Learning to Generate Wikipedia Summaries for Underserved Languages from Wikidata

Lucie-Aimée Kaffee[1†]   **Hady Elsahar**[2†]   **Pavlos Vougiouklis**[1†]
Christophe Gravier[2]   **Frédérique Laforest**[2]   **Jonathon Hare**[1]   **Elena Simperl**[1]

[1] School of Electronics and Computer Science, University of Southampton, UK
{kaffee, pv1e13, jsh2, e.simperl}@ecs.soton.ac.uk
[2] Laboratoire Hubert Curien, CNRS, UJM-Saint-Étienne, Université de Lyon, France
{hady.elsahar, christophe.gravier, frederique.laforest}@univ-st-etienne.fr

Abstract

While Wikipedia exists in 287 languages, its content is unevenly distributed among them. In this work, we investigate the generation of open domain Wikipedia summaries in underserved languages using structured data from Wikidata. To this end, we propose a neural network architecture equipped with copy actions that learns to generate single-sentence and comprehensible textual summaries from Wikidata triples. We demonstrate the effectiveness of the proposed approach by evaluating it against a set of baselines on two languages of different natures: Arabic, a morphological rich language with a larger vocabulary than English, and Esperanto, a constructed language known for its easy acquisition.

1 Introduction

Despite the fact that Wikipedia exists in 287 languages, the existing content is unevenly distributed. The content of the most under-resourced Wikipedias is maintained by a limited number of editors – they cannot curate the same volume of articles as the editors of large Wikipedia language-specific communities. It is therefore of the utmost social and cultural interests to address languages for which native speakers have only access to an impoverished Wikipedia. In this paper, we propose an automatic approach to generate textual summaries that can be used as a starting point for the editors of the involved Wikipedias. We propose an end-to-end trainable model that generates a textual summary given a set of KB triples as input. We apply our model on two languages that have a severe lack of both editors and articles on Wikipedia: Esperanto is an easily acquired artificially created language which makes it less data needy and a more suitable starting point

for exploring the challenges of this task. Arabic is a morphologically rich language that is much more challenging to work, mainly due to its significantly larger vocabulary. As shown in Table 1 both Arabic and Esperanto suffer a severe lack of content and active editors compared to the English Wikipedia which is currently the biggest one in terms of number of articles. Our research is mostly related to previous work on adapting the general encoder-decoder framework for the generation of Wikipedia summaries (Lebret et al., 2016; Chisholm et al., 2017; Vougiouklis et al., 2017). Nonetheless, all these approaches focus on task of biographies generation, and only in English – the language with the most language resources and knowledge bases available. In contrast with these works, we explore the generation of sentences in an open-domain, multilingual context. The model from (Lebret et al., 2016) takes the Wikipedia infobox as an input, while (Chisholm et al., 2017) uses a sequence of slot-value pairs extracted from Wikidata. Both models are only able to generate single-subject relationships. In our model the input triples go beyond the single-subject relationships of a Wikipedia infobox or a Wikidata page about a specific item (Section 2). Similarly to our approach, the model proposed by (Vougiouklis et al., 2017) accepts a set of triples as input, however, it leverages instance-type-related information from DBpedia in order to generate text that addresses rare or unseen entities. Our solution is much broader since it does not rely on the assumption that unseen triples will adopt the same pattern of properties and entities' instance types pairs as the ones that have been used for training. To this end, we use copy actions over the labels of entities in the input triples. This relates to previous works in machine translation which deals with rare or unseen word problem for translating names and numbers in text. (Luong et al., 2015)

[†]The authors contributed equally to this work.

	Arabic	Esperanto	English
# of Articles	541,166	241,901	5,483,928
# of Active Users	7,818	2,849	129,237
Vocab. Size	2.2M	1.5M	2.0M

Table 1: Recent page statistics and number of unique words (vocab. size) of Esperanto, Arabic and English Wikipedias.

propose a model that generates positional placeholders pointing to some words in source sentence and copy it to target sentence (*copy actions*). (Gulcehre et al., 2016) introduce separate trainable modules for copy actions to adapt to highly variable input sequences, for text summarisation. For text generation from tables, (Lebret et al., 2016) extend positional copy actions to copy values from fields in the given table. For Question Generation, (Serban et al., 2016) use a placeholder for the subject entity in the question to generalise to unseen entities.

We evaluate our approach by measuring how close our synthesised summaries can be to actual summaries in Wikipedia against two other baselines of different natures: a language model, and an information retrieval template-based solution. Our model substantially outperforms all the baselines in all evaluation metrics in both Esperanto and Arabic. In this work we present the following contributions: i) We investigate the task of generating textual summaries from Wikidata triples in underserved Wikipedia languages across multiple domains, and ii) We use an end-to-end model with copy actions adapted to this task. Our datasets, results, and experiments are available at: https://github.com/pvougiou/Wikidata2Wikipedia.

2 Model

Our approach is inspired by similar encoder-decoder architectures that have already been employed on similar text generative tasks (Serban et al., 2016; Vougiouklis et al., 2017).

2.1 Encoding the Triples

The encoder part of the model is a feed-forward architecture that encodes the set of input triples into a fixed dimensionality vector, which is subsequently used to initialise the decoder. Given a set of un-ordered triples $F_E = \{f_1, f_2, \ldots, f_R : f_j = (s_j, p_j, o_j)\}$, where s_j, p_j and o_j are the one-hot vector representations of the respective sub-

ject, property and object of the j-th triple, we compute an embedding h_{f_j} for the j-th triple by forward propagating as follows:

$$h_{f_j} = q(\mathbf{W_h}[\mathbf{W_{in}}s_j; \mathbf{W_{in}}p_j; \mathbf{W_{in}}o_j]) \ , \quad (1)$$
$$h_{F_E} = \mathbf{W_F}[h_{f_1}; \ldots; h_{f_{R-1}}; h_{f_R}] \ , \quad (2)$$

where h_{f_j} is the embedding vector of each triple f_j, h_{F_E} is a fixed-length vector representation for all the input triples F_E. q is a non-linear activation function, $[\ldots; \ldots]$ represents vector concatenation. $\mathbf{W_{in}}, \mathbf{W_h}, \mathbf{W_F}$ are trainable weight matrices. Unlike (Chisholm et al., 2017), our encoder is agnostic with respect to the order of input triples. As a result, the order of a particular triple f_j in the triples set does not change its significance towards the computation of the vector representation of the whole triples set, h_{F_E}.

2.2 Decoding the Summary

The decoder part of the architecture is a multi-layer RNN (Cho et al., 2014) with Gated Recurrent Units which generates the textual summary one token at a time. The hidden unit of the GRU at the first layer is initialised with h_{F_E}. At each timestep t, the hidden state of the GRU is calculated as follows:

$$h_t^l = \text{GRU}(h_{t-1}^l, h_t^{l-1}) \quad (3)$$

The conditional probability distribution over each token y_t of the summary at each timestep t is computed as the softmax$(\mathbf{W_{out}}h_t^L)$ over all the possible entries in the summaries dictionary, where h_t^L is the hidden state of the last layer and $\mathbf{W_{out}}$ is a biased trainable weight matrix.

A summary consists of words and mentions of entity in the text. We adapt the concept of *surface form tuples* (Vougiouklis et al., 2017) in order to be able to learn an arbitrary number of different lexicalisations of the same entity in the summary (e.g. "aktorino", "aktoro"). Figure 1 shows the architecture of our generative model when it is provided with the three triples of the idealised example of Table 2.

2.3 Copy Actions

Following (Luong et al., 2015; Lebret et al., 2016) we model all the copy actions on the data level through a set of special tokens added to the basic vocabulary. Rare entities identified in text and existing in the input triples are being replaced by the token of the property of the relationship to which it

Triples	Q490900 (Floridia)	P31 (estas)	Q747074 (komunumo de Italio)
	Q490900 (Floridia)	P17 (ŝtato)	Q38 (Italio)
	Q30025755 (Floridia)	P1376 (ĉefurbo de)	Q490900 (Floridia)
Textual Summary	Floridia estas komunumo de Italio.		
Vocab. Extended	[[Q490900, Floridia]] estas komunumo de [[P17]].		

Table 2: Training example: a set of triples about *Floridia*. Subsequently, our system summarises the input set in the form of text. The vocabulary extended summary is the one on which we train our model.
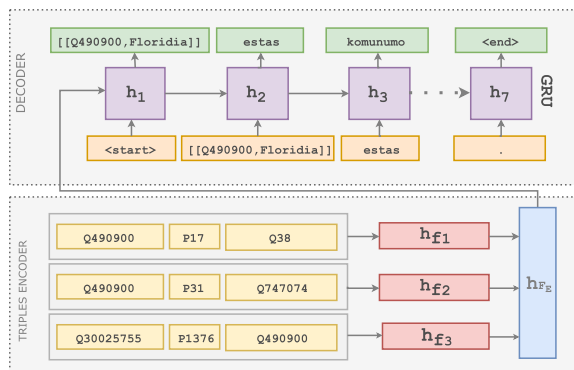

Figure 1: Model Overview

was matched. We refer to those tokens as *property placeholders*. In Table 2, [[P17]] in the vocabulary extended summary is an example of property placeholder – would it be generated by our model, it is replaced with the label of the object of the triple with which they share the same property (i.e. Q490900 (Floridia) P17 (ŝtato) Q38 (Italio)). When all the tokens of the summary are sampled, each property placeholder that is generated is mapped to the triple with which it shares the same property and is subsequently replaced with the textual label of the entity. We randomly choose an entity, in case there are more than one triple with the same property in the input triples set.

2.4 Implementation and Training Details

We implemented our neural network models using the Torch[1] package.

We included the $15,000$ and $25,000$ most frequent tokens (i.e. either words or entities) of the summaries in Esperanto and Arabic respectively for target vocabulary of the textual summaries. Using a larger size of target dictionary in Arabic is due to its greater linguistic variability – Arabic vocabulary is 47% larger than Esperanto vocabulary (cf. Table 1). We replaced any rare enti-

ties in the text that participate in relations in the aligned triples set with the corresponding property placeholder of the upheld relations. We include all property placeholders that occur at least 20 times in each training dataset. Subsequently, the dictionaries of the Esperanto and Arabic summaries are expanded by 80 and 113 property placeholders respectively. In case the rare entity is not matched to any subject or object of the set of corresponding triples it is replaced by the special <resource> token. Each summary is augmented with the respect start-of-summary <start> and end-of-summary <end> tokens.

For the decoder, we use 1 layer of GRUs. We set the dimensionality of the decoder's hidden state to 500 in Esperanto and 700 in Arabic. We initialise all parameters with random uniform distribution between -0.001 and 0.001, and we use Batch Normalisation before each non-linear activation function and after each fully-connected layer (Ioffe and Szegedy, 2015) on the encoder side (Vougiouklis et al., 2017). During training, the model tries to learn those parameters that minimise the sum of the negative log-likelihoods of a set of predicted summaries. The networks are trained using mini-batch of size 85. The weights are updated using Adam (Kingma and Ba, 2014) (i.e. it was found to work better than Stochastic Gradient Descent, RMSProp and AdaGrad) with a learning rate of 10^{-5}. An l_2 regularisation term of 0.1 over each network's parameters is also included in the cost function.

The networks converge after the 9th epoch in the Esperanto case and after the 11th in the Arabic case. During evaluation and testing, we do beam search with a beam size of 20, and we retain only the summary with the highest probability. We found that increasing the beam size resulted not only in minor improvements in terms of performance but also in a greater number of fully-completed generated summaries (i.e. summaries for which the special end-of-summary <end> to-

---

[1]Torch is a scientific computing package for Lua. It is based on the LuaJIT package.

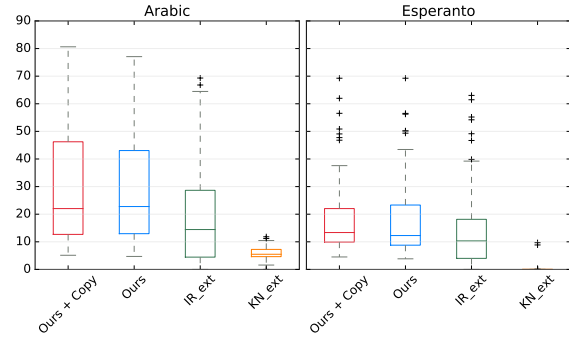|	Arabic	Esperanto
Avg. # of Tokens per Summary	28.1 (±28.8)	26.4 (±22.7)
Avg. # of Triples per Summary	8.1 (±11.2)	11.0 (±13.8)
Avg. # of Linked Named Entities	2.2 (±1.0)	2.4 (±1.1)
Avg. # of Aligned Triples	0.1 (±0.4)	0.2 (±0.5)
Vocabulary Size	344,827	226,447
Total # of Summaries	255,741	126,714

Table 3: Dataset statistics in Arabic and Esperanto.

Figure 2: A box plot showing the distribution of BLEU 4 scores of all systems for each category of generated summaries.

ken is generated).

3 Dataset

In order to train our models to generate summaries from Wikidata triples, we introduce a new dataset for text generation from KB triples in a multilingual setting and align it with the triples of its corresponding Wikidata Item. For each Wikipedia article, we extract and tokenise the first introductory sentence and align it with triples where its corresponding item appears as a subject or an object in the Wikidata truthy dump. In order to create the *surface form tuples* (i.e. Section 2.3), we identify occurrences of entities in the text along with their verbalisations. We rely on keyword matching against labels from Wikidata expanded by the global language fallback chain introduced by Wikimedia[2] to overcome the lack of non-English labels in Wikidata (Kaffee et al., 2017).

For the *property placeholders*, we use the distant supervision assumption for relation extraction (Mintz et al., 2009). Entities that participate in relations with the main entity of the article are being replaced with their corresponding property placeholder tag. Table 3 shows statistics on the two corpora that we used for the training of our systems.

4 Baselines

To demonstrate the effectiveness of our approach, we compare it to two competitive systems.

KN is a 5-gram Kneser-Ney (KN) (Heafield et al., 2013) language model. KN has been used before as a baseline for text generation from structured data (Lebret et al., 2016) and provided competitive results on a single domain in English. We

also introduce a second KN model (KN_{ext}), which is trained on summaries with the special tokens for copy actions. During test time, we use beam search of size 10 to sample from the learned language model.

IR is an Information Retrieval (IR) baseline similar to those that have been used in other text generative tasks (Rush et al., 2015; Du et al., 2017). First, the baseline encodes the list of input triples using TF-IDF followed by LSA (Halko et al., 2011). For each item in the test set, we perform K-nearest neighbors to retrieve the vector from the training set that is the closest to this item and output its corresponding summary. Similar to KN baseline, we provide two versions of this baseline IR and IR_{ext}.

5 Results and Discussion

We evaluate the generated summaries from our model and each of the baselines against their original counterparts from Wikipedia. Triples sets whose generated summaries are incomplete[3] (i.e. summaries for which the special end-of-summary `<end>` token is generated) are excluded from the evaluation. We use a set of evaluation metrics for text generation: BLEU (Papineni et al., 2002), METEOR (Denkowski and Lavie, 2014) and $ROUGE_L$ (Lin, 2004). As displayed in Table 4, our model shows a significant enhancement compared to our baselines across the ma-

---

[3]Around ≤ 1% and 2% of the input validation and test triples sets in Arabic and Esperanto respectively led to the generation of summaries without the `<end>` token. We believe that this difference is explained by the limited size of the Esperanto dataset that increases the level of difficulty that the trained models (i.e. with or without Copy Actions) to generalise on unseen data.

	Model	BLEU 1		BLEU 2		BLEU 3		BLEU 4		ROUGE$_L$		METEOR	
		Valid.	Test	Valid.	Test	Valid.	Test	Valid.	Test	Valid.	Test	Valid.	Test
Arabic	KN	12.84	12.85	2.28	2.4	0.95	1.04	0.54	0.61	17.08	17.09	29.04	29.02
	KN$_{ext}$	28.93	28.84	21.21	21.16	16.78	16.76	13.42	13.42	28.57	28.52	30.47	30.43
	IR	41.39	41.73	34.18	34.58	29.36	29.72	25.68	25.98	43.26	43.58	32.99	33.33
	IR$_{ext}$	49.87	48.96	42.44	41.5	37.29	36.41	33.27	32.51	51.66	50.57	34.39	34.25
	Ours	53.61	54.26	47.38	48.05	42.65	43.32	38.52	39.20	64.27	64.64	45.89	45.99
	+ Copy	**54.10**	**54.40**	**47.96**	**48.27**	**43.27**	**43.60**	**39.17**	**39.51**	**64.60**	**64.69**	**46.09**	**46.17**
Esperanto	KN	18.12	17.8	6.91	6.64	4.18	4.0	2.9	2.79	37.48	36.9	31.05	30.74
	KN$_{ext}$	25.17	24.93	16.44	16.3	11.99	11.92	8.77	8.79	44.93	44.77	33.77	33.71
	IR	43.01	42.61	33.67	33.46	28.16	28.07	24.35	24.3	46.75	45.92	20.71	20.46
	IR$_{ext}$	**52.75**	**51.66**	43.57	42.53	37.53	36.54	33.35	32.41	58.15	57.62	31.21	31.04
	Ours	49.34	49.40	42.83	42.95	38.28	38.45	34.66	34.85	66.43	**67.02**	40.62	**41.13**
	+ Copy	50.22	49.81	**43.57**	**43.19**	**38.93**	**38.62**	**35.27**	**34.95**	**66.73**	66.61	**40.80**	40.74

Table 4: Automatic evaluation of our model against all other baselines using BLEU 1-4, ROUGE and METEOR for both Arabic and Esperanto Validation and Test set

jority of the evaluation metrics in both languages. We achieve at least an enhancement of at least 5.25 and 1.31 BLEU 4 score in Arabic and Esperanto respectively over the IR$_{ext}$, the strongest baseline. The introduction of the copy actions to our encoder-decoder architecture enhances our performance further by $0.61 - 1.10$ BLEU (using BLEU 4). In general, our copy actions mechanism benefits the performance of all the competitive systems.

Generalisation Across Domains. To investigate how well different models can generalise across multiple domains, we categorise each generated summary into one of 50 categories according to its main entity instance type (e.g. village, company, football player). We examine the distribution of BLEU-4 scores per category to measure how well the model generalises across domains (Figure 2). We show that i) the high performance of our system is not skewed towards some domains at the expense of others, and that ii) our model has a good generalisation across domains – better than any other baseline. Despite the fact that the Kneser-Ney template-based baseline (KN$_{ext}$) has exhibited competitive performance in a single-domain context (Lebret et al., 2016), it is failing to generalise in our multi-domain text generation scenario.

6 Conclusions

In this paper, we show that with the adaptation of the encoder-decoder neural network architecture for the generation of summaries we are able to overcome the challenges introduced by working with underserved languages. This is achieved by leveraging data from a structured knowledge base and careful data preparation in a multilingual fashion, which are of the utmost practical interest for our under-resourced task, that would have otherwise required a substantial additional amount of data. Our model was able to perform and generalise across domains better than a set of strong baselines.

Acknowledgements

References

Andrew Chisholm, Will Radford, and Ben Hachey. 2017. Learning to generate one-sentence biographies from Wikidata. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Association for Computational Linguistics, Valencia, Spain, pages 633–642.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR* abs/1406.1078.

Michael J. Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation, WMT@ACL 2014, June 26-27, 2014, Baltimore, Maryland, USA*. pages 376–380.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*. pages 1342–1352.

Caglar Gulcehre, Marcin Moczulski, Misha Denil, and Yoshua Bengio. 2016. Noisy activation functions. In *International Conference on Machine Learning*. pages 3059–3068.

Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. 2011. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review* 53(2):217–288.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 2: Short Papers*. pages 690–696.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*. PMLR, Lille, France, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456.

Lucie-Aimée Kaffee, Alessandro Piscopo, Pavlos Vougiouklis, Elena Simperl, Leslie Carr, and Lydia Pintscher. 2017. A Glimpse into Babel: An Analysis of Multilinguality in Wikidata. In *Proceedings of the 13th International Symposium on Open Collaboration*. ACM, page 14.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980. http://arxiv.org/abs/1412.6980.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*. pages 1203–1213.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*. Barcelona, Spain, volume 8.

Thang Luong, Ilya Sutskever, Quoc V. Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*. pages 11–19.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore*. pages 1003–1011.

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 on Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '02, pages 311–318. https://doi.org/10.3115/1073083.1073135.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*. pages 379–389.

Iulian Vlad Serban, Alberto García-Durán, Çaglar Gülçehre, Sungjin Ahn, Sarath Chandar, Aaron C. Courville, and Yoshua Bengio. 2016. Generating factoid questions with recurrent neural networks: The 30m factoid question-answer corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Pavlos Vougiouklis, Hady ElSahar, Lucie-Aimée Kaffee, Christophe Gravier, Frédérique Laforest, Jonathon S. Hare, and Elena Simperl. 2017. Neural wikipedian: Generating textual summaries from knowledge base triples. *CoRR* abs/1711.00155.