МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего

образования

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Отчет по заданию №5

по дисциплине

«ГЛУБИННОЕ ОБУЧЕНИЕ»

**«Задание №5 – Анализ текстовых данных средствами BERT и стека LSTM»**

Выполнил

Студент гр. 932226

\_\_\_\_\_\_\_\_\_\_\_Савенкова М. М. «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г

Проверил

\_\_\_\_\_\_\_\_ к.т.н. Аксёнов С.В.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Томск – 2023

Оглавление

[1 Индивидуальное задание 3](#_Toc154595447)

[2 Цель работы 4](#_Toc154595448)

[3 Листинг программ 5](#_Toc154595449)

[4 Входные данные 6](#_Toc154595450)

[5 Построение моделей 7](#_Toc154595451)

[5.1 BERT 7](#_Toc154595452)

[5.1.1 Параметры архитектуры и обучения 8](#_Toc154595453)

[5.1.2 Полученные результаты 9](#_Toc154595454)

[5.2 Fine-tuned BERT 10](#_Toc154595455)

[5.2.1 Параметры архитектуры и обучения 10](#_Toc154595456)

[5.2.2 Полученные результаты 10](#_Toc154595457)

[5.3 LSTM 11](#_Toc154595458)

[5.3.1 Параметры архитектуры и обучения 11](#_Toc154595459)

[5.3.2 Полученные результаты 12](#_Toc154595460)

[5.4 Двунаправленная LSTM 13](#_Toc154595461)

[5.4.1 Параметры архитектуры и обучения 13](#_Toc154595462)

[5.4.2 Полученные результаты 14](#_Toc154595463)

[6 Выводы 16](#_Toc154595464)

# 1 Индивидуальное задание

Задачи – Обучить нейросетевые модели BERT, стека LSTM, двунаправленную LSTM, предсказывающие авторство текстов по фрагменту текстов.

Выбрать 4-x поэтов из списка: Цветаева М.И., Некрасов Н.А., Лермонтов М.Ю., Тютчев Ф.И., Фет А.А., Бунин И.А., Есенин С.А., Маяковский В.В., Бродский И.А., Ахматова А.А., Блок А.А., Заболоцкий Н.А. (Можно взять любых других писателей). Для каждого выбранного поэта взять не менее 50 фрагментов из произведений, каждый фрагмент содержит не менее 100 символов.

Получить токенизатор и обученную модель BERT c HuggingFace.

Выполнить загрузку и предварительную обработку данных из текстов.

Разделить каждую выборку на обучающую, тестовую и валидационную.

Добавить к архитектуре BERT полносвязный (или два полносвязных) слой(я) и обучить полученную модель в следующих вариантах:

1) Заморозить веса в BERT и настроить добавленный(е) полносвязный(ые) слой(и). (Это – модель 1а)

2) Выполнить Fine-tuning полученной архитектуры. (Это – модель 1b)

Обучить вторую модель – стек LSTM (две последовательные ячейки LSTM, добавить Dropout) для классификации текстовых данных.

Обучить третью модель – двунаправленную ячейку LSTM для классификации текстовых данных.

Привести кривые изменения функции потерь (Loss) и верности (Accuracy) во время обучения на обучающем и валидационном наборах для всех моделей.

Привести результаты обучения моделей и сделать выводы.

# 2 Цель работы

Цель работы – получить навыки обучения и использования архитектур BERT, стека LSTM и двунаправленной LSTM для задач классификации текстовых данных.

# 3 Листинг программ

Листинги программ представлены в отдельном файле: Poetry\_classification\_(Savenkova\_932226).ipynb.

# 4 Входные данные

В качестве входных данных был выбран датасет «19 000 Russian poems», расположенный по ссылке <https://www.kaggle.com/datasets/grafstor/19-000-russian-poems>.

Данные представлены в табличном виде. Набор содержит три поля: автор стихотворения, название стихотворения и текст стихотворения.

Из набора были извлечены данные четырёх авторов – Есенина, Лермонтова, Ахматовой и Блока. Для обучения из датасета были отобраны стихотворения длиной более 100 символов.

Так как имелся дисбаланс количества стихотворений для каждого автора (от 300 до 1400 записей), был произведен ресэмплинг и в каждом классе было отобрано 500 объектов.

Также из набора данных были удалены все строки с пропущенными значениями, а сами тексты стихотворений были очищены от знаков пунктуации и приведены к нижнему регистру.

# 5 Построение моделей

В ходе выполнения данной лабораторной работы были построены четыре нейросетевых модели:

* BERT с заключительными полносвязными слоями;
* Fine-tuned BERT;
* Стек LSTM;
* Двунаправленная LSTM.

5.1 BERT

BERT (Bidirectional Encoder Representations from Transformers) –архитектура нейронной сети, представляющая собой двунаправленную модель кодировщика на основе трансформеров.

Основные особенности архитектуры BERT:

* Двунаправленный контекст. BERT учитывает контекст как слева, так и справа от каждого слова.
* Маскирование слов. BERT обучается на входных данных, в которых случайно замаскирована часть слов и модель должна предсказать маскированные токены, что позволяет модели обучаться на основе контекста, учитывая окружающие слова.
* Сегментные эмбеддинги (эмбеддинги, используемые для различных сегментов входа). BERT использует сегментные эмбеддинги для разделения предложений в парах предложений, которые играют важную роль в задачах понимания текста.

Для использования BERT необходимо предварительно токенизировать текст. Токенизация текста является процессом разделения текста на отдельные токены или слова.

Текст разделяется на отдельные токены, а затем каждый токен преобразуется в числовое представление, чтобы его можно было передать в модель BERT. Токенизация важна, так как BERT требует фиксированный размер входных данных, поэтому длинные тексты должны быть разделены на более мелкие части.

BERT использует так называемые "WordPiece" токены, которые могут включать отдельные слова, корни слов или суффиксы. Это позволяет учесть особенности морфологии языка и улучшить общую способность модели понимать и обрабатывать текст.

5.1.1 Параметры архитектуры и обучения

Нейронная сеть состоит из слоя BERT и двух полносвязных слоёв со 128 и 4 (количество авторов) нейронами соответственно. Функция активации первого полносвязного слоя – relu, выходного – softmax.

В качестве оптимизатора процесса обучения был выбран Adam с коэффициентом обучения равным 0.01 (значение 0.001 или использование RMSprop с обоими значениями негативно сказалось на процессе обучения).

Функция потерь – SparseCategoricalCrossentropy, выбранная метрика – SparseCategoricalAccuracy.

Обучение производилось на 60 эпохах. В процессе обучения работали две callback-функции: одна сохраняла модель с лучшими значениями точности, вторая – останавливала обучение, если на протяжении 30 эпох результат не улучшится.

5.1.2 Полученные результаты

На рисунке 1 представлены графики точности и процесса обучения на обучающей и валидационной выборках.

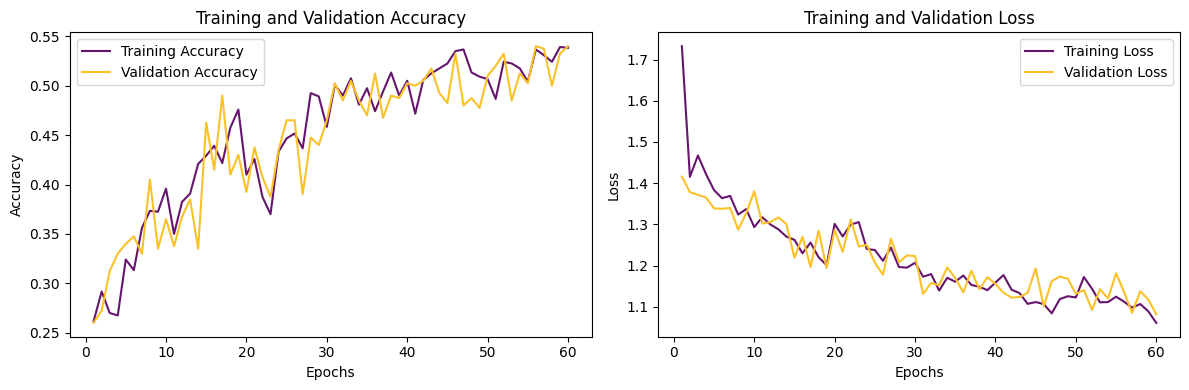


Рисунок 1 — Графики процесса обучения

На тестовой выборке точность составила 0.5250.

5.2 Fine-tuned BERT

5.2.1 Параметры архитектуры и обучения

Для fine-tuning были разморожены 4 последних слоя BERT из 12. Остальные параметры архитектуры и обучения остались прежними.

5.2.2 Полученные результаты

На рисунке 2 представлены графики точности и процесса обучения на обучающей и валидационной выборках.

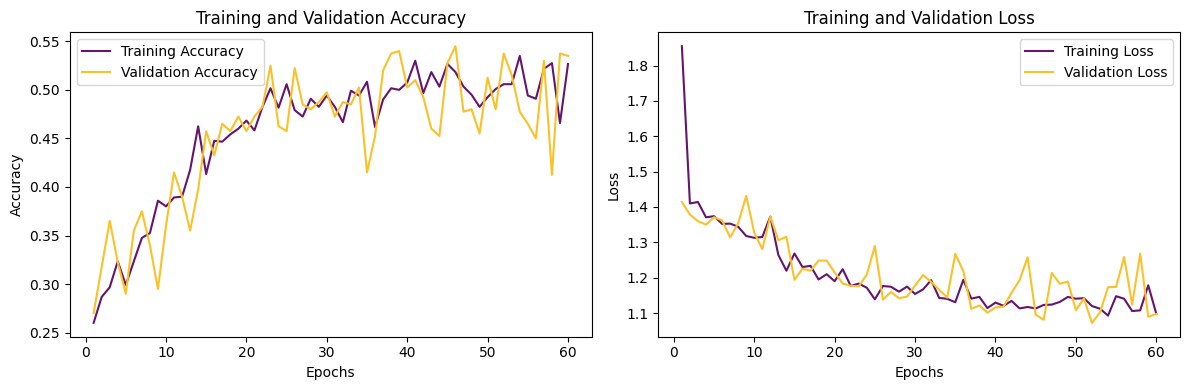


Рисунок 2 — Графики процесса обучения

На тестовой выборке точность составила 0.5425.

Вопреки ожиданиям, точность увеличилась лишь на сотые значения.

5.3 LSTM

Нейронные сети архитектуры LSTM (Long Short-Term Memory) используются для распознавания и классификации текста, так как LSTM блоки обеспечивают модели сохранение информации на протяжении длинных последовательностей и могут эффективно обрабатывать текстовые данные с учетом контекста.

5.3.1 Параметры архитектуры и обучения

Нейронная сеть состоит из следующих слоёв:

* Embedding – специальный входной слой, позволяющий выполнить эмбеддинг текстовых данных, то есть преобразовать каждое слово в числовой вектор определенной длины.
* LSTM – слой LSTM с 256 ячейками, возвращающий последовательности.
* Dropout – слой прореживания с параметром 0.1.
* LSTM – слой LSTM с 256 ячейками.
* Dropout – слой прореживания с параметром 0.001.
* Dense – полносвязный слой с 4 нейронами (количество поэтов) и функцией активации softmax.

В качестве оптимизатора процесса обучения был выбран Adam с коэффициентом обучения равным 0.01 (значение 0.001 или использование RMSprop с обоими значениями негативно сказалось на процессе обучения).

Функция потерь – SparseCategoricalCrossentropy, выбранная метрика – SparseCategoricalAccuracy.

Обучение производилось на 60 эпохах. В процессе обучения работали две callback-функции: одна сохраняла модель с лучшими значениями точности, вторая – останавливала обучение, если на протяжении 20 эпох результат не улучшится.

5.3.2 Полученные результаты

На рисунке 3 представлены графики точности и процесса обучения на обучающей и валидационной выборках.

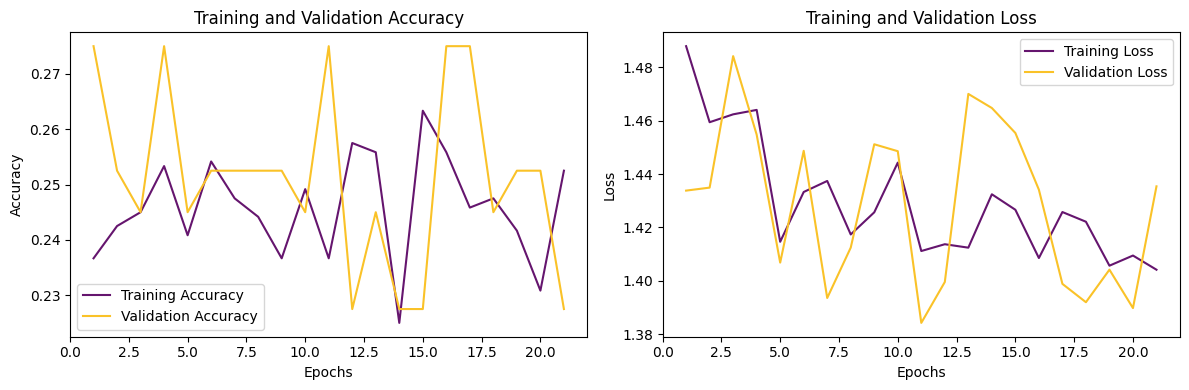


Рисунок 3 — Графики процесса обучения

На тестовой выборке точность составила 0.2825.

5.4 Двунаправленная LSTM

Двунаправленная LSTM (BiLSTM) является модификацией рекуррентной нейронной сети LSTM (Long Short-Term Memory), которая обрабатывает последовательности данных. Она состоит из двух LSTM слоев: одного, который просматривает последовательность данных в прямом порядке (forward LSTM), и другого, который просматривает последовательность в обратном порядке (backward LSTM).

Forward LSTM анализирует последовательность данных слева направо, постепенно обрабатывая каждый элемент и сохраняя информацию о контексте. Backward LSTM работает в обратном порядке, начиная с последнего элемента последовательности, и также сохраняет информацию о контексте. Затем два направления объединяются путем конкатенации выходных векторов каждого LSTM слоя.

Использование BiLSTM позволяет модели учитывать информацию не только из прошлого, но также и из будущего, что может помочь в задачах, требующих понимания контекста и зависимостей в последовательностях данных. BiLSTM широко применяется в области обработки естественного языка (Natural Language Processing), в задачах, таких как машинный перевод, определение тональности, именованное сущностное распознавание и другие.

5.4.1 Параметры архитектуры и обучения

Нейронная сеть состоит из следующих слоёв:

* Embedding – специальный входной слой, позволяющий выполнить эмбеддинг текстовых данных, то есть преобразовать каждое слово в числовой вектор определенной длины.
* Bidirectional LSTM – слой LSTM с 128 ячейками, возвращающий последовательности.
* Dropout – слой прореживания с параметром 0.2.
* Bidirectional LSTM – слой LSTM с 128 ячейками.
* Dense – полносвязный слой с 4 нейронами (количество поэтов) и функцией активации softmax.

В качестве оптимизатора процесса обучения был выбран Adam.

Функция потерь – SparseCategoricalCrossentropy, выбранная метрика – SparseCategoricalAccuracy.

Обучение производилось на 60 эпохах. В процессе обучения работали две callback-функции: одна сохраняла модель с лучшими значениями точности, вторая – останавливала обучение, если на протяжении 20 эпох результат не улучшится.

5.4.2 Полученные результаты

На рисунке 4 представлены графики точности и процесса обучения на обучающей и валидационной выборках.

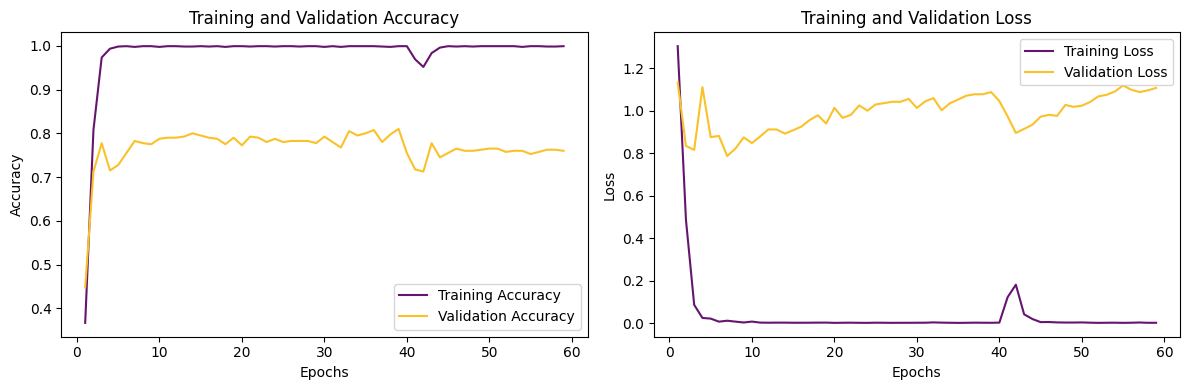


Рисунок 4 — Графики процесса обучения

На тестовой выборке точность составила 0.7475.

# 6 Выводы

Качество предсказаний модели, анализирующей данные в виде текста, сильно зависит от того, насколько качественно был «очищен» текст. Существует несколько способов обработать данные:

* Удаление стоп-слов: удаление наиболее часто встречающихся слов, которые не несут полезной смысловой информации, например, артиклей, предлогов.
* Приведение к нормальной форме: приведение слов к их базовой форме, например, применение стемминга или лемматизации. Стемминг обрезает слова до их основы, а лемматизация приводит слова к их словарной форме.
* Очистка данных: удаление символов пунктуации, преобразование текста в нижний регистр, удаление лишних пробелов или специальных символов.
* Выделение признаков: определение и извлечение особенностей текста, таких как n-граммы, частоты встречаемости слов, синтаксические шаблоны и другие параметры, которые могут быть полезны для модели.
* Морфологический анализ: определение грамматической структуры и морфологических характеристик слов. Это может быть полезно для определения смысла слова в контексте.

Поскольку в данной работе текст был только очищен от знаков пунктуации и приведен к нижнему регистру, это могло послужить причиной невысокой точности.

Также причиной таких результатов могли послужить особенности данных текстов – стихотворений конкретных авторов. При визуализации облака слов (рисунок 5) можно увидеть пересечение тем.

|  |  |
| --- | --- |
| C:\MARINA\STUDY\Магистратура\семестр 3\глубинное обучение\задание 5\esen.png | C:\MARINA\STUDY\Магистратура\семестр 3\глубинное обучение\задание 5\lerm.png |
| C:\MARINA\STUDY\Магистратура\семестр 3\глубинное обучение\задание 5\ahma.png | C:\MARINA\STUDY\Магистратура\семестр 3\глубинное обучение\задание 5\blok.png |
| Рисунок 5 — Облака слов | |