ISSN 0234-0453 • ИНФОРМАТИКА И ОБРАЗОВАНИЕ • 2022 • Том 37 № 3

DOI: 10.32517/0234-0453-2022-37-3-12-20

СИСТЕМА ЗАДАНИЙ ДЛЯ ПЕРВОЙ ВСЕРОССИЙСКОЙ ОЛИМПИАДЫ ШКОЛЬНИКОВ ПО ИСКУССТВЕННОМУ ИНТЕЛЛЕКТУ

С. Г. Григорьев[1], **И. А. Калинин**[2], **Н. Н. Самылкина**[3] ✉

[1] ***Московский городской педагогический университет, г. Москва, Россия***

[2] ***Московский государственный лингвистический университет, г. Москва, Россия***

[3] ***Московский педагогический государственный университет, г. Москва, Россия***

✉ *nsamylkina@yandex.ru*

Аннотация

В статье рассматриваются основные подходы к разработке и использованию системы заданий для первой всероссийской олимпиады школьников по искусственному интеллекту. Первая олимпиада школьников по искусственному интеллекту, проводимая Министерством просвещения Российской Федерации, имела свои отличительные особенности по процедуре проведения и используемым в ходе соревнований заданиям, которые могут быть применены для сравнительного анализа и экспертной оценки проводимых в России интеллектуальных соревнований для школьников. Тематика заданий олимпиады — обработка текста на естественном языке (Natural Language Processing), компьютерное зрение (Computer Vision), наука о данных (Data Science).

Использовался язык программирования Python. Разработанные задания были ориентированы на формирование индивидуальной траектории обучающегося для подготовки к заключительному этапу олимпиады в области искусственного интеллекта, т. е. соблюдались тематическая преемственность на всех этапах и практическая направленность заданий с применением датасетов на втором и третьем этапах. Для оценивания использовалась рейтинговая оценка (как в профессиональном сообществе Kaggle.com) индивидуальных возможностей школьников в комплексном применении математических и цифровых навыков при реализации задач искусственного интеллекта.

Статья может быть полезна учителям информатики, включающим тему в углубленный курс информатики и педагогам дополнительного образования, готовящим школьников к олимпиадам по программированию.

Ключевые слова: всероссийская олимпиада школьников, искусственный интеллект, информатика, Python, наука о данных.

Для цитирования:

Григорьев С. Г., Калинин И. А., Самылкина Н. Н. Система заданий для первой всероссийской олимпиады школьников по искусственному интеллекту. *Информатика и образование*. 2022;37(3):12–20. DOI: 10.32517/0234-0453-2022-37-3-12-20

THE TASK SYSTEM FOR THE FIRST ALL-RUSSIAN OLYMPIAD IN ARTIFICIAL INTELLIGENCE FOR SCHOOLCHILDREN

S. G. Grigoriev[1], **I. A. Kalinin**[2], **N. N. Samylkina**[3] ✉

[1] ***Moscow City University, Moscow, Russia***

[2] ***Moscow State Linguistic University, Moscow, Russia***

[3] ***Moscow Pedagogical State University, Moscow, Russia***

✉ *nsamylkina@yandex.ru*

Abstract

The article discusses the main approaches to the development and use of the task system for the first All-Russian Olympiad in artificial intelligence for schoolchildren. The first Olympiad in artificial intelligence for schoolchildren, held by the Ministry of Education of the Russian Federation, had distinctive features in the procedure of the competition and in competition tasks. These features can be used for comparative analysis and expert evaluation of intellectual competitions for schoolchildren held in Russia. The subjects of the Olympiad tasks are Natural Language Processing, Computer Vision, and Data Science.

The Python programming language was used. The tasks were focused on the formation of an individual path for the student to prepare for the final stage of the Olympiad in artificial intelligence, i. e. thematic continuity was observed at all stages and the practical orientation of tasks using data sets was observed in the second and third rounds. For the assessment, a rating assessment was used (as in the professional community Kaggle.com) of the individual abilities of schoolchildren in the complex application of mathematical and digital skills in solving artificial intelligence tasks.

The article may be useful for informatics teachers who include the topic in an advanced informatics course and teachers of additional education who prepare schoolchildren for Olympiads in programming.

ISSN 0234-0453 • INFORMATICS AND EDUCATION • 2022 • Volume 37 No 3

Keywords: All-Russian Olympiad for schoolchildren, artificial intelligence, informatics, Python, data science.
For citation:
Grigoriev S. G., Kalinin I. A., Samylkina N. N. The task system for the first All-Russian Olympiad in artificial intelligence for schoolchildren. *Informatics and Education.* 2022;37(3):12–20. (In Russian.) DOI: 10.32517/0234-0453-2022-37-3-12-20

1. Введение

Всероссийская олимпиада по искусственному интеллекту для обучающихся VIII—XI классов общеобразовательных организаций проводилась в октябре-ноябре 2021 года Министерством просвещения Российской Федерации в рамках Федерального проекта «Искусственный интеллект» национальной программы «Цифровая экономика Российской Федерации» [1]. Материалы олимпиады размещены по ссылке: https://olimp.edsoo.ru/

Министерство просвещения Российской Федерации впервые проводило такую олимпиаду и, вполне возможно, что она займет свое постоянное место в списке всероссийских олимпиад школьников. На основе полученного авторами опыта участия в подготовке и проведении Всероссийской олимпиады школьников по искусственному интеллекту будет представлен материал об актуальности тематики олимпиады, ее структуре, целях и особенностях используемых заданий.

Искусственный интеллект (ИИ) сегодня применяется повсеместно, в том числе в образовании. По данным Web of Science Core Collection количество научных публикаций по теме искусственного интеллекта и анализа данных с 2016 по 2020 год возросло в 2,6 раза в сравнении с периодом 2011–2015 годов [2, 3]. Этот показатель вдвое больше роста количества таких исследований в мире, где общий прирост был в 1,3 раза, что подчеркивает темп развития данной области в мире. Научные разработки и проекты **науки о данных** (Data Science), инструменты **анализа данных** (Data Analysis) стали более доступными и используются для анализа перспективных разработок в крупных ИТ-компаниях и на различных производствах [4–6].

В настоящее время развитие науки и потребности бизнеса в цифровых компетенциях, сформированных у выпускников вузов, влияют на необходимость расширения вопросов искусственного интеллекта в содержании школьного курса информатики [7–11]. Количество публикаций о необходимости изучения вопросов искусственного интеллекта в общем образовании неуклонно возрастает [12–15]. Искусственный интеллект и анализ больших данных рассматриваются как перспективные сквозные цифровые технологии в федеральном проекте «Цифровые технологии», являющемся не только одним из важнейших проектов цифровизации страны, но и важным государственным ориентиром. Среди семи дорожных карт развития цифровых технологий, утвержденных президиумом Правительственной комиссии по цифровому развитию, использованию информационных технологий для улучшения качества жизни и условий ведения предпринимательской деятельности, две связаны напрямую с данной областью: «Нейротехнологии и искусственный интеллект», «Компоненты робототехники и сенсорика», остальные документы включают элементы ИИ. Образовательная робототехника получила мощную поддержку государственных и частных компаний в России. В настоящее время присутствует в основных и дополнительных образовательных программах общего образования, а также олимпиадах школьников и студентов [16].

На момент организации олимпиады по искусственному интеллекту в России существовали еще две такие олимпиады, проводимые вузами, — это Национальная технологическая олимпиада и Олимпиада Университета Иннополис. Их особенностями являются, в частности, ориентация на командную работу, использование ресурсов самих вузов в подготовке команд, что существенно влияет и на тематику задач, решаемых участниками этих олимпиад.

Всероссийская олимпиада по искусственному интеллекту для обучающихся VIII–XI классов общеобразовательных организаций, проводимая Министерством просвещения Российской Федерации, наряду с общими целями аналогичных проектов — поддержки талантливых школьников, а именно: выявления и развития творческих способностей обучающихся, развития познавательного интереса к технологиям искусственного интеллекта, создания условий для поддержки одаренных школьников, ставила целью повышение у обучающихся мотивации к углубленному изучению информатики и математики как основе предпрофессиональной подготовки в области информационных технологий.

В отличие от уже упомянутых выше олимпиад, Всероссийская олимпиада по искусственному интеллекту была ориентирована на индивидуальные соревнования учащихся (а не на подготовку команд), причем должна была предоставить возможность участия наиболее широкому кругу участников — в том числе тем, кто интересуется тематикой искусственного интеллекта самостоятельно и не имеет доступа к вычислительной и технической базе крупных университетов или технологических компаний.

При планировании олимпиады и разработке заданий организационный и методический комитеты предполагали, что количество участников (по крайней мере, на первом этапе) будет велико — настолько, что организация ручной проверки работ и проведения апелляций в приемлемые сроки будет невозможной. Таким образом, с самого начала задания олимпиады были ориентированы на автоматизированную проверку — с помощью специально разработанной среды.

Исходя из таких вводных параметров, было запланировано три этапа проведения олимпиады по пять дней каждый. Этапы проводились в дистанционном формате, но при разных временных условиях.

ISSN 0234-0453 • ИНФОРМАТИКА И ОБРАЗОВАНИЕ • 2022 • Том 37 № 3

2. Характеристика этапов Всероссийской олимпиады по искусственному интеллекту

Первый, **отборочный этап** олимпиады длился пять дней. Решения всех заданий можно было в произвольном порядке в течение всего промежутка времени загрузить для проверки. При неограниченных возможностях доступа к самой разной информации на различных ресурсах риски использования чужих решений были очевидны, и их необходимо было максимально снизить, что и было сделано на следующих этапах.

Второй, **основной этап** олимпиады предусматривал более «жесткий» формат работы с заданиями — их можно было сделать в любое время пяти дней этапа, но в один временной период — четыре часа.

На втором этапе олимпиады было целью отобрать тех участников, которые не только владеют общими навыками решения задач, но и знакомы с направлением «Искусственный интеллект и решение интеллектуальных задач», с некоторыми основными его методами и подходами, могут применить их для решения задач [10, 12, 15, 17, 18, 19].

Третий, **заключительный этап** продолжался также в течение пяти дней. Можно было решить задания в любое время в один временной интервал — двенадцать часов в условиях прокторинга. На этом этапе участники соревновались между собой — для предложенных заданий не существовало «полностью правильного» решения — необходимо было набрать наибольшее количество баллов при тестировании модели.

Другие организационные особенности, описанные в положении об олимпиаде, были направлены на вовлечение администрации образовательных организаций и учителей информатики в процесс подготовки и участия одаренных школьников в олимпиаде. Для обеспечения прозрачности контингента участников использовалась их регистрация через образовательные организации, которые своим приказом направляют своих обучающихся для участия в олимпиаде. При этом предполагалась возможность организации доступа к контенту олимпиады на площадках образовательных организаций при консультативной поддержке педагогов.

Как мы уже указывали, олимпиада предусматривала полностью автоматизированную проверку результатов, в которой участник не видит результата выполнения своей программы, а видит только оценку соответствия ее выводу предусмотренным критериям. От участников требовалось подготовить программу, которая либо наберет наибольшее количество баллов (т. е. пройдет наибольшее количество тестов) — на первых двух этапах, либо будет выдавать результаты, максимально близкие к «человеческой» оценке.

Задания первого отборочного этапа проверяли сформированность базовых навыков решения задач на языке Python. Это пока единственный доступный учащимся язык программирования, для которого есть полный набор средств решения задач искусственного интеллекта.

Задачей первого, отборочного **этапа**, как следует из названия, было не выявить победителя, а отобрать участников, владеющих в необходимом объеме технической базой, т. е. навыками решения задач на языке Python с использованием его стандартных библиотек (например, регулярных выражений) [20].

Важно отметить, что стремились отобрать тех, кто не просто технически освоил приемы программирования, но и осознанно применяет его специальные возможности, может поставить задачу, выдвинуть правдоподобные предположения и добиться результата. Участникам предлагалось десять заданий, можно было набрать до тридцати трех баллов.

Проверялись навыки:

1. Ввода данных и вывода результатов.
2. Использования файлов для ввода и вывода.
3. Обработки числовой информации.
4. Обработки текстовой информации, в том числе работы с различными кодировками.
5. Организации хранения данных в памяти, планирования структуры хранения данных для последующей обработки этих данных.
6. Решения переборных задач.

На этом и на последующих этапах ввод и вывод данных осуществлялся через стандартные потоки ввода-вывода (т. е. «ввести с клавиатуры» и «вывести на экран»). Код запускался в режиме консольного приложения.

Решением задания считался программный код на языке Python, соответствующий требованиям, описанным в условиях.

Участнику предоставлялась возможность проверить работоспособность кода на одном тесте неограниченное количество раз. После окончания времени на решение заданий основного тура олимпиады представить на проверку задания было невозможно, но каждая сданная вовремя на проверку задача продолжала проверяться на серии тестов. Каждый пройденный тест давал участнику один балл.

Каждый тест представлял собой набор входных данных. Каждый параметр передавался в отдельной строке через поток стандартного ввода (функция input()).

Тест считался пройденным, если программа выдавала на стандартный вывод (функцией print()) указанные в задании данные, **точно совпадающие с эталонным значением**. Любые дополнительные символы в любом месте ответа считались ошибкой.

Все решения проверялись набором автоматических тестов, выполняемых в изолированной среде, это позволило не подгонять код под конкретные выходные данные (как иногда случается на олимпиадах по программированию).

Среда включала язык Python с его набором стандартных модулей, а также библиотек NLTK, Pillow, NumPy, Pandas, SciKit-Learn, TensorFlow (включая Keras), PyTorch. Доступа к сети Интернет среда не имела.

ISSN 0234-0453 • INFORMATICS AND EDUCATION • 2022 • Volume 37 No 3

Критерием присвоения балла являлось прохождение теста.

На втором, **основном этапе** олимпиады были предложены задания для выделения тех участников, которые действительно интересуются, специализируются или могут быстро ознакомиться с основными методами интеллектуальной обработки данных. Предлагалось пять задач на использование специализированных библиотек, ориентированных на работу с числовыми, текстовыми и графическими данными, с ориентацией на подготовку и анализ наборов данных для последующего решения задач интеллектуальной обработки. Предусматривается возможность использования библиотек NumPy, Pandas, Pillow, SkLearn, NLTK, Apyory. Задания предусматривали оценку уровня знакомства школьников с возможностями этих библиотек, умения применить их для решения ранее не встречавшейся задачи.

Использовались следующие классы задач:

- классификации;
- кластеризации;
- выявления ассоциативных правил;
- регрессии.

Предполагалось, что участники в своем решении:

1) анализируют или формируют набор данных для решения задачи;
2) выбирают модель для ее решения;
3) формируют обучающий и (при необходимости) тестовый набор данных;
4) проводят обучение и при необходимости тестирование модели;
5) готовят программу, применяющую модель для формирования ответа и выводящую ответ в стандартный поток вывода.

Задачи не предусматривали самостоятельной реализации участниками алгоритмов перечисленных классов. Все необходимые для решения модели и средства имеются в составе библиотек, но ими необходимо было грамотно воспользоваться.

Всего в основном туре можно было набрать двадцать баллов.

Таким образом, перед заключительным этапом участники были подготовлены к необходимости разработки программной модели обработки данных в соответствии с заданием. На **третьем этапе** участникам предложили два задания, из которых необходимо было выполнить хотя бы одно. То есть было выбрано не одно узкое направление искусственного интеллекта, а два: компьютерное зрение (сегментация изображений) и обработка текста на естественных языках (задача оценки тональности).

Хотя предлагаемые для участников олимпиады задания включают в себя хорошо известные задачи с большим количеством различных методов и подходов к их решению, целью участников олимпиады была не разработка нового общего метода (что далеко выходит за возможности школьников), а демонстрация свободного владения имеющимися методами и знания особенностей их применения, что позволяет путем проведения вычислительных экспериментов выбрать и настроить наилучший вариант решения для конкретной ситуации и набора данных. Задания оценивались рейтинговым способом, т. е. не подразумевалось наличие абсолютно верного решения, речь шла о сравнении эффективности решений на реальных наборах данных.

Еще одной особенностью заданий третьего этапа было то, что одна из задач позволяла добиться результата большим количеством вычислительных экспериментов с помощью специализированного рабочего места и тому подобных средств, а вторая могла быть решена с помощью сравнительно небольшого объема вычислений, но требовала вдумчивого анализа, подбора и настройки методов выделения и анализа данных.

3. Разбор примеров заданий каждого этапа олимпиады

3.1. Пример задачи отборочного этапа на проверку общих навыков программирования

Есть текстовый файл литературного произведения на русском языке, в кодировке UTF-8 разбитый на строки. В файле возможно использование переносов. Объем файла — более 100 Кб.

Подготовьте программу, которая получит из стандартного ввода (функция input()) имя файла и выведет имена главных героев — первые пять по частоте словоупотребления. (Возможно, одно имя будет в этом списке упоминаться в разных формах несколько раз. Имя и фамилия считаются разными словами.) Имена вывести через пробел в нижнем регистре.

Разбор и решение задачи.

Начнем с чтения файла:

```
textFile = input()
with open(textFile,'r',encoding='utf-8') as f:
strings = f.readlines()
```

Результат выполнения этого кода — список строк из файла. Нам потребуется проанализировать эти строки — т. е. разбить их на отдельные слова и подсчитать их количество. Воспользуемся механизмом регулярных выражений и словарем:

```
import re
textFile = input("Введите имя файла для анализа=>")
words = {}
wordSplit = re.compile(r"[А-Яа-я]+")
with open(textFile,'r',encoding='utf-8') as f:
strings = f.readlines()
text = ' '.join(strings)
text = re.sub(r"([^\s])-\n\s{1}",'\g<1>',text)
for word in re.findall(wordSplit,text):
if word in words:
words[word] = words[word] + 1
else:
words[word] = 1
```

В этой части используем очень простой способ выделения слов — выделяем любую последовательность букв русского языка (можно обоснованно предположить, что в тексте на русском языке и главные

ISSN 0234-0453 • ИНФОРМАТИКА И ОБРАЗОВАНИЕ • 2022 • Том 37 № 3

герои будут иметь имена, записанные кириллицей): `[А-Яа-я]+`

Чуть сложнее дополнение, в котором учитываются переносы. Перенос — это знак «-», стоящий в конце строки, причем перед ним не пробел (слово делится на части). Поэтому мы соединяем все строки в одну строку (т. е. в единый строковый объект) через пробел, а потом выполняем замену частей этой строки, подпадающих под шаблон, т. е. убираем комбинацию из переноса, символа новой строки и пробела (появился при соединении строк): заменяем `([^\s])-\n\s{1}` на первую группу.

В получившейся «суперстроке» мы перебираем все слова (прямо по шаблону `[А-Яа-я]+`) и проверяем: если слова нет в словаре — добавляем его туда, в количестве одна штука, если есть, увеличиваем количество.

Полученный словарь отсортируем и выведем (пока для изучения результата) 50 самых частых слов.

```
sortedDict = [k for k, v in sorted(words.
items(),
  key=lambda item: item[1],reverse=True)
  if k[0] == k[0].upper()]
print(' '.join(sortedDict[0:50]))
```

В полученном списке (он зависит от файла, в данном случае были взяты первый и второй тома произведения Л. Н. Толстого «Война и мир») имеется очень много лишних слов (т. е. не имен), причем на самом деле эти слова от произведения зависят мало [21]. Создается список таких «неинформативных» с точки зрения задачи слов, взятый с сайта Национального корпуса русского языка [22].

Создадим для этого дополнительный словарь и отфильтруем слова перед подсчетом:

```
import re
textFile = input("Введите имя файла для анализа=>")
words = {}
wordSplit = re.compile(r"[А-Яа-я]+")
stop = {}
with open('stopWords.txt','r',encoding='utf-8')
  as f:
for line in f:
word=line.lower().strip()
if not word in stop:
stop[word] = 1
with open(textFile,'r',encoding='utf-8') as f:
strings = f.readlines()
text = ' '.join(strings)
text = re.sub(r"([^\s])-\n\s{1}",'\g<1>',text)
for word in re.findall(wordSplit,text):
if not word.lower() in stop:
if word in words:
words[word] = words[word] + 1
else:
words[word] = 1
sortedDict = [k for k, v in sorted(words.
items(),
  key=lambda item: item[1],reverse=True)
  if k[0] == k[0].upper()]
print(' '.join(sortedDict[0:5]))
```

Поскольку файл достаточно большой, статистические закономерности позволяют вполне уверенно предположить, что это будет работать. Конечно, для уверенности (и, возможно, для корректировки словаря) стоит провести несколько экспериментов с разными файлами.

По условиям первого тура загрузить в качестве правильного решения задачи можно было только один файл — поэтому словарь стоп-слов тоже нужно внести в текст программы.

Необходимо также обратить внимание на соблюдение условий ввода и вывода.

3.2. Примеры задач основного этапа олимпиады

Так же как и в первом этапе, предусматривались несколько уровней трудности заданий и по возможности разнообразная тематика, охватывающая как можно больше направлений ИИ.

Пример задачи на работу с набором данных.

Aditya Kadiwal собрал набор данных о нормальных и мошеннических транзакциях с банковскими картами: https://www.kaggle.com/adityakadiwal/credit-card-fraudulent-transactions.

Напишите программу, которая введет с клавиатуры имя файла с такими данными, значения параметров, а после этого по аналогичным данным сможет принять решение — разрешать транзакцию (ответ «LEGIT») или отправить на проверку человеком (ответ «FRAUD»).

Обратите, пожалуйста, внимание на то, что самое важное НЕ пропустить ложную транзакцию.

Пример задачи на работу с изображениями.

Имеется фотография груши на светлом фоне, скачанная с сайта (в виде файла формата JPEG или PNG). Листьев у груши на фото нет.

Написать программу, которая получит из стандартного потока ввода имя файла и выведет цвет груши: красная, желтая или зеленая одним словом в стандартный поток вывода.

Разбор и решение задачи.

Первое действие — найти несколько фотографий груш (либо сразу на белом фоне, либо стираем ненужное). Как бы ни решалась задача, потребуется каким-то образом формировать и проверять решение.

Второе — подбираются средства для ее решения из имеющегося списка библиотек. Прочитать файл JPEG можно с помощью Pillow. Как будет найдена и представлена груша? Подбирается метод из средств, которые предоставляет библиотека машинного обучения Scikit-learn: классификация не подходит, регрессия тоже, а кластеризация может быть использована — изображение состоит из точек, и кластеров этих точек (по условию) у нас ровно два: груша и фон. Остается выяснить, какие точки к какому кластеру относятся.

Используется самый известный метод кластеризации — метод K-средних:

```
from PIL import Image
import numpy as np
```

ISSN 0234-0453 • INFORMATICS AND EDUCATION • 2022 • Volume 37 No 3

```
import math
from sklearn.cluster import KMeans
img = Image.open(input(''))
colorPixel = []
for i in range(img.size[0]):
for j in range(img.size[1]):
pixel = img.getpixel((i, j))
colorPixel.append(pixel)
colors = np.array( colorPixel )
kmeans = KMeans(n_clusters = 2)
kmeans.fit(colors)
for l in kmeans.cluster_centers_:
print(l)
```

Прочитав изображение, создается список из всех его пикселей. Каждый пиксель является «точкой» в цветовом пространстве, его место на фотографии не важно, представляет интерес только цвет.

Превратив его в массив Numpy и выучив на этом массиве модель K-средних [10], на экране печатаются центры этих кластеров (их ровно два), т. е. «средний» цвет каждого из них.

После прогона через это преобразование нескольких изображений получается представление о том, какие это будут кластеры: «почти белый» — фон, и цвет — зависящий от груши. Остается усреднить кластеры в нескольких изображениях, и получается способ оценки цвета груши:

```
from PIL import Image
import numpy as np
import math
from sklearn.cluster import KMeans
colorSet = { 'зеленая' : (95,134,55),
             'желтая' : (225,190,44),
             'красная' : (176,68,73),
             'белый фон' : (255,255,255)}
img = Image.open(input(''))
def distanse( a, b):
r = (a[0]-b[0])**2+(a[1]-b[1])**2+(a[2]-b[2])**2
return math.sqrt( r )
colorPixel = []
for i in range(img.size[0]):
for j in range(img.size[1]):
pixel = img.getpixel((i, j))
colorPixel.append(pixel)
colors = np.array( colorPixel )
kmeans = KMeans(n_clusters = 2)
kmeans.fit(colors)
clusterSize = {}
for label in kmeans.labels_:
if label in clusterSize:
clusterSize[label] += 1
else:
clusterSize[label] = 1
for l in kmeans.cluster_centers_:
mD = 'зеленое'
for color in colorSet:
if distanse( colorSet[color], l ) <
   distanse( colorSet[mD], l ):
mD = color
if mD.find('фон') == -1:
print(mD)
```

Чтобы найти «ближайший» цвет для каждого кластера, используется Евклидово расстояние. В словаре с каждым кластером связано название, так что сразу получается и результат для вывода — цвет «нефонового» кластера, т. е. цвет груши.

3.3. Примеры задач заключительного этапа олимпиады

Мария собрала из разных источников много отзывов на фильмы. Каждый отзыв хранится в отдельном текстовом файле, в кодировке UTF-8. Кроме отзыва файл ничего не содержит (оценок в баллах там тоже нет).

Мария устала сортировать отзывы вручную и желает получить программу, которая будет:

1) вводить с клавиатуры имя текстового файла;
2) выводить в стандартный вывод тип отзыва: положительный, отрицательный или нейтральный. Результат надо вывести на экран одним словом: “positive”, “negative”, “neutral”.

Мария отсортировала 200 отзывов и на них проверит работу программы. Результат оценки 200 файлов будет сравниваться с эталоном.

Итоговая оценка программы вычисляется на основе метрики F1. Программа получит F1*100 баллов (т. е., если метрика F1 = 0.7356687, в рейтинг пойдет оценка 73.56687).

Для обучения модели можно использовать опубликованный Михаилом Клеминым набор данных: собранные отзывы с сайта «Кинопоиск» [23].

Не имея возможности в рамках статьи полностью разобрать решение этой задачи, указываются некоторые ее особенности.

Во-первых, это очень известная задача определения тональности текста. В сети или в любом курсе, посвященном обработке текста на естественном языке, эта задача рассматривается очень часто, поскольку имеет множество реальных применений: от социологического анализа до рекомендательных систем.

Тем не менее универсального метода ее решения не существует, и в каждом конкретном случае у нее есть свои особенности, от которых будет зависеть качество полученного результата. Именно это позволяет организовать соревнование между участниками.

Во-вторых, в приведенной нами постановке задачи есть несколько особенностей, которые нужно было учесть при выборе метода решения. Речь в задаче идет о трехклассовом анализе, т. е. нужно разделить не только полярные мнения, но и нейтральные. Тексты отзывов достаточно длинные (в среднем больше тысячи слов), и применение трансформеров, таким образом, требует заметной адаптации. При этом, что неочевидно, специфика задачи не дает возможности воспользоваться часто применяемыми средствами предварительной обработки текста: нельзя использовать обычный список стоп-слов (потому что, например, в него попадет слово «НЕ», влияющее как раз на категорию текста), нельзя выполнить лемматизацию (точнее, можно, но при этом точность снизится), и т. д.

В-третьих, объем предоставленного набора данных был достаточно велик, но модель для классификации должна была уместиться в архив объемом

ISSN 0234-0453 • ИНФОРМАТИКА И ОБРАЗОВАНИЕ • 2022 • Том 37 № 3

не более 100 Мб. Таким образом, «в лоб» задача решалась с не очень высоким качеством, а для повышения качества требовала понимания применяемых средств.

Упрощенно схема одного из возможных решений может быть описана таким образом:

1. Считать все тексты, представить как набор признаков (слов).
2. Перевести данные в векторное представление (использовался метод TF-IDF, как наиболее частый).
3. Выполнить обучение модели-классификатора этих векторов (наилучшие результаты показал классификатор на базе линейных опорных векторов).

Приведем основной фрагмент одного из вариантов подготовки такой модели в виде последовательности обработки:

```
from sklearn.model_selection import
  train_test_split
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text
  import TfidfVectorizer
from sklearn.metrics import confusion_matrix
from sklearn.svm import LinearSVC
from sklearn.metrics import accuracy_score,
  f1_score
import os, re, pickle
stopWords = {} # Загрузка словаря-фильтра,
  сформированного для сокращения модели
with open('wordGain.txt','r',encoding='utf-8')
  as f:
for line in f:
word=line.lower().strip()
stopWords[word] = 1
wordSplit = re.compile(r"[А-ЯЁа-яё]+") # Регу-
  лярное выражение для выделения слов
def preprocess_file( filename,enc):
words = ''
for string in open(filename,'r',encoding=enc):
  # Формирование "отрицательных слов"
string = string.replace(' не ',' не_')
string = string.replace(' ни ',' ни_')
string = string.replace(' и т. д. ',' и_т_д ')
string = string.replace(' и т. п. ',' и_т_п ')
for word in re.findall(wordSplit,string):
if word.lower() in stopWords:
words = words + ' '+word
return words.strip()
basePath =/ИИ/text processing/dataset/'# Ката-
  лог с отзывами
paths = ['pos','neg','neu']
corpus = []
target = []
count = 1
for path in paths: # Обработка каталогов
  с отзывами
for file in os.listdir(basePath+path): # Пере-
  бор файлов в каталоге
corpus.append(preprocess_file(basePath+path+'\\
  '+file,"utf-8"))
target.append(path)
print (count,file,path)
count += 1
docs_train, docs_test, class_train,
  class_test = train_test_split(corpus, target,
  test_size=0.20, random_state=None)# Разделе-
  ние набора на обучающую и тестовую выборки
textClasser = Pipeline([
('vect',TfidfVectorizer(min_df=2, max_df=0.95,
  ngram_range=(1,2), lowercase=False,
  stop_words=['а','в','у','и','то'])),
('clf', LinearSVC(class_weight='balanced'))
])
textClasser.fit(docs_train,class_train)

with open('sentiment-model.bf','wb') as f:
  # Сохранение модели
pickle.dump(textClasser,f)
# Оценка результатов на тестовом наборе
predicted = textClasser.predict(docs_test)
print(accuracy_score(class_test,predicted))
print(confusion_matrix(class_test,predicted,
  labels=['pos','neg','neu']))
print(f1_score(class_test,predicted,labels=
  ['pos','neg','neu'],average='weighted'))
```

Фактически в задаче требовалось определить наиболее эффективные методы классификации и их параметры, а потом разработать способ «ужатия» модели, не потеряв значительно в качестве ее работы. Решать эти задачи можно было по-разному: можно было воспользоваться знанием сравнительных исследований на эту тему и приведенными в них приемами, можно было провести серию вычислительных экспериментов — если позволяло оборудование и подготовка.

Как и в предыдущих задачах, решение надо было отправить на проверку в виде стартового файла на Python и необходимых для его работы файлов (в приведенном примере — файла модели и списка слов для фильтра). Именно этому файлу и передавались для оценки файлы контрольного набора, как и описано в условии задачи. Обучение модели к этой задаче во время проверки на третьем этапе не предусматривалось (объем данных для этого был слишком велик).

По условию второй задачи участникам необходимо было написать алгоритм подсчета «количества звезд» на фотографиях, учитывая, что на таких фотографиях могут быть недостоверные источники света, такие как фонарь, луна (или свет фар автомобиля). При оценке результатов решения этой задачи оценивалось отклонение от заранее подсчитанного числа звезд, и на каждом тесте можно было набрать до десяти очков.

Сумма накопленных баллов при решении заданий заключительного этапа позволяет получить двести баллов — по сто за каждое задание, соответственно, топ-3 участников, набравших максимальное количество баллов, занимают призовые места в соответствии со своей позицией в образовавшемся рейтинге.

Кроме того, следует отметить, что особое внимание заслужили участники, набравшие максимальное количество баллов за решение отдельной задачи, как первой, так и второй. Это показывает, что каждый из них был сосредоточен на максимальном качестве разрабатываемой модели для решения заданий.

Всего в заключительный этап олимпиады прошли пятьдесят человек, из них сорок семь просто приняли участие, а двадцать пять смогли справиться хотя бы с одной из поставленных задач.

4. Заключение

Заключительный этап олимпиады стал экспериментом и для участников, вовлеченных в решение задач, и для организаторов — с точки зрения разработки нетривиальных заданий, формат которых ранее не использовался в аналогичных мероприятиях.

Существующие олимпиады по направлению искусственного интеллекта предлагают участникам разработку решения с использованием готовой модели данных, которую необходимо «научить» решать поставленную задачу. Однако рассматриваемая Всероссийская олимпиада по искусственному интеллекту дала возможность участникам поучаствовать в эксперименте по созданию собственных моделей с нуля, обучая их на предоставленных или самостоятельно сформированных наборах данных в условиях ограниченных ресурсов. Школьники, основываясь на существующих решениях, как правило, предлагаемых крупными ИТ-компаниями, разработали свои решения. Инженерных прорывов не ожидалось и не обнаружилось. Но это самостоятельные осознанные решения школьников, увлеченных темой искусственного интеллекта, которые продолжат свое профессиональное обучение в ИТ-сфере.

Приведенные примеры заданий каждого этапа олимпиады были разобраны разработчиками на обучающих вебинарах, которые проводились после окончания каждого этапа.

Список источников / References

1. Программа «Цифровая экономика Российской Федерации»: утверждена распоряжением Правительства Российской Федерации от 28.07.2017 г. № 1632-р. Режим доступа: http://static.government.ru/media/files/9gFM4FHj4PsB79I5v7yLVuPgu4bvR7M0.pdf

[The program “Digital Economy of the Russian Federation”: approved by the Decree of the Government of the Russian Federation No. 1632-r dated July 28 2017. Available at: http://static.government.ru/media/files/9gFM4FHj4PsB79I5v7yLVuPgu4bvR7M0.pdf]

2. *Yang D., Shi B., Samylkin A.* Graphical Neural Networks for the Global Economy with Microsoft DeepGraph. *WSDM 22: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. NY, Association for Computing Machinery; 2022:1655. DOI: 10.1145/3488560.3510020

3. *Machalica M., Samylkin A., Port M., Chandra S.* Predictive Test Selection. *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*. 2019:91–100. DOI: 10.1109/ICSE-SEIP.2019.00018

4. *Рассел С., Норвиг П.* Искусственный интеллект: современный подход: 2-е изд. М.: Вильямс; 2007. 1408 с.

[*Russell S., Norvig P.* Artificial Intelligence: a modern approach: 2nd ed. Moscow, Williams; 2007. 1408 p. (In Russian.)]

5. *Брусард М.* Искусственный интеллект: пределы возможного. М.: Альпина нон-фикшн; 2020. 362 с.

[*Broussard M.* Artificial intelligence: the limits of the possible. Trans. from English. Moscow, Alpina non-fiction; 2020. 362 p. (In Russian.)]

6. *Джарратано Д., Райли Г.* Экспертные системы: принципы разработки и программирование: 4-е изд. М.: Вильямс; 2007. 1152 с.

[*Giarratano D., Riley G.* Expert systems: principles of development and programming: 4th ed. Moscow, Williams; 2007. 1152 p. (In Russian.)]

7. *Калинин И. А., Самылкина Н. Н.* Информатика. Углубленный уровень. 11 класс. М.: БИНОМ. Лаборатория знаний; 2013. 216 с.

[*Kalinin I. A., Samylkina N. N.* Informatics. Advanced level. 11th grade. Moscow, BINOM. Laboratory of Knowledge; 2013. 216 p. (In Russian.)]

8. *Ясницкий Л. Н.* Введение в искусственный интеллект. М.: Академия; 2008. 176 с.

[*Yasnitskiy L. N.* Introduction to artificial intelligence. Moscow, Academy; 2008. 176 p. (In Russian.)]

9. *Ясницкий, Л. Н.* Искусственный интеллект. Элективный курс: учеб. пособие. М.: БИНОМ. Лаборатория знаний; 2012. 197 с.

[*Yasnitsky, L. N.* Artificial intelligence. Elective course: textbook. Moscow: BINOM. Laboratory of knowledge; 2012. 197 p. (In Russian.)]

10. *Самылкина Н. Н., Салахова А. А.* Обучение основам искусственного интеллекта и анализа данных в курсе информатики на уровне среднего общего образования: монография. Москва: МПГУ; 2022. 228 с. DOI: 10.31862/9785426310643

[*Samylkina N. N., Salakhova A. A.* Teaching the basics of artificial intelligence and data analysis in an informatics course at the level of secondary general education: monograph. Moscow, MPGU; 2022. 228 p. (In Russian.) DOI: 10.31862/9785426310643]

11. *Stoyanov S., Glushkova T., Papancheva R.* Source intellect. Knowledge representation through logic. Bourgas, Logic Programming LLC Art Publishing House; 2021. 248 p.

12. *Джоши П.* Искусственный интеллект с примерами на Python. СПб.: Диалектика; 2019. 448 с.

[Joshi P. Artificial Intelligence with examples in Python. St. Petersburg, Dialectics; 2019. 448 p. (In Russian.)]

13. *Левченко И. В.* Основные подходы к обучению элементам искусственного интеллекта в школьном курсе информатики. *Информатика и образование*. 2019;34(6):7–15. DOI: 10.32517/0234-0453-2019-34-6-7-15

[*Levchenko I. V.* Basic approaches to teaching elements of artificial intelligence in the school course of informatics. *Informatics and Education*. 2019;34(6):7–15. (In Russian.) DOI: 10.32517/0234-0453-2019-34-6-7-15]

14. *Богданова А. Н.* Элективный курс «Основы искусственного интеллекта» для учащихся старших классов. *Информатика в школе*. 2021;20(7):27–33. DOI: 10.32517/2221-1993-2021-20-7-27-33

[*Bogdanova A. N.* The elective course “basics of artificial intelligence” for high school students. *Informatics in School*. 2021;20(7):27–33. (In Russian.) DOI: 10.32517/2221-1993-2021-20-7-27-33]

15. *Stoyanov S., Glushkova T., Papancheva R.* Artificial Intelligence. Problem solving through search. Bourgas, Logic Programming LLC Art Publishing House; 2019. 312 p.

16. Примерная основная образовательная программа основного общего и среднего общего образования. Режим доступа: https://fgosreestr.ru/

[Approximate basic educational program of basic general and secondary general education. (In Russian.) Available at: https://fgosreestr.ru/]

17. *Agrawal R., Srikant R.* Fast algorithms for mining association rules in large databases. *Proc. of the 20th Int. Conf. on Very Large Data Bases (VLDB). Santiago, Chile.* 1994:487–499. Available at: https://vldb.org/conf/1994/P487.PDF

18. *Пиковер К.* Искусственный интеллект. Иллюстрированная история. От автоматов до нейросетей. М.: Синдбад; 2021. 250 с.

[*Pickover K.* Artificial intelligence. Illustrated history. From automata to neural networks. Moscow, Sindbad; 2021. 250 p. (In Russian.)]

ISSN 0234-0453 • ИНФОРМАТИКА И ОБРАЗОВАНИЕ • 2022 • Том 37 № 3

19. Положение о всероссийской олимпиаде по искусственному интеллекту для обучающихся общеобразовательных организаций, утверждено протоколом оргкомитета № ТВ-47/04пр от 25.10.2021г. Режим доступа: https://edu.prosv.ru/pl/fileservice/user/file/download/h/70b99a32d3bfbb8526cf4a99b1c54ad6.pdf

[The regulations on the All-Russian Olympiad in Artificial Intelligence for students of general education organizations, approved by the protocol of the organizing committee No. TV-47/04pr dated October 25 2021. (In Russian.) Available at: https://edu.prosv.ru/pl/fileservice/user/file/download/h/70b99a32d3bfbb8526cf4a99b1c54ad6.pdf]

20. *Фридл Дж.* Регулярные выражения. СПб.: Символ-Плюс; 2008. 608 с.

[*Friedl J.* Regular Expressions. St. Petersburg, Symbol-Plus; 2008. 608 p. (In Russian.)]

21. *Толстой Л. Н.* Война и мир. Т 1, 2. Режим доступа: https://avidreaders.ru/book/voyna-i-mir-toma-1-i.html

[Tolstoy L. N. War and Peace. V 1, 2. (In Russian.) Available at: https://avidreaders.ru/book/voyna-i-mir-toma-1-i.html]

22. Национальный корпус русского языка. Режим доступа: https://ruscorpora.ru/new/corpora-freq.html

[The National Corpus of the Russian Language. (In Russian.) Available at: https://ruscorpora.ru/new/corpora-freq.html]

23. Kinopoisk's movies reviews. Available at: https://www.kaggle.com/mikhailklemin/kinopoisks-movies-reviews

Информация об авторах

Григорьев Сергей Георгиевич, член-корреспондент РАО, доктор тех. наук, профессор, профессор департамента информатики, управления и технологий, Институт цифрового образования, Московский городской педагогический университет, г. Москва, Россия; *ORCID:* https://orcid.org/-0000-0002-0034-9224; *e-mail:* grigorsg@yandex.ru

Калинин Илья Александрович, канд. пед. наук, доцент, начальник управления информатизации, доцент кафедры международной информационной безопасности, Институт информационных наук, Московский государственный лингвистический университет, г. Москва, Россия; *ORCID:* https://orcid.org/0000-0001-6710-8188; *e-mail:* kalininIlya@mail.ru

Самылкина Надежда Николаевна, доктор пед. наук, доцент, профессор кафедры теории и методики обучения математике и информатике, Институт математики и информатики, Московский педагогический государственный университет, г. Москва, Россия; *ORCID:* https://orcid.org/0000-0003-0797-5532; *e-mail:* nsamylkina@yandex.ru

Information about the authors

Sergey G. Grigoriev, Corresponding Member of RAE, Doctor of Sciences (Engineering), Professor, Professor at the Department of IT, Management and Technologies, Institute of Digital Education, Moscow City University, Moscow, Russia; *ORCID:* https://orcid.org/0000-0002-0034-9224; *e-mail:* grigorsg@yandex.ru

Ilya A. Kalinin, Candidate of Sciences (Education), Docent, Head of the Informatization Department, Associate Professor at the International Information Security Department, Institute of Information Sciences, Moscow State Linguistic University, Moscow, Russia; *ORCID:* https://orcid.org/0000-0001-6710-8188; *e-mail:* kalininIlya@mail.ru

Nadezhda N. Samylkina, Doctor of Sciences (Education), Docent, Professor at the Department of Theory and Methods of Teaching Mathematics and Informatics, Institute of Mathematics and Informatics, Moscow Pedagogical State University, Moscow, Russia; *ORCID:* https://orcid.org/0000-0003-0797-5532; *e-mail:* nsamylkina@yandex.ru

Поступила в редакцию / Received: 03.04.2022.
Поступила после рецензирования / Revised: 25.04.2022.
Принята к печати / Accepted: 26.04.2022.