# Taking up the Gaokao Challenge: An Information Retrieval Approach

**Gong Cheng**[*†], **Weixi Zhu**[*], **Ziwei Wang, Jianghui Chen, Yuzhong Qu**
National Key Laboratory for Novel Software Technology,
Nanjing University, Nanjing 210023, China

## Abstract

Answering questions in a university's entrance examination like Gaokao in China challenges AI technology. As a preliminary attempt to take up this challenge, we focus on multiple-choice questions in Gaokao, and propose a three-stage approach that exploits and extends information retrieval techniques. Taking Wikipedia as the source of knowledge, our approach obtains knowledge relevant to a question by retrieving pages from Wikipedia via string matching and context-based disambiguation, and then ranks and filters pages using multiple strategies to draw critical evidence, based on which the truth of each option is assessed via relevance-based entailment. It achieves encouraging results on real-life questions in recent history tests, significantly outperforming baseline approaches.

## 1 Introduction

A recently very hot AI challenge is to have the computer pass entrance examinations at different levels of education. The Todai Robot Project [Fujita *et al.*, 2014] aims to develop a problem solving system that can pass the University of Tokyo's entrance examination. China has launched a similar project focusing on four out of the nine subjects tested in the National Higher Education Entrance Examination (commonly known as Gaokao), namely Chinese, mathematics, geography, and history. Recently, the Project Aristo [Clark, 2015] invites contributions to passing the Elementary School Science and Math Exams.

As a preliminary attempt to take up the Gaokao challenge, we aim to develop an approach to automatically answering multiple-choice questions in Gaokao. Figure 1 shows an example question in history tests, consisting of a stem and four options as possible answers. The stem is different from and more complex than one-sentence-long factoid questions handled by existing approaches to question answering. It starts with several background sentences that are indispensable for understanding and answering the question; without that, the lead-in sentence at the end of the stem would not be a self-contained question. For instance, the lead-in sentence in Fig. 1 would be confusing without knowing from the previous background sentences that *those relatives* refer to those being feoffed. Besides, such a stem often includes domain-specific expressions, like quotes in Classical Chinese in history (which are difficult to understand by speakers of modern Chinese), formulas in mathematics, and maps in geography. Multiple-sentence-long questions[1] and their domain-specific expressions would invalidate existing solutions to semantic parsing and question answering [Kolomiyets and Moens, 2011; Kwiatkowski *et al.*, 2013; Berant and Liang, 2014; Yih *et al.*, 2015]. The task here also differs from the reading comprehension tasks in QA4MRE [Peñas *et al.*, 2013] and MCTest [Richardson *et al.*, 2013]. In that task, the correct answer resides in a given document, whereas to answer a question in Gaokao, searching external resources for additional knowledge[2] would be an essential component of the problem solving process. For instance, the question in Fig. 1 aims to test whether a student is aware that *feoffments in the Zhou Dynasty mainly consisted of land and population*; this fact is certainly not given in the stem.

> The *Effectiveness of Confucianism* chapter of the *Xunzi* said: "(The King of Zhou) ruled the land and founded seventy-one (feudal) states, of which fifty-three (governors) were from the Ji family." It shows that in the Zhou Dynasty, feoffments were mainly granted to relatives. The King of Zhou would invest those relatives with
>
> A. title and cattle B. land and cattle
> C. land and population (the correct answer) D. population and iron plows

Figure 1: An example multiple-choice question in history tests, translated from Chinese. In particular, the quote underlined is translated from Classical Chinese.

The above distinctive features of the challenge would call for a new problem solving framework for automatically answering a multiple-choice question in Gaokao. We propose a three-stage framework as shown in Fig. 2. Firstly, knowledge that is relevant to a given question is recollected from a memory. Secondly, after filtering out non-essential knowledge, ev-

[*]Gong Cheng and Weixi Zhu are co-first authors.
[†]Corresponding author: Gong Cheng (gcheng@nju.edu.cn).

[1]For example, in the 2014 Gaokao in Beijing, the average number of sentences contained in the stem of a multiple-choice question in the history test is 2.5, and the maximum number is 5.

[2]The term "knowledge" is used in a common sense in this paper.

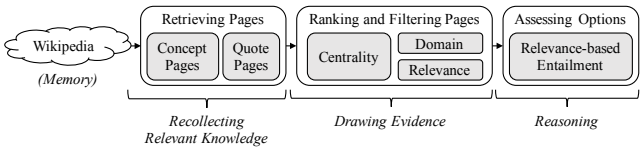


Figure 2: Overview of the approach.

idence is drawn. Finally, based on the evidence, reasoning is performed to check each option; and the option that is most likely to be true will be returned as the answer.

In our implementation of the framework, as shown in Fig. 2, (Chinese) Wikipedia is taken as the memory since it covers a wide range of knowledge and is freely accessible. Relevant knowledge is recollected by retrieving a set of pages from Wikipedia that match the question in two specific ways and thus may provide useful knowledge for answering the question. The subset of pages that are truly useful are regarded as evidence, which is drawn by ranking the retrieved pages according to multiple strategies for evaluating their usefulness, and filtering out bottom-ranked ones. The contents of the remaining pages will entail each option to some extent, based on which its truth is assessed.

Our contribution is threefold.

- We are among the first to take up the Gaokao challenge. After showing its difficulty and its difference from existing research problems, we propose a three-stage framework for answering multiple-choice questions.
- Our implementation of the framework exploits and extends information retrieval techniques. It retrieves pages from Wikipedia by string matching and context-based disambiguation, ranks and filters pages according to centrality, problem domain, and relevance, and accesses the truth of each option via relevance-based entailment.
- Our approach achieves encouraging results on a set of real-life questions collected from recent history tests. The dataset is accessible to the research community.

## 2 Approach

A multiple-choice question $q$ consists of a string $s_q$ called stem and a set of strings $O_q$ called options; only one of $O_q$ is the correct answer to $s_q$ to be found. Our approach to answering $q$ comprises three stages: retrieving pages, ranking and filtering pages, and assessing options.

### 2.1 Retrieving Pages

To acquire knowledge for answering $q$, two types of pages that match $q$ in different ways are retrieved from Wikipedia: concept pages and quote pages.

#### Retrieving Concept Pages

Some concepts mentioned in $q$, like *the Zhou Dynasty* and *feoffment* in Fig. 1, can give clues about $q$; their descriptions may provide useful knowledge for answering $q$. The description of a concept can be acquired from a Wikipedia page titled this concept, called a *concept page*. To retrieve concept pages, we match the titles of Wikipedia pages to $q$. Specifically, we adopt the simple yet effective leftmost longest principle. As shown in Algorithm 1, by repeatedly searching the stem $s_q$ (as the input $t$) for leftmost longest matches of page titles and skipping those on a predefined stopword list, a set of concept pages $P_C(s_q)$ are retrieved. Each option $o \in O_q$ is processed analogously, resulting in $P_C(o)$.

**Algorithm 1:** Retrieving Concept Pages

**Data:** A set of Wikipedia pages $P$ and a string $t$ to be matched.
**Result:** A set of concept pages $P_C(t) \subseteq P$.

```
Initialize P_C(t) to an empty set;
while t is not an empty string do
    Find the subset of pages P' ⊆ P whose titles are prefixes of t;
    if P' is not an empty set then
        Find p ∈ P' that has the longest title lt;
        if lt is not on the stopword list then
            Add p to P_C(t);
        Remove the prefix lt from t;
    else
        Remove the first character (or word) from t;
return P_C(t);
```

Feudalism may refer to:
- Feudalism (in China), existed during the Shang and the Zhou dynasty, and was replaced by centralization of authority during and after the Qin dynasty…
- Feudalism (in Europe), prevailed in the Middle Ages from the 5th to the 15th century…
- Feudalism (in Japan), originated from ritsuryo in the Heian period from the 8th to the 12th century…

Figure 3: The contents of an example disambiguation page, translated from Chinese. (Links are underlined.)

A retrieved concept page may belong to two categories of special pages in Wikipedia: disambiguation page and redirect page. Such a page needs to be replaced by another page as follows, to acquire the description of the desired concept.

Firstly, a concept name can be ambiguous. For instance, *Feudalism* has different forms in different countries, corresponding to different concepts. A retrieved concept page titled such an ambiguous concept name is usually a *disambiguation page*, as illustrated in Fig. 3. It contains not the detailed description of any concept but links to other pages that describe concepts having the same name. Each link is accompanied by a short description (usually one or several sentences) of the corresponding concept for disambiguation purposes. Therefore, to acquire the detailed description of the desired concept, a retrieved disambiguation page is to be replaced by one of the concept pages it links to, i.e., to disambiguate an ambiguous concept name in $q$. To this end, we leverage other concepts appearing in $q$ as contextual information for disambiguation.

Formally, given a retrieved disambiguation page $dp(cn)$ titled an ambiguous concept name $cn$ appearing in $q$, which contains links to a set of disambiguated concept pages $P(cn)$, we need to replace $dp(cn)$ with one page in $P(cn)$. To find the correct one, let $PT(cn)$ be the set of other page titles

found in $q$ by Algorithm 1. Let $abstr(p)$ and $body(p)$ be the abstract and the body of a candidate concept page $p \in P(cn)$, respectively. Let $sd(p)$ be the short description accompanying the link to $p$ in $dp(cn)$. Let $tf(pt, t)$ be the number of times a page title $pt$ appears in text $t$. The page $p \in P(cn)$ to replace $dp(cn)$ would have the highest ranking score:

$$\Sigma_{pt \in PT(cn)}(\alpha \cdot tf(pt, abstr(p)) + \beta \cdot tf(pt, body(p)) + \gamma \cdot tf(pt, sd(p)))\,, \quad (1)$$

where $\alpha, \beta, \gamma \in [0, 1]$ are weights to be tuned, and are empirically set to $\alpha = 0.8, \beta = 0.5, \gamma = 1.0$ in our experiments.

Secondly, a retrieved concept page can be a *redirect page*. Such a page simply forwards the reader to another page without providing any other information. Therefore, to acquire the description of the desired concept, a retrieved redirect page will be replaced by the page it is redirected to.

For instance, the redirect page *Feoffment* matches the question in Fig. 1 and is retrieved. It is then replaced by the page it is redirected to, namely *Feudalism*, which is a disambiguation page as shown in Fig. 3. Among the three disambiguated concept pages, *Feudalism (in China)* is selected to replace the disambiguation page because its contents and short description in the disambiguation page mention many other concepts appearing in the question, such as *the Zhou dynasty*.

**Retrieving Quote Pages**

Quotes in $q$, like the one in Fig. 1, can give clues about $q$. They widely exist in Chinese, politics, geography, and history tests in Gaokao. However, quotes are often written in Classical Chinese, which is difficult to understand by speakers of modern Chinese and the computer because it uses different lexical items, and appears extremely concise and ambiguous. Apart from a quote itself, the context out of which the quote has been used may also provide useful knowledge for answering $q$. Such contextual information can be acquired from Wikipedia pages whose contents contain an exact match of the quote, called a *quote page*. Specifically, in the stem $s_q$, text in (angle) quotation marks are identified as quotes. A quote page that matches the largest (non-zero) number of these quotes is retrieved, denoted by $p_Q(s_q)$. Each option $o \in O_q$ is processed analogously, resulting in $p_Q(o)$; in particular, if $o$ does not contain (angle) quotation marks, it as a whole will be treated as a quote to retrieve $p_Q(o)$ because in this case (angle) quotation marks may be omitted.

For instance, the contents of the page *King Cheng of Zhou* match the quote in Fig. 1, and it is retrieved as $p_Q(s_q)$.

## 2.2 Ranking and Filtering Pages

Let $P_R(s_q)$, $P_R(o)$,$P_R(O_q)$ be the sets of concept and quote pages retrieved based on $s_q$, $o \in O_q$, and $O_q$, respectively:

$$\begin{aligned} P_R(s_q) &= P_C(s_q) \cup \{p_Q(s_q)\}\,, \\ P_R(o) &= P_C(o) \cup \{p_Q(o)\}\,, \\ P_R(O_q) &= \bigcup_{o \in O_q} P_R(o)\,. \end{aligned} \quad (2)$$

Some of these pages are not truly useful for answering $q$, but contain noise information that may negatively affect the subsequent stage of our approach. We develop three ranking strategies to filter them out: centrality-based, domain-based, and relevance-based.

**Centrality-based Strategy**

Each question $q$ usually focuses on one particular theme. We represent the theme of $q$ by the center of all the $n$ retrieved pages ($n = |P_R(s_q) \cup P_R(O_q)|$) in a vector space, and rank pages according to the cosine similarity between them and the center. Formally, each retrieved page $p_i$ is represented by a vector $F(p_i)$, and is ranked by

$$cosine(F(p_i), \frac{1}{n}\sum_{j=1}^{n}\frac{F(p_j)}{\|F(p_j)\|})\,. \quad (3)$$

In the following, we discuss three ways of defining $F(p_i)$: $F^W(p_i)$, $F^L(p_i)$, and $F^C(p_i)$. Their effectiveness will be compared in the experiments.

Firstly, we consider the words that occur in the contents of $p_i$. Each dimension of the vector corresponds to a separate word $w_j$, and its value is the term frequency-inverse document frequency (TF-IDF) weight of $w_j$:

$$F_j^W(p_i) = tf(w_j, p_i) * (1 + \log\frac{N}{1 + df_W(w_j)})\,, \quad (4)$$

where $tf(w_j, p_i)$ is the number of times $w_j$ occurs in the contents of $p_i$, $df_W(w_j)$ is the number of Wikipedia pages where $w_j$ occurs, and $N$ is the total number of Wikipedia pages.

Secondly, we consider the links to other pages in $p_i$. Each dimension of the vector corresponds to a separate link $l_j$, and its value is computed in a way similar to TF-IDF in Eq. (4):

$$F_j^L(p_i) = lf(l_j, p_i) * (1 + \log\frac{N}{1 + df_L(l_j)})\,, \quad (5)$$

where $lf(l_j, p_i)$ is the number of times $l_j$ occurs in $p_i$, $df_L(l_j)$ is the number of Wikipedia pages where $l_j$ occurs, and $N$ is the total number of Wikipedia pages.

Thirdly, we consider the categories that $p_i$ belongs to, enabled by Wikipedia's category system. Each dimension of the vector corresponds to a separate category $c_j$, and its value is computed in a way inspired by the IDF part in Eq. (4):

$$F_j^C(p_i) = 1 + \log\frac{N}{1 + df_C(c_j)}\,, \quad (6)$$

where $df_C(c_j)$ is the number of Wikipedia pages belonging to $c_j$ or any of $c_j$'s descendant categories, and $N$ is the total number of Wikipedia pages.

**Domain-based Strategy**

When $q$ is known to be in a specific domain like history, Wikipedia's category system can be used to filter out the retrieved pages not belonging to any historical categories. Historical categories consist of all the categories whose names contain the word *history*, and their descendant categories.

**Relevance-based Strategy**

A retrieved page that is useful for answering $q$ should be relevant to both the stem $s_q$ and the options $O_q$. To measure relevance, we represent $s_q$, $O_q$, and each retrieved page $p_i$ by

word vectors $F^W$ weighted by Eq. (4). Depending on the part of $q$ based on which $p_i$ is retrieved, $p_i$ is ranked by

$$\begin{aligned} &\delta_s \cdot cosine(F^W(p_i), F^W(O_q)) \\ &+ \delta_o \cdot cosine(F^W(p_i), F^W(s_q)) , \end{aligned} \tag{7}$$

in which

$$\delta_s = \begin{cases} 1 & \text{if } p_i \in P_R(s_q), \\ 0 & \text{otherwise,} \end{cases} \quad \delta_o = \begin{cases} 1 & \text{if } p_i \in P_R(O_q), \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

In the experiments, we will test the effectiveness of the three strategies and their combination, and explore an appropriate number of top-ranked pages to keep.

### 2.3 Assessing Options

Let $P_F(s_q) \subseteq P_R(s_q)$ and $P_F(o) \subseteq P_R(o)$ be the subsets of pages that remain after filtering by using a combination of the aforementioned strategies. Based on that, the truth of an option $o \in O_q$ is assessed according to the extent to which $q$ together with the contents of $P_F(s_q)$ and $P_F(o)$ can entail that $o$ is the correct answer to $q$. The option in $O_q$ that is most likely to be true will be selected as the answer to $q$.

We decompose the entailment into two parts:

- the extent to which a combination of $s_q$ and $o$ can be entailed from $s_q$ and $P_F(s_q)$, and
- the extent to which a combination of $s_q$ and $o$ can be entailed from $o$ and $P_F(o)$.

By assuming independence of factors within each side of the entailment and then canceling common factors on both sides, it suffices to measure

- the extent to which $o$ can be entailed from $P_F(s_q)$, and
- the extent to which $s_q$ can be entailed from $P_F(o)$.

We transform the above entailment into relevance measurement. We represent $s_q$, each option $o \in O_q$, and each retrieved page $p_i$ by word vectors $F^W$ weighted by Eq. (4). We assess the truth of $o$ according to the following scores:

$$\begin{aligned} Score_O(o) &= cosine(F^W(o), \sum_{p_i \in P_F(s_q)} F^W(p_i)) , \\ Score_S(o) &= cosine(F^W(s_q), \sum_{p_i \in P_F(o)} F^W(p_i)) . \end{aligned} \tag{9}$$

If $P_F(s_q) = \emptyset$ (or $P_F(o) = \emptyset$), the second item in $cosine$ has to be replaced by $F^W(s_q)$ (or $F^W(o)$).

In the experiments, we will test the effectiveness of $Score_O$, $Score_S$, and their combination.

## 3 Experiments

### 3.1 Dataset

We evaluated our approach based on real-life questions appearing in recent history tests in Beijing. Specifically, 577 multiple-choice questions (each with four options, including a known correct answer) were collected: 101 from 10 history tests in Gaokao from 2005 to 2014, and 476 from 50 history tests in mock Gaokao from 2012 to 2015. Three human experts were invited to collectively answer those questions only based on the contents of Chinese Wikipedia pages (dump on 20141105). As a result,

- 123 questions (21.32%), denoted by QS-A, could be successfully answered, i.e., the experts could leverage the contents of some Wikipedia pages to entail the correct answer to each of those questions;
- 454 questions (78.68%), denoted by QS-B, went beyond the scope of Wikipedia, and could not be correctly answered without referring to the contents of history textbooks or other resources.

Both QS-A and QS-B were used to evaluate our approach to answering questions. QS-A was also used to separately evaluate the first stage (i.e., retrieving pages) and the second stage (i.e., ranking and filtering pages) of the approach. To this end, for each question in QS-A, the experts were asked to provide a minimal set of Wikipedia pages they used to entail the correct answer, as a gold standard for evaluation. They provided a total of 345 pages, or 2.80 per question. On the one hand, for 46 questions (37.40%), human experts correctly answered each of them based on the contents of only 1 page. On the other hand, for 4 questions (6.50%), they required 8 pages for answering each of them.

We have made our dataset online accessible to the research community[3].

### 3.2 Metrics

To evaluate the first and the second stage of our approach based on QS-A, let $G$ be the set of gold-standard pages used by the experts to entail the correct answer to a question, and let $A$ be the set of pages found by an automatic approach. We measure the precision ($P$), recall ($R$), and F-score ($F$) of $A$:

$$P = \frac{|G \cap A|}{|A|} \quad R = \frac{|G \cap A|}{|G|} \quad F = 2 \cdot \frac{P \cdot R}{P + R} . \tag{10}$$

To evaluate our full approach based on QS-A and QS-B, we calculate the accuracy of question answering, namely the percentage of correctly answered questions.

### 3.3 Results

**Retrieving Pages**

In Algorithm 1, we used a stopword list consisting of the widely used HIT's list, five domain-specific stopwords, and all the single characters. The algorithm retrieved 2,383 concept pages for the questions in QS-A, or 19.37 per question. We randomly selected and manually checked 200 of them; only 13 matches (6.50%) were incorrect due to inaccurate boundaries, demonstrating the effectiveness of the algorithm.

Among the retrieved concept pages, 151 (6.34%) were disambiguation pages, 91 of which were replaced by correct disambiguated concept pages. Among incorrect replacement, 23 were due to not-on-the-list; that is, Wikipedia provided a short description of the correct concept in the disambiguation page without creating a separate page for it or linking to that page (as illustrated by *Feudalism (in Japan)* in Fig. 3), so that correct replacement was impossible.

Finally, as shown in Table 1, compared with the gold standard for the questions in QS-A, the retrieved concept pages achieved an average recall of 0.807, showing satisfactory

[3] http://ws.nju.edu.cn/gaokao/ijcai-16/GaokaoHistory577.xml

Table 1: Quality of Unfiltered Concept and Quote Pages

| | Precision | Recall | F-score |
|---|---|---|---|
| Concept pages | 0.184 | 0.807 | 0.278 |
| Quote pages | 0.256 | 0.229 | 0.228 |
| Their union | 0.176 | 0.832 | 0.272 |

Table 2: Accuracy of Question Answering

| | QS-A | QS-B |
|---|---|---|
| $Score_O$ | 31.71% | 22.91% |
| $Score_S$ | 36.59% | 30.84% |
| $Score_O + Score_S$ | 43.09% | 31.28% |
| baseline (random) | 25.00% | 25.00% |

completeness of the results, though the precision was very low, indicating the necessity of ranking and filtering pages in the second stage of our approach.

The total number of quote pages retrieved for the questions in QS-A was 244, or 1.98 per question. As shown in Table 1, they formed an effective complement to concept pages; their union achieved a higher recall of 0.832.

**Ranking and Filtering Pages**

We tested various combinations of the proposed strategies for ranking and filtering the retrieved concept and quote pages. Figure 4 shows their precision-recall curves achieved by the resulting $k$ top-ranked pages, with $k$ varying from 1 to 40. Among the three centrality-based strategies based on different kinds of vectors (i.e., word, link, and category), word vector outperformed the others. Its effectiveness was also demonstrated by the considerable difference between the combination of centrality-based, domain-based, and relevance-based strategies and the combination of only domain-based and relevance-based strategies. The former, firstly filtering out pages not belonging to any historical categories and then ranking the remaining pages by the product of centrality-based (using word vector) and relevance-based ranking scores, was also the best-performing one among all the tested combinations. In particular, it exceeded a baseline approach that treated the text of a question as a keyword query and ranked pages by using BM25 [Robertson *et al.*, 1994], a function extensively used in information retrieval.

Similar results were observed on F-scores shown in Fig. 5, which peaked at 0.406, achieved by the aforementioned best-performing combination when $k = 6$; that was noticeably higher than the F-scores achieved by unfiltered pages shown in Table 1 (0.228–0.278), demonstrating the effectiveness of our ranking strategies.

**Assessing Options and Answering Questions**

For each question in QS-A and QS-B, based on top-ranked pages ($k = 6$) given by the best-performing ranking strategy in the previous experiment, the truth of each option was assessed according to Eq. (9); the option achieving the highest score was selected as the answer. Table 2 shows the accuracy of the answers. In QS-A, in which the correct answer to each question could be entailed by the experts from some Wikipedia pages, around one third of the questions were correctly answered solely based on $Score_O$ or $Score_S$, outperforming a natural baseline approach which randomly picked an option from four as the answer and was expected to correctly answer one fourth of the questions. Further, by combining $Score_O$ and $Score_S$, the accuracy increased to 43.09%, which was very encouraging as a preliminary attempt to take up the Gaokao challenge.

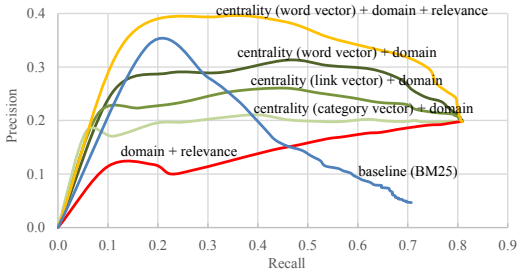


Figure 4: Precision-recall curves of ranking strategies.

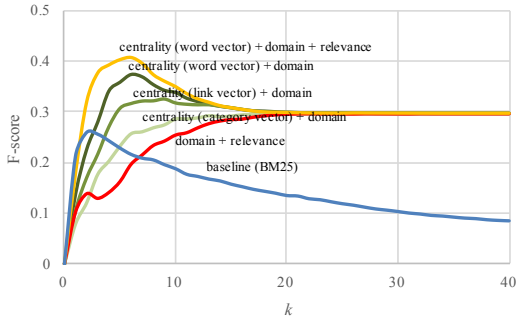


Figure 5: F-scores of ranking strategies.

In QS-B, although those questions more or less went beyond the scope of Wikipedia, our approach still correctly answered 31.28% of them by exploiting knowledge available in Wikipedia, notably outperforming the baseline approach.

## 3.4 Discussion

We analyzed the questions in QS-A on which our ranking strategies achieved low F-scores, and found that some pages covering a broad spectrum of historical topics (e.g., *China*) were ranked high because (a) their contents overlapped with those of many other retrieved pages and thus they were favored by our centrality-based strategy, (b) they belonged to historical categories and were favored by our domain-based strategy, and (c) they contained many words appearing in the question and were favored by our relevance-based strategy. However, the general knowledge provided by such pages was often not truly useful for answering a specialized question in Gaokao. It would inspire future work to not consider the contents of a page as a whole in ranking but look for its small parts that could closely match a question. In other words, one could retrieve and rank not pages but sentences or paragraphs.

In the experiments, our approach was configured to consistently use $k$ top-ranked pages for answering every question, whereas according to the gold standard, the number of pages needed for entailing the correct answer to a question

varied from 1 to 8. Therefore, it would motivate future work to study how to automatically configure the approach to use an appropriate $k$ for each individual question in practice, to avoid insufficiency of knowledge given too few pages or noise information brought by too many pages.

Even though our approach retrieved the right pages for a question, it could still fail to entail the correct answer due to limitations of our entailment method based on relevance measurement. Some questions required to be more profoundly understood, e.g., expressing the negation of a fact and thus reversing the truth of an option. Some other questions asked about the chronological order of a set of historical events, for which relevance measurement was not suitable. The diversity of questions in Gaokao would call for multiple solvers, and our approach would be just one tool in the toolbox.

We compared our approach with two baseline approaches: a natural baseline approach to answering multiple-choice questions by randomly picking an option as the answer, and BM25 which was extensively used for ranking pages in information retrieval. Existing solutions to semantic parsing and question answering [Kolomiyets and Moens, 2011; Kwiatkowski *et al.*, 2013; Berant and Liang, 2014; Yih *et al.*, 2015] were not involved because there was a mismatch between the type of questions they could handle (i.e., one-sentence-long factoid questions) and the multiple-choice questions in Gaokao (i.e., several-sentence-long complex questions). Besides, recent efforts to semantic parsing were assumed to answer a question against a formal knowledge base, whereas to the best of our knowledge, Chinese knowledge bases that were freely accessible could not offer satisfactory coverage of the knowledge needed for Gaokao.

## 4 Related Work

Having the computer pass entrance examinations at different levels of education, as an increasingly hot AI challenge, has been taken up by researchers in several countries, dealing with various subjects. As an early attempt, the Project Halo [Friedland *et al.*, 2004] aims to create a Digital Aristotle that encompasses scientific knowledge and is capable of solving complex problems. As part of this effort, the OntoNova system [Angele *et al.*, 2003] represents chemistry knowledge in F-Logic and can answer formal queries regarding complex chemical concepts and reactions via rule reasoning. Later, physics and biology knowledge is covered, and less formal questions described in a controlled natural language are supported [Gunning *et al.*, 2010]. Another line of research addresses mathematical questions described in natural language. Early attempts mainly employ hand-crafted rules to transform natural language into formal queries that can be processed by an inference engine [Mukherjee and Garain, 2008]; recent efforts start to explore learning techniques [Kushman *et al.*, 2014; Hosseini *et al.*, 2014].

The above formal methods may not be suitable for subjects like history, which largely rely on textual knowledge. In the Todai Robot Project [Fujita *et al.*, 2014], a yes-no question in history tests is converted to a set of factoid questions; their answers and confidence values returned by an existing question answering system are aggregated to determine the correctness of the original proposition [Kanayama *et al.*, 2012]. A multiple-choice question in history tests is transformed into recognizing textual entailment between a description in Wikipedia and each option of the question [Miyao *et al.*, 2012]; the general idea is similar to our work. However, it is a mere partial solution to the problem since it is not clear which description or page should be used for entailment. By comparison, what we present in this paper is a complete solution; in particular, retrieving, ranking, and filtering concept and quote pages is a major technical contribution of our work.

It is essential to distinguish between questions in entrance examinations like Gaokao and one-sentence-long factoid questions handled by existing solutions to semantic parsing and question answering [Kolomiyets and Moens, 2011; Kwiatkowski *et al.*, 2013; Berant and Liang, 2014; Yih *et al.*, 2015]. A question in Gaokao is more complex, comprising multiple sentences and requiring more sophisticated discourse analysis such as coreference resolution; existing techniques are not ready to truly understand such a question. In addition, such a question often includes domain-specific expressions like quotes in Classical Chinese, which are even more difficult to understand by the computer. For these reasons, existing approaches can hardly apply to Gaokao directly, though their ideas may be borrowed. To practically meet the challenge, our three-stage approach exploits and extends information retrieval techniques, and has achieved encouraging results in the experiments.

## 5 Conclusion

We are among the first to take up the Gaokao challenge. The challenge itself is related to many applications for computer-aided education. In addition, it requires solving AI problems like natural language understanding and (complex) question answering, thereby contributing to next-generation AI applications. To meet the challenge, our approach focuses on but is not restricted to multiple-choice questions; answering other types of questions like fill-in-the-blank can build upon our technique for retrieving, ranking, and filtering pages. In future work we will explore that in history tests as well as similar subjects like human geography, based on not only Wikipedia but also other resources like textbooks and pages from domain-specific websites.

Our approach can be improved in several directions. As discussed in Sect. 3.4, we will consider retrieval and ranking at sentence or paragraph level. We will also consider the use of textual entailment methods [Androutsopoulos and Malakasiotis, 2010] for assessing the truth of an option, but have to overcome the difficulty that a complex option may require synthesizing information from multiple pages for entailment. As a longer-term goal, to truly understand and correctly answer a question in Gaokao, we need to integrate discourse analysis, semantic parsing, and formal representation of domain knowledge.

## Acknowledgments

This work was supported in part by the 863 Program under Grant 2015AA015406, in part by the NSFC under Grant

61572247 and 61223003, and in part by the Fundamental Research Funds for the Central Universities.

## References

[Androutsopoulos and Malakasiotis, 2010] Ion Androutsopoulos and Prodromos Malakasiotis. A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research*, 38:135–187, May–August 2010.

[Angele *et al.*, 2003] Jürgen Angele, Eddie Moench, Henrik Oppermann, Steffen Staab, and D. Wenke. Ontology-based query and answering in chemistry: OntoNova @ project Halo. In *Proceedings of the 2nd International Semantic Web Conference*, pages 913–928, Sanibel Island, Florida, October 2003. Springer.

[Berant and Liang, 2014] Jonathan Berant and Percy Liang. Semantic parsing via paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1415–1425, Baltimore, Maryland, June 2014. ACL.

[Clark, 2015] Peter Clark. Elementary school science and math tests as a driver for AI: Take the Aristo challenge! In *Proceedings of the 27th Conference on Innovative Applications of Artificial Intelligence*, pages 4019–4021, Austin, Texas, January 2015. AAAI.

[Friedland *et al.*, 2004] Noah S. Friedland, Paul G. Allen, Gavin Matthews, Michael Witbrock, David Baxter, Jon Curtis, Blake Shepard, Pierluigi Miraglia, Jürgen Angele, Steffen Staab, Eddie Moench, Henrik Oppermann, Dirk Wenke, David Israel, Vinay Chaudhri, Bruce Porter, Ken Barker, James Fan, Shaw Yi Chaw, Peter Yeh, Dan Tecuci, and Peter Clark. Project Halo: Towards a digital Aristotle. *AI Magazine*, 25(4):29–48, Winter 2004.

[Fujita *et al.*, 2014] Akira Fujita, Akihiro Kameda, Ai Kawazoe, and Yusuke Miyao. Overview of Todai robot project and evaluation framework of its NLP-based problem solving. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, pages 2590–2597, Reykjavik, Iceland, May 2014. ELRA.

[Gunning *et al.*, 2010] David Gunning, Vinay K. Chaudhri, Peter Clark, Ken Barker, Shaw-Yi Chaw, Mark Greaves, Benjamin Grosof, Alice Leung, David McDonald, Sunil Mishra, John Pacheco, Bruce Porter, Aaron Spaulding, Dan Tecuci, and Jing Tien. Project Halo update - progress toward digital Aristotle. *AI Magazine*, 31(3):33–58, Fall 2010.

[Hosseini *et al.*, 2014] Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren Etzioni, and Nate Kushman. Learning to solve arithmetic word problems with verb categorization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 523–533, Doha, Qatar, October 2014. ACL.

[Kanayama *et al.*, 2012] Hiroshi Kanayama, Yusuke Miyao, and John Prager. Answering yes/no questions via question inversion. In *Proceedings of the 24th International Conference on Computational Linguistics*, pages 1377–1392, Mumbai, India, December 2012. ACL.

[Kolomiyets and Moens, 2011] Oleksandr Kolomiyets and Marie-Francine Moens. A survey on question answering technology from an information retrieval perspective. *Information Sciences*, 181(24):5412–5434, December 2011.

[Kushman *et al.*, 2014] Nate Kushman, Yoav Artzi, Luke Zettlemoyer, and Regina Barzilay. Learning to automatically solve algebra word problems. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 271–281, Baltimore, Maryland, June 2014. ACL.

[Kwiatkowski *et al.*, 2013] Tom Kwiatkowski, Eunsol Choi, Yoav Artzi, and Luke Zettlemoyer. Scaling semantic parsers with on-the-fly ontology matching. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1545–1556, Seattle, Washington, October 2013. ACL.

[Miyao *et al.*, 2012] Yusuke Miyao, Hideki Shima, Hiroshi Kanayama, and Teruko Mitamura. Evaluating textual entailment recognition for university entrance examinations. *ACM Transactions on Asian Language Information Processing*, 11(4):13, December 2012.

[Mukherjee and Garain, 2008] Anirban Mukherjee and Utpal Garain. A review of methods for automatic understanding of natural language mathematical problems. *Artificial Intelligence Review*, 29(2):93–122, Aprial 2008.

[Peñas *et al.*, 2013] Anselmo Peñas, Eduard Hovy, Pamela Forner, Álvaro Rodrigo, Richard Sutcliffe, and Roser Morante. QA4MRE 2011-2013: Overview of question answering for machine reading evaluation. In *Proceedings of the 4th International Conference of the CLEF Initiative*, pages 303–320, Valencia, Spain, September 2013. Springer.

[Richardson *et al.*, 2013] Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. MCTest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, Seattle, Washington, October 2013. ACL.

[Robertson *et al.*, 1994] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. Okapi at TREC-3. In *Proceedings of the 3rd Text REtrieval Conference*, pages 109–126, Gaithersburg, Maryland, November 1994. NIST.

[Yih *et al.*, 2015] Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pages 1321–1331, Beijing, China, July 2015. ACL.