Humans Keep It One Hundred: an Overview of AI Journey

Tatiana Shavrina[1,3], Anton Emelyanov[1,4], Alena Fenogenova[1], Vadim Fomin[1,3],
Vladislav Mikhailov[1,3], Andrey Evlampiev[1], Valentin Malykh[2], Vladimir Larin[1],
Alex Natekin[6,7], Aleksandr Vatulin[6,7], Peter Romov[6,7],
Daniil Anastasyev[4,5], Nikolai Zinov[4,5], Andrey Chertok[1]
[1]Sberbank, [2]Huawei, [3]NRU HSE, [4]MIPT, [5]Yandex, [6]Data Souls, [7]ODS.ai
Moscow, Russia
rybolos@gmail.com

Abstract

Artificial General Intelligence (AGI) is showing growing performance in numerous applications - beating human performance in Chess and Go, using knowledge bases and text sources to answer questions and even pass school student examination. In this paper, we describe the results of AI Journey, a competition of AI-systems aimed to improve AI performance on linguistic knowledge evaluation, reasoning and text generation. Competing systems have passed Unified State Exam (USE, in Russian), including versatile grammar tasks (test and open questions) and an essay: a combined solution consisting of the best performing models have achieved a high score of 69%, with 68% being an average human result. During the competition, a baseline for the task and essay parts was proposed, and 98 systems were submitted, showing different approaches to task solving and reasoning. All the data and solutions can be found on github https://github.com/sberbank-ai/combined_solution_aij2019

Keywords: artificial intelligence, agi, artificial general intelligence, Russian language, question answering

1. Introduction

Since the Turing Test was introduced (Machinery, 1950), the number of different ways AI systems are assessed has significantly grown (Potthast et al., 2013; Bellemare et al., 2013; Caputo et al., 2014). Recently, the Robot College Student Test has been proposed to confirm human-level artificial general intelligence (AGI) on the capability to enrol in a university and take exams in the same way humans do (Goertzel, 2014). The test requires advanced comprehension of natural language (NLU), along with the capability to support reasoning, use of commonsense knowledge and answer generation. The AI Journey competition is designed to test AI systems on passing the Unified State Exam (USE) in the Russian language in full concordance with the official guidelines and knowledge assessment system, including an automatically evaluated knowledge testing and human-based evaluation of the essays.

2. Motivation

Previously, a few exam-oriented question answering contests were organized in English (Clark, 2015), Chinese (Cheng et al., 2016; Guo et al., 2017), and Japanese (Strickland, 2013; Fujita et al., 2014) languages. At the most large-scale among all such contests, Allen AI Science Challenge (Clark, 2015), systems were designed to answer standard 8th grade multiple choice science questions. More sophisticated AI knowledge and reasoning abilities were assessed in the AI2 Reasoning Challenge (ARC) as described in (Clark et al., 2018). The Aristo system (Clark et al., 2019) has achieved remarkable success on the Grade 8 New York Regents Science Exam covering more than 90% of the exam's multiple choice and non-diagram questions, and scored 83% on the Grade 12 Science Exam questions with 8 different approaches to human-like reasoning developed. Another work (Saxton et al., 2019) presented a challenge for mathematical reasoning evaluation on different mathematics question types. One of the state-of-the-art Transformer models (Vaswani et al., 2017) scored 14 out of 40 questions selected from publicly-available math exam variants for British school students of age 16.

However, the multifaceted nature of the Robot College Student Test has still remained a landmark challenge. Rich variety of different question types and tasks, still not considered by current systems, offers a milestone in exploring the capabilities of AI (specifically, any tasks other than multiple choice ones without diagrams are excluded from the examination (Clark et al., 2019)).

In this paper, we introduce a new challenge that extends exam-oriented question answering to multiple versatile task types. A satisfactory solution on the USE requires skills and knowledge acquired at school in spelling, orthoepy, text logic, grammar, punctuation, stylistics, semantics and text interpretation as well as writing essays. Thus, the competition includes the examination procedure fully equivalent to that of the human test-takers in order to test AI capabilities of natural language understanding, reasoning, text generation and commonsense knowledge.

3. Competition Methods

AI systems competition based on tasks tailored for human examination requires a new framework which allows for automatic and manual solution evaluation, data leakage and cheating prevention, and unique submission procedure.

The submissions are rated in full concordance with the official USE assessment guide. The score of each

task is summed and the resulting number is called a primary score. Primary scores lie within the range from 0 to 58 (greater is better). After the primary score is calculated, an official mapping table is used to calculate the secondary score that lies between 0 and 100 (greater is better). For instance, the primary score of 1 is mapped to the secondary score of 3, 27 is mapped to 50, and 58 is mapped to 100. The mapping is monotonous (the larger the primary score, the larger the secondary score), but it is not linear. We also refer to the scores in specific tasks as primary scores since such they have not passed the primary-to-secondary mapping yet. The secondary score of a test-taker is the resulting grade of their solution. A solution as well as a human test-taker can achieve a score up to 100 points. According to the statistics for the students evaluation, the average USE score is 68 points. A score of 36 points allows applying to a university, while a score of 24 points is required to get a graduation certificate.

3.1. Submission Procedure

In order to prevent potential data leakage and manual exam solving, we propose a competition format which allows the test set to be hidden from the participants and the submissions to be assessed in the isolated environment using Docker[1]. The solution is required to be archived in a ZIP file and contain:

- a meta-link to a publicly available Docker image of the solution from Docker Hub[2] thus allowing for developing systems with any set of preferred libraries and programming languages, and
- a script deploying an HTTP service which supports the following HTTP requests: GET (checking for the solution initialization, e.g. loading models) and POST (sequential receiving of the examination variants and sending the answers back in JSON format).

All the submissions are run under the same restrictions:

- 4 vCPU;
- RAM: 16 GB;
- Any access to the Internet is blocked;
- GET request time before task inference: 10 minutes;
- POST request time: 30 minutes;
- A single POST request should be completed before sending the next one;
- Maximal unarchived solution size: 20 GB;
- the Docker image size of 20 GB.

[1]https://docker.com
[2]https://hub.docker.com/


```
{
  "id": "11",
  "meta": {
    "language": "ru",
    "source": "real_test"
  },
  "text": "Укажите варианты ответов, в которых во всех словах одного ряда пропущена одна и та же буква. Запишите номера ответов.",
  "attachments": [],
  "question": {
    "type": "multiple_choice",
    "min_choices": 1,
    "choices": [
      {
        "id": "1",
        "text": "1) неразборч..вый, гел..вый"
      },
      {
        "id": "2",
        "text": "2) подмиг..вавший, исслед..вать"
      },
      {
        "id": "3",
        "text": "3) рассте..вавшийся, молодц..ватая"
      },
      {
        "id": "4",
        "text": "4) затейл..вый, постук..вает"
      },
      {
        "id": "5",
        "text": "5) скле..вающий, выколач..вают"
      }
    ]
  },
  "solution": {
    "correct": [
      "4",
      "5"
    ]
  },
  "score": 1
},
```


Figure 1: An example of examination task format.

3.2. Data and format

The exam consists of 27 question types:

- 26 versatile test type tasks on different high school curriculum themes (orthoepy, grammar, punctuation, stylistics, text analysis, etc.);
- writing an essay based on a text extracted from fiction.

Each examination question includes the following meta-fields: task id, text (question task text), question type, attachments (a set of attached files, if any), meta (question source, originating exam topic), choices (arbitrary key-value pairs of choice id and choice extracted from the question task text), answer type (the format for automatic answer evaluation), solution (question task answer) and score (maximum number of points for the task).

Answer type could be the following: 1) choice (choosing one option from the list); 2) multiple choice (choosing a subset of options from the list); 3) matching (correct matching of objects from two sets); 4) text (open answer in the form of arbitrary text).

The answer type is a string or an array of strings. If a task answer is an arbitrary phrase, it is required to be lowercased and contain no white spaces. For example, '["1", "3"]' or 'делочести' ('matterofhonour').

The training set of 135 unique variants was collected from publicly available sources[3], with subsequent formatting (see Fig. 2). The participants were allowed to use any additional data to develop the systems.

Since using publicly available variants of the USE as a test set could result in a data leakage, experts from the Higher School of Economics created 60 unique variants of the same methodological standard instead. The variants are randomly split into the public test set (30 variants) and the private test set (30 variants).

[3]https://rus-ege.sdamgia.ru, https://yandex.ru/tutor

3.3. Evaluation pipeline

Check phase
The submissions are evaluated on publicly available set of questions with known answers. For the check phase, a small sample from the training set is used. This phase is important for testing the solutions for potential errors and issues in evaluation system interaction. Evaluation result and system output are fully available for the participant.

Public Test
The submissions are evaluated on a hidden set of questions manually created by experts. Results on the public test set form a leader board during the active stage of the competition. Tasks and answer options within tasks are randomly rearranged each evaluation for further defence against the leader board probing and retrieving any information from the hidden test data.

Private Test
The submissions are evaluated on another hidden set of questions manually created by experts. Results on the private test set determine competition winners.

Evaluation objectives
Each examination question is evaluated by task type specific metrics: choice - accuracy; multiple choice - union / intersection; matching - the proportion of correctly matched pairs; text - special evaluation function, followed by a request for human-expert assessment.

3.4. Essay evaluation

Essay evaluation procedure comprises of two stages: automatic preliminary evaluation and manual expert assessment. The automatic preliminary evaluation is a helpful utility for the assessors which provides basic superficial evaluation of the generated texts for them to meet the following criteria:

- no plagiarism of either fiction texts or human-written essays;
- correct spelling;
- good sentence connectivity, absence of tautology;
- language errors (slang, swearing);
- paragraph structure;
- text length (in words).

If a submitted essay's originality score is less than 60%, it is automatically scored 0 points. The participant gets informed about it and is proposed to submit a new solution for expert assessment.
Manual essay assessment is carried out by experts who follow the official USE guidelines[4]. The guidelines require an essay to meet the criteria listed below (in addition to the automatically scored ones):

1. a problem is stated in the source text;
2. there are comments to the problem with at least two examples provided;
3. author's attitude is spotted;
4. there are comments to the author's attitude;
5. the essay is semantically integral;
6. the writing style is accurate and expressive;
7. punctuation is correct;
8. the essay conforms with
   - language norms;
   - writing norms;
   - ethical norms;
9. the essay is factually accurate.

The essay should be of not less than 150 words and should be thematically and problematically related to the short text given in the task - usually some excerpt from a fiction book included in the high school curriculum, containing a moral or philosophical problem.
Essays are evaluated by the experts in the assessment system developed by the organizers. Each essay is checked by three experts independently. The expert assessments are automatically compared, and in case of significant assessment difference the submission is additionally evaluated by an expert who have not seen this essay.

4. Baseline

The baseline proposed by the organizers' team[5] scored 30 points out of 100 and was organized as follows:

- the task classifier - receives JSON with the input task, determines the task type and then calls a script that solves tasks of the specific type;
- solver script for each of the 27 types of tasks. Each script imports embedding models (embedders), classifiers and knowledge bases relevant to the given task from a common pool;
- embedders: BERT embedder multilingual(Devlin et al., 2019) model for obtaining vector representations at word-, sentence-, and text-level;
- language models - we used the n-gram frequency base of the Russian National Corpus (RNC)[6] for tasks of grammatical error detection and spelling;
- morphology and syntax parsers - pymorphy2 (Korobov, 2015) and UDPipe(Straka and Straková, 2017) were used to determine the part of speech, case, number, gender, the normal form of a word and the connections between words; classifiers for

[4] http://obrnadzor.gov.ru/common/upload/news/infomaterial/ESOCO_eng_Print.pdf

[5] https://github.com/sberbank-ai/ai-journey-2019

[6] http://www.ruscorpora.ru/

making specific decisions: binary classifiers for punctuation tasks to where to put a comma, a dash, a colon, etc.

- knowledge bases - an orthoepic dictionary (a dictionary of the accepted pronunciation including word stress) - as in the school dictionaries. a dictionary of tropes - literary means: synonyms, antonyms, paronyms, idioms, etc., collected from publicly available resources. a collection of school essays on classical literature - for finetuning the generative model.

- essay models - the baseline model for writing included the following solution:

    1. LDA(Blei et al., 2003) thematic modelling + templates

    2. TextRank(Mihalcea and Tarau, 2004) + templates

    3. ULMFit AWD LSTM(Howard and Ruder, 2018) model

  The final essay was obtained by following these steps.

    1. At first, LDA was used to determine the theme of the source text and a corresponding introduction template was selected, e. g., "The topic of parents and children is covered by many classics of literature".
    2. Then, TextRank was used to extract the most important sentences of the source text, and they were inserted into templates such as: "The position of the author is expressed in the following sentences:", "This paragraph outlines the opinion of the author..."
    3. Subsequently, the resulting tests with filled gaps were given as an input to the AWD LSTM fine-tuned on the school essays, which generated a continuation up to a limit of 450 words.

5. Participants' Solutions Review

During the competition, 98 teams submitted their solutions, each of which could receive an automatic assessment of the test part an unlimited number of times, and receive a manual assessment of the essay 12 times. There were 2355 submissions in total.[7]

5.1. Best Approaches to Specific Tasks

We present the analysis of the best solutions of the top 10 teams below, considering all types of exam tasks.

[7] https://contest.ai-journey.ru/en/leaderboard

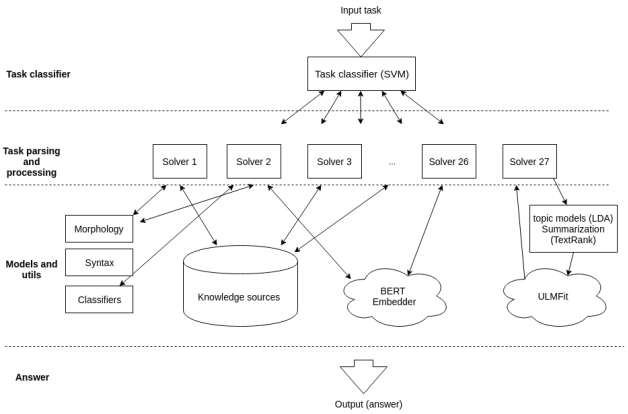

Figure 2: Baseline solution architecture.

5.1.1. Semantics - tasks 1, 3, 24

Task 1 - select one or more sentences containing the general information on the task text with 5 choices provided (Answer type: multiple choice)
Baseline Solver: maximum cosine similarity between sentence-level embedding of the choice and text-level embedding of the task text.
Best solutions:
The two commands with the best solutions approached this task by

1. choosing the options with the highest cosine similarity upon the fasttext(Bojanowski et al., 2016) vectors,

2. by returning the two closest options, since all of the sentences relevant to the text are expected to be close (closeness is determined by the cosine similarity of the options' BERT embeddings).

Task 3 - select the most relevant word meaning in the given context with 5 choices provided (Answer type: choice)
Baseline Solver: maximum cosine similarity between sentence-level embedding of the word meaning and text-level embedding of the task-text.
The best solutions approached this problem either by treating the problem as a masked task and solved it with BERT or by crawling a large additional amount of test variants.
Task 24 - find specific literary means in the given range of enumerated sentences; typically, contextual synonyms, contextual antonyms, phraseological units, etc. (Answer type: text)
Baseline Solver: a combination of knowledge base retrieved from publicly available synonym, antonym, and phraseology dictionaries, sentence preprocessing procedure and rules.
Best solutions include a component-based approach where synonyms and antonyms are found with fasttext, idioms are extracted by means of dictionary lookup, and in all other cases, the system simply returns the least frequent word in the text that is also not a proper name. They also include a component-based

approach that combines rules, morphological analysis by the Python library Mystem[8], Word2Vec(Mikolov et al., 2013) and dictionary lookup.

5.1.2. Orthoepy - task 4

Task 4 - select one word with correct or incorrect stress out of 5 marked words (Answer type: text)
Baseline Solver: use of the knowledge base retrieved from the train tasks.
Best solutions: an improved version of the baseline approach where dictionary lookup is combined with memorizing correct and erroneous word stresses and improving the rules to choose the right option; using dictionaries and rules to score candidates.

5.1.3. Grammar - tasks 5-8

Task 5 - select and replace an incorrect word with a paronym (i. e. a word of similar spelling and pronunciation but different meaning) within 5 sentences (Answer type: text)
Baseline Solver: a combination of UDPipe to extract syntactic relations, knowledge base retrieved from publicly available paronym dictionaries to get candidates and bigram frequency dictionary to rank the candidates.
Three systems scored the highest upon this task. All three use dictionary lookup to retrieve potential candidates. Although they use different approaches to score those candidates. One system treats the problem as a masked task and uses BERT to score potential replacements. The other extracts several features (including morphological properties and ngram frequency data) and scores the candidates with a custom formula. The third system does the scoring by means of fasttext, which is used to calculate the similarity between each word in the sentence and its context, represented as the average of the fasttext vectors of all other words. In the same manner, the similarity between potential candidates and their supposed contexts is calculated. The candidate with the highest difference between the similarity of original word and the similarity of the replacement is then retrieved.
Task 6 - select and exclude (typically, a redundant word) or replace a grammatically incorrect word with a correct word form (Answer type: text)
Baseline Solver: word exclusion based on a combination of sentence preprocessing procedure and maximum cosine similarity on sentence-level embeddings of generated bigrams.
Best solutions: pairwise comparison of fasttext embeddings of all nouns and verbs with exception of stop words; a dictionary lookup approach with a fallback to word2vec and cosine similarity in case the lookup is failed.
Task 7 - select and replace a grammatically incorrect word with a relevant word form within the given context from 5 word phrases (Answer type: text).
Baseline Solver: few-shot classification on word-level embeddings of the task choices with the incorrect word considered as the less frequent word in the Russian National Corpus.
Best solutions: scoring candidates with n-gram models and morphological analysis; a sophisticated system of rules that includes custom dictionaries; an improved version of the baseline that includes dictionary lookup.
Task 8 - one-to-one matching of 5 grammatical error types with 9 provided sentences (Answer type: matching).
Baseline Solver: a combination of UDPipe to extract grammatical and syntactic relations, sentence preprocessing procedure and rules to generate grammatical error candidate.
Best solutions: using BERT for multi-class classification; a complex rule-based approach.

5.1.4. Spelling - tasks 9-15

Task 9 - select one or more word sets; there is a gap in each word root corresponding to vowels in easily misspelled positions (Answer type: multiple choice).
Baseline Solver: a combination of rules and knowledge base retrieved from the train tasks.
Best solutions: a dictionary, a morphological analyzer and frequencies of word n-grams; an intricate system of rules based on regular expressions.
Task 10 - select one or more word rows in which all the words should have the same letter instead of a gap; the gap is within a prefix or morpheme boundary (Answer type: multiple choice).
Baseline Solver: use of knowledge base retrieved from pymorphy2 dictionaries.
Best solutions: morphological analysis and dictionary lookup; a version of the baseline approach with more complex rules; a version of the baseline approach with a custom knowledge base.
Task 11 - select one or more word rows in which all the words (typically, nouns and adjectives) should be completed with the same letter; the open gap is placed within a prefix or morpheme boundary (Answer type: multiple choice).
Baseline Solver: use of knowledge base retrieved from pymorphy2 dictionaries.
Best solutions: a complex component-based approach with custom dictionaries, morphological analysis and rules; a logistic regression fit on word features to predict the missing letter.
Task 12 - select one or more word rows in which all the words (typically, verbs and gerunds) should be completed with the same letter; the open gap is placed within a suffix or morpheme boundary (Answer type: multiple choice).
Baseline Solver: use of knowledge base retrieved from pymorphy2 dictionaries.
Best solutions: same as in task 11.
Task 13 - select one out of 5 sentences in which the specified word is written separately with the previous one in the given context (Answer type: text).
Baseline Solver: few-shot classification on word-level embeddings of the task choices with representatives retrieved from train tasks.

[8] https://pypi.org/project/pymystem3/

Best solutions: using BERT for binary classification (remarkably, this solution achieved 0 per cent error rate for that task on the private test set).

Task 14 - select one out of 5 sentences in which two specific words (typically, complex conjunctions) are written separately in the given context (Answer type: text).
Baseline Solver: few-shot classification on word-level embeddings of the task choices with representatives retrieved from train tasks.
Best solutions: a combination of morphological analysis, language n-gram models and rule-based approach; getting the impossible spellings out of consideration by means of pymorphy2 and scoring the remaining candidate spellings with BERT after replacing the candidate word with a [MASK] token.

Task 15 - select gaps (up to 9 gaps in a sentence) corresponding to the specified spelling, typically "н" or "нн" letter combination within an affix or morpheme boundary in the given context (Answer type: text).
Baseline Solver: few-shot classification on word-level embeddings of the words containing gaps - classification is carried out on the embeddings of representatives retrieved from train tasks
Best solutions: a combination of morphological analysis, language n-gram models and rule-based approach; BERT classifier trained to predict whether a masked gap stands for a letter combination mentioned in the task definition.

5.1.5. Punctuation - tasks 16-21

Task 16 - restore the punctuation in 5 task choices and select one or more sentences containing only one comma (Answer type: multiple choice).
Baseline Solver: CatBoostClassifier[9] trained on the features obtained with CountVectorizer[10] POS-tag ngram_range of 4.
Best solutions: feature extraction with dependency parsing with UDPipe and classification with a random forest; an LGBM classifier upon a bag of word n-grams and a bag of pos-tag n-grams.

Tasks 17-20 - restore sentence punctuation and select the gaps (up to 11 gaps) corresponding to the comma in the given context (Answer type: multiple choice)
Baseline Solver: CatBoostClassifier trained to predict the comma given the POS-tag window of 3 as categorical features.
Best solutions: replacing each placeholder with a [MASK] token, using BERT's output to decide if this placeholder replaces a comma, carefully chosen probability thresholds (individual for each task).

Task 21 - select 2 or more sentences that share the same syntactic rule on the use of versatile punctuation marks (Answer type: multiple choice).
Baseline Solver: siamese bi-LSTM network trained to predict whether the given pair of sentences share the same syntactic rule. This network takes two different sentences as input and then use a single bi-LSTM followed by two dense layers to perform the binary classification. The weights of the bi-LSTM and the dense layers are shared for both of the sentences.
Best solutions: morphological analysis (pymorphy2) and rule-based approach; an LGBM classifier fit on TF-IDF and morphological features from pymorphy2.

5.1.6. Logic - tasks 2, 22

Task 2 - fill in a gap between sentences or text parts with the most relevant logical connector or a conjunction without choices provided (Answer type: text).
Baseline Solver: multi-layer perceptron classifier trained on sentence-level embedding of the task text trained to predict a connector as a categorical feature.
Best solutions: using a custom list of candidates and scoring them as a masked task with BERT; combining BERT's masked tasks, morphological analysis (pymorphy2) and a custom list of candidates.

Task 22 - select one or more statements relevant to a task text content with 5 choices provided (Answer type: multiple choice).
Baseline Solver: maximum cosine similarity between sentence-level embedding of a choice and text-level embedding of the task text.
Best solutions: training BERT for a binary classification that outputs 1 if a choice is relevant to the task text, using BERT embeddings of options and their closest neighbours in the task text as features.

5.1.7. Discourse and text analysis - tasks 23, 25-26

Task 23 - select one or more relevant or irrelevant statements concerning versatile discourse types of task text sentences (Answer type: multiple choice).
Baseline Solver: a combination of rule-based multi-class discourse type classification, text preprocessing procedure and Logistic Regression trained to predict whether the statement is or is not of the specific discourse type.
Best solutions: scoring the candidates upon the cosine similarity of word2vec embeddings; a fine-tuned approach based on the baseline solver.

Task 25 - select a sentence which is linked to the previous one with a versatile connector within the specified sentences range, if any (Answer type: choice).
Baseline Solver: a combination of sentence preprocessing procedure and rule-based classification.
Best solutions: a rule-based approach based upon dictionary lookup and custom list of connectors of

[9] https://catboost.ai/docs/concepts/python-reference_catboostclassifier.html
[10] scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

various types.

Task 26 - one-to-one matching of 4 sentences with 9 out of 40 possible versatile literary means (Answer type: matching).
Baseline Solver: an ensemble of rule-based target class unification, sentence preprocessing procedure and Logistic Regression trained to predict whether a sentence is of specific literary means
Best solutions: using BERT for multi-class classification; combining the baseline approach with the rules.

5.1.8. Essay - task 27

Almost all of the best solutions use templates (in one way or another) rather than generation – largely due to the human evaluation procedure, which turned out to be strict for generative models - the experts reacted very negatively to the generation errors in the text. In the framework of the experiments, the participants applied not only ULMFit but also GPT2, but these solutions did not receive enough points. The generation of non-existent books and characters, inhuman grammatical errors made a logical boundary, beyond which the generation result is not considered a text already, but a meaningless bag of words.

The best solution achieved 68% of the maximum score on average of 3 topics.
The solution is based on original templates for an introduction, author's attitude, an agreement with the author's attitude, a conclusion. The theme classifier determines the subject of the text based on the match with the given keywords and corresponding theme names. The author's full name is extracted from the source text and the task so that the binding to the topic looks natural. For part of the argumentation, a knowledge base was compiled containing argumentation on existing topics, collected from textbooks and websites. All templates are randomly selected from a subset, joined together, the author's full name and argument are substituted. The output of the solution is a completely coherent and logical text containing 70% of the original material.
Second Best Essay - 59% of the maximum score on average of 3 topics.
The second-best solution is similarly based on templates, but it utilizes multilingual Universal Sentence Encoder[11] for the problem and author's attitude retrieval. Firstly, the most relevant problem and author's attitude are obtained, where relevance is measured by cosine similarity between the text and problems Universal Sentence Encoder embeddings. Then, a supportive argument for the author's attitude is found similarly.

The most interesting solutions with lower scores have the following interesting architectural solutions:

- NER for extracting the names of the mentioned heroes in the source text and binding to them,
- a generative retelling of the source text using the summa library[12],
- a classifier of possible topics on fasttext,
- periphrases of sentences with the help of synonyms and joining of these pre-prepared sentences from original school essays.

5.2. Best Solution

In this section, we briefly describe the most notable distinctions of the winning solution which is based on the baseline and has scored 59 out of 100 on the private test set. Various applications of RuBERT (Kuratov and Arkhipov, 2019) embeddings and RuBERT model fine-tuning on specific tasks and use of additional data has demonstrated superiority over other task solving methods:

- Combination of cosine similarity over RuBERT word contextual embeddings and use of knowledge base (Task 3);
- Combination of RuBERT MaskedLM and knowledge base (Task 5);
- RuBERT binary classifier trained to predict a masked suffix in the given context (Task 15);
- RuBERT binary classifier fine-tuned to predict if a masked placeholder contains a comma in the given context (Task 16-21);
- Fine-tuned RuBERT multi-class classifier (Task 8, 26);
- Combination of the multilingual Universal Sentence Encoder (Cer et al., 2018) embeddings and fine-tuned RuBERT binary classifier (Task 22).

The winning solution is widely used by the organizers for assembling the best collective solution which has achieved a score of 69 points, with the best solver for each type of task.

5.3. Best Results for each task

The table below shows the final best scores for all exam tasks collected from all participants' solutions.

[11] https://tfhub.dev/google/universal-sentence-encoder/1

[12] https://pypi.org/project/summa/

task	primary score[13]	task	primary score[5]
task_1	0,70	task_15	0,83
task_2	0,53	task_16	0,90
task_3	0,70	task_17	0,90
task_4	0,97	task_18	0,56
task_5	0,66	task_19	0,86
task_6	0,43	task_20	0,90
task_7	0,90	task_21	0,50
task_8	0,86	task_22	0,63
task_9	0,83	task_23	0,30
task_10	0,97	task_24	0,33
task_11	0,90	task_25	0,73
task_12	0,77	task_26	0,75
task_13	1,00	task_27	0,68
task_14	0,83		

6. Competition Results

The table 1 below shows the results of the top 10 teams. Points for test tasks and for the essay part are taken into account separately. The final score was obtained using the official scale (nonlinear) for the transfer of the total scores for the test and the essay to a 100 point scale.[14] Final grades for the test are obtained by averaging the score for 30 test variants; final grades for the essay were obtained using manual assessment and averaging the grades on a sample of 3 essay themes.

As a result of the competition, there were presented many approaches to solving NLU problems; the best solution had 63 points out of 100, including all examination parts that humans pass - open and closed questions, and essay. AI Journey is the first competition of its kind, which can slightly detract from the disadvantage that some of the solutions are definitely using pure engineering approach. Many individual tasks show approximately the same error rate for the solutions based on the engineering approach in the one hand and universal models (for example, BERT) on the other hand, but universal solutions are much easier to scale and transfer to tasks of slightly different formulations, which makes them certainly better, although this criterion not evaluated in the competition. It should be noted that for two months of the competition, participants of various levels, from student teams to industrial companies, showed a high quality of their solutions, moreover, the level of solutions has been gradually rising until the very end of the competition. We now plan to make the leaderboard permanent and support the submissions of new solutions, and welcome participants who want to get the highest score with us.

We hope that the data, baseline and open source solutions will be a new start for the community of scientists and NLP-developers, and the resulting technologies will contribute to question-answer systems, knowledge bases, education applications, text writing assistants and so on[16].

7. Discussion

The proposed competition design provides a reusable framework for future competitions and environments for general question answering problems. Containerized format solves multiple common issues in competition organization: reproducible results (Tatman et al., 2018; Likhomanenko et al., 2015), secure environment with hidden test data, high flexibility in tools and approaches used by participants. Proposed data and evaluation format is suitable for many question answering problems including other knowledge domains like natural sciences and computer programming. This format could be easily extended to more sophisticated problems like Visual Question Answering (Gordon et al., 2017) by simple addition of relevant attachment files.

Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. (2013). The arcade learning environment: An evaluation platform for general agents. Journal of Artificial Intelligence Research, 47:253–279.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. Journal of machine Learning research, 3(Jan):993–1022.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606.

Caputo, B., Müller, H., Martinez-Gomez, J., Villegas, M., Acar, B., Patricia, N., Marvasti, N., Üsküdarlı, S., Paredes, R., Cazorla, M., et al. (2014). Imageclef 2014: Overview and analysis of the results. In International Conference of the Cross-Language Evaluation Forum for European Languages, pages 192–211. Springer.

Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., et al. (2018). Universal sentence encoder. arXiv preprint arXiv:1803.11175.

Cheng, G., Zhu, W., Wang, Z., Chen, J., and Qu, Y. (2016). Taking up the gaokao challenge: An information retrieval approach. In IJCAI, pages 2479–2485.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. (2018). Think you have solved question answering? try arc, the ai2 reasoning challenge. arXiv preprint arXiv:1803.05457.

Clark, P., Etzioni, O., Khot, T., Mishra, B. D., Richardson, K., Sabharwal, A., Schoenick, C., Tafjord, O., Tandon, N., Bhakthavatsalam, S., et al. (2019). From'f'to'a'on the ny regents science exams: An overview of the aristo project. arXiv preprint arXiv:1909.01958.

[13] Primary and secondary scores are discussed in Section 3.

[14] https://4ege.ru/novosti-ege/4023-shkala-perevoda-ballov-ege.html

[16] https://github.com/sberbank-ai/combined_solution_aij2019

Team name	Place	Test Score	Essay Score	Total (max 100)	Number of submissions
qbic	1	0.59	0.59	59.77	48
Bilbo Bagging	2	0.61	0.53	58.47	170
Magic City	3	0.43	0.68	55.63	115
borsden	4	0.46	0.56	53.4	74
Ololosh AI	5	0.49	0.44	50.93	150
nice	6	0.37	0.58	49.7	40
Niw	7	0.62	0.20	49.2	111
stickman	8	0.50	0.21	43.77	70
Orcs	9	0.38	0.37	43.77	19
Lamoda.AI	10	0.53	0.12	42.73	206
Best combined[15]		0.69	0.68	69	-

Table 1: Top 10 teams from final leader board for the AI Journey competition.

Clark, P. (2015). Elementary school science and math tests as a driver for ai: take the aristo challenge! In Twenty-Seventh IAAI Conference.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In NAACL-HLT.

Fujita, A., Kameda, A., Kawazoe, A., and Miyao, Y. (2014). Overview of todai robot project and evaluation framework of its nlp-based problem solving. World History, 36:36.

Goertzel, B. (2014). Artificial general intelligence: concept, state of the art, and future prospects. Journal of Artificial General Intelligence, 5(1):1–48.

Gordon, D., Kembhavi, A., Rastegari, M., Redmon, J., Fox, D., and Ali, F. (2017). Iqa: Visual question answering in interactive environments. arXiv preprint arXiv:1712.03316.

Guo, S., Zeng, X., He, S., Liu, K., and Zhao, J. (2017). Which is the effective way for gaokao: Information retrieval or neural networks? In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 111–120.

Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. arXiv preprint arXiv:1801.06146.

Korobov, M. (2015). Morphological analyzer and generator for russian and ukrainian languages. In Mikhail Yu. Khachay, et al., editors, Analysis of Images, Social Networks and Texts, volume 542 of Communications in Computer and Information Science, pages 320–332. Springer International Publishing.

Kuratov, Y. and Arkhipov, M. (2019). Adaptation of deep bidirectional multilingual transformers for russian language. arXiv preprint arXiv:1905.07213.

Likhomanenko, T., Rogozhnikov, A., Baranov, A., Khairullin, E., and Ustyuzhanin, A. (2015). Improving reproducibility of data science experiments. ICML 2015 AutoML Workshop.

Machinery, C. (1950). Computing machinery and intelligence-am turing. Mind, 59(236):433.

Mihalcea, R. and Tarau, P. (2004). Textrank: Bringing order into text. In Proceedings of the 2004 conference on empirical methods in natural language processing, pages 404–411.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In NIPS.

Potthast, M., Hagen, M., Gollub, T., Tippmann, M., Kiesel, J., Rosso, P., Stamatatos, E., and Stein, B. (2013). Overview of the 5th international competition on plagiarism detection. In CLEF Conference on Multilingual and Multimodal Information Access Evaluation, pages 301–331. CELCT.

Saxton, D., Grefenstette, E., Hill, F., and Kohli, P. (2019). Analysing mathematical reasoning abilities of neural models. arXiv preprint arXiv:1904.01557.

Straka, M. and Straková, J. (2017). Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipe. In Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, pages 88–99, Vancouver, Canada, August. Association for Computational Linguistics.

Strickland, E. (2013). Can an ai get into the university of tokyo? IEEE Spectrum, 50(9):13–14.

Tatman, R., VanderPlas, J., and Dane, S. (2018). A practical taxonomy of reproducibility for machine learning research. Reproducibility in Machine Learning Workshop at ICML 2018, Stockholm, Sweden.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems, pages 5998–6008.