GQS: Generating Questions with Slang

Maya Angelova
Question Processing
University of Potsdam, Summer 2020,
mangelova@uni-potsdam.de

Abstract

The main question posed by this work is to find out what does a deep neural network model learn from slang and how it generates questions based on noisy every day data such as tweets. Further the main motivation is curiosity for how Transformer models which showed superior performance on many tasks behave in noisy data settings as in internet slang. The results are not satisfactory: there is a need for a dedicated slang dataset for the specific task with a predefined format. Human evaluation is to be performed, since automatical evaluation is not a reliable source. The results suggest that slang can be learned, even from scratch with existing pre-trained available models.

1 Introduction

Question generation is the task of automatically generating questions given a certain context: an image, a knowledge graph, a text paragraph, tables, etc. These generated question should have their answers embedded in the given context. The task can serve to train a chat dialog system, used for educational purposes or simply enriching and creating other Question-Answering training datasets and thus improving Question Answering models.

In this paper the task of generating questions given natural language tweets and the corresponding answers is being analyzed and discussed. The argumentation behind using dataset of tweets is that every day tweet messages more strongly resemble usual every day communication, embedding slang terms and noisy linguistic data.

The paper is organized as follows: a general introduction to the topic as given in this section, followed by background and definitions of slang. An overview over previous relevant encountered work is given in Section 3. Section 4 takes a closer look at available datasets, where Section 5 describes the chosen approach. Section 6 presents the evaluation of the results, whereas Section 7 discusses some of the possible problems for the poor performance and Section 8 points to the Github repository containing all the scripts and data.

2 Slang

Slang as defined by (Kulkarni and Wang, 2018) is extra-grammatical word formation phenomena. Or as defined by Mirriam-Webster dictionary [1]:

> "an informal nonstandard vocabulary composed typically of coinages, arbitrarily changed words, and extravagant, forced, or facetious figures of speech"

as well as "language peculiar to a particular group". And to exhaust other options, the definition given by Cambridge Dictionary [2]:

> "Slang is vocabulary that is used between people who belong to the same social group and who know each other well. Slang is very informal language. It can offend people if it is used about other people or outside a group of people who know each other well. We usually use slang in speaking rather than writing. Slang normally refers to particular words and meanings but can include longer expressions and idioms."

Example of such are *alphabetisms*, *blends* or *clippings* or *reduplicatives*. *Alphabetisms* are shortenings of a multiword sequence, an abbreviation which is read letter by letter. An example would be *UN* which stands for *United Nations*. *Blends* or portmanteaus, are formed by merging parts of existing words. For example,

[1] https://www.merriam-webster.com/
[2] https://dictionary.cambridge.org

edutainment is a blend of *education* and *entertainment*. *Clippings* are constructed by shortening lexemes. For example, *berg* is a clipping of *iceberg*, *gym* is a clipping of *gymnasium*. *Reduplicatives* are word pairs constructed by either repeating a word (*boo boo*) or by alternating certain vowels or consonants so that they are phonologically similar *clickety-clackety*, *teenie-weenie*, *itsy-bitsy*.

Slang is difficult to process and model because of irregularities and noisy linguistic data. But such noisy data is the norm not the exception in human communication and natural language systems utilized in personal assistants such as Siri or Alexa are forced to handle noisy data on a daily basis.

3 Previous work

3.1 Urban Dictionary Embeddings

A study from the year 2020 – *Urban Dictionary Embeddings for Slang NLP Applications* (Steven Wilson, 2020) from the University of Edinburgh and the Alan Turing Institute presents learned word embeddings using the Urban Dictionary Corpora [3]: a crowd-sourced dictionary for slang words and phrases. The authors report that sentiment analysis and sarcasm detection tasks with classifiers initialized with the these embeddings resulted in improved performance compared to initializing with a range of other well-known, pre-trained embeddings. The context and background provided by the Urban Dictionary embeddings contributed for this performance improvement.

3.2 SlangNet

Another paper of interest is *SlangNet: A WordNet like Resource for English Slang* (Dhuliawala et al., 2016) where the authors presented a WordNet like structured resource for slang words and internet neologisms. This corpus was nowhere to be found or downloaded unfortunately.

3.3 Simple Models for Word Formation

Simple Models for Word Formation in English Slang (Kulkarni and Wang, 2018) from the year 2018 use simple generative models for three types of extra-grammatical word formation phenomena abounding in English slang: blends, clippings, and reduplicatives. Their gold standard datasets are human annotated. Their approach is described as a data-driven one incorporating linguistic knowledge, achieving state of the art performance.

[3]https://www.urbandictionary.com/

3.4 Preliminary results and existing approaches

To my knowledge at this moment there are no known previous study or baselines that generate questions with the main focus on slang. There are several available papers on Question Generation which also provide code.

3.4.1 ERNIE-GEN

The paper *ERNIE-GEN: An Enhanced Multi-Flow Pre-training and Fine-tuning Framework for Natural Language Generation*, (Xiao et al., 2020) from the year 2020 presents a multi-flow language generation framework which can be utilized both for pre-training and fine-tuning. The novelty of the approach is hidden in three elements.

- Span-by-span generation pre-training task implemented using a multi-flow attention architecture, based on the fact that language phrases are organized in a coherent manner, and therefore the aim to generate a complete span by span sequence rather than word by word.
- Infilling generation mechanism and a noise-aware generation method addressing the exposure bias. [4]
- Multi-granularity target fragments sampling which enforces the decoder part to rely more on the encoder representations and not primarily on the previously generated ones in the pre-training stage.

The authors published three pre-trained models:

1. ERNIE-GEN base (lower-cased — 12-layer, 768-hidden, 12-heads, 110M parameters) pre-trained on English Wikipedia and BookCorpus (totally 16GB)
2. ERNIE-GEN large (lower-cased — 24-layer, 1024-hidden, 16-heads, 340M parameters) pre-trained on English Wikipedia and BookCorpus (totally 16GB)
3. ERNIE-GEN large with 430G (lower-cased — 24-layer, 1024-hidden, 16-heads, 340M parameters) pre-trained on a 430GB corpus. The text corpora is extracted from the

[4]The exposure bias is caused by teacher forcing and produces a training-generation discrepancy

corpus used by RoBERTa (Liu et al., 2019), T5 (Raffel et al., 2019) and ALBERT (Lan et al., 2020).

For the Question Generation downstream task the authors report on the SQuAD 1.1 dataset (Rajpurkar et al., 2018) using ERNIE-GEN large with 430G BLEU-4 of 25.41, METEOR of 26.77 and Rouge-L of 52.91.

Unfortunately I was not able to get any of the ERNIE-GEN models to run and replicate their results or to fine-tune on the TweetQA dataset due to several exceptions such as not being able to find cuda devices, due partially to different available python ERNIE-GEN modules. For this more time and ultimately spare thousands of dollars for computer power are needed.

3.4.2 Question Generation by Transformers

The study *Question Generation by Transformers* from 2019 (Kriangchaivech and Wangperawong, 2019) presents a model which automatically generates questions from Wikipedia passages using the Transformer architecture. The framework is pretrained on the inverted SQuAD 1.1 dataset (Rajpurkar et al., 2018). The outcome are simple generated questions about unseen passages with answers containing an average of 8 words per question.

The *bert_qa* python module via *pip* can be installed, unfortunately does not bring the desired outcome on such a short notice. On a local virtual machine based on Debian 10, a MemoryError occurres. Running the available BERT QA jupyter notebook on google colab had different errors, the last one of which was *ImportError: cannot import name 'network'*.

Locally compile the code and installing the module on google colab was successful, unfortunately still, the training takes too long, as well as the inference (without fine-tuning), so this was unfeasible for the given time and cost frame. I suppose that the problems could be with the underlying hardware, as Bert QA was not able to run from scratch on distributed GPUs, single GPU was too slow without a any available log feedback output of the current process. Using TPU should be manually adjusted in the code to meet the specific google colab requirements. Errors such as *AttributeError: module 'tensorflow_core._api.v2.data.experimental' has no attribute 'AutoShardPolicy'* occurred.

4 Dataset

4.1 Urban Dictionary

The crowdsourced online Urban Dictionary contains slang words and phrases, as defined by the community [5]. The following slang formation are encountered in the Urban Dictionary: alphabetisms, such as *bc*, short for *because*, blends such as *edutainment*, clippings such as *gym* and reduplicatives such as *itsy-bitsy*.

4.2 Stanford Question Answering Dataset

Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2016) consists of questions posed by crowd workers on a set of Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading passage. The data set comprises 100,000+ questions.

4.3 TweetQA

TWEETQA: A Social Media Focused Question Answering Dataset (Xiong et al., 2019) from the year 2019 addresses the issue of the predominantly formal text based linguistic corpora used for training Question Answering systems. The paper presents a large-scale corpus based on social media, and in particular Twitter. The tweets that were gathered were only written by journalists who used them to write an article and were manually annotated by creating questions answers pairs.

An example tweet from the dataset can be seen in Table 1 and corpora statistics in Tables 2 and 3.

Passage: Oh man just read about Paul Walkers death. So young. Ugggh makes me sick especially when it's caused by an accident. God bless his soul. – Jay Sean (@jaysean) December 1, 2013
Q: why is sean torn over the actor's death? **A**: walker was young

Table 1: An example from the TweetQA dataset. The specifics are outlined as the tweet being informal and the occurrence of twitter handles as usernames.

Note Some of the answers to the questions contain mistakes such as one listed in Table 4.

[5] https://www.urbandictionary.com/

# of Training triples	10692
# of Development triples	1086
# of Test triples	1979
Average question length (#words)	6.95
Average answer length (#words)	2.45

Table 2: Basic statistics of TweetQA.

Question Type	Percentage
What	42.33%
Who	29.36%
How	7.79%
Where	7.00%
Why	2.61%
Which	2.43%
When	2.16%
Others	6.32%

Table 3: Question Type statistics of TweetQA

Passage: Just LISTEN to the words! Ain't nothing but Citizens appealing to their future VPOTUS! WITH RESPECT an CLARITY!" #AMENandAMEN Christopher Jackson (@ChrisisSingin) November 19, 2016
Q: what are the citizens appealing to? A_1: to theri future A_2: their future vpotus

Table 4: Example of an erroneous tweet

Passage: Iowa cafe creates Trump burger for the guy who likes to ham it up.
Q: .@FoxNewsLeisure (@fxnleisure) January 27, 2016 A_1: iowa A_2: iowa

Table 5: Example of an erroneous tweet

Others contain no questions at all but statements instead as one listed in Table 5.

4.4 MetroLyrics

The original kaggle dataset from MetroLyrics is contains 380.000+ lyrics from different artists and genres, organized by artist, year, song. Unfortunately the data can no longer be found online and downloaded. A Github repository from a Computer Science project at the University of Berkley, California [6] contains sample datasets.

4.5 Other datasets

Datasets that can be taken into considerations include WebQuestions dataset (Jonathan Berant and Liang, 2013) and SimpleQuestions dataset (Antoine Bordes and Weston, 2015). The first is as the authors point out specifically for benchmarking QA models, based on structural knowledge bases. The later one is prepared by human annotators based on questions and facts extracted from the Knowledge Base Freebase. Such datasets are not really applicable for the topic of interest, since they rarely if at all contain slang words or words that can be categorized as such.

5 Approach

The current work is based on the Github repository[7], an open source project investigating Question Generation with pre-trained transformers (seq-2-seq models). The repository contains simple data processing and training scripts and easy to use pipelines for inference. There are three pipelines:

- question-generation: for single task Question Generation models.
- multitask-qa-qg: for multi-task Question Answering, Question Generation models.
- e2e-qg: for end-to-end Question Generation.

The repository is based on the popular repository[8] containing state-of-the-art general-purpose architectures (BERT, GPT-2, RoBERTa, XLM, DistilBert, XLNet, T5) for Natural Language Understanding and Natural Language Generation, but to this day lacking a pipeline for Question Generation. This is incorporated by Patil Suraj in the above Github repository. The default setting of Question Generation pipeline is to use the *valhalla/t5-small-qg-hl* model with a highlight qg format. It is an answer aware question generation method, where the model knows the text and an answer and has to generate a corresponding question.

5.1 Highlight Format

This format takes a text data and highlights the corresponding answer, as seen in Table 6.

[6] https://github.com/hiteshyalamanchili/SongGenreClassification

[7] https://github.com/patil-suraj/question_generation

[8] https://github.com/huggingface/transformers

<hl> 42 <hl> is the answer to life, the universe and everything.

Table 6: The answer *42* is highlighted within the text with special highlight tokens.

5.2 Data Pre-processing

In order to be able to distinguish slang from non-slang words, the Urban Dictionary was utilized. A file in the comma-separated-value format was downloaded[9] and made available in form of one list containing the lower case urban words. Every tweet was than split up into words and every word was checked against the Urban Dictionary.

5.2.1 Steps

1. Annotation of the original TweetQA data with available slang words in the tweets as seen in the Urban Dictionary via a python script
2. Preparation

   convert the TweetQA data to a SQUAD format so it can be easily deployed with the already available pipeline scripts .

   the idea behind this step is to be able to use the highlight format of an answer aware question generation model, based on the paper *A Recurrent BERT-based Model for Question Generation* (Chan and Fan, 2019) as explained further above.

5.2.2 Problems

Passage: The #endangeredriver would be a sexy bastard in this channel if it had water. Quick turns. Narrow. (I'm losing it) John D. Sutter (@jdsutter) June 21, 2014
Q: what is this user "losing"? A_1: it A_2: I'm losing it

Table 7: Problematic tweet Nr. 1

There are several problems arising from the previous described data pre-processing. First and fore most the SQUAD data format expects the starting point of the answer within the given context. This is a problem for training, since the answers in the TweetQA dataset are not perfectly aligned to the tweets and are as well as the questions somewhat free composed and abstract.

An example would be the tweet as seen in Table 7, where there are two answers to the question *what is this user "losing"?*: namely *it* and *he is losing it*. For the first one, the answer starts at the position 85 in the tweet, whereas for the second one there is no such text in the tweet and therefore no relevant position can be found.

Passage: And I don't think those things are in conflict, bc a progressive populist candidate focused on the economy and AHCA can do both— Dan Pfeiffer (@danpfeiffer) June 21, 2017
Q: what can a progressive populist do both of? A_1: pay attention to the economy and ahca A_2: focus on the economy

Table 8: Problematic tweet Nr. 2

Another example of a similar tweet is given in Table 8, with the first answer being *pay attention to the economy and ahca* with the position of 95, marking the start of the phrase *economy and ahca* and the second one: *focus on the economy* with the position of 85, marking the beginning of the word *focus* in the given tweet. Luckily there is the variable *is_impossible* in the SQUAD JSON format standard which can be utilized in such ambiguous situations. This attribute is set to true when any the available answers cannot be assigned a position in the given tweet.

Another problem with the automatic preprocessing is the simple yet powerful approach of looking for the first occurrence of the current relevant answer in the tweet. That has two problems: first there are parts of speech such as definite and indefinite article (*the, a, an*) which are very common in every day written and spoken language. The answer may also occur in different parts of the tweet with different meanings, thus not reflecting the answer to the question and can introduce ambiguity. An example for the previously described problem would be the tweet as seen in Table 9, where the first answer-phrase: *a dog.* cannot be found in the tweet, but only the second one: *the dog.*. Removing these articles and searching only for the word *dog* in the tweet for the starting posi-

[9] https://www.kaggle.com/therohk/urban-dictionary-words-dataset/datachro

Passage: .@CNN What do I do about this dog? Eat the pizza, eat the dog, or both? Adam Davis (@amdhit) July 23, 2014
Q: who does davis propose eating? A_1: a dog. A_2: the dog

Table 9: Problematic tweet Nr. 3

tion, makes it easier to find of the nearly correct position of the main answer. The second problem is illustrated by the two occurrences of the answer-word *dog*, where the second occurrence would be the correct one and not the first.

Passage: .@CNN What do I do about this dog? Eat the pizza, eat the dog, or both? Adam Davis (@amdhit) July 23, 2014
Q: who is davis directing their question to? A_1: cnn. A_2: cnn.

Table 10: Problematic tweet Nr. 4

Problematic is also the punctuation in a given answer. An example would be the tweet as seen in Table 10 with two given identical answers *cnn.*. Therefore the script removes the most frequent punctuation symbols from the answer before looking the word in the Urban Dictionary to classify it as slang.

Another problem with the Urban Dictionary is that there are too many words considered urban or (as defined by this work) slang, but which cannot be straightforward categorized as such. Such words would be: *basketball* with the given definition in the Urban Dictionary *"a sport"*, which is not considered a slang word or is at least ambiguous to do so.

5.2.3 Manually editing TweetQA

Because of the above stated problems there is a need to go over the automatically processed Tweet QA data and manually adjust for errors: correct the starting position and remove unwanted words from the slang list. For a case such as in Table 11, the slang as found in the Urban Dictionary is basically the whole tweet. It will be changed to an empty list.

Following rules were adopted:

- Similar answers are treated the same and thus getting the same position in text. (An example would be the answers *#endangeredriver* and *#enddangeredriver*)
- Similar answer and portion of text are treated as the same word as in answer: *sen. gillebrand* and text: *@SenGillibrand*)
- The attribute *is_impossible* is set to *true* only when **all** the available answers cannot be assigned a position in the given tweet.
- New slang words which were not found in the Urban Dictionary were introduced such as TP for *toilet paper*, or *ellentube*.
- Wrong positions of the answers were corrected if encountered
- Unanswerable tweets are left as is. Example: *how many people liked this tweet?*: 202
- Tweet with the *this tweet is not showing* instead of QA pairs is left as is

Passage: This forecast is deflated as much as New England Patriots footballs!I apologize. W NJ has the most to lose. Dave Curren (@DaveCurren) January 27, 2015	
slang:	"this","forecast","is","deflated", "as", "much", "new", "england", "patriots", "i", "w", "nj", "has", "the", "most", "to", "dave", "curren", "january", "2015"

Table 11: Problematic tweet Nr. 4

The development set of the TweetQA SQUAD format was manually edited according to the rules above. Only around 2100 out of the 10689 question answer pairs were manually edited in the train dataset, due to the time consuming and dull nature of the task.

5.3 Model

5.3.1 T5

T5 or "Text-to-Text Transfer Transformer" is presented in *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer* (Raffel et al., 2019) from 2019. The main objective of the model is handling the input *and* the output as a text. In comparison to the BERT architecture T5 is using a casual decoder and additional alternative pre-training tasks instead of the original cloze task.

5.3.2 T5 model specifics

The parameters for the pre-trained T5 small and base model (both End-to-End and Question Generation) are given in Table 12.

Parameter	Models	
	Small	Base
dropout rate	0.1	0.1
heads	8	12
hidden layers	6	12
intermediate FF layer	2048	3072
maximum sequence length	512	512
vocab size	32102	32102

Table 12: Parameters for both T5 small and base model.

5.4 Training

The two basic models used were Question Generation and End-to-End Question Generation, explained further above. The parameters used for training of the models (*valhalla/t5-small-qg-hl, valhalla/t5-base-qg-hl, valhalla/t5-small-e2e-qg, valhalla/t5-base-e2e-qg*) using the available pipeline scripts are given in Table 13.

Parameter	Value
training batch size	16
evaluation batch size	16
epochs	15
learning rate	1e-4
evaluation during training	true

Table 13: Training parameters

6 Evaluation

6.1 Automatic Evaluation

6.1.1 Standard evaluation metrics

The standard evaluation metrics BLEU (1-4), METEOR, REOUGE_L and CIDEr were computed using the *nlg-eval* python module.

Results of the evaluation of automatically created SQUAD formatted TweetQA data As we can see the automatic evaluation measurements perform quite poorly, but still the fine-tuning on the pre-trained models did show some improvement. The results are displayed in Tables 14 and 15 for the qg models and Tables 16 and 17 for the e2e models.

Metric	Outcome small qg model	
	original	fine-tuned
Bleu 1	0.165630	0.346000
Bleu 2	0.099125	0.241399
Bleu 3	0.065634	0.178444
Bleu 4	0.045238	0.138167
METEOR	0.198628	0.222483
ROUGE_L	0.179915	0.367536
CIDEr	0.532227	1.431858

Table 14: Evaluation performed on automatically created TweetQA data, SQUAD format

Metric	Outcome base qg model	
	original	fine-tuned
Bleu 1	-	0.394109
Bleu 2	-	0.284917
Bleu 3	-	0.214532
Bleu 4	-	0.166719
METEOR	-	0.248101
ROUGE_L	-	0.417516
CIDEr	-	1.789576

Table 15: Evaluation performed on automatically created TweetQA data, SQUAD format

Results of the evaluation of manually edited SQUAD formatted TweetQA data As we can see the automatic evaluation measurements perform quite poorly, but still the fine-tuning on the pre-trained models did show some improvement. The results are displayed in Tables 18 and 19 for the qg models and Tables 20 and 21 for the e2e models.

Unfortunately there are no evaluation results for the original base qg model for both scenarios, since for the evaluation produces an error *RuntimeError: CUDA error: CUBLAS_STATUS_ALLOC_FAILED when calling 'cublasCreate(handle)'.*

6.1.2 BLEURT

As presented in 2020 the paper *BLEURT: Learning Robust Metrics for Text Generation* (Sellam et al., 2020) introduces BLEURT, a learned evaluation metric based on BERT. The model was trained on a public collection of ratings based on the WMT Metrics Task dataset.

The evaluation results for the manually edited TweetQA data for two T5 models are given in Table 22.

Metric	Outcome small e2e model	
	original	fine-tuned
Bleu 1	0.053403	0.332793
Bleu 2	0.026391	0.227647
Bleu 3	0.015307	0.164830
Bleu 4	0.009273	0.125187
METEOR	0.109512	0.212231
ROUGE_L	0.083067	0.351536
CIDEr	0.026161	1.306929

Table 16: Evaluation performed on automatically created TweetQA data, SQUAD format

Metric	Outcome small qg model	
	original	fine-tuned
Bleu 1	0.168221	0.351202
Bleu 2	0.101524	0.246024
Bleu 3	0.067962	0.182722
Bleu 4	0.047070	0.141755
METEOR	0.200778	0.225531
ROUGE_L	0.183403	0.371697
CIDEr	0.552163	1.479602

Table 18: Evaluation performedon manual edited TweetQA data, SQUAD format

Metric	Outcome base e2e model	
	original	fine-tuned
Bleu 1	0.064967	0.396125
Bleu 2	0.036087	0.285684
Bleu 3	0.022533	0.214982
Bleu 4	0.014561	0.166605
METEOR	0.127423	0.247074
ROUGE_L	0.100446	0.418279
CIDEr	0.050683	1.788318

Table 17: Evaluation performed on automatically created TweetQA data, SQUAD format

Metric	Outcome base qg model	
	original	fine-tuned
Bleu 1	-	0.400382
Bleu 2	-	0.293443
Bleu 3	-	0.222170
Bleu 4	-	0.173488
METEOR	-	0.252304
ROUGE_L	-	0.423129
CIDEr	-	1.833456

Table 19: Evaluation performed on manual edited TweetQA data, SQUAD format

6.2 Manual Evaluation

According to the paper *Best practices for the human evaluation of automatically generated text* from 2019 (van der Lee et al., 2019) some of the criteria listed in Table 23 are of interest when evaluating natural language models. Further they report a survey on NLG papers between 2005 and 2014, where 80% use automatic measures such as BLEU, METEOR, and ROUGE. The benefit of using such methods is a cheap, reproducible and fast way to perform checks. The problems with automatic measures are that they are often uninterpretable (ranging from not capable to capture the semantic and syntactic meaning and thus implement meaningful comparisons for a variety of NLP tasks such as text generation to having differing implementations of automatic metric evaluation across studies) and do not correlate with human evaluation.

The advise of the authors of (van der Lee et al., 2019) are to use either multiple-item 7-point Likert scales or a continuous ranking measurement. For the criteria choices the authors recommend to use separate ones rather than overall quality assessments and to properly define those.

For this work the chosen Likert Scale is a multiple-item 5-point one. As I was not able to find any clear definitions from any of the above criteria, I will define them myself. Inspirational were the following studies: *An evaluation of online machine translation of arabic into english news headlines: Implications on students' learning purposes* (Kadhim et al., 2013), *COSTA MT Evaluation Tool: An Open Toolkit for Human Machine Translation Evaluation* (Chatzitheodorou, 2013), *An Evaluation of Output Quality of Machine Translation (Padideh Software vs. Google Translate)* (Azer and Aghayi, 2015), *HEVAL: Yet Another Human Evaluation Metric*(Joshi et al., 2013) and *RankME: Reliable Human Ratings for Natural Language Generation* (Novikova et al., 2018).

Fluency & Naturalness How "fluent" and "natural" the generated question appears to be, without taking into account the correctness of the information? Could the question have been produced by a native speaker? The score description can be seen in Table 24.

Relevance How relevant is the generated question to the given text? Does it make sense

Metric	Outcome small e2e model	
	original	fine-tuned
Bleu 1	0.053089	0.333872
Bleu 2	0.026385	0.230703
Bleu 3	0.015345	0.169098
Bleu 4	0.009345	0.129525
METEOR	0.109112	0.215522
ROUGE_L	0.082893	0.353338
CIDEr	0.025654	1.348371

Table 20: Evaluation performed on manual edited TweetQA data, SQUAD format

Metric	Outcome base e2e model	
	original	fine-tuned
Bleu 1	0.065108	0.399541
Bleu 2	0.035949	0.291154
Bleu 3	0.022479	0.219786
Bleu 4	0.014611	0.170864
METEOR	0.127798	0.251499
ROUGE_L	0.101185	0.422600
CIDEr	0.048939	1.826197

Table 21: Evaluation performed on manual edited TweetQA data, SQUAD format

given the context? Here the correctness of the information should be taken into account. The score description can be seen in Table 25.

Grammaticality Is the generated question grammatically correct, without taking into account the correctness of the information? The score description can be seen in Table 26.

Readability Is the generated question easily read? The score description can be seen in Table 27.

Syntactic Correctness Is the syntax of the generated question correct? The score description can be seen in Table 28.

Non-redundancy Is there any redundancy in the generated question? The score description can be seen in Table 29.

Quality & Meaning How is the overall quality of the generated question? Is the question meaningful? How is the overall quality of the question in terms of its grammatical correctness, fluency, adequacy and other important factors? The score description can be seen in Table 30.

Slang Count How many slang words are in the generated question? Numerical unbounded

Metric	Outcome BLEURT	
	original	manually corrected
qg base t5		-0.397222
qg small t5		-0.540658
qg base t5 fine-tuned		-0.216805
qg small t5 fine-tuned		-0.368003

Table 22: Evaluation performed on manually edited TweetQA data, SQUAD format, T5 Question Generation model, pre-trained and fine-tuned

Criterion	Total
Fluency	13%
Naturalness	8%
Quality	5%
Relevance	5%
Grammaticality	5 %
Readability	4 %
Clarity	3%
Syntactic Correctness	3 %
Non-redundancy	2 %

Table 23: Chosen criteria used for human evaluation from all presented in (van der Lee et al., 2019).

outcome.

6.2.1 Results

The answers to the above described metrics were obtained using an interactive python script.

TweetQA dataset The model with the best results was the fine-tuned base T5 Question Generation model as shown in Table 19. Thus the generated questions from that model were taken and manually evaluated. The results are shown in Appendix A. According to the manual evaluation the generated questions were predominantly fluent, grammatically and syntactically correct, readable and contained almost no redundancy or any answers. The overall quality was also predominantly good. These evaluation findings contradict the automatically computed performance metrics from the previous section. The average length of a generated question is 39 characters, with 60 slang terms found in 1085 generated questions. An example of some of the generated questions with slang can be seen in Table 35.

An example of automatically creating a slang from a non-slang context and reference ques-

Score	Description
5	perfectly fluent and natural
4	almost fluent and natural
3	partially fluent and natural
2	almost not fluent and unnatural
1	totally not fluent and unnatural

Table 24: Fluency & Naturalness Scores

Score	Description
5	perfectly relevant
4	almost relevant
3	partially relevant
2	almost not relevant
1	totally not relevant

Table 25: Relevance Scores

Score	Description
5	perfectly grammatically correct
4	almost grammatically correct
3	partially grammatically correct
2	almost not grammatically correct
1	totally not grammatically correct

Table 26: Grammaticality Scores

Score	Description
5	perfectly readable
4	almost readable
3	partially readable
2	almost not readable
1	totally not readable

Table 27: Readability Scores

tions can be seen in Table 31, where *Madison Square Garden* is automatically abbreviated to *msg*, an example of an alphabetism. Also the reverse process occurred: from a tweet with slang, a non-slang question was generated, as seen in Table 31, which was presumably learned from the available reference questions.

An explanation about the difference in performance between manual and automatic evaluation is clearly illustrated in Table 33, where the reference questions are very different from the generated one, and thus the automatic metrics will perform poorly and yet the generated question is a meaningful one. Another similar example is also depicted in Table 34.

MetroLyrics The sample of lyrics used contains only 52 non redundant items: *english_cleaned_lyrics_edit.csv*.

The results of the manual evaluation regarding the four T5 models: question generation, small and base, as well as end-to-end question generation small and base are given in Appendix B, Appendix C, Appendix D and Appendix E. As one can see, the quality of the generated questions is not good and the results of the evaluation are very dispersed. Many of the questions generated with all of the four model already contained the answers from the lyrics in them, as well many redundancies. The overall quality is less than satisfactory.

This is all due to using the pre-trained models from scratch and not fine-tuning, due to lack of a proper corpus.

An example of some of the generated questions with slang can be seen in Table 36.

7 Discussion

7.1 Data

There are a lot of further improvements that can be done, such as using learned slang embeddings such as done by (Steven Wilson, 2020), initialize and pre-train an already available Transformer model, after which fine-tune on a particular downstream task such as Question Generation with Slang.

TweetQA may also not be representative of slang formation, and thus does not represent a noisy dataset. Another approach would be to scrap internet forums, public chats and subreddits for a representative slang dataset and create one tailored to the needs of the specific task. A rap lyrics dataset is also a very noisy slang linguistic corpus. The difficulties that arise from this approach however are the cost and time needed for manually creating and annotating such a dataset. Interesting resources that can be helpful are Online Slang Dictionary [10] as well as the project of Matt Bierner from the year 2016 [11] where he uses character level Recurrent Neural Network to generate new Urban Dictionary definitions.

Another idea is to incorporate emojis and their meanings, either via an available dictionary, or learn embeddings using ViLBERT. Following this idea it could be interesting to use the

[10] http://onlineslangdictionary.com/

[11] https://blog.mattbierner.com/urban-dictionary-neural-network/

Score	Description
5	has perfectly correct syntax
4	has almost correct syntax
3	has partially correct syntax
2	has almost incorrect syntax
1	has totally incorrect syntax

Table 28: Syntactic Correctness Scores

Score	Description
5	perfectly non-redundant
4	almost non-redundan
3	partially non-redundan
2	almost redundant
1	totally redundant

Table 29: Non-redundancy Scores

pictures accompanying the tweets again using ViLBERT (Lu et al., 2019).

7.2 Evaluation -DONE

The problem with the standard automatically computed measures should be clear by now, given the examples so far.

The problem with BLEURT is its bias: the authors state in the abstract of their paper that the metric is being trained on thousands of probably biased examples, and that the number of the examples is also not enough. Another negative point as pointed out by Priyanshu Sinha in his blog [12] would be that almost all state of the art natural language processing neural models make use of the transformer architecture and thus BLEURT provides zero explainability.

7.2.1 Manual Evaluation

Human evaluation comes with its own problems, one of which is the subjectivity of the assessed evaluation. In this particular case of course it is not relevant since there is no representative group of the human population, but just one person - me. So the sample size is 1, meaning that there is no statistical power whatsoever. I am also not a native speaker in English but one that is fluent enough to communicate and understand complex matters.

The subjectivity of the manual evaluation makes it unreliable. Some time a question may seem more readable than other times, depending on cognitive fatigue, motivation etc.

[12] https://medium.com/vision-ml/bleurt-failures-494623b26352

Score	Description
5	has perfect quality & is meaningful
4	has almost perfect quality & is almost meaningful
3	has partial perfect quality & is partially meaningful
2	has somewhat quality & is somewhat meaningful
1	has no quality & is not meaningful at all

Table 30: Quality & Meaning Scores

Tweet: Lacey Holsworth and her family made the trip to Madison Square Garden to watch Adreian Payne and the Spartans. Spartan Basketball (@MSU_Basketball) March 29, 2014
$Qref_1$: who went to madison square garden to watch adreian payne? A_1^1: lacey holsworth and her family A_2^1: lacey holsworth and her family
$Qref_2$: where did the spartans play? A_1^2: madison square garden A_2^2: madison square garden
Qgenerated: who made the trip to msg?

Table 31: Generated slang question and original tweet data.

8 Conclusion

The scripts and Jupyter notebooks can be found at Github [13].

9 Acknowledgement

My thanks go to the previous work done by Suraj Patil on Github [14].

References

Sumit Chopra Antoine Bordes, Nicolas Usunier and Jason Weston. 2015. Large-scale simple question answering with memory networks. *arXiv*.

Haniyeh Sadeghi Azer and Mohammad Bagher Aghayi. 2015. An evaluation of output quality of machine translation (padideh software

[13] https://github.com/mayaang/question_generation

[14] https://github.com/patil-suraj/question_generation

Tweet: @InStyle: On KWs Cover: Beautiful statement. Thank u 4 opening this convo. Its an important 1 that needs to be had.— kerry washington (@kerrywashington) February 5, 2015
$Qref_1$: what does kerry washington think of instyle's statement? A_1^1: it is beautiful A_2^1: it is beautiful
$Qref_2$: what is kerry thankful for? A_1^2: instyle opening the conversation A_2^2: instyle opening the conversation
Qgenerated: how does kerry washington feel about opening the conversation?

Table 32: Generated question from slang and original tweet data.

Tweet: .@pressclubdc statement: Turkish President Erdogan doesn't get to export abuse against journalists to America NPC President (@NPCPresident) March 31, 2016
$Qref_1$: what is being done to journalists? A_1^1: abuse A_2^1: abuse
$Qref_2$: what is the name of the turkish president? A_1^2: erdogan A_2^2: erdogan
Qgenerated: who doesn't get to export abuse against journalists to america?

Table 33: Generated question and original tweet data.

Tweet: I am currently wearing two scarves and a jacket at work. It's not a #fashion statement. I'm freezing. In June. #bostonweathersucks— April Grudier (@aprieve) June 6, 2017
$Qref_1$: what is she wearing at work? A_1^1: two scarves and a jacket A_2^1: two scarves and a jacket
$Qref_2$: what month is it in the tweet? A_1^2: june A_2^2: june
Qgenerated: when is aprie grudier freezing?

Table 34: Generated question and original tweet data.

vs. google translate). *Advances in Language and Literary Studies*, 6(4):226–237.

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, Hong Kong, China, November. Association for Computational Linguistics.

Konstantinos Chatzitheodorou. 2013. Costa mt evaluation tool: An open toolkit for human machine translation evaluation. *The Prague Bulletin of Mathematical Linguistics*, 100, 09.

Shehzaad Dhuliawala, Diptesh Kanojia, and Pushpak Bhattacharyya. 2016. Slangnet: A wordnet like resource for english slang. 05.

Roy Frostig Jonathan Berant, Andrew Chou and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. *EMNLP*.

Nisheeth Joshi, Iti Mathur, Hemant Darbari, and Ajai Kumar. 2013. Heval: Yet another human evaluation metric. *ArXiv*, abs/1311.3961.

Kais Kadhim, L.S. Habeeb, Ahmad Arifin, Zaharah Hussin, and M.M.R.T.L. Abdullah. 2013. An evaluation of online machine translation of arabic into english news headlines: Implications on students' learning purposes. *Turkish Online Journal of Educational Technology*, 12:39–50, 04.

Kettip Kriangchaivech and Artit Wangperawong. 2019. Question generation by transformers. *arXiv preprint arXiv:1909.05017*.

Vivek Kulkarni and William Wang. 2018. Simple models for word formation in english slang. 04.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. *ArXiv*, abs/1909.11942.

Y. Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Jiasen Lu, Dhruv Batra, D. Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *ArXiv*, abs/1908.02265.

Jekaterina Novikova, Ondrej Dusek, and Ver-

ena Rieser. 2018. Rankme: Reliable human ratings for natural language generation. In *NAACL-HLT*.

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

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. *arXiv e-prints*, page arXiv:1606.05250.

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

Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. 2020. Bleurt: Learning robust metrics for text generation. In *ACL*.

Barbara McGillivray Kiran Garimella Gareth Tyson Steven Wilson, Walid Magdy. 2020. Urban Dictionary Embeddings for Slang NLP Applications. *Proceedings of The 12th Language Resources and Evaluation Conference*.

Chris van der Lee, Albert Gatt, Emiel van Miltenburg, Sander Wubben, and Emiel Krahmer. 2019. Best practices for the human evaluation of automatically generated text. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 355–368, Tokyo, Japan, October–November. Association for Computational Linguistics.

Dongling Xiao, Han Zhang, Yukun Li, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2020. Ernie-gen: An enhanced multi-flow pre-training and fine-tuning framework for natural language generation. *ArXiv*, abs/2001.11314.

Wenhan Xiong, Jiawei Wu, Hong Wang, Vivek Kulkarni, Mo Yu, Xiaoxiao Guo, Shiyu Chang, and William Yang Wang. 2019. Tweetqa: A social media focused question answering dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

A Appendices

B TweetQS manual evaluation

In this appendix are the results from the manual evaluation performed on TweetQA, summarized in Figures 1, 2, 3, 4, 5, 6, 7 and 8.

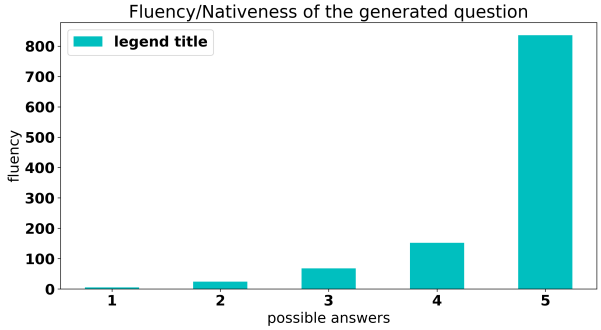

Figure 1: Do you find this generated question fluent? Could the question have been produced by a native speaker?

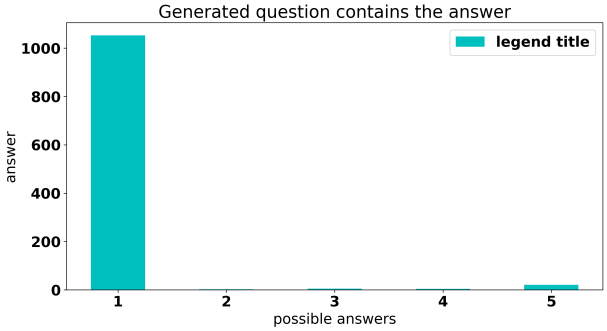

Figure 2: Does this generated question contain the answer?

C E-2-E base T5 model

In this appendix are the results from the manual evaluation performed on MetroLyrics, summarized in Figures 9, 10, 11, 12, 13, 14, 15 and 16. The average length of a generated question is 64 characters, with 23 slang terms found in 131 generated questions.

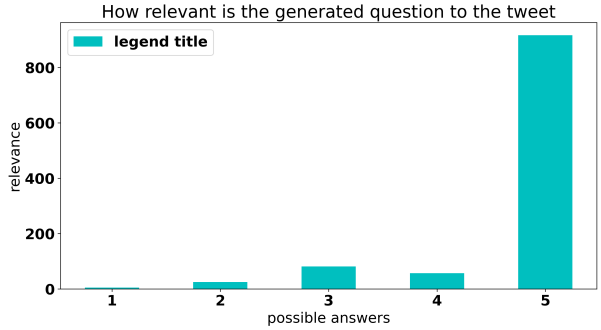

Figure 3: How relevant is the generated question to the given tweet?

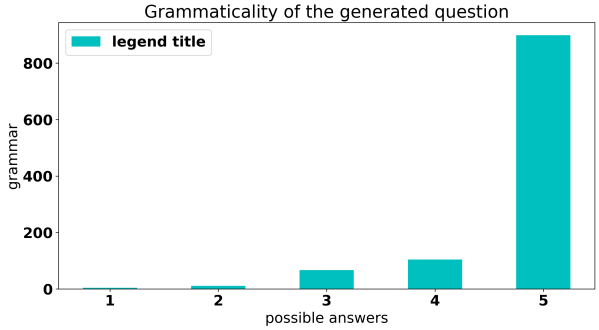

Figure 4: Is the generated question grammatically correct, without taking into account the correctness of the information?

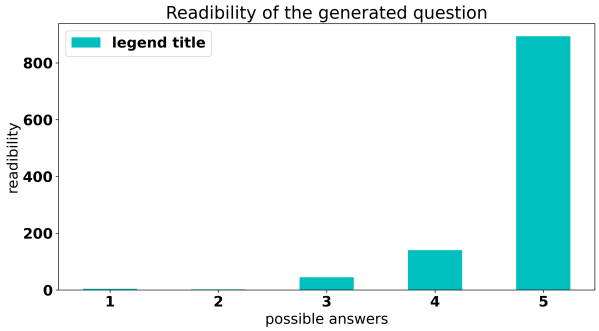

Figure 5: Is the generated question easily read?

D E-2-E small T5 model

In this appendix are the results from the manual evaluation performed on MetroLyrics, summarized in Figures 17, 18, 19, 20, 21, 22, 23 and 24. The average length of a generated question is 119 characters, with 29 slang terms found in 84 generated questions.

E QG small T5 model

In this appendix are the results from the manual evaluation performed on MetroLyrics,

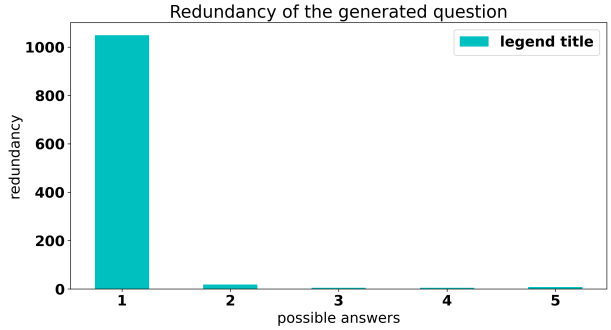

Figure 6: Is there any redundancy in the generated question?

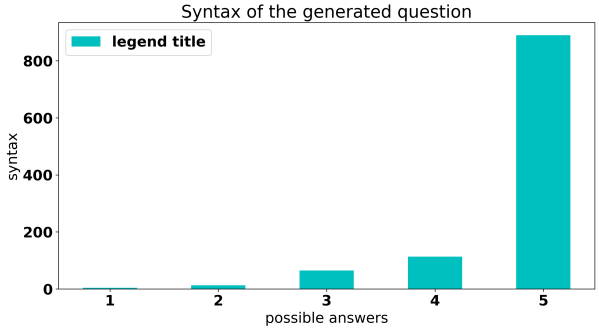

Figure 7: Is the syntax of the generated question correct?

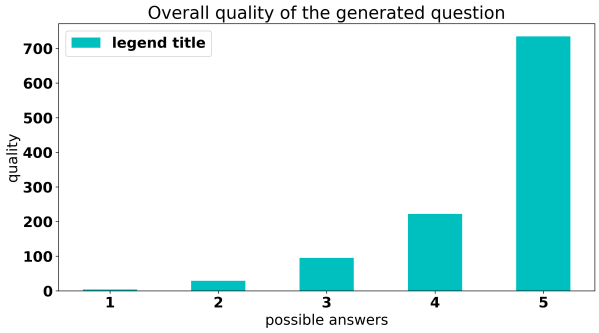

Figure 8: How is the overall quality of the generated question? Is the question meaningful?

summarized in Figures 25, 26, 27, 28, 29, 30, 31 and 32. The average length of a generated question is 73 characters, with 12 slang terms found in 34 generated questions.

F QG base T5 model

In this appendix are the results from the manual evaluation performed on MetroLyrics, summarized in Figures 33, 34, 35, 36, 37, 38, 39 and 40. The average length of a generated

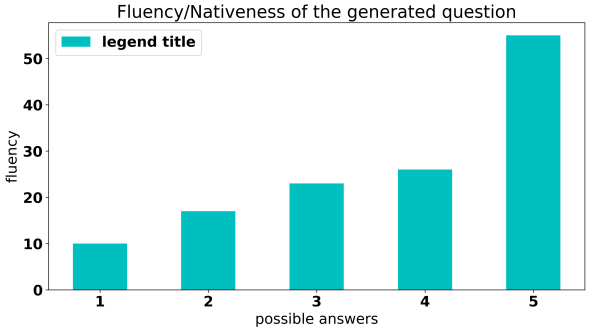

Figure 9: Do you find this generated question fluent? Could the question have been produced by a native speaker?

Sample questions with slang
who is elizabeth warren trying to bully, intimidate and slut-shame?
how far did miguel ace the 15th?
who is being cyberbullied?
what will we wear during bp?
what hashtag does alexa penavega use in her twe
who posted this video of him on ellentube?
where will john legere take his rants?
who is @bassellalsad shooing away?
who is reese witherspoon snuggling with?
future has flat out said who ain't his main priorit
what movie is the prop from?
who is in the 1st intermission?
what did selena & abel release?
what is @usatopinion doing to make cops 2 b cops?

Table 35: Questions with slang

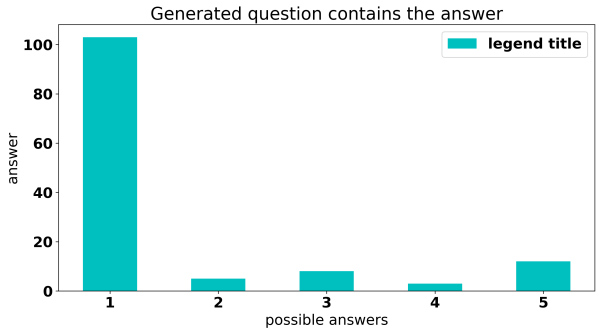

Figure 10: Does this generated question contain the answer?

question is 46 characters, with 2 slang terms found in 33 generated questions.

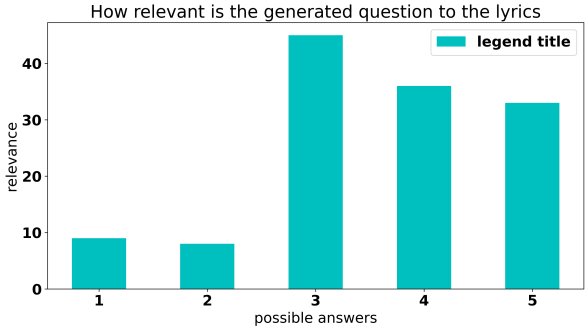

Figure 11: How relevant is the generated question to the given lyric?

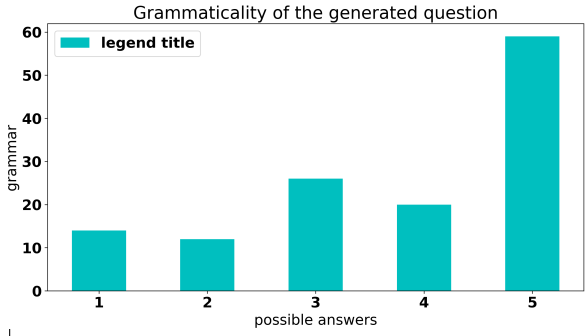

Figure 12: Is the generated question grammatically correct, without taking into account the correctness of the information?

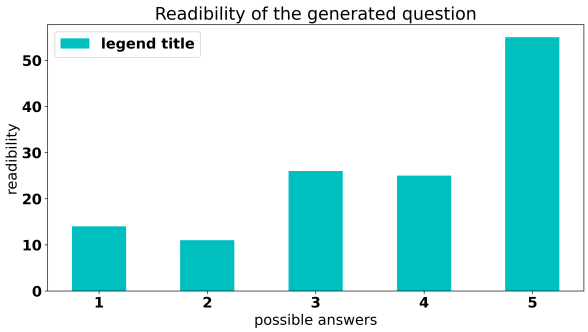

Figure 13: Is the generated question easily read?

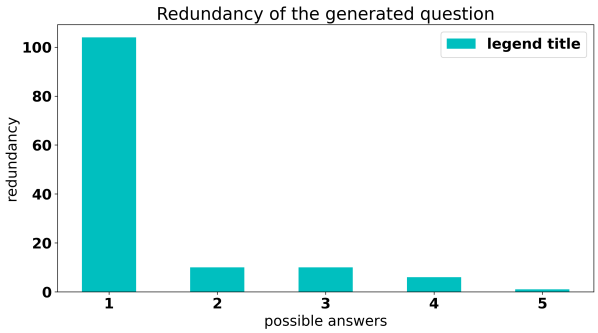

Figure 14: Is there any redundancy in the generated question?

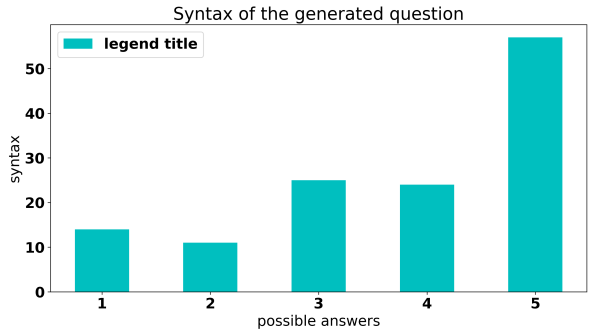

Figure 15: Is the syntax of the generated question correct?

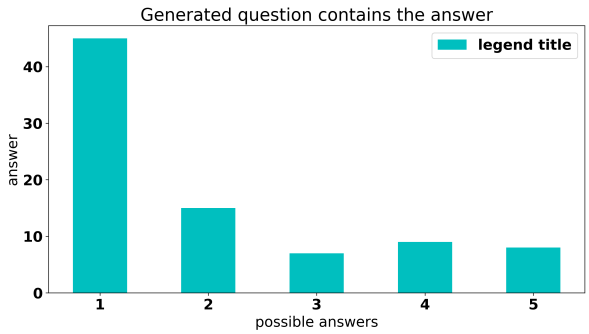

Figure 18: Does this generated question contain the answer?

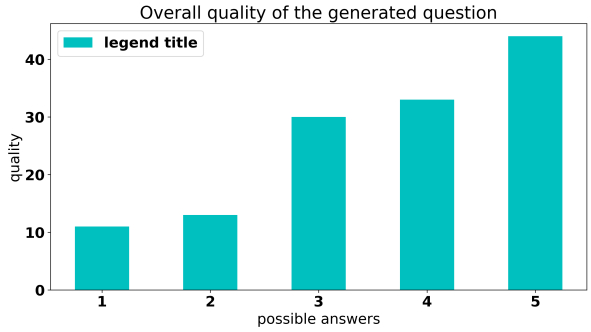

Figure 16: How is the overall quality of the generated question? Is the question meaningful?

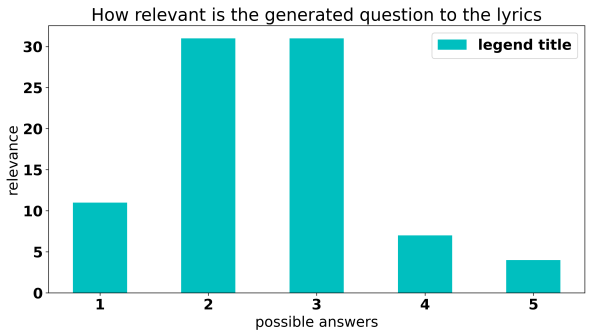

Figure 19: How relevant is the generated question to the given lyric?

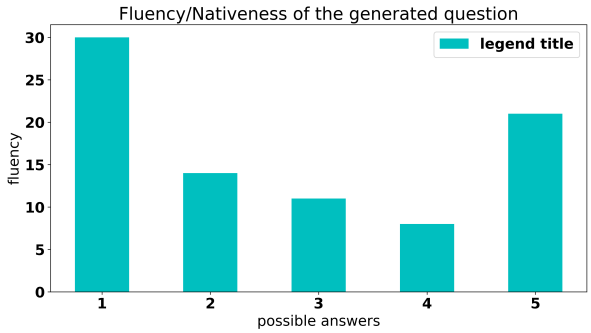

Figure 17: Do you find this generated question fluent? Could the question have been produced by a native speaker?

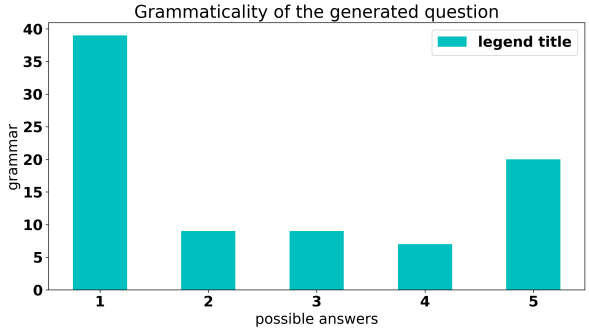

Figure 20: Is the generated question grammatically correct, without taking into account the correctness of the information?

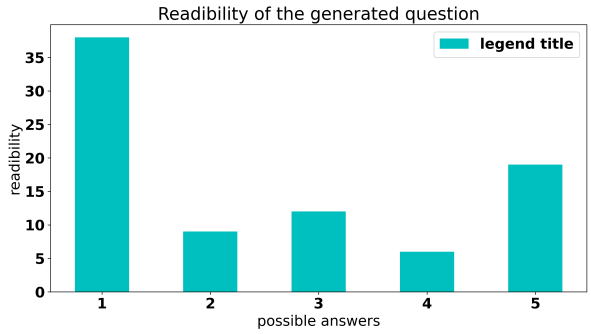

Figure 21: Is the generated question easily read?

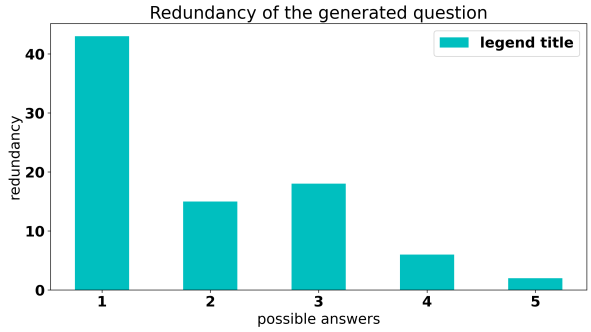

Figure 22: Is there any redundancy in the generated question?

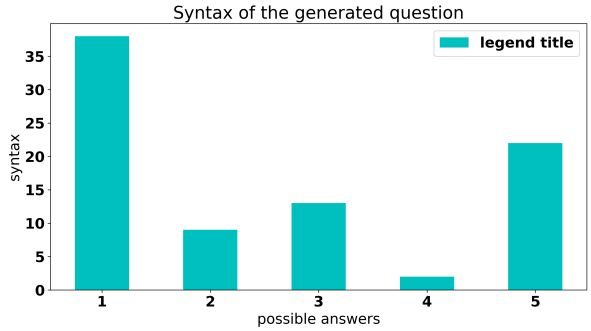

Figure 23: Is the syntax of the generated question correct?

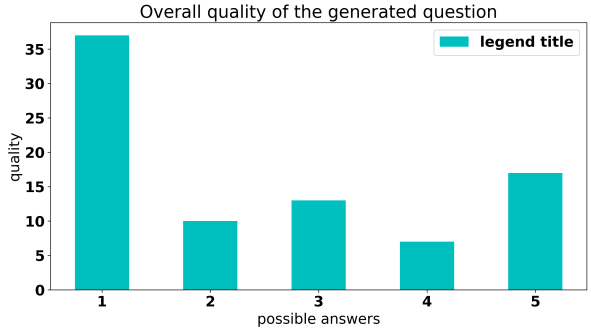

Figure 24: How is the overall quality of the generated question? Is the question meaningful?

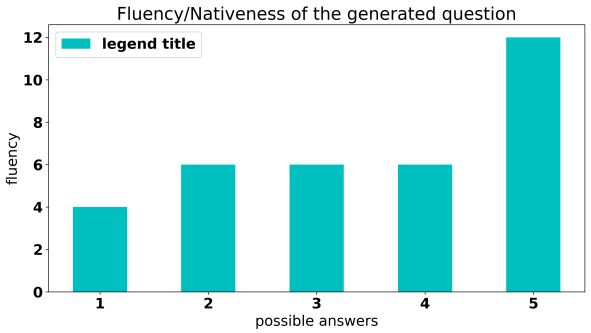

Figure 25: Do you find this generated question fluent? Could the question have been produced by a native speaker?

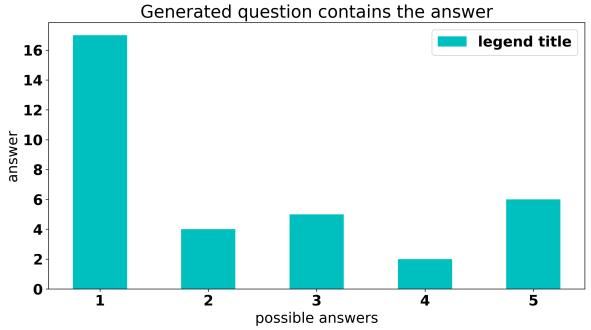

Figure 26: Does this generated question contain the answer?

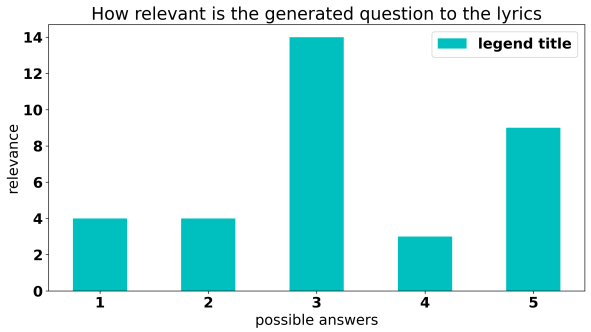

Figure 27: How relevant is the generated question to the given lyric?

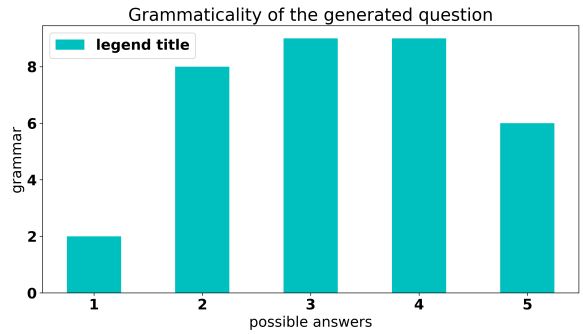

Figure 28: Is the generated question grammatically correct, without taking into account the correctness of the information?

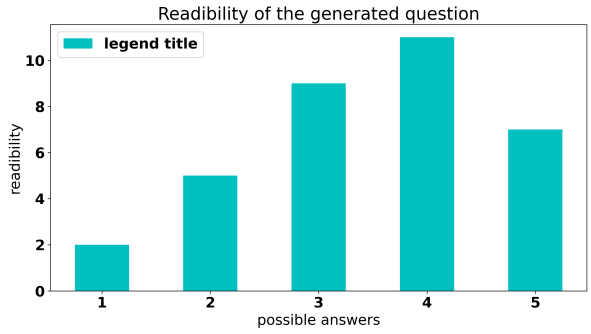

Figure 29: Is the generated question easily read?

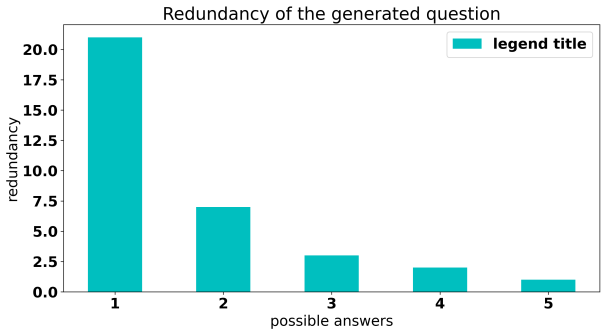

Figure 30: Is there any redundancy in the generated question?

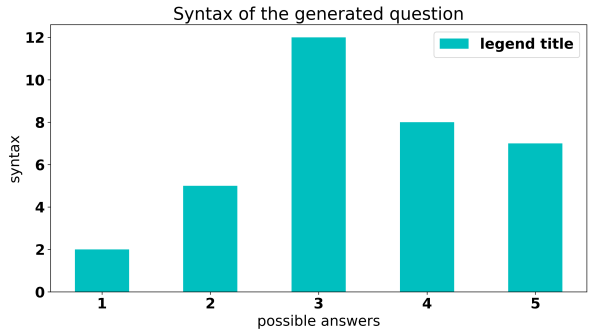

Figure 31: Is the syntax of the generated question correct?

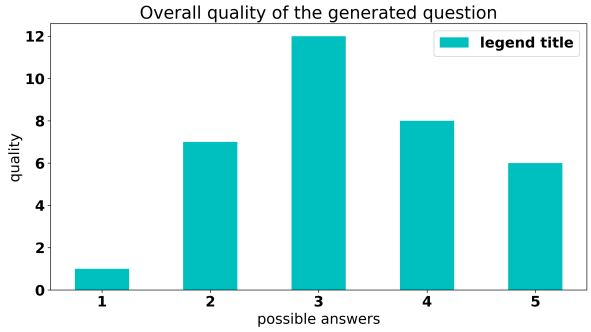

Figure 32: How is the overall quality of the generated question? Is the question meaningful?

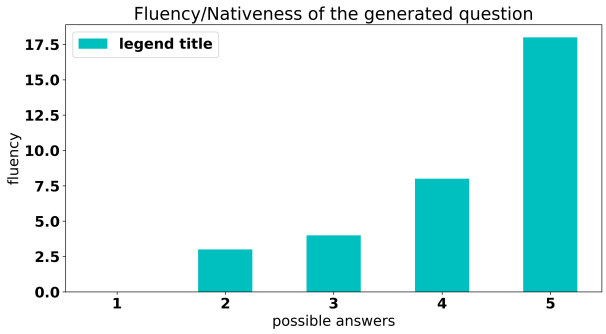

Figure 33: Do you find this generated question fluent? Could the question have been produced by a native speaker?

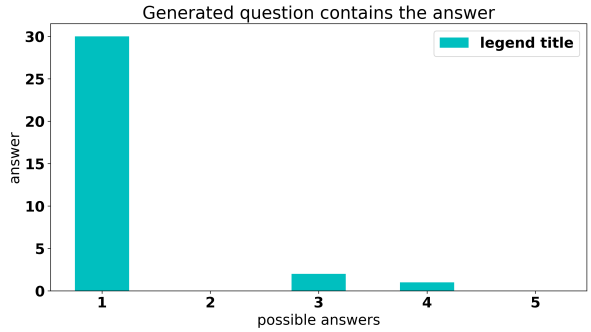

Figure 34: Does this generated question contain the answer?

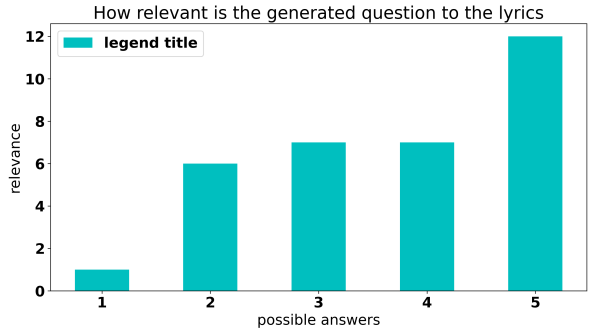

Figure 35: How relevant is the generated question to the given lyric?

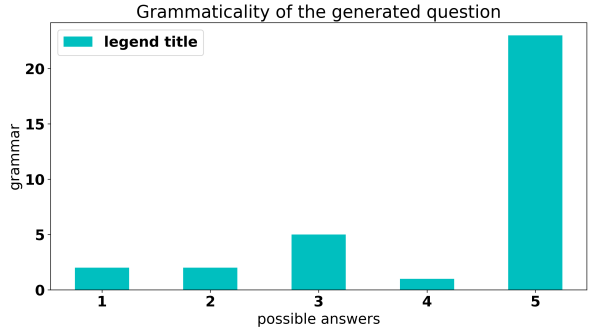

Figure 36: Is the generated question grammatically correct, without taking into account the correctness of the information?

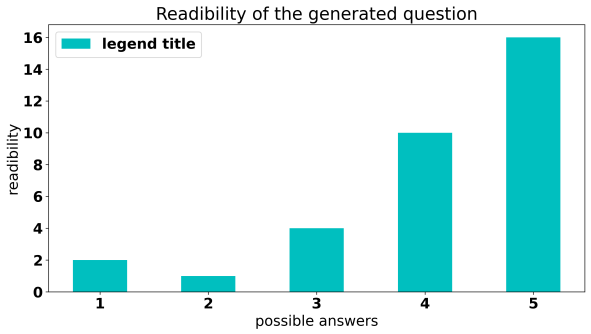

Figure 37: Is the generated question easily read?

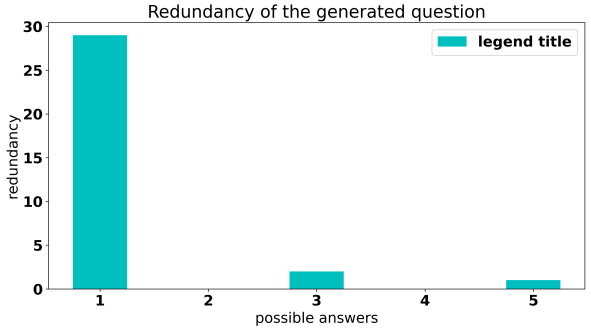

Figure 38: Is there any redundancy in the generated question?

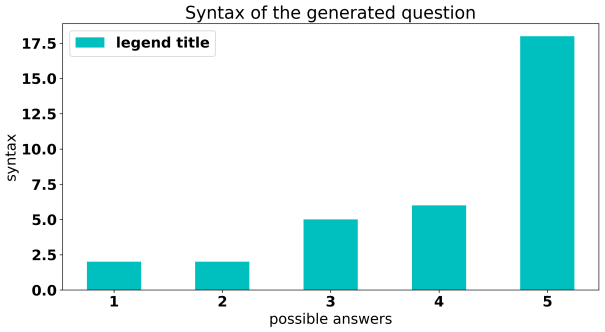

Figure 39: Is the syntax of the generated question correct?

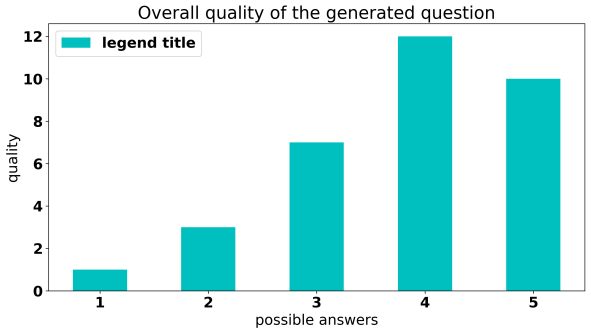

Figure 40: How is the overall quality of the generated question? Is the question meaningful?

Sample questions with slang	model
What is your favorite song they're gonna play?	e2e base
What kind of love just ain't enough?	e2e base
What is the name of the party that popping no sit around?	e2e small
What does Ay Ay Nobody like to be played?	e2e small
What do you say about leaving em alone because he's gonna hurt you?	qg small
What does Uhh Welcome to Hollywood baby do?	qg small
What's the name of the song that's gonna cut right to the chase?	qg base

Table 36: Questions with slang, Metrolyrics