**Министерство науки и высшего образования Российской Федерации**
**Федеральное государственное бюджетное образовательное учреждение**
**высшего образования**
**«Московский государственный технический университет имени Н.Э. Баумана**
**(национальный исследовательский университет)»**
**(МГТУ им. Н.Э. Баумана)**

УТВЕРЖДАЮ
Заведующий кафедрой ИУ-7
_____ И. В. Рудаков
«16» сентября 2023 г.

# З А Д А Н И Е
## на выполнение научно-исследовательской работы

по теме

**«Сравнение существующих методов классификации текста»**

Студент группы **ИУ7И-74Б**

**Динь Вьет Ань**

Направленность НИР

**учебная**

Источник тематики

**НИР кафедры**

График выполнения НИР: 25% к 6 нед., 50% к 9 нед., 75% к 12 нед., 100% к 15 нед.

***Техническое задание***

*Провести обзор существующих методов классификации текста. Провести анализ предметной области классификации текста. Сформулировать критерии оценки результата классификации текста и на их основе сравнить рассмотренных методов.*

***Оформление научно-исследовательской работы:***

Расчетно-пояснительная записка на **12-20** листах формата А4.

***Перечень графического (иллюстративного) материала:***

Презентация на **6-10** слайдах.

Дата выдачи задания «16» сентября 2023 г.

| | | |
|---|---|---|
| **Руководитель НИР** | _____ | **Павельев А. А.** |
| | (Подпись, дата) | (Фамилия И. О.) |
| **Студент** | _____ | **Динь Вьет Ань** |
| | (Подпись, дата) | (Фамилия И. О.) |

# РЕФЕРАТ

Расчетно-пояснительная записка 23 с., 1 рис..

Ключевые слова: классификация текста, анализ текста, текстовое представление, категоризация текста, анализ текста, классификация документов, машинное обучение, глубокое обучение.

# СОДЕРЖАНИЕ

**РЕФЕРАТ** **3**

**ВВЕДЕНИЕ** **5**

**1 Анализ предметной области** **6**

1.1 Задача классификации текстов . . . . . 6

1.2 Процесс классификации текста . . . . . 7

**2 Предобработка текста и извлечение признаков** **8**

2.1 Необходимость предобработки текста . . . . . 8

2.2 Очистка и предобработка текста . . . . . 8

2.2.1 Токенизация . . . . . 8

2.2.2 Удаление шума (стоп-слов) . . . . . 8

2.3 Извлечения признаков . . . . . 9

2.3.1 Bag of Words (BoW) . . . . . 9

2.3.2 Word2Vec . . . . . 10

2.3.3 GloVe . . . . . 10

2.3.4 FastText . . . . . 11

**3 Существующие методы классификации текста** **12**

3.1 KNN . . . . . 12

3.2 Метод опорных векторов . . . . . 13

3.3 Decision Tree and Random Forest . . . . . 13

3.4 CNN . . . . . 14

**4 Сравнение существующих методов классификации текста** **16**

**ЗАКЛЮЧЕНИЕ** **19**

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ** **20**

**ПРИЛОЖЕНИЕ А** **23**

# ВВЕДЕНИЕ

В последние годы наблюдается экспоненциальный рост количества сложных документов и текстов, требующих более глубокого понимания, чтобы иметь возможность точно классифицировать тексты во многих приложениях. Многие подходы к машинному обучению достигли превосходных результатов в обработке естественного языка. Успех этих методов зависит от их способности понимать сложные модели и нелинейные связи внутри данных. Однако поиск подходящих структур, архитектур и методов классификации текста является непростой задачей.

Цель данной рабате — проведение краткого обзора методов классификации текста, формулирование критерий оценки классификации текста и на их основе сравнить рассмотренных методов.

В рамках выполнения работы необходимо решить следующие задачи:

— провести анализ предметной области;
— провести краткий обзор существующих методов классификации текста;
— сравнить рассмотренных методов по преимуществам и недостаткам.

# 1 Анализ предметной области

## 1.1 Задача классификации текстов

Классификация текста —– это процесс присвоения предопределенной категории или метки предложениям, абзацам, текстовым отчетам или неструктурированного текста.

За последние несколько десятилетий проблемы классификации текста широко изучались и решались во многих практических приложениях. Многие исследователи теперь заинтересованы в разработке приложений, использующих преимущества методов классификации текста, особенно в связи с недавними достижениями в области обработки естественного языка.

Некоторые задачи классификации текста в реальном [1]:

1) анализ настроений — задача понимания аффективных состояний и субъективной информации, содержащейся в фрагменте текста;
2) маркировка тем — задача распознавания одной или нескольких тем фрагмента текста (т. е. его тем);
3) классификация новостей — задача присвоения новостям категорий;
4) ответ на вопрос — задача выбора ответа на вопрос, выбора из потенциальных предложений - кандидатов (обычно извлекаемых из контекстного документа);
5) вывод на естественном языке — задача определения того, влекут ли два предложения друг друга (классификация, происходит ли следование в одном из двух направлений или ни в одном из них);
6) распознавание именованных объектов — задача поиска именованных объектов в неструктурированном тексте и маркировка их заранее определенными категориями;
7) синтаксический анализ — серия задач, связанных с прогнозированием морфо - синтаксических свойств слов.

## 1.2 Процесс классификации текста

Большинство процессов классификации текста, обычно, состоят из следующих трёх шагов: предобработка текста, извлечение признаков и классификация текста с помощью некоторого алгоритма.

Ниже, на рисунке 1, представлены этапы процесса классификации текста.

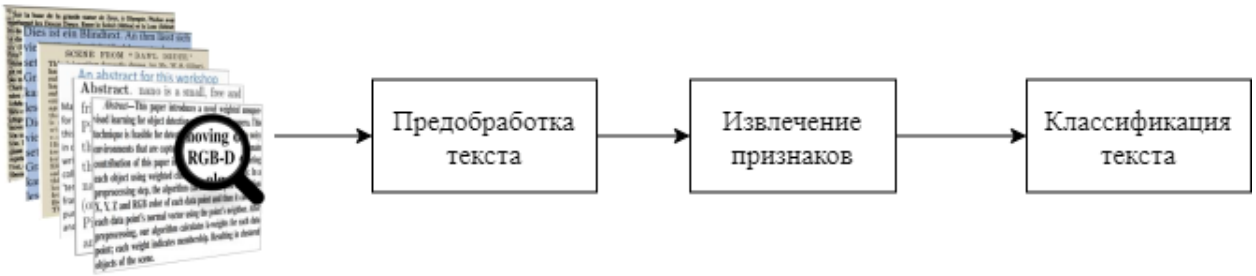


Рисунок 1 – Этапы процесса классификации текста.

Система классификации текстов содержит четыре различных уровня области применения, которые можно применять.

1) Уровень документа. На этом уровне документа алгоритм получает соответствующие категории полного документа.
2) Уровень абзаца. На этом уровне абзаца алгоритм получает соответствующие категории одного абзаца (части документа).
3) Уровень предложения. На этом уровне предложения получают соответствующие категории одного предложения (части абзаца).
4) Уровень подпредложения. На этом уровне подпредложения алгоритм получает соответствующие категории подвыражений внутри предложения.

# 2 Предобработка текста и извлечение признаков

## 2.1 Необходимость предобработки текста

Входные данные для задач на естественном языке состоят из необработанного неструктурированного текста. Текстовая информация, в отличие от других типов данных, таких как изображения или временные ряды, не обладает числовым представлением, поэтому перед подачей ее в какой-либо классификатор она должна быть спроецирована в соответствующее пространство признаков. Поэтому процедуры предварительной обработки имеют особое значение, поскольку без них не существует основы ни для процедур выделения признаков, ни для алгоритмов классификации.

## 2.2 Очистка и предобработка текста

### 2.2.1 Токенизация

Токенизация — самая базовая операция предварительной обработки, которую необходимо применить к тексту. Этот процесс определяет уровень детализации анализа текстовых данных и в целом может быть описан как процесс предварительной обработки, целью которого является разделение текстового потока на слова, фразы, символы или другие значимые элементы, называемые токенами. Разделение основано на правилах и может быть простым, как разделение пробелами или знаками препинания.

Например, предложение:

After eating, I decided to start working.

В данном случае токены следующие:

{“After”, “eating”, “I”, “decided”, “to”, “start”, “working”}.

### 2.2.2 Удаление шума (стоп-слов)

Стоп-слова (текстовые шумы) —– это неинформативные слова, которые встречаются в большом количестве, но не имеют семантического значения. Например, слова “и”, “в”, “только” не несут никакой ценности и только добавляют шум в данные. Множество токенов, полученный после процесса токенизации,

может содержать множество ненужных или бессмысленных элементов. Удаление стоп-слов необходимо, поскольку это уменьшит количество различных элементов в пространстве признаков.

Обычно тексты содержат разные грамматические формы одного и того же слова, а также могут встречаться однокоренные слова. Лемматизация и стемминг преследуют цель привести все встречающиеся словоформы к одной, нормальной словарной форме.

Стемминг — это грубый эвристический процесс, который отрезает “лишнее” от корня слов, часто это приводит к потере словообразовательных суффиксов.

Лемматизация — это более тонкий процесс, который использует словарь и морфологический анализ, чтобы в итоге привести слово к его канонической форме — лемме.

Отличие в том, что стеммер (конкретная реализация алгоритма стемминга) действует без учёта контекста и, соответственно, не делает разницы между словами, которые имеют разный смысл в зависимости от части речи. Однако у стеммеров есть своё преимущество — они работают быстрее.

## 2.3 Извлечения признаков

### 2.3.1 Bag of Words (BoW)

Модель Bag of Words (модель BoW) —– это уменьшенное и упрощенное представление текстового документа из выбранных частей текста на основе определенных критериев, таких как частота слов.

При всей простоте реализации данный подход имеет ряд недостатков:

— для больших наборов текстов размерность словаря, а, следовательно, и размерность вектора, представляющего текст, может исчисляться сотнями тысяч, а иногда и миллионами;

— не учитывается контекст слова в документе.

### 2.3.2 Word2Vec

Word2Vec (Word to Vector) —– это метод, используемый для преобразования слов в векторы, тем самым фиксируя их значение, семантическое сходство и взаимосвязь с окружающим текстом. Этот метод помогает компьютерам изучать контекст и значение выражений и ключевых слов из больших текстовых коллекций, таких как новостные статьи и книги.

Основная идея Word2Vec состоит в том, чтобы представить каждое слово как многомерный вектор, где положение вектора в этом многомерном пространстве отражает значение слова.

Word2Vec использует модель мелкой нейронной сети для изучения значения слов из большого массива текстов. В отличие от глубоких нейронных сетей, которые имеют несколько скрытых слоев, мелкие нейронные сети имеют только один или два скрытых слоя между входом и выходом. Это делает обработку быстрой и прозрачной. Неглубокая нейронная сеть Word2Vec может быстро распознавать семантические сходства и идентифицировать слова-синонимы, что делает ее быстрее глубоких нейронных сетей.

### 2.3.3 GloVe

GloVe (Global Vector) — алгоритм обучения без учителя для получения векторных представлений слов. Обучение проводится на основе агрегированной глобальной статистики частоты совпадения слов из корпуса, и полученные представления демонстрируют интересные линейные подструктуры векторного пространства слов. Преимущества GloVe:

— Простая архитектура без нейронной сети.

— Модель быстрая, и этого может быть достаточно для простых приложений.

— GloVe улучшает Word2Vec. Она добавляет частоту встречаемости слов и опережает Word2Vec в большинстве приложений.

— Осмысленные эмбеддинги.

Недостатки алгоритма:

— Хотя матрица совместной встречаемости предоставляет глобальную информацию, GloVe остаётся обученной на уровне слов и даёт немного данных о предложении и контексте, в котором слово используется.
— Плохо обрабатывает неизвестные и редкие слова.

### 2.3.4 FastText

FastText —– это созданная в Facebook библиотека, содержащая предобученные готовые векторные представления слов и классификатор, то есть алгоритм машинного обучения разбивающий тексты на классы.

К основной модели Word2Vec добавлена модель символьных n-грамм. Каждое слово представляется композицией нескольких последовательностей символов определённой длины. Например, слово they в зависимости от гиперпараметров, может состоять из “th”, “he”, “ey”, “the”, “hey”. По сути, вектор слова – это сумма всех его n-грамм.

Результаты работы классификатора хорошо подходят для слов с небольшой частотой встречаемости, так как они разделяются на n-граммы. В отличие от Word2Vec и Glove, модель способна генерировать эмбеддинги для неизвестных слов.

# 3 Существующие методы классификации текста

## 3.1 KNN

Метод k-ближайших соседей (k-nearest neighbors) — это простой алгоритм машинного обучения с учителем, который можно использовать для решения задач классификации и регрессии.

Алгоритм K-NN сохраняет все доступные данные и классифицирует новую точку данных на основе сходства. Это означает, что когда появляются новые данные, их можно легко классифицировать по категории наборов с помощью алгоритма K-NN.

Согласно принципу алгоритма KNN, структура классификатора включает в себя 4 параметра: данные для классификации, набор выборочных данных, набор выборочных меток и значение K. Затем вычислить расстояние между новыми данными и выборочными данными, упорядочить расстояния от наименьшего к наибольшему, возьмите первые K ближайших данных. Наиболее часто встречающаяся метка может быть идентифицирована как новая метка данных путем определения количества вхождений каждого введенного типа данных в K первых точках.

Преимущества метода.

- Алгоритм прост и легко реализуем.
- Нет необходимости строить модель, настраивать несколько параметров или делать дополнительные допущения.
- Алгоритм универсален. Его можно использовать для обоих типов задач: классификации и регрессии.

Недостатки метода.

- Алгоритм работает значительно медленнее при увеличении объема выборки, предикторов или независимых переменных.
- Из аргумента выше следуют большие вычислительные затраты во время выполнения.

– Всегда нужно определять оптимальное значение k.

### 3.2 Метод опорных векторов

Метод опорных векторов (англ. support vector machine, SVM) — один из наиболее популярных методов обучения, который применяется для решения задач классификации и регрессии. Основная идея метода заключается в построении гиперплоскости, разделяющей объекты выборки оптимальным способом.

– Преимущества метода:
  + хорошо работает с пространством признаков большего размера;
  + хорошо работает с данными небольшего объема;
  + метод находит разделяющую полосу максимальной ширины, позволяет в дальнейшем осуществлять более уверенную классификацию.
– Недостатки метода:
  + долгое время обучения (для больших наборов данных);
  + неустойчивость к шуму: выбросы в исходных данных становятся опорными объектами-нарушителями и напрямую влияют на построение разделяющей гиперплоскости.

### 3.3 Decision Tree and Random Forest

Деревья решений (Decision Tree)[2] являются одними из самых ранних и популярных классификаторов. Структура этого метода представляет собой иерархическую декомпозицию пространства данных[3], [4]. Основная идея заключается в создании дерева на основе атрибута для категоризированных точек данных, но основная задача дерева решений заключается в том, какой атрибут или функция может находиться на родительском уровне, а какой должен быть на дочернем уровне.

Дерево решений —– это очень быстрый алгоритм как для обучения, так и для прогнозирования, но он также чрезвычайно чувствителен к небольшим изменениям в данных[5]. Эта модель также имеет проблемы с прогнозированием вне выборки[6].

Метод случайных лесов (Random Forest) —– это метод обучения для клас-

сификации текста. Случайные леса представляют собой наборы деревьев решений, обученных с использованием случайных подмножеств признаков, которые достигли гораздо более высокой производительности и чаще используются на практике. Этот метод очень быстро обучается работе с наборами текстовых данных по сравнению с другими методами, такими как глубокое обучение, но довольно медленным для создания прогнозов после обучения[7]. Таким образом, чтобы добиться более быстрой структуры, количество деревьев в лесу необходимо уменьшить, поскольку большее количество деревьев в лесу увеличивает временную сложность на этапе прогнозирования.

### 3.4 CNN

Сверточная нейронная сеть (CNN) —– это архитектура глубокого обучения, которая обычно используется для иерархической классификации документов [8, 6]. Хотя CNN изначально были созданы для обработки изображений, они также эффективно использовались для классификации текста[9, 10].

В базовой CNN для обработки изображений тензор изображения свернут с набором ядер размера d × d. Эти слои свертки называются картами объектов и могут объединяться для предоставления нескольких входных фильтров. Чтобы снизить сложность вычислений, CNN используют пуллинг для уменьшения размера выходных данных от одного уровня сети к другому. Различные методы объединения используются для уменьшения выходных данных при сохранении важных функций[11]. Чтобы передать объединенные выходные данные составных избранных карт на следующий слой, карты сводятся в один столбец. Последние слои CNN обычно полностью связаны. В общем, на этапе обратного распространения ошибки сверточной нейронной сети корректируются как веса, так и фильтры детектора признаков. Потенциальная проблема, которая возникает при использовании CNN для классификации текста, это количество «каналов» S (размер пространства признаков). Хотя приложения классификации изображений обычно имеют мало каналов (например, только 3 канала RGB), S может быть очень большим (например, 50000) для приложений классификации

текста [189], что приводит к очень высокой размерности.

Было предложено множество подходов, одним из самых популярных является TextCNN[12], сравнительно простая модель на основе CNN с однослойной структурой свертки, которая размещается поверх вложений слов.

# 4 Сравнение существующих методов классификации текста

Таким образом, рассматривая решения задачи классификации текста, можно выделить два подхода к решению: с помощью машинного обучения и глубокого обучения.

Машинное обучение включает модели: KNN, метод опорных векторов (SVM), деревья решений и случайные леса. KNN —– это метод классификации, который легко реализовать и который адаптируется к любому типу пространства признаков. Эта модель также естественным образом обрабатывает случаи с несколькими классами[13, 14]. Однако KNN ограничен ограничениями по хранению данных для больших задач поиска ближайших соседей. Кроме того, производительность KNN зависит от поиска значимой функции расстояния, что делает этот метод сильно зависимым от данных алгоритмом[15, 16]. SVM —– мощный алгоритм классификации текста, но он требует соответствующей предварительной обработки данных и может не подходить для нелинейных или сложно структурированных данных. SVM способен хорошо работать в пространствах объектов большой размерности. При работе с текстовыми данными пространство признаков часто бывает большим, поскольку количество слов или фраз может быть очень большим. SVM хорошо справляется с этим случаем и дает точные результаты классификации. Однако SVM требует предварительной обработки данных для извлечения признаков. Дерево решений — это очень быстрый алгоритм как для обучения, так и для прогнозирования, но он также чрезвычайно чувствителен к небольшим изменениям в данных[5]. Случайные леса (т. е. набор деревьев решений) очень быстро обучаются по сравнению с другими методами, но довольно медленно создают прогнозы после обучения. Таким образом, чтобы добиться более быстрой структуры, количество деревьев в лесу необходимо уменьшить, поскольку большее количество деревьев в лесу увеличивает временную сложность на этапе прогнозирования.

Глубокое обучение — один из самых мощных методов искусственного

интеллекта (ИИ), и многие исследователи и ученые сосредоточены на архитектурах глубокого обучения, чтобы повысить надежность и вычислительную мощность этого инструмента. Однако архитектуры глубокого обучения также имеют некоторые недостатки и ограничения при применении к задачам классификации текста. Одна из основных проблем этой модели заключается в том, что глубокое обучение не способствует всестороннему теоретическому пониманию процесса обучения[17]. Хорошо известным недостатком методов глубокого обучения является их природа «черного ящика»[18, 19]. То есть метод, с помощью которого методы глубокого обучения получают свернутый результат, не совсем понятен. Еще одним ограничением глубокого обучения является то, что для него обычно требуется гораздо больше данных, чем для традиционных алгоритмов машинного обучения, а это означает, что этот метод нельзя применять для задач классификации небольших наборов данных [20, 21]. Кроме того, огромный объем данных, необходимых для алгоритмов классификации глубокого обучения, еще больше усугубляет вычислительную сложность на этапе обучения[22].

Исходя из выше сказанно, можно сделать сравнения.

— По сложности: Машинное обучение и глубокое обучение имеют разные сложности в процессе обучения и развертывания модели. В машинном обучении часто используются традиционные алгоритмы, такие как KNN, машины опорных векторов (SVM) или случайные леса. Для глубокого обучения модель нейронной сети будет иметь множество скрытых слоев, что создает более сложную сеть и требует больше вычислительных ресурсов.

— По обобщению: Глубокое обучение часто имеет лучшую способность к обобщению, чем машинное обучение. Это означает, что глубокое обучение способно изучать сложные функции и автоматически извлекать информацию из данных. Машинное обучение также может дать хорошие результаты, но оно во многом зависит от ручного выбора функций и из-

влечения их из данных.

— По требуемым данным: для достижения хороших результатов глубокое обучение часто требует большого объема обучающих данных. Благодаря машинному обучению меньшее количество обучающих данных может дать лучшие результаты. Однако, если данных достаточно, глубокое обучение может изучить более сложные закономерности и дать более точные результаты.

— По времени обучения: обучение глубокому обучению часто занимает больше времени, чем машинное обучение. При использовании глубокой нейронной сети процесс обучения может длиться от нескольких часов до нескольких дней и даже недель. Между тем, машинное обучение позволяет быстро обучаться и давать хорошие результаты при меньшем объеме данных.

**Вывод**

На основе сравнения можно сделать вывод, что методы глубокого обучения требуют больше данных и времени на обучение, чем методы машинного обучения. Методы машинного обучения также имеют лучшее обобщение, чем методы глубокого обучения.

# ЗАКЛЮЧЕНИЕ

В результате выполнения работы были проведены анализ предметной области и обзор существующих методов решения задачи классификации текста. Также было проведено сравнение преимуществ и недостаток рассмотренных методов.

Был сделан вывод, что методы глубокого обучения требуют больше данных и времени на обучение, чем методы машинного обучения. Методы машинного обучения также имеют лучшее обобщение, чем методы глубокого обучения.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ


1. A Survey on Text Classification: From Traditional to Deep Learning / Qian Li, Hao Peng, Jianxin Li [и др.] // ACM Transactions on Intelligent Systems and Technology (TIST). 2020. Т. 13. С. 1–41.
2. Whitney Donna L., Evans Bernard W. Abbreviations for names of rock-forming minerals // American Mineralogist. 2010. Т. 95. С. 185–187.
3. Aggarwal C.C. Zhai C. Mining Text Data. Springer New York, NY, 2012.
4. Morgan James N., Sonquist John A. Problems in the Analysis of Survey Data, and a Proposal // Journal of the American Statistical Association. 1963. Т. 58. С. 415–434.
5. Towards an aggregator that exploits big data to bid on frequency containment reserve market / Christian Giovanelli, Xin Liu, S. Sierla [и др.] // IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society. 2017. С. 7514–7519.
6. Jasim Dalia Sami. Data mining approach and its application to dresses sales recommendation // Researchgate. Net. 2016.
7. Social network analytics for contemporary business organizations / Himani Bansal, Gulshan Shrivastava, Gia Nhu Nguyen [и др.]. IGI Global, 2018.
8. Recurrent Convolutional Neural Networks for Text Classification / Siwei Lai, Liheng Xu, Kang Liu [и др.] // AAAI Conference on Artificial Intelligence. 2015.
9. LeCun Yann, Bengio Yoshua, Hinton Geoffrey E. Deep Learning // Nature. 2015. Т. 521. С. 436–444.

10. Gradient-based learning applied to document recognition / Yann LeCun, Léon Bottou, Yoshua Bengio [и др.] // Proc. IEEE. 1998. Т. 86. С. 2278–2324.

11. Scherer Dominik, Müller Andreas C., Behnke Sven. Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition // International Conference on Artificial Neural Networks. 2010.

12. Kim Yoon. Convolutional Neural Networks for Sentence Classification // Conference on Empirical Methods in Natural Language Processing. 2014.

13. Divya Sahgal A. Ramesh. On Road Vehicle Detection Using Gabor Wavelet Features with Various Classification Techniques. // Proceedings of the 14th International Conference on Digital Signal Processing Proceedings. 2002.

14. Patel Divyesh, Srivastava Tanuja. Ant Colony Optimization Model for Discrete Tomography Problems // International Conference on Soft Computing for Problem Solving. 2013.

15. Sahgal Divya, Parida Manoranjan. Object Recognition Using Gabor Wavelet Features with Various Classification Techniques // International Conference on Soft Computing for Problem Solving. 2013.

16. Comparing Existing Methods for Predicting the Detection of Possibilities of Blood Cancer by Analyzing Health Data / Gandhi Priyank Sanjay, Viral Nagori, GP Sanjay [и др.] // Int. J. Innov. Res. Sci. Technol. 2018. Т. 4. С. 10–14.

17. Shwartz-Ziv Ravid, Tishby Naftali. Opening the Black Box of Deep Neural Networks via Information // ArXiv. 2017. Т. abs/1703.00810.

18. MacDonell Stephen G., Gray A. R. Alternatives to regression models for estimating software projects. 1996.

19. Shrikumar Avanti, Greenside Peyton, Kundaje Anshul. Learning Important

Features Through Propagating Activation Differences // International Conference on Machine Learning. 2017.

20. Anthes Gary. Deep learning comes of age // Commun. ACM. 2013. T. 56. C. 13–15.

21. Lampinen Andrew Kyle, McClelland James L. One-shot and few-shot learning of word embeddings // ArXiv. 2017. T. abs/1710.10280.

22. Severyn Aliaksei, Moschitti Alessandro. Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks // Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2015.

# ПРИЛОЖЕНИЕ А