Exploring Input Representation Granularity for Generating Questions Satisfying Question-Answer Congruence

Madeeswaran Kannan, Haemanth Santhi Ponnusamy,
Kordula De Kuthy, Lukas Stein, Detmar Meurers
University of Tübingen
{mkannan,hsp,kdk,lstein,dm}@sfs.uni-tuebingen.de

Abstract

In question generation, the question produced has to be well-formed and meaningfully related to the answer serving as input. Neural generation methods have predominantly leveraged the distributional semantics of words as representations of meaning and generated questions one word at a time. In this paper, we explore the viability of form-based and more fine-grained encodings such as character or subword representations for question generation.

We start from the typical seq2seq architecture using word embeddings presented by De Kuthy et al. (2020), who generate questions from text so that the answer given in the input text matches not just in meaning but also in form, satisfying question-answer congruence. We show that models trained on character and subword representations substantially outperform the published results based on word embeddings, and they do so with fewer parameters.

Our approach eliminates two important problems of the word-based approach: the encoding of rare or out-of-vocabulary words and the incorrect replacement of words with semantically-related ones. The character-based model substantially improves on the published results, both in terms of BLEU scores and regarding the quality of the generated question. Going beyond the specific task, this result adds to the evidence weighing different form- and meaning-based representations for natural language processing tasks.

1 Introduction

Question generation (QG) is a challenging NLP task, where both language form and meaning play a vital role in the production of questions that have to be well-formed and meaningfully related to the envisaged answer. Neural models have been shown to be very promising for QG, with most recent approaches formulating the task as a sequence learning problem with the goal of mapping a sentence onto a question (e.g., Zhao et al., 2018; Chan and Fan, 2019; Xie et al., 2020). The research typically targets QG in the context of Question Answering, where the task is to generate a question that is related to the information in a given paragraph. The QA task ensures a general functional link between the question and the meaning of the passage that answers it. The datasets designed for such question answering/generation provide paragraph-level contexts for each question that span multiple sentences or even multiple passages. Note that the question here is related to the information expressed in the text passage, not to the way in which this information is structured and expressed in the text.

Consider the example from the SQuAD dataset shown in Figure 1. The first question pertains to the first sentence of the passage. While the concept *gravity* mentioned in that sentence is needed to answer the question, the question cannot be answered using the first sentence as such. For the second question, the information needed to answer the question is expressed in a sentence that is more in line with the question, but still falls short of the so-called question-answer congruence (Stechow, 1990; Sugawara, 2016) required for the sentence to serve as a direct answer to the question.

Context:	In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity. The main forms of precipitation include drizzle, rain, sleet, snow, graupel and hail.
Q_1:	What causes precipitation to fall? gravity
Q_2:	What is another main form of precipitation besides drizzle, rain, snow, sleet and hail? graupel

Figure 1: Example question-answer pairs from the SQuAD dataset (Rajpurkar et al., 2016)

Complementing QG in the prominent QA context, there are other strands of QG research that aim

Proceedings of the 14th International Conference on Natural Language Generation (INLG), pages 24–34,
Aberdeen, Scotland, UK, 20-24 September 2021. ©2021 Association for Computational Linguistics

at generating questions that can be answered by a sentence as given in the text, putting a premium on question-answer congruence. This includes QG work in the educational application domain, where the perspective of the question is supposed to reflect the perspective of the author of a given text passage that the student is supposed to learn about (Heilman and Smith, 2010; Heilman, 2011; Rus et al., 2012). Recent work under this perspective includes Stasaski et al. (2021), who propose a neural question generation architecture for the generation of cause-and-effect questions. They extract cause and effect relations from text, which are then used as answers for the neural question generation, aiming at direct question-answer congruence.

A second strand of work for which the relation between the question and the answer sentence as expressed in the text plays a crucial role is the research interested in discourse. An early example of research investigating the role of discourse structure for question generation is Agarwal et al. (2011). They identify discourse relations in a text as cues motivating the generation of a question and then formulate questions that can be answered by the sentences with those discourse relations, while ensuring direct question answer congruence. In a related vein, approaches making use of so-called Questions under Discussion (QuDs) to identify the information structure of a sentence in a given discourse also rely on such a direct relationship between question and answer. In a recent paper pursuing this perspective, De Kuthy et al. (2020) show that a seq2seq based neural approach can successfully generate meaningful, well-formed questions that can function as Questions under Discussions in a formal theory of discourse. Similarly, Pyatkin et al. (2020) showed that using question-answer pairs obtained through crowdsourcing can be used reliably to annotate discourse. Based on their crowdsourced data, they train a pipeline of neural models to directly generate such question-answer pairs from text. The overall goal of question generation supporting discourse analysis is to generate a question for every sentence in a text to explicitly characterize the evolving discourse.

Viewed from the perspective of question generation for tasks requiring question-answer congruence, the QG task in essence consists of two steps: (i) replace the answer phrase in the source sentence with a matching question word and (ii) transform the rest of the sentence into a well-formed question. All the words that the generated question consists of are already given, so only the question word that matches the answer phrase needs to be generated anew. The sentence-question pair in example (1) taken from De Kuthy et al. (2020) illustrates this.

(1) A: Auch Otto Graf Lambsdorf ist **gegen zweierlei Wahlrecht.**
also Otto Graf Lambsdorf is against double voting rights
Otto Graf Lambsdorf is also against double voting rights.

Q: **Wogegen** ist auch Otto Graf Lambsdorf?
what against is also Otto Graf Lambsdorf
What is Otto Graf Lambsdorf against, too?

Except for the answer phrase *gegen zweierlei Wahlrecht* ('against double voting rights'), all words from the source sentence reappear in the generated question, including the named entity *Otto Graf Lambsdorf*. The only new material in the question is the question word *wogegen* ('what against').

While the text thus includes all the language needed to successfully generate the question, for seq2seq-based approaches based on word embeddings, the challenge arises that words present in the source sentence which do not appear in the material the embeddings were trained on are not adequately represented. As admitted in De Kuthy et al. (2020), unknown and rare words are therefore a problem and cannot be correctly generated in the question. Rare words are often replaced by semantically related words that are inappropriate in the given context.

In this paper, we explore an alternative: characters and subwords as form-based and more fine-grained representations of both the input and output of the question generation task. We will show that this avoids the unknown/rare word problem and results in a substantial improvement both in a quantitative BLEU evaluation and in terms of a qualitative analysis of the questions. Going beyond the particular QG task, our results contribute to the general endeavour of exploring the best choices of form or meaning-based input and output representations for neural approaches for a range of NLP tasks depending on their characteristics.

2 Related Work

Traditional question generation approaches that leveraged syntactic structures and linguistic features (Liu et al., 2010; Curto et al., 2012; Heilman, 2011) to define transformation rules on parse trees are inherently limited in their scope and ability to deal with authentic language data. Deep learning has, in recent years, supplanted such methods given its ability to learn the syntactic and semantic properties and characteristics of language when provided with large amounts of natural language text.

Neural question generation is generally realised as a sequence learning problem, so a sequence-to-sequence (*seq2seq*) architecture (Sutskever et al., 2014) is a logical fit for this type of task. Here, the encoder network learns the latent representation of the source sentence and the decoder network generates the target question one word at a time. The work done by Du et al. (2017) introduces two such models, which are provided with the source sentence and paragraph-level information that encodes the context of the generated question. Borrowing from reinforcement learning, the work by Kumar et al. (2018) introduces policy gradients along with POS tags and named entity mentions to assign task-specific rewards to the training objective. Pointer-generator networks (Gu et al., 2016; See et al., 2017) with gated self-attention have been deployed to address the problem of rare and out-of-vocabulary words and larger contexts (Zhao et al., 2018).

The neural question generation models mentioned above, and many more in this vein, primarily focus on generating questions in English and consider words to be the atomic unit of meaning. They consequently approach the representation learning and text generation tasks at the word level. This assumption does not necessarily hold for languages such as Chinese, where the individual characters contain rich internal information. Neural language models that are trained on character-level inputs have been shown to capture more salient information about morphology than their word-level counterparts (Huang et al., 2016; Marra et al., 2018). Character-aware question answering systems (Golub and He, 2016; Lukovnikov et al., 2017) have similarly been shown to be resilient to the unknown word problem. To capture and combine information about language form and meaning, Bojanowski et al. (2017) proposed treating words as bags of character n-grams to enrich word embeddings with subword information. Byte-pair encoding (Shibata et al., 1999) has seen a recent resurgence in the context of generative language models where it is employed to perform subword segmentation without the necessity of tokenization or morphological analysis. Subword-level embeddings learned with the help of this method have been competitive in many downstream NLP tasks (Sennrich et al., 2015; Heinzerling and Strube, 2018; Xu et al., 2019).

To test performance and trade-offs between character-, subword-, and word-level representations in the context of question generation, we use the German question generation task proposed by De Kuthy et al. (2020), aimed at generating a Question under Discussion for each sentence in a discourse. The required question-answer congruence with the meaning and form requirements this entails, together with the relative morpho-syntactic richness and partially flexible word order of the German language make it an interesting experimental setting for exploring the potential advantages of character and subword representations.

3 Data

In terms of datasets for neural question generation models, contemporary approaches are generally trained on datasets created in the question answering context. These datasets, such as SQuAD (Rajpurkar et al., 2016), Quac (Choi et al., 2018), and Coqa (Reddy et al., 2019), are not well-suited for training models for tasks requiring high question-answer congruence, and they focus on English. Multilingual datasets like XQUAD (Artetxe et al., 2019), MLQA (Lewis et al., 2019), XNLI (Conneau et al., 2018), and TyDi QA (Clark et al., 2020) are similarly unsuitable as they contain only little data, intended as benchmark for the evaluation of question answering systems.

Given these limitations of the established English datasets for the research goals we are pursuing, we instead obtained the German QA answer corpus created by De Kuthy et al. (2020) and base our explorations on that dataset. The corpus contains 5.24 million sentence-question-answer triples which were generated by a transformation-based question generation system (Kolditz, 2015) on articles from the German newspaper *Die Tageszeitung (taz,* `http://taz.de`*)*. The corpus exhibits over 30 different types of questions, the most common of which are wh-questions asking for subject and object phrases (such as who or what questions in English) as well as various types of questions asking for adverbial modifiers (such as, for example, when or where questions). Some typical question-answer pairs will be discussed later in section 5.

4 Our Character and Subword-based Neural QG Approach

As the starting point and baseline of our approach, we take the same basic architecture as De Kuthy et al. (2020), a word-embedding based sequence-to-sequence model (Sutskever et al., 2014) with multiplicative attention (Luong et al., 2015). This was done in order to ensure comparability of our results with theirs. Furthermore, any fundamental changes to the neural architecture – such as using a Transformer (Vaswani et al., 2017) or a pointer-generator (Zhao et al., 2018) network – would make it more difficult to distinguish between any improvements offered exclusively by the change in input representation and those by the change in architecture.

To introduce character– and subword–level tokens, we defined an input pipeline consisting of the following steps: 1) UTF-8 text normalization was performed on the input sentence, 2) the normalized input sentence was parsed using *spaCy*'s (Honnibal et al., 2020) `de_core_news_sm` model to perform word-level tokenization and part-of-speech (POS) tagging, 3) a second tokenization pass was performed on each word token to generate character and subword tokens, and 4) each character and subword token pertaining to a given word token was assigned the latter's POS tag and the answer phrase indicator.

For character-level tokenization, each word was decomposed into a list of its component Unicode codepoints. Subword tokenization was performed with the *HuggingFace* Tokenizer library (Wolf et al., 2020). The library provides byte-pair encoding (BPE, Shibata et al., 1999) and unigram (Kudo, 2018) tokenization algorithms. BPE first constructs a baseline vocabulary with all unique symbols in a corpus. Then, merge rules that combine two symbols in the base vocabulary into a new symbol are learned iteratively until a desired final vocabulary size is reached. Conversely, unigram tokenization starts with a large initial vocabulary from which it repeatedly removes symbols that have the least effect on a loss function defined over the training data of a unigram language model. To reduce the size of the base vocabulary in both models, base symbols are directly derived from bytes rather than (all) Unicode codepoints. The library also includes the SentencePiece (Kudo and Richardson, 2018) algorithm, which processes the input as raw string sequences obviating the need for pre-tokenization.

Finally, bidirectional LSTM was used as the recurrent unit in the encoder as we expect the contextual information provided by the backward pass to not only enrich the sentential representation learned in the encoder but also lower the effective reduction in learnable parameters caused by the smaller vocabulary sizes of the character- and subword-level models. The per-timestep input to the encoder is the concatenation of the token embedding, POS embedding, and the answer phrase indicator. The final outputs of the encoder (hidden state, sequences, cell state) is the concatenation of the respective backward and forward layers of each output.

For the character-level models, a fixed-size vocabulary consisting of all the unique codepoints in the QA corpus was generated. Similarly, the subword tokenizers were trained on the entire corpus to generate vocabularies with 10K symbols each.[1]

5 Evaluation

For a comprehensive comparison, we trained five models: a word-level model to replicate De Kuthy et al. (2020), three subword models with different tokenization algorithms (byte-level BPE, SentencePiece BPE, and SentencePiece Unigram), and a character model. All models were trained on the same 400K training samples from the QA corpus for 20 epochs, and validation was performed on 40K samples. For each type of input representation, the model with the lowest validation loss was was evaluated on a held-out test set of 15K samples.

For their original model, De Kuthy et al. (2020) implemented a post-processing copy module to replace OOV marker tokens in the generated question with the original tokens from the source sentence; this behaviour was replicated for our word-level model. As model hyperparameters, we used: batch size: 128, encoder: Bi-LSTM, decoder: LSTM, encoder/decoder hidden size: 256/512, encoder/decoder dropout: 0.5, word/subword/character embedding dim: 300, decoder beam search width: 5. Table 1 shows the BLEU scores from comparing the ground-truth questions of the test set with corresponding model-generated questions. We used the standard *SacreBLEU* library (Post, 2018)[2] for the calculation of the BLEU scores.

[1] The subword vocabularies also include the base symbols found in the character vocabulary. In both cases, special meta tokens such as unknown, sentence-start and end markers were additionally added to each vocabulary.

[2] Version 1.4.10 with default parameters.

Model	BLEU 1/2/3/4	Cumulative
Word (De Kuthy et al., 2020)	93.8/86.5/81.0/76.5	84.24
Word (replication)	93.8/86.5/81.0/76.5	84.20
Subword (Byte BPE)	98.2/93.4/90.0/87.4	**91.97**
Subword (SentPiece BPE)	97.0/91.4/87.3/84.1	89.35
Subword (SentPiece Unigram)	98.1/93.3/89.8/87.2	91.76
Character	97.2/91.8/88.0/85.1	90.18
Subword-level (Byte BPE NoPOS)	98.0/93.0/89.4/86.7	91.48
Subword (SentPiece BPE NoPOS)	97.8/92.3/88.5/85.7	90.67
Subword (SentPiece Unigram NoPOS)	98.0/92.7/88.9/86.1	90.84
Character (NoPOS)	97.4/91.8/87.9/84.9	90.34

Table 1: Quantitative evaluation results

The word-level QG model with our modifications is able to produce results essentially identical to those of the baseline model by De Kuthy et al. (2020). Both models use the post-processing copy step to address the problem of out-of-vocabulary tokens, but neither is able to fully overcome it due to the intrinsic weaknesses of such extra-modular, non-neural solutions. The character- and subword-level models, on the other hand, entirely sidestep this issue by generating the target sequence one character or subword at a time. We additionally trained variants of the character- and subword-level models without POS tags (the NoPOS models in the table). Even with fewer learnable parameters and without the linguistic information provided by the POS tags, the models are able to achieve scores very close to those of their POS-aware counterparts. The effect of different subword tokenization algorithms on the quantitative performance of the model appears to be minimal.

5.1 Error Analysis

To analyze the quality of the results produced by our models and compare them to those of the baseline word-level model, we performed a manual evaluation of the questions generated for the same sample of 500 sentences of De Kuthy et al. (2020).

The quality of the generated questions was manually evaluated by two human annotators, both trained linguists and native speakers of German. They were asked to provide a binary judgment: whether the question is well-formed and satisfies question-answer congruence with the source sentence. The two conjoined criteria were expressed in the annotation manual as (i) Well-Formedness: Is the question grammatically correct and would I formulate it that way as a native speaker of German? and (ii) Question-Answer Congruence: Is the question answered by the associated sentence as a whole? The annotators were instructed to take into account all aspects of grammaticality, including word order, verb forms, punctuation, and also spelling and capitalization errors. For the evaluation of question-answer congruence, the annotators checked whether the generated question was answerable by the full source sentence, in particular whether the question word matched the given answer phrase and whether the question did not contain any semantically different words. The resulting annotation showed good inter-annotator agreement ($\kappa = 0.74$).

The results of this evaluation (Table 2) reveal how model performance increases with more fine-grained in input granularity. The baseline word-level model posted the worst score among all trained models, generating well-formed questions for only 54.2% of the 500 sentences in the evaluation set. The best subword model improves upon this substantially with 61.0% well-formed questions, and the character model adds a further, small improvement. Curiously, removing POS tags as input features from the subword model results in a slight performance increase, but the opposite for the character model. The effect is even more pronounced in the SentencePiece BPE subword model. To investigate this further, we performed systematic error analysis of the most frequently encountered errors (Table 3). Note that the overall sums differ slightly from the percentages in Table 2 since one question can contain multiple types of errors.

Model	Well-formed Questions
Word	54.2%
Subword (SentPiece Unigram)	59.6%
Subword (SentPiece Unigram No POS)	61.0%
Character	**61.4%**
Character (No POS)	59.6%

Table 2: Results per question for the evaluation set

Despite the post-processing copy mechanism, the questions from the word model still contained

Error Type	Word	Subword	Subword	Character	Character
		(SentPiece Unigram)	(SentPiece Unigram NoPOS)		(NoPOS)
Question word	82	108	100	109	117
Unknown Word	35	0	0	0	0
Word Order	29	20	23	21	23
Different Word	35	16	5	1	0
Different Subword	0	1	2	0	0
Missing Word	2	8	10	7	4
Missing Subword	0	0	2	0	0
Repeated Word	4	4	4	10	5
Verb Form	8	9	15	13	17
Source Sentence	13	13	13	13	13
Answer Phrase	23	31	31	24	26
Spelling	0	3	2	0	4
Total	231	217	205	197	213

Table 3: Distribution of error types in the evaluation samples

unknown words in 35 cases. For example, rare words such as *süffisant* (*smug*), *listenreich* (*cunning*), *Naschwerk* (*sweet delicacy*), *Erbtanten* (*rich aunt from which one inherits*). The subword and character models did not have this problem at all. Unwanted word replacements with different words occurred in 35 samples with the word model, for example, *unbegreiflich* (*incomprehensible*) was replaced by *geschehen* (*happen*), *Adelheid Streidel* (proper name of a terrorist) by *extremistischen Streidel* (*extremist Streidel*), and *bewilligt* (*approved*) by *beantragt* (*requested*). The subword models reduce this to as few as five occurrences, and in the character models this type of error does not occur at all. By far, the biggest error source for all models is the production of incorrect question words. This is a hard objective since the question word depends on aspects of form (e.g., does it refer to a nominal phrase or a prepositional phrase) and meaning (e.g., does it refer to an animate or inanimate referent) of the given answer phrase. The word-level model had fewer problems with question word generation than the other models, so the word embeddings encode sufficient form and meaning information for the model to learn the question word patterns.

There appears to be no single, clear pattern across all models that explains the effect of POS tags. Nevertheless, the quality of question words does consistently suffer when they are removed from the input. The character-based model without POS tags generated the highest number of questions with an incorrect question word - an aspect of question generation that relies on meaning-related information, evidently provided by the latter. One potential explanation could be rooted in how the models process the POS features: By assigning to each subword or character the POS tag of its parent word, the model has to contend with increased noise in the training data due to weak correlation between the tags and specific subwords or characters. For instance, the subword unit *her* in *herkommen (to come from)* would take the latter's POS tag `VB (verb)` but will be assigned `JJ (adjective)` when appearing in *herrlich (superb)*.

To gain a better understanding of when a model generates a new form and when it copies tokens from the input, in the following we discuss indicative examples together with the softmax-activated attention scores between the source sentence and the question. In the figures below, the x-axis and y-axis of each plot correspond to the tokens in the generated question and the source sentence, respectively. Each pixel corresponds to the alignment weight w_{xy} of the y-th source token and x-th target token, ranging from 0 (purple) to 1 (yellow). The red tokens on the y-axis indicate the phrase in the source sentence that answers the question.

Example (2) shows a typical sentence-question pair from the evaluation sample. Both the subword models and the character models produced the correct question in (2-b), given the answer phrase (marked in bold font).

(2) a. Bis dahin seien **die Länder der DDR**
until then would be the states of the GDR
pleite.
bankrupt

b. Wer ist bis dahin pleite?
who is until then bankrupt

For the correct question (2-b), the models have to produce the question word *Wer (who)* in place of the answer phrase *die Länder der DDR (the states of the GDR)*, and they have to transform the plural *seien (were)* into the singular verb *ist (is)*. The sentence initial *Bis dahin (until then)* must be placed after the verb and lower cased. The attention plots in Figures 2 and 3 directly showcase this. The tokens in the answer phrase, particularly the first one, have higher alignment weights for the question word than the other tokens in the sentence. Similarly, the model specifically attends to the verb in the source sentence when generating the same in the question. The tokens that are copied as-is from the source sentence have strong, monotonic weights that appear as diagonals.

Example (3) shows another sentence-question pair from the evaluation set. The character model predicted the correct question (3-b), but the subword model predicted the incorrect question (3-c), in which the adverb *danach (thereafter)* is repeated and the numeral *1988* from the input is missing.

(3) a. Danach sollte Ende 1988 **mit der Produktion**
thereafter should end 1988 with the production
der U-Boote und mit der Teillieferung
of the submarines and with the partial deliveries
begonnen werden.
started be

Subsequently, production of the submarines and partial deliveries were to begin at the end of 1988.

b. Womit sollte danach Ende 1988 begonnen
with what should thereafter end 1988 started
werden ?
be

What should be started thereafter at the end of 1988?

c. Womit sollte danach Ende danach
with what should thereafter end thereafter
begonnen werden ?
started be

The corresponding attention scores are shown in Figure 4 for the correct question (3-b) generated by the character model and Figure 5 for the erroneous question (3-c) produced by the subword model. Once again, in order to produce the question word *Womit (with what)*, both models assign a strong weight to the preposition *mit (with)*, which is the first token of the given answer phrase. While the character model then continues to correctly annotate the tokens in the source sentence, the subword model's alignments show more ambiguity. For the token *danach (thereafter)*, it additionally attends to *Ende (end)* in the source sentence - another word that carries a temporal meaning. At the position of the numeral *1988*, the model assigns significant weights to all three temporally related words, but the weight for *Ende* is diminished due to its occurrence in the previous timestep. Neverthe-

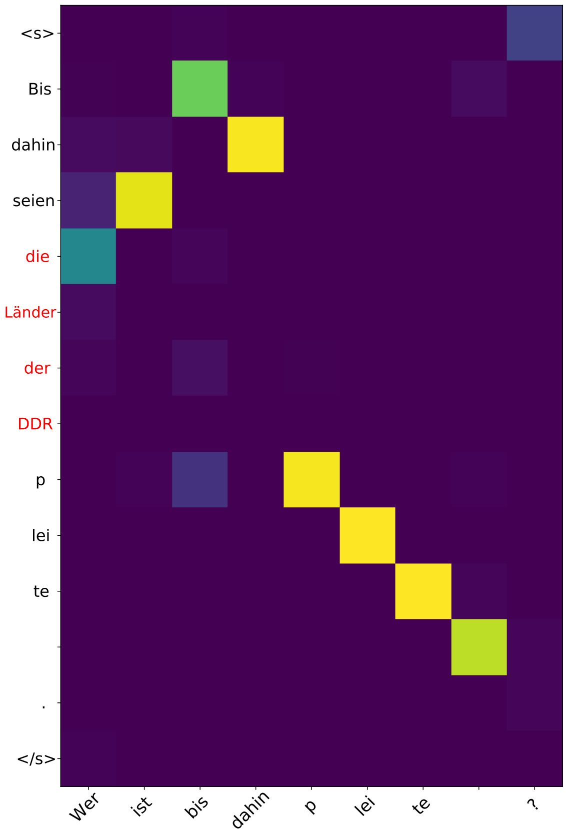

Figure 2: Subword attention plot for example (2)

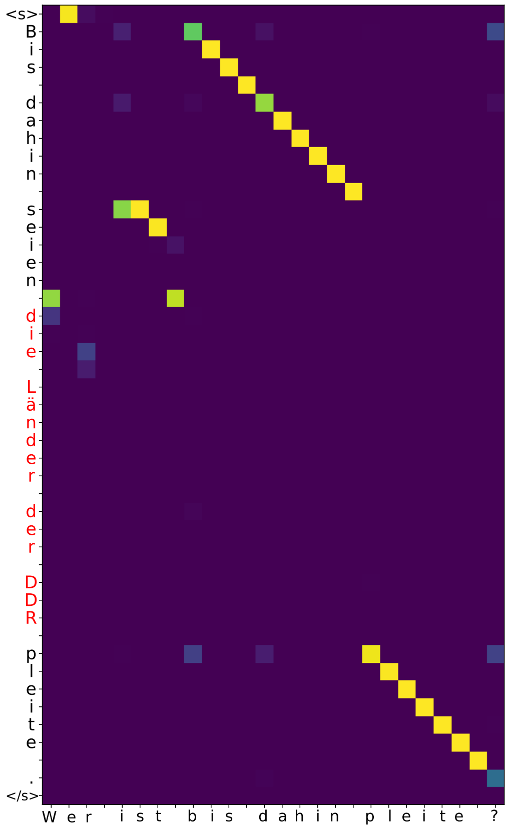

Figure 3: Character attention plot for example (2)

less, the model ultimately shows higher confidence in *danach* than in the rest and thus (mis-)predicts it a second time.

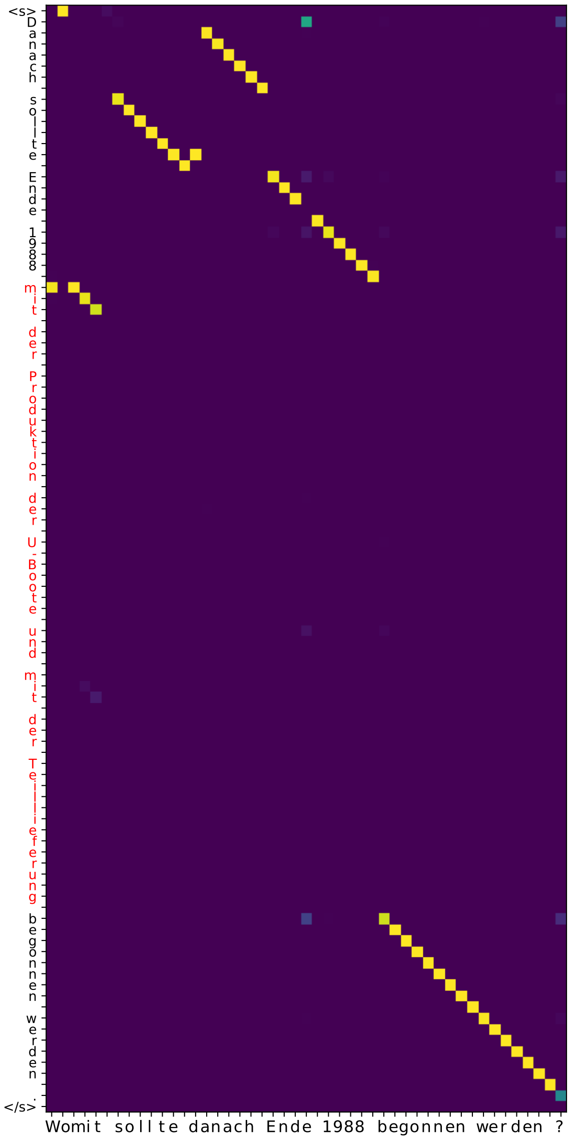

Figure 4: Character attention plot for example (3)

One potential problem of a purely form-based approach using characters is that it can produce character strings that do not correspond to any word in the given language. This hardly ever occurred in the questions generated by our character model with the exception of one interesting example where the model created a new question word, illustrated here in example (4).

(4) a. Dies dürfte sich mit der Schaffung des
this is likely itself with the creation of the
Binnenmarktes **ab** **1993** ändern.
single market from 1993 change

This is likely to change with the creation of the single market from 1993.

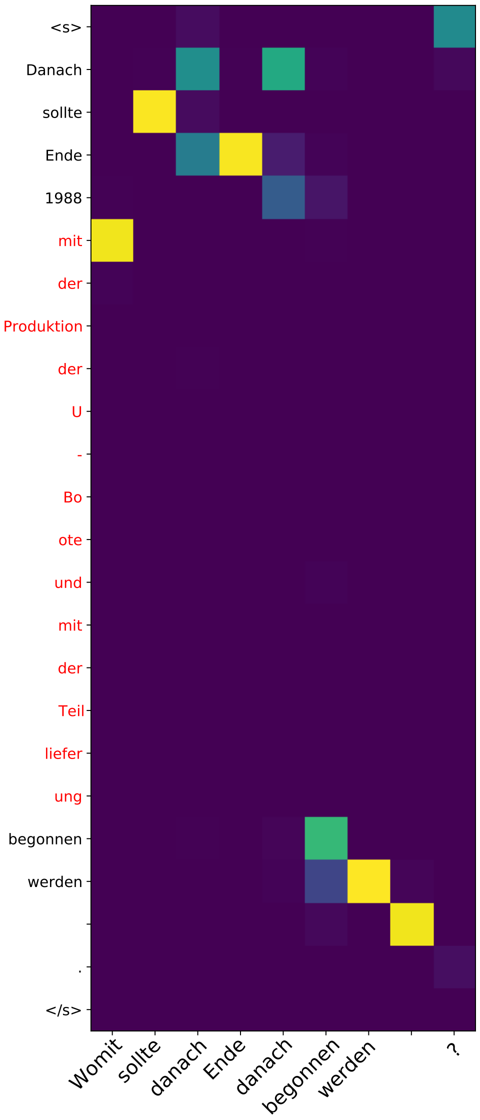

Figure 5: Subword attention plot for example (3)

b. Worab durfte sich dies mit der
where from was likely itself this with the
Schaffung des Binnenmarktes ändern?
creation of the single market change

From when was this likely to change with the creation of the single market?

Given the answer phrase *ab 1993*, the model produced the question word *Worab* – a concatenation of the (existing) words *wo (where)* and the preposition *ab (on)* – instead of the required question phrase *ab wann (from when)*. Such concatenations of a question word and a preposition actually exist in German, e.g., in the question word *woran (what of)*, so the character model apparently picked up this pattern of generating question words from prepositions, but applied it to a non-existing case.

6 Conclusion

We explored the prospect of neural question generation at the character- and subword-level using finer-grained input representations than word tokens by adopting De Kuthy et al. (2020)'s task of generating Questions under Discussion for German. The models that were trained on character and subword tokens showed significant leaps in BLEU scores in comparison to the baseline word-level model, even in the absence of extra linguistic information.

In addition to eliminating the problem of out-of-vocabulary and rare words, our manual analysis of the generated questions revealed that those models were able to learn and exploit both semantic and orthographic information with fewer parameters, producing questions with fewer errors relating to word order and word replacement. The character model, in particular, is able to fully eliminate the latter error category.

Considering the relevance of the research beyond the specific question generation task, the results reported in this paper provide further evidence and motivation to consider the advantages of form-focused neural representations and character-level natural language generation for tasks such as machine translation and extractive text summarization.

References

Manish Agarwal, Rakshit Shah, and Prashanth Mannem. 2011. Automatic question generation using discourse cues. In *Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–9, Portland, OR. Association for Computational Linguistics.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2019. On the cross-lingual transferability of monolingual representations. *CoRR*, abs/1910.11856.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Ying-Hong Chan 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.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. *arXiv preprint arXiv:1808.07036*.

Jonathan H Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages. *arXiv preprint arXiv:2003.05002*.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Sérgio Curto, Ana Cristina Mendes, and Luísa Coheur. 2012. Question generation based on lexico-syntactic patterns learned from the web. *Dialogue & Discourse*, 3(2):147–175.

Kordula De Kuthy, Madeeswaran Kannan, Haemanth Santhi Ponnusamy, and Detmar Meurers. 2020. Towards automatically generating questions under discussion to link information and discourse structure. In *Proceedings of the 28th International Conference on Computational Linguistics*, Barcelona, Spain.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. *arXiv preprint arXiv:1705.00106*.

David Golub and Xiaodong He. 2016. Character-level question answering with attention. *arXiv preprint arXiv:1604.00727*.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*.

Michael Heilman. 2011. *Automatic factual question generation from text*. Ph.D. thesis, Carnegie Mellon University.

Michael Heilman and Noah A. Smith. 2010. Extracting simplified statements for factual question generation. In *In Proceedings of the Third Workshop on Question Generation*.

Benjamin Heinzerling and Michael Strube. 2018. BPEmb: Tokenization-free Pre-trained Subword Embeddings in 275 Languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Jiangping Huang, Donghong Ji, Shuxin Yao, Wenzhi Huang, and Bo Chen. 2016. Learning phrase representations based on word and character embeddings. In *Neural Information Processing*, pages 547–554, Cham. Springer International Publishing.

Tobias Kolditz. 2015. Generating questions for German text. Master thesis in computational linguistics, Department of Linguistics, University of Tübingen.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. *arXiv preprint arXiv:1804.10959*.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Vishwajeet Kumar, Ganesh Ramakrishnan, and Yuan-Fang Li. 2018. A framework for automatic question generation from text using deep reinforcement learning. *arXiv preprint arXiv:1808.04961*.

Patrick Lewis, Barlas Oğuz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2019. Mlqa: Evaluating cross-lingual extractive question answering. *arXiv preprint arXiv:1910.07475*.

Ming Liu, Rafael A Calvo, and Vasile Rus. 2010. Automatic question generation for literature review writing support. In *International Conference on Intelligent Tutoring Systems*, pages 45–54. Springer.

Denis Lukovnikov, Asja Fischer, Jens Lehmann, and Sören Auer. 2017. Neural network-based question answering over knowledge graphs on word and character level. In *Proceedings of the 26th international conference on World Wide Web*, pages 1211–1220.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Giuseppe Marra, Andrea Zugarini, Stefano Melacci, and Marco Maggini. 2018. An unsupervised character-aware neural approach to word and context representation learning. *Lecture Notes in Computer Science*, page 126–136.

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

Valentina Pyatkin, Ayal Klein, Reut Tsarfaty, and Ido Dagan. 2020. QADiscourse - Discourse Relations as QA Pairs: Representation, Crowdsourcing and Baselines. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2804–2819, Online. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Vasile Rus, Brendan Wyse, Paul Piwek, Mihai Lintean, Svetlana Stoyanchev, and Cristian Moldovan. 2012. A detailed account of the first question generation shared task evaluation challenge. *Dialogue & Discourse*, 3(2):177–204.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Yusuxke Shibata, Takuya Kida, Shuichi Fukamachi, Masayuki Takeda, Ayumi Shinohara, Takeshi Shinohara, and Setsuo Arikawa. 1999. Byte pair encoding: A text compression scheme that accelerates pattern matching. Technical report, Technical Report DOI-TR-161, Department of Informatics, Kyushu University.

Katherine Stasaski, Manav Rathod, Tony Tu, Yunfang Xiao, and Marti A. Hearst. 2021. Automatically generating cause-and-effect questions from passages. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 158–170, Online. Association for Computational Linguistics.

Arnim von Stechow. 1990. Focusing and backgrounding operators. In Werner Abraham, editor, *Discourse Particles*, pages 37–84. John Benjamins, Amsterdam.

Ayaka Sugawara. 2016. *The role of question-answer congruence (QAC) in child language and adult sentence processing*. Ph.D. thesis, Massachusetts Institute of Technology.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on*

Empirical Methods in Natural Language Processing: System Demonstrations, pages 38–45, Online. Association for Computational Linguistics.

Yuxi Xie, Liangming Pan, Dongzhe Wang, Min-Yen Kan, and Yansong Feng. 2020. Exploring question-specific rewards for generating deep questions.

BinChen Xu, Lu Ma, Liang Zhang, HaoHai Li, Qi Kang, and MengChu Zhou. 2019. An adaptive wordpiece language model for learning chinese word embeddings. In *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, pages 812–817. IEEE.

Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. 2018. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3901–3910.