ADAPT Centre Cone Team at IJCNLP-2017 Task 5: A Similarity-Based Logistic Regression Approach to Multi-choice Question Answering in an Examinations Shared Task

Daria Dzendzik
ADAPT Centre
School of Computing
Dublin City University
Dublin, Ireland
daria.dzendzik@adaptcentre.ie

Alberto Poncelas
ADAPT Centre
School of Computing
Dublin City University
Dublin, Ireland
alberto.poncelas@adaptcentre.ie

Carl Vogel
School of Computer Science and Statistics
Trinity College Dublin the University of Dublin
Dublin, Ireland
vogel@tcd.ie

Qun Liu
ADAPT Centre
School of Computing
Dublin City University
Dublin, Ireland
qun.liu@adaptcentre.ie

Abstract

We describe the work of a team from the ADAPT Centre in Ireland in addressing automatic answer selection for the Multi-choice Question Answering in Examinations shared task. The system is based on a logistic regression over the string similarities between question, answer, and additional text. We obtain the highest grade out of six systems: 48.7% accuracy on a validation set (vs. a baseline of 29.45%) and 45.6% on a test set.

1 Introduction

This paper describes the participation of the ADAPT Centre team in the Multi-choice Question Answering in Examinations Shared Task 2017.[1] This is a typical question answering task that aims to test how accurately the answers of the questions in exams could be selected. Any additional sources, such as knowledge base, textbooks or articles, can be used to exploit support information. The English subset contains 5,367 questions from five domains: Biology, Chemistry, Earth Science, Life Science and Physical Science. The questions come from real exams. Every question has four answer candidates which may be a word, a value, a phrase or a sentence. This challenge is an important step towards a rational, quantitative assessment of natural language understanding capabilities.

[1] See http://www.nlpr.ia.ac.cn/cip/ijcnlp/Multi-choice_Question_Answering_in_Exams.html – October 2017.

Our approach is to extract relevant information from Wikipedia[2] and to apply logistic regression over string similarities. We used an adaptation of methods developed for a comparable task in relation to question answering about films (Dzendzik et al., 2017). The features we employ are similarities of a term frequency–inverse document frequency (TF-IDF) metric, character n-grams (with n ranging from 2 to 5), bag of words, and windows slide (a ratio between answer and substrings of extracted data) – the window slide ratio is described in further detail below (section 2.3). We train our model in two ways: combining training over all domains and separate model training for each domain. The second way yields a better result on the validation set: 48.7% of accuracy vs. 43.6%. Finally, we obtained 45.6% accuracy on the test set, and this is the best result for the English dataset of the six competing systems.

The paper is organized as follows: We detail our approach in Section 2. In Section 3, we describe experiments and results on the training, validation and test data. In Section 4, we compare our approach to previous research in answering examination questions. Finally, Section 5 contains conclusions and directions of future work.

2 Approach

We constructed our model as a four-step pipeline:

1. Preprocessing – cleaning of the input data.
2. Data selection – based on key words from the question we extract relative sentences from

[2] wikipedia.org– September 2017

Proceedings of the 8th International Joint Conference on Natural Language Processing, Shared Tasks, pages 67–72,
Taipei, Taiwan, November 27 – December 1, 2017.

Wikipedia.

3. Feature Vector Concatenation – for every question, a feature vector is built as a concatenation of vectors of similarities between the answer candidates and sentences obtained in the previous step.
4. Logistic Regression – a logistic regression over the feature vector.

Later in this section, we describe our approach to multi-choice question answering in detail.

2.1 Preprocessing

The first step is to clean the question and answers statements. Using regular expressions, the existing serial numbers and letters of the question and answer candidates are deleted. See the examples (1)-(3).

(1) *"3. What ..." → "What ... "*

(2) *"8) Who ..." → "Who ... "*

(3) *"a) Paper: Paper degrades ... " → "Paper: Paper degrades ..."*

2.2 Data Selection

A list of keywords are extracted from the question statement using Natural Language Toolkit (nltk)[3] implementation of the Rapid Automatic Keyword Extraction (RAKE) algorithm (Rose et al., 2010). These words are used to retrieve a list of sentences from an information source. For each question, we select the top-50 sentences as ranked by containing (unweighted) keywords related to the item

The information source used in this system is a set of sentences from articles of Wikipedia. Each sentence is stored as a document in an inverted index data structure using Lucene.[4]

2.3 Feature Vector Creation

The data is organized in triplets (q, a, S) where a is a candidate answer, q is the question that a belongs to, and S is the set of sentences retrieved from Wikipedia by querying the keywords of the question q.

Then, for every triplet a feature vector is created by concatenating different features. Each feature is a function of two arguments and encodes the similarity between two strings. The first argument is either the answer a or a concatenation of the question and the answer $q + a$ and the second argument is S. The value of every feature is between 0 and 1: the higher the value, the more similar the strings are.

The five similarities are based on the work of (Dzendzik et al., 2017), and are described below.

- **TF-IDF cosine similarity** - the cosine similarity between `TF-IDF` representations as defined in Equation (4).

$$cos\_tfidf(a, S) = \frac{w_a \cdot w_S}{|w_a||w_S|} \quad (4)$$

  where w_a and w_S are `TF-IDF` vector representations of the answer (or question + answer) and the sentences correspondingly.

- **Bag of words ratio similarity** - a bag of words measure shows the ratio of sentences words which exist in the answer (or question + answer) as shown in Equation (5).

$$bow(a, S) = \frac{|W_a \cap W_S|}{|W_a|} \quad (5)$$

  where W_a - bag of words from the answer (or the question + the answer) and W_S bag of words from sentences.

- **Window slide ratio similarity** - returns the highest ratio of sequence match between answer (or question + answer) and all sentences substrings. The window of the substrings has a size equal to a length of the answer. See the Equation (6).

$$wSlide(a, S) = max_i(\frac{2 * M_i}{T_i}) \quad (6)$$

  where $T_i = |a| + |s_i|$ is the total number of character elements in both sequences: the answer a and s_i. s_i is i–substring of S, $\forall i s_i \in S, |s_i| = |a|$ and M_i is the number of matches between all substrings of a and s_i.

- **Character N-gram** - similar to `Window Slide` but on character level. The size of the window is limited by parameter n (We consider n = 2,3,4,5 characters). As a result, we get the ratio of n-grams overlap including white spaces in the answer (or the question + the answer) and the sentences.

[3] https://github.com/csurfer/rake-nltk – September 2017

[4] https://lucene.apache.org/core/ – September 2017

- **Word2vec cosine similarity** The cosine distance similarity (as in Equation (7)) between skip-gram representations (Mikolov et al., 2013).

$$w2v\_cos(a, S) = \frac{v_a \cdot v_S}{|v_a||v_S|} \quad (7)$$

where v_a and v_S are `Word2vec` representations of the answer (or the question + the answer) and the sentences correspondingly.

We use a pre-trained word2vec model based on small subset of Wikipedia (Tapaswi et al., 2016).

Using this metrics 17 different features are built:

1. $f_1 = w2v\_cos(q, S) + w2v\_cos(a, S)$
2. $f_2 = w2v\_cos(a, S)$
3. $f_3 = cos\_tfidf(a, S)$
4. $f_4 = bow\_overlap(a, S)$
5. $f_5 = windowSlide(a, S)$
6. $f_6 = charNgramm_2(a, S)$
7. $f_7 = charNgramm_3(a, S)$
8. $f_8 = charNgramm_4(a, S)$
9. $f_9 = charNgramm_5(a, S)$
10. $f_{10} = w2v\_cos(q + a, S)$
11. $f_{11} = cos\_tfidf(q + a, S)$
12. $f_{12} = bow\_overlap(q + a, S)$
13. $f_{13} = windowSlide(q + a, S)$
14. $f_{14} = charNgramm_2(q + a, S)$
15. $f_{15} = charNgramm_3(q + a, S)$
16. $f_{16} = charNgramm_4(q + a, S)$
17. $f_{17} = charNgramm_5(q + a, S)$

Here $q+a$ is the concatenation of q and a. Some questions from the dataset are presented as sentences with one or many gaps. If the question q includes gaps, instead of concatenating, the candidate answer a will be used to fill the gaps in the question. See the example (8).

(8) Question: *"_____ obtain energy by using the chemical energy stored in inorganic compounds"*

Answer candidates:

1. *Photoautotrophs*
2. *Chemoautotrophs*
3. *Heterotrophs*
4. *None of the above*

Concatenation strings:

1. *Photoautotrophs obtain energy by using the chemical energy stored in inorganic compounds*
2. *Chemoautotrophs obtain energy by using the chemical energy stored in inorganic compounds*
3. *Heterotrophs obtain energy by using the chemical energy stored in inorganic compounds*
4. *None of the above obtain energy by using the chemical energy stored in inorganic compounds*

For a question q the similarity features of its answer candidates a_1, a_2, a_3, a_4 are concatenated into one single vector as shown in Figure 1.

Following this method, for each question, there is one feature vector which contains information for all answer candidates inside.

2.4 Logistic Regression

The final step of our system is logistic regression over the vector. It returns the eventual answer. We use a scikit-learn[5] implementation with liblinear core and one-versus-rest schemes.

3 Experiments

In this section, we describe the data and results of our experiments.

3.1 Data

The dataset provides examination questions in two languages: English and Chinese. We focus our research on English subset of data which contains 5,367 questions from five domains: Biology, Chemistry, Earth Science, Life Science and Physical Science. Table 1 presents the division of the dataset into training, validation and test sets.

As mentioned before, we use Wikipedia dump as source of additional data.

[5] `http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html` – September 2017

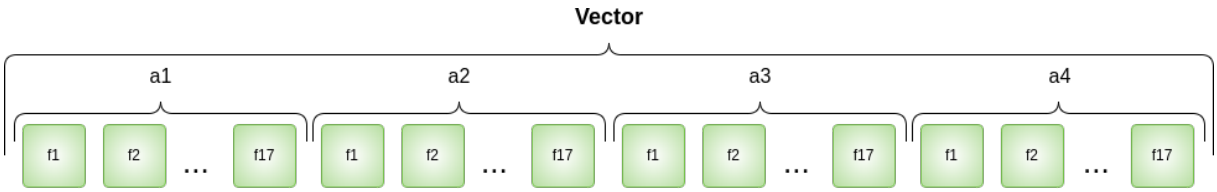

Figure 1: Feature vector concatenation.

Domain	Train	Val	Test	Total
Biology	281	70	210	561
Chemistry	775	193	581	1549
Physical	299	74	224	597
Earth Science	830	207	622	1659
Life Science	501	125	375	1001
Total	2686	669	2012	5367

Table 1: The number of questions for each domain and a division for training, validation and test sets.

3.2 Results

The systems built are the following:

- **Combined training:** Data from all domains is combined together to train one single model.
- **Domain training:** Use separate models trained in each domain. The parameters of the logistic regressions are the same for every domain.

The results obtained by these systems and the baseline[6] are presented in tables 2 and 3.

Using the Combined training approach we obtained 44.82% accuracy on the training data and 43.6% on the validation set (table 2 line 2). This result significantly outperforms the baseline (table 2 line 1). Using the Domain training approach we observe that the average score is better (51.71%) than Combined training. In addition, this approach enables evaluation of the method in each domain. The best performance is obtained in the Chemical subset – 69.55% accuracy. However, for Earth Science and for Life Science we obtained 41.08% and 42.91%, respectively. Unfortunately, we can obtain the results of separate domains only for the trainin set; we cannot compare it with the published baseline on the validation set. Finally, by concatenating the results from separate domains we obtain 48.7% accuracy on the validation set and 45.6% on the test set.

System	Train	Valid	Test
Baseline	–	29.45	–
Combined training	44.82	43.6	–
Domain training	**51.71**	**48.7**	**45.6**

Table 2: Results of all English subset of the baseline system on the validation set and our system for combined and domain training on the training, validation and test sets.

	Bio	Chem	Phys	Earth	Life
Baseline					
Valid	30	21.24	25.68	31.88	40
Domain training					
Train	49.47	69.55	51.84	41.08	42.91

Table 3: Results of the baseline system on the validation set and results of domain training on the train set for each domain.

4 Related Work

As mentioned before, this system is based on the work of (Dzendzik et al., 2017). The main difference is the data selection module: in the earlier work we select sentences from movie plot using similarities; here we extract relevant sentences from Wikipedia. Another difference is that we do not build semantic features in this shared task.

The core of our method is text similarity. We consider only five types of similarities. Gomaa and Fahmy (2013) consider more than 25 text similarity metrics in five categories: character-based similarity, term-based similarity, corpus-based similarity, knowledge-based similarity, and hybrid similarity measures. They also mention cosine similarity (for the mathematical function that determines the metric), `Tf-IDf` (which we deem to be a hybrid of term-based and corpus-based)

[6]The baseline is provided by the shared task organizers– see footnote 1.

and N-grams similarities (which may be character-based or term based), all of which we use, too.

There is interest in answering examination questions automatically. Wang et al. (2014) describe CMUs UIMA-based[7] modular automatic question answering system to automatically answer multiple-choice questions for the entrance exam in world history in English. The approach relies on two different test collection: the original test collection provided by NTCIR (NII Testbeds and Community for Information access Research)[8] organizers and the collection created by the authors.

Li et al. (2013) described the system that was used in the Entrance Exams task of Question Answering for Machine Reading Evaluation on Conference and Labs of the Evaluation Forum 2013 (QA4MRE CLEF) (Sutcliffe et al., 2013). It consists of three components, Character Resolver, Sentence Extractor and Recognizing Textual Entailment. In the system, the documents are processed by the Character Resolver in order to tag each story with a character as ID. The Sentence Extractor then extracts related sentences for each question and creates a Hypothesis (H) and Text (T). Finally it inputs this T/H pair into the Recognizing Textual Entailment system to select an answer.

5 Conclusions and Future Work

In this paper, we described the work of ADAPT Centre for the multi-choice question answering in examination shared task. In this work we have used a sentence retrieval approach with combination of logistic regression over string similarities. Our submission shows an improvement over the baseline system. According to the shared task leader board, six teams submitted their results for English subset. Our submission shows the best result on validation and test dataset and significantly outperform the baseline system.

At the same time, we believe that our system can be improved in the future: a) Using more in-domain data, for this system a set of Wikipedia articles has been used as the information source, the accuracy may be improved by using technical books or manuals; b) exploring different methods for selecting sentences from the information source (such as considering the keywords from the candidate answers); or c) Extracting a different set of keywords (for example, maximizing semantic distance among keywords selected).

[7] https://uima.apache.org/ – September 2017

[8] http://research.nii.ac.jp/ntcir/index-en.html – September 2017

Acknowledgments

This research is supported by the Science Foundation Ireland (Grant 12/CE/I2267) as part of ADAPT centre (www.adaptcentre.ie) at Dublin City University. The ADAPT Centre for Digital Content Technology is funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is cofunded under the European Regional Development Fund.

References

Daria Dzendzik, Carl Vogel, and Qun Liu. 2017. Who framed roger rabbit? answering questions about movie plot. The Joint Video and Language Understanding Workshop: MovieQA and The Large Scale Movie Description Challenge (LSMDC), at ICCV 2017, 23th of October, Venice, Italy.

Wael H. Gomaa and Aly A. Fahmy. 2013. Article: A survey of text similarity approaches. *International Journal of Computer Applications*, 68(13):13–18. Full text available.

Xinjian Li, Ran Tian, Ngan L. T. Nguyen, Yusuke Miyao, and Akiko Aizawa. 2013. Question answering system for entrance exams in QA4MRE. In *Working Notes for CLEF 2013 Conference , Valencia, Spain, September 23-26, 2013*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic keyword extraction from individual documents.

Richard FE Sutcliffe, Anselmo Peñas, Eduard H Hovy, Pamela Forner, Álvaro Rodrigo, Corina Forascu, Yassine Benajiba, and Petya Osenova. 2013. Overview of qa4mre main task at clef 2013. In *CLEF (Working Notes)*.

Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2016. Movieqa: Understanding stories in movies through question-answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Di Wang, Leonid Boytsov, Jun Araki, Alkesh Patel, Jeff Gee, Zhengzhong Liu, Eric Nyberg, and Teruko Mitamura. 2014. CMU multiple-choice question answering system at NTCIR-11 qa-lab. In *Proceedings of the 11th NTCIR Conference on Evaluation*

of Information Access Technologies, NTCIR-11, National Center of Sciences, Tokyo, Japan, December 9-12, 2014.