

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
”ВЫСШАЯ ШКОЛА ЭКОНОМИКИ”»

**Московский институт электроники и математики**

Юткин Дмитрий Игоревич, группа БИВ-144  
Вдовкин Василий Алексеевич, группа БИВ-144

**Классификация и агрегация новостных статей, используя методы и модели  
обработки естественного языка**

Междисциплинарная курсовая работа  
по направлению 09.03.01 Информатика и вычислительная техника  
студентов образовательной программы бакалавриата  
«Информатика и вычислительная техника»

Студент \_\_\_\_\_ Д.И. Юткин  
Студент \_\_\_\_\_ В.А. Вдовкин

Руководитель  
старший преподаватель  
Л.Л. Волкова  
\_\_\_\_\_

Москва 2017 г.

## **Аннотация**

В данной работе изучаются алгоритмы классификации и агрегации текста, основанные на методах и моделях обработки языка, и применяются на русскоязычных новостных статьях. В процессе работы удалось собрать и обработать большой корпус новостей, на котором обучены следующие модели: TFIDF, SVM, FastText, word2vec, Kmeans. Классификаторы показали точность 86-88%. Для визуализации работы перечисленных алгоритмов на практике реализован новостной агрегатор в виде web-сервиса, агрегирующий и классифицирующий актуальные новости с сайтов российских СМИ.

## **Abstract**

The main purposes of this work are the natural language processing models and algorithms of the text aggregation and classification and application of these models and algorithms on the russian media news articles. In the course of this work a dataset with the significant number of news was gathered and processed. On this dataset TFIDF, SVM, FastText, word2vec, Kmeans models were trained. The accuracy of the classifiers is between 86 and 88 depending on a model. To show how these algorithms are working on practice the news aggregation web-service was implemented, which aggregates and classifies articles in real-time.

## Оглавление

<b>1</b>	<b>Введение</b>	<b>4</b>
<b>2</b>	<b>Цель и задачи курсовой работы</b>	<b>5</b>
<b>3</b>	<b>Сбор и подготовка данных</b>	<b>5</b>
3.1	Получение данных из новостных источников	5
3.2	Нормализация новостных статей	6
<b>4</b>	<b>Классификация новостных статей</b>	<b>8</b>
4.1	Линейная модель на TF-IDF признаках	8
4.2	Градиентный бустинг деревьев на word2vec	8
<b>5</b>	<b>Агрегация новостных статей</b>	<b>8</b>
5.1	Кластеризация с помощью алгоритмов машинного обучения	8
5.2	Объединение в связные компоненты графа	8
<b>6</b>	<b>Разработка веб-приложения</b>	<b>8</b>
<b>7</b>	<b>Заключение</b>	<b>8</b>
	<b>Список литературы</b>	<b>9</b>
	<b>Приложение</b>	<b>10</b>

# 1 Введение

С каждым годом вычислительные мощности современных компьютеров и сервисов, предлагающие облачные вычисления, позволяют обрабатывать всё большие массивы данных. Благодаря этому происходит быстрое развитие алгоритмов анализа данных и машинного обучения. Результат работы этих алгоритмов можно увидеть в нашей повседневной жизни: сервисы прогноза погоды, которые предсказывают направление движения облаков, персонализированная реклама, подстраивающиеся под интересы пользователя, автомобили с автопилотом и т.д. — в основе всех этих разработок лежат алгоритмы интеллектуального анализа данных и машинного обучения.

Одна из важнейших проблем, возникающая перед исследователями и разработчиками подобных систем, заключается в поиске данных для обучения моделей, которые в будущем будут использоваться для анализа новой информации.

Решением данной проблемы являются тексты из сети Интернет. Петабайты информации, записанной на естественном языке за последние несколько десятилетий, доступны любому исследователю. Книги, новостные статьи, блоги, посты в социальных сетях — всё это является источником легкодоступных данных.

Именно поэтому обработка естественного языка (Natural Language Processing, NLP) получила большое развитие. За последние несколько лет появилось множество специализированных библиотек и сервисов для анализа естественного языка. К таким сервисам, например, относится IBM Watson — набор платных продуктов, предлагающий различные инструменты работы с текстом: от извлечения важной информации и классификации, до его генерации. Яркими примерами свободнораспространяемых библиотек являются: NLTK, Gensim, MyStem, rumorphy и многие другие.

В данной работе решаются две задачи:

- а) классификация тем новостных статей;
- б) агрегация новостных статей по смысловой близости.

Для придания работе новизны и оригинальности задача решается для русскоязычных новостей.

Практическая значимость работы доказывается на примере реализации в виде web-сервиса новостного агрегатора, который в реальном времени, с помощью

алгоритмов и подходов предложенных в данной работе, обрабатывает публикации русскоязычных интернет-СМИ.

## **2 Цель и задачи курсовой работы**

Целью курсовой работы является разработка веб-сервиса, который:

- а) в реальном времени получает статьи из русскоязычных новостных источников;
- б) классифицирует полученные статьи по общим темам;
- в) агрегирует по схожести содержания статьи из различных источников.

Агрегация и классификация основана на исследуемых в работе алгоритмы.

Для достижения поставленной цели, должны быть выполнены следующие задачи:

- а) Изучить методы и модели автоматической обработки текста и естественного языка;
- б) Собрать корпус новостных статей для обучения моделей;
- в) Исследовать и реализовать различные подходы к классификации и агрегации текстовых документов;
- г) Разработать back-end и front-end инфраструктуру сервиса.

## **3 Сбор и подготовка данных**

### **3.1 Получение данных из новостных источников**

Для получения робастных моделей машинного обучения, требуется достаточно большой корпус новостных статей, содержащий нескольких сотен тысяч документов. Обычно существуют готовые коллекции документов, но из-за выбранных ограничений к документам, а именно: новости на литературном русском языке, содержащие тег – слово, описывающие тему документа, найти такой корпус практически невозможно, поэтому необходимо собрать данные самостоятельно.

В качестве новостных источников были выбраны следующие популярные СМИ: «Газета.Ru», «Lenta.ru», «ТАСС», «Новая Газета», «ВЕДОМОСТИ» и «СПОРТ-ЭКСПРЕСС». Последнее было выбрано по причине малого количества спортивных новостей от других источников.

При анализе сайтов СМИ стало понятно, что они имеют схожую структуру: для отображения ссылок на статьи, используются страницы со списком новостей

(«Лента новостей»), по которым можно итерироваться, изменяя параметры запросов, например, дату последней новости на странице или количество показанных статей. Для извлечения данных из источников реализован набор парсеров, которые опираются на описанную структуру.

Стоит отметить, что во многих случаях для получения чистого текста приходится ждать ответа от сервера и обрабатывать HTML содержание страниц, и это сильно замедляет работу, поэтому парсеры одновременно обрабатывают множество ссылок на статьи в параллельных процессах, что значительно ускоряет работу. Впоследствии эти парсеры удалось также внедрить в web-сервер для получения недавних новостей.

При формировании корпуса, все новостные статьи сопровождалось различными метаданными: название СМИ, ссылка на статью, дата публикации, тег и заголовок. В результате было получено более 1,1 млн. новостей от 1999 до 2017 года, многие из которых имели неправильно проставленные темы или не имели тем вовсе. Причин тому может быть несколько, например, ошибки редакторов или технические ограничения веб-сайтов новостных агентств. Например, большинство новостей на сайте «Новой Газеты» помечены тегом «политика», хотя речь в них идет часто совсем о другом. После детального анализа тегов стало ясно, что относить новости к рубрикам редакторы стали только после определённого времени, а все уже имеющие статьи на сайте отнесли в «политику». Некорректные данные пришлось удалить.

В итоговую выборку, которая в дальнейшем использовалась для обучения и тестирования алгоритмов, вошло 133 тыс. статей, помеченных 32 различными тегами. Распределение СМИ и тем на отобранных данных отражено на рисунке 1.

### **3.2 Нормализация новостных статей**

Нормализация является одной из важнейших стадий обработки естественного языка. Вкратце, нормализация приводит текст в более информативный для моделей вид. В зависимости от языка процесс нормализации может отличаться. Для русского языка, например, достаточно удалить пунктуацию, стоп-слова и лемматизировать. Стоп-слова – это слова, которые примерно одинаково распределены по всему корпусу языка, чаще всего ими являются местоимения, предлоги и союзы. Лемматизация – приведение слова в начальную форму. Часто вместо лемматизации

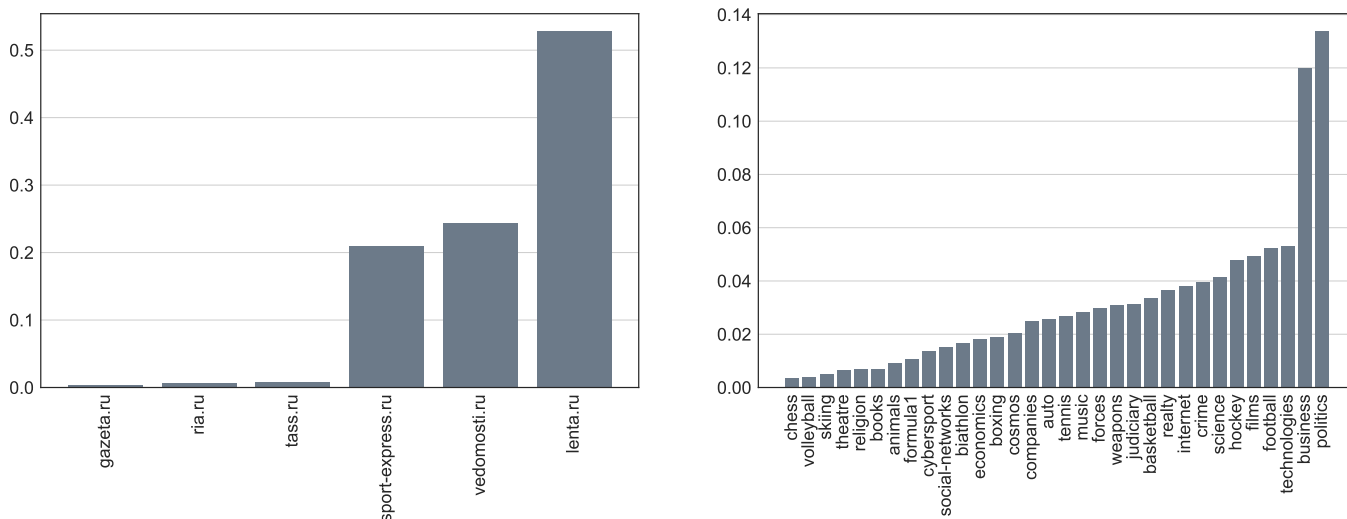


Рисунок 1 — Распределение СМИ и тем в данных

используется стемминг – алгоритм, который убирает части слова, влияющие на его форму, например, окончание. Данные алгоритмы не только опираются на словари, но и на определённые правила, зависящие от языка, так как в корпусе могут встречаться слова в разных формах, которых нет в словаре, например, неологизмы, но образованы они по правилам языка.

Процесс обработки текста в данной работе состоит из нескольких последовательных этапов:

- а) Приведение текста в нижний регистр.
- б) Удаление чисел и символов пунктуации. Дефис сохраняется.
- в) Удаление стоп-слов. В данный набор входят наиболее часто употребляемые слова русского и английского языков, а также названия новостных агентств («лента», «тасс», «риа» и т.д.), сохранение которых приводит к переобучению моделей.
- г) Лемматизация каждого слова с помощью библиотеки MyStem<sup>1</sup>.

<sup>1</sup><https://tech.yandex.ru/mystem/>

## **4 Классификация новостных статей**

### **4.1 Линейная модель на TF-IDF признаках**

### **4.2 Градиентный бустинг деревьев на word2vec**

## **5 Агрегация новостных статей**

### **5.1 Кластеризация с помощью алгоритмов машинного обучения**

### **5.2 Объединение в связные компоненты графа**

## **6 Разработка веб-приложения**

Веб-приложение состоит из двух основных частей: back-end и front-end. Back-end – это сам веб-сервер, который осуществляет обработку запросов пользователей, получение и обработку данных. Front-end – это пользовательский интерфейс, визуализирующий полученные данные от back-end в понятный вид. С помощью этого интерфейса пользователь способен не только получать, но и передавать данные на back-end.

## **7 Заключение**



## **Список литературы**

## **Приложение**