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 преподователь
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

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 (Natrual 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, pymorphy 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 несколько сотен тысяч документов. Обычно существуют готовые коллекции текстов, но из-за выбранных ограничений к документам, а именно: новости на русском языке вместе с тег — словом, описывающим тему документа, найти такой корпус не удалось, поэтому было принято решение собрать данные самостоятельно.

В качестве новостных источников были выбраны следующие популярные СМИ: «Газета.Ru», «Lenta.ru», «ТАСС», «Новая Газета», «ВЕДОМОСТИ», «РИА Новости» 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 алгоритмов, 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, что сильно замедляет работу, поэтому алгоритмы получения новостей одновременно обрабатывают в параллельных процессах множество веб-страниц, что значительно ускоряет работу. Данные алгоритмы также используются в основном web-сервисе для получения недавних новостей.

При формировании корпуса, все новостные статьи сопровождались различными метаданными: название СМИ, ссылка на статью, дата публикации, тег и заголовок. В результате было получено более 1,1 млн. новостей с 1999 по 2017 год, 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», что зачастую не совпадает с истинным содержанием статьи. 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 133 529 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. Не формально, нормализация приводит текст в более информативный для моделей вид. В зависимости от языка процесс нормализации может отличаться. Например, для русского языка зачастую удаляют пунктуацию и стоп-слова, а также производят лемматизацию. Стоп-слова — это слова, которые примерно одинаково распределены по всему корпусу языка, чаще всего ими являются место-

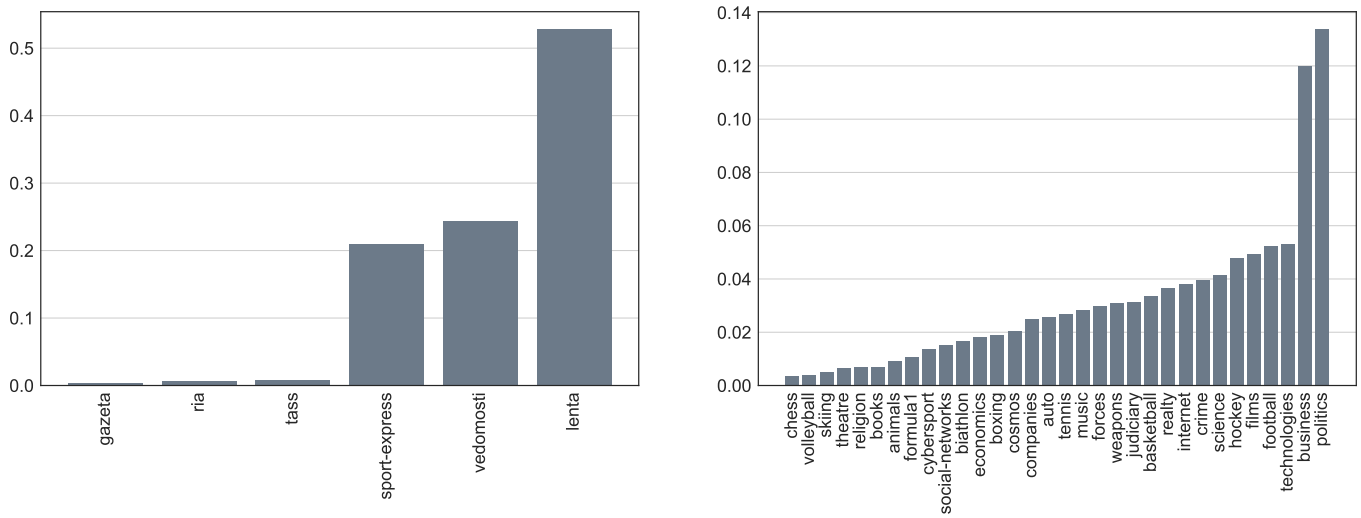

Рисунок 1 — Распределение СМИ и тем в данных

имения, предлоги и союзы. Лемматизация — приведение слова в начальную форму (лемма):

- для существительных — именительный падеж, единственное число;
- для прилагательных — именительный падеж, единственное число, мужской род;
- для глаголов, причастий, деепричастий — глагол в инфинитиве.

Часто вместо лемматизации используется стемминг — алгоритм, который убирает части слова, влияющие на его форму, например, окончание. В результате применения данной процедуры однокоренные слова, как правило, преобразуются к одинаковому виду. 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[1].

4 Классификация новостных статей

За последние два десятка лет, в результате активных исследований в области машинного обучения, было изобретено множество успешных алгоритмов классификации. Например, такие модели как support vector machines (SVM) [1], градиентный бустинг деревьев и нейронные сети [2], были успешно применены к задачам классификации текстов. В данной работе для классификации новостных статей использовались две из перечисленных выше модели — это SVM и градиентный бустинг деревьев.

Анализ текста является важной частью машинного обучения, однако сырые данные, а именно последовательности символов переменной длины, не могут быть переданы на вход алгоритму в явном виде т.к. большинство моделей ожидают численный вектор признаков фиксированной длинны.

Векторизация — метод трансформации коллекции текстовых документов в числовые вектора признаков. Существуют различные подходы к векторизации текста, в данной работе были применены два самых популярных: TF-IDF и word2vec.

4.1 Линейная модель на TF-IDF признаках

Одним из самых простых подходов к решению задачи классификации текстовых документов является обучение линейного классификатора на TF-IDF признаках, посчитанных на корпусе документов.

TF-IDF — статистическая мера, используемая для оценки важности слова в контексте документа, являющегося частью коллекции документов или корпуса. [3] TF-IDF — это произведение двух статистик: TF (term frequency) и IDF (inverse document frequency). На сегодняшний день, TF-IDF один из самых популярных способов взвешивания слов, входящих в корпус документов. Например, 83% рекомендательных систем цифровых библиотек используют TF-IDF [4].

Существует множество способов подсчёта TF-IDF, в данной работе использовался следующий:

$$\mathrm{tf}(t, d) = \frac{n_t}{\sum_k n_k},$$

[1] https://tech.yandex.ru/mystem/

где n_t есть число вхождений слова t в документ, а $\sum_k n_k$ — общее число слов в данном документе.

$$\text{idf}(t, D) = \log \frac{|D|}{|\{d_i \in D \mid t \in d_i\}|},$$

где $|D|$ — число документов в корпусе, $|\{d_i \in D \mid t \in d_i\}|$ — число документов из коллекции D, в которых встречается t (когда $n_t \neq 0$).

Таким образом, мера TF-IDF является произведением двух сомножителей:

$$\text{TF-IDF}(t, d, D) = \text{tf}(t, d) \cdot \text{idf}(t, D).$$

Признаковым описанием одного объекта $d \in D$ будет вектор

$$\big(\text{TF-IDF}(t,d,D)\big)_{t \in V},$$

где V — словарь всех слов, встречающихся в коллекции D.

Для преобразования новостных статей в числовые признаки, использовался класс `TfidfVectorizer` из библиотеки машинного обучения Scikit-learn [5]. В качестве параметров векторизации использовались следующие значения: `min_df=3` — учитываются слова, встретившиеся суммарно во всех документах минимум 3 раза и `ngram_range=(1,2)` — учитываются как отдельные слова, так и би-граммы.

После векторизации новостных статей, была получена разрежённая матрица размера $133\,529$ строк на $1\,025\,919$ столбцов. По причине большого количества признаков ($\gg$ обучающих примеров), в качестве классификатора было решено использовать линейный SVM. Не смотря на то, что данный алгоритм был впервые описан более пятидесяти лет назад, сегодня он по прежнему показывает одни из самых высоких результатов в задачах классификации текста. Как было показано в [6], особенно высокое качество удаётся получить при использовании би-грамм.

В данной работе в качестве SVM использовался `SGDClassifier` из библиотеки Scikit-learn. Данный класс реализует различные линейные классификаторы, параметры которых оптимизируются с помощью алгоритма стохастического градиентного спуска [7].

Классификатор обучался со следующими параметрами: `loss="hinge"` — функционал качества линейного SVM, `n_iter=70` — число итераций оптимизационного алгоритма, `alpha=1e-5` — коэффициент регуляризации.

Обучение классификатора происходило на 70% новостных статей, остальные 30% использовались для валидации. Оценка качества классификации проводилась с помощью метрик accuracy и F1-меры с макро-усреднением по каждому классу. На валидационном множестве линейному SVM удалось получить accuracy $= 0.8792$ и F1 $= 0.8860$. Матрица потерь приведена на рисунке 2.

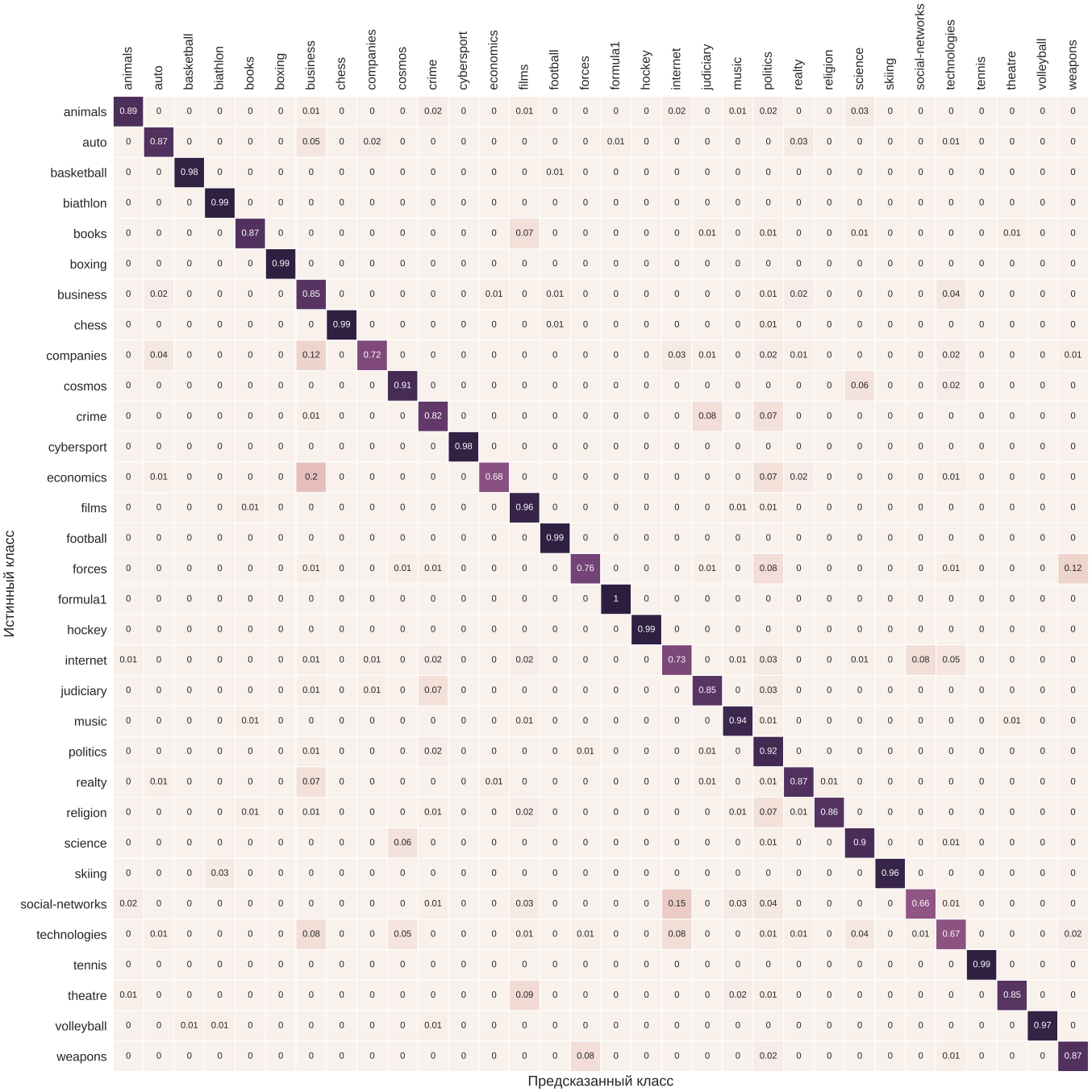

Рисунок 2 — Матрица потерь линейного SVM

Распределение accuracy для каждого из классов отражено на рисунке 3.

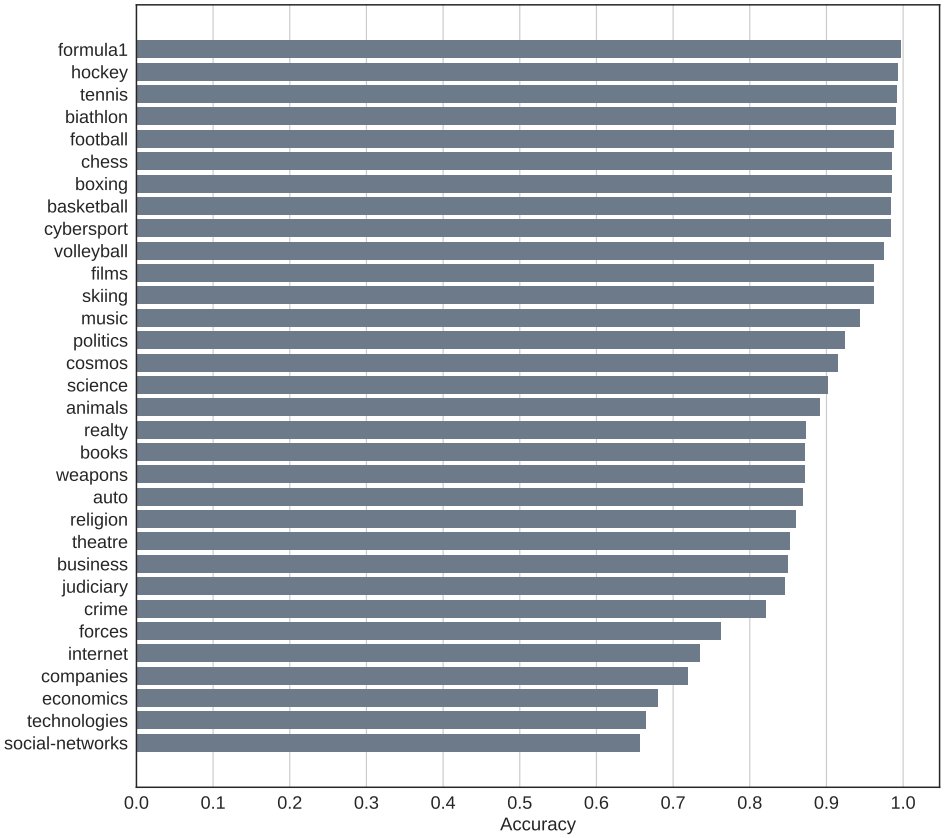

Рисунок 3 — Распределение accuracy линейного SVM для каждого из классов

Веса признаков в линейной модели в случае, если признаки отмасштабированы, характеризуют степень их влияния на значение целевой переменной. В задаче классификации текстов, кроме того, признаки являются хорошо интерпретируемыми, поскольку каждый из них соответствует конкретному слову, поэтому из линейного SVM для каждого класса были извлечены топ слова, характеризующие каждую из 32-ух тем (таблица 1).

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 поль-

animals	жить	вольер	хозяин	животный	зоопарк	питомец	кличка	животное
auto	авторынок	осаго	автомобильный	автопроизводитель	автопром	камаз	автоваз	автомобиль
basketball	рфб	евробаскет	центровой	кубок европа	баскетбол	баскетболист	евролига	нба
biathlon	эстафета	биатлонистка	шипулин	сбр	хохфильцен	биатлон	ibu	биатлонист
books	библиотека	произведение	писательница	роман	книга	литературный	поэт	писатель
boxing	алоян	лебзяк	мма	поединок	поветкин	бой	бокс	боксер
business	россельхознадзор	fifa	ржд	газпром	туроператор	formula	ритейлер	оао
chess	карякин	шахматист	карякина	магнус	шахматы	карлсен	фид	шахматный
companies	тысяча автомобиль	компания	миллиард кубометр	миллиард	тысяча	процент акция	ретейлер	процент
cosmos	космонавт	светить	прогресс	вселенная	космос	астрофизик	марс	астронавт
crime	грабитель	изымать	группировка	полиция	убивать	летний	преступник	тюрьма
cybersport	gaming	team	valve	киберфутбол	dota	киберспорт	киберспортивный	киберспортсмен
economics	мрот	греция	бюджет	пенсия	ввп	минфин	экономика	инфляция
films	мультфильм	сериал	актриса	кино	картина	режиссер	актер	фильм
football	поле	стадион	нападающий	матч тур	фифа	уефа	футболист	полузащитник
forces	развертывать	выполнение	военнослужащий	военный	шойгу	генштаб	конашенков	минобороны
formula1	манор	цитировать	рено	феррари	макларен	пилот	мерседес	формула
hockey	авангард	ска	шайба	нападающий	хоккей	хоккеист	нхл	кхл
internet	википедия	сервис	ресурс	youtube	сайт	хакер	блогер	интернет
judiciary	стража	статья	арестовывать	колония	следственный комитет	следствие	скр	комитет россия
music	композитор	евровидение	песня	певец	концерт	альбом	певица	музыкант
politics	лидер	кремль	парламентарий	партия	депутат	госдума	глава	мид
realty	объект	строительный	жилищный	строительство	жкх	ипотека	жилье	недвижимость
religion	монастырь	собор	церковный	муфтий	святой	христиан	митрополит	патриарх
science	университет	журнал	математик	физик	научный	археолог	исследователь	ученый
skiing	вяльбе	нортуг	лыжник	fis	легков	йохауг	лахти	устюгов
social-networks	некоторые	юзер	facebook	twitter	пользователь сеть	пользователь	вконтакте	соцсеть
technologies	vimpelcom	apple	оператор	wifi	говорить	мтс	робот	контакт
tennis	кубок федерация	open	шарапов	теннис	корт	теннисист	теннисистка	кубок дэвис
theatre	росгосцирк	цирковой	балет	постановка	театральный	мюзикл	театр	спектакль
volleyball	факел	маричев	алекно	суперлига	казанский	белогорье	волейболист	волейбол
weapons	jane	использоваться	defense news	министерство оборона	миллиметр	миллиметровый	defense	тип

Таблица 1: Топ слова для каждой темы

зователей, получение и обработку данных. Front-end – это пользовательский интерфейс, визуализирующий полученные данные от back-end в понятный вид. С помощью этого интерфейса пользователь способен ни только получать, но и передавать данные на back-end.

7 Заключение

Список литературы

1. *Weston Jason, Watkins Chris*. Support Vector Machines for Multi-class Pattern Recognition // Proc. European Symposium on Artificial Neural Networks. — 1999. — Pp. 219–224.

2. Very Deep Convolutional Networks for Natural Language Processing / Alexis Conneau, Holger Schwenk, Loïc Barrault, Yann LeCun // *CoRR*. — 2016. — Vol. abs/1606.01781. http://arxiv.org/abs/1606.01781.

3. *Jones Karen Sparck*. A statistical interpretation of term specificity and its application in retrieval // *Journal of Documentation*. — 1972. — Vol. 28, no. 1. — Pp. 11–21. http://dx.doi.org/10.1108/eb026526.

4. Research-paper recommender systems: a literature survey / Joeran Beel, Bela Gipp, Stefan Langer, Corinna Breitinger // *International Journal on Digital Libraries*. — 2016. — Vol. 17, no. 4. — Pp. 305–338.

5. Scikit-learn: Machine Learning in Python / F. Pedregosa, G. Varoquaux, A. Gramfort et al. // *Journal of Machine Learning Research*. — 2011. — Vol. 12. — Pp. 2825–2830.

6. *Wang Sida I., Manning Christopher D.* Baselines and Bigrams: Simple, Good Sentiment and Topic Classification // Proceedings of the ACL. — 2012. — Pp. 90–94.

7. *Bottou Léon*. Large-Scale Machine Learning with Stochastic Gradient Descent // Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers / Ed. by Yves Lechevallier, Gilbert Saporta. — Heidelberg: Physica-Verlag HD, 2010. — Pp. 177–186. http://dx.doi.org/10.1007/978-3-7908-2604-3_16.

Приложение