Mind the (Language) Gap: Generation of Multilingual Wikipedia Summaries from Wikidata for ArticlePlaceholders

Lucie-Aimée Kaffee[1](✉), Hady Elsahar[2], Pavlos Vougiouklis[1], Christophe Gravier[2], Frédérique Laforest[2], Jonathon Hare[1], and Elena Simperl[1]

[1] School of Electronics and Computer Science,
University of Southampton, Southampton, UK
{pv1e13,jsh2,e.simperl}@ecs.soton.ac.uk, kaffee@soton.ac.uk

[2] Laboratoire Hubert Curien, CNRS UJM-Saint-Étienne,
Université de Lyon, 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. It is therefore of utmost social and cultural importance to focus efforts on languages whose speakers only have access to limited Wikipedia content. We investigate supporting communities by generating summaries for Wikipedia articles in underserved languages, given structured data as an input.

We focus on an important support for such summaries: ArticlePlaceholders, a dynamically generated content pages in underserved Wikipedias. They enable native speakers to access existing information in Wikidata. To extend those ArticlePlaceholders, we provide a system, which processes the triples of the KB as they are provided by the ArticlePlaceholder, and generate a comprehensible textual summary. This data-driven approach is employed with the goal of understanding how well it matches the communities' needs on two underserved languages on the Web: Arabic, a language with a big community with disproportionate access to knowledge online, and Esperanto, an easily-acquainted, artificial language whose Wikipedia content is maintained by a small but devoted community. With the help of the Arabic and Esperanto Wikipedians, we conduct a study which evaluates not only the quality of the generated text, but also the usefulness of our end-system to any underserved Wikipedia version.

Keywords: Multilinguality · Wikipedia · Wikidata
Natural language generation · Esperanto · Arabic · Neural networks

L.-A. Kaffee, H. Elsahar and P. Vougiouklis—These authors contributed equally to this work.

A. Gangemi et al. (Eds.): ESWC 2018, LNCS 10843, pp. 319–334, 2018.
https://doi.org/10.1007/978-3-319-93417-4_21

1 Introduction

Despite the fact that Wikipedia exists in 287 languages, its 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 in the large Wikipedia communities. Part of this problem has been addressed by Wikidata, the KB supporting Wikipedia with structured data in a cross-lingual manner. Recently, Wikimedia introduced **ArticlePlaceholders** [12] in order to integrate Wikidata's knowledge into the Wikipedias of underserved languages and help in reducing the language gap. ArticlePlaceholders display Wikidata triples in a tabular-based way in the target Wikipedia language and are currently deployed to 11 underserved Wikipedias[1]. When a user searches for a topic on Wikipedia that has a Wikidata item, but no Wikipedia article yet, they are led to the ArticlePlaceholder[2] on the topic. Compared to stub articles[3], ArticlePlaceholders have the advantage of being dynamically updated in real time to accommodate information changes in Wikidata. This means less maintenance for small communities of editors. Since Wikidata is one central, language-independent place to edit information and each item or property has to be translated only once, any contribution in Wikidata has an impact on the ArticlePlaceholders. For example, an editor speaking only English can connect the existing items Q1299 (*The Beatles*) with the item Q145 (*United Kingdom*) via the property P495 (*country of origin*). This will automatically add the same triple with their Esperanto labels: *The Beatles – eldonit/ata en – Unuiĝinta Reĝlando*. Nonetheless, ArticlePlaceholders currently only display information in the form of tables.

In this paper, we propose an automatic approach to enrich ArticlePlaceholders with textual summaries that can serve as a starting point for the Wikipedia editors to write their article. The summaries resemble the first sentence of a Wikipedia article, that gives a reader an overview of the topic. We pose the following research questions:

RQ1. Given the challenges concerning underserved languages, can we generate textual summaries that match the quality and style of Wikipedia content?

RQ2. Can we generate summaries that are useful for Wikipedia editors of underserved language communities?

We adapt an end-to-end trainable model, which generates a monolingual textual summary (i.e. only in English) given a set of KB triples as input, for multilingual support. To this end, we introduce a new "property placeholders" feature and put them under distant supervision in order to enable our system to verbalise even rare or "unseen" entities. Since the summaries are generated explicitly based on

[1] cy, eo, lv, nn, ht, kn, nap, gu, or, sq, and bn.

[2] Example as of online now, without the integration of generated summaries: https://gu.wikipedia.org/wiki/special:AboutTopic/Q7186.

[3] https://en.wikipedia.org/wiki/Wikipedia:Stub.

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

Page statistic	Esperanto	Arabic	English	Wikidata
Articles	241,901	541,166	5,483,928	37,703,807
Avg edits/page	11.48	8.94	21.11	14.66
Active users	2,849	7,818	129,237	17,583
Vocab. size	1.5M	2.2M	2.0M	–

the input triples, potential changes in the respective triples can manifest themselves immediately to the textual content of the summary without the inclusion of the translation loop. Furthermore, since we do not transfer any information from a source language, our model learns to generate Wikipedia content that captures the linguistic peculiarities of our target underserved Wikipedias.

We apply our model on two languages that have a severe lack of both editors and articles on Wikipedia: Esperanto and Arabic. Esperanto is an artificially created language, with an easy acquisition, which makes it a suitable starting point to explore challenges of our task. On the other hand, Arabic is a morphologically rich language with a significantly larger vocabulary. Arabic is the 5th most spoken language in the world [8], however as shown in Table 1 the Arabic Wikipedia suffers a severe lack of content compared to the English.

We propose a novel evaluation framework that assesses the usefulness of the summaries via a multitude of metrics, computed against strong baselines and involving readers and editors of underserved Wikipedias. We start our evaluation by measuring how close our synthesized summaries are to actual summaries in Wikipedia. We compare our model to two strong baselines of different natures: MT and a template-based solution. Our model substantially outperforms the baselines in all evaluation metrics in both Esperanto and Arabic. In addition, we developed three studies with the Wikipedia community, in which we ask for their feedback about the generated summaries, in terms of their fluency, appropriateness for Wikipedia, and engagement with editors. We believe that given the promising results achieved in the automatic and human evaluations, our approach along with the datasets, the baselines, and the experimental design of the human evaluation can serve as a starting point for the research community to further improve and assist in solving this critical task. Our code and experiments are available: https://github.com/pvougiou/Mind-the-Language-Gap.

2 Related Work

Multilingual Text Generation. Many existing techniques for text generation and RDF verbalization rely on templates. These templates are generated using linguistic features such as grammatical rules [26], or are hand-crafted [7]. These approaches face many challenges when scaling for a language-independent system, as templates need to be fine-tuned to any new languages they are ported to. This is especially difficult for the few editors of underserved Wikipedias since templates need extra attention. They would have to create and maintain templates while this time could be invested in the creation of an actual article. Recognizing this problem, the authors of [5,6] introduce a distant-supervised approach to verbalize triples. The templates are learned from existing Wikipedia articles. This makes the approach more suitable for language-independent tasks. However, templates always assume that items will always have the appropriate triples to fill the slots of the template. This assumption is not always necessarily true. In our experiments, we implement a template-learning baseline and we show that adapting to the varying triples available can achieve better performance.

Text Generation for Wikipedia. Pochampally et al. and Sauper et al. proposed the generation of Wikipedia summaries by harvesting sentences from the Internet [20,23]. Existing Wikipedia articles are used to automatically derive templates for the topic structure of the summaries and the templates are afterward filled using Web content. Such approaches are limited to only one or two domains and only in English. The lack of Web resources for underserved languages prevents these approaches to scale to undeserved languages in multiple domains [16]. Meanwhile, KBs have been used as a resource for NLG [2,5,19,25]. These techniques leverage linguistic information from KBs to build a dataset of triples aligned with equivalent sentences from Wikipedia. This alignment is used at subsequent steps to train NLG systems.

The most relevant work to our proposed model are the recent approaches by Lebret et al. [15], Chisholm et al. [2], and Vougiouklis et al. [25], who all propose adaptations of the general encoder-decoder neural network framework [3,24]. They use structured data from Wikidata and DBpedia as input and generate one sentence summaries that match the Wikipedia style in English in only a single domain. The first sentence of Wikipedia articles in a single domain exhibit a relatively narrow domain of language in comparison to other text generation tasks such as translation. However, Chisholm et al. [2] show that this task is still challenging and far from being solved. In contrast with these works, in our paper we extend those research work to include open-domain, multilingual summaries.

Evaluating Text Generation. Evaluating generated text is challenging and there have been different approaches proposed by the literature. Automatic scores [15], expert evaluation and crowdsourcing [2,14] have been employed. Additionally, similar to Sauper and Barzilay [23], we extend our evaluation to usefulness of the summaries for Wikipedia editors by measuring the amount of reuse of the generated summaries. This concept has been widely investigated in fields such as journalism [4] and plagiarism detection [21].

Table 2. The ArticlePlaceholder provides our system with a set of triples about *Floridia*, whose either subject or object is related to the item of Floridia. Subsequently, our system summarizes the input set of triples as text. We train our model using the summary with the extended vocabulary.

ArticlePlaceholder triples	f_1: `Q490900` (Floridia)	`P17` (ŝtato)	`Q38` (Italio)
	f_2: `Q490900` (Floridia)	`P31` (estas)	`Q747074` (komunumo de Italio)
	f_3: `Q30025755` (Floridia)	`P1376` (ĉefurbo de)	`Q490900` (Floridia)
Textual summary	Floridia estas komunumo de Italio		
Vocab. extended summary	`[[Q490900, Floridia]]` estas komunumo de `[[P17]]`		

3 Methods

We use a neural network in order to understand the impact of adding automatically generated text to ArticlePlaceholders in underserved language Wikipedias.

3.1 Our System

Our system is adapted from our encoder-decoder architecture introduced in [25] that has already been used on a similar text generative task. The architecture of the generative model is displayed in Fig. 1. The encoder is a feed-forward architecture which encodes an input set of triples into a vector of fixed dimensionality. This is used at a later stage to initialise the decoder. The decoder is an RNN that uses Gated Recurrent Units (GRUs) [3] to generate the textual summary one token at a time.

An example is presented in Table 2. The ArticlePlaceholder provides our system with a set of triples about the Wikidata item of *Floridia* (i.e. `Q490900` (Floridia) is either the subject or the object of the triples in the set). Figure 1 displays how our model generates a summary from those triples, f_1, f_2, and f_3. A vector representation h_{f_1}, h_{f_2}, and h_{f_3} for each of the input triples is computed by processing their subject, predicate and object. These vector representations are used to compute a vector representation for the whole input set h_{F_E}. h_{F_E}, along with the special start-of-summary <`start`> token, are used to initialise the decoder that sequentially predicts tokens ("`[[Q490900, Floridia]]`", "estas", "komunumo" etc.).

Formally, let F_E be the set of triples provided by the ArticlePlaceholder for the item E (i.e. item E is either the subject or the object of the triples in the set), our goal is to learn a model that generates a summary Y_E about E. We regard Y_E as a sequence of T tokens such that $Y_E = y_1, y_2, \ldots, y_T$ and compute the conditional probability $p(Y_E|F_E)$:

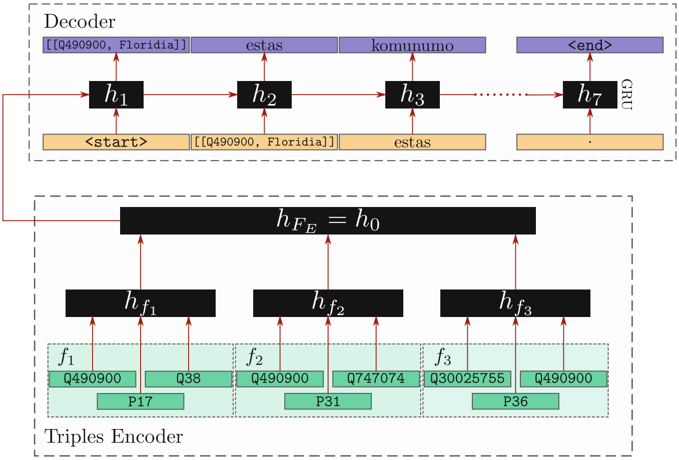

Fig. 1. The triple encoder computes a vector representation for each one of the three input triples from the ArticlePlaceholder, h_{f_1}, h_{f_2} and h_{f_3}. Subsequently, the decoder is initialized using the concatenation of the three vectors, $[h_{f_1}; h_{f_2}; h_{f_3}]$. The purple boxes represent the tokens of the generated summary. Each summary starts and ends with the respective start-of-summary <start> and end-of-summary <end> tokens. (Color figure online)

$$p(Y_E|F_E) = \prod_{t=1}^{T} p(y_t|y_1, \ldots y_{t-1}, F_E) \quad . \tag{1}$$

3.1.1 Generating a Summary

Our model learns to make a prediction about the next token by using the negative cross-entropy criterion. We define a maximum number of triples per summary. Input sets with fewer triples are padded with zero vectors, which are consistently ignored by the encoder. During training our architecture predicts the sequence of tokens that make up the summary. During testing, the ArticlePlaceholder provides our model with a set of unknown triples. After the vector representation h_{F_E} for the unknown set of triples is computed, we initialize the decoder with a special start-of-sequence <start> token.

We adopt a beam-search decoder [15,24,25] which provides us with B-most-probable summaries for each triple set F_E.

3.1.2 Vocabulary Extensions

Each summary consist of words and mentions of named entities. Mapping those entities to words is hard since an entity can have several surface forms and the system may face rare/unseen entities at prediction time. We adopt the concept of *surface form tuples* to learn a number of different verbalisations of the same entity in the summary [25]. In Table 2, `[[Q490900, Floridia]]` in the vocabulary extended summary is an example of a surface form tuple where the entity Q490900 is associated with the surface form of "Floridia".

Additionally, we address the problem of learning embeddings for rare entities in text [17] by training our model to match the occurrence of rare entities in the text to the corresponding triple. To this end, we introduce *property placeholders*. The property placeholders are inspired by the *property-type placeholders* [25]. However, their applicability is much broader since they do not require any instance type-related information about the entities that appear in the triples. In the vocabulary extended summary of Table 2, [[P17]] is an example of property placeholder. In case it is 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)).

Further details regarding the fundamental components of our neural architecture, such as the triples encoder and the surface form tuples, can be found in our previous work [25].

4 Training and Automatic Evaluation

In this section, we describe the dataset that we built for our experiments along with the results of the automatic evaluation of our neural network architecture against the baselines.

4.1 Dataset

In order to train and evaluate our system, we created a new dataset for text generation from KB triples in a multilingual setting. We wish to explore the robustness of our approach to variable datasets with respect to language complexity and size of available training data. Consequently, we worked with two linguistically distinct Wikipedias of different sizes (see Table 1) and different language support in Wikidata [13].

This dataset aligns Wikidata triples with the first, introductory sentence of its corresponding Wikipedia articles. For each Wikipedia article, we extracted and tokenized the first sentence using a multilingual Regex tokenizer from the NLTK toolkit [1]. Afterwards, we retrieved the corresponding Wikidata item to the article and queried all triples where the item appeared as a subject or an object in the Wikidata truthy dump[4].

In order to create the *surface form tuples* (i.e. Sect. 3.1.2), we identify occurrences of entities in the text along with their verbalisations. We rely on keyword matching against labels from Wikidata from the corresponding language, due to the lack of reliable entity linking tools for underserved languages.

For the *property placeholders* (described in more detail in Sect. 3.1.2), we use the distant-supervision assumption for relation extraction [18]. After identifying the rare entities that participate in relations with the main entity of the article, they are replaced from the introductory sentence with their corresponding

[4] https://dumps.wikimedia.org/wikidatawiki/entities/.

property placeholder tag (e.g. `[[P17]]` in Table 2). During testing, any property placeholder token that is generated by our system is replaced by the label of the entity of the relevant triple (i.e. triple with the same property as the generated token).

4.2 Automatic Evaluation

To evaluate how well our system generates textual summaries for Wikipedia, we evaluated the generated summaries against two baselines on their original counterparts from Wikipedia. We use a set of evaluation metrics for text generation BLEU 1, BLEU 2, BLEU 3, BLEU 4, METEOR and $\mathrm{ROUGE_L}$. BLEU calculates n-gram precision multiplied by a brevity penalty which penalizes short sentences to account for word recall. METEOR is based on the combination of uni-gram precision and recall, with recall weighted over precision. It extends BLEU by including stemming, synonyms and paraphrasing. $\mathrm{ROUGE_L}$ is a recall-based metric which calculates the length of the most common subsequence between the generated summary and the reference.

4.3 Baselines for Automatic Evaluation

Due to the variety of approaches for text generation, we demonstrate the effectiveness of our system by comparing it against two baselines of different nature. Both baselines are reproducible and the code is provided in the GitHub repo.

Machine Translation (MT). For the MT baseline, we used Google Translate on English Wikipedia summaries. Those translations are compared to the actual target language's Wikipedia entry. This limits us to articles that exist in both English and the target language. In our dataset, the concepts in Esperanto and Arabic that are not covered by English Wikipedia account for 4.3% and 30.5% respectively. This indicates the content coverage gap between different Wikipedia languages [10].

Template Retrieval (TP). Similar to template-based approaches for text generation [6,22], we build a template-based baseline that retrieves an output summary from the training data based on the input triples. First, the baseline encodes the list of input triples that corresponds to each summary in the training/test sets into a sparse vector of TF-IDF weights [11]. Afterwards, it performs LSA [9] to reduce the dimensionality of that vector. Finally, for each item in the test set, we employ the K-nearest neighbors algorithm to retrieve the vector from the training set that is the closest to this item. The summary that corresponds to the retrieved vector is used as the output summary for this item in the test set. We provide two versions of this baseline. The first one (TP) retrieves the raw summaries from the training dataset. The second one (TP_{ext}) retrieves summaries with the special tokens for vocabulary extension. A summary can act as a template after replacing its entities with their corresponding *Property Placeholders* (see Table 2).

Table 3. Participation numbers: total number of participants (P), total number of sentences (S), number of P that evaluated at least 50% of S, and average number of S evaluated per P.

		#P	#S	#P: S > 50%	Avg. #S/P	All ann.
Arabic	Fluency	27	60	5	15.03	406
	Approp	27	60	5	14.78	399
	Editors	7	30	2	4	33
Esper.	Fluency	27	60	3	8.7	235
	Approp	27	60	3	8.63	233
	Editors	8	30	2	4.75	38

5 Community Study

Automatic measures of text quality such as BLEU can give an indication of how close a generated text is to the source of a summary. Complementary, working with humans is generally more trusted when it comes to quality evaluation of generated text, and captures the direct response of the community. We ran a community study for a total of 15 days to answer our research questions. To address the question whether the textual summaries can match the quality of Wikipedia (RQ1), we define text quality as fluency and appropriateness. Fluency describes the quality in terms of understandability and grammatical correctness. Appropriateness describes how well a summary fits into Wikipedia, i.e. whether a reader can identify it as part of a Wikipedia article. We assess editors reuse to answer whether we can generate summaries that are useful for Wikipedia editors (RQ2). Our evaluation targets two different communities: (1) *readers*: Any speaker of Arabic and Esperanto, that reads Wikipedia, independent of their activity on Wikipedia, and (2) *editors*: any active contributor to Arabic and Esperanto Wikipedia. Readers were asked to fill one survey combining fluency and appropriateness. Editors were also asked to fill an additional survey[5]. To sample only participants with previous activity on Wikipedia, we asked them for their reading and editing activity on Wikipedia. The survey instructions[6] and announcements[7] were translated in Arabic and Esperanto.

Recruitment. For the recruitment of readers, we wanted to reach fluent speakers of the language. For Arabic, we got in contact with Arabic speaking researchers from research groups working on Wikipedia related topics. For Esperanto, as there are fewer speakers and they are harder to reach, we promoted the survey

[5] Example questions: https://github.com/pvougiou/Mind-the-Language-Gap/tree/master/crowdevaluation/Examples.

[6] All instructions for the surveys: https://tinyurl.com/y7cgmesk.

[7] https://github.com/luciekaffee/Announcements.

Table 4. Automatic evaluation of our model against all other baselines using BLEU 1–4, ROUGE and METEOR on the validation and the test set for both Arabic and Esperanto.

	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	MT	31.12	33.48	19.31	21.12	12.69	13.89	8.49	9.11	29.96	30.51	31.05	30.1
	TP	41.39	41.73	34.18	34.58	29.36	29.72	25.68	25.98	43.26	43.58	32.99	33.33
	TP_{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	**53.18**	**52.94**	**45.86**	**45.64**	**40.38**	**40.21**	**35.7**	**35.55**	**57.9**	**57.99**	**39.22**	**39.37**
Esperanto	MT	5.35	5.47	1.62	1.62	0.59	0.56	0.26	0.23	4.67	4.79	0.66	0.68
	TP	43.01	42.61	33.67	33.46	28.16	28.07	24.35	24.3	46.75	45.92	20.71	20.46
	TP_{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	**56.51**	**56.96**	**47.72**	**48.1**	**41.8**	**42.13**	**37.24**	**37.52**	**64.36**	**64.69**	28.35	28.76

on social media such as Twitter and Reddit[8] using the researchers' accounts. For the recruitment of editors, we posted on the editors' mailing-lists[9]. Additionally, for Esperanto we posted on the Wikipedia discussion page[10]. The Arabic editors survey was also promoted at WikiArabia, the conference for the Arabic speaking Wikipedia community. The numbers of participation in all surveys can be found in Table 3.

Fluency. We answer whether we can generate summaries that match the quality and style of Wikipedia content in a study with 54 Wikipedia readers from two different Wikipedia languages. We created a corpus consisting of 60 summaries of which 30 are generated through our approach, 15 are from news, 15 from Wikipedia summaries of the training dataset. For news in Esperanto, we chose introduction sentences of articles in the Esperanto version of Le Monde Diplomatique[11]. For news in Arabic, we chose introduction sentences of the RSS feed of BBC Arabic[12]. Each participant was asked to assess the fluency of the text. We employ a scale from 0 to 6, where: **(6) Excellent**: the given sentence has no grammatical flaws and the content can be understood with ease; **(3) Moderate**: the given sentence is understandable, but has minor grammatical issues; **(0) Non-understandable**: the given sentence cannot be understood. For each sentence, we calculate the mean quality given by all participants and then averaging over all summaries in each corpus.

Appropriateness. As we used the same survey for both fluency and appropriateness, participants answered questions regarding the appropriateness over the same set of sentences. They were asked to assess whether the displayed sentence

[8] https://www.reddit.com/r/Esperanto/comments/75rytb/help_in_a_study_using_ai_to_create_esperanto/.

[9] Esperanto: eliso@lists.wikimedia.org, Arabic: wikiar-l@lists.wikimedia.org.

[10] https://eo.wikipedia.org/wiki/Vikipedio:Diskutejo/Diversejo#Help_in_a_study_improving_Esperanto_text_for_Editors.

[11] http://eo.mondediplo.com/, accessed 28. September 2017.

[12] http://feeds.bbci.co.uk/arabic/middleeast/rss.xml, accessed 28 Sep 2017.

Table 5. Results for fluency and appropriateness.

		Fluency		**Appropriateness**	
		Mean	**SD**	**Part of Wikipedia**	
Arabic	Ours	4.7	1.2	77%	🟪
	Wikipedia	4.6	0.9	74%	🟩
	News	5.3	0.4	35%	🟦
Esper.	Ours	4.5	1.5	69%	🟪
	Wikipedia	4.9	1.2	84%	🟩
	News	4.2	1.2	52%	🟦

could be part of a Wikipedia article. We test whether a reader can tell the difference from just one sentence whether a text is appropriate for Wikipedia, using the news sentences as a baseline. This gives us an insight on whether the text produced by the neural network "feels" like Wikipedia text (appropriateness). Participants were asked not to use any external tools for this task. Readers have just two options to choose from (Yes and No).

Editors Reuse. We randomly choose 30 items from our test set. For each item, each editor was offered the generated summary and its corresponding set of triples and was asked to write a paragraph of 2 or 3 sentences. Editors had the freedom to copy from the generated summary, or completely work from scratch. We assessed how editors used our generated summaries in their work by measuring the amount of text reuse. To quantify the amount of reuse in text we use the Greedy String-Tiling (GST) algorithm [27]. GST is a substring matching algorithm that computes the degree of reuse or copy from a source text and a dependent one. GST is able to deal with cases when a whole block is transposed, unlike other algorithms such as the Levenshtein distance, which calculates it as a sequence of single insertions or deletions rather than a single block move. Given a generated summary $S = s_1, s_2, \ldots$ and an edited one $D = d_1, d_2, \ldots$, each consisting of a sequence of tokens, GST will identify a set of disjoint longest sequences of tokens in the edited text that exist in the source text (called ***tiles***) $T = \{t_1, t_2, \ldots\}$. It is expected that there will be common stop words appearing in both the source and the edited text. However, we are rather interested in knowing how much of real structure of the generated summary is being copied. Thus, we set minimum match length factor $mml = 3$ when calculating the tiles, s.t. $\forall t_i \in T : t_i \subseteq S \wedge t_i \subseteq D \wedge |t_i| \geq mml$ and $\forall t_i, t_j \in T | i \neq j : t_i \cap t_j = \emptyset$. This means that copied sequences of single or double words will not count in the calculation of reuse. We calculate a reuse score *gstscore* by counting the lengths of the detected tiles, and normalize by the length of the generated summary.

$$gstscore(S, D) = \frac{\sum_{t_i \in T} |t_i|}{|S|} \tag{2}$$

Table 6. Percentage of summaries in each category of reuse. A generated summary (top) and after it is was edited (bottom). Solid lines represent reused tiles, while dashed lines represent overlapping sub-sequences not contributing to the *gstscore*.

	Category	Examples	%
Arabic	WD	.خماسي كلوريد الزرنيخ مركب كيميائي له الصيغة(كلمة ناقصة)، ويكون على شكل بلورات بيضاء خماسي كلوريد الزرنيخ هو مُركب كميائي له الصيغة(AtClu2085)، ويكون على شكل بلورات بيضاء.	45.45%
	PD	بيتش باتوم أوهايو (بالإنجليزية (كلمة ناقصة) Ohio)هي منطقة سكنية تقع في الولايات المتحدة في(كلمة ناقصة.) بيتش باتوم (بالإنجليزية:Beach Batom)هي قرية تقع في الولايات المتحدة الامريكية في برووك كاونتي.	33.33%
	ND	دير علا هي بلدة تقع في جنوب غرب إيران. دير علا، أو بيثر، هي قرية أردنية	21.21%
Esperanto	WD	Zederik estas komunumo en la nederlanda provinco Zuid-Holland. Zederik estas komunumo en la nederlanda provinco Zuid-Hooland kaj estas ĉirkaŭata de la municipoj Lopik kaj Zederik.	78.98%
	PD	Nova Pádua estas municipo en la brazila subŝtato Suda Rio-Grando, kiu havis (manka nombro) loĝantojn en (jaro). Nova Pádua estas municipo en la brazila subŝtato Suda Rio-Grando.	15.79%
	ND	Ibiúna estas municipo de la brazila subŝtato San-Paŭlio, kiu taksis (manka nombro) enloĝantojn en (jaro). Ibiúna estas brazila [[municipo]] kiu troviĝas en la administra unuo [[San-Paŭlo]].	5.26%

We classify each of the edits into three groups according to the *gstscore* as proposed by [4]: (1) **Wholly Derived (WD):** the summary structure has been fully reused in the composition of the editor's text ($gstscore \geq 0.66$); (2) **Partially Derived (PD):** the summary has been partially used ($0.66 > gstscore \geq 0.33$); (3) **Non Derived (ND):** The summary has been changed completely ($0.33 > gstscore$).

6 Results and Discussions

In this section, we will report and discuss our experimental findings with respect to the two research questions.

6.1 Automatic Evaluation

As displayed in Table 4, our model shows a significant enhancement compared to our baselines across the majority of the evaluation metrics in both languages. We achieve a **3.01** and **5.11** enhancement in BLEU 4 score in Arabic and Esperanto respectively over TP_{ext}, the strongest baseline. MT of English summaries is not competitive. We attribute this result to the differences in the way of writing across different Wikipedia languages – this inhibits MT from being sufficient for Wikipedia document generation. The results show that generating language directly from the knowledge base triples is a much more suitable approach.

6.2 Community Study

We present the results of the community study in order to find whether we could generate textual summaries that match the quality and style of Wikipedia (RQ1) and can support editors (RQ2).

Fluency *(Table 5)***.** Overall, the quality of our generated summaries is high (4.7 points in average in Arabic, 4.5 in Esperanto). In Arabic, 63.3% of the summaries were evaluated to have at least 5 (out of 6) in average. In Esperanto, 50% of the summaries have at least a quality of 5 (out of 6) in average, with 33% of all summaries given a score of 6 by all participants. This means the majority of our summaries is highly understandable and grammatically correct. Furthermore, our generated summaries are also considered by participants to have a similar average quality as Wikipedia summaries and news from widely read media organizations.

Appropriateness *(Table 5)***.** 77% (resp. 69%) of the generated Arabic (resp. Esperanto) summaries were categorized as being part of Wikipedia. In comparison, news sentences were identified more likely to not fit. In only 35% (Arabic) and 52% (Esperanto) of cases, readers have mistaken them for Wikipedia sentences. Wikipedia sentences were clearly recognized as such (77% and 84%) with scores that are closely matching the one from the generated summaries from our model. Wikipedia has a certain writing style, that seems to differ clearly from news. Our summaries are able to reflect this writing style, being more likely evaluated as Wikipedia sentences than the news baseline – we can expect the generated summaries to melt seamlessly with other Wikipedia content.

Editors Reuse *(Table 6)***.** Our summaries were highly reused. **79%** of the Arabic generated summaries and **93%** of the Esperanto generated summaries were either wholly (**WD**) or partially (**PD**) reused by editors. For the wholly derived edits, editors tended to copy the generated summary with minimal modifications such as Table 6 subsequences A and B in Arabic or subsequence G in Esperanto. One of the common things that hampers the full reusability are "rare" tokens, (كلمة ناقصة) in Arabic and *(mankas vorto)* in Esperanto. Usually, these tokens are yielded when the output word is not in the model vocabulary, it has not been seen frequently by our model such as names in different languages. As it can be seen in tiles E and D in the Arabic examples in Table 6, editors prefer in those cases to adapt the generated sentences. This can also go as far as making the editor to delete the whole subsentence if it contains a high number of such tokens (subsequence H in Table 6). By examining our generated summaries we find that such missing tokens are more likely to appear in Arabic than in Esperanto (2.2 times more). The observed reusability by editors of the Esperanto generated summaries (78.98% **WD**) in comparison to Arabic (45.45% **WD**) can be attributed to this. This can be explained as follows. First, the significant larger vocabulary size of Arabic, which lowers the probability of a word to be seen by the Arabic model. Second, since the majority of rare tokens are named entities mentioned in foreign languages and since the Latin script of Esperanto is similar to many other languages, the Esperanto model has an advantage over the Arabic one when capturing words representing named entities.

7 Conclusions

We introduce a system that extends Wikipedia's ArticlePlaceholder with multilingual summaries automatically generated from Wikidata triples for underserved language on Wikipedia. We show that with the encoder-decoder architecture that we propose is able to perform better than strong baselines of different natures, including MT and a template-based baseline. We ran a community evaluation study to measure to what extent our summaries match the quality and style of Wikipedia articles, and whether they are useful in terms of reuse by Wikipedia editors. We show that members of the targeted language communities rank our text close to the expected quality standards of Wikipedia, and are likely to consider the generated text as part of Wikipedia. Lastly, we found that the editors are likely to reuse a large portion of the generated summaries, thus emphasizing the usefulness of our approach to its intended audience.

Acknowledgements. This research is partially supported by the Answering Questions using Web Data (WDAqua) project, a Marie Skłodowska-Curie Innovative Training Network under grant agreement No 642795, part of the Horizon 2020 programme.

References

1. Bird, S.: NLTK: the natural language toolkit. In: ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 17–21 July 2006, Sydney, Australia (2006)
2. Chisholm, A., Radford, W., Hachey, B.: Learning to generate one-sentence biographies from Wikidata. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Long Papers, vol. 1, pp. 633–642. Association for Computational Linguistics, Valencia, April 2017
3. Cho, K., van Merrienboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. CoRR abs/1406.1078 (2014)
4. Clough, P.D., Gaizauskas, R.J., Piao, S.S.L., Wilks, Y.: METER: MEasuring TExt reuse. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, 6–12 July 2002, Philadelphia, PA, USA, pp. 152–159 (2002)
5. Duma, D., Klein, E.: Generating natural language from linked data: unsupervised template extraction. In: IWCS, pp. 83–94 (2013)
6. Ell, B., Harth, A.: A language-independent method for the extraction of RDF verbalization templates. In: INLG 2014 - Proceedings of the Eighth International Natural Language Generation Conference, Including Proceedings of the INLG and SIGDIAL 2014 Joint Session, 19–21 June 2014, Philadelphia, PA, USA, pp. 26–34 (2014)

7. Galanis, D., Androutsopoulos, I.: Generating multilingual descriptions from linguistically annotated OWL ontologies: the NaturalOWL system. In: Proceedings of the Eleventh European Workshop on Natural Language Generation, pp. 143–146. Association for Computational Linguistics (2007)
8. Gordon, R.G., Grimes, B.F., et al.: Ethnologue: Languages of the world, vol. 15. SIL International, Dallas (2005)
9. Halko, N., Martinsson, P., Tropp, J.A.: Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions. SIAM Rev. **53**(2), 217–288 (2011)
10. Hecht, B., Gergle, D.: The tower of Babel meets web 2.0: user-generated content and its applications in a multilingual context. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 291–300. ACM (2010)
11. Joachims, T.: A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In: Proceedings of the Fourteenth International Conference on Machine Learning (ICML 1997), Nashville, Tennessee, USA, 8–12 July 1997, pp. 143–151 (1997)
12. Kaffee, L.A.: Generating article placeholders from Wikidata for Wikipedia: increasing access to free and open knowledge. Bachelor's thesis, HTW Berlin (2016)
13. Kaffee, L.A., Piscopo, A., Vougiouklis, P., Simperl, E., Carr, L., Pintscher, L.: A glimpse into Babel: an analysis of multilinguality in Wikidata. In: Proceedings of the 13th International Symposium on Open Collaboration, p. 14. ACM (2017)
14. Kondadadi, R., Howald, B., Schilder, F.: A statistical NLG framework for aggregated planning and realization. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4–9 August 2013, Sofia, Bulgaria, Long Papers, vol. 1, pp. 1406–1415 (2013)
15. Lebret, R., Grangier, D., Auli, M.: 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, 1–4 November 2016, pp. 1203–1213 (2016)
16. Lewis, W.D., Yang, P.: Building MT for a severely under-resourced language: white hmong. In: Association for Machine Translation in the Americas, October 2012
17. Luong, T., Sutskever, I., Le, Q.V., Vinyals, O., Zaremba, W.: 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, 26–31 July 2015, Beijing, China, Long Papers, vol. 1, pp. 11–19 (2015)
18. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: 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, pp. 1003–1011 (2009)
19. Mrabet, Y., Vougiouklis, P., Kilicoglu, H., Gardent, C., Demner-Fushman, D., Hare, J., Simperl, E.: Aligning texts and knowledge bases with semantic sentence simplification (2016)
20. Pochampally, Y., Karlapalem, K., Yarrabelly, N.: Semi-supervised automatic generation of Wikipedia articles for named entities. In: Wiki@ ICWSM (2016)

21. Potthast, M., Gollub, T., Hagen, M., Kiesel, J., Michel, M., Oberländer, A., Tippmann, M., Barrón-Cedeño, A., Gupta, P., Rosso, P., Stein, B.: Overview of the 4th international competition on plagiarism detection. In: CLEF 2012 Evaluation Labs and Workshop, Online Working Notes, Rome, Italy, 17–20 September 2012 (2012)
22. Rush, A.M., Chopra, S., Weston, J.: 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, 17–21 September 2015, pp. 379–389 (2015)
23. Sauper, C., Barzilay, R.: Automatically generating Wikipedia articles: a structure-aware approach. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, ACL 2009, vol. 1, pp. 208–216. Association for Computational Linguistics, Stroudsburg (2009)
24. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems, vol. 27, pp. 3104–3112. Curran Associates, Inc. (2014)
25. Vougiouklis, P., ElSahar, H., Kaffee, L., Gravier, C., Laforest, F., Hare, J.S., Simperl, E.: Neural Wikipedian: generating textual summaries from knowledge base triples. CoRR abs/1711.00155 (2017). http://arxiv.org/abs/1711.00155
26. Wanner, L., Bohnet, B., Bouayad-Agha, N., Lareau, F., Nicklaß, D.: MARQUIS: generation of user-tailored multilingual air quality bulletins. Appl. Artif. Intell. **24**(10), 914–952 (2010)
27. Wise, M.J.: YAP 3: improved detection of similarities in computer program and other texts. ACM SIGCSE Bull. **28**(1), 130–134 (1996)