Transfer Learning and Masked Generation for Answer Verbalization

Sebastien Montella
Orange Innovation / Lannion, France
Aix-Marseille Univ. CNRS, LIS / Marseille, France
sebastien.montella@orange.com

Lina M. Rojas-Barahona
Orange Innovation / Lannion, France
linamaria.rojasbarahona@orange.com

Frederic Bechet
AMU/CNRS/LIS, Marseille, France
frederic.bechet@lis-lab.fr

Johannes Heinecke
Orange Innovation / Lannion, France
johannes.heinecke@orange.com

Alexis Nasr
AMU/CNRS/LIS, Marseille, France
alexis.nasr@lis-lab.fr

Abstract

Structured Knowledge has recently emerged as an essential component to support fine-grained Question Answering (QA). In general, QA systems query a Knowledge Base (KB) to detect and extract the raw answers as final prediction. However, as lacking of context, language generation can offer a much informative and complete response. In this paper, we propose to combine the power of transfer learning and the advantage of entity placeholders to produce high-quality verbalization of extracted answers from a structured KB. We claim that such approach is especially well-suited for answer generation. Our experiments show 44.25%, 3.26% and 29.10% relative gain in BLEU over the state-of-the-art on the VQuAnDA, ParaQA and VANiLLa datasets, respectively. We additionally provide minor hallucinations corrections in VANiLLa standing for 5% of each of the training and testing set. We witness a median absolute gain of 0.81 SacreBLEU. This strengthens the importance of data quality when using automated evaluation.

1 Introduction

Question Answering (QA) has witnessed a massive number of stupendous improvements over the past few years which marked a new era of QA. At the core of this significant progress is the huge leap in the use of Pretrained Language Model (PLM). On several benchmarks, state-of-the art QA systems perform on par with human according to reported evaluation metrics. However, despite remarkable accuracy in answer detection and extraction, few works have considered returning a verbalized response to the user. Indeed, most of QA systems outputs over Knowledge Bases (KBs) are utterly bereft of any context. To this extent, more works progressively tackled the Answer Verbalization task (AV) which consists in generating a verbalized form of the answer. As a consequence, the user may benefit from a more contextualized response.

Recently, there have been few techniques proposed to perform surface realisation of a raw answer. With the lack of paired training data, Akermi et al. (2020) investigated an unsupervised method to obtain answer verbalizations for both English and French languages. An initial step was to first check whether the question marker (e.g. *Who, What*) could be *straightforwardly* substituted with the raw answer or not. For instance, with the question *"Who is the president of the U.S.?"*, its raw answer *"Joe Biden"* can directly replace the question marker *"who"* with the question mark substituted with a period. If this is not the case, the question is segmented into chunks based on the syntactic tree parsed with UDPipeFuture (Straka, 2018; Akermi et al., 2021). After defining the raw answer as a new chunk, all possible permutations of the chunks are collected. The most likely permutation is identified with a PLM such as GPT2 (Radford et al., 2019). Finally, Akermi et al. (2020) use BERT (Devlin et al., 2019) to find any possibly missing function words around the raw answer such as *a, an, to, with, in* etc. In spite of its appealing unsupervised mechanism, this method is computationally expensive because of the cost of estimating the likelihood of all (distinct) permutations. Moreover, the likelihood is computed with potential absent words which may jeopardize the final ranking of permutations.

Proceedings of the Workshop on Structured and Unstructured Knowledge Integration (SUKI), pages 47 - 54
July 14, 2022 ©2022 Association for Computational Linguistics

Following, multiple datasets were released to spur the community to apply end-to-end learning (Kacupaj et al., 2020, 2021a; Biswas et al., 2021). Kacupaj et al. (2021c) introduced VOGUE, an end-to-end model based on a dual encoder-decoder architecture. More precisely, the input question is encoded with a first Transformer encoder (Vaswani et al., 2017). On top of that, a logical form of the question is also encoded with an additional Transformer encoder. The logical form is a simplified representation of the question, similar to a query, inspired from Plepi et al. (2021) and Kacupaj et al. (2021b). Taking our aforementioned question example, its logical form is `find(president, U.S)`. During the decoding phase, VOGUE uses entities placeholders[1] for both the raw answer and the subject entity to generate an abstract version of the response. Following the previous example, the generated verbalization would be *"`[ANS]` is the president of `[ENT]`"*. In our work, we utilize a comparable mechanism fused with large-scaled pretrained models to leverage efficient transfer learning.

Specifically, our contribution is twofold:

- We propose a masked answer verbalization coupled with transfer learning to verbalize extracted answers over KBs. Placeholders are generated instead of the correct raw answer. This allows a better generalization and scalability of the model. Then, a post-processing step is applied which consists of replacing the placeholder with the raw answer.

- We provide a minor revision of the VANILLA dataset by correcting entity hallucinations in 5% of the verbalizations. We show evidence that erroneous references may be the culprit of 0.13% absolute median SacreBLEU drop in evaluation and up to 0.81 absolute median gain in SacreBLEU when trained on corrected training data.

2 Our Approach

In this section, we present our method based on transfer learning and masked generation. We consider an input question $X = \{x_1, x_2, \dots, x_{N-1}, x_N\}$ with x_i the i^{th} word and its raw answer $A = \{a_1, a_2, \dots, a_{K-1}, a_K\}$ with a_j the j^{th} word of the answer[2]. The goal is to generate a verbalized answer $Y = \{y_1, y_2, \dots, y_{M-1}, y_M\}$ We model the generation of each token as a conditional θ-parameterized probability distribution. More precisely, we estimate θ such that $P_\theta(y_i|X, A, y_1, y_2, \dots, y_{i-1})$ is maximized.

As mentioned in Dai and Le (2015), Howard and Ruder (2018) and Montella et al. (2020), NLG has significantly benefited from transfer learning and very large PLMs (Devlin et al., 2019; Radford et al., 2019). The generalization ability to unseen data has tremendously improved over the last decades due to the use of excessively large training corpora. As a consequence, we consider two recent PLMs for generation to leverage transfer learning:

- **BART** (Lewis et al., 2020) is based on a Transformer architecture (Vaswani et al., 2017). More specifically, its encoder and decoder correspond to BERT (Devlin et al., 2019) and to GPT (Radford et al., 2019), respectively. During training, BART is pretrained with a denoising objective. It consists in corrupting the input of the model (masking, reordering, etc) and to reconstruct the original, i.e. denoised, input.

- **T5** (Raffel et al., 2020) is similar to the Transformer-based model (Vaswani et al., 2017) with minor changes. For instance, as positional embeddings, a single scalar is added to the logits used for attention weights computation. Also, a simplified layer normalization is utilized. T5 is trained on multiple tasks at once such as question answering, language modeling, span extraction, paraphrasing, sentiment analysis, etc. To do so, all text processing tasks are cast in a text-to-text framework which allows to reuse the same model, loss function, optimizer and so on. Both input and target are textual content or transformed as text. Thus, for binary, numerical or categorical data types, T5 maps such format to strings. Moreover, a specificity of T5 is that the task is informed within the input thanks to a prefix, e.g. *"translate English to German:"* or *"summarize:"*. While finetuning, it is a good practice to reuse the same prefix as the downstream task for efficient transfer learning.

In order to verbalize the answer, a first step consists in encoding X with the encoder part of T5 or BART model. Then, the decoder part takes learned

[1] We use the term *placeholder* and *mask* interchangeably.

[2] The raw answer can be of multiple words.

representations to generate Y. In our case, a placeholder is generated in Y which will be replaced by the raw answer A as explained in next section.

2.1 Masked Answer Verbalization

As humans, our ability to generate a response is independent and agnostic to our own knowledge. For instance, given the question *"What is the capital of Ghana?"*, although the answer, i.e. *"Accra"*, might not be known, one is still able to generate the response *"The capital of Ghana is* [ANSWER]*"* where [ANSWER] stands for a placeholder of the correct raw answer. Therefore, this paradigm could remain when modeling any question answering system. This is a two-stage process. First, a template of the verbalized answer is generated. Secondly, we replace the mask with the corresponding raw answer, i.e. a single or several entities, of the input question. We are aware that this approach works especially well in English, but would require adjustment to other languages such as French or German because of gender agreement. However, several benefits can be pointed out. It alleviates the training of the model since it principally learns to generate templates. In addition, it avoids misspelling of entities during the generation. It has been shown that unseen entities are not handled properly by the generative system (Ferreira et al., 2020). This is further critical when a copy mechanism is not applied. On top of that, using placeholders reduces the complexity of the model by shrinking its vocabulary dimension (last layer). This is also significant regarding training time since a softmax layer is usually applied which is known to be time consuming.

3 Datasets

More and more efforts have been made to construct and annotate new QA datasets. However, most of proposed corpora do not include a well-formed and informative response. In fact, no verbalization of the retrieved answer is usually given. Only the raw answer acts as the final prediction which puts a curb on possible downstream generation task. To this end, we explore newly released datasets equipped with a natural language form of the response:

- **VQuAnDa** (Kacupaj et al., 2020) is based on the Large scale Complex Question Answering Dataset (LC-QuAD). VQuAnDa provides a set of 5000 *complex* questions with their SPARQL queries and their corresponding answer verbalization. A semi-automatic process is used to derive the answer verbalization of each question. The available templates of the questions in LC-QuAD dataset are paraphrased using strict rules (use of active voice, synonyms, order rearranging, etc.) to get natural response templates. Then, a second step consists in extracting raw answers from DBpedia using the SPARQL queries. In case that the number of retrieved answers is greater than 15, the list of answers is replaced with a single token [answer] to avoid long sequences. Lastly, entities and predicates are filled accordingly to generate the final verbalization. To ensure correctness, resulting verbalization are checked *manually* according to (Kacupaj et al., 2021a). There are totally 4000 and 1000 pairs for training and testing sets, respectively.

- **ParaQA** (Kacupaj et al., 2021a) extends VQuAnDa by proposing multiple verbalizations for each question. This paraphrasing task was done using different techniques such as back-translation. At least two verbalizations per questions are given, and up to 8 unique paraphrases are provided in some cases. Thus, more pairs in training set can be found for the same question. We record a total of 12,637 pairs in training. Note that the training and testing splits of ParaQA are different than VQuAnDa.

- **VANiLLa** (Biswas et al., 2021) is a compelling dataset due to its size. Covering more than 300 relations, it was built using a semi-automatic framework. First, direct questions with single entity as answer were extracted from the Complex Sequential Question Answering (CSQA) (Saha et al., 2018) and SimpleQuestions[3] Datasets. After clustering similar questions based on 4-grams, a template-based verbalization of a single instance of each cluster was manually annotated thanks to Amazon Mechanical Turk (AMT). Finally, a post-processing aims at using the resulting templates to infer the verbalization for other similar questions in corresponding clusters. Totally, VANiLLa gathers 85,732 and 21,434 pairs for training and testing.

[3] Available at https://github.com/davidgolub/SimpleQA/tree/master/datasets/SimpleQuestions

	Train	Test
VQuAnDa	4,000	1,000
ParaQA	12,637	1,000
VANiLLa	85,732	21,434

Table 1: Datasets Statistics

These datasets are therefore suitable for the response generation task. Nonetheless, because of the semi-automatic framework, these corpora are prone to errors as we will show in Section 4.4

4 Experiments

In our experiments, we provide empirical results on the introduced datasets in Section 3. In Section 4.2, we compare our transfer learning approach over the existing literature using T5 and BART embedded with a masking strategy. Then, we explore the advantage of placeholders in Section 4.3. Our inputs and outputs with and without our masking approach are depicted in Table 2.

4.1 Training Settings

We use the pretrained BART and T5 models from HuggingFace. For both PLMs, we use their base models, i.e. `facebook/bart-base` and `t5-base` configurations. The input questions and target responses are all lower-cased. Since no validation sets are provided regarding the official splits, we arbitrarily set our hyperparameters for all of our experiments and do not validate them. We choose to finetune models on 10 epochs using a batch size of 32. We use the cross entropy loss and Adam optimizer for optimization. The initial learning rates are set to 1.0×10^{-5} and 1.0×10^{-4} for BART and T5 respectively.[4] For T5, we prefix each question with the prefix *"question:"* as it has already been used during T5 pretraining for question-answering. During generation, we use a greedy decoding (no beam search or sampling is applied). For better reproducibility, our code is available at `https://github.com/Anonymous1911272/answerverbalization`.

4.2 Results

Evaluation of natural language remains a critical issue since it is difficult to automate. Besides, human annotations are usually costly and time-consuming. For fair comparison, we follow exactly the same evaluation protocol and metrics as Kacupaj et al. (2021c), using BLEU (on 4-grams) (Papineni et al., 2002) and METEOR (Banerjee and Lavie, 2005)[5]. Since our predicted verbalizations contain placeholders, we replace them with the raw answers included in the dataset. Therefore, our evaluation does not differ between unmasked approaches. Results on VQuAnDa, ParaQA and VANiLLa datasets are depicted in Table 3.

We can see that transfer learning methods *systematically* show best (bold) or second-to-best (underlined) performances on all datasets. This is not surprising as large pretraining has shown massive improvements over standard approaches. BART exhibits much better performances than T5 on VQuAnDa and ParaQA. On the contrary, T5 is slightly better on VANiLLa. We conjecture that BART is well-fitted to map question to its answer verbalization. Question and response usually share similar words, but in different orders and few words or preprosition could be missing to go from one to another. This exactly corresponds to the denoising objective on which BART has been pretrained. Therefore, the input question can be viewed as a noisy version of the answer verbalization from which BART attempts to reconstruct. Regardless, pretrained models on average results in 44.25%, 3.26% and 29.10% relative gain in BLEU over VOGUE on VQuAnDa, ParaQA and VANiLLa respectively. VOGUE nonetheless shows interesting results despite its size and no pretraining. This is also explained by the logical form of the question which boils down the question to a simple abstraction. Furthermore, we observe that the unsupervised strategy by Akermi et al. (2020) has strong shortcoming to compete with a basic RNN. Their method is sensitive to the syntax and length of the input question. The longer the question, the worst the generation. While the verbalizations in VQuAnDa and ParaQA are 17 tokens long on average, this might be the reason of low performances on these datasets. Moreover, unnatural questions, as included in VANiLLa, are not handled properly because of the use of PLMs to gauge the likelihood of permutations.

In the following, our interest lies in measuring the real gain of using placeholders.

[4] We witness divergence when learning rate is set to 1.0×10^{-4} for BART.

[5] Kacupaj et al. (2021c) average the BLEU and METEOR of each verbalization.

	Input	Output
w/o mask	*Who is the president of the U.S.?* `[SEP]` *J. Biden*	*The president of the U.S. is J. Biden.*
w/ mask	*Who is the president of the U.S.?*	*The president of the U.S. is* `[ANSWER]` .

Table 2: Examples of model input and output with and without placeholders. During evaluation, the placeholder is replaced with the raw answer *J. Biden*.

	BLEU ↑			METEOR ↑		
Models	VQuAnDa	ParaQA	VANiLLa	VQuAnDa	ParaQA	VANiLLa
RNN†	15.43	22.45	16.66	53.15	58.41	58.67
Transformer†	18.37	23.61	30.80	56.83	59.63	62.16
Akermi et al. (2020)	22.70	18.25	18.30	48.04	44.27	48.27
VOGUE†	28.76	32.05	35.46	67.21	**68.85**	65.04
T5 (masking)	39.07	30.62	**45.87**	67.70	59.81	**67.15**
BART (masking)	**43.90**	**35.57**	45.69	**71.92**	65.40	66.71

Table 3: Answer Verbalization Results. (†) results are taken from Kacupaj et al. (2021c).

		BLEU ↑	METEOR ↑	SacreBLEU ↑	Chrf++ ↑	TER ↓
T5	w/o mask	30.06	58.39	35.25	56.44	52.67
	w/ mask	**39.07**	**67.70**	**58.26**	**73.87**	**42.45**
BART	w/o mask	33.93	61.43	39.19	59.26	49.22
	w/ mask	**43.90**	**71.92**	**60.85**	**75.45**	**35.36**

Table 4: Results with and without placeholders on VQuAnDa.

		BLEU ↑	METEOR ↑	SacreBLEU ↑	Chrf++ ↑	TER ↓
T5	w/o mask	25.55	53.55	33.49	53.84	56.04
	w/ mask	**30.62**	**59.81**	**47.01**	**66.39**	**49.87**
BART	w/o mask	30.56	57.80	37.61	56.95	52.41
	w/ mask	**35.57**	**65.40**	**50.70**	**68.86**	**43.50**

Table 5: Results with and without placeholders on ParaQA.

		BLEU ↑	METEOR ↑	SacreBLEU ↑	Chrf++ ↑	TER ↓
T5	w/o mask	42.91	64.56	53.33	70.19	44.41
	w/ mask	**45.87**	**67.15**	**57.67**	**73.40**	**41.59**
Bart	w/o mask	43.14	65.13	54.16	71.36	43.99
	w/ mask	**45.69**	**66.71**	**57.41**	**73.07**	**42.00**

Table 6: Results with and without placeholders on VANiLLa.

4.3 To Mask or not to Mask?

In this section, we investigate the impact of using a masking mechanism. We conduct a comparative study between masked and non-masked generation.

To do so, we finetune BART and T5 with the same hyperparameters as previous experiments. For non-masked generation, the input question is concatenated with its raw answer. To differentiate question and answer, we make use of the separator token `[SEP]`. With this setting, models should learn to combine input question and input answer accordingly to form a grammatically correct verbalization. We adopt additional evaluation metric, i.e. SacreBLEU (Post, 2018), Chrf++ (Popović, 2015) and TER (Snover et al., 2006), yielding much fine-grained analysis. The experiment results for VQuAnDa, ParaQA and VANiLLa are shown in Table 4, 5 and 6.

On the three datasets, we observe that using a placeholder leads to systematic gain for all reported metrics. More importantly, the gap can be considerably significant when masking the raw answer.

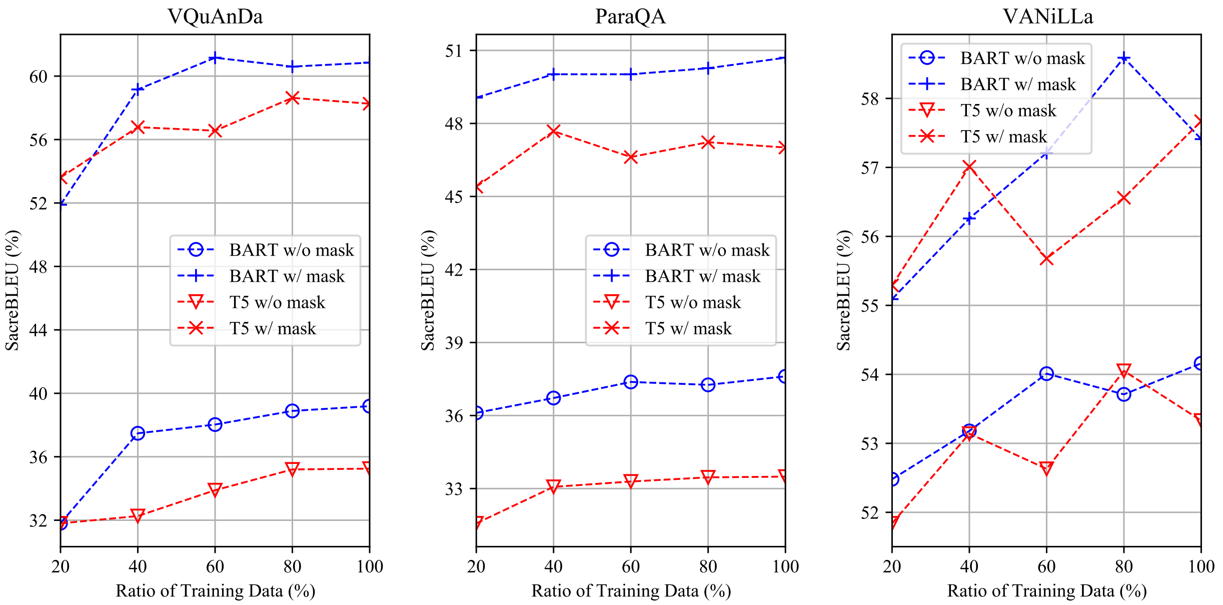

Figure 1: Tuning proportion of training data

		Test Set	
		raw	corrected
T5	w/o mask	53.33	**53.45**
	w/ mask	57.67	**57.81**
BART	w/o mask	54.16	**54.30**
	w/ mask	**57.41**	57.16

		Test Set	
		raw	corrected
T5	w/o mask	52.72	**53.58**
	w/ mask	57.48	**58.23**
BART	w/o mask	54.96	**55.81**
	w/ mask	58.36	**59.13**

Table 7: SacreBLEU scores of T5 and BART trained on raw VANiLLa (left) and corrected VANiLLa (right)

For T5 and BART, we note 23.01%, 13.52%, 4.34% and 21.56%, 13.09%, 3.25% absolute gain in SacreBLEU for VQuAnDa, ParaQA and VANiLLa respectively. Thus, generating a more abstract verbalization alleviates the learning. Following, we inspect the effect of the size of training set. We thereby finetune BART and T5 on different (random) proportion of training data. We report SacreBLEU scores for each portion of training data in Fig. 1. At first glance, the gap between masked and non-masked generation remains very distinctive despite using less training data. We remark for T5 and BART about 23.09%, 13.81%, 3.44% and 21.65%, 13.00%, 3.40% absolute gain on average in SacreBLEU on VQuAnDa, ParaQA and VANiLLa while tuning amount of data fed to models. We observe that both masked and unmasked strategies keep increasing performances when new samples are given. Contrary to expectation, despite the use of placeholder, masked generation keeps benefiting of some significant performance leaps. For BART on VQuAnDa and ParaQA, SacreBLEU reaches a limit with only 40% of the training data in both configurations. On VANiLLa, models show much more variance, but a positive trend remains overall.

4.4 References are not Innocent

Semi-automatic dataset construction is a convenient yet effective technique to automatically generate sizeable corpora. Few handcrafted annotations are needed as initial seed. However, resulting samples are highly prone to errors or not natural. This remains a major drawback in the NLG community where the low quality or diversity of the available data jeopardize comparison between approaches. Within the VANiLLa dataset, we particularly reveal some verbalizations where the subject entity of the question differs with the subject entity of the reference. For example, given the question *“Which sex does Doris Miller belong to?”*, the assigned reference is *“Sterjo is a male”*, with *“Sterjo”* a hallucinated entity, which should be corrected with *“Doris Miller”*. Those hallucinated entities in gold references especially occur with specific and redundant entities (e.g. *“Sterjo”*). We assume the semi-automatic pipeline to be the culprit of such mismatch. Fortunately, those errors can be corrected automatically since the subject entity of each question is explicitly inquired in the original dataset. We identified 12 repeated hallucinated entities over the whole training set of VANiLLa. We thus interchange erroneous entities with correct ones. This stands for 5% for each of the training and testing set. The quality and diversity of references was proved to be at the core of variations of automated metrics outcomes (Freitag et al., 2020). Errors in references directly jeopardize resulting performances of models. Indeed, good predictions might be rated as bad quality while being correct. Furthermore, automatic metrics are critically sensitive to any changes in chosen words in target verbalization. We hence investigate the shift in reported results with corrected references. Precisely, we finetune T5 and BART with same hyperparameters as previously mentioned in Section 4.1. We train and evaluate models on the original VANiLLa dataset (*"Raw"*) and the corrected version (*"Corrected"*). The SacreBLEU scores are given in Table 7.

With only 5% of corrections in both training and testing sets, we record small improvements in SacreBLEU. Although the increases are relatively insignificant, those results clearly indicates that the quality of the references is crucial to precisely assess models performances. More and more works are competing in improving those metrics. Several contributions in generation considered slight improvements as predominance of their approaches over previous methods. However, we show in Table 7 that evaluating models on a corrected version lead to different results that are not systematically better. In contrast, when trained on much higher quality samples, results on corrected testing examples exhibits more important gain as seen in Table 7. The absolute median gain reaches 0.81 SacreBLEU with a cleaner training set while barely 0.13 with the standard training set. As a consequence, it is hard to compare and to draw any conclusions between models on noisy datasets. It is then important to raise awareness toward automatic dataset construction.

5 Conclusion

We proposed to verbalize answers usually returned by any question-answering system from a structured knowledge base. We combined the advantages of transfer learning and masked generation. We compared our strategies with and without masks using T5 and BART. We showed that using massively pretrained models with answer placeholders alleviates the learning and led to unprecedented results on VQuAnDa, ParaQA and VANiLLa datasets. Furthermore, we revealed multiple redundant entity hallucinations in the VANiLLa dataset. By automatically correcting 5% of them, we observed shifts in performances. This further demonstrates the limitation of automatic metrics when references are not reliable.

References

Imen Akermi, Johannes Heinecke, and Frédéric Herledan. 2020. Transformer based natural language generation for question-answering. In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, page 349–359, Dublin, Ireland (Virtual). Association for Computational Linguistics.

Imen Akermi, Johannes Heinecke, and Frédéric Herledan. 2021. Génération automatique de texte en langage naturel pour les systèmes de questions-réponses. *Traitement Automatique des Langues*, 62(1):13–37.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Debanjali Biswas, Mohnish Dubey, Md Rashad Al Hasan Rony, and Jens Lehmann. 2021. Vanilla: Verbalized answers in natural language at large scale. arXiv:2105.11407.

Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Stroudsburg, PA, USA. Association for Computational Linguistics.

Thiago Castro Ferreira, Claire Gardent, Nikolai Ilinykh, Chris Van Der Lee, Simon Mille, Diego Moussallem, and Anastasia Shimorina. 2020. The 2020 Bilingual, Bi-Directional WebNLG+ Shared Task Overview and Evaluation Results (WebNLG+ 2020). In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, Dublin/Virtual, Ireland.

Markus Freitag, David Grangier, and Isaac Caswell. 2020. BLEU might be guilty but references are not innocent. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 61–71, Online. Association for Computational Linguistics.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Endri Kacupaj, Barshana Banerjee, Kuldeep Singh, and Jens Lehmann. 2021a. Paraqa: A question answering dataset with paraphrase responses for single-turn conversation. In *ESWC 2021*, pages 598–613.

Endri Kacupaj, Joan Plepi, Kuldeep Singh, Harsh Thakkar, Jens Lehmann, and Maria Maleshkova. 2021b. Conversational question answering over knowledge graphs with transformer and graph attention networks. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 850–862, Online. Association for Computational Linguistics.

Endri Kacupaj, Shyamnath Premnadh, Kuldeep Singh, Jens Lehmann, and Maria Maleshkova. 2021c. Vogue: Answer verbalization through multi-task learning. In *Machine Learning and Knowledge Discovery in Databases. Research Track*, pages 563–579, Cham. Springer International Publishing.

Endri Kacupaj, Hamid Zafar, Jens Lehmann, and Maria Maleshkova. 2020. Vquanda: Verbalization question answering dataset. In *The Semantic Web*, pages 531–547, Cham. Springer International Publishing.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Sebastien Montella, Betty Fabre, Tanguy Urvoy, Johannes Heinecke, and Lina Rojas-Barahona. 2020. Denoising pre-training and data augmentation strategies for enhanced RDF verbalization with transformers. In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 89–99, Dublin, Ireland (Virtual). Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Joan Plepi, Endri Kacupaj, Kuldeep Singh, Harsh Thakkar, and Jens Lehmann. 2021. Context transformer with stacked pointer networks for conversational question answering over knowledge graphs. In *The Semantic Web*, pages 356–371, Cham. Springer International Publishing.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. https://openai.com/blog/better-language-models/.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Amrita Saha, Vardaan Pahuja, Mitesh M. Khapra, Karthik Sankaranarayanan, and Sarath Chandar. 2018. Complex sequential question answering: Towards learning to converse over linked question answer pairs with a knowledge graph. arXiv:1801.10314.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

Milan Straka. 2018. UDPipe 2.0 Prototype at CoNLL 2018 UD Shared Task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.