CSGS: Adapting a short answer scoring system for multiple-choice reading comprehension exercises

Simon Ostermann, Nikolina Koleva, Alexis Palmer, and Andrea Horbach

Department of Computational Linguistics, Saarland University, Saarbrücken, Germany
(simono,nikkol,apalmer,andrea)@coli.uni-saarland.de
home page: http://www.coli.uni-saarland.de

Abstract. This paper describes our system submission to the CLEF Question Answering Track 2014 Entrance Exam shared task competition, where the task is to correctly answer multiple choice reading comprehension exercises. Our system is a straightforward adaptation of a model originally designed for scoring short answers given by language learners to reading comprehension questions. Our model implements a two step procedure, where both steps use the same set of metrics for evaluating similarity between pairs of input sentences/questions. In the first step, we automatically select the sentence of the reading text that best matches the question. In the second step, the selected sentence is compared to each of the four answers, and the answer with the highest similarity score is chosen as the correct answer. Although the model has not been tuned to this specific task, we obtain scores that are competitive with other top-performing systems in the challenge. Additionally, we make no use of the training material but rather treat the task as one of general determination of semantic similarity between text sentences and provided answers.

1 Introduction

Reading comprehension exercises are a widely-used and important means of assessing students' ability to understand the material they read. Questions in reading comprehension exercises generally span a range of difficulty levels, from simple extraction of facts contained in reading texts to more sophisticated inference, requiring both information in the text and general (or subject-specific) background knowledge. As such, they provide a challenging context for automated analysis.

The CLEF Question Answering Track 2014 Entrance Exam shared task asks systems to read a given document and answer a set of multiple-choice questions based on the reading text. We approach this task by adopting a model originally designed for a different reading comprehension context: scoring short answers to reading comprehension questions given by language learners. The tasks have in

common that they require assessing the suitability of answers to questions based on reading texts, but they differ in their inputs and expected outputs.

For short answer scoring, the system is provided with a reading text, a set of questions, a target answer for each question, and a set of learner answers to be scored as correct or incorrect. Most short answer scoring systems work by comparing learner answers to a sample solution (aka target answer), but language learners tend to replicate chunks of the reading text in their answers. Thus, it is often straightforward to match a sentence of the text to an answer written by a learner. In previous work we used this tendency to develop a scoring model that incorporated features based on the relationship between reading text sentences and learner answers [3].

In the Entrance Exam challenge, the system is again provided with a reading text and a set of questions based on the text. Instead of a target answer, though, there is a set of four answers: one best answer and three distractor answers. Often there is high similarity between the best answer and one or more other answers. To accomplish this task, we again use a model that evaluates similarity between text sentences and both the question and the set of answers. The basic idea of the model (which is described in more detail in Section 3) is to use a common set of similarity metrics (following [4]) in a two-step procedure. First, we automatically identify the sentence of the reading text that best matches the question, on the assumption that this sentence has a reasonable likelihood of containing the question's answer. We will see (in Section 5) that this assumption does not always hold. Second, we choose as the best answer the one our system evaluates as most similar to the selected sentence from the text.

It should be noted that this approach requires no training material, as it simply relies on evaluating similarity between either the question and a sentence from the reading text or a reading text sentence and each answer from the set of four in the multiple-choice setting. Although our model has not been tuned to the specific task, it still performs at a level that is comparable to other top-performing systems in the challenge. This suggests that the general approach captures key aspects of semantic similarity.

2 Task and Data

The aim of the given task is to provide a computational solution towards automatic question answering. The data consist of reading comprehension exercises which were part of Japanese university entrance exams. Those exams are meant to be used for checking new student's capabilities of various skills by testing them with the help of reading exercises and are collected from the Japanese Center Tests of 2013 and 2014.[1]

The task now is to detect the right answer for a given question and reading text. Contrary to earlier shared tasks from this scenario, questions are rather unconstrained here and range from "simple" comprehension questions, that require the student or computer just to find a paraphrase, over sentence completion

[1] http://nlp.uned.es/entrance-exams/

questions up to complex questions that demand a deeper knowledge of text coherence.

The data consists of 12 test documents with 56 questions and 4 answers for each question. The texts vary in terms of length, complexity and content. Task organizers provided two data sets: one for training and development, with the correct answer for every question indicated, and a second for testing and ranking systems participating in the challenge. Though we used training material to test the general feasibility of using our pre-existing model for this new task, we do not use correct-answer annotations for any actual model training or even parameter setting.

3 Our Model

As described above, our system was originally developed for the related task of short answer scoring. In short answer scoring, a number of different criteria, ranging from token overlap to various syntactic and semantic features, are used to determine whether a given student answer is correct or incorrect. A crucial difference between short answer scoring and answering multiple-choice questions is the absence in the latter of an "ideal" target answer. When we use this model for short answer scoring, we compare each student answer to the target answer, and then do supervised classification to learn how this comparison looks for correct vs. incorrect answers. In this section we describe how we adapt the model for scoring answers when we don't have a target answer to compare to.

Figure 1 schematically shows the workflow of the system. In general the system consists of two components. In the first step, answers, texts and questions are preprocessed and annotated with linguistic information. The output of this preprocessing afterwards serves as the input for the alignment module.
Our adapted alignment model for evaluating answers as such consists of two sub-modules again, the **sentence selection** module and the **answer selection** module. Both modules rely on alignment between sentences. For sentence selection, we find the best reading text sentence for a given question via alignment ; for answer selection we align each answer with this best sentence. We first describe the alignment model and then the general workflow of our two-step answer evaluation model.

3.1 Alignment model

In our alignment model we follow the methodology that has been proposed by [4] for grading short answer questions. In such a task, the content of a learner answer is aligned to that of a target answer, and features measuring the overlap between target and learner answer are extracted in order to approximate the determination of semantic equivalence between target answer and learner answer. During alignment, the model identifies similarities between a learner answer and its corresponding target answer on a number of pairs on a number of linguistic levels: tokens, chunks, and dependency triples. For the current task, aligning

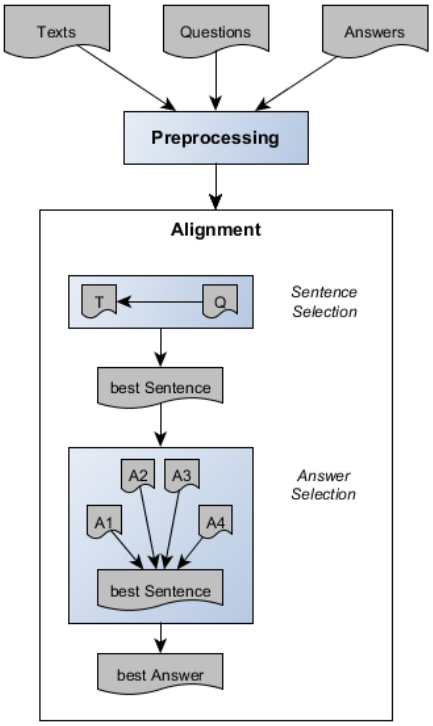

Fig. 1: System architecture (abbreviations: (Q)uestions, (A)nswers (1-4) and (T)exts)

answers and questions to text sentence, we mainly consider alignments between tokens.

We preprocess all material (texts, questions and answers) using standard NLP tools: for sentence splitting (OpenNLP) and tokenization (Stanford CoreNLP),[2][3] POS tagging and stemming (both Treetagger [6]),[4] NP chunking (Treetagger) and synonym extraction (WordNet [1]).[5] For synonyms we use not only words that occur in the same synset but also words that are in a hypernym relation and have maximally one node in between them.

On the token level, we use several different metrics for identity between tokens, with each metric associated with a certain alignment weight. We use the following types of identity (id), weighted in descending order: token id > lemma id > synonym id. After weights have been determined for all possible token pairs,

[2] http://opennlp.apache.org/index.html
[3] http://nlp.stanford.edu/software/corenlp.shtml
[4] http://www.cis.uni-muenchen.de/ schmid/tools/TreeTagger/
[5] http://wordnet.princeton.edu/

the best applicable weight is used as input for a traditional marriage alignment algorithm [2].

The token alignment is afterwards refined by chunk-alignment: For chunk alignment, we use the percentage of aligned tokens between chunks for two sentences as input for the alignment process. Two chunks can only be aligned if at least one of the tokens has been aligned. If an aligned token pair from the previous token alignment step ends up in two not-aligned chunks, the token alignment is split up again.

In short-answer scoring, the resulting alignment is used to extract a variety of overlap features, like e.g. how many tokens of the learner answer can be found in the target answer. In the task at hand, where we have to identify the one sentence out of a set of sentences that fits some other sentence best, we instead use the alignment directly to compute one overall alignment weight by summing up the weights of all token-level links of the final alignment between two sentences.

3.2 Answer Evaluation Workflow

Sentence Selection. In this first step of our evaluation, we aim at finding the text sentence which best matches the lexical material in the question. To do so, we align each sentence in the reading text with the question and compute the overall alignment weight for each pairing. The text sentence with the highest weight is then assumed to be the sentence that carries the most crucial information for answering the question.

Answer Selection. After the best-matching sentence has been identified, we align this sentence with each of the four potential answers. Again the answer with the highest alignment weight is proposed as the correct answer.

Figure 2 schematically illustrates the process of selecting the correct answer. The black arrow indicates the selection of the best sentence by aligning it with the question. The answer that aligns best with this sentence is taken to be correct. Other answers that were not selected might potentially link to other regions of the text (as indicated with red arrows).

One technical problem that may occur is that two answers to the same question can end up with the same alignment weight. In the test data, this happens for 11 out of 56 questions. We investigate two different ways of handling this outcome, submitting them as two runs of our model. In the first run (**csgs-1** in Table 1), we simply choose the first (in linear sequence) of the equally-weighted answers. In the second run (**csgs-2**), we mark such questions as unanswered.

Example (1) shows an example for such a set of answers that all receive the same weight, because none of them has any overlap with the proposed text sentence that could be picked up by our system.

(1) a. **Question:** At the beginning of the story, what did the boy think George did for a living?
Text sentence: “Are you a carpenter, sir” the boy asked, looking

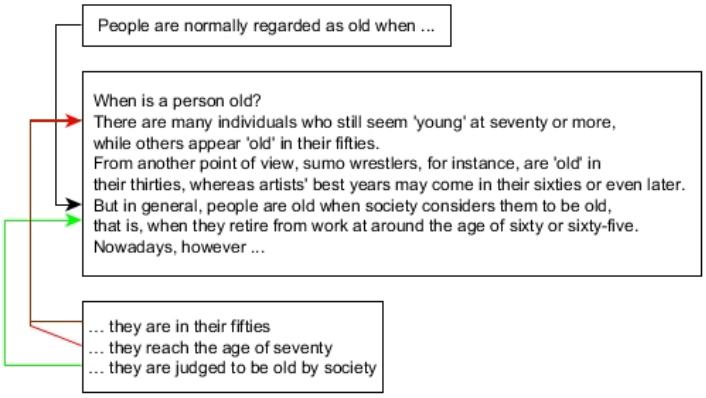

Fig. 2: Visualization of the alignment model

up at the old man's face.

Answers:

–He thought he made and repaired wooden objects.
–He thought he painted pictures or houses.
–He thought he fixed musical instruments.
–He thought he grew plants and flowers.

4 Results

Table 1 summarizes the evaluation of the systems that participated in the English-language portion of the Question Answering Track for Entrance Exams (cut down to roughly the top 50% of runs). In total five systems participated in the shared task, each submitting multiple runs. Each run is given a c@1 score according to [5]. This measure is an extension of the accuracy measure but also accounts for deciding not to answer a question if a system is not confident enough about it. Since it is an extension of the accuracy measure, the higher the score, the better the performance.

Our two runs (**csgs-1** and **csgs-2**, as described above) are highlighted in bold. The c@1 score gives a slight boost for not answering a question rather than submitting an incorrect answer. Therefore it is not surprising that **csgs-2** achieves the better performance of our two runs, since it takes the intuitive and simple strategy of declining to answer when it cannot confidently select one best answer from the set of four possible answers.

5 Discussion and Error Analysis

In this section we provide error analysis to show and discuss the strengths and weaknesses of our model in the context of this task. In order to analyze the

Table 1: List of submitted systems that achieved c@1 => 0.25 on the English data set.

c@1	System Run
0.446	EE2014–SynapseDeveloppement–1–Output english.xml
0.375	EE2014–DIPF–7–dipf1407enen.xml
0.375	EE2014–cicnlp–8–cicnlp-8.xml
0.362	**EE2014–csgs–2–02_1.xml**
0.357	**EE2014–csgs–1–01_1.xml**
0.357	EE2014–cicnlp–7–cicnlp-7.xml
0.339	EE2014–cicnlp–2–cicnlp-2.xml
0.304	EE2014–cicnlp–4–cicnlp-4.xml
0.304	EE2014–cicnlp–3–cicnlp-3.xml
0.286	EE2014–DIPF–5–dipf1405enen.xml
0.286	EE2014–DIPF–3–dipf1403enen.xml
0.286	EE2014–cicnlp–6–cicnlp-6.xml
0.286	EE2014–cicnlp–1–cicnlp-1.xml
0.25	EE2014–LIMSI-CNRS–4–dude4.xml
0.25	EE2014–DIPF–6–dipf1406enen.xml

source of the errors, we annotate the English data set with the best sentence that matches the meaning of the correct answer for each question. In this manner, we establish a gold standard (GS) for the first step of our model, which is the sentence selection described in Section 3.2. We also discuss the performance of the two modules of the system for more detailed evaluation.

5.1 Additional Annotations

Our answer selection mechanism comprises two steps: the selection of a relevant text sentence that potentially contains the correct answer and the alignment of the proposed answer alternatives against this passage.

For a better understanding of these two components, we assess their contributions separately: We first assess how often our sentence selection module finds the correct sentence in the text. In a second step we examined how good the alignment of answers to the text would be given that we had oracle information about this best sentence. In order to be able to conduct these experiments we need some additional gold standard information we marked the text sentence that we thought contained the relevant material necessary to answer the question, i.e. the gold standard for the best text sentence.

These annotations have been done by two annotators each, in case of a disagreement they have been adjudicated by a third annotator. The two annotators agreed in 61% of the cases. This shows that the task for selecting the best sentence is not trivial even for humans. As a matter of fact usually one needs more than one sentence to answer a question. Especially for a general question like "What is this story about?" it is necessary to infer the main point of the story, which would require a deeper semantic analysis that would need to include processing on the discourse level.

Having the GS for the best matching sentence we re-ran the classification of the answers and obtained an accuracy of 50%, which is an improvement of 16% compared to the performance with the automatically selected sentences. Even using GS sentences from the text we are able to detect the correct answer for only half of the questions.

5.2 Error Analysis

In this section we show examples for which our model worked and also didn't work and discuss the reasons for that.

Example (2) shows a case where the correct answer was selected. The third answer highlighted in italic bold has the highest weight and thus it is classified as correct by our system. The best sentence in this case is also the one that we obtained in the gold standard. Therefore the comparison leads us to the correct answer. Here we can see the importance of the synonymy check. Although there is a high overlap of identical words it is particularly important to recognize that the verbs *go away* and *leave* in the answer and the best sentence, respectively, have the same meaning in order to score the answer higher.

(2) a. **Question:**
What was the problem the author had with his house?

b. **Best Sentence:**
My son and I were trying to sell the house we had restored but in the barn attached to it there were bats and they wouldn't leave.

c. **Answers ordered by alignment weight, best fitting first:**

–***Bats were living in the barn and wouldn't go away.***
–The author and his son might not be able to stay for the season.
–The author and his son couldn't sleep well because of the muttering sounds.
–The house was still badly in need of repair.

Another case that is well handled by our model is not a direct question but rather a completion of a sentence like in example (3). The detection of the best matching sentence works well and consequently the answer selection that is based on the comparison to the best text sentence is good.

(3) a. **Question:**
Rats that live with their brothers and sisters during their early days

b. **Best Sentence:**
It has been found that while baby rats kept with their brothers and sisters engage in a lot of rough play, those raised alone with their mothers play just a little.

c. **Answers ordered by alignment weight, best fitting first:**

–***spend a lot of the time playing roughly with them.***
–hurt each other a lot through their rough play.
–quickly learn to be independent of their mothers.
–still want to play with their mothers.

In case our system fails to detect the best sentence, the consequence is that the selection of the correct answer also fails. The reason for that is because we compare the answers to a different piece of the given text. The incorrect answers in the multiple choice are also related to the text but not to the relevant part for the question under consideration. This idea is illustrated with example (4). The best sentence selected by the system gives a higher alignment weight when it is compared to an incorrect answer. In case we know the best sentence according to the gold standard, our system is able to select the correct answer. It is interesting to observe that the weight of the overlap with the correct answer is the same as before but now it is the highest one.

(4) a. **Question:**
How did the author obtain Margaret's address?

b. **System Best Sentence:**
"I don't know how my address got into a magazine in Japan, because I have never asked for a pen pal, but it's so nice hearing from someone in such a fascinating country, and I look forward to corresponding with you."

c. **GS Best Sentence:**
I was reading a popular youth magazine when I noticed a list of addresses of young people from all over the world who were seeking pen pals in Japan.

d. **Answers:**

–He wrote to a popular magazine for her address.
(**best for system sentence**)
–He found it in a popular magazine.
(**best for gold standard sentence**)
–He received it from one of his classmates.
–He selected it from a list given by his teacher.

5.3 Performance of the Sentence Detection Unit

We compared how many of the sentences in the GS are also selected by the sentence detection component of our system. It turned out that only 11 of 56 sentences matched the GS sentences. In other words the accuracy on the sentence

selection task is 24%. It is particularly challenging for an automatic method to distinguish in case there is a high overlap with the lexical material in the question, whether the sentence contains the information relevant for answering the question or not. In example (6) approximately half of the lexical material that occurs in the question overlaps with the best automatically selected sentence and thus it matches the sentence in the GS.

(5) a. **Question:**
Why was the writer in Arizona during World War II?

b. **System Best Sentence:**
I'd been sent to a special camp in Arizona for Japanese-Americans during World War II, before I joined the army.

c. **GS Best Sentence:**
I'd been sent to a special camp in Arizona for Japanese-Americans during World War II, before I joined the army.

In contrast, in example (6) the system picks a different system because the overlap of the lexical material is higher than with the GS sentence corresponding to this question. In this example we see the subject of the sentences has the surface representation "I" and it refers to "the author". Our model does not apply co-reference resolution but this may improve the performance of the system. If a model is able to figure out that the pronoun "I" in the GS Best sentence is referring to the author and to measure the overlap with the noun phrase then the GS Best Sentence would have a better chance to be selected. One other factor that influences the choice of a wrong sentence is that in general our model prefers shorter sentences as it computes the percentage overlap.

(6) a. **Question:**
Why did the author ask Margaret for her picture?

b. **System Best Sentence:**
Margaret had asked her friend to send it only in the case of her death.

c. **GS Best Sentence:**
I knew it would be impolite to ask a girl her age, but thought it would be all right to ask her to send a picture.

6 Conclusions and Future Work

In this paper we have described the adaption of a model originally developed for short answer scoring to the task of answering multiple choice questions in an entrance exam scenario. The model has been shown to be suitable for both

tasks, since both require evaluating a set of responses according to how well they answer reading comprehension questions. With no task-specific tuning, and without using the training material provided, the system achieves performance comparable to the best-performing runs submitted to the CLEF Question Answering Track 2014 Entrance Exam shared task.

That said, there is ample room for improving the system in order to better handle the task at hand. In our two-step approach, aspects of both modules could be improved. Error analysis shows that the first step of the procedure, sentence selection, performs quite poorly compared to gold standard annotations. High overlap with the question material is simply not enough to choose the sentence from the text that contains the highest proportion of the answer material. For this task we need deeper semantic analysis. In particular, a first step would be to incorporate a co-reference system; this would be beneficial for the many sentences in which pronouns occur instead of full noun phrases. We could further improve this module by taking into account the type of the question and the corresponding expected answer type.

One particular weakness of our approach is that the system, by maximizing percentage overlap, tends to prefer shorter sentences to longer ones. One potential way to address this problem would be to use some metric for identifying the most important words in the reading text and then give these terms more weight when determining the overall alignment score. We would also like to expand our approach to lexical similarity to better identify words taht are not covered by Wordnet synset relations.

References

1. Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.
2. D. Gale and L. S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.
3. Andrea Horbach, Alexis Palmer, and Manfred Pinkal. Using the text to evaluate short answers for reading comprehension exercises. In *Second Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 286–295, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics.
4. Detmar Meurers, Ramon Ziai, Niels Ott, and Stacey Bailey. Integrating parallel analysis modules to evaluate the meaning of answers to reading comprehension questions. *Special Issue on Free-text Automatic Evaluation. International Journal of Continuing Engineering Education and Life-Long Learning (IJCEELL)*, 21(4):355–369, 2011.
5. Anselmo Peñas and Alvaro Rodrigo. A simple measure to assess non-response. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1415–1424, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.
6. Helmut Schmid. Improvements in part-of-speech tagging with an application to German. In *In Proceedings of the ACL SIGDAT-Workshop*, pages 47–50, 1995.