МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Методические указания по выполнению лабораторных работ

по дисциплине «Разработка диалоговых систем»

для студентов направления подготовки
09.03.02 «Информационные системы и технологии»
Профиль «Прикладное программирование в интеллектуальных информационных системах»

Ставрополь, 2026

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

1. Цели и задачи освоения дисциплины. Цель изучения дисциплины: формирование у студентов системы теоретических представлений о современных подходах к построению диалоговых систем на основе моделей машинного обучния.

Задачами дисциплины «Разработка диалоговых систем» являются:

1) изучение теоретических принципов обработки естественного языка;

2) исследование эффективности языковых моделей;

3) изучение типологии языковых моделей;

4) получение практических навыков по применению интеллектуальных моделей обработки естественного языка для построения диалоговых систем;

5) получение практических навыков по проектированию приложений на основе моделей NLP.

Пособие предназначено для студентов, обладающих теоретическими знаниями в области проектирования приложений и практическими навыками программирования (предпочтительно языки C, C++, C#, Python, R). Цель метдических указаний: сформировать у студентов целостный взгляд на современные тенденции в области искусственных нейронных сетей; сформировать систему компетенций.

В пособии рассмотрены практические аспекты проектирования и разработки диалоговых систем

ЛАБОРАТОРНАЯ РАБОТА 1. НАСТРОЙКА СРЕДЫ РАЗРАБОТКИ

Цели и задачи

Цель лабораторной работы: изучение программных средств для организации рабочего места разработчика систем обработки естественного языка.

Основные задачи:

– установка и настройка среды разработки Python;

– изучение принципов токенизации;

– загрузка и тестирование предобученной модели.

Теоретическое обоснование

В данной работе необходимо реализовать следующие этапы:

- установка Transformer с Anaconda;
- работа с языковыми моделями и токенизаторами;
- работа с моделями, предоставленными сообществом;
- работа с тестами и наборами данных;
- измерение быстродействия и потребления памяти.

Оборудование и материалы

Для выполнения лабораторной работы рекомендуется использовать персональный компьютер со следующими программными средствами разработки: Библиотека Transformers, TensorFlow и PyTorch, Python 3.6x, Jupyter Notebook, Google Colaboratory, Docker, Locust.io, Git.

Методика и порядок выполнения работы

1. Выполните настройку среды разработки в среде Google Colab.

После настройки необходимо проверить работоспособность модели следующим образом:

```
>>> from Transformer import BERTTokenizer
>>> tokenizer = BERTTokenizer.from_pretrained('BERT-base-uncased')
```

Первая строка этого фрагмента кода импортирует токенизатор BERT, а вторая строка загружает предварительно обученный токенизатор для базовой версии BERT. Обратите внимание, что версия, не учитывающая регистр (uncased), обучается с использованием букв без регистра, поэтому не имеет значения, отображаются ли буквы в верхнем или нижнем регистре. Чтобы проверить код и увидеть результат, вы должны ввести в окне интерпретатора Python следующие строки:

```
>>> text = "Using Transformer is easy!"
>>> tokenizer(text)
```

Вы получите следующий вывод:

{'input_ids': [101, 2478, 19081, 2003, 3733, 999, 102], 'token_type_ids': [0, 0, 0, 0, 0, 0, 0], 'attention_mask': [1, 1, 1,1, 1, 1, 1]}

2. Токенизатор можно использовать для моделей Transformer как на основе PyTorch, так и на основе TensorFlow. Чтобы выводить данные для каждой из них, в return_tensors необходимо использовать ключевые слова pt и tf. Например, вы можете использовать токенизатор, просто выполнив следующую команду:

```
>>> encoded_input = tokenizer(text, return_tensors="pt")
```

encoded_input содержит токенизированный текст, который предназначен для модели PyTorch. Чтобы запустить модель, например базовую модель BERT, можно использовать следующий код для загрузки модели из репозитория моделей huggingface:

```
>>> from Transformer import BERTModel
```

```
>>> model = BERTModel.from_pretrained("BERT-base-uncased")
```

Выходные данные токенизатора можно передать загруженной модели с помощью следующей строки кода:

```
>>> output = model(**encoded_input)
```

Мы получим результат работы модели в виде векторных представлений и выходов перекрестного внимания.

3. В данной работе необходимо настроить среду разработки, загрузить и использовать предварительно обученную модель BERT, и понять основы токенизации, а также проанализировать разницу между версиями моделей PyTorch и TensorFlow.

Содержание отчета и его форма

Отчет по лабораторной работе должен содержать:

1. Номер и название лабораторной работы; задачи лабораторной работы.

2. Реализация каждого пункта подраздела «Методика и порядок выполнения работы» с приведением исходного кода программы, диаграмм и графиков для визуализации данных.

3. Ответы на контрольные вопросы.

4. Экранные формы (консольный вывод) и листинг программного кода с комментариями, показывающие порядок выполнения лабораторной работы, и результаты, полученные в ходе её выполнения.

Отчет о выполнении лабораторной работы подписывается студентом и сдается преподавателю.

Контрольные вопросы

1. Пусть имеется функция XOR, в которой две точки {(0, 0), (1, 1)} принадлежат к одному классу, а две другие точки {(1, 0), (0, 1)} – к другому. Покажите, как разделить два этих класса, используя функцию активации ReLU

2. Покажите, что для функции активации в виде сигмоиды и гиперболического тангенса (обозначенной как Ф(·) в каждом из этих случаев) справедливы приведенные ниже равенства:

А. Сигмоида: Ф(-v)=l-Ф(v).

Б. Гиперболический тангенс: Ф(-v) = -Ф (v).

В. Спрямленный гиперболический тангенс: Ф(-v) = -Ф (v).

3. Покажите, что гиперболический тангенс – это сигмоида, масштабированная вдоль горизонтальной и вертикальной осей и смещенная вдоль вертикальной оси:

$$\tanh(v) = 2 \cdot \text{sigmoid}(2v) - 1.$$

4. Пусть имеется набор данных, в котором две точки {(-1, -1), (1, 1)} принадлежат к одному классу, а две другие точки {(1, -1), (-1, 1)} - к другому. Начните со значений параметра перцептрона (0, 0) и выполните несколько обновлений по методу стохастического градиентного спуска с $\alpha = 1$. В процессе получения обновленных значений циклически обходите тренировочные точки в любом порядке.

А. Сходится ли алгоритм в том смысле, что изменения значений целевой функции со временем становятся предельно малыми?

Б. Объясните, почему наблюдается ситуация, описанная в п. А.

5. Определите для набора данных из упражнения 4, в котором двумя признаками являются (x_1, x_2), новое одномерное представление z, обозначаемое как:

$$z = x_1 * x_2$$

Является ли набор данных линейно разделимым в терминах одномерного представления, соответствующего z? Объясните важность нелинейных преобразований в задачах классификации.

6. Покажите, что значение производной сигмоиды не превышает 0,25, независимо от значения аргумента. При каком значении аргумента сигмоида имеет максимальное значение?

7. Покажите, что значение производной функции активации tanh не превышает 1, независимо от значения аргумента. При каком значении аргумента функция активации tanh имеет максимальное значение?

8. Пусть имеется сеть с двумя входами, x_1 и x_2. Сеть имеет два слоя, каждый из которых содержит два элемента. Предположим, что веса в каждом слое устанавливаются таким образом, что в верхнем элементе каждого слоя для суммирования его входов применяется сигмоида, а в нижнем – функция активации tanh. Наконец, в единственном выходном узле для суммирования двух его входов применяется активация ReLU. Запишите выход этой нейронной сети в замкнутой форме в виде функции x_1 и x_2.

9. Вычислите частную производную по x_1 функции в замкнутой форме, полученной в предыдущем упражнении. Практично ли вычислять производные для градиентного спуска в нейронных сетях, используя выражения в замкнутой форме (как в традиционном машинном обучении)?

10. Пусть имеется двухмерный набор данных, в котором все точки с $x_1 > x_2$ принадлежат к положительному классу, а все точки с $x_1 <= x_2$ – к отрицательному. Истинным разделителем для этих двух классов является линейная гиперплоскость (прямая линия), определяемая уравнением $x_1 - x_2 = 0$. Создайте набор тренировочных данных с 20 точками, сгенерированными случайным образом в положительном квадранте единичного квадрата. Снабдите каждую точку меткой, указывающей на то, превышает или не превышает ее первая координата x_1 вторую координату x_2.

А. Реализуйте алгоритм перцептрона без регуляризации, обучите его на полученных выше 20 точках и протестируйте его точность на 1000 точках, случайно сгенерированных в единичном квадрате. Используйте для генерирования тестовых точек ту же процедуру, что и для тренировочных.

Б. Замените критерий перцептрона на кусочно-линейную функцию потерь в своей реализации тренировки и повторите определение точности вычислений на тех же тестовых точках, которые использовали перед этим. Регуляризация не используется.

В. В каком случае и почему вам удалось получить лучшую точность?

Г. Как вы считаете, в каком случае классификация тех же 1000 тестовых точек не изменится значительно, если использовать другой набор из 20 тренировочных точек?

Список литературы

Для выполнения лабораторной работы, при подготовке к защите, а также для ответа на контрольные вопросы рекомендуется использовать следующие источники: [1-3, 9, 10].

ЛАБОРАТОРНАЯ РАБОТА 2. ТРАНСФОРМЕРЫ

Цели и задачи

Цель лабораторной работы: разработка на основе предобученной модели BERT.

Основные задачи:

– загрузка модели;

–загрузка данных;

– обучение модели;

– сохранение конфигурации.

Теоретическое обоснование

Токенизаторы – одна из самых важных частей многих приложений NLP. Для BERT применяется токенизация WordPiece. Стоит отметить, что WordPiece, SentencePiece и BytePairEncoding (BPE) являются тремя наиболее широко известными токенизаторами, используемыми в различных архитектурах на основе трансформеров. Мы еще вернемся к ним в следующих разделах. Основная причина того, что BERT или любая другая трансформерная архитектура использует токенизацию частей слов, – это способность таких токенизаторов работать с неизвестными токенами.

BERT также использует позиционное кодирование, чтобы гарантировать, что модель учитывает положение токенов. BERT и подобные модели не учитывают порядок входной последовательности. Обычные модели, такие как LSTM и RNN, получают позицию в соответствии с порядком токенов в последовательности. Чтобы донести эту дополнительную информацию в BERT, применяется позиционное кодирование.

Например, в случае задачи классификации последовательности, такой как анализ тональности или классификация предложений, в оригинальной статье про BERT предлагается использовать представление [CLS] из

последнего уровня. Однако есть другие исследования, в которых проводят классификацию при помощи различных методов с использованием BERT (с использованием усредненного представления токенов на основе всех токенов, развертывания LSTM на последнем уровне или даже использования CNN поверх последнего уровня). Представление [CLS] на последнем уровне для классификации последовательностей может использоваться любым классификатором, но чаще всего он представляет собой плотный слой с размером ввода, равным размеру представления окончательного токена, и размером вывода, равным количеству классов с функцией активации Softmax. Альтернативным вариантом является использование сигмоидной функции, когда вывод может быть многозначным, а сама задача заключается в классификации с несколькими метками.

На рис. 2.1 представлен пример задачи NSP:

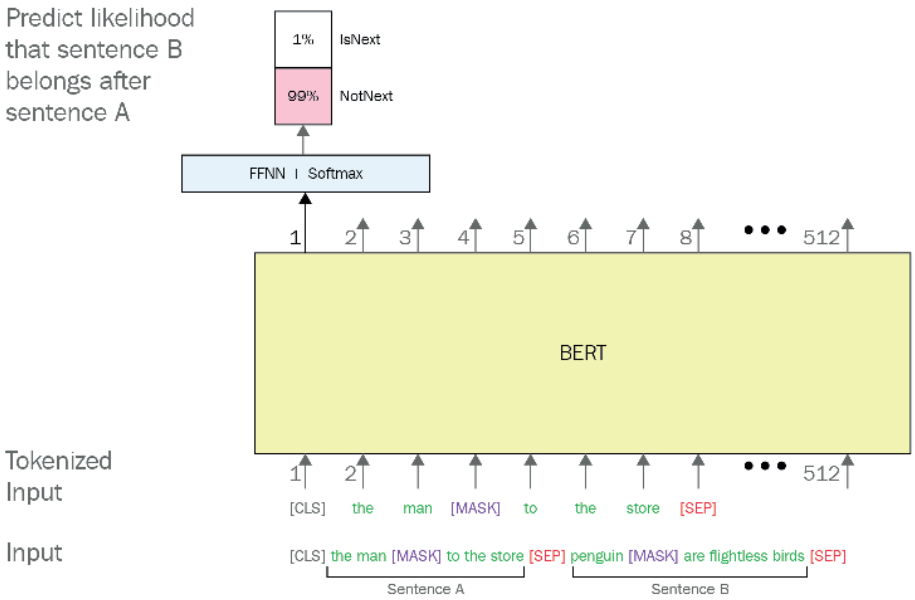
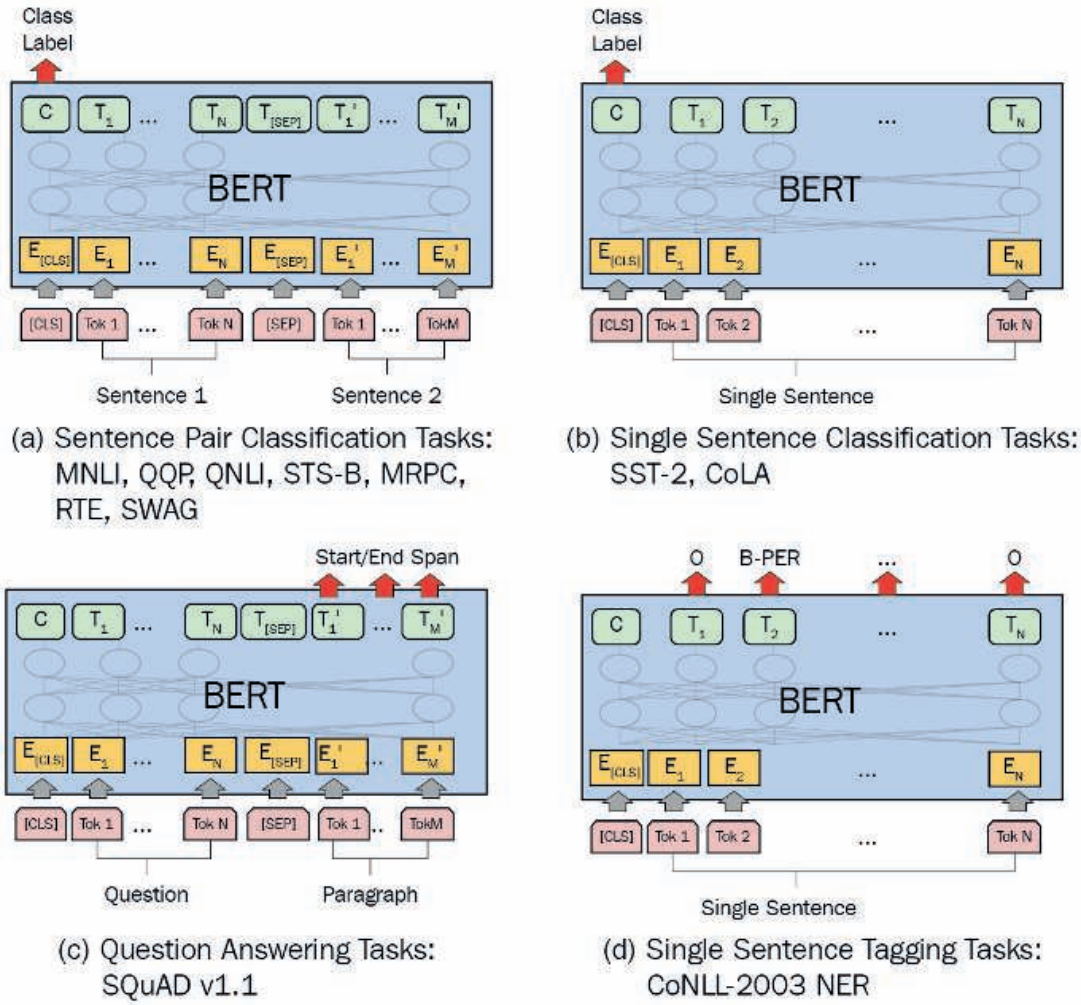


Рисунок 2.1 – Пример модели BERT для задачи NSP

Модель BERT встречается в различных вариантах с разными настройками. Например, можно менять размер входных данных. В предыдущем примере задано значение 512, и, следовательно, максимальный размер последовательности, которую модель может получить в качестве входных данных, равна 512. Однако в последовательность входят специальные токены [CLS] и [SEP], поэтому фактический размер данных

уменьшается до 510. С другой стороны, использование WordPiece в качестве токенизатора дает на выходе токены частей слов, и после токенизации размер увеличится, потому что токенизатор разбивает слова на части (как правило, если они не встречались в корпусе предварительного обучения).

На рис. 2.2 представлены схемы модели BERT для различных задач. Для задачи NER вместо [CLS] используется вывод каждого токена. Если стоит задача ответа на вопрос, то вопрос и ответ объединяются при помощи токена-разделителя [SEP], а ответ аннотируется при помощи данных Начало/Конец и Промежуток с последнего уровня. В данном случае Абзац – это контекст, в котором задается Вопрос.



(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

(b) Single Sentence Classification Tasks: SST-2, CoLA

(c) Question Answering Tasks: SQuAD v1.1

(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

Рисунок 2.2 – Модель BERT для различных задач NLP

Независимо от всех этих задач, наиболее важной способностью BERT является контекстное представление текста. Причина, по которой эта модель

столь успешна в различных задачах, заключается в архитектуре энкодера Transformer, который представляет входные данные в виде плотных векторов. Эти векторы можно легко преобразовать в выходные данные с помощью очень простых классификаторов.

Оборудование и материалы

Для выполнения лабораторной работы рекомендуется использовать персональный компьютер со следующими программными средствами разработки: Библиотека Transformers, TensorFlow и PyTorch, Python 3.6x, Jupyter Notebook, Google Colaboratory, Docker, Locust.io, Git.

Методика и порядок выполнения работы

Набор данных IMDB из 50 тыс. обзоров фильмов (доступен по адресу https://www.kaggle.com/lakshmi25npathi/sentiment-analysis-of-imdb-movie-reviews) – это достаточно большой набор данных для анализа тональности, но небольшой, если используете его в качестве корпуса для обучения вашей языковой модели.

1. Загрузить набор данных и сохранить его в формате .txt для обучения языковой модели и токенизатора, используя следующий код:

```
import pandas as pd
imdb_df = pd.read_csv("IMDB Dataset.csv")
reviews = imdb_df.review.to_string(index=None)
with open("corpus.txt", "w") as f:
    f.writelines(reviews)
```

2. После подготовки корпуса необходимо обучить токенизатор. Библиотека tokenizers обеспечивает быстрое и простое обучение токенизатора WordPiece. Чтобы обучить его на своем корпусе, необходимо запустить следующий код:

```
>>> from tokenizers import BertWordPieceTokenizer
```

>>> bert_wordpiece_tokenizer =BertWordPieceTokenizer()

>>> bert_wordpiece_tokenizer.train("corpus.txt")

3. Теперь есть обученный токенизатор. Вы можете получить доступ к обученному словарю с помощью функции get_vocab() обученного объекта tokenizer. Вы можете извлечь вокабуляр (словарный запас) при помощи следующей строки кода:

>>> bert_wordpiece_tokenizer.get_vocab()

Так выглядит результат ее выполнения:

{'almod': 9111, 'events': 3710, 'bogart': 7647, 'slapstick': 9541, 'terrorist': 16811, 'patter': 9269, '183': 16482, '##cul': 14292, 'sophie': 13109, 'thinki': 10265, 'tarnish': 16310, '##outh': 14729, 'peckinpah': 17156, 'gw': 6157, '##cat': 14290, '##eing': 14256, 'successfully': 12747, 'roomm': 7363, 'stalwart': 13347,...}

4. Важно сохранить токенизатор для дальнейшего использования. Используя функцию save_model() объекта и указав ей каталог, сохраните словарь токенизатора для дальнейшего использования:

>>> bert_wordpiece_tokenizer.save_model("tokenizer")

5. Теперь вы можете повторно загрузить словарь, используя функцию from_file():

>>> tokenizer = \ BertWordPieceTokenizer.from_file("tokenizer/vocab.txt")

6. Вы можете использовать токенизатор, как в этом примере кода:

>>> tokenized_sentence = \tokenizer.encode("Oh it works just fine")

>>> tokenized_sentence.tokens['[CLS]', 'oh', 'it', 'works', 'just', 'fine','[SEP]']

Специальные токены [CLS] и [SEP] будут автоматически добавлены в список токенов, потому что они нужны BERT для обработки ввода.

7. Давайте попробуем токенизировать другое предложение, используя наш токенизатор:

>>> tokenized_sentence = \tokenizer.encode("ohoh i thougt it might be working well")['[CLS]', 'oh', '##o', '##h', 'i', 'thoug', '##t', 'it', 'might', 'be', 'working', '##g', 'well', '[SEP]']

8. Похоже, у вас получился хороший токенизатор для зашумленного текста с ошибками. Теперь, когда у вас есть готовый и сохраненный токенизатор, вы можете обучить свою собственную модель BERT. Первый шаг – использовать BertTokenizerFast из библиотеки Transformers. Вам необходимо загрузить обученный токенизатор из предыдущего шага, используя следующие строки кода:

```
>>> from Transformers import BertTokenizerFast
>>> tokenizer = \ BertTokenizerFast.from_pretrained("tokenizer")
```

Мы использовали BertTokenizerFast, потому что это рекомендовано в документации библиотеки HuggingFace. Также есть BertTokenizer, который, согласно определению из документации библиотеки, реализуется в коде не так быстро, как быстрая версия. В большинстве документации и карточек предварительно обученных моделей настоятельно рекомендуется использовать версию BertTokenizerFast.

9. На следующем шаге мы подготовим корпус для ускоренного обучения с помощью следующей команды:

```
>>> from Transformers import LineByLineTextDataset
>>> dataset = \LineByLineTextDataset(tokenizer=tokenizer,
file_path="corpus.txt",
block_size=128)
```

10. Также необходимо предоставить систематизатор данных (data collator) для маскированного моделирования языка:

```
>>> from Transformers import DataCollatorForLanguageModeling
>>> data_collator = DataCollatorForLanguageModeling(
tokenizer=tokenizer,
mlm=True,
mlm_probability=0.15)
```

Подборщик данных получает данные и готовит их к обучению. Например, указанный выше подборщик берет данные и подготавливает их для маскированного моделирования языка с вероятностью 0,15. Цель

использования такого механизма – выполнять предварительную обработку на лету, что позволяет использовать меньше ресурсов. С другой стороны, это замедляет процесс обучения, потому что каждая выборка должна быть предварительно обработана на лету во время обучения.

11. На этапе обучения мы предоставляем информацию для тренера, которую можно задать в виде аргументов обучения с помощью следующего кода:

```
>>> from Transformers import TrainingArguments
>>> training_args = TrainingArguments(
output_dir="BERT",
overwrite_output_dir=True,
num_train_epochs=1,
per_device_train_batch_size=128)
```

12. Теперь мы создадим модель BERT, которую собираемся использовать с конфигурацией по умолчанию (количество итераций внимания, уровни энкодера Transformer и т. д.):

```
>>> from Transformers import BertConfig, BertForMaskedLM
>>> bert = BertForMaskedLM(BertConfig())
```

13. И последний подготовительный шаг – создание объекта тренера:

```
>>> from Transformers import Trainer
>>> trainer = Trainer(
model=bert,
args=training_args,
data_collator=data_collator,
train_dataset=dataset)
```

14. Наконец, вы можете обучить свою языковую модель при помощи следующей команды:

```
>>> trainer.train()
```

Она выведет на экран индикатор выполнения, отображающий ход обучения (рис. 2.3)

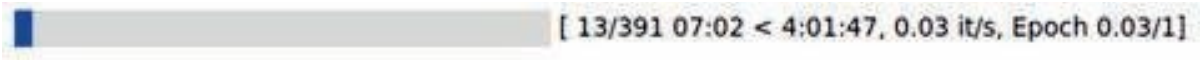


Рисунок 2.3 – Индикатор хода обучения модели BERT

15. После завершения обучения вы можете сохранить модель при помощи следующей команды:

```
>>> trainer.save_model("MyBERT")
```

К этому моменту вы узнали, как обучить BERT с нуля любому конкретному языку, который вам нужен. Вы узнали, как обучить токенизатор и модель BERT, используя подготовленный вами корпус.

16. Конфигурация по умолчанию, которую вы предоставили для BERT, является наиболее важной частью этого процесса обучения, определяющей архитектуру BERT и ее гиперпараметры. Вы можете взглянуть на эти параметры, используя следующий код:

```
>>> from Transformers import BertConfig
>>> BertConfig()
```

17. После выполнения пунктов 1-17 продемонстрируйте работоспособность модели, сохраните конфигурацию обучения.

Содержание отчета и его форма

Отчет по лабораторной работе должен содержать:

1. Номер и название лабораторной работы; задачи лабораторной работы.

2. Реализация каждого пункта подраздела «Методика и порядок выполнения работы» с приведением исходного кода программы, диаграмм и графиков для визуализации данных.

3. Ответы на контрольные вопросы.

4. Экранные формы (консольный вывод) и листинг программного кода с комментариями, показывающие порядок выполнения лабораторной работы, и результаты, полученные в ходе её выполнения.

Отчет о выполнении лабораторной работы подписывается студентом и сдается преподавателю.

Контрольные вопросы

1. Предположим, функция потерь для тренировочной пары $(\bar{X}, y)$ имеет следующий вид:

$$L = \max\{0, \alpha - y(\bar{W} \cdot \bar{X})\}.$$

Результаты тестовых примеров предсказываются как $\hat{y} = \{\bar{W} \cdot \bar{X}\}$. Значение $\alpha = 0$ соответствует критерию перцептрона, а значение $\alpha = 1$ – SVM. Покажите, что любое значение $\alpha > 0$ приводит к SVM с неизменным оптимальным решением, если регуляризация не задействуется. Что произойдет в случае использования регуляризации?

2. Основываясь на результатах упражнения 1, сформулируйте обобщенную целевую функцию для SVM Уэстона-Уоткинса.

3. Рассмотрим нерегуляризируемое обновление перцептрона для бинарных классов со скоростью обучения α. Покажите, что изменение значения α ни на что существенно не влияет в том смысле, что это приводит лишь к масштабированию вектора весов с коэффициентом α. Покажите, что этот результат остается справедливым также для мультиклассового случая. Останутся ли результаты такими же в случае использования регуляризации?

4. Покажите, что в случае применения SVM Уэстона-Уоткинса к набору данных с k = 2 классами результирующие обновления эквивалентны бинарным обновлениям SVM.

5. Покажите, что в случае применения мультиномиальной логистической регрессии к набору данных с k = 2 классами результирующие обновления эквивалентны обновлениям логистической регрессии.

6. Реализуйте классификатор Softmax, используя библиотеку глубокого обучения, с которой вы работаете.

7. В моделях ближайших соседей на основе линейной регрессии рейтинговая оценка объекта предсказывается в виде взвешенной комбинации оценок других объектов, оцененных тем же пользователем, где специфические для объектов веса обучаются с помощью линейной регрессии. Покажите,

какую архитектуру автокодировщика вы использовали бы для создания модели этого типа. Обсудите связь этой архитектуры с архитектурой матричной факторизации.

8. Логистическая матричная факторизация. Рассмотрим автокодировщик, в котором имеется входной слой, один скрытый слой, содержащий представление пониженной размерности, и выходной слой с линейной активацией.

А. Задайте функцию потерь на основе отрицательного логарифма правдоподобия для случая, когда известно, что матрица входных данных содержит бинарные значения из набора {0,1}.

Б. Задайте функцию потерь на основе отрицательного логарифма правдоподобия для случая, когда известно, что матрица входных данных содержит вещественные значения из интервала [0,1].

Список литературы

Для выполнения лабораторной работы, при подготовке к защите, а также для ответа на контрольные вопросы рекомендуется использовать следующие источники: [1-4].

ЛАБОРАТОРНАЯ РАБОТА 3. ЯЗЫКОВЫЕ МОДЕЛИ AR

Цели и задачи

Цель лабораторной работы: изучение принципов функционирования языковых моделей AR.

Теоретическое обоснование

Первоначально архитектура трансформеров была предназначена для решения задач Seq2Seq, таких как машинный перевод или резюмирование, но теперь она широко применяется в различных задачах NLP, начиная от классификации токенов и заканчивая разрешением кореферентности (coreference). Впоследствии в моделях начали использовать по отдельности и более творчески левую и правую части архитектуры. Общая цель, также известная как цель шумоподавления (denoising objective), заключается в том, чтобы полностью восстановить исходный вход из поврежденного входа в двунаправленном режиме, как показано в левой части рис. 3.1.

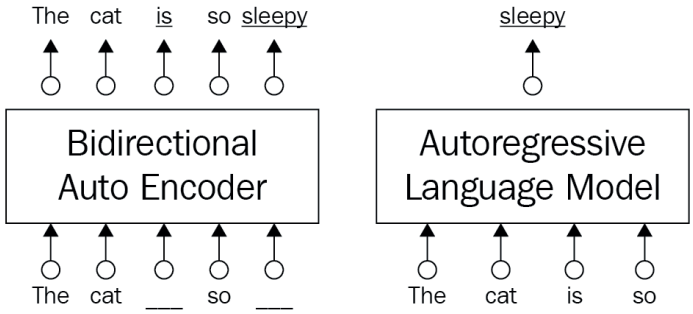


Рисунок 3.1 – Языковые модели AE и AR

Как видно из архитектуры BERT, которая является ярким примером автоэнкодерных моделей, они могут включать контекст, окружающий слово с обеих сторон. Однако первая проблема заключается в том, что искажающие символы [MASK], которые используются на этапе предварительного обучения, отсутствуют в данных на этапе тонкой настройки, что приводит к

несоответствию между предварительным обучением и тонкой настройкой. Во-вторых, модель BERT небезосновательно предполагает, что замаскированные токены независимы друг от друга.

В свою очередь, авторегрессивные модели избегают таких предположений относительно независимости и, естественно, не страдают от несоответствия предварительного обучения, поскольку они полагаются на цель предсказания следующего токена, обусловленного предыдущими токенами без маскировки. Они просто используют декодирующую часть трансформера с маскированным самовниманием. Они предотвращают доступ модели к словам справа от текущего слова в прямом направлении (или слева от текущего слова в обратном направлении), что называется однонаправленностью. Их также называют каузальными языковыми моделями (causal language models, CLM) из-за их однонаправленности. На рис. 3.1 показана разница между моделями AE и AR.

Модель GPT и две построенные на ее основе модели (GPT-2, GPT-3) – Transformer-XL и XLNet – являются одними из наиболее упоминаемых в литературе популярных моделей AR. Несмотря на то что модель XLNet основана на авторегрессии, ей каким-то образом удается использовать обе контекстные стороны слова двунаправленным образом с помощью языковой цели, основанной на перестановках. Далее вы познакомитесь с этими моделями поближе и узнаете, как обучать модели с помощью различных экспериментов. Начнем с моделей на основе GPT.

Левая часть рис. 3.2 (подготовленного по мотивам оригинальной статьи) иллюстрирует архитектуру трансформера и цели обучения, представленные в оригинальной статье про GPT. В правой части показано, как трансформировать ввод для тонкой настройки на несколько задач.

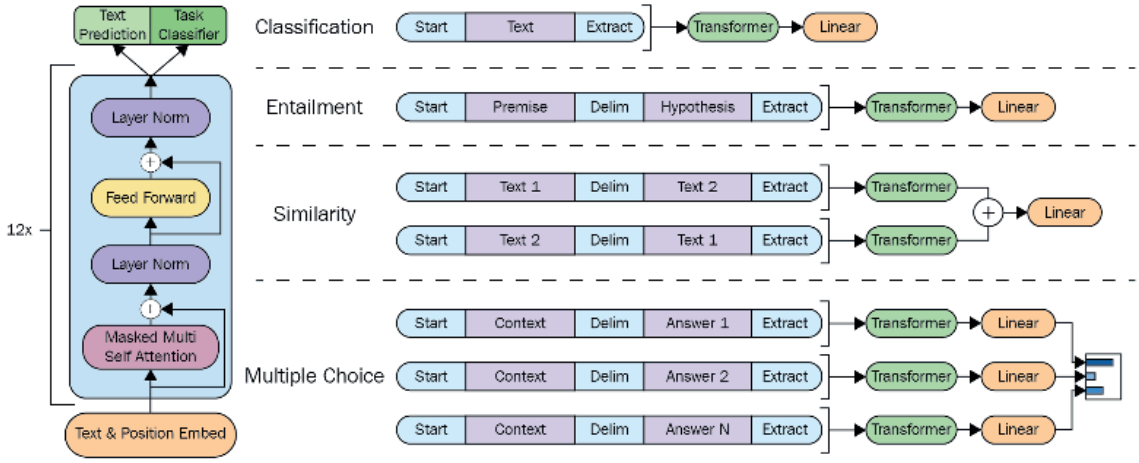


Рисунок 3.2 – Трансформация текстового ввода (на основе оригинальной статьи)

Оборудование и материалы

Для выполнения лабораторной работы рекомендуется использовать персональный компьютер со следующими программными средствами разработки: Библиотека Transformers, TensorFlow и PyTorch, Python 3.6x, Jupyter Notebook, Google Colaboratory, Docker, Locust.io, Git.

Методика и порядок выполнения работы

Требуется обучить собственные авторегрессивные языковые модели. Мы начнем с GPT-2 и более подробно рассмотрим его различные функции обучения, используя библиотеку transformers.

Вы можете выбрать любой текстовый корпус для обучения собственной модели GPT-2, но для этого примера мы использовали любовный роман «Эмма» Джейн Остин. Если вы хотите получить модель для генерации текста с более широким универсальным охватом, мы настоятельно рекомендуем выполнить обучение на гораздо более обширном корпусе.

1. Загрузите необработанный текст романа «Эмма», используя следующую команду:

```
wget https://raw.githubusercontent.com/teropa/nlp/master/resources/corpora/gutenberg/austen-emma.txt
```

2. Первым шагом является обучение токенизатора BytePairEncoding для GPT-2 на том же корпусе, на котором вы собираетесь в дальнейшем обучать свой GPT-2. Следующий код импортирует токенизатор BPE из библиотеки tokenizers:

```
from tokenizers.models import BPE
from tokenizers import Tokenizer
from tokenizers.decoders import ByteLevel as ByteLevelDecoder
from tokenizers.normalizers import Sequence, Lowercase
from tokenizers.pre_tokenizers import ByteLevel
from tokenizers.trainers import BpeTrainer
```

3. Как видите, в этом примере мы намерены обучить более продвинутый токенизатор, добавив дополнительные функции, такие как нормализация Lowercase (нижний регистр). Чтобы создать объект tokenizer, вы можете использовать следующий код:

```
tokenizer = Tokenizer(BPE())
tokenizer.normalizer = Sequence([Lowercase()])
tokenizer.pre_tokenizer = ByteLevel()
tokenizer.decoder = ByteLevelDecoder()
```

Первая строка создает токенизатор из класса токенизатора BPE. К части нормализации добавилась функция Lowercase, а для атрибута pre_tokenizer установлено значение ByteLevel, чтобы использовать байты в качестве входных данных. Атрибут decoder также должен иметь значение ByteLevelDecoder для правильного декодирования.

4. Обучите модель. Продемонстрируйте ее работоспособность.

Содержание отчета и его форма

Отчет по лабораторной работе должен содержать:

1. Номер и название лабораторной работы; задачи лабораторной работы.

2. Реализация каждого пункта подраздела «Методика и порядок выполнения работы» с приведением исходного кода программы, диаграмм и графиков для визуализации данных.

3. Ответы на контрольные вопросы.

4. Экранные формы (консольный вывод) и листинг программного кода с комментариями, показывающие порядок выполнения лабораторной работы, и результаты, полученные в ходе её выполнения.

Отчет о выполнении лабораторной работы подписывается студентом и сдается преподавателю.

Контрольные вопросы

1. Рассмотрим следующую рекурсию:

$$(x_{t+1}, y_{t+1}) = (f(x_t, y_t), g(x_t, y_t))$$

где f и g – функции многих переменных.

А. Представьте производную $\frac{\partial x_{i+2}}{\partial x_i}$ в виде выражения, включающего только x_t и y_t;

Б. Можете ли вы начертить архитектуру нейронной сети, соответствующей приведенной выше рекурсии, если *t* принимает значения в интервале от 1 to 5? Предположите, что нейроны могут вычислять любую требуемую функцию.

2. Рассмотрим нейрон с двумя входами, который перемножает входы x_1 и x_2 для получения выхода σ. Пусть L – функция потерь, вычисляемая в узле σ. Предположим, вам известно, что $\frac{\partial L}{\partial \sigma} = 5$, $x_1 = 2$, а $x_2 = 3$. Вычислите $\frac{\partial L}{\partial x_1}$ и $\frac{\partial L}{\partial x_2}$.

3. Рассмотрим нейронную сеть с двумя слоями, включая входной слой. Первый (входной) слой содержит четыре входа: x_1, x_2, x_3 и x_4. Второй слой имеет шесть скрытых элементов, соответствующих всем операциям попарного умножения. Выходной узел σ просто суммирует значения шести скрытых

элементов. Пусть L – функция потерь выходного узла. Предположим, вам известно, что $\frac{\partial L}{\partial \sigma} = 2,\ x_1 = 1,\ x_2 = 2,\ x_3 = 3,\ x_4 = 4$. Вычислите $\frac{\partial L}{\partial x_i}$ для каждого i.ё

4. Как бы вы выполнили предьщущее упражнение с измененным условием, при котором выход σ вычисляется как максимальное из шести его значений, а не как их сумма?

5. Рассмотрим функцию потерь $L = x^2 + y^{10}$. Реализуйте простой алгоритм крутейшего спуска для графического построения координат по мере их изменения от начальной точки до оптимального значения 0. Рассмотрите две различные начальные точки (0,5; 0,5) и (2; 2) и отобразите на графике траектории в этих двух случаях при постоянной скорости обучения. Что можно сказать о поведении алгоритма в этих двух случаях?

Список литературы

Для выполнения лабораторной работы, при подготовке к защите, а также для ответа на контрольные вопросы рекомендуется использовать следующие источники: [1-5].

ЛАБОРАТОРНАЯ РАБОТА 4. ГЕНЕРАЦИЯ ТЕКСТА С ИСПОЛЬЗОВАНИЕМ АВТОРЕГРЕССИВНЫХ МОДЕЛЕЙ

Цели и задачи

Цель лабораторной работы: научиться использовать авторегрессивные модели для генерации текстового вывода.

Основные задачи:

– исследование возможностей предобученной модели для генерации текста;

– оценка эффективности модели.

Теоретическое обоснование

В предыдущей работе было показано как обучать авторегрессивную модель на собственном корпусе. В результате вы обучили собственную версию GPT-2. В данной работе данная моель будет применена на практике.

Оборудование и материалы

Для выполнения лабораторной работы рекомендуется использовать персональный компьютер со следующими программными средствами разработки: Библиотека Transformers, TensorFlow и PyTorch, Python 3.6x, Jupyter Notebook, Google Colaboratory, Docker, Locust.io, Git.

Методика и порядок выполнения работы

1. Сгенерируем предложения из модели, которую вы обучили, при помощи такого кода:

```
def generate(start, model):
    input_token_ids = tokenizer_gpt.encode(start, return_tensors='tf')
    output = model.generate(input_token_ids, max_length = 500,
```

```
        num_beams = 5, temperature = 0.7, no_repeat_ngram_size=2,
        num_return_sequences=1
    )
    return tokenizer_gpt.decode(output[0])
```

Функция generate, определенная в этом фрагменте кода, принимает строку start и генерирует последовательности символов, следующие за этой строкой. Вы можете изменить такие параметры, как max_length, чтобы установить меньший размер последовательности, или num_ return_sequences, чтобы сгенерировать другой текст.

2. Запустим этот код с пустой строкой:

generate(" ", model)

Должен получиться вывод, показанный на рис. 4.1.

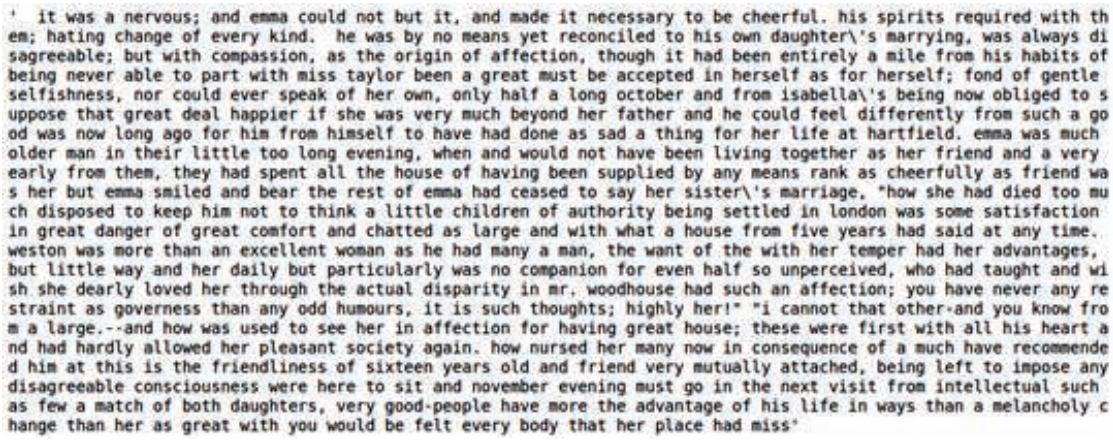


Рисунок 4.1 – Пример генерации текста с помощью GPT-2

Как видно из предыдущего вывода на рис. 4.1, модель генерирует длинный текст, и хотя его довольно трудно читать, во многих случаях синтаксис почти правильный.

3. Протестируйте модель. Изучите возможности по генерации текстоового вывода. Предложите рекомендации по улучшению качества вывод предобученной модели.

Содержание отчета и его форма

Отчет по лабораторной работе должен содержать:

1. Номер и название лабораторной работы; задачи лабораторной работы.

2. Реализация каждого пункта подраздела «Методика и порядок выполнения работы» с приведением исходного кода программы, диаграмм и графиков для визуализации данных.

3. Ответы на контрольные вопросы.

4. Экранные формы (консольный вывод) и листинг программного кода с комментариями, показывающие порядок выполнения лабораторной работы, и результаты, полученные в ходе её выполнения.

Отчет о выполнении лабораторной работы подписывается студентом и сдается преподавателю.

Контрольные вопросы

1. Рассмотрим две нейронные сети, используемые для регрессионного моделирования, которые имеют одинаковую структуру входного слоя и включают по 10 скрытых слоев, каждый из которых содержит по 100 элементов. В обоих случаях выходным узлом является одиночный элемент с линейной активацией. Единственным отличием является то, что в одной из сетей в скрытых слоях используются линейные активации, а в другой – сигмоидные. В какой модели дисперсия предсказаний будет больше?

2. Рассмотрим ситуацию, когда имеются четыре атрибута $x_1 \dots x_4$, а зависимая переменная y такова, что $y = 2 \cdot x_1$. Создайте небольшой тренировочный набор данных, включающий 5 различных примеров, в котором линейная регрессионная модель будет иметь бесконечное количество решений для коэффициентов с $w_1 = 0$. Проанализируйте, как эта модель будет работать с данными, не входящими в тренировочный набор. Почему в данном случае пригодится регуляризация?

3. Реализуйте перцептрон с регуляризацией и без регуляризации. Протестируйте обе разновидности перцептрона на тренировочных данных и на данных, не входящих в тренировочные, используя для этого набор данных

Ionosphere из хранилища UCI Machine Leaming Repository. Что можно сказать относительно влияния регуляризации в обоих случаях? Повторите этот эксперимент с меньшими примерами из тренировочных данных Ionosphere и запишите свои наблюдения.

4. Реализуйте автокодировщик с одним скрытым слоем. Реконструируйте входы для набора данных Ionosphere из предыдущего примера, используя два варианта: а) без добавления шума, но с регуляризацией весов, и б) с добавлением гауссовского шума, но без регуляризации весов.

5. Рассмотрите сжимающий автокодировщик с одним скрытым слоем и активацией ReLU. Проанализируйте, как изменяются обновления в случае использования ReLU-активации.

6. Предположим, у вас есть модель, обеспечивающая точность около 80% как на тренировочном наборе данных, так и на тестовых данных, не входящих в тренировочный набор. Рекомендовали бы вы увеличение набора данных или настройку модели для повышения точности?

7. Добавление гауссовского шума во входные признаки в линейной регрессии эквивалентно L_2-регуляризации линейной регрессии. Подумайте, почему добавление гауссовского шума во входные данные шумоподавляющего автокодировщика с одним скрытым слоем и линейными элементами примерно эквивалентно L_2-регуляризации сингулярного разложения.

8. Рассмотрим сеть с одним входным слоем, двумя скрытыми слоями и одним выходом, предсказывающим бинарную метку. Все скрытые слои используют сигмоиду, но регуляризация не используется. Входной слой содержит d элементов, а каждый скрытый слой – p элементов. Предположим, вы добавили дополнительный скрытый слой между двумя имеющимися скрытыми слоями, и этот дополнительный скрытый слой содержит q линейных элементов.

А. Объясните, почему мощность этой модели уменьшится, если $q < p$, несмотря на то что добавление скрытого слоя увеличивает количество параметров.

Б. Увеличится ли мощность модели, если $q > p$?

9. Боб разделил помеченные классифицированные данные на две части: одну для построения модели, другую - для ее валидации. Затем Боб выполнил тестирование 1000 нейронных архитектур, обучая параметры (с помощью алгоритма обратного распространения ошибки) на части данных, предназначенной для построения модели, и тестируя точность на валидационной части данных. Подумайте, почему результирующая модель, скорее всего, продемонстрирует для тестовых данных, не входящих в тренировочный набор, худшую точность, чем для валидационных данных, несмотря на то что валидационные данные не использовались для обучения параметров. Имеются ли у вас какие-либо рекомендации для Боба относительно использования результатов его 1000 валидационных тестов?

10. Повышается ли точность классификации на тренировочных данных с увеличением их объема? А что можно сказать о поточечном усреднении функции потерь на тренировочных примерах? В какой точке полученные в процессе тренировки и тестирования значения точности сблизятся? Объясните свой ответ.

11. Какое влияние на точность тренировки и тестирования оказывает увеличение параметра регуляризации? В какой точке полученные в процессе тренировки и тестирования значения в точности сблизятся?

Список литературы

Для выполнения лабораторной работы, при подготовке к защите, а также для ответа на контрольные вопросы рекомендуется использовать следующие источники: [1–5].

ЛАБОРАТОРНАЯ РАБОТА 5. КЛАССИФИКАЦИЯ ТЕКСТА

Цели и задачи

Цель лабораторной работы: изучение принципов построения информационных систем с использованием линейных методов машинного обучения.

Основные задачи:

– тонкая настройка модели BERT для двоичной классификации с одним предложением;

– обучение модели классификации с помощью PyTorch;

– тонкая настройка BERT для многоклассовой классификации с пользовательскими наборами данных;

– тонкая настройка BERT для регрессии пар предложений.

Теоретическое обоснование

Классификация текста (также известная как категоризация текста) – это способ сопоставления документа (предложения, публикации в Twitter, главы книги, содержимого электронной почты и т. д.) с категорией из предопределенного списка (классов). В случае двух классов, которые имеют положительные и отрицательные метки, это называется бинарной классификацией (binary classification), или, если использовать более строгие термины, анализом тональности (sentiment analysis) или анализом эмоциональной окраски текста. Если классов больше двух, это называется многоклассовой классификацией (multi-class classification), когда классы являются взаимоисключающими, или классификацией с несколькими метками (multi-label classification), когда классы не являются взаимоисключающими и документ может получать более одной метки. Например, контент новостной статьи может быть связан со спортом и

политикой одновременно. Помимо этой классификации, мы можем захотеть оценить документы в диапазоне [–1,1] или ранжировать их в диапазоне [1–5].

Мы можем решить эту проблему с помощью регрессионной модели, в которой вывод является числовым, а не категориальным. К счастью, трансформеры позволяют эффективно решать эти проблемы. Для задач, связанных с парой предложений, таких как схожесть документов или выявление следствия, вводом является не одно предложение, а два предложения, как показано на рис. 5.1. Мы можем оценить, насколько два предложения семантически похожи, или предсказать, являются ли они семантически похожими. Еще одна задача, связанная с парами предложений, – это выявление следствия, когда смысл второго предложения логически вытекает из первого. Такая задача относится к многоклассовой классификации. Например, в тесте GLUE на рис. 5.1 используются две последовательности, оцениваемые по классам следует/противоречит/нейтрально.

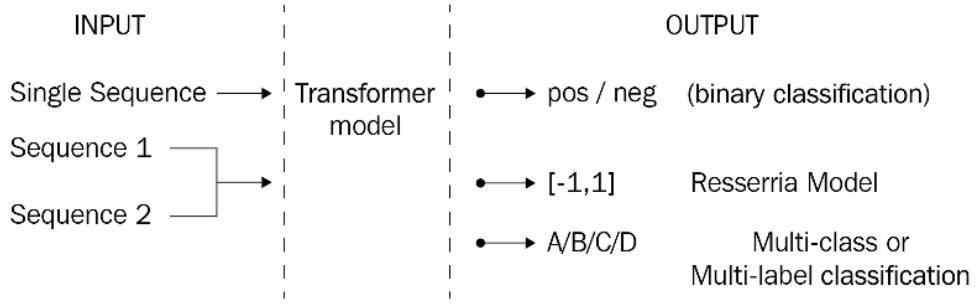


Рисунок 5.1 – Схема классификации текста

Оборудование и материалы

Для выполнения лабораторной работы рекомендуется использовать персональный компьютер со следующими программными средствами разработки: Библиотека Transformers, TensorFlow и PyTorch, Python 3.6x, Jupyter Notebook, Google Colaboratory, Docker, Locust.io, Git.

Методика и порядок выполнения работы

В данной работе рассмотривается настройка предварительно обученной модели BERT для анализа тональности с помощью популярного набора данных IMDb sentiment. Работа с графическим процессором ускорит наш процесс обучения, но если у вас нет таких ресурсов, вы также можете выполнить тонкую настройку с помощью обычного ЦП.

1. Чтобы получить и сохранить информацию о текущем устройстве, выполните следующие строки кода:

```
from torch import cuda
device = 'cuda' if cuda.is_available() else 'cpu'
```

2. Будем использовать здесь класс DistilBertForSequenceClassification, который унаследован от класса DistilBert со специальным дополнением для классификации последовательностей поверх основного кода. Мы можем использовать этот классификатор для обучения моделей классификации, у которых количество классов по умолчанию равно 2:

```
from transformers import DistilBertTokenizerFast, DistilBertForSequenceClassification
model_path= 'distilbert-base-uncased'
tokenizer = DistilBertTokenizerFast.from_pre-trained(model_path)
model = \ DistilBertForSequenceClassification.from_pre-trained(
model_path,
id2label={0:"NEG", 1:"POS"},
label2id={"NEG":0, "POS":1})
```

3. Обратите внимание, что два параметра, называемых id2label и label2id, передаются модели для использования во время вывода. В качестве альтернативы мы можем создать конкретный объект config и передать его модели следующим образом:

```
config = AutoConfig.from_pre-trained(....)
SequenceClassification.from_pre-trained(....config=config)
```

4. Получим популярный набор данных классификации тональности под названием IMDB Dataset. Исходный набор данных состоит из двух наборов данных: 25 000 примеров для обучения и 25 примеров для тестирования. Мы разделим набор данных на два блока: тестовый и проверочный. Обратите внимание, что примеры для первого блока положительны, а примеры второго блока – отрицательны. Мы можем распределить примеры следующим образом:

```
from datasets import load_dataset
imdb_train= load_dataset('imdb', split="train")
imdb_test= load_dataset('imdb', split="test[:6250]+test[-6250:]")
imdb_val= load_dataset('imdb', split="test[6250:12500]+tst[-12500:-6250]")
```

5. Проверим форму набора данных:

```
>>> imdb_train.shape, imdb_test.shape, imdb_val.shape
((25000, 2), (12500, 2), (12500, 2))
```

6. Если вы располагаете ограниченными вычислительными ресурсами, то можете взять небольшую часть набора данных. Например, вы можете запустить следующий код, чтобы выбрать 4000 примеров для обучения, 1000 для тестирования и 1000 для проверки:

```
imdb_train= load_dataset('imdb', split="train[:2000]+train[-2000:]")
imdb_test= load_dataset('imdb',split="test[:500]+test[-500:]")
imdb_val= load_dataset('imdb', split="test[500:1000]+test[-1000:-500]")
```

7. Теперь мы можем пропустить эти наборы данных через модель tokenizer, чтобы подготовить их к обучению:

```
enc_train = imdb_train.map(lambda e: tokenizer(e['text'],
padding=True, truncation=True),
batched=True, batch_size=1000)
enc_test = imdb_test.map(lambda e: tokenizer( e['text'], padding=True, truncation=True), batched=True, batch_size=1000)
enc_val = imdb_val.map(lambda e: tokenizer( e['text'], padding=True, truncation=True), batched=True, batch_size=1000)
```

8. Посмотрим, как выглядит тренировочный набор. Токенизатор добавил в набор данных маску внимания и входные идентификаторы, чтобы модель BERT могла его обрабатывать:

```
import pandas as pd
pd.DataFrame(enc_train)
```

Результат выполнения этих строк кода показан на рис. 5.2.

	attention_mask	input_ids	label	text
0	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...	[101, 22953, 2213, 4381, 2152, 2003, 1037, 947...	1	Bromwell High is a cartoon comedy. It ran at t...
1	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...	[101, 11573, 2791, 1006, 2030, 2160, 24913, 20...	1	Homelessness (or Houselessness as George Carli...
2	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...	[101, 8235, 2058, 1011, 3772, 2011, 23920, 575...	1	Brilliant over-acting by Lesley Ann Warren. Be...
3	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...	[101, 2023, 2003, 4089, 1996, 2087, 2104, 9250...	1	This is easily the most underrated film inn th...
4	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...	[101, 2023, 2003, 2025, 1996, 5171, 11463, 837...	1	This is not the typical Mel Brooks film. It wa...
...	...	...	...	...
24995	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...	[101, 2875, 1996, 2203, 1997, 1996, 3185, 1010...	0	Towards the end of the movie, I felt it was to...
24996	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...	[101, 2023, 2003, 1996, 2785, 1997, 3185, 2008...	0	This is the kind of movie that my enemies cont...
24997	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...	[101, 1045, 2387, 1005, 6934, 1005, 2197, 2305...	0	I saw 'Descent' last night at the Stockholm Fi...
24998	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...	[101, 2070, 3152, 2008, 2017, 4060, 2039, 2005...	0	Some films that you pick up for a pound turn o...
24999	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...	[101, 2023, 2003, 2028, 1997, 1996, 12873, 435...	0	This is one of the dumbest films, I've ever se...

25000 rows × 4 columns

Рисунок 5.2 – Закодированный набор обучающих данных

Дообучите полученную модель на данных.

Содержание отчета и его форма

Отчет по лабораторной работе должен содержать:

1. Номер и название лабораторной работы; задачи лабораторной работы.

2. Реализация каждого пункта подраздела «Методика и порядок выполнения работы» с приведением исходного кода программы, диаграмм и графиков для визуализации данных.

3. Ответы на контрольные вопросы.

4. Экранные формы (консольный вывод) и листинг программного кода с комментариями, показывающие порядок выполнения лабораторной работы, и результаты, полученные в ходе её выполнения.

Отчет о выполнении лабораторной работы подписывается студентом и сдается преподавателю.

Контрольные вопросы

1. Даже несмотря на то, что для обучения модели используется дискретное семплирование алгоритма контрастивной дивергенции, окончательная фаза вывода выполняется с использованием вещественных сигмоид и Softmax-активаций. Опишите, как бы вы воспользовались этим фактом для тонкой настройки обученной модели с помощью алгоритма обратного распространения ошибки.

2. Реализуйте алгоритм контрастивной дивергенции ограниченной машины Больцмана. Также реализуйте алгоритм вывода, позволяющий получить распределение вероятности скрытых элементов для тестового примера. Используйте Python или любой другой язык программирования по своему выбору.

3. Рассмотрим машину Больцмана без двудольного ограничения (RBM), но с тем ограничением, что все элементы являются видимыми. Как это ограничение упрощает процесс тренировки машины Больцмана?

4. Предложите подход, позволяющий использовать RBM для обнаружения выбросов.

5. Получите обновления весов для тематического моделирования с использованием подхода на основе RBM.

6. Предложите способ расширения RBM для задач коллаборативной фильтрации с помощью дополнительных слоев.

7. Известна и хорошо описана машина Больцмана для классификации. Однако этот подход ограничен бинарной классификацией. Предложите способ его расширения на задачи многоклассовой классификации.

Список литературы

Для выполнения лабораторной работы, при подготовке к защите, а также для ответа на контрольные вопросы рекомендуется использовать следующие источники: [1–5].

ЛАБОРАТОРНАЯ РАБОТА 6. НАСТРОЙКА ЯЗЫКОВЫХ МОДЕЛЕЙ ДЛЯ КЛАССИФИКАЦИИ ТОКЕНОВ

Цели и задачи

Цель лабораторной работы: получить опыт тонкой настройки языковых моделей для классификации токенов.

Основные задачи:

– распознавание именованных объектов;

– тегирование частей речи (part-of-speech, POS);

– ответы на вопросы (question answering, QA);

– тонкую настройку модели.

Теоретическое обоснование

Задача классификации каждого токена в последовательности токенов называется классификацией токенов (token classification). В этой задаче говорится, что конкретная модель должна уметь относить каждый токен к какому-либо классу. POS и NER – две наиболее известные задачи по этому критерию. Однако эту категорию попадает еще одна важная задача NLP – QA.

Одной из хорошо известных задач в категории классификации токенов является NER – распознавание того, является токен объектом или нет, и определение типа каждого обнаруженного объекта. Например, текст может одновременно содержать несколько объектов – имена людей, местоположения, организации и другие типы. Следующий текст является ярким примером NER: Джордж Вашингтон – один из президентов Соединенных Штатов Америки.

Джордж Вашингтон – это имя человека, а Соединенные Штаты Америки – это название места. Ожидается, что модель маркировки последовательности пометит все слова тегами, каждый из которых содержит информацию о теге.

Для стандартных задач NER повсеместно используются теги BIO. В табл. 6.1 представлен список тегов и их описание.

Таблица 6.1 – Теги BIO и их описание

Тег	Описание
O	Outside of entity / За пределами объекта
B-PER	Beginning of Person entity / Начало объекта «Персона»
I-PER	Inside of Person entity / Внутри объекта «Персона»
B-LOC	Beginning of Location entity / Начало объекта «Местоположение»
C	Inside of Location entity / Внутри объекта «Местоположение»
B-ORG	Beginning of Organization entity / Начало объекта «Организация»
I-ORG	Inside of Organization entity / Внутри объекта «Организация»
B-MISC	Beginning of Miscellaneous entity / Начало объекта «Прочее»
I-MISC	Inside of Miscellaneous entity / Внутри объекта «Прочее»

В этой таблице B обозначает начало тега, I обозначает внутреннюю часть тега, а O – внешнюю часть объекта. Из этих трех букв составлено название аннотации BIO. Например, так будет выглядеть приведенное выше предложение, если его аннотировать с помощью BIO:

[B-PER|Джордж] [I-PER|Вашингтон] [O|один] [O|из]

[O|президентов] [B-LOC|Соединенных] [I-LOC|Штатов]

[I-LOC|Америки] [O|.]

Соответственно, входная последовательность должна быть размечена в формате BIO. Образец набора данных может выглядеть, как показано на рис. 6.1.

```
SOCCER NN B-NP O
- : O O
JAPAN NNP B-NP B-LOC
GET VB B-VP O
LUCKY NNP B-NP O
WIN NNP I-NP O
, , O O
CHINA NNP B-NP B-PER
IN IN B-PP O
SURPRISE DT B-NP O
DEFEAT NN I-NP O
. . O O

Nadim NNP B-NP B-PER
Ladki NNP I-NP I-PER

AL-AIN NNP B-NP B-LOC
, , O O
United NNP B-NP B-LOC
Arab NNP I-NP I-LOC
Emirates NNPS I-NP I-LOC
1996-12-06 CD I-NP O
```

Рисунок 6.1 – Набор данных CONLL2003

В дополнение к тегам NER, которые мы видели, в этом наборе данных доступны теги POS.

Теги POS, или грамматические теги (grammar tagging) – это аннотирование слова в заданном тексте в соответствии с его соответствующей частью речи. Для простоты в этой книге мы будем считать, что каждое слово относится к одной из четырех POS-категорий: существительное, прилагательное, наречие и глагол. Однако с лингвистической точки зрения существует много других ролей, помимо этих четырех. В случае тегов POS существуют варианты, среди которых самым известным является набор тегов Penn Treebank POS. В табл. 6.2 показано краткое изложение и соответствующее описание этих ролей.

Таблица 6.2 – POS-теги Penn Treebank

CC	coordinating conjunction соединительный союз	and
CD	cardinal numberколичественное числительное	1, third
DT	determiner определяющее слово	the
EX	existential thereпространственно-временная отсылка	there is
FW	foreign wordиностранное слово	les
IN	preposition, subordinating conjunctionпредлог, подчинительный союз	in, of, like

IN/that	that as subordinatorthat как подчиненный	that
JJ	adjective прилагательное	green
JJR	adjective, comparativeприлагательное, сравнительная степень	greener
JJS	adjective, superlativeприлагательное, превосходная степень	greenest
LS	list markerмаркер списка	D
MD	modalмодальный (глагол)	could, will
NN	noun, singular or masscуществительное, единственное или множественное	table
NNS	noun pluralсуществительное, множественное число	tables
NP	proper noun, singularимя собственное, единственное число	John

Оборудование и материалы

Для выполнения лабораторной работы рекомендуется использовать персональный компьютер со следующими программными средствами разработки: Библиотека Transformers, TensorFlow и PyTorch, Python 3.6x, Jupyter Notebook, Google Colaboratory, Docker, Locust.io, Git.

Методика и порядок выполнения работы

1. Для загрузки набора данных используйте следующие строки кода:

```
import datasets
conll2003 = datasets.load_dataset("conll2003")
```

Появится индикатор выполнения загрузки, и после завершения загрузки и кеширования набор данных будет готов к использованию. На рис. 6.2 показаны индикаторы выполнения.

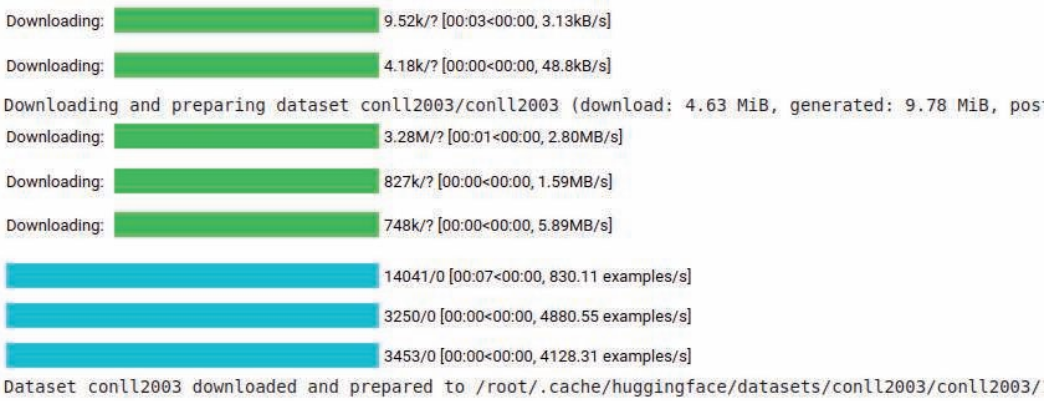


Рисунок 6.2 – Загрузка и подготовка набора данных

2. Вы можете легко перепроверить набор данных, запросив образец обучающих данных с помощью следующей строки кода:

>>> conll2003["train"][0]

Вы должны получить следующий результат выполнения этой строки кода – пример обучающих данных CONLL2003:

{'chunk_tags': [11, 21, 11, 12, 21, 22, 11, 12, 0],

'id': '0',

'ner_tags': [3, 0, 7, 0, 0, 0, 7, 0, 0],

'pos_tags': [22, 42, 16, 21, 35, 37, 16, 21, 7],

'tokens': ['EU','rejects','German','call','to','boycott','British','lamb','.']}

3. В примере обучающих данных показаны соответствующие теги для POS и NER. Здесь мы будем использовать только теги NER. Вы можете использовать следующую команду, чтобы получить теги NER, доступные в этом наборе данных:

>>> conll2003["train"].features["ner_tags"]

4. В результате будут выведены все девять тегов BIO:

>>> Sequence(feature=ClassLabel(num_classes=9, names=['O', 'B-PER', 'I-PER', 'B-ORG', 'I-ORG', 'B-LOC', 'I-LOC', 'B-MISC', 'I-MISC'], names_file=None, id=None), length=-1, id=None)

5. Следующим шагом будет загрузка токенизатора BERT:

from transformers import BertTokenizerFast

tokenizer = BertTokenizerFast.from_pretrained("bert-baseuncased")

6. Класс tokenizer также может работать с предложениями, размеченными пробелами. Нам нужно настроить наш токенизатор на работу с предложениями, размеченными пробелами, потому что задача NER имеет основанную на токене метку для каждого токена. Токены в этой задаче обычно представляют собой слова, обозначенные пробелами, а не BPE или любыми другими токенами токенизатора. Вот как токенизатор можно использовать с предложением, размеченным пробелами:

```
>>> tokenizer(["Oh","this","sentence","is","tokenized",
"and","splitted","by","spaces"],
is_split_into_words=True)
```

Как видите, чтобы решить проблему, достаточно просто установить для is_split_into_words значение True.

7. Перед применением обучающих данных их нужно обработать. Для этого мы должны использовать следующую функцию и отобразить весь набор данных:

```
def tokenize_and_align_labels(examples):
tokenized_inputs = tokenizer(examples["tokens"],
truncation=True, is_split_into_words=True)
labels = []
for i, label in enumerate(examples["ner_tags"]):
word_ids = tokenized_inputs.word_ids(batch_index=i)
previous_word_idx = None
label_ids = []
for word_idx in word_ids:
if word_idx is None:
label_ids.append(-100)
elif word_idx != previous_word_idx:
        label_ids.append(label[word_idx])
else:
        label_ids.append(label[word_idx] \
```

```
        if label_all_tokens else -100) \
        previous_word_idx = word_idx \
        labels.append(label_ids)
    tokenized_inputs["labels"] = labels
    return tokenized_inputs
```

8. Эта функция гарантирует, что наши токены и метки будут выровнены правильно. Выравнивание необходимо, потому что слова должны быть цельными. Чтобы проверить эту функцию на деле, вы можете запустить ее, предоставив ей один образец обучающих данных:

```
q = tokenize_and_align_labels(conll2003['train'][4:5])
print(q)
```

Результат будет следующим:

```
>>> {'input_ids': [[101, 2762, 1005, 1055, 4387, 2000,
1996, 2647, 2586, 1005, 1055, 15651, 2837, 14121, 1062,
9328, 5804, 2056, 2006, 9317, 10390, 2323, 4965, 8351,
4168, 4017, 2013, 3032, 2060, 2084, 3725, 2127, 1996,
4045, 6040, 2001, 24509, 1012, 102]], 'token_type_ids':
[[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0]], 'attention_mask': [[1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1]], 'labels': [[-100, 5, 0,
-100, 0, 0, 0, 3, 4, 0, -100, 0, 0, 1, 2, -100, -100, 0,
0, 0, 0, 0, 0, 0, -100, -100, 0, 0, 0, 0, 5, 0, 0, 0, 0,
0, 0, 0, -100]]}
```

Дообучите модель тегов POS, а также протестируйте эту модель.

Содержание отчета и его форма

Отчет по лабораторной работе должен содержать:

1. Номер и название лабораторной работы; задачи лабораторной работы.

2. Реализация каждого пункта подраздела «Методика и порядок выполнения работы» с приведением исходного кода программы, диаграмм и графиков для визуализации данных.

3. Ответы на контрольные вопросы.

4. Экранные формы (консольный вывод) и листинг программного кода с комментариями, показывающие порядок выполнения лабораторной работы, и результаты, полученные в ходе её выполнения.

Отчет о выполнении лабораторной работы подписывается студентом и сдается преподавателю.

Контрольные вопросы

1. Загрузите приведенную RNN, работающую на уровне символов:
https://github.com/karpathy/char-rnn
и обучите ее на наборе данных "tiny Shakespeare ", досгупном по тому же адресу. Создайте выходы языковой модели после выполнения тренировки в течение 1) 5 эпох, 2) 50 эпох и 3) 500 эпох. Какие существенные различия наблюдаются между этими тремя выходами?

2. Рассмотрим эхо-сеть, в которой скрытые состояния разделены на К групп по *p/K* элементов каждая. Скрытым состояниям отдельной группы разрешается образовывать связи лишь в пределах собственной группы в следующий момент времени. Обсудите, как этот подход связан с ансамблевым методом, в котором создаются К независимых эхо-сетей и предсказания этих сетей усредняются.

3. Рассмотрим рекуррентную сеть, в которой скрытые состояния имеют размерность 2. Каждый элемент матрицы W_{hh} размером 2x2, которая осуществляет преобразование между скрытыми состояниями, равен 3,5. Кроме того, между скрьпыми состояниями различных временных слоев используется сигмоидная активация. Будет ли такая сеть подвержена проблемам затухающих и взрывных градиентов?

4. Предположим, вы располагаете большой базой биологических данных с информацией об азотистых основаниях, которая содержится в виде строк кодовых последовательностей, состоящих из символов {A, C, T, G} (A – аденин, C – цитозин, T – тимин, G – гуанин). Некоторые из этих строк содержат необычные мутации, представляющие изменения в азотистых основаниях. Предложите метод обучения без учителя (т.е. нейронную архитектуру), использующий RNN для обнаружения таких мутаций.

5. Как бы вы изменили архитектуру из предыдущего вопроса, если бы вам была предоставлена тренировочная база данных, в которой позиции мутаций в каждой последовательности помечены, но в тестовой базе данных оставлены непомеченными?

6. Предложите рекомендации относительно предварительного обучения входного и выходного слоев в подходе для машинного перевода с применением обучения «последовательность в последовательность».

7. Изменятся ли обученные веса после масштабирования тренировочного набора данных с использованием некоторого коэффициента масштабирования в случае пакетной или послойной нормализации? Каков был бы ваш ответ, если бы масштабированию был подвергнуто лишь небольшое подмножество точек тренировочных данных? Повлияет ли на обученные веса в случае любого метода нормализации изменение центрирования набора данных?

Список литературы

Для выполнения лабораторной работы, при подготовке к защите, а также для ответа на контрольные вопросы рекомендуется использовать следующие источники: [1–5].

ЗАКЛЮЧЕНИЕ

Методические указания к лабораторным работам по дисциплине «Разработка диалоговых систем» для студентов направления 09.03.02 «Информационные системы и технологии» охватывает теоретические аспекты построения информационных систем на основе искусственных нейронных сетей, а также предлагает студентам практические рекомендации по разработке систем на основе ИНС.

В методических указаниях рассмотрены практические аспекты проектирования, обучения и применения ИНС для решения професиональных задач. Студенту предлагается реализовать ИНС различноы архитектур.

СПИСОК ЛИТЕРАТУРЫ

Список основной литературы

1. Уэс, Маккинли. Python и анализ данных Электронный ресурс / Маккинли Уэс ; пер. А. А. Слинкин. - Python и анализ данных,2021-04-19. - Саратов : Профобразование, 2017. - 482 c. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 978-5-4488-0046-7, экземпляров неограниченно.

2. Сузи, Р.А. Язык программирования Python Электронный ресурс : учебное пособие / Р.А. Сузи. - Язык программирования Python,2020-07-28. - Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. - 350 c. - Книга находится в базовой версии ЭБС IPRbooks. - ISBN 5-9556-0058-2, экземпляров неограниченно

Список дополнительной литературы

3. Стенли, Липпман. Язык программирования C++ Электронный ресурс : Полное руководство / Липпман Стенли, Лажойе Жози ; пер. А. Слинкин. - Язык программирования C++,2021-04-19. - Саратов : Профобразование, 2017. - 1104 c. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 978-5-4488-0136-5, экземпляров неограниченно

4. Седжвик, Р. Алгоритмы на C++ / Р. Седжвик. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 1773 с., экземпляров неограниченно

5. https://archive.ics.uci.edu/ml/index.html – Репозиторий наборов данных для машинного обучения (Центр машинного обучения и интеллектуальных систем).

6. https://www.kaggle.com – Портал и система проведения соревнований по проблемам анализа данных.

7. https://www.mockaroo.com – Сайт для генерации наборов данных.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Методические указания

по организации и проведению самостоятельной работы
по дисциплине

«Разработка диалоговых систем»

для студентов направления подготовки

09.03.02 «Информационные системы и технологии»

направленность (профиль)

«Прикладное программирование в интеллектуальных информационных системах»

Ставрополь, 2026

1. Общие положения

Самостоятельная работа – планируемая учебная, учебно-исследовательская, научно-исследовательская работа студентов, выполняемая во внеаудиторное (аудиторное) время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия (при частичном непосредственном участии преподавателя, оставляющем ведущую роль за работой студентов).

Самостоятельная работа студентов (СРС) в ВУЗе является важным видом учебной и научной деятельности студента. Самостоятельная работа студентов играет значительную роль в рейтинговой технологии обучения.

К основным видам самостоятельной работы студентов относятся:

− формирование и усвоение содержания конспекта лекций на базе рекомендованной лектором учебной литературы, включая информационные образовательные ресурсы (электронные учебники, электронные библиотеки и др.);

− написание докладов;

− подготовка к семинарам, практическим и лабораторным работам, их оформление;

− составление аннотированного списка статей из соответствующих журналов по отраслям знаний (педагогических, психологических, методических и др.);

− выполнение учебно-исследовательских работ, проектная деятельность;

− подготовка практических разработок и рекомендаций по решению проблемной ситуации;

− выполнение домашних заданий в виде решения отдельных задач, проведения типовых расчетов, расчетно-компьютерных и индивидуальных работ по отдельным разделам содержания дисциплин и т.д.;

− компьютерный текущий самоконтроль и контроль успеваемости на базе электронных обучающих и аттестующих тестов;

− выполнение курсовых работ (проектов) в рамках дисциплин;

− выполнение выпускной квалификационной работы и др.

Методика организации самостоятельной работы студентов зависит от структуры, характера и особенностей изучаемой дисциплины, объема часов на ее изучение, вида заданий для самостоятельной работы студентов, индивидуальных качеств студентов и условий учебной деятельности.

Процесс организации самостоятельной работы студентов включает в себя следующие этапы:

- подготовительный (определение целей, составление программы, подготовка методического обеспечения, подготовка оборудования);
- основной (реализация программы, использование приемов поиска информации, усвоения, переработки, применения, передачи знаний, фиксирование результатов, самоорганизация процесса работы);
- заключительный (оценка значимости и анализ результатов, их систематизация, оценка эффективности программы и приемов работы, выводы о направлениях оптимизации труда).

Самостоятельная работа по дисциплине «Разработка диалоговых систем» направлена на формирование следующих **компетенций**:

Код, формулировка компетенции	Код, формулировка индикатора
ПК-3 Способен использовать интеллектуальные модели при решении задач профессиональной деятельности	ПК-3$_{ИД-3.1}$ – Демонстрирует знания теоретических основ функционирования интеллектуальных моделей и особенности их обучения и оптимизации
	ПК-3$_{ИД-3.2}$ – Анализирует и применяет интеллектуальные методы согласно решаемой задаче

	ПК-3$_{ИД-3.3}$ – Применяет инструментальные средства визуализации данных, в том числе больших данных
	ПК-3$_{ИД-3.4}$ – Умеет настраивать параметры и гиперпараметры интеллектуальных моделей для достижения высокой эффективности обучения и функционирования
	ПК-3$_{ИД-3.5}$ – Применяет инструментальные средства интеллектуальной обработки данных различной природы (текстовые, изображения, видео, аудио)
ПК-4 Способен анализировать предметную область и обосновывать эффективность применения интеллектуальных моделей различного типа	ПК-4$_{ИД-4.1}$ – Демонстрирует знания в области теории функционирования обучаемых алгоритмов и интеллектуальных моделей
	ПК-4$_{ИД-4.2}$ – Демонстрирует знание стандартов оценки качества систем искусственного интеллекта
	ПК-4$_{ИД-4.3}$ – Осуществляет обоснованный отбор моделей искусственного интеллекта для решения профессиональных задач
ПК-9 Способен проводить научно-исследовательскую работу на основе анализа передового опыта в сферах проектирования и разработки информационных систем, в том числе интеллектуальных	ПК-9$_{ИД-9.1}$ – Проводит анализ отечественного и зарубежного опыта по тематике исследований
	ПК-9$_{ИД-9.2}$ – Составляет обзоры, рефераты, отчеты, готовит научные публикации и доклады на научных конференциях и семинарах по тематике своих исследований
	ПК-9$_{ИД-9.3}$ – Реализует вычислительные эксперименты в рамках исследований с использованием передовых технологий и методик в области интеллектуальных информационных систем

2. Цель и задачи самостоятельной работы

Ведущая цель организации и осуществления СРС совпадает с целью обучения студента – формирование набора общенаучных, профессиональных и специальных компетенций будущего бакалавра по соответствующему направлению подготовки

При организации СРС важным и необходимым условием становятся формирование умения самостоятельной работы для приобретения знаний, навыков и возможности организации учебной и научной деятельности. Целью самостоятельной работы студентов является овладение фундаментальными знаниями, профессиональными умениями и навыками деятельности по профилю, опытом творческой, исследовательской деятельности. Самостоятельная работа студентов способствует развитию самостоятельности, ответственности и организованности, творческого подхода к решению проблем учебного и профессионального уровня.

Задачами СРС являются:

- систематизация и закрепление полученных теоретических знаний и практических умений студентов;
- углубление и расширение теоретических знаний;
- формирование умений использовать нормативную, правовую, справочную документацию и специальную литературу;
- развитие познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;
- формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- развитие исследовательских умений;

- использование материала, собранного и полученного в ходе самостоятельных занятий на семинарах, на практических и лабораторных занятиях, при написании курсовых и выпускной квалификационной работ, для эффективной подготовки к итоговым зачетам и экзаменам.

3. Технологическая карта самостоятельной работы студента

Коды реализуемых компетенций	Вид деятельности студентов	Средства и технологии оценки	Объем часов, в том числе (акад.)		
			СРС	Контактная работа с преподавателем	Всего
8 семестр					
ПК-3, ПК-4, ПК-9	Самостоятельное изучение литературы и источников	Собеседование	6	2	8
ПК-3, ПК-4, ПК-9	Подготовка к лабораторным занятиям	Защита ЛР	48	12	60
Итого за 8 семестр			54	14	68
Итого			54	14	68

4. Порядок выполнения самостоятельной работы студентом

4.1. Методические рекомендации по работе с учебной литературой

При работе с книгой необходимо подобрать литературу, научиться правильно ее читать, вести записи. Для подбора литературы в библиотеке используются алфавитный и систематический каталоги.

Важно помнить, что рациональные навыки работы с книгой - это всегда большая экономия времени и сил.

Правильный подбор учебников рекомендуется преподавателем, читающим лекционный курс. Необходимая литература может быть также указана в методических разработках по данному курсу.

Изучая материал по учебнику, следует переходить к следующему вопросу только после правильного уяснения предыдущего, описывая на бумаге все выкладки и вычисления (в том числе те, которые в учебнике опущены или на лекции даны для самостоятельного вывода).

При изучении любой дисциплины большую и важную роль играет самостоятельная индивидуальная работа.

Особое внимание следует обратить на определение основных понятий курса. Студент должен подробно разбирать примеры, которые поясняют такие определения, и уметь строить аналогичные примеры самостоятельно. Нужно добиваться точного представления о том, что изучаешь. Полезно составлять опорные конспекты. При изучении материала по учебнику полезно в тетради (на специально отведенных полях) дополнять конспект лекций. Там же следует отмечать вопросы, выделенные студентом для консультации с преподавателем.

Выводы, полученные в результате изучения, рекомендуется в конспекте выделять, чтобы они при перечитывании записей лучше запоминались.

Опыт показывает, что многим студентам помогает составление листа опорных сигналов, содержащего важнейшие и наиболее часто употребляемые формулы и понятия. Такой лист помогает запомнить формулы, основные положения лекции, а также может служить постоянным справочником для студента.

Чтение научного текста является частью познавательной деятельности. Ее цель – извлечение из текста необходимой информации. От того на сколько осознанна читающим собственная внутренняя установка при обращении к печатному слову (найти нужные сведения, усвоить информацию полностью или частично, критически проанализировать материал и т.п.) во многом зависит эффективность осуществляемого действия.

Выделяют ***четыре основные установки в чтении научного текста***:

информационно-поисковый (задача – найти, выделить искомую информацию)

усваивающая (усилия читателя направлены на то, чтобы как можно полнее осознать и запомнить как сами сведения излагаемые автором, так и всю логику его рассуждений)

аналитико-критическая (читатель стремится критически осмыслить материал, проанализировав его, определив свое отношение к нему)

творческая (создает у читателя готовность в том или ином виде – как отправной пункт для своих рассуждений, как образ для действия по аналогии и т.п. – использовать суждения автора, ход его мыслей, результат наблюдения, разработанную методику, дополнить их, подвергнуть новой проверке).

Основные виды систематизированной записи прочитанного:

Аннотирование – предельно краткое связное описание просмотренной или прочитанной книги (статьи), ее содержания, источников, характера и назначения;

Планирование – краткая логическая организация текста, раскрывающая содержание и структуру изучаемого материала;

Тезирование – лаконичное воспроизведение основных утверждений автора без привлечения фактического материала;

Цитирование – дословное выписывание из текста выдержек, извлечений, наиболее существенно отражающих ту или иную мысль автора;

Конспектирование – краткое и последовательное изложение содержания прочитанного.

Конспект – сложный способ изложения содержания книги или статьи в логической последовательности. Конспект аккумулирует в себе предыдущие виды записи, позволяет всесторонне охватить содержание книги, статьи. Поэтому умение составлять план, тезисы, делать выписки и другие записи определяет и технологию составления конспекта.

Методические рекомендации по составлению конспекта:

1. Внимательно прочитайте текст. Уточните в справочной литературе непонятные слова. При записи не забудьте вынести справочные данные на поля конспекта;
2. Выделите главное, составьте план;
3. Кратко сформулируйте основные положения текста, отметьте аргументацию автора;
4. Законспектируйте материал, четко следуя пунктам плана. При конспектировании старайтесь выразить мысль своими словами. Записи следует вести четко, ясно.
5. Грамотно записывайте цитаты. Цитируя, учитывайте лаконичность, значимость мысли.

В тексте конспекта желательно приводить не только тезисные положения, но и их доказательства. При оформлении конспекта необходимо стремиться к емкости каждого предложения. Мысли автора книги следует излагать кратко, заботясь о стиле и выразительности написанного. Число дополнительных элементов конспекта должно быть логически обоснованным, записи должны распределяться в определенной

последовательности, отвечающей логической структуре произведения. Для уточнения и дополнения необходимо оставлять поля.

Овладение навыками конспектирования требует от студента целеустремленности, повседневной самостоятельной работы.

4.2. Методические рекомендации по подготовке к практическим и лабораторным занятиям

Для того чтобы практические и лабораторные занятия приносили максимальную пользу, необходимо помнить, что упражнение и решение задач проводятся по вычитанному на лекциях материалу и связаны, как правило, с детальным разбором отдельных вопросов лекционного курса. Следует подчеркнуть, что только после усвоения лекционного материала с определенной точки зрения (а именно с той, с которой он излагается на лекциях) он будет закрепляться на практических занятиях как в результате обсуждения и анализа лекционного материала, так и с помощью решения проблемных ситуаций, задач. При этих условиях студент не только хорошо усвоит материал, но и научится применять его на практике, а также получит дополнительный стимул (и это очень важно) для активной проработки лекции.

При самостоятельном решении задач нужно обосновывать каждый этап решения, исходя из теоретических положений курса. Если студент видит несколько путей решения проблемы (задачи), то нужно сравнить их и выбрать самый рациональный. Полезно до начала вычислений составить краткий план решения проблемы (задачи). Решение проблемных задач или примеров следует излагать подробно, вычисления располагать в строгом порядке, отделяя вспомогательные вычисления от основных. Решения при необходимости нужно сопровождать комментариями, схемами, чертежами и рисунками.

Следует помнить, что решение каждой учебной задачи должно доводиться до окончательного логического ответа, которого требует условие, и по возможности с выводом. Полученный ответ следует проверить способами, вытекающими из существа данной задачи. Полезно также (если возможно) решать несколькими способами и сравнить полученные результаты. Решение задач данного типа нужно продолжать до приобретения твердых навыков в их решении.

4.3. Методические рекомендации по самопроверке знаний

После изучения определенной темы по записям в конспекте и учебнику, а также решения достаточного количества соответствующих задач на практических занятиях и самостоятельно студенту рекомендуется, провести самопроверку усвоенных знаний, ответив на контрольные вопросы по изученной теме.

В случае необходимости нужно еще раз внимательно разобраться в материале.

Иногда недостаточность усвоения того или иного вопроса выясняется только при изучении дальнейшего материала. В этом случае надо вернуться назад и повторить плохо усвоенный материал. Важный критерий усвоения теоретического материала - умение решать задачи или пройти тестирование по пройденному материалу. Однако следует помнить, что правильное решение задачи может получиться в результате применения механически заученных формул без понимания сущности теоретических положений.

4.4. Методические рекомендации по написанию научных текстов (докладов, докладовов, эссе, научных статей и т.д.)

Перед тем, как приступить к написанию научного текста, важно разобраться, какова истинная цель вашего научного текста - это поможет вам разумно распределить свои силы и время.

Во-первых, сначала нужно определиться с идеей научного текста, а для этого необходимо научиться либо относиться к разным явлениям и фактам несколько критически

(своя идея – как иная точка зрения), либо научиться увлекаться какими-то известными идеями, которые нуждаются в доработке (идея – как оптимистическая позиция и направленность на дальнейшее совершенствование уже известного). Во-вторых, научиться организовывать свое время, ведь, как известно, свободное (от всяких глупостей) время – важнейшее условие настоящего творчества, для него наконец-то появляется время. Иногда именно на организацию такого времени уходит немалая часть сил и талантов.

Писать следует ясно и понятно, стараясь основные положения формулировать четко и недвусмысленно (чтобы и самому понятно было), а также стремясь структурировать свой текст. Каждый раз надо представлять, что ваш текст будет кто-то читать и ему захочется сориентироваться в нем, быстро находить ответы на интересующие вопросы (заодно представьте себя на месте такого человека). Понятно, что работа, написанная «сплошным текстом» (без заголовков, без выделения крупным шрифтом наиболее важным мест и т, п.), у культурного читателя должна вызывать брезгливость и даже жалость к автору (исключения составляют некоторые древние тексты, когда и жанр был иной и к текстам относились иначе, да и самих текстов было гораздо меньше – не то, что в эпоху «информационного взрыва» и соответствующего «информационного мусора»).

Объем текста и различные оформительские требования во многом зависят от принятых в конкретном учебном заведении порядков.

Доклад - это самостоятельное исследование студентом определенной проблемы, комплекса взаимосвязанных вопросов.

Доклад не должна составляться из фрагментов статей, монографий, пособий. Кроме простого изложения фактов и цитат, в доклад е должно проявляться авторское видение проблемы и ее решения.

Рассмотрим основные этапы подготовки

а студентом.

Выполнение доклада начинается с выбора темы.

Затем студент приходит на первую консультацию к руководителю, которая предусматривает:

- обсуждение цели и задач работы, основных моментов избранной темы;
- консультирование по вопросам подбора литературы;
- составление предварительного плана.

Следующим этапом является работа с литературой. Необходимая литература подбирается студентом самостоятельно.

После подбора литературы целесообразно сделать рабочий вариант плана работы. В нем нужно выделить основные вопросы темы и параграфы, раскрывающие их содержание.

Составленный список литературы и предварительный вариант плана уточняются, согласуются на очередной консультации с руководителем.

Затем начинается следующий этап работы - изучение литературы. Только внимательно читая и конспектируя литературу, можно разобраться в основных вопросах темы и подготовиться к самостоятельному (авторскому) изложению содержания доклада. Конспектируя первоисточники, необходимо отразить основную идею автора и его позицию по исследуемому вопросу, выявить проблемы и наметить задачи для дальнейшего изучения данных проблем.

Систематизация и анализ изученной литературы по проблеме исследования позволяют студенту написать работу.

Рабочий вариант текста доклада предоставляется руководителю на проверку. На основе рабочего варианта текста руководитель вместе со студентом обсуждает возможности доработки текста, его оформление. После доработки доклад сдается на кафедру для его оценивания руководителем.

Требования к написанию доклада

Написание 1 доклада является обязательным условием выполнения плана СРС по любой дисциплине профессионального цикла.

Тема доклада может быть выбрана студентом из предложенных в рабочей программе или фонде оценочных средств дисциплины, либо определена самостоятельно, исходя из интересов студента (в рамках изучаемой дисциплины). Выбранную тему необходимо согласоваться с преподавателем.

Доклад должен быть написан научным языком.

Объем доклада должен составлять 20-25 стр.

Структура доклада:

- Введение (не более 3-4 страниц). Во введении необходимо обосновать выбор темы, ее актуальность, очертить область исследования, объект исследования, основные цели и задачи исследования.
- Основная часть состоит из 2-3 разделов. В них раскрывается суть исследуемой проблемы, проводится обзор мировой литературы и источников Интернет по предмету исследования, в котором дается характеристика степени разработанности проблемы и авторская аналитическая оценка основных теоретических подходов к ее решению. Изложение материала не должно ограничиваться лишь описательным подходом к раскрытию выбранной темы. Оно также должно содержать собственное видение рассматриваемой проблемы и изложение собственной точки зрения на возможные пути ее решения.
- Заключение (1-2 страницы). В заключении кратко излагаются достигнутые при изучении проблемы цели, перспективы развития исследуемого вопроса
- Список использованной литературы (не меньше 10 источников), в алфавитном порядке, оформленный в соответствии с принятыми правилами. В список использованной литературы рекомендуется включать работы отечественных и зарубежных авторов, в том числе статьи, опубликованные в научных журналах в течение последних 3-х лет и ссылки на ресурсы сети Интернет.
- Приложение (при необходимости).

Требования к оформлению:

- текст с одной стороны листа;
- шрифт Times New Roman;
- кегль шрифта 14;
- межстрочное расстояние 1,5;
- поля: сверху 2,5 см, снизу – 2,5 см, слева - 3 см, справа 1,5 см;
- доклад должен быть представлен в сброшюрованном виде.

Порядок защиты доклада:

Защита доклада проводится на практических занятиях, после окончания работы студента над ним и исправления всех недочетов, выявленных преподавателем в ходе консультаций. На защиту доклада отводится 5-7 минут времени, в ходе которого студент должен показать свободное владение материалом по заявленной теме. При защите доклада приветствуется использование мультимедиа-презентации.

Оценка доклада

Доклад оценивается по следующим критериям:

• соблюдение требований к его оформлению;

• необходимость и достаточность для раскрытия темы приведенной в тексте доклада информации;

• умение студента свободно излагать основные идеи, отраженные в докладе;

• способность студента понять суть задаваемых преподавателем и сокурсниками вопросов и сформулировать точные ответы на них.

Критерии оценки:

Оценка «отлично» выставляется студенту, если в докладе студент исчерпывающе, последовательно, четко и логически стройно излагает материал; свободно справляется с задачами, вопросами и другими видами применения знаний; использует для написания

доклада современные научные материалы; анализирует полученную информацию; проявляет самостоятельность при написании доклада.

Оценка «хорошо» выставляется студенту, если качество выполнения доклада достаточно высокое. Студент твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопросы по теме доклада.

Оценка «удовлетворительно» выставляется студенту, если материал доклада излагается частично, но пробелы не носят существенного характера, студент допускает неточности и ошибки при защите доклада, дает недостаточно правильные формулировки, наблюдаются нарушения логической последовательности в изложении материала.

Оценка «неудовлетворительно» выставляется студенту, если он не подготовил доклад или допустил существенные ошибки. Студент неуверенно излагает материал доклада, не отвечает на вопросы преподавателя.

4.5. Методические рекомендации по выполнению исследовательских проектов

Исследовательская проектная работа – это групповая работа, для выполнения которой необходим выбор и приложение научной методики к поставленной задаче, получение собственного теоретического или экспериментального материала, на основании которого необходимо провести анализ и сделать выводы об исследуемом явлении. Выполнение проекта – это всегда коллективная, творческая практическая работа, предназначенная для получения определенного продукта или научно-технического результата. Такая работа подразумевает четкое, однозначное формирование поставленной задачи, определение сроков выполнения намеченного, определение требований к разрабатываемому объекту.

Выполнение 1 группового проекта является обязательным условием выполнения самостоятельной работы по любой дисциплине профессионального цикла. Тема проектного задания может быть выбрана студентом из предложенных в рабочей программе или фонде оценочных средств дисциплины, либо определена самостоятельно, исходя из интересов студента (в рамках изучаемой дисциплины). Выбранную тему необходимо согласоваться с преподавателем.

Требования по выполнению и оформлению проекта

При выполнении проекта приветствуется работа в группе (2-3 человека). Проект – это исследовательская работа, в ходе которой студенты должны продемонстрировать владение навыками научного исследования, умения проводить анализ, обобщать информацию, делать выводы, предлагать свои решения проблемы, рассматриваемой в проекте.

При подготовке материалов проекта студенты должны продемонстрировать владение современными методами компьютерной обработки данных.

Критерии оценки работы участника проекта.

Для каждого из участников проекта оцениваются:

• профессиональные теоретические знания в соответствующей области;

• умение работать со справочной и научной литературой, осуществлять поиск необходимой информации в Интернет;

• умение работать с техническими средствами;

• умение пользоваться соответствующими выполняемому проекту информационными технологиями;

• умение готовить материалы проекта для презентации: составлять и редактировать тексты, формировать презентацию проекта;

• умение работать в команде;

• умение публично представлять результаты собственной деятельности;

• коммуникабельность, инициативность, творческие способности.

Критерии выставления оценки участникам проекта

Оценка	Профессиональные компетенции	Компетенции, связанные с использованием соответствующих выполняемому проекту технических средств и информационных технологий	Иные универсальные компетенции (коммуникабельность, инициативность, умение работать в «команде», управленческие навыки и т.д.)	Отчетность
«Отлично»	Работа выполнена на высоком профессиональном уровне. Представленный материал в основном фактически верен, допускаются негрубые фактические неточности. Студент свободно отвечает на вопросы, связанные с проектом.	Технические средства и информационные технологии освоены и использованы для реализации проекта полностью	Студент проявил инициативу, творческий подход, способность к выполнению сложных заданий, навыки работы в коллективе, организационные способности.	Проект представлен полностью и в срок.
«Хорошо»	Работа выполнена на достаточно высоком профессиональном уровне. Допущено до 4–5 фактических ошибок. Студент отвечает на вопросы, связанные с проектом, но недостаточно полно.	Обнаруживаются некоторые ошибки в использовании соответствующих технических средств и информационных технологий	Студент достаточно полно, но без инициативы и творческих находок выполнил возложенные на него задачи.	Проект представлен достаточно полно и в срок, но с некоторыми недоработками.
«Удовлетворительно»	Уровень недостаточно высок. Допущено до 8 фактических ошибок. Студент может ответить лишь на некоторые из заданных вопросов, связанных с проектом.	Обнаруживает недостаточное владение навыками работы с техническими средствами и соответствующими информационными технологиями	Студент выполнил большую часть возложенной на него работы.	Проект сдан со значительным опозданием (более недели) и не полностью
«Неудовлетворительно»	Работа не выполнена или выполнена на низком уровне. Допущено более 8 фактических ошибок. Ответы на связанные с проектом вопросы	Навыков работы с техническими средствами нет, информационные технологии не освоены	Студент практически не работал, не выполнил свои задачи или выполнил лишь отдельные не существенные	Проект не сдан.

Оценка	Профессиональные компетенции	Компетенции, связанные с использованием соответствующих выполняемому проекту технических средств и информационных технологий	Иные универсальные компетенции (коммуникабельность, инициативность, умение работать в «команде», управленческие навыки и т.д.)	Отчетность
	обнаруживают непонимание предмета и отсутствие ориентации в материале проекта.		поручения в групповом проекте.	

Студенты должны: защитить проект в режиме презентации, предъявить файлы выполненного проекта, уметь рассказать о технологиях, использованных ими при выполнении проекта, дать оценку работы каждого члена группы (*если проект групповой*).

4.6. Методические рекомендации по подготовке к экзаменам и зачетам

Изучение многих общепрофессиональных и специальных дисциплин завершается экзаменом. Подготовка к экзамену способствует закреплению, углублению и обобщению знаний, получаемых, в процессе обучения, а также применению их к решению практических задач. Готовясь к экзамену, студент ликвидирует имеющиеся пробелы в знаниях, углубляет, систематизирует и упорядочивает свои знания. На экзамене студент демонстрирует то, что он приобрел в процессе обучения по конкретной учебной дисциплине.

Экзаменационная сессия - это серия экзаменов, установленных учебным планом. Между экзаменами интервал 3-4 дня. Не следует думать, что 3-4 дня достаточно для успешной подготовки к экзаменам.

В эти 3-4 дня нужно систематизировать уже имеющиеся знания. На консультации перед экзаменом студентов познакомят с основными требованиями, ответят на возникшие у них вопросы. Поэтому посещение консультаций обязательно.

Требования к организации подготовки к экзаменам те же, что и при занятиях в течение семестра, но соблюдаться они должны более строго. Во-первых, очень важно соблюдение режима дня; сон не менее 8 часов в сутки, занятия заканчиваются не позднее, чем за 2-3 часа до сна. Оптимальное время занятий - утренние и дневные часы. В перерывах между занятиями рекомендуются прогулки на свежем воздухе, неутомительные занятия спортом. Во-вторых, наличие хороших собственных конспектов лекций. Даже в том случае, если была пропущена какая-либо лекция, необходимо во время ее восстановить (переписать ее на кафедре), обдумать, снять возникшие вопросы для того, чтобы запоминание материала было осознанным. В-третьих, при подготовке к экзаменам у студента должен быть хороший учебник или конспект литературы, прочитанной по указанию преподавателя в течение семестра. Здесь можно эффективно использовать листы опорных сигналов.

Вначале следует просмотреть весь материал по сдаваемой дисциплине, отметить для себя трудные вопросы. Обязательно в них разобраться. В заключение еще раз целесообразно повторить основные положения, используя при этом листы опорных сигналов.

Систематическая подготовка к занятиям в течение семестра позволит использовать время экзаменационной сессии для систематизации знаний.

Контроль самостоятельной работы студентов

Контроль самостоятельной работы проводится преподавателем в аудитории.

Предусмотрены следующие виды контроля: собеседование, оценка доклада, оценка презентации, оценка участия в круглом столе, оценка выполнения проекта.

Подробные критерии оценивания компетенций приведены в Фонде оценочных средств для проведения текущей и промежуточной аттестации.

Список литературы для выполнения СРС

Перечень основной литературы:

1. Уэс, Маккинли. Python и анализ данных Электронный ресурс / Маккинли Уэс ; пер. А. А. Слинкин. - Python и анализ данных,2021-04-19. - Саратов : Профобразование, 2017. - 482 c. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 978-5-4488-0046-7, экземпляров неограниченно.

2. Сузи, Р.А. Язык программирования Python Электронный ресурс : учебное пособие / Р.А. Сузи. - Язык программирования Python,2020-07-28. - Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. - 350 c. - Книга находится в базовой версии ЭБС IPRbooks. - ISBN 5-9556-0058-2, экземпляров неограниченно

Перечень дополнительной литературы:

3. Стенли, Липпман. Язык программирования C++ Электронный ресурс : Полное руководство / Липпман Стенли, Лажойе Жози ; пер. А. Слинкин. - Язык программирования C++,2021-04-19. - Саратов : Профобразование, 2017. - 1104 c. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 978-5-4488-0136-5, экземпляров неограниченно

4. Седжвик, Р. Алгоритмы на C++ / Р. Седжвик. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 1773 с., экземпляров неограниченно

Методическая литература:

1. Методические рекомендации для самостоятельной работы студентов по дисциплине «Разработка диалоговых систем»
2. Методические указания к лабораторным работам по дисциплине «Разработка диалоговых систем»

Интернет-ресурсы:

1. http://el.ncfu.ru/ – система управления обучением ФГАОУ ВО СКФУ. Дистанционная поддержка дисциплины «Разработка диалоговых систем».
2. https://archive.ics.uci.edu/ml/index.html – Репозиторий наборов данных для машинного обучения (Центр машинного обучения и интеллектуальных систем).
3. https://www.kaggle.com – Портал и система проведения соревнований по проблемам анализа данных.
4. https://www.mockaroo.com – Сайт для генерации наборов данных.