CoMiC: Exploring Text Segmentation and Similarity in the English Entrance Exams Task

Ramon Ziai and Björn Rudzewitz

Universität Tübingen, SFB 833,
Nauklerstr. 35, 72070 Tübingen, Germany
`{rziai,brzdwtz}@sfs.uni-tuebingen.de`
`http://purl.org/icall/comic`

Abstract. This paper describes our contribution to the English Entrance Exams task of CLEF 2015, which requires participating systems to automatically solve multiple choice reading comprehension tasks. We use a combination of text segmentation and different similarity measures with the aim of exploiting two observed aspects of tests: 1) the often linear relationship between reading text and test questions and 2) the differences in linguistic encoding of content in distractor answers vs. the correct answer.
Using features based on these characteristics, we train a ranking SVM in order to learn answer preferences. In the official 2015 competition we achieve a c@1 score of 0.29, a medium but encouraging result. We identify two main issues that pave the way towards further research.

Keywords: Short Answer Assessment, Question Answering, Reading Comprehension

1 Introduction

Computational approaches to comparing and evaluating meaning of language have been the focus of much research in the last years, as demonstrated by shared tasks such as Recognizing Textual Entailment [6] and various SemEval tasks, e.g. Semantic Textual Similarity [1]. While earlier work concentrated on comparing the meaning of sentences and text fragments in isolation, there has been a trend towards contextualizing such tasks in a concrete application scenario, such as evaluating learner answers to comprehension questions, as in the 2013 Joint Student Response Analysis and Recognizing Textual Entailment task [7]. Our core research focus lies in the same area, and we have participated competitively in the latter task [13].

In this paper, we describe our contribution to the 2015 English Entrance Exams task, a relatively new development in the area. It is similar to our core task of Short Answer Evaluation in that both require evaluation of answers to questions about a text, but differs in that it is a multiple choice task and hence does not include reference answers to which student input can directly be compared. The objective of the Entrance Exams task is to build a system which assumes the role of a test-taker in multiple choice reading comprehension exams. As a subtask of the CLEF Question Answering track, the task involves both pinpointing the location of relevant text passages and recognizing the meaning equivalence or difference of answer candidates, making it an especially challenging undertaking.

However, the task provides an interesting testbed for various research questions concerning reading comprehension. Among these, one is of particular interest to us: Given a question and a text, how can one accurately and robustly pinpoint to the relevant part in the text? In participating in the Entrance Exams task, we hope to cover some way towards answering that question.

The structure of the paper is as follows: section 2 gives an overview of the data used, and section 3 explains our approach to the task. Section 4 then discusses results and problems before section 6 concludes the paper.

2 Data

The data as provided by the task organizers is comprised of English reading tests posed as an entrance requirement at Japanese universities. Every reading test consists of a text, three to seven questions, and four answer candidates to every question. Out of the answer candidates, exactly one is correct, which sets the random baseline to 25%. The texts are mostly fictional in nature, which is likely intentional because world knowledge would interfere with language competence in factual texts: for example, a specialized text on spiders would be easier for students with prior knowledge of zoology than for those without, regardless of their language skills.

Figure 1 shows an example of a test with one question and its four answer candidates. The correct answer to *People are normally regarded as old when* according to the text is *c) they are judged to be old by society.* The relevant information can be found in the fourth sentence of the text: *But in general, people are old when society considers them to be old, that is, when they retire from work at around the age of sixty or sixty-five.* In this case, the test-taker needs to recognize that *consider* and *judge* are synonyms according to the context.

The task organizers provided the data from the 2013 and 2014 editions of the Entrance Exams task as training data, which are both approximately equal in size and do not overlap in language material. The 2015 data set is larger, with 19 reading tests as compared to the 12 in the previous editions. Table 1 gives an overview of the three data sets.

Data set	# tests	# questions	Ø questions/text	Ø words/text
2013	12	60	5.0	624.3
2014	12	56	4.7	520.0
2015	19	89	4.7	481.6

Table 1. Statistics for the 2013, 2014 and 2015 data sets

Text:	When is a person old? There are many individuals who still seem 'young' at seventy or more, while others appear 'old' in their fifties. From another point of view, sumo wrestlers, for instance, are 'old' in their thirties, whereas artists' best years may come in their sixties or even later. But in general, people are old when society considers them to be old, that is, when they retire from work at around the age of sixty or sixty-five. Nowadays, however, the demand for new work skills is making more and more individuals old before their time. Although older workers tend to be dependable, and have much to offer from their many years of experience, they are put at a disadvantage by rapid developments in technology. Older people usually find it more difficult to acquire the new skills required by technological changes, and they do not enjoy the same educational opportunities as young workers. When they finally leave work and retire, people face further problems. The majority receive little or no assistance in adjusting to their new situation in the community. Moreover, since society at present appears to have no clear picture of what place its older members should occupy, it is unable to offer them enough opportunities to have satisfying social roles after they retire. In the past, the old used to be looked upon as experts in solving various problems of life. Today, however, they are no longer regarded as such and are seldom expected to play significant roles in social, economic and community affairs. With the number of older people in the population rapidly increasing, we need greatly to increase and improve the opportunities provided for them so that they can participate in society with dignity and respect.
Question:	People are normally regarded as old when
Answer Candidates:	a) they are in their fifties b) they are judged to be old by to society *(correct)* c) they consider themselves too old to work d) they reach the age of seventy

Fig. 1. Example reading test with text, question and answer candidates

3 Our Approach

As mentioned in the introduction, the Entrance Exams task can be seen as a two-step problem: one needs to 1) identify the part of the text that a question is about and 2) identify the answer whose meaning is expressed in that text part. While 1) is mainly needed to narrow down the search space, 2) is where the test-taker needs to demonstrate their grasp of the content.

We will first give a short overview of our approach before going into more detail on how each step was handled. Part 1), which we will call *Text Segment Identification* for the purposes of this paper, was accomplished by i) using a text segmentation algorithm to partition the text into meaningful paragraphs and ii) comparing the question to each paragraph using a similarity metric. The result is an ordering of paragraphs by similarity to the question. Part 2), which will be called *Answer Selection* was tackled by a) extracting different similarity features of each answer candidate to each paragraph in the order determined before, and b) using these features to train a ranking SVM model for pairwise answer candidate comparison.

3.1 Pre-processing and Architecture

As a prerequisite to the later steps, a certain amount of pre-processing needs to be done. The whole architecture of our system was realized using the UIMA framework [8], with DKPro Core [3] for the pre-processing components and DKPro Similarity [2] for the similarity metrics. We experimented with different options, but the final set of NLP components was the one shown in Table 2. Most of the pre-processing is needed in order to perform coreference resolution, which we use to resolve each coreferent expression to its first mention before we apply any similarity metrics.

Task	Component
Sentence Segmentation, Tokenization, Named Entity Recognition, Lemmatization, POS tagging, Coreference Resolution	Stanford CoreNLP [11]
Constituency Parsing	Berkeley Parser [15]
Dependency Parsing	MaltParser [12]
Semantic Role Labeling	Clear SRL [5]

Table 2. NLP components used in our system via DKPro Core

3.2 Text Segment Identification

In order to find meaningful text segments, we employed the C99 text segmentation algorithm [4], which groups sentences based on their similarity. The algorithm optionally takes the desired number of segments as an argument, which we used in order to get the same number of segments, and hence answer features computed, for every question-text combination. By applying C99 without that parameter first and observing its behaviour, we found that four text segments were chosen in most cases and used that setting.

In order to determine the similarity of the question and the different text fragments, we employed the `VectorIndexSourceRelatedness` measure from DKPro, a vector-based metric using an index derived from the English Wiktionary corpus. We complemented this measure by exploiting a fact that we observed about the reading tests: in many cases, the questions follow the text in a linear fashion, i.e. question 1 will likely be answered in the beginning of the text whereas question 4 will probably be answered near the end. Thus, we calculated a weight w for the similarity metric as follows:

$$w = min(num_q, num_t)/max(num_q, num_t) \tag{1}$$

where num_q is the number of the current question and num_t is the number of the current text fragment. Using this weight, we penalize similarity scores of question–fragment combinations whose relative position is very different, while leaving those whose relative position is the same unchanged.

The result of the procedure above is an ordering of text fragments for each question by weighted similarity score.

3.3 Answer Selection

In the Answer Selection step, our goal is to extract features for each answer candidate that allow a characterization of its adequacy. To that end, we compare each answer candidate to each text fragment using three different similarity metrics: the vector-space measure we already used in the previous step, the Resnik similarity measure based on WordNet [16], and the Greedy String Tiling algorithm [18] with a minimum match length of 3. The idea behind using these measures of differing linguistic depth was to capture whether answer candidates are expressed literally in the text or whether more language understanding was necessary to compare their meaning to that in the text. Following some manual inspection of training data, we hypothesized that false answer candidates would often be those that can be found literally in the text whereas the correct one would require more work for the test-taker. Consider again the example in Figure 1, where the correct answer requires the knowledge that *consider* and *judge* are synonymous in the context.

The three similarity measures were extracted for every combination of answer candidate and text fragment, in the order determined by the *Text Segment Identification* step. The ordering is crucial because the feature positions then encode whether a given measure was obtained on a fragment close to the question or not. In addition to the similarity measures, we also calculated the overlap of dependency head words and the overlap of semantic roles between answer candidate and text fragment.

The features obtained were used to train a ranking Support Vector Machine [10]. We used the SVMRank implementation[1] with hyperparameter $c = 20$. Ranking was chosen as the preferred approach because it allows characterization of answer candidates in relation to each other and it eliminates the problem of ties, i.e. that two given answer candidates are judged to be equally good and no winner can be determined.

4 Results

In this section, we report and discuss the results we obtained. We first briefly describe the experiment setup before turning to the results proper.

4.1 Experiment Setup

We used the 2014 data set as our training set and the 2013 data set as the development set. This was purely arbitrary, as in principle any setup that has completely distinct training and development sets is valid. Both data sets are approximately equal in size with 12 reading tests and 58 ± 2 questions. We used the development set to select the combination of components and measures we would use for the final submission, and we submitted one run only. For training the final model and testing on the official 2015 test data, we naturally used both training and development data. We did not make use of the possibility not to answer some of the questions, so accuracy and the official evaluation measure c@1 [14] are the same in our case.

4.2 Quantitative Results

Table 3 shows the results of our system in relation to the best runs according to overall c@1 of each participating team on the official 2015 test set. Additionally, we included the random baseline and the worst submission. A reading test counts as passed if on the total of its questions one achieves a c@1 score of at least 0.5.

Team	Overall c@1	# tests passed
Synapse	0.58 (52/89)	16/19
LIMSI-QAEE	0.36 (32/89)	8/19
cicnlp	0.30 (27/89)	6/19
NTUNLG	0.29 (26/89)	6/19
CoMiC	0.29 (26/89)	5/19
Random	0.25 (22/89)	N/A
Worst	0.21 (19/89)	3/19

Table 3. Quantitative results of our submission in relation to others

[1] http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

As can be observed in the table, our system clearly beats the random baseline. However, on the one hand, it does not reach the performance level of the currently top-performing systems. On the other hand, one can observe that there is a steep drop in c@1 from the top system (Synapse) to the runner-up (LIMSI-QAEE). With our result of 0.29, we place ourselves exactly in the middle if one considers all 17 submissions. In the next section, we will try to provide some insight into what the biggest issues are for our system and what can be done to improve performance.

5 Discussion

We proceed by first discussing two particular reading tests of the 2015 test set – one where our system did badly, and one where we achieved the best c@1 score among all participants. The remainder of this section is dedicated to discussing the performance of the *Text Segment Identification* sub-module, since it is of special interest to our research agenda.

5.1 Example Reading Tests

The text of reading test 2 is about tofu, how it came from China to Japan, is used differently in both cuisines, and how it became more popular in western countries as well. Our system did not answer a single question correctly, a fact which we attribute to two main problems: first, the text is quite short with only 312 words which means there is less room for error in identifying the right text fragments – if the algorithm is off by one or two sentences, the fragment will not contain all the necessary information needed to confirm or reject an answer. Second, the vocabulary is not very diverse across text passages, which makes it hard to compare meaning based on the essentially term-based similarity metrics we employ. It seems that what is needed here is a comparison of relations between terms, not just the terms themselves.

Test 13 is in some ways the opposite of this: it has a long text (738 words) on the success story of a rock-climber who lost his legs but still wants to climb the “highest vertical cliff on earth”. Our system achieved a c@1 score of 0.83 on this test – the best result among all participants. The text contains some rather specialized vocabulary which differs across text passages, so we assume this is why our approach does well here.

5.2 Evaluation of Text Segment Identification

As described in section 3.2, our system first ranks the automatically determined segments of a text according to their similarity to every question. This similarity additionally is weighted by a linearity weight (see equation 1) to prefer selections of segments parallel to the linear order of questions to this text.

In order to evaluate the performance of the system's text segment identification module in isolation, the official 2015 test questions were manually annotated for this subtask after applying the C99 text segmentation algorithm. We then compared the system's prediction for the segment containing the answer to a given question against the manually annotated gold standard segment. For 48 out of 89 questions, the system predicted the correct text segment (micro-average = 0.54). The macro-average with respect to individual tests was 0.57.

A manual inspection of the test data indicated a less strong parallelism between the questions and the corresponding segment than we had seen in the training data. Therefore we ran our system without the linearity weight and compared the results. Although the total number of correctly predicted text segments did not change (micro-average = 0.54), the distribution of correctly predicted segments did decrease (macro-average = 0.53). Moving from accuracy to correlation, the difference between the system with and without the linearity weight becomes even more evident: both measures exhibit a drop from 0.57 to 0.37, showing that even if our system does not always identify the correct segment, it gets much closer to doing so by exploiting this task-specific feature.

We also analyzed the tests for which our system predicted all segments correctly (tests 12 and 16). While test 12 has 4 questions, and the order of the corresponding segments shows a perfect positive correlation with the question IDs, test 16 only has 3 questions, but 4 segments. Our system correctly identified segment 3 as being irrelevant for answering any of the questions.

This analysis shows that our system can distinguish between relevant and irrelevant segments in certain cases. In a direct comparison, the linearity feature proved to help modeling aspects of the data, although the system was too biased towards modeling a linear order, which indicates a need for fine-tuning the weight of this feature.

5.3 Additional Synonym Feature

Recall our hypothesis from section 3.3 that correct answers tend to paraphrase content from the text in order to make the recognition task more challenging for humans. In order to further investigate this hypothesis and better distinguish between similarity at the word surface level and deeper lexical similarity, we added an additional feature after our official submission: a similarity score where all lemmas occurring in both the answer candidate and the text segment were filtered out before passing the remaining lemmas to the previously mentioned vector space measure. We tested the result on our development set and found an improvement of 3.3% (2 out of 60 questions) in c@1 compared to our previous model, which suggests that it is beneficial to examine how exactly correct answers differ from false ones.

6 Conclusion

We have presented a new approach to the 2015 English Entrance Exams task. The system was developed from scratch and is built on the idea of exploiting reading comprehension task characteristics, such as text structure and the way distractor answers are constructed. We achieve a c@1 score of 0.29 which is a medium but encouraging result considering the limited time in which the system was put together and the various ways in which it could be improved.

As identified by the results discussion, our system could be improved mainly in two ways. First, a better tuning of our *Text Fragment Identification* component, concerning the number and size of text fragments as these are dependent on the task and data set. Also, while clearly useful, the linearity weight we used in order to exploit question order introduced a strong bias. The second strand concerns the introduction of a representation that allows for the comparison of relations between entities in answer and text instead of only employing term-based similarity metrics. One way of achieving this could be to explore the use of Lexical Resource Semantics [17] in answer comparison, a formalism already used successfully in Short Answer Assessment [9].

Acknowledgments

We are grateful to Cornelius Fath for manual inspection of the training data and to Detmar Meurers and Kordula De Kuthy for helpful insights during the system development and feedback on the report. We would also like to thank one anonymous reviewer for comments.

Bibliography

[1] Agirre, E., Banea, C., Cardie, C., Cer, D., Diab, M., Gonzalez-Agirre, A., Guo, W., Mihalcea, R., Rigau, G., Wiebe, J.: Semeval-2014 task 10: Multilingual semantic textual similarity. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014). pp. 81–91. Association for Computational Linguistics (2014), http://aclweb.org/anthology/S14-2010

[2] Bär, D., Zesch, T., Gurevych, I.: Dkpro similarity: An open source framework for text similarity. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations. pp. 121–126. Association for Computational Linguistics, Sofia, Bulgaria (August 2013), http://www.aclweb.org/anthology/P13-4021

[3] de Castilho, R.E., Gurevych, I.: A broad-coverage collection of portable nlp components for building shareable analysis pipelines. In: Ide, N., Grivolla, J. (eds.) Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT (OIAF4HLT) at COLING 2014. pp. 1–11. Association for Computational Linguistics and Dublin City University, Dublin, Ireland (Aug 2014)

[4] Choi, F.Y.Y.: Advances in domain independent linear text segmentation. In: Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference. pp. 26–33. NAACL 2000, Association for Computational Linguistics, Stroudsburg, PA, USA (2000), http://dl.acm.org/citation.cfm?id=974305.974309

[5] Choi, J.D., Palmer, M.: Transition-based semantic role labeling using predicate argument clustering. In: Proceedings of ACL workshop on Relational Models of Semantics (RELMS'11). pp. 37–45 (2011)

[6] Dagan, I., Glickman, O., Magnini, B.: The pascal recognising textual entailment challenge. In: Candela, J.Q., Dagan, I., Magnini, B., d'Alché Buc, F. (eds.) Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification and Recognizing Textual Entailment. Lecture Notes in Computer Science, vol. 3944, pp. 177–190. Springer (2006), http://u.cs.biu.ac.il/~dagan/publications/RTEChallenge.pdf

[7] Dzikovska, M., Nielsen, R., Brew, C., Leacock, C., Giampiccolo, D., Bentivogli, L., Clark, P., Dagan, I., Dang, H.T.: Semeval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment challenge. In: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013). pp. 263–274. Association for Computational Linguistics, Atlanta, Georgia, USA (June 2013), http://aclweb.org/anthology/S13-2045

[8] Ferrucci, D., Lally, A.: UIMA: An architectural approach to unstructured information processing in the corporate research environment. Natural Language Engineering 10(3–4), 327–348 (2004),

http://journals.cambridge.org/action/displayAbstract?fromPage=online&aid=252253&fulltextType=RA&fileId=S1351324904003523

[9] Hahn, M., Meurers, D.: Evaluating the meaning of answers to reading comprehension questions: A semantics-based approach. In: Proceedings of the 7th Workshop on Innovative Use of NLP for Building Educational Applications (BEA-7) at NAACL-HLT 2012. pp. 94–103. Montreal (2012), http://purl.org/dm/papers/hahn-meurers-12.html

[10] Joachims, T.: Optimizing search engines using clickthrough data. In: Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD). pp. 217–226. ACM (2002)

[11] Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations. pp. 55–60. Association for Computational Linguistics, Baltimore, Maryland (June 2014), http://www.aclweb.org/anthology/P/P14/P14-5010

[12] Nivre, J., Nilsson, J., Hall, J., Chanev, A., Eryigit, G., Kübler, S., Marinov, S., Marsi, E.: MaltParser: A language-independent system for data-driven dependency parsing. Natural Language Engineering 13(1), 1–41 (2007), http://w3.msi.vxu.se/~nivre/papers/nle07.pdf

[13] Ott, N., Ziai, R., Hahn, M., Meurers, D.: CoMeT: Integrating different levels of linguistic modeling for meaning assessment. In: Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval). pp. 608–616. Association for Computational Linguistics, Atlanta, GA (2013), http://aclweb.org/anthology/S13-2102.pdf

[14] Peñas, A., Rodrigo, A.: A simple measure to assess non-response. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. pp. 1415–1424. Association for Computational Linguistics (2011), http://aclweb.org/anthology/P11-1142

[15] Petrov, S., Barrett, L., Thibaux, R., Klein, D.: Learning accurate, compact, and interpretable tree annotation. In: Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (2006), http://www.eecs.berkeley.edu/~petrov/data/acl06.pdf

[16] Resnik, P.: Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. Journal of Artificial Intelligence Research (JAIR) 11, 95–130 (1999), http://www.cs.cmu.edu/afs/cs/project/jair/pub/volume11/resnik99a.pdf

[17] Richter, F., Sailer, M.: Basic concepts of lexical resource semantics. In: Beckmann, A., Preining, N. (eds.) European Summer School in Logic, Language and Information 2003. Course Material I, Collegium Logicum, vol. 5, pp. 87–143. Publication Series of the Kurt Gödel Society, Wien (2004)

[18] Wise, M.J.: String similarity via greedy string tiling and running karp-rabin matching. Online Preprint 119 (December 1993)