshwar[2] [iD],
Thiago Castro Ferreira[3,4] [iD], and Axel-Cyrille Ngonga Ngomo[1] [iD]

[1] Data Science Group, University of Paderborn, Paderborn, Germany
{diego.moussallem,axel-cyrille.ngomo}@upb.de
[2] DL Group, Manipal Institute of Technology, Manipal, India
dwarakasharma@gmail.com
[3] Federal University of Minas Gerais (UFMG), Belo Horizonte, Brazil
[4] Tilburg Center for Cognition and Communication (TiCC), Tilburg University,
Tilburg, The Netherlands
tcastrof@tilburguniversity.edu

Abstract. The RDF-to-text task has recently gained substantial attention due to continuous growth of Linked Data. In contrast to traditional pipeline models, recent studies have focused on neural models, which are now able to convert a set of RDF triples into text in an end-to-end style with promising results. However, English is the only language widely targeted. We address this research gap by presenting NABU, a multilingual graph-based neural model that verbalizes RDF data to German, Russian, and English. NABU is based on an encoder-decoder architecture, uses an encoder inspired by Graph Attention Networks and a Transformer as decoder. Our approach relies on the fact that knowledge graphs are language-agnostic and they hence can be used to generate multilingual text. We evaluate NABU in monolingual and multilingual settings on standard benchmarking WebNLG datasets. Our results show that NABU outperforms state-of-the-art approaches on English with 66.21 BLEU, and achieves consistent results across all languages on the multilingual scenario with 56.04 BLEU.

Keywords: Knowledge Graphs · Natural Language Generation · Semantic Web

1 Introduction

Natural Language Generation (NLG) is the process of generating coherent natural language text from non-linguistic data [38]. Despite community agreement on the text and speech output of these systems, there is far less consensus on what

---

the input should be [20]. A large number of inputs have hence been employed for NLG systems, including images [48], numeric data [22], and Semantic Web (SW) data [34]. Practical applications can be found in domains such as weather forecasts [30], feedback for car drivers [8], diet management [1].

Presently, the generation of natural language from Resource Description Framework (RDF) data has gained substantial attention [7]. The RDF-to-text task has hence been proposed to investigate the quality of automatically generated texts from RDF Knowledge Graphs (KGs) [11]. With the emergence of neural methods, end-to-end data-to-text models have been introduced to learn input-output mappings directly. These approaches rely much less on explicit intermediate representations compared to rule-based approaches [21].

Although Neural NLG models have been achieving very good results [19], English is the only language that has been widely targeted. In this work, we alleviate this language limitation by proposing a multilingual approach, named NABU. The motivation behind multilingual models lies in several directions, mainly in (1) transfer learning; when low-resource language pairs are trained together with high-resource languages, the translation quality improves; (2) zero-shot translation, where multilingual models are able to translate between language pairs from similar families that were never seen during training; (3) Easy deploy, a multilingual model achieving same performance on many languages in comparison to several separate language-specific models are much more desirable for companies in terms of deployment [24].

Our approach, NABU, is based on the fact that knowledge graphs are language-agnostic and hence can be used on the encoder side to generate multilingual text. NABU consists of an encoder-decoder architecture which incorporates structural information of RDF triples using an encoding mechanism inspired by Graph Attention Network (GAT) [47]. In contrast to recent related work [39], NABU relies on the use of a reification strategy for modeling the graph structure of RDF input. The decoder part is based on the vanilla Transformer model [46] along with an unsupervised tokenization model.

We evaluate NABU on the standard benchmarking WebNLG datasets [18] in three settings: monolingual, bilingual and multilingual. For the monolingual setting, we compare NABU with state-of-the-art English approaches and also perform experiments on Russian and German. The goal of the bilingual setting is to analyze the performance of NABU for language families. To achieve this goal, we train and evaluate bilingual models using NABU on English-German and on English-Russian. In the multilingual setting, we compare NABU with a multilingual Transformer model on English, German and Russian. Our results show that NABU outperforms state-of-the-art approaches on English and achieves 66.21 BLEU. NABU also achieves consistent results across all languages on multilingual settings with 56.04 BLEU. In addition, NABU presents promising results on the bilingual models with 61.99 BLEU. Our findings suggest that NABU is able to generate multilingual text with similar quality to that generated by humans. The main contributions of this paper can be summarized as follows:

– We present a novel approach dubbed NABU based on a GAT-Transformer architecture for generating multilingual text from RDF KGs.
– NABU outperforms English state-of-the-art approaches with consistent average improvements of +10 BLEU, METEOR and chrF3 on the WebNLG datasets.
– NABU exploits the benefits of modeling of language families in the generation task.

The version of NABU used in this paper and also all experimental data are publicly available.[1]

2   Related Work

A significant body of research has investigated the generation of Natural Language (NL) texts from RDF data. A plenty of research is based on template- and rule-based approaches such as [10,14,15,34]. Recently, the WebNLG [11] challenge made this research area more prominent by providing a benchmark corpus of English texts verbalizing RDF triples in 15 different semantic domains. Among the participating models, the works based on sequence-to-sequence Neural Networks (NNs) achieved some of the best results [32,43]. Moreover, RDF has also been showing promising benefits to the generation of benchmarks for evaluating NLG systems [33].

The choice of neural architectures for RDF-to-text has evolved constantly along the last couple of years. All end-to-end models submitted to the WebNLG challenge [19] received the set of triples in a linearized form as input. However, researchers have recently been experimenting with graph-based approaches, which take the RDF input formatted as a graph, with promising results. Marcheggiane and Perez [29] proposed a structured data encoder based on Graph Convolutional Network (GCN) that directly exploits the graph structure and presented better results than Long Short-Term Memories (LSTM) models. Distiawan et al. [13] presented a GTR-LSTM architecture which captures the global information of a KG by encoding the relationships both within a triple and between the triples. Ferreira et al. [17] introduced a systematic comparison between neural pipeline and end-to-end data-to-text approaches for the generation of text from RDF triples. Although Marcheggiane and Perez [29] showed that the linearisation of the input graph has several drawbacks, the authors implemented Gated recurrent unit (GRU) and Transformer architectures which showed results superior to those of the former architecture. Recently, Ribeiro et.al [39] devised an unified graph attention network structure which investigates graph-to-text architectures that combined global and local graph representations to improve fluency in text generation. Their experiments demonstrated significant improvements on seen categories in the WebNLG dataset.

Despite the plethora of graph-based neural approaches on handling RDF data, English is the only language which has been widely targeted. Recent efforts

---

[1] https://github.com/dice-group/NABU.

were made to create German and Russian language versions of WebNLG [16,42]. However, no work that investigates these languages has been published at the time of writing. To the best of our knowledge, NABU is hence the first approach which tackles multilinguality in the RDF-to-text task.

3 The NABU Approach

NABU tackles RDF-to-text based on the formal description of a translation problem. The RDF-to-text task takes an RDF graph as input and generates an output text which reflects its meaning. Figure 1 depicts an example of a set of 3 RDF triples and the corresponding text. Therefore, the underlying idea behind our approach is as follows: *Given that KGs are language-agnostic and represent facts often extracted from text, we can regard the facts (i.e., RDF triples) as sentences and train a model to translate the facts from a language-agnostic graph representation to several languages.* In the following, we give an overview of GAT architecture and Transformer. Thereafter, we present NABU in detail. Throughout the description of our methodology and our experiments, we use DBpedia [2] as reference Knowledge Base (KB) since the benchmarking datasets are based on this KB.
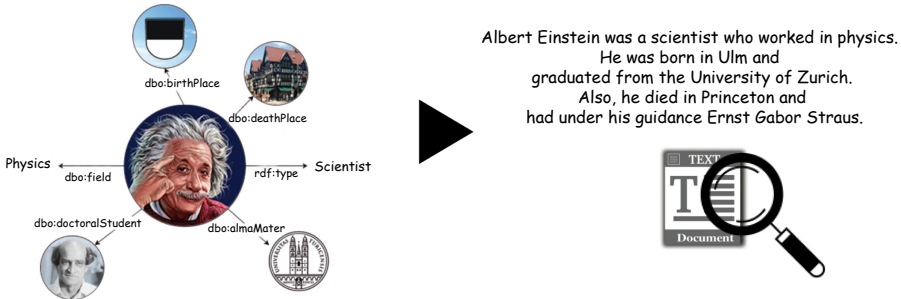
Fig. 1. Example of a set of triples (left) and the corresponding verbalization (right).

3.1 Background

Transformer. Transformer-based models consist of an encoder and a decoder, i.e., a two-tier architecture where the encoder reads an input sequence $x = (x_1, ..., x_n)$ and the decoder predicts a target sequence $y = (y_1, ..., y_n)$. The encoder and decoder interact via a soft-attention mechanism [3,28], which comprises one or multiple attention layers. We follow the notations from Tang et al. [45] in the subsequent sections: Let m stand for the word embedding size and n for the number of hidden units. Further, let K be the vocabulary size of the source language. Then, h_i^l corresponds to the hidden state at step i of layer l. h_{i-1}^l represents the hidden state at the previous step of layer l while

h_i^{l-1} means the hidden state at i of layer $l-1$. $E \in \mathbb{R}^{m \times K}$ is a word embedding matrix, $W \in \mathbb{R}^{n \times m}$, $U \in \mathbb{R}^{n \times n}$ are weight matrices, E_{x_i} refers to the embedding of x_i, and $e_{pos,i}$ indicates a positional embedding at position i.

Transformer models rely deeply on self-attention networks. Each token is connected to every other token in the same sentence directly via self-attention. Thus, the path length between any two tokens is 1. Due to lack of recurrence found in Recurrent Neural Network (RNN), Transformers implement *positional encoding* to input and output. Additionally, these models rely on multi-head attention to feature attention networks, which are more complex in comparison to the 1-head attention mechanism used in RNNs. In contrast to RNN, the positional information is also preserved in positional embeddings. Equation 1 describes the hidden state h_i^l, which is calculated from all hidden states of the previous layer. f represents a feed-forward network with the rectified linear unit (ReLU) as the activation function and layer normalization. The first layer is implemented as $h_i^0 = W E_{x_i} + e_{pos,i}$. Moreover, the decoder has a multi-head attention over the encoder's hidden states:

$$h_i^l = h_i^{l-1} + f(\text{self-attention}(h_i^{l-1})). \tag{1}$$

Graph Attention Networks. Deep Learning on non-euclidean data has recently gained substantial research interest due to the abundance of its availability. A plethora of problems can be solved efficiently by representing data in a data structure that can utilize the inherent structure and inter-entity relationships. Kipf and Welling [26] introduced GCN, through which they generalize the convolution operation of Convolutional Neural Network (CNN) to graph structures. Every layer in a GCN has a weight matrix W that transforms nodes feature vectors from a low-dimensional representation space to high-dimensional representation space, which aims to preserve the structure of the graph.

Consider a graph of z nodes and a set of node features $(\boldsymbol{h_1}, \boldsymbol{h_2}, .., \boldsymbol{h_z})$. A GCN layer computes a net set of features $(\boldsymbol{h_1'}, \boldsymbol{h_2'}, .., \boldsymbol{h_z'})$. First the feature matrix is multiplied with W $\boldsymbol{g} = W\boldsymbol{h}$. Then, the aggregated sum of node features are normalized using normalization constant $\frac{1}{c_{ij}}$ to stabilize the update rule. Finally,

$$\boldsymbol{h_i'} = \sigma \left(\sum_{j \epsilon N_i} \frac{1}{c_{ij}} \boldsymbol{g_j} \right)$$

However, the convolution operation in GCN does not take into account the fact that some nodes are more important than others to generate a particular segment of the target sentence. To alleviate this problem, Velickovic et al. [47] devised GAT, which converts the normalization constant into dynamic attention coefficients. The attention coefficients are calculated by applying *self-attention* over node features. In one forward pass, a GAT layer calculate a score of a given node that quantifies the importance of neighbors to its representation:

$$e_{ij} = a\left(\boldsymbol{h_i}, \boldsymbol{h_j}\right).$$

The attention scores are then normalized using softmax:

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k \epsilon N_i} exp(e_{ik})}.$$

3.2 Approach

Graph-based NNs have been used successfully to parse and support the generation of natural-language sentences from RDF KG. Although GAT models have shown to alleviate the loss of node information, the network still suffers from parameter explosion depending on the size of the graph structure [5]. To alleviate the parameters explosion problem, we follow the same strategy used in [29], named reification,[2] to slightly modify how the RDF graph is encoded. We describe below how reification is applied. Afterward, we explain the encoder and decoder parts of NABU. An overview of NABU architecture after reification can be found in Fig. 2.
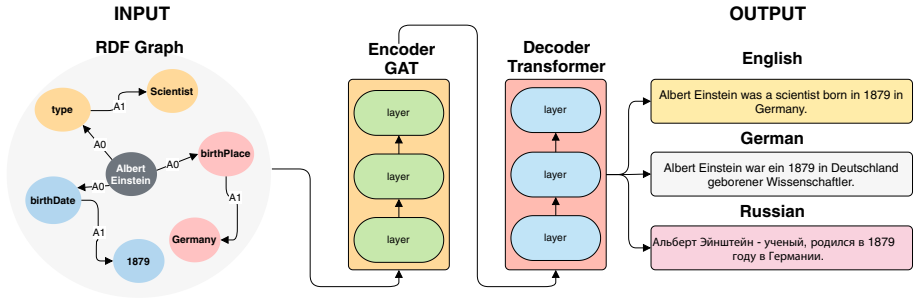

Fig. 2. NABU architecture

Reification. RDF triples are represented as a graph in which (i) the subjects and objects are nodes and (ii) predicates (relationships) between them are labeled edges. For example, `<Albert_Einstein, birthPlace, Germany>` can be seen as a sub-KG in DBpedia where `Albert_ Einstein` and `Germany` are the nodes and `birthPlace` is the edge. However, the edges are encoded as parameters by the GAT, and the parameters explosion problem stated by Beck et al. [5] often occurs.

Therefore, we follow the reification strategy, which maps the relations to nodes in the KG and creates new binary relations for each relation in the RDF triples. We rely on two binary relations, which model the relationship between the subject and predicate (A0) and predicate and object (A1) only. For example, ⟨`Albert_Einstein`, `birthPlace`, `Germany`⟩ becomes ⟨`Albert_Einstein`, A0,

---

[2] Not to be confused with RDFS reification.

`birthPlace⟩` and `⟨birthPlace, A1, Germany⟩`. Apart from handling the parameter explosion problem, reification is useful in two ways. First, the encoder generates a hidden state for each relation in the input. Second, it allows for modeling an arbitrary number of edges (predicates) efficiently. Figure 3 illustrates the reification strategy for our example.

Encoder. Here, the reified graph is sent as input to the GAT that applies a self-attention mechanism to compute the importance of each node in the graph. The GAT encoder represents nodes in a high-dimensional vector space whilst taking into account the representations of their neighbors. Note that NABU follows the same strategy of recent literature on multilingual Neural Machine Translation (NMT) models in which a special token is used in the encoder to determine to what target language to translate [44]. Figure 4 shows how a single forward step/pass works in NABU approach.
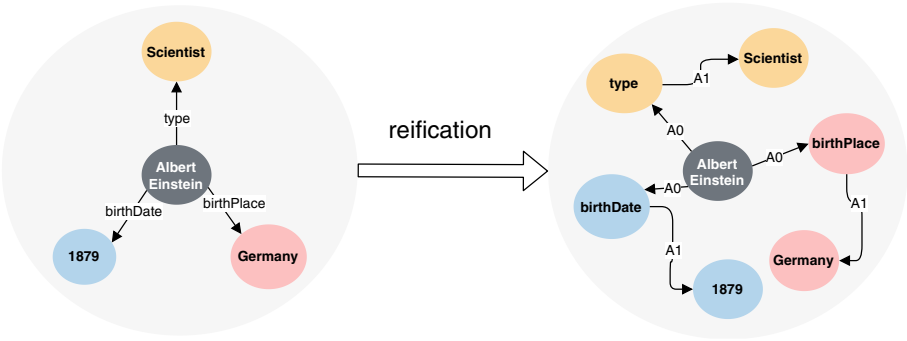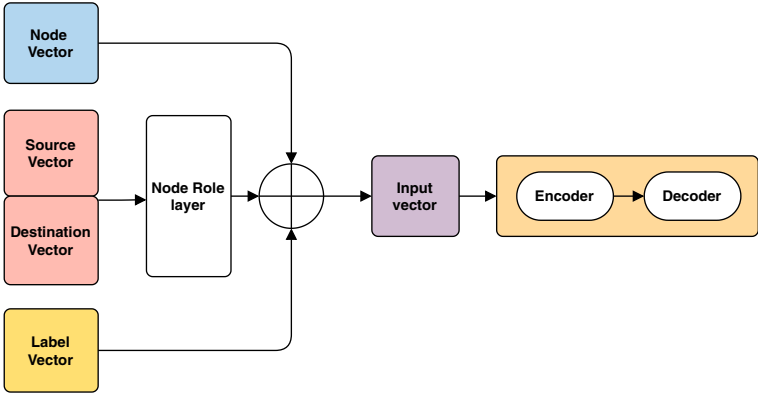
Fig. 3. Reification used on our example.

Fig. 4. An overview of a single forward pass in NABU.

In one forward pass of our model, we have four dense vectors as inputs, namely (i) the *node vector* $\boldsymbol{H} = (\boldsymbol{h_1}, \boldsymbol{h_2}, .., \boldsymbol{h_z})$ with embeddings of all nodes in the graphs, (ii) the *source vector*, $\boldsymbol{S} = (\boldsymbol{s_1}, \boldsymbol{s_2}, .., \boldsymbol{s_z})$ with embeddings of source nodes in edges of the graph, (iii) the *destination vector*, $\boldsymbol{D} = (\boldsymbol{d_1}, \boldsymbol{d_2}, .., \boldsymbol{d_z})$ with embeddings of target nodes in edges of the graphs and (iv) the *label vector*, $\boldsymbol{L} = (\boldsymbol{l_1}, \boldsymbol{l_2}, .., \boldsymbol{l_z})$ with embedding labels. The source $\boldsymbol{S}$ and destination $\boldsymbol{D}$ vectors are concatenated and are passed through dense layer which encodes them into a vector of the same shape as the label vector. We call this new vector the *edge vector*, $\boldsymbol{E}$. We then add the edge vector ($\boldsymbol{E}$), node vector ($\boldsymbol{H}$) and label vector ($\boldsymbol{L}$) to form the input vector to our encoder:

$$\boldsymbol{E} = f(\boldsymbol{S}, \boldsymbol{D}), \text{ and}$$

$$\boldsymbol{H}^{'} = \|_{h \epsilon \eta} G(\boldsymbol{H} + \boldsymbol{L} + \boldsymbol{E}),$$

where η is the number of heads in the multi-head attention layer.

Decoder. Our decoder follows the standard architecture of the Transformer decoder, which takes into account the intermediate representation generated by the encoder. The decoder gives a probability distribution over the target language's vocabulary. We also rely on an unsupervised tokenizer, which implements yte Pair Encoding (BPE) [41] and unigram language model [27] for handling multilinguality and out-of-vocabulary words. Afterward, we apply a beam search for selecting the most likely word in the output sentence.

4   Evaluation

4.1   Goals

In our evaluation, we address the following research questions:

Q1: How does our multilingual approach compare with state-of-the-art results in English?

Q2: Is NABU able to generate bilingual text while modelling two languages from distinct families?

Q3: How accurate are the multilingual texts generated by NABU?

We designed our evaluation as follows: First, we measured the performance of NABU on English by using the WebNLG dataset and compared it with state-of-the-art approaches. Additionally, we evaluated NABU on two other languages—German and Russian. Second, we evaluated NABU on bilingual models—English-German and English-Russian. Third, we combined all three languages in a multilingual setting and compared it with a multilingual Transformer baseline model. For measuring the quality of our approach, we used the automatic evaluation metrics BLEU, METEOR, and CHRF++.

4.2   Data

The experiments presented in this work were conducted on the WebNLG corpus [18, 19], which consists of sets of RDF triples mapped to target texts. In comparison with other popular NLG benchmarks [6, 31, 35], WebNLG is the most semantically varied corpus. Its English version contains 25,298 texts which describe 9,674 sets of up to 7 RDF triples in 15 domains: Astronaut, University, Monument, Building, Comics Character, Food, Airport, Sports Team, Written Work, City, Athlete, Artist, Means of Transportation, Celestial Body and Politician. Out of these domains, five (Athlete, Artist, MeanOfTransportation, Celestial-Body, Politician) are exclusively present in the test set, being unseen during the training and validation processes.

For German and Russian, we relied on the translated versions of WebNLG corpus [9, 42]. The German version comprises 20,370 texts describing 7,812 sets of up to 7 RDF triples in 15 domains. Additionally, the German datasets provide gold-standard representations for traditional pipeline steps, such as discourse ordering (i.e., the order in which the source triples are verbalized in the target text), text structuring (i.e., the organization of the triples into paragraph and sentences), lexicalization (i.e., verbalization of the predicates) and referring expression generation (i.e., verbalization of the entities). The Russian datasets contain 20,800 texts describing 5,185 sets of up to 7 RDF triples in 9 domains. Both were automatically created and manually analyzed. The English and Russian datasets abide by the criteria to gold standards as they were manually assessed by several native speakers. The German version can be regarded as a silver standard given that it did not go through the same process and contains some known errors. For the monolingual experiments, we relied on the standard WebNLG parts of train, dev, and test sets across all languages. Note that the German version does not contain a test set originally. Therefore we relied on a k-Fold Cross-Validation technique to create the test set. For the multilingual set of experiments, we concatenated all English, German and Russian datasets and shuffled their training sets randomly to facilitate an end-to-end training of the model.

4.3   Tasks

We designed three tasks for carrying out our evaluation, (1) Monolingual, (2) Bilingual, (3) Multilingual. (1) In the monolingual task, we train our models to work in each language separately. Hence, we generate three models, one for English, one for German, and another for Russian. Each model receives RDF triples from its given DBPedia language version. For example, the German model receives triples from the German DBpedia. Afterward, we evaluate the models on each WebNLG language-specific dataset. (2) The bilingual task was divided into two sets; the first set, we train one English-German model. This model receives RDF triples from the English and German DBpedia versions as input and has to generate text in English and German, as output. For the second set, we trained

one English-Russian model that receives RDF triples from the English and Russian DBpedia versions and generates text in English and Russian, respectively. (3) In the third task, we train one multilingual model which receives as input the triples from the English, German, and Russian DBpedia versions. This model has to output text in three languages, English, German, and Russian, respectively. The input relies on WebNLG triples containing resources from the English, German, and Russian DBpedia KGs, all entities are found across the three KGs via `sameAs` relations for the sake of completeness.

4.4   Model Settings

In this section, we describe the parameters and hyper-parameters used to train NABU models. We experimented with two encoder-decoder architectures for RDF verbalization. First, Transformer$_{baseline}$ which is an encoder-decoder model with a pure transformer architecture used to both encode triples into intermediate representation and decode it into tokens. Second, NABU$_{GAT-Trans}$, which comprises a GAT encoder and Transformer as the decoder.

For both models, we relied on the same settings. We used a Transformer 6-layer encoder-decoder model with an 8-headed multi-head attention mechanism [46]. The training used a batch size of 32 and Adam optimizer with an initial maximum learning rate of 0.001. We set a source and target word embedding's size of 256, and hidden layers to size 256, dropout = 0.3 (naive). We used a vocabulary of 32000 words for the word based models and a beam size of 5. All our vocabularies were trained using the sentencepiece library.[3] In addition, we used a copy mechanism for investigating the out-of-vocabulary (OOV) words issue. This mechanism first tries to substitute the OOV words with target words that have the highest attention weight according to their source words [28]. If the words are not found, it copies the source words to the position of the not-found target word [23]. Note that we added an extra language token at the beginning of our input sentences for the Transformer model, and a language node to the input graph in our GAT model for performing the bilingual and multilingual experiments. This technique of adding a special language token is in line with [44].

4.5   Evaluation Metrics

We used three automatic Machine Translation (MT) standard metrics to ensure consistent and clear evaluation of the common evaluation datasets of the WebNLG challenge. BLEU [36] uses a modified precision metric for comparing the MT output with the reference (human) translation. The precision is calculated by measuring the n-gram similarity (n = 1,..4) at the word level. BLEU also applies a brevity penalty by comparing the length of the MT output with the reference translation. METEOR [4] was mainly introduced to overcome the semantic weakness of BLEU. To this end, METEOR considers stemming and

---

[3] https://github.com/google/sentencepiece.

paraphrasing along with exact standard word (or phrase) matching. The synonymy overlap through a shared WordNet synset of the words. Along with exact standard word (or phrase) matching, it has additional features, i.e., stemming and paraphrasing. CHRF++ [37] exploits the use of character n-gram precision and recall (F-score) for automatic evaluation of MT outputs. chrF++ has shown a good correlation with human rankings of different MT outputs and is simple and does not require any additional information. Additionally, chrF++ is language- and tokenization-independent.

4.6   Results

Monolingual. Our experiments report that NABU consistently outperforms state-of-the-art models on English data. Table 1 shows that NABU achieved a BLEU score of 66.21, which is 28.15% higher than the previous state-of-the-art Transformer model [17]. We decided to run our experiments on all WebNLG categories to elucidate the strengths and limitations of NABU. According to [17], the main drawback in current NN models is the incapability of generating text for unseen entities and that the experiments should be on all categories. NABU, in turn, shows that it is capable of predicting correctly both seen and unseen entities and their relations. In addition, NABU shows an improvement in METEOR up to +2 points. We report NABU's chrF++ as our intention is to follow recent literature which has adopted this metric due to its good correlation with human results. We can now answer [Q1] as follows: NABU surpasses state-of-the-art results on WebNLG in English.

Table 2 shows that NABU outperforms the transformer baseline on German and Russian. It is important to note that our Transformer baseline, Transformer$_{baseline}$, already outperforms the previous state-of-the-art approaches on English. The difference between our Transformer$_{baseline}$ and the Transformer presented by [17] is that we rely on BPE and character-level tokenizer on the decoder side. Our results suggest that we can refrain from running the related work (see Table 1) on the German and Russian datasets, especially as they were designed and tested to work on English, thus there is currently no baseline for German and Russian. With these results, NABU demonstrates its

Table 1. Results on WebNLG English test set with all categories (seen and unseen), comparison with the state-of-the-art approaches

Model	BLEU	METEOR	chrF++
UPF–FORGe	38.65	39.00	–
Melbourne	45.13	37.00	–
Moryossef et al., (2019)	47.40	39.00	–
Castro et al. (2019)	51.68	32.00	–
NABU$_{GAT-Trans}$	**66.21**	**41.11**	**71.98**

language agnosticism and presents improvements in German and Russian over the baseline.

Table 2. Monolingual results on WebNLG language testsets

Models	Language	BLEU	METEOR	chrF++
Monolingual				
Transformer$_{baseline}$	ENG	54.96	38.43	69.11
	GER	50.07	34.51	63.48
	RUS	46.42	27.74	56.80
NABU$_{GAT-Trans}$	ENG	**66.21**	**41.47**	**71.98**
	GER	**53.08**	**37.42**	**64.57**
	RUS	**46.86**	**28.84**	**58.37**

Bilingual. Table 3 presents the results of NABU$_{GAT-Trans}$ on two bilingual models. The results show that NABU on English-German outperformed the Transformer$_{baseline}$ on all metrics. On English-Russian, NABU$_{GAT-Trans}$ presented worse results on BLEU and METEOR than Transformer$_{baseline}$. However, NABU$_{GAT-Trans}$ showed superior results on chrF++ which is the metric that best correlates with human results. On the one hand, we analyzed that the English-German model leveraged both languages properly due to their vocabulary overlap. German and English share a word vocabulary of 33%, thus training both languages with NABU$_{GAT-Trans}$, which employs a graph representation on the encoder side and a character level on decoder could actually model both languages correctly and generate coherent text. On the other hand, English-Russian presented inconsistent results because both languages are significantly different, and they do not share any vocabulary. We reckoned these conflicting scores are due to the language family of both languages. Looking manually at the results, we concluded that encoding distinct language families requires additional features, and we, therefore, plan to investigate this phenomenon in the future. The results presented herein answer our second research question, [Q2], by showing that NABU is capable of modeling languages from distinct families in a bilingual approach, but a deeper investigation is required.

Table 3. Bilingual results on WebNLG language test sets

Models	Language	BLEU	METEOR	chrF++
Bilingual				
Transformer$_{baseline}$	ENG-GER	58.30	36.46	66.72
NABU$_{GAT-Trans}$	ENG-GER	**61.99**	**39.51**	**69.68**
Transformer$_{baseline}$	ENG-RUS	**55.30**	**37.90**	61.63
NABU$_{GAT-Trans}$	ENG-RUS	49.15	33.41	**64.00**

Multilingual. Table 4 shows that NABU$_{GAT-Trans}$ performed better than Transformer$_{baseline}$ by presenting consistent improvement of $+2$ BLEU, METEOR, and chrF++. This result exhibits that NABU can effectively generate multilingual text, thus answering our third research question, [Q3]. Comparing the multilingual results of NABU with its bilingual results on English-Russian, we concluded that the characteristics of the German language, namely its three gender types, contributed to the better alignment of the languages in the decoder side of multilingual NABU model. Russian also contains three genders as German; therefore, NABU made use of it as features for generating coherent texts. We also noticed that the English texts generated by the multilingual NABU model are comparable to those of the English state-of-the-art models. NABU's multilingual model is also better than the previous English state-of-the-art by 4 BLEU and presents comparable results on METEOR. This result also reaffirms the capability of NABU for achieving English state-of-the-art results and contributes to our first research question, [Q1].

Table 4. Multilingual Results on WebNLG language testsets

Models	Language	BLEU	METEOR	chrF++
Multilingual				
Transformer$_{baseline}$	ENG-GER-RUS	53.39	36.86	60.72
NABU$_{GAT-Trans}$	ENG-GER-RUS	**56.04**	**38.34**	**62.04**

Time-Performance. All models were trained on NVIDIA Tesla P100. Both NABU$_{GAT-Trans}$ and Transformer$_{baseline}$ models took the same amount of time since they contain the same number of weights. Therefore, the monolingual models took 6 h to be trained, while the multilingual models took 8 h on average. This difference of 2 h lies in the size of the multilingual training dataset, which contains all English, German, and Russian training sets.

4.7   Error Analysis and Discussion

In this section, we report some of the errors found in NABU's output while carrying out a human evaluation. First, we analyzed the discrepancy between BLEU, METEOR, and chrF++: NABU outperformed the previous state-of-the-art approach for English by roughly 15 BLEU, while the difference in METEOR is considerable smaller. Our analysis shows that some entities contained typos and were not generated correctly by NABU. In addition, we found a low variance in the generated synonyms. BLEU ignores these aspects while METEOR penalizes based on them, thus explaining the discrepancy between the scores.

Additionally, we noticed some wrong verbalization of similar predicates (edges) that were responsible for decreasing NABU scores across all languages. For example, NABU was sometimes not able to generate text correctly in the

Artist domain. The problem lies in the triples which contain both `dbo:artist` or `dbo:producer` relations as predicates. Both predicates are often verbalized to "artist". This happens because the predicates share the same domain and range and therefore have a similar vector representation in the embeddings. We plan to address this issue in future work by using a more appropriate embedding model.

We also analyzed the multilingual texts generated by $NABU_{GAT-Trans}$ and $Transformer_{baseline}$. We noticed that the $NABU_{GAT-Trans}$ performed better at structuring the RDF graph as input and verbalizing a structured set of RDF triples, whereas $Transformer_{baseline}$ presented better results than $NABU_{GAT-Trans}$ at ordering (also known as Discourse Ordering step) the triples for a better verbalization. The advantage of $Transformer_{baseline}$ over $NABU_{GAT-Trans}$ in Discourse Ordering seems to be related to the linearized form of its input, which explicitly represents in what order the triples have to be verbalized. Additionally, our reification strategy affected the Discourse Ordering, we noticed it by analyzing the generated text from an input with two equal predicates for different subjects. For example, "Albert_Einstein dbo:birthPlace Germany" and "Michael_Jackson dbo:birthPlace USA". $NABU_{GAT-Trans}$ verbalized this two triples as "Albert Einstein was born in the United States of America and Michael Jackson was born in Germany". This problem occurs because NABU can not identify the subjects of each predicate correctly as they are identical in the encoder side. We plan to address this drawback by investigating new approaches for the structuring and ordering steps.

Another interesting insight is related to the inflections of words in German, similar to [9]. The possessive was often a source of errors when verbalizing into German. The translation "Elliot See's Besatzung war ein Testpilot." is not perfect as the apostrophe ('s) is placed wrongly. However, this problem did not happen when generating the sentence, "Bill Oddies Tochter ist Kate Hardie", where the possessive of "Oddie" is built correctly. Similar insights can be derived pertaining to the preposition "von" (en: of). For example, the entity `Texas_University` was wrongly verbalized as "Universität von Texas" instead of the correct form "Universität Texas". The possessive and related constructions are well-known challenges in MT from English to German. Therefore, we plan to explore this phenomenon in future research deeply.

On the Russian results, we observed that the main challenge was related to the verbalization of unseen entities. In $NABU_{GAT-Trans}$, some entities were copied from their source sentences due to the use of the copy mechanism in NABU. For example, the entity "Visvesvaraya_Technological_University" was generated as "Visvesvaraya Technical University" in the English form instead of being verbalized in the Russian language. Additionally, we perceived that $NABU_{GAT-Trans}$ displayed problems similar to those reported in [42] for generating Entities. However, these problems were mostly detected in the unseen category. Our current hypothesis is that the generation of unseen entities in Russian is more challenging than German and English due to the Cyrillic alphabet.

5   Conclusion

We presented a multilingual RDF verbalizer which relies on graph attention NN along with a reification strategy. Our experiments suggest that our approach, named NABU, outperforms state-of-the-art approaches in English. Additionally, NABU presented consistent results across the languages used in our evaluation. NABU is language-agnostic, which means it can be ported easily to languages other than those considered in this paper. To the best of our knowledge, we are the first approach to exploit and achieve the multilinguality successfully in the RDF-to-text task. As future work, we aim to exploit other graph-based neural architecture and other reification approaches for improving NABU's performance. Additionally, we plan to investigate how to deal with the similarity of relations by combining language models and new evaluation metrics [40]. Moreover, we plan to investigate our methodology in the context of low-resource scenarios as well as on different KGs [12, 25].

References

1. Anselma, L., Mazzei, A.: Designing and testing the messages produced by a virtual dietitian. In: Proceedings of the 11th International Conference on Natural Language Generation, Tilburg University, The Netherlands, November 2018, pp. 244–253. Association for Computational Linguistics (2018)
2. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: a nucleus for a web of open data. In: Aberer, K., et al. (eds.) ASWC/ISWC -2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76298-0_52
3. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
4. 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 MT and/or Summarization, pp. 65–72. ACL (2005)
5. Beck, D., Haffari, G., Cohn, T.: Graph-to-sequence learning using gated graph neural networks. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 273–283 (2018)
6. Belz, A., White, M., Espinosa, D., Kow, E., Hogan, D., Stent, A.: The first surface realisation shared task: overview and evaluation results. In: Proceedings of the 13th European Workshop on Natural Language Generation, Nancy, France, pp. 217–226. Association for Computational Linguistics (2011)

7. Bouayad-Agha, N., Casamayor, G., Wanner, L.: Natural language generation in the context of the semantic web. Semant. Web **5**(6), 493–513 (2014)

8. Braun, D., Reiter, E., Siddharthan, A.: SaferDrive: an NLG-based behaviour change support system for drivers. Nat. Lang. Eng. **24**(4), 551–588 (2018)

9. Ferreira, T.C., Moussallem, D., Krahmer, E., Wubben, S.: Enriching the WebNLG corpus. In: Proceedings of the 11th International Conference on Natural Language Generation, pp. 171–176. Association for Computational Linguistics (2018)

10. Cimiano, P., Lüker, J., Nagel, D., Unger, C.: Exploiting ontology lexica for generating natural language texts from RDF data. In: Proceedings of the 14th European Workshop on Natural Language Generation, Sofia, Bulgaria, August 2013, pp. 10–19. ACL (2013)

11. Colin, E., Gardent, C., Mrabet, Y., Narayan, S., Perez-Beltrachini, L.: The WebNLG challenge: generating text from DBPedia data. In: Proceedings of the 9th INLG Conference, pp. 163–167 (2016)

12. Moussallem, D., et al.: RDF2PT: generating Brazilian Portuguese texts from RDF data. In: The 11th Edition of the Language Resources and Evaluation Conference, Miyazaki (Japan), 7–12 May 2018 (2018)

13. Distiawan, B., Qi, J., Zhang, R., Wang, W.: GTR-LSTM: a triple encoder for sentence generation from RDF data. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1627–1637 (2018)

14. Duma, D., Klein, E.: Generating natural language from linked data: unsupervised template extraction. In: IWCS, pp. 83–94 (2013)

15. Ell, B., Harth, A.: A language-independent method for the extraction of RDF verbalization templates. In: INLG, pp. 26–34 (2014)

16. Ferreira, T.C., Moussallem, D., Krahmer, E., Wubben, S.: Enriching the WebNLG corpus. In: Proceedings of the 11th International Conference on Natural Language Generation, pp. 171–176 (2018)

17. Ferreira, T.C., van der Lee, C., van Miltenburg, E., Krahmer, E.: Neural data-to-text generation: a comparison between pipeline and end-to-end architectures. 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), pp. 552–562 (2019)

18. Gardent, C., Shimorina, A., Narayan, S., Perez-Beltrachini, L.: Creating training corpora for NLG micro-planners. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 179–188. Association for Computational Linguistics (2017)

19. Gardent, C., Shimorina, A., Narayan, S., Perez-Beltrachini, L.: The WebNLG challenge: generating text from RDF data. In: Proceedings of the 10th International Conference on Natural Language Generation, pp. 124–133 (2017)

20. Gatt, A., Krahmer, E.: Survey of the state of the art in natural language generation: core tasks, applications and evaluation. arXiv preprint arXiv:1703.09902 (2017)

21. Gehrmann, S., Dai, F., Elder, H., Rush, A.: End-to-end content and plan selection for data-to-text generation. In: Proceedings of the 11th International Conference on Natural Language Generation, Tilburg University, The Netherlands, November 2018, pp. 46–56. Association for Computational Linguistics (2018)

22. Gkatzia, D., Hastie, H.F., Lemon, O.: Comparing multi-label classification with reinforcement learning for summarisation of time-series data. In: ACL, no. 1, pp. 1231–1240 (2014)

23. 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, vol. 1, pp. 1631–1640 (2016)

24. Johnson, M., et al.: Google's multilingual neural machine translation system: enabling zero-shot translation. Trans. Assoc. Comput. Linguist. **5**, 339–351 (2017)

25. Kaffee, L.-A., et al.: Mind the (language) gap: generation of multilingual Wikipedia summaries from Wikidata for ArticlePlaceholders. In: Gangemi, A., et al. (eds.) ESWC 2018. LNCS, vol. 10843, pp. 319–334. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_21

26. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)

27. Kudo, T.: Subword regularization: improving neural network translation models with multiple subword candidates. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 66–75 (2018)

28. Luong, T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1412–1421. ACL (2015)

29. Marcheggiani, D., Perez, L.: Deep graph convolutional encoders for structured data to text generation. In: Proceedings of the 11th International Conference on Natural Language Generation, pp. 1–9. Association for Computational Linguistics (2018)

30. 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, HLT-NAACL 2016, San Diego, California, pp. 720–730. Association for Computational Linguistics (2016)

31. Mille, S., Belz, A., Bohnet, B., Graham, Y., Pitler, E., Wanner, L.: The first multilingual surface realisation shared task (SR'18): overview and evaluation results. In: Proceedings of the First Workshop on Multilingual Surface Realisation, Melbourne, Australia, July 2018, pp. 1–12. Association for Computational Linguistics (2018)

32. Mrabet, Y., et al.: Aligning texts and knowledge bases with semantic sentence simplification. In: WebNLG 2016 (2016)

33. Ngonga Ngomo, A.-C., Röder, M., Moussallem, D., Usbeck, R., Speck, R.: BENGAL: an automatic benchmark generator for entity recognition and linking. In: Proceedings of the 11th International Conference on Natural Language Generation, pp. 339–349 (2018)

34. Ngonga Ngomo, A.-C., Moussallem, D., Bühman, L.: A holistic natural language generation framework for the semantic web. In: Proceedings of the International Conference Recent Advances in Natural Language Processing, p. 8. ACL (Association for Computational Linguistics) (2019)

35. 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, pp. 201–206 (2017)

36. Papineni, K., Roukos, S., Ward, T., Zhu, W.-J.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (2002)

37. Popović, M.: chrF++: words helping character n-grams. In: Proceedings of the Second Conference on Machine Translation, pp. 612–618 (2017)

38. Reiter, E., Dale, R.: Building Natural Language Generation Systems. Cambridge University Press, Cambridge (2000)

39. Ribeiro, L.F.R., Zhang, Y., Gardent, C., Gurevych, I.: Modeling global and local node contexts for text generation from knowledge graphs. arXiv preprint arXiv:2001.11003 (2020)
40. Sellam, T., Das, D., Parikh, A.: BLEURT: learning robust metrics for text generation. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, July 2020, pp. 7881–7892. Association for Computational Linguistics (2020)
41. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2016, Berlin, Germany, pp. 1715–1725. Association for Computational Linguistics (2016)
42. Shimorina, A., Khasanova, E., Gardent, C.: Creating a corpus for Russian data-to-text generation using neural machine translation and post-editing. In: Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing, pp. 44–49 (2019)
43. Sleimi, A., Gardent, C.: Generating paraphrases from DBPedia using deep learning. In: WebNLG 2016, p. 54 (2016)
44. Tan, X., Ren, Y., He, D., Qin, T., Zhao, Z., Liu, T.-Y.: Multilingual neural machine translation with knowledge distillation. arXiv preprint arXiv:1902.10461 (2019)
45. Tang, G., Müller, M., Rios, A., Sennrich, R.: Why self-attention? A targeted evaluation of neural machine translation architectures. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 4263–4272 (2018)
46. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
47. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
48. Xu, K., et al.: Show, attend and tell: neural image caption generation with visual attention. In: International Conference on Machine Learning, pp. 2048–2057 (2015)