

Multiple-choice Question Generation for the Chinese Language

Yicheng Sun¹, Hejia Chen² and Jie Wang¹ ^a

¹Department of Computer Science, University of Massachusetts, Lowell, MA, 01854, U.S.A.

²School of Computer Science and Technology, Xidian University, Xi'an, P.R.C.

Keywords: Multiple-choice Question Generation, Natural Language Processing.

Abstract: We present a method to generate multiple-choice questions (MCQs) from Chinese texts for factual, eventual, and causal answer keys. We first identify answer keys of these types using NLP tools and regular expressions. We then transform declarative sentences into interrogative sentences, and generate three distractors using geographic and aliased entity knowledge bases, Synonyms, HowNet, and word embeddings. We show that our method can generate adequate questions on three of the four reported cases that the SOTA model has failed. Moreover, on a dataset of 100 articles randomly selected from a Chinese Wikipedia data dump, our method generates a total of 3,126 MCQs. Three well-educated native Chinese speakers evaluate these MCQs and confirm that 76% of MCQs, 85% of question-answer pairs, and 91% of questions are adequate and 96.5% of MCQs are acceptable.

1 INTRODUCTION

Generating MCQs on a given article is often used to assess reading comprehension. An MCQ consists of a question-answer pair (QAP) and a number of distractors. An *adequate* MCQ should satisfy three requirements: (1) The question, answer, and distractors are contextually fit, grammatically correct, and conformed to native speakers. (2) The answer matches the question. (3) Each distractor provides enough confusion so that the right answer could be chosen only with some understanding of the underlying texts. An *adequate* QAP means that requirements 1 and 2 are satisfied. An *adequate question* means that requirement 1 is satisfied. An MCQ is *acceptable* if it is either adequate or can be made adequate with a minor effort such as changing, adding, or deleting one word or two. Contentwise, MCQs shall cover the main points of a given article.


Research on question generation (QG), first studied for the English language (Wolfe, 1976), has been focused on generating questions on declarative sentences.

QG research for the Chinese language, although started late, has made significant progress thanks to the recent developments of NLP tools and text-to-text transformers for the Chinese language. However,

we still face the following issues: (1) Some generated questions are syntactically or semantically incorrect or contain answers in the questions. (2) Some generated answers do not match the underlying questions. (3) Some generated distractors are not distracting enough. For example, the SOTA results produced by the Chinese Neural Question Generation (CNQG) (Liu and Zhang, 2022), a transformer-based method, generates questions with only 71.5% being adequate, which is substantially lower than the 90+% of generated QAPs being adequate for the English language (Zhang et al., 2022).

We attempt to narrow this gap with the following contributions: We present a novel method using SRL (semantic-role-labeling), POS (part-of-speech), and NER (named-entity-recognition) tags, as well as regular expressions, event extractions, and causal relations to generate a substantially larger number of adequate QAPs over declarative sentences. We can generate answers to be a sentence segment or a complete sentence to describe an event. We devise algorithms and create domain knowledge bases for geographic entities, alias of notable people, and use Synonyms, HowNet, and word embeddings to generate distractors that are semantically adequate with better distracting effect.

We implement these methods as a system called CMCQG (Chinese MCQ Generator) and show that CMCQG can generate adequate questions for three of

^a  <https://orcid.org/0000-0003-1483-2783>

the four reported cases that CNQG has failed (Liu and Zhang, 2022). Three native Chinese speakers with advanced degrees evaluate each generated MCQ on the quality of its question, answer, and distractors. On a dataset of 100 articles selected independently at random from a Chinese Wikipedia data dump, CMCQG generates a total of 3,126 MCQs. The evaluation results indicate that 76% of MCQs, 85% of QAPs, and 91% of questions are adequate, while 96.5% of MCQs are acceptable.

2 RELATED WORK

We summarize related work for the Chinese language. Questions may be generated from a given declarative sentence or a given passage using transformative methods, generative methods, or both. Transformative methods are based on syntax, semantics, and templates. Generative methods are neural-network models based on text-to-text transformers. In general, transformative methods have a better chance of generating grammatically correct QAPs than generative methods, while generative methods have been widely used for conversational dialogues.

Most existing question-generation methods are focused on factual WH-questions based on syntactic and semantic information. For example, the generation-and-ranking method (Liu et al., 2016)

transforms declarative sentences into questions and then selects questions based on their ranking. This method generates questions by straightforward replacement of a target word (the answer key) with an interrogative pronoun based on POS tags generated by the Language Technology Platform (LTP) (Che et al., 2010), which often results in coarse target selection and leads to unsatisfactory questions, particularly on more complex sentences.

The out-of-vocabulary problem is common in processing Chinese documents. A Stroke-Aware Copy Network model (Li et al., 2019) based on sequence-to-sequence (seq2seq) neural networks (Bahdanau et al., 2014; Cho et al., 2014) was devised to resolve this problem and treat question generation as translation. An interesting approach to handling logograms in its own right, the quality of QAPs generated by this method, however, falls in a similar category of low accuracy as that of the generative neural-network models for English (Du et al., 2017; Duan et al., 2017; Harrison and Walker, 2018; Sachan and Xing, 2018). Since a self-attention mechanism is used, the questions generated are likely to contain answers in them.

Incorporating rule-based transformation and neural-network models has produced promising re-

sults. One method transforms a declarative sentence into a question using a template-based method, and then using a multi-feature neural network to rank questions (Zheng et al., 2018). Another method (Liu et al., 2017) uses templates to generate questions from a knowledge graph and then uses a seq2seq neural network to modify them so that they would look more natural.

Built on extraction of subject, predicate, and object from an input sentence and a Chinese version of T5 (Text-To-Text Transfer Transformer) (Raffel et al., 2020), a Chinese Neural Question Generation was devised recently (Liu and Zhang, 2022) using multi-encoder sequence-to-sequence neural-net model augmented with knowledge graph triples, which generates a slightly over 70% ratio of adequate QAPs among those generated.

Much previous effort on generating distractors has focused on finding some forms of distractors, instead of making them look more distracting. To generate distractors with reasonable distractions, POS tags, word frequency, WordNet, domain ontology, distributional hypothesis, pattern matching, and semantic analysis have been used (Rao and Saha, 2018; Zhang et al., 2020).

3 CMCQG ARCHITECTURE

CMCQG uses a number of existing tools, including LTP to carry out Chinese NLP tasks, Semantic SentenceRank (SSR) (Zhang and Wang, 2021) or Contextual Network and Topic Analysis Rank (CNATAR) (Zhang et al., 2021) to rank sentence of a given document according to their relative importance and topic diversity, as well as Synonyms and HowNet to help generate distractors. CMCQG consists of three components: (1) Preprocessing. (2) QAP Generation, with three sub-modules: factual QAP (on person, time, number, and location), eventual QAP, and causal QAP. (3) Distractor Generation. Figure 1 depicts the architecture and data flow of CMCQG.

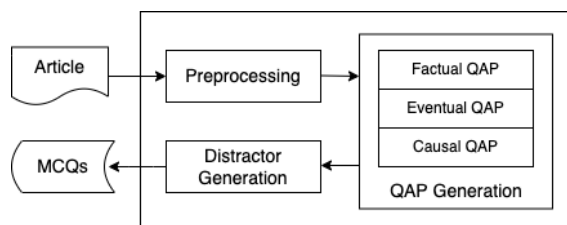


Figure 1: CMCQG architecture and data flow diagram.

LTP (Che et al., 2010) is a Chinese NLP platform that provides common NLP tools, including segmen-

tation of words and sentences, as well as POS, NER, and SRL tagging. LTP uses POS tags designed for the Chinese language and SLR tags defined in Prop-Bank. SRL tags consist of five argument (ARG) tags, denoted by A0 to A4., one REL tag for verbs, and a number of modification tags (ARGM). A0 represents agent; A1 patient; A2 instrument, benefactive, attribute; A3 starting point, benefactive, attribute; A4 ending point. In a sentence, A0 represents a subject, A1 an object, and A2–A4 represent other other types of arguments of predicates.

LTP can now produce POS, NER, and SRL tags with pretty high accuracy.

We use SSR (Zhang and Wang, 2021) and CNATAR (Zhang et al., 2021) to select important declarative sentences. These are unsupervised algorithms to rank sentences according to their relative importance and the underlying topic diversity in a given article.

We use Synonyms and HowNet to generate distractors. Synonyms is a Chinese synonym toolkit, which enables searching for synonyms and comparing sentence similarities using Word2vec. Its current vocabulary consists of 125,792 words. HowNet is an online knowledge base that provides interconceptual relations and interattribute relations of concepts, expressed in both Chinese and English bilingual lexicons. The sememe set of HowNet is determined by extracting, analyzing, merging and filtering semantics of thousands of Chinese characters. And the sememe set can also be adjusted or expanded in the subsequent process of annotating words. Each sememe in HowNet is represented by a word or phrase in both Chinese and English.

In what follows, we use “Q” for “question”, “A” for “answer”, and “D” for “distractor”. When using examples to illustrate concepts and algorithms, in addition to providing a Chinese sentence/phrase and its English translation, we also use an English-word-in-Chinese-sentence (EWICS) sentence for the purpose of illustrating various tags on different words or segments of the original sentence. An EWICS sentence is a Chinese sentence with each Chinese word in the sentence replaced with its English counterpart wherever possible. Verbs in Chinese do not have tense, participle, or person, and so the original form of verbs are always used in such a sentence. Chinese also have names, dates, and addresses in the reverse order. Thus, an EWICS sentence could look odd to native English speakers, and so we display it in italic to remind the reader of this fact.

4 PREPROCESSING

On a given document, the Preprocessing component first do the followings: (1) Segment words and tag each word or segment with an SRL tag, a POS tag, and an NER tag (where applicable) using LTP. (2) Rank sentences using CNATAR or SSL. (3) Remove sentences unsuitable for generating questions—only declarative sentences that include a complete set of SRL tags for subject, predicate, and object are deemed suitable for generating questions. This is done by removing interrogative sentences and imperative sentences based on ending punctuation, and sentences with one or more SR tags missing for subject, predicate, and object.

4.1 Segmenting Phrases

To generate adequate QAPs, it is necessary to segment phrases. For example, to generate a “who” question from “伊丽莎白女王出席了那个仪式 (Queen Elizabeth attended the ceremony)”, the question is “谁出席了那个仪式 (who attended the ceremony)?” The answer should be “伊丽莎白女王 (Queen Elizabeth)”, instead of just “女王 (Queen)” or “伊丽莎白 (Elizabeth)”. We use regular expressions of POS tags to recognize phrasal nouns and merge consecutive nouns representing a person name, time, location, and number into a phrasal name, phrasal time, phrasal location, and phrasal number. Likewise, we also merge consecutive verb, adverbs, and auxiliaries into a phrasal verb. The regular expressions we use to recognize such phrases are given below, where “/” indicates the end of a segmented word.

Phrase	Regular expression
Phrasal name	(ns/)*(n/)*(b/)*n/nh/
Phrasal Time	nt/(nt/)*nt/
Phrasal Location	ns/(ns/)*ns/
Phrasal Number	(m/wp/)*m/
Phrasal Verb	(d/)*v/u/

The POS tags used in these regular expressions and their meanings are shown below:

POS tag	Meaning
b	noun modifier
d	adverb, including negatives
m	numbers, numerical or ordinal
n	general noun
nh	person name
ns	geographic name
nt	temporal noun
u	denotes auxiliary
v	verb
wp	punctuation

We assign recognized phrases with a new POS tag for each type of phrase as, respectively, *nh*/, *nt*/, *ns*/, *m*/, and *v*/. For example, (1) 贝克 (Baker) (*nh*/) 州长 (governor) (*n*) is a phrasal name, and is merged to a phrase 贝克州长 (Governor Baker) (*nh*/). (2) 2022 (*nt*/) 年 (year) (*nt*/) 7 (*nt*/) 月 (month) (*nt*/) 4 (*nt*/) 日 (day) (*nt*/) is a phrasal time, and is merged to 2022年7月4日 (July 4, 2022) (*nt*/). (3) 日本 (Japan) (*ns*/) 东京 (Tokyo) (*ns*/) is a phrasal location, and is merged to 日本东京 (Tokyo, Japan) (*ns*/). (4) 100 (*m*/) , (/wp) 000 (*m*/) is a phrasal number, and is merged to 100,100 (*m*/). (5) 参加 (attend) (*v*/) 了 (/u) is a phrasal verb, and is merged to 参加了 (attended) (*v*/).

4.2 Segmenting Complex Sentences

Let S be a declarative sentence after phrases are segmented and new POS tags are assigned. We say that S is *simple* if it consists of one subject, one predicate, and one subject, with both subject and object being a single word or a phrase; S is *complex* if it contains at least two predicates. Complex sentences can be classified into three types: (1) S is *type-1 complex* if it has a subject-predicate-object structure with the subject, object, or both being a simple sentence. (2) S is *type-2 complex* if it consists of a main clause that is either a simple sentence or a type-1 complex sentence, and a few subordinate clauses each with a predicate and an object but without a subject, where the object in a subordinate clause may also be a simple sentence. (3) S is *type-3 complex* if it consists of a few independent sentences separated by commas or conjunctions.

Where there is no confusion, the word “complex” is omitted. We treat type-1 sentences as simple sentences. We segment a type-2 sentence into a set of simple sentences as follows: First extract the common subject shared by subordinate clauses, which is a prefix of the sentence ending at the word having an ARG tag A0. Then for each pair of nearest REL and A1 (i.e., no other REL or A1 tags between them), extract the words from REL to A1 and the words after until a punctuation of a comma or a period, and add the common subject in front of the extraction to form a sentence. Likewise, we segment a type-3 sentence into a set of sentences. We first extract the time and location phrases (if any) before the first subject as the common phrase of the new sentences. For each set of the nearest A0, REL, and A1 tags, we extract the word string from A0 to A1 plus the words after until a conjunction, a comma, or a period, and add the common phrases to this word string to form a new sentence. If a new sentence from segmentation is short (e.g., if it contains less than five words), then ignore it, for it

doesn't generate good questions. If all the new sentences are short, then keep the original sentence.

5 QAP GENERATION

In Chinese, a question is generated from a declarative sentence using an appropriate interrogative pronoun to replace the text to be asked. Different from English, there is no need to require subject-auxiliary inversion or verb segmentation. The QAP Generation component decides what to ask, and the content to be asked is the correct answer to a question to be generated. CMCQG generates two kinds of QAPs: (1) factual questions about people, time, locations, quantities, and general nouns, where answers are typically a single word or a single phrase; (2) event and causal questions, where answers are typically a sentence or a sentence segment.

5.1 Domain Knowledge Bases

To help generate adequate questions and distractors, we create two domain knowledge bases: Geographic Knowledge Base (GKB) and Alias Knowledge Base (AKB). GKB is a knowledge base for countries, states, provinces, and cities, as well as relations between them. We create GKB using a location detection dataset from Tencent (LocList.xml in the QQ program file), a phase abbreviation dataset (<https://baike.baidu.com/item/世界国家与地区一览表/850195>), a dataset for capitals of nations and provinces in China (<https://baike.baidu.com/item/省会/2089891>) and the states in the US (<https://baike.baidu.com/item/美国/125486>). We organize these datasets in a graph structure and devise an algorithm for fast searching. GKB consists of 145 countries, 878 states and provinces, and 6,295 cities worldwide, as well as standard abbreviations. AKB is a knowledge base for aliases of famous people. It is customary for the Chinese people in the past to have a name, a courtesy name, and a pseudonym, used interchangeably. It is also customary to use surname followed by a position to refer to a person. We create AKB by writing a web crawler to collect aliases from Baidu Baike based on the category of people in THUOCL (<http://thuocl.thunlp.org/>), which consists of 13,658 names sorted by frequencies. We organize these data in a graph structure and devise a search algorithm for fast searching.

Table 1: (Section 5.2) Regular expressions to identify single or multiple consecutive phrasal nouns, where c, p, q are POS tags representing, respectively, conjunction, prepositions, and quantity, $C = (\epsilon | p | (d)^*(u | v))$ means with or without p/, (d/)*u/, or (d/)*v/, and $C' = (\epsilon | p | u | v)$ means with or without p/, u/, or v/.

Type	Regular Expression	Comment
Person	nh/C	single nh/ followed by C
	nh/(wp/nh)*c/nh/C	multiple nh/'s followed by C
Location	(?<!p/ns)/C	single ns/ followed by C without /p in front
	(?<!p/ns)/(wp/ns)*c/ns/C	multiple ns/'s followed by C without p/ in front
	p/ns/(wp/ns)*c/ns/	multiple ns/'s with p/ in front
Time	(?<!p/c)nt/C'	single nt/ followed by C' and without /p or c/ in front
	(?<!p)nt/c/nt/C'	multiple nt/'s followed by C' and without /p in front
	p/nt/($\epsilon c nt$)	single or multiple nt/ with /p in front
Number	m/q/	numerical or ordinal number followed by quantity
	m/%/	numerical or ordinal number followed by percentage

5.2 Factual QAP

We use regular expressions shown in Table 1 to identify single or multiple consecutive such phrasal nouns for selecting answer keys appropriately after preprocessing with four types of phrasal nouns identified and new POS tags assigned. For example, to generate a factual question about one person or multiple persons (separated by a conjunction or a punctuation) appearing in a declarative sentence, we first determine which case the answer key belongs to. For a single person we could use interrogative pronoun 谁 (who) and for multiple persons we use 哪些人 (a plural form of who in Chinese).

To generate a factual question about time, we would need to determine if the answer is for a single point in time or a time range. For the former we could use 什么时间 (what time) as an interrogative pronoun, and for the latter we could use 在哪个时间段 (in which time frame). To generate a question that conforms better to what native speakers would say, we would need to determine a time unit so that we can use a proper word. For example, if the time unit is 年 (year), we could use 在哪年 (in what year) as an interrogative pronoun, and if it is 多年 (years), then we could use 在哪些年 (in what years) or 在哪几年 (in which years).

To generate a factual question about a single location or multiple locations, we would need to determine which case the answer key belongs to. We also want to know if the location is a country, a state/province, or a city using the GKB knowledge base. For example, asking about one country we could use 哪个国家 (which country), about several countries we could use 哪些国家 (which countries), and about several countries and regions we could use 哪些国家与地区 (which countries and regions).

To generate factual questions about numbers, numerical or ordinal, we would need to determine if it is

about people, organizations, animals, money, or different kinds of things to determine the correct quantifiers, for the Chinese language uses different quantifiers to represent different things and beings. For example, we could use 多少钱 (how much money) for money, 多少头 (how many) for farm animals, 多少只 (how many) for chickens, 第几名 (what rank) for people or things, and 百分之多少 (what is the percentage) for percentage. We do so using NER tags and a database on quantify words we created for various entities.

For subjects or objects that are nouns or phrasal nouns different from any of the above entities, we could use a generic interrogative pronoun 什么 (what) to replace a subject or an object to form a question. This would generate a syntactically correct question, but may not always read well, particularly when asking about an subject without a proper named entity. For example, for a sentence 苹果发布了新的手机产品 (Apple announced a new cellphone product), asking 哪家公司发布了新的手机产品 (Which company announced a new cellphone product) is semantically better than asking 什么发布了新的手机产品 (What announced a new cellphone product). We would need finer named-entity recognition and a new knowledge base on named entities. We would also need a better model to infer that 苹果 (Apple) in the context is a software company.

5.3 Eventual QAP

In a declarative sentence, if its subject or an object contain a subject-predicate-object structure, a subject-predicate structure, or a predicate-object structure, then such a segment, called a p-segment (shorthand for predicate-segment), is an event and we may ask about the subject or an object as follows:

At the top level of SRL, by assumption, both A0 and A1 exist. Without loss of generality, assume that

Table 2: (Section 5.3) SRLs on “房价 (home price) 上涨 (surging), 导致 (cause) 很多人 (many people) 无法 (unable) 买房 (buy house)”.

Sentence	房价 (home price)	上涨 (surging)	导致 (cause)	很多 (many)	人 (people)	无法 (unable)	买 (buy)	房 (house)
SRL sentence	A0		REL	A1				
SRL p-statement 1	A0	REL						
SRL p-statement 2				A0		REL	A1	

Table 3: (Section 5.4) Causal relationships and regular expressions to identify them, where %s is for matching a causal word.

Causal relationship	Regular Expression
cause-and-effect pair in the form of “[c-word][cause], [e-word][effect]”	(.*)((%s)(.*)((%s)(.*)((%s)(.*)((%s)(.)))
cause-and-effect pair in the form of “[e-word][effect], [c-word][cause]”	same as above
cause-and-effect pair in the form of “[effect-1][e-word][effect-2], [c-word][cause]”	same as above
causal word in the form of “[c-word][cause], [effect]”	(.*)((%s)(.*)((%s)(.*)((%s)(.)))
causal word in the form of “[effect], [c-word][cause]”	(.*)((%s)(.*)((%s)(.*)((%s)(.)))
causal word in the form of “[cause], [e-word][effect]”	(.*)((%s)(.*)((%s)(.*)((%s)(.)))

A0 appears on the left-hand side of REL and A1 on the right-hand side. Then the prefix that ends right before REL is the subject and the suffix that starts right after REL is the object. If the subject contains an REL tag, then call it p-statement 1. Likewise, if the object contains an REL tag, then call it p-statement 2.

We can ask about p-statement 1 (if it exists) by replacing it with an interrogative pronoun 什么事情 (what things) in the original sentence to generate a question, with p-segment 1 being the answer. Likewise, if p-segment 2 exists, then replace it with an interrogative pronoun 什么 (what) in the original sentence to generate a question, with p-segment 2 being the answer. This is the same as asking a noun object or a phrasal-noun object. For example, in sentence 房价上涨, 导致很多人无法买房 (home price surging, many people are unable to buy a house), the EWICS sentence is “home price surging, cause many people unable buy house”. Table 2 shows SRLs on this sentence. Then p-statement 1 is 房价上涨 (home price surging) and p-statement 2 is 很多人无法买房 (many people unable buy house). We generate the following two QAPs:

Q1 = 什么事导致很多人无法购买房屋? (what things cause many people unable buy house?)

A1 = 房价上涨 (home price surging).

Q2 = 房价上涨导致什么? (home price surging cause what?)

A2 = 很多人无法买房 (many people unable buy house).

5.4 Causal QAP

Given a cause-and-effect sentence, to generate a causal QAP, it is critical to extract the cause segment and the effect segment. We construct a list of causal

words and cause-and-effect pairs extracted from a Chinese Wikipedia dump and other documents, and devise regular expressions to extract cause-and-effect segments. We use a cause-and-effect segment and the underlying causal word or cause-and-effect pair to represent a causal statement.

In the Chinese language, there are three common types of cause-and-effect pairs of words and three types of causal words. A cause-and-effect pair of words has a causal word to lead the cause segment, denoted by c-word, and an effect word to lead the effect segment, denoted by e-word. Denote by [c-word] the c-word and [e-word] the e-word in the sentence, where [c-word] or [e-word] could be null, but cannot be both be null in a cause-and-effect sentence. Let [cause] denote the cause segment and [effect] the effect segment. In Chinese, the [cause] in a sentence can either appear before [effect] or after. A cause-and-effect sentence could either contain a cause-and-effect pair of words or just one causal word. For the latter, if it is a c-word, then [cause] can either appear before [effect] or after. If it is an e-word, then [effect] typically appears after [cause]. Listed in Table 3 are the common types of causal relationships and the corresponding regular expressions for extracting [cause] after a c-word and [effect] after an e-word based on a set of the causal words and cause-and-effect pairs collected from a Chinese Wikipedia dump.

To generate a causal question, we could either ask about the cause or about the effect using different interrogative pronouns depending on where [cause] appears before [effect] or after in a causal relationship.

Case 1: [cause] appears before [effect]. (1) Form a causal QAP about the cause as follows: Q = 什么 (what happened) [e-word][effect]? A = [cause]. (2)

Form a causal QAP about the effect as follows: Q = [cause][e-word]什么 (what)? A = [effect].

For example, from the sentence 因为澳大利亚发生了山火, 导致了很多人死亡 (a wildfire in Australia has caused many animals to die) we obtain [c-word] = 因为 (because), [e-word] = 导致了 (cause), [cause] = 澳大利亚发生了山火 (a wildfire in Australia), and [effect] = 很多人死亡了 (many animals have died). We generate a causal QAP about the cause: Q = 什么[e-word][effect] = 什么导致了很多人死亡 (what happened that caused many animals to die)? A = [cause] = 澳大利亚发生了山火 (a wildfire in Australia). We generate a causal QAP about the effect: Q = [cause][e-word]什么 = 澳大利亚发生了山火导致了什么 (What did a wildfire in Australia cause)? A = [effect] = 很多人死亡了 (many animals have died).

Case 2: [cause] appears after [effect]. (1) Form a causal QAP about the cause as follows: Q = 为什么 (why is that)[effect] ? A = [cause]. (2) Form a causal QAP about the effect as follows: Q = [cause]的结果什么 (outcome is what)? A = [effect].

For example, from the sentence 这家餐馆的点心之所以广受欢迎是因为他们有秘方 (this restaurant's dim sum dishes are [hence] so welcomed because they have secret recipes) we obtain the followings: [c-word] = 是因为 (because), [e-word] = 之所以 (hence), [cause] = 他们有秘方 (they have secret recipes), and [effect] = 这家餐馆的点心广受欢迎 (this restaurant's dim sum dishes are [hence] so welcomed by customers). We generate a causal QAP about the cause: Q = 为什么[effect] = 为什么这家餐馆的点心广受欢迎 (why is that the dim sum dishes in this restaurant are so welcomed)? A = [cause] = 他们有秘方 (they have secret recipes). We generate a causal QAP about the effect: Q = [cause]的结果是什么 = 他们有秘方的结果是什么 (what is the outcome of their secret recipes)? . The answer is 这家餐馆的点心广受欢迎 (the dim sum dishes in this restaurant are so welcomed).

6 DISTRACTOR GENERATION

For each QAP generated, we would want to generate three adequate distractors. One way to generate a distractor is to substitute an answer word (phrase) with an appropriate word or a phrase that maintains the original part of speech in the answer. For convenience, we refer to such a word or phrase as a *target* word. If the target word is a person name or names, use Synonyms and the alias knowledge base AKB to find distractors. For example, if the target word is a person, then distractors should be persons related

to the target word. For example, suppose the target word is 李小龙 (Bruce Lee) in a sentence about Kung Fu star, then distracting Kung Fu stars could be 杨斯 (Bolo Yeung), 成龙 (Jackie Chan), and 李连杰 (Jet Li).

If the target word is a location or locations, use the GKB knowledge base to find distractors. For example, if the target word is a city such as 纽约 (New York), then distractors should be cities related to the target word. In this case, distracting cities in the same league would be 波士顿 (Boston), 费城 (Philadelphia), and 芝加哥 (Chicago).

If the target word is a point in time, a time range, a numerical number, or an ordinal number, we use several algorithms to alter time and number in the same format, and randomly select one of these algorithms when generating distractors. For example, we may change the answer value at random, change the answer value at random but in a small range of values around the answer, and increase or decrease the answer value by one or two units.

For other nouns as target words in QAPs (such as those from eventual and causal QAPs), we devise a definition-based method to find distractor candidates from HowNet, and then select distractors based on Word2vec similarities. Each word in HowNet has a definition (DEF) tag that can be used as a hypernym of the word, making it easier to search for hyponyms under the same hypernym. In so doing, we can also avoid retrieving words that look similar but in different categories. For example, the DEF for 香蕉 (banana) is fruit, indicating that we should look for similar types of fruit as distractors.

In particular, given a target word w with its POS tag, we first search for w in HowNet to get its DEF tag. We then search for the words under this DEF tag as candidates of distractors. Finally, we use a pre-trained word-embedding database to retrieve the embedding vector for w and each candidate w' , denoted by $emb(w)$ and $emb(w')$, respectively. Remove w' if the cosine similarity of $emb(w)$ and $emb(w')$ is beyond a certain range (e.g., beyond the interval (0.5, 0.8)). The remaining candidates are distractors. If the number of distractors is less than 3, then change the similarity range until at least three distractors are found. Choose at random three distractors to be the final selection.

It is possible to use a list of word embeddings to find distractors. For example, let L be a database of Word2vec embeddings. Given a target word w , first search for its word-embedding representation $emb(w)$, then search for a number of embeddings with cosine similarities to $emb(w)$ within a certain range (not too big and not too small), and retrieve at ran-

Table 4: Among all four samples that CNQG produces inadequate questions (Liu and Zhang, 2022), CMCQG rights three but fails one, where CNQG-Q and CMCQG-Q mean, respective, the question generated by CNQG and CMCQG.

Sample 1	
Sentence	北美洲是世界上湖泊最多的洲，同时也是世界上最大的淡水湖区。(North America has the largest number of lakes among all continents, as well as the largest freshwater lake areas.)
CNQG-Q	世界上最大的淡水湖是什么？(What is the largest freshwater lake in the world?)
CMCQG-Q	哪个洲是世界上湖泊最多的洲？(Which continent has the largest number of lakes in the world?)
Sample 2	
Sentence	1925年3月，闻一多先生写下了名篇《七子之歌》，其中第五章是“广州湾”。(In March 1925, Mr. Wen Yiduo wrote his famous poem “The Song of Seven Sons”, where Chapter 5 is “Guangzhou Bay”.)
CNQG-Q	现代诗人是谁？(Who is the contemporary poet?)
CMCQG-Q	什么时候，闻一多先生写下了名篇《七子之歌》，其中第五章是“广州湾”？(When did Mr. Wen Yiduo write his famous poem “The Song of Seven Sons”, where Chapter 5 is “Guangzhou Bay”?)
Sample 3	
Sentence	李贺有“诗鬼”之称，是与“诗圣”之称杜甫、“诗仙”之称李白、以及“诗佛”王维相齐的唐代著名诗人。(Nicknamed “Poet Ghost”, Li He is a famous poet in the Tang dynasty and equally popular as “Poet Saint” Du Pu, “Poet Fairy” Li Bai, and “Poet Buddha” Wang Wei.)
CNQG-Q	谁被称为诗鬼称为？(Who is nicknamed poet ghost nicknamed?)
CMCQG-Q	是谁有“诗鬼”之称，是与“诗圣”之称杜甫、“诗仙”之称李白、以及“诗佛”王维相齐的唐代著名诗人？(Who is nicknamed “Poet Ghost”, a famous poet in the Tang dynasty and equally popular as “Poet Saint” Du Pu, “Poet Fairy” Li Bai, and “Poet Buddha” Wang Wei?)
Sample 4	
Sentence	鲁智深，小说《水浒传》中重要人物，人称花和尚。(Lu Zhisheng, a critical character in the novel “Water Margin”, nicknamed “Flowery Monk”.)
CNQG-Q	水浒传里的花是谁？(Who is the flower in novel “Water Margin”?)
CMCQG-Q	None. Reason: the SRL tags of the sentence do not contain a complete set of subject, predicate, and object.

dom three of these embeddings as distractors. However, this approach is more likely to produce inadequate distractors than using HowNet. For example, for a target word “Coca Cola”, the word “McDonald” would be selected as a distractor while the question is about drinks.

7 EVALUATIONS

The CNQG model paper (Liu and Zhang, 2022) provide four concrete samples that CNQG fails to generate adequate QAPs from input sentences. On these sentences, CMCQG is able to generate adequate QAPs for three but fails for one because that sentence does not produce the desired SRL tags (see Table 4).

We use a Chinese Wikipedia data dump at github.com/brightmart/nlp_chinese_corpus that consists of over 1 million articles to evaluate CMCQG. In particular, we select from it 100 articles independently at random, which contain a total of 2,767 sentences. Removing interrogative sentences, imperative

sentences, and declarative sentences with fewer than 8 words, there are 1,273 declarative sentences that contain a subject, a predicate, and an object. Segmenting type-2 and type-3 complex sentence except causal-relationship sentences, CMCQG generates 1,465 sentences. CMCQG generates factual QAPs on person, location, time, and number (excluding questions with subject being a general noun) from applicable simple and type-1 simple sentences whenever possible. For the remaining sentences CMCQG generates eventual and causal QAPs whenever possible. A total of 3,243 QAPs and a total of 3,126 MCQs are generated. CMCQG failed to generate three distractors for 117 QAPs (i.e., it only generated less than three distractors for these QAPs). The breakdown of the 3,126 generated MCQs is as follows: Person: 643; Location: 627; Time: 397; Number: 599; Eventual: 871; Causal: 79.

Three native Chinese speakers with advanced degrees evaluated the quality of these MCQs. To maintain a balance between categories, we selected 100 MCQs independently at random from each category

Table 5: Overall evaluation results.

	A-MCQ	P-MCQ	P-QAP	P-Q	UA-Q	UA-A	UA-D	UA-MCQ
Number	545	440	494	526	0	0	4	0
Percentage	96.5	76.0	85.3	90.8	0	0	0.7	0

Table 6: The percentage of adequate MCQs in each category.

Person	Location	Time	Number	Eventual	Causal
85	73	80	82	65	69.6

Table 7: The percentage of adequate questions, adequate answers, and adequate distractors in each category.

	Person	Location	Time	Number	Eventual	Causal	Overall
Question	95	95	90	90	87	87	90.8
Answer	95	95	89	88	90	88.6	91.0
Distractors	89	91	91	85	80	83.5	86.7

Table 8: Average scores in each category (category max: 2; overall max: 6).

	Person	Location	Time	Number	Eventual	Causal	Average
Question	1.980	1.983	1.963	1.870	1.820	1.860	1.920
Answer	1.980	1.983	1.927	1.950	1.933	1.911	1.950
Distractors	1.840	1.917	1.940	1.870	1.840	1.850	1.880
Overall	5.800	5.883	5.830	5.660	5.593	5.621	5.749

with over 100 questions, for a total of 579 MCQs. Listed below are evaluation criteria:

A question is (1) *adequate* if it is grammatically correct (syntactically and semantically) and relevant to the meaning of the underlying declarative sentence; (2) *acceptable* if it is relevant, but contains a minor grammatical error or does not conform to what a native speaker would say; (3) *unacceptable* if it is either irrelevant or contains serious grammatical errors.

An answer is (1) *adequate* if it is grammatically correct and match well with the underlying question; (2) *acceptable* if it is understandable and can be made adequate after a minor modification; and (3) *unacceptable* if it does not make sense.

Distractors are (1) *adequate* if each of the three distractors is grammatically correct and relevant to the question with distracting effects; (2) *acceptable* if one or two distractors are adequate; and (3) *unacceptable* if none of the distractors is adequate.

For each part of an MCQ, assign 2, 1, and 0 to, respectively, adequate, acceptable, and unacceptable. An MCQ is adequate if all its three parts are adequate, and acceptable if each of its three parts is at least acceptable. Adequate QAP and Q are similarly defined. Tables 5–8 depict the average evaluation results of the three judges, where prefixes P, A, and UA stand for “adequate”, “acceptable”, and “unacceptable”, respectively.

Since replacing a selected content with an interrogative pronoun is syntactically correct in Chinese, all generated questions are indeed syntactically cor-

rect. Thus, when a question is grammatically incorrect it means that it is semantically incorrect, meaning that it asks for a wrong thing, which is due to incorrectly POS tagging. For example, the Japanese name 五味太郎 (Gomi Taro) should have been given a POS tag *nh/* to indicate it is a person name, but it is wrongly tagged as a number *m/* because it has a number 五 (five) in it, causing a wrong thing to be asked. This problem can be fixed with a better named-entity recognition.

We can make some inadequate QAPs adequate using a better knowledge base for units of measurement and more accurate POS and NE taggers. Some acceptable answers include more information than what is asked for due to the added information to new sentences from segmentation of complex sentences. Thus, improving sentence segmentation would help resolve this problem. Some acceptable distractors can be easily fixed by improving the coverage of GKB and AKB. Fixing other errors would need a better algorithm that can identify more appropriate synonyms of a target word according to its context, particularly for polysemous words.

8 CONCLUSIONS

Built on successes of NLP research on the Chinese language, CMCQG generates MCQs on direct points from declarative sentences of a given article with sat-

isfactory results. CMCQG is transformative and it tends to generate well-formed QAPs, but the interrogative sentences it generates, while being grammatically correct, tend to be rigid and dogmatic. Generative methods based on text-to-text transformers tend to generate interrogative sentences that are more vivid, they also tend to generate silly questions. It would therefore be interesting to investigate how to combine these two seemingly opposite approaches and construct a complementary method.

Generating MCQs on derived points of a given article is more interesting and much more difficult. Machine inference over a set of declarative sentences that derives aggregate QAPs for certain types of questions may be a fruitful direction. For example, we may be able to identify cause-and-effect relationships among multiple sentences and generate MCQs based on such relations.

REFERENCES

- Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Che, W., Li, Z., and Liu, T. (2010). LTP: A Chinese language technology platform. In *COLING 2010, 23rd International Conference on Computational Linguistics, Demonstrations Volume, 23-27 August 2010, Beijing, China*.
- Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.
- Du, X., Shao, J., and Cardie, C. (2017). Learning to ask: neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1342–1352, Vancouver, Canada. Association for Computational Linguistics.
- Duan, N., Tang, D., Chen, P., and Zhou, M. (2017). Question generation for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 866–874, Copenhagen, Denmark. Association for Computational Linguistics.
- Harrison, V. and Walker, M. (2018). In *Neural generation of diverse questions using answer focus, contextual and linguistic features*. Association for Computational Linguistics.
- Li, W., Kang, Q., Xu, B., and Zhang, L. (2019). Sac-net: Stroke-aware copy network for chinese neural question generation. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*.
- Liu, M., Rus, V., and Liu, L. (2016). Automatic chinese factual question generation. *IEEE Transactions on Learning Technologies*, 10(2):1–1.
- Liu, M. and Zhang, J. (2022). Chinese neural question generation: Augmenting knowledge into multiple neural encoders. *Applied Sciences*.
- Liu, T., Wei, B., Chang, B., and Sui, Z. (2017). Large-scale simple question generation by template-based seq2seq learning. In *National CCF Conference on Natural Language Processing and Chinese Computing*.
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer.
- Rao, D. and Saha, S. K. (2018). Automatic multiple choice question generation from text : A survey. *IEEE Transactions on Learning Technologies*, pages 14–25.
- Sachan, M. and Xing, E. (2018). Self-training for jointly learning to ask and nnsver questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 629–640, New Orleans, Louisiana. Association for Computational Linguistics.
- Wolfe, J. H. (1976). Automatic question generation from text-an aid to independent study. In *Proceedings of the ACM SIGCSE-SIGCUE technical symposium on Computer science and education*, pages 104–112.
- Zhang, C., Sun, Y., Chen, H., and Wang, J. (2020). Generating adequate distractors for multiple-choice questions. In *The 12th International Conference on Knowledge Discovery and Information Retrieval (KDIR)*.
- Zhang, C., Zhang, H., Sun, Y., and Wang, J. (2022). Transformer generation of question-answer pairs with pre-processing and postprocessing pipelines. In *The 22th ACM Symposium on Document Engineering (DocEng)*. ACM.
- Zhang, H. and Wang, J. (2021). An unsupervised semantic sentence ranking scheme for text documents. *Integrated Computer-Aided Engineering*, pages 17–33.
- Zhang, H., Zhou, Y., and Wang, J. (2021). Contextual networks and unsupervised ranking of sentences. In *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 1126–1131. IEEE.
- Zheng, H.-T., Han, J., Chen, J., and Sangaiah, A. K. (2018). A novel framework for automatic chinese question generation based on multi-feature neural network model. *Computer Science and Information Systems*, 15:487–499.