Synthetic Data Generation for Multilingual Domain-Adaptable Question Answering Systems

Alina Kramchaninova, Arne Defauw
CrossLang
Kerkstraat 106, 9050 Gentbrugge, Belgium
{firstname.lastname}@crosslang.com

Abstract

Deep learning models have significantly advanced the state of the art of question answering systems. However, the majority of datasets available for training such models have been annotated by humans, are open-domain, and are composed primarily in English. To deal with these limitations, we introduce a pipeline that creates synthetic data from natural text. To illustrate the domain-adaptability of our approach, as well as its multilingual potential, we use our pipeline to obtain synthetic data in English and Dutch. We combine the synthetic data with non-synthetic data (SQuAD 2.0) and fine-tune multilingual BERT models on the question answering task. Models trained with synthetically augmented data demonstrate a clear improvement in performance when evaluated on the domain-specific test set, compared to the models trained exclusively on SQuAD 2.0. We expect our work to be beneficial for training domain-specific question-answering systems when the amount of available data is limited.

1 Introduction

Recent advances in tackling the problem of question answering (QA) rely on large-scale, open-domain datasets (Bartolo et al., 2021), annotated by humans and composed primarily in English (e.g. SQuAD 1.0 (Rajpurkar et al., 2016), and SQuAD 2.0 (Rajpurkar et al., 2018)). Despite some indications of poor robustness and generalisation (Bartolo et al., 2021), models trained on such datasets are capable of providing topic-agnostic, general-purpose assistance to their users (Ruder and Sil, 2021).

Nevertheless, most industrial applications of QA systems are domain-specific, and often need to be able to operate in multilingual environments. Data collection and manual composition of datasets for each domain and language is most definitely a laborious task, not to mention that certain domains are of little academic or commercial interest and are only of use for some low-resource communities (Rogers et al., 2021). Moreover, while the current synthetic data generation systems focus on augmenting QA data in the SQuAD format,[1] little research has been done on either the generation of synthetic data from natural plain text, or in multiple languages.

Furthermore, most machine reading comprehension (MRC) benchmarks focus primarily on the creation of questions with multi-word factoid answers (e.g. SQuAD 2.0 pairs each factoid question with a Wikipedia paragraph), as well as unanswerable questions (Liu et al., 2020). However, in a real-world scenario, a QA system should ideally be able to provide a response on semantically complex questions such as "I am an EU citizen living in the UK. What changes for me after Brexit?", and questions containing grammar and spelling errors (e.g. questions asked by a non-native speaker, or containing mistakes caused by dyslexia).

In this work, we introduce a domain-adaptable end-to-end pipeline for generic synthetic data generation that requires no manual textual preprocessing, and allows for the integration of mul-

[1] A tuple (c, q, a) where c refers to the context—text segment in which the answer a to the question q should be found.

tilingual features. We utilise this pipeline to create domain-specific training sets in English (EN) and Dutch (NL) from the web scraped data of the Single Digital Gateway and Your Europe portal,[2] which provides information on rules and procedures for citizens and businesses in the EU, in all European languages. We combine the obtained data with SQuAD 2.0 in English and its machine-translated-into-Dutch version to fine-tune multiple instances of a BERT (Bidirectional Encoder Representations from Transformers) model (Devlin et al., 2019) on the QA task. We then cross-evaluate the performance of these models on the relevant test sets, and observe improvements on the QA task when evaluated on the domain-specific test sets, while remaining competitive against models trained on the SQuAD-only counterparts in both languages.

2 Related Work

Existing approaches to synthetic data generation often view question and answer generation as dual tasks (Tang et al., 2017; Shakeri et al., 2020), where one task can improve the other and vice-versa. Roundtrip consistency (Alberti et al., 2019) is one of the methods that combines question generation and question answering models to, first, generate a question conditioned on a pre-selected answer span and its context, and then match against it an answer predicted by a QA system. If there is a match, the triplet (i.e. context, question and answer) is considered valid.

Cloze generation (Dhingra et al., 2018) is a more intuitive approach: it logically splits a document in ratio of 20:80, with the introduction being the first 20% of the input text. It is assumed that the introduction contains answer candidates that are likely to occur in the remainder of the document. Potential answer candidates are consequently selected by matching multi-word spans between introductory sentences and the rest of the text.

These approaches, however, focus on potential answers that are primarily named entities or noun phrases (Tang et al., 2017; Alberti et al., 2019; Puri et al., 2020; Shakeri et al., 2020). For our use-case, we are interested in finding answers of longer spans that might contain administrative procedures in a multilingual setting (e.g. answers to such questions as "how do I request an international passport?" or "Waar kan ik mijn wagen registreren?" - "Where can I register my car?"). Moreover, we are interested in finding right answers in a document that might contain multiple procedures, i.e. the introduction might not match the subsequent content at all, unlike the assumption of the methods proposed in (Dhingra et al., 2018).

OG	Results tend to be scattered across different websites that often lack any guarantee of quality or reliability, and significant *information gaps* remain in many areas, leaving important questions unanswered
MT	Resultaten zijn meestal verspreid over verschillende websites die vaak geen enkele garantie voor kwaliteit of betrouwbaarheid hebben, en er blijven op veel gebieden aanzienlijke *informatielacunes*, waardoor belangrijke vragen onbeantwoord blijven
OG	information gaps
MT	informatie hiaten

Table 1: Translation of Segments via Google Translate

In this work we propose the use of a combination of question generation (QG), question paraphrasing (QP) and unsupervised filtering methods to solve these limitations of previous work. We present techniques for building models and filtering methods in any language using machine translation (MT). For both QP and QG we rely on a T5 model (Raffel et al., 2019) fine-tuned on the respective downstream task (we refer to Section 3).

With regard to the sub-task of QP, we note that on its own it is not an area of active research, although paraphrasing as a data augmentation technique has been explored in both academic (Witteveen and Andrews., 2019) and applied contexts. For instance, Rasa Open Source,[3] a framework for building chatbots and voice-based virtual assistance, researches paraphrasing as a data augmentation technique, to ensure the recognition and anticipation of different variations of the same intent,[4] as small variations in questions, e.g. the use of synonyms, may yield different answers (Dong et al., 2017).

Although multilingual QA remains a relatively unexplored problem, there exist various datasets for the fine-tuning and evaluation of multilingual

[2] https://ec.europa.eu/growth/single-market/single-digital-gateway_en

[3] https://rasa.com/open-source

[4] https://forum.rasa.com/t/paraphrasing-for-nlu-data-augmentation-experimental/27744

QA systems, such as the human-composed TyDi QA (Clark et al., 2020), or MLQA (Lewis et al., 2020) that was created using translation alignments.

Whereas MT may appear as a possible solution to the scarcity of the data for each domain and language, three issues remain. First, and the most evident one, is the quality of MT output, e.g. such problems as the preservation of the word order of the source language might occur (Clark et al., 2020). The second issue lies in the potential misalignment of answer spans (Carrino et al., 2020; Lee et al., 2018) caused by differences between translations of answer segments within the context, and outside of it (see Table 1 where 'OG' stands for 'original' and 'MT' for 'machine-translated'). The bigram "information gaps" was translated to "informatielacunes" within context, but to "informatie hiaten" as a standalone term.[5] As a consequence, it becomes more difficult to determine the offsets (i.e. the position in the context) of such answer spans, and potentially renders the segment useless. Lastly, it must be noted that even though there exist large language models that can generalise across languages, language similarity (Pires et al., 2019) is an important factor that affects the performance of certain architectures across multiple languages.

3 Methodology

We developed a synthetic data generation pipeline that converts plain text into question answering pairs via the following steps: passage detection, keyword filtering, question generation and question paraphrasing.

3.1 Passage Detection

For our use-case, we extracted text from html pages scraped from the web using the Trafilatura[6] library. Next, a rule-based approach was used to parse plain text into chunks (paragraphs and sentences) that can be used as input for the question generator (see Section 3.3). First, we split the text extracted via the Trafilatura library using the newline delimiter, after which we evaluated the start and end characters of each resulting text chunk: if a chunk ends with a question mark or colon, we concatenated the chunk with the subsequent chunk; if it starts with a character that indicates enumeration (e.g. a dash, an asterisk...), the chunk was concatenated with the previous chunk. Chunks containing less than one sentence were discarded. This rule-based approach discards any processing noise that might have occurred during the extraction of text, and delivers semantically charged, coherent paragraphs.

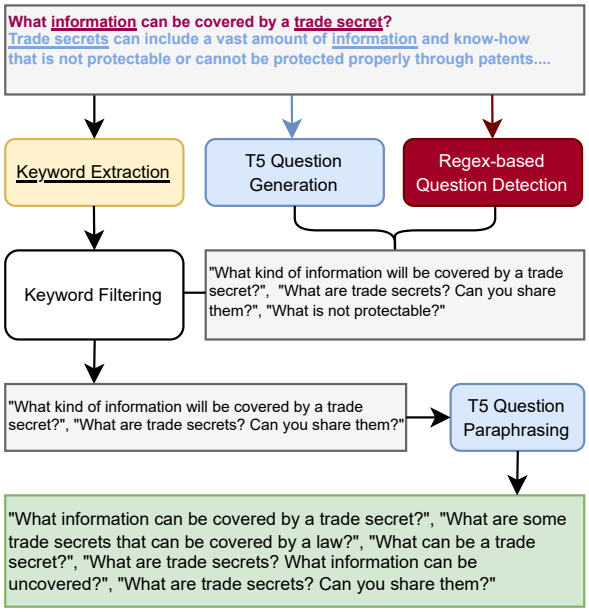

Figure 1: Synthetic Data Generation Pipeline

Consequently, via a sentence-splitter[7] we split the obtained paragraphs into sentences. Both paragraphs and the sentences they contain are fed to the QG model (see Section 3.3): in this way, due to the length differences of sentences and paragraphs, we generate QA pairs of different degrees of complexity. To recreate the SQuAD format for the composition of the synthetic data, for each resulting QA pair, where the input to the QG model is considered the answer, and the output the corresponding question, we also add its context. If the input (i.e. the resulting answer) to the QG model is a paragraph, the context is the document containing that paragraph. If the input is a sentence, the context is the paragraph containing that sentence.

3.2 Keyword Filtering

Once we have obtained the to-be-processed chunks (sentences and paragraphs), we use the YAKE! (Campos et al., 2020) library to extract the most meaningful n-grams from each chunk, one at a

[5]Similarly, in morphologically rich languages, standalone terms could be translated to their base forms while inflected within a context.

[6]https://github.com/adbar/trafilatura

[7]https://pypi.org/project/sentence-splitter

time. The library implements an unsupervised approach that can be applicable to various languages, without a need for external knowledge such as dictionaries or corpora. YAKE! builds upon features extracted from the document (or text chunk in our case) such as casing, word frequency, word relatedness to the document, and how often a candidate n-gram appears within different sentences. YAKE! then heuristically combines these features to calculate a score for each n-gram—the lower the score, the more meaningful the keyword. From this list of generated n-grams, we compute the average score and select the entities with a lower than average score. This final list for each text chunk is cached and used to filter question candidates of the corresponding chunk, after both the QG (see Section 3.3) and QP (see Section 3.4) steps.

3.3 Question Generation

For question generation we used a pre-trained T5 model fine-tuned on the downstream task of QG. For our English pipeline, we used an existing and publicly available T5 based QG model[8]. For QG in Dutch, we fine-tuned a pretrained multilingual T5 model (mT5) (Xue et al., 2020) on the downstream task of QG on the following datasets machine-translated (see Section 3.5) into Dutch[9]: SQuAD 2.0 (Rajpurkar et al., 2018), RACE (Lao et al., 2017), CoQA (Reddy et al., 2019), and MSMARCO (Bajaj et al., 2016). The mT5-Base model pre-trained on 101 languages, is a 580-million parameter model, the fine-tuning of which is very expensive memory-wise. To limit the resources used, we pruned the model by removing the unused vocabulary from other languages than the desired one (Dutch) via an update of the tokenizer and embedding layer.

The potential answers (paragraphs and sentences) obtained via the passage detection step (Section 3.1) are used as input to these T5 based QG models, resulting in a QA pair. By adding the context (i.e. the document if the input chunk is a paragraph, a paragraph if the input chunk is a sentence, also see Section 3.1), we further obtain a synthetic data point in the SQuAD format.

Although sometimes overlooked in the literature, we did not discard questions already present in the web scraped data, but extracted them using a pre-defined regular expression (regex), e.g. "What information can be covered by a trade secret?" in Figure 1. If a question is detected in a given paragraph, it is split into two at the end index of the detected question, and the first part is cached as a question instance, while the second part is considered being the answer to the question.

Your Europe	**EN**	**NL**
Documents	308	171
Total Q before Key. Filt.	57,182	38,751
Q via Regex	20	16
+ QG (sentence)	3,861	439
+ QP (sentence)	18,828	2,080
+ QG (paragraph)	701	86
+ QP (paragraph)	3,900	304
= Total Q after Key. Filt.	27,310	2,925

Table 3: Synthetic data overview.

Q	welke nationaliteit is verantwoordelijk voor sociale zekerheid?
A	Welk land er verantwoordelijk is voor uw sociale zekerheid, dus ook uw gezinstoelagen (kinderbijslag, opvoedingstoelagen, ouderschapsverlof enz.), hangt in de EU af van uw economische situatie en uw woonplaats, niet van uw nationaliteit.

Table 2: Accepted semantically incorrect synthetic question

We then used the keyword filter, described in the previous section, to decide which generated and/or detected questions are kept and eventually paraphrased (see Section 3.4). In other words, if a generated or detected question contains any word from the keyword list, the question is considered valid.

We empirically observed that the quality of the generated questions in Dutch is vastly dependent on the quality of the translation. However, unlike previous work that focuses on evaluating the quality of generated questions (Chen et al., 2020), (Chan and Fan, 2019), in our training set we allow questions that are grammatically incorrect or contain made up or confusing words, e.g. the word "land" (country) was replaced by the word "nationaliteit" (nationality) in Table 2.

3.4 Question Paraphrasing

In a similar way as for the QG sub-task, we used an existing T5 model fine-tuned on the downstream task of QP. For English we used an existing QP

[8] https://huggingface.co/valhalla/t5-small-e2e-qg

[9] See https://huggingface.co/datasets/iarfmoose/question_generator for the original EN dataset.

Type	Text
Context	YES - A medicine available in one EU country might not be sold in another EU country, or it might be sold under a different brand name. When asking for a prescription from your doctor that you intend to dispense in another EU country, you should ensure they use the common name for the prescribed product wherever possible. This will enable a pharmacist in another EU country to prescribe you the equivalent product in that country. To find out if your medicine is available in other EU countries, you can check with your country's national contact point for cross-border healthcare. \n This depends on national law in each European country and will therefore vary throughout the EU. Check with the National Enforcement body in the country concerned or a national consumer centre for more information.\n YES — in all EU countries. Switzerland still applies restrictions on Bulgarian, Croatian and Romanian nationals.\n Ask the host-country liaison office for posted workers. \n Whenever certain conditions have to be fulfilled before you become entitled to health coverage, the national health insurance body examining your claim must take account of periods of insurance, residence or employment completed under the legislation of other EU countries. This ensures that you will not lose your healthcare coverage when changing jobs or moving to another country. \n You can get child benefits from Switzerland or Germany; you won't get full benefits from more than one country. If entitlement in both countries is based on work, even if your children live in yet another country, you will get your benefits from whichever of the two countries where you work that pays the most.
Question	I am unemployed and I come from Bulgaria. Am I allowed to look for work in another EU country and have my benefits transferred there?
Answer	YES — in all EU countries. Switzerland still applies restrictions on Bulgarian, Croatian and Romanian nationals.
$QA_{S,\text{EN-NL}}$	Switzerland or Germany
$QA_{SGP,\text{EN-NL}}$	YES — in all EU countries

Table 4: Your Europe test example.

model[10] fine-tuned on the Quora Question Pairs (QQP) dataset[11]. For Dutch, we fine-tuned a separate mT5 model on the machine-translated QQP dataset.

The detected and/or generated questions that have passed the keyword filter (see Section 3.3) are fed to these QP models individually, without any consideration for the answer or the context. We once again applied the keyword filter to select the most meaningful paraphrased questions.

3.5 Machine Translation

In order to obtain multilingual datasets for the QG and QP task, we rely on transformer-based neural MT models provided via the CEF eTranslation service.[12]. The CEF eTranslation sevice provides translation in 24 official European languages.

4 Experiments

We fine-tuned the multilingual distilled version of BERT (Sanh et al., 2019) (mDistilBERT) on the QA task using the synthetic data obtained using the methods described in Section 3 and the SQuAD 2.0 datasets (we refer to Section 4.1). Full overview of the training data, its sources and size, can be found in Tables 3 and 5. As multilingual BERT models are known to perform better on tasks in English (Riabi et al., 2021), we performed separate experiments with English and Dutch data, as well as experiments with the bilingual data combined. All models were tested on four test sets, two in each language.

4.1 Datasets

Train sets SQuAD 2.0 is a benchmark dataset for question-answering systems. In addition to the 86,821 question answering pairs, the dataset contains 43,498 unanswerable questions. As we are interested in creating a robust QA model that will be able to detect answers in a document, and not interested in unanswerable questions, we omit the latter type of questions, resulting in a non-synthetic training set of length 86,821 for English.

For our Dutch experiments, we used the publicly available machine-translated version of SQuAD 2.0.[13] This dataset contains 53,376 positive and 41,768 negative examples, the latter being omitted.

We further create a synthetic dataset from the web scraped data from the Your Europe portal using the pipeline described in Section 3. In Table 3 we show statistics of our resulting synthetic dataset, and the number of questions (and corresponding answers and context) generated in each step. The second row of Table 3 show the number of documents scraped for both English and Dutch. Next, the total number of questions generated via QG, QP and regex is shown, before filtering via keyword extraction (Total Q before Key. Filt.). The

[10] https://github.com/ramsrigouthamg/Paraphrase-any-question-with-T5-Text-To-Text-Transfer-Transformer-

[11] https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs

[12] https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/eTranslation

[13] https://gitlab.com/niels.rouws/dutch-squad-v2.0

Dataset	$QA_{S, EN}$	$QA_{SG, EN}$	$QA_{SGP, EN}$	$QA_{S, NL}$	$QA_{SGP, NL}$	$QA_{S, EN\text{-}NL}$	$QA_{SGP, EN\text{-}NL}$
S_{EN}	**114,131**	86,821	86,821	-	-	86,821	**59,511**
QG_{EN}	-	**27,310**	4,582	-	-	-	4,582
QP_{EN}	-	-	22,728	-	-	-	22,728
S_{NL}	-	-	-	**56,301**	53,376	53,376	**50,451**
QG_{NL}	-	-	-	-	541	-	541
QP_{NL}	-	-	-	-	2,384	-	2,384
Total	114,131	114,131	114,131	56,301	56,301	140,197	140,197

Table 5: Overview of data composition per trained model. Numbers in bold refer to the randomly oversampled (columns $QA_{S, EN}$, $QA_{SG, EN}$, $QA_{S, NL}$) and undersampled data (column $QA_{SGP, EN\text{-}NL}$).

following rows show the resulting number of questions, after keyword filtering (see Section 3.2), created in each step of the pipeline, both when using sentences and paragraphs as input chunks to the pipeline.

We may observe a difference in the number of generated synthetic questions in English and Dutch. This is primarily caused by the quality of generated and paraphrased questions filtered via keyword extraction: due to the compounding nature of the Dutch language, a great number of questions were filtered out, e.g. if the word "huwelijksaanvraag" (marriage application) is in the original text segment while the generated question might contain the word "huwelijksaangifte" (marriage declaration).

Test sets We evaluate both on the SQuAD 2.0 dataset, and on a domain-specific test set. For the evaluation on SQuAD in English, we held out 5,875 positive examples from the original dataset, while for the evaluation in Dutch, 3,522 positive examples were selected from the machine-translated-into-Dutch SQuAD 2.0 dataset.

To test our pipeline in a setting that would be as close as possible to real-world scenarios, we used a subset of the Your Europe data that was excluded from the training set, and similarly not used as input to the synthetic data generation pipeline. We specifically chose the pages that contained Frequently Asked Questions (FAQ) to retrieve 333 English questions and 265 Dutch questions, and the corresponding answers. These questions were not simplistic call-to-action questions, but mostly compound questions such as "I work in Germany, my husband works in Switzerland, and we live with our children in Austria. Where can we get child benefits from?" The QA pairs were then manually evaluated to ensure that every question is paired with a semantically correct answer.

As the QA pairs were mostly gathered from the FAQ pages of the Your Europe portal, we decided to create an artificial context for each QA pair: we randomly selected five potential answers from other QA pairs, and randomly concatenated them to the single right answer for the given question. An example of such a context and its corresponding QA pair can be seen in Table 4.

4.2 Models

For an objective evaluation of the impact of the different steps of our pipeline for synthetic data generation on the performance of QA models, we have trained several QA models on various combinations of data (see Table 5). In the column 'Dataset' we refer to SQuAD (S) and synthetic training datasets that consist of generated (QG) and paraphrased questions (QP) per language, as indicated in the name of each dataset, we also refer to Table 3.

The model names (first row of Table 5) equally contain the language code of the corresponding dataset, although every fine-tuned QA model uses the same base language model (mDistilBERT) in order to objectively compare the results of each model.

In Table 5, the resulting English QA model $QA_{SGP, EN}$ is trained on both the English SQuAD dataset (86,821 segments=S_{EN}) and the full set of English synthetic data (27,310 segments=QG_{EN}+QP_{EN}), where 'S' stands for SQuAD, 'G' for segments obtained via QG and regex, and 'P' for segments obtained via QP. Similarly, $QA_{S, EN}$ was trained exclusively on the non-synthetic SQuAD training data, randomly oversampled to 114,131 segments to prevent potential differences in performance due to the size of the training data. To analyse the importance of

Context	Bills can be introduced to Parliament in a number of ways; the Scottish Government can introduce new laws or amendments to existing laws as a bill; a committee of the Parliament can present a bill in one of the areas under its remit; a member of the Scottish Parliament can introduce a bill as a private member; or a private bill can be submitted to Parliament by an outside proposer. Most draft laws are government bills introduced by ministers in the governing party. Bills pass through Parliament in a number of stages:
Question	A member of what parliament can introduce a bill as a public member?
$QA_{S,\ EN}$	Scottish
$QA_{SG,\ EN}$	Scottish Government can introduce new laws or amendments to existing laws as a bill ; a committee of the Parliament can present a bill in one of the areas under its remit ; a member of the Scottish Parliament can introduce a bill as a private member
$QA_{SGP,\ EN}$	a member of the Scottish Parliament can introduce a bill as a private member

Table 6: Predictions of different QA models, trained only using SQuAD data ($QA_{S,\ EN}$) and QA models trained on a combination of SQuAD and synthetic data ($QA_{SG,\ EN}$ and $QA_{SGP,\ EN}$), on a segment from the held out EN SQuAD test set.

Model	BLEU	F1	SemSim
$QA_{S,\ EN}$	**0.2033**	**0.2538**	**0.4420**
$QA_{SG,\ EN}$	0.1673	0.2120	0.4138
$QA_{SGP,\ EN}$	0.1789	0.2272	0.4175
$QA_{S,\ EN\text{-}NL}$	**0.2058**	**0.2580**	**0.4382**
$QA_{SGP,\ EN\text{-}NL}$	0.1795	0.2293	0.4219

Table 7: Scores obtained by the various QA models on the held out EN SQuAD test set

Model	BLEU	F1	SemSim
$QA_{S,\ NL}$	0.1866	0.2315	0.4779
$QA_{SGP,\ NL}$	**0.1928**	**0.2369**	**0.4863**
$QA_{S,\ EN\text{-}NL}$	**0.1733**	**0.2132**	**0.4559**
$QA_{SGP,\ EN\text{-}NL}$	0.1478	0.1828	0.4427

Table 8: Scores obtained by the various QA models on the held out NL SQuAD test set

Model	BLEU	F1	SemSim
$QA_{S,\ EN}$	0.0772	0.1165	0.1995
$QA_{SG,\ EN}$	0.1438	0.1898	0.2813
$QA_{SGP,\ EN}$	**0.1557**	**0.1997**	**0.3145**
$QA_{S,\ EN\text{-}NL}$	0.0712	0.1107	0.1734
$QA_{SGP,\ EN\text{-}NL}$	**0.1903**	**0.2588**	**0.4018**

Table 9: Scores obtained by the various QA models on the EN domain-specific (Your Europe) test set

Model	BLEU	F1	SemSim
$QA_{S,\ NL}$	0.0681	0.1033	0.1635
$QA_{SGP,\ NL}$	**0.1650**	**0.2236**	**0.3320**
$QA_{S,\ EN\text{-}NL}$	0.0706	0.1001	0.1429
$QA_{SGP,\ EN\text{-}NL}$	**0.1892**	**0.2556**	**0.3689**

Table 10: Scores obtained by the various QA models on the NL domain-specific (Your Europe) test set

QP as a pipeline feature, we also performed an ablation study, training $QA_{SG,\ EN}$ on SQuAD data in combination with the oversampled synthetic QG_{EN} dataset.

An identical strategy was applied in order to obtain the Dutch QA models $QA_{S,\ NL}$ and $QA_{SGP,\ NL}$. Similarly, for our bilingual models, we combined the English and Dutch versions of SQuAD, and synthetic datasets to train the bilingual $QA_{S,\ EN\text{-}NL}$ and $QA_{SGP,\ EN\text{-}NL}$ models. Note that for a fair comparison of models trained exclusively on SQuAD ($QA_{S,\ EN\text{-}NL}$) with $QA_{SGP,\ EN\text{-}NL}$, we randomly undersampled the English and Dutch SQuAD dataset in this case.

4.3 Metrics

The performance of the various QA models listed in Table 5 was evaluated using the following metrics: sentence BLEU (Papineni et al., 2002), Rouge-L (Lin, 2004) that measures the longest common subsequence to calculate f1-measure, and the cosine similarity calculated using multilingual Sentence-BERT embeddings (Reimers and Gurevych, 2019). We use these metrics to measure the predicted answer against the gold standard answer.

5 Discussion of Results

In this section we compare the performance of the QA models trained on both non-synthetic (i.e. SQuAD) and synthetic data, and models trained exclusively on non-synthetic data. As discussed in Section 4.2, we present results for both English and Dutch. We also evaluate the performance of a bilingual QA model.

In Table 7 we show the scores of our QA models trained on EN and a combination of EN and NL data obtained on the held out EN SQuAD test set. We observe that $QA_{S,\ EN}$ trained on the

EN SQuAD data, achieved the best performance. Nevertheless, despite slightly lower scores, models trained on the combination of SQuAD and synthetic data, do not demonstrate a large regression in performance. This is also illustrated by the example shown in Table 6: we notice that predictions by $QA_{SG, EN}$ and $QA_{SGP, EN}$ tend to be of longer spans, causing this small drop in performance when evaluated on the gold standard answer 'Scottish'. Similar results are obtained for the QA models trained on NL and a combination of EN and NL data (Table 8), although in this case the $QA_{SGP, NL}$ model achieves slightly better scores than the model trained on non-synthetic data only ($QA_{S, NL}$).

More interestingly, in Tables 9 and 10, we present the results on the domain-specific (Your Europe) test sets for EN and NL. We observe that models trained on non-synthetic data only ($QA_{S, EN}$, $QA_{S, NL}$, $QA_{S, EN\text{-}NL}$) demonstrate an overall lower performance compared to the models also trained on synthetic data ($QA_{SG, EN}$, $QA_{SGP, EN}$, $QA_{SGP, NL}$ and $QA_{SGP, EN\text{-}NL}$). Comparing scores achieved by $QA_{SG, EN}$ and $QA_{SGP, EN}$ we can also conclude that adding synthetic segments obtained via QP results in an increase in performance, consistent across all metrics. Finally, from these results we also see that bilingual models trained on synthetic and non-synthetic data achieve better performance than their monolingual version (i.e. $QA_{SGP, EN}$ and $QA_{SGP, NL}$ versus $QA_{SGP, EN\text{-}NL}$).

6 Conclusion

In this paper we presented a novel multilingual domain-adaptable pipeline for the generation of synthetic training data for QA models. Our experiments demonstrate that models trained with synthetic data achieved improved performance on domain-specific test sets that included not solely factual, but semantically complex questions, both in English and Dutch. As our pipeline incorporates two mT5 models fine-tuned on task- and language-specific datasets, we demonstrate that it is possible to make use of MT and apply our approach to any language supported by mT5.

One of the remaining challenges of this approach is the quality monitoring of the generated synthetic questions, especially for languages other than English. It would be useful to experiment with more advanced filtering methods than the method based on keyword extraction proposed in this work. For instance, a semantic similarity feature could potentially detect questions that might not include specific keywords, but also questions containing synonyms of extracted keywords or semantically close paraphrases. We also assume that it would be useful to introduce an additional feature to evaluate the chunks that are processed by our pipeline for synthetic data generation, as not every input paragraph or sentence would serve as an answer to a potential question in a real-world scenario.

Acknowledgements

This work was performed in the framework of the CEFAT4Cities project (2019-EU-IA-0015), funded by the CEF Telecom programme (Connecting Europe Facility).

References

Alberti, Chris, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins 2019. Synthetic QA Corpora Generation with Roundtrip Consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6168–6173, Florence, Italy. Association for Computational Linguistics.

Bajaj, Payal, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *ArXiv*, abs/1611.09268.

Bartolo, Max, Tristan Thrush, Robin Jia, Sebastian Riedel, Pontus Stenetorp, and Douwe Kiela 2021. Improving Question Answering Model Robustness with Synthetic Adversarial Data Generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8830–8848, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Campos, Ricardo, Vítor Mangaravite, Arian Pasquali, Alípio Jorge, Celia Nunes, and Adam Jatowt. 2020. Yake! keyword extraction from single documents using multiple local features. In *Information Sciences*.

Carrino, Casimiro Pio, Marta R. Costa-jussa, José A. R. Fonollosa. 2020. Automatic Spanish Translation of the SQuAD Dataset for Multilingual Question Answering. *ArXiv*, abs/1912.05200.

Chan, Ying-Hong, and Yao-Chung Fan. 2019. A Recurrent BERT-based Model for Question Generation. In *Proceedings of the 2nd Workshop on Machine*

Reading for Question Answering, pages 154–162, Hong Kong, China. Association for Computational Linguistics.

Chen, Yu, Lingfei Wu, Mohammed J. Zaki. 2020. Reinforcement Learning Based Graph-to-Sequence Model for Natural Question Generation. *ArXiv*, abs/1908.04942.

Clark, Jonathan H., Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A Benchmark for Information-Seeking Question Answering. In *Transactions of the Association for Computational Linguistics*, 8:454–470.

Devlin, Jacob, 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, Minneapolis, Minnesota. Association for Computational Linguistics.

Dhingra, Bhuwan, Danish Danish, and Dheeraj Rajagopal. 2018. Simple and Effective Semi-Supervised Question Answering. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Volume 2 (Short Papers), pages 582–587, New Orleans, Louisiana. Association for Computational Linguistics.

Dong, Li, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. 2017. Learning to Paraphrase for Question Answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 875–886, Copenhagen, Denmark. Association for Computational Linguistics.

Klein, Tassilo, and Moin Nabi. 2019. Learning to Answer by Learning to Ask: Getting the Best of GPT-2 and BERT Worlds. *ArXiv*, abs/1911.02365.

Lai, Guokun, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding Comprehension Dataset From Examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Lee, Kyungjae, Kyoungho Yoon, Sunghyun Park, and Seung-won Hwang. 2018. Semi-supervised Training Data Generation for Multilingual Question Answering. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Lewis, Patrick, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: Evaluating Cross-lingual Extractive Question Answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330, Online. Association for Computational Linguistics.

Lin, Chin-Yew. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Liu, Dayiheng, Yeyun Gong, Jie Fu, Yu Yan, Jiusheng Chen, Jiancheng Lv, Nan Duan, and Ming Zhou. 2020. Tell Me How to Ask Again: Question Data Augmentation with Controllable Rewriting in Continuous Space. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5798–5810, Online. Association for Computational Linguistics.

Papineni, Kishore, 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.

Pires, Telmo, Eva Schlinger, and Dan Garrette. 2019. How Multilingual is Multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Puri, Raul, Ryan Spring, Mohammad Shoeybi, Mostofa Patwary, and Bryan Catanzaro. 2020. Training Question Answering Models From Synthetic Data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5811–5826, Online. Association for Computational Linguistics.

Radford, Alec, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners.

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *ArXiv*, abs/1910.10683.

Rajpurkar, Pranav, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Rajpurkar, Pranav, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *ArXiv*, abs/1806.03822.

Reddy, Siva, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A Conversational Question Answering Challenge. *ArXiv*, abs/1808.07042.

Reimers, Nils, and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Riabi, Arij, Thomas Scialom, Rachel Keraron, Benoît Sagot, Djamé Seddah, and Jacopo Staiano. 2021. Synthetic Data Augmentation for Zero-Shot Cross-Lingual Question Answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7016–7030, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Rogers, Anna, Matt Gardner, and Isabelle Augenstein. 2021. QA Dataset Explosion: A Taxonomy of NLP Resources for Question Answering and Reading Comprehension. *ArXiv*, abs/2107.12708.

Ruder, Sebastian and Avi Sil. 2021. Multi-Domain Multilingual Question Answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, pages 17–21, Punta Cana, Dominican Republic & Online. Association for Computational Linguistics.

Sanh, Victor, Lysandre Debyt, Julien Chaumond and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arxiv pre-print arXiv:1910.01108*.

Shakeri, Siamak, Cicero Nogueira dos Santos, Henry Zhu, Patrick Ng, Feng Nan, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2020. End-to-end synthetic data generation for domain adaptation of question answering systems. *ArXiv*, abs/2010.06028.

Tang, Duyu, Nan Duan, Tao Qin, and Ming Zhou. 2017. Question answering and question generation as dual tasks. *ArXiv*, abs/1706.02027.

Witteveen, Sam and Martin Andrews. 2019. Paraphrasing with Large Language Models. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 215–220, Hong Kong. Association for Computational Linguistics.

Xue, Linting, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua and Colin Raffel. 2020. mT5: A massively multilingual pre-trained text-to-text transformer. *ArXiv*, abs/2010.11934.