

С. И. СМЕТАНИН

**Оценка эффективности анализа
тональности текстов на русском
языке с использованием Google
Cloud Prediction API**

УДК 004.912

Национальный
исследовательский
университет «Высшая
школа экономики»,
г. Москва

В статье оценивается эффективность использования API облачного сервиса Google Cloud Prediction для решения задачи анализа тональности русскоязычных текстов. Проведен эксперимент по обучению модели на различных наборах данных, который направлен на исследование эффективности использования Prediction API и сравнение с существующими базовыми алгоритмами. Результаты эксперимента продемонстрировали высокие показатели качества работы Prediction API.

1. Введение

Социальные сети, как распространенный и открытый источник данных для анализа, особенно интересны тем, что имеют незначительную задержку между внешним событием и реакцией на него. Другими словами, сообщение о каком-либо значимом событии из реальной жизни практически моментально публикуется в социальных сетях. Одной из наиболее актуальных задач в области анализа текстовых сообщений в сайтах с элементами социальных сетей является задача распознавания эмоциональной окраски текста, которая позволяет извлечь из текстовой информации полярность мнения человека об объекте высказывания. Анализ тональности широко используется не только в бизнес секторе, но и в социальных и политических исследованиях. К примеру, для мониторинга настроения пользователей сети Интернет на основе

сообщений из социальных сетей [1]; для прогнозирования показателя спроса на товар либо услугу на основе реакции пользователей сети Интернет на рекламную кампанию по выходу на рынок [2]; для принятия решений в сфере трейдинга и аналитики [3] на основе мнений пользователей в социальных сетях; для оценки эффективности рекламных компаний [4].

Наиболее распространенными классификаторами в области анализа эмоциональной окраски текстов являются наивный Байесовский классификатор с мультиномиальным распределением, который основывается на теореме Байеса со строгим предположением о независимости. Согласно проведенному в работе [5] эксперименту по сравнению точности классификации при использовании униграмм, биграмм и триграмм, биграммы показали наиболее высокие показатели качества работы. Авторы обосновывают это тем, что биграммы находят оптимальный баланс между лексическим охватом словаря и способностью выявлять шаблоны проявления эмоционально окрашенных высказываний.

Облачные провайдеры, такие как Google, Amazon, Microsoft и IBM, широко применяют технологии машинного обучения в своих продуктах и сервисах. Более того, в качестве одной из своих услуг они предоставляют разработчикам платформу для облачного машинного обучения и облачные сервисы искусственного интеллекта, такие как компьютерное зрение, распознавание речи, анализ текста, интеллектуальный поиск и т.д.

Исследователи в работе [6] использовали облачный сервис Microsoft Azure Machine Learning для анализа отношения индийских пользователей Twitter к правительственной программе «Цифровая Индия», целью которой является трансформация страны в общество, основанное на цифровых технологиях, и переход к экономике знаний, использующей ИКТ в качестве движущей силы. Алгоритм классификации Two-Class Bayes Point Machine продемонстрировал наивысшие показатели эффективности в критериях точности, полноты и F-меры. Результаты анализа показали, что большинство людей дали положительные отзывы о программе «Цифровая Индия», что является индикатором успеха реализации проекта в восприятии целевой группы. Классификация эмоциональной окраски твитов с помощью Azure Machine Learning

также рассматривается в статье [7], где алгоритм Support Vector Machine был обучен на датасете Sentiment140 [8]. Получившаяся модель может классифицировать сообщения с точностью 77,8%, полнотой 81,1% и *F*-мерой 79,4%.

Использование облачных сервисов IBM Watson, Amazon Machine Learning и Google Prediction API описывается в работах [9], [10] и [11] соответственно. За счет делегирования построения и оптимизации модели компаниям провайдерам облачных услуг данные сервисы показывают высокие показатели качества и скорости работы. Однако при выборе провайдера разработчикам, как правило, не хватает информации о показателях качества работы сервисов с определенным естественным языком на определенных наборах данных.

В данной статье оценивается эффективность бинарного анализа тональности русскоязычных текстов на основе использования облачного сервиса для машинного обучения Google Cloud Prediction API [12], который предоставляется разработчиком по модели обслуживания Machine Learning as a Service (MLaaS). В первом разделе приведено описание MLaaS. Во втором разделе подробно описан функционал и механика работы Prediction API. В третьем разделе приведена информация об обучении Prediction API и об оценке качества работы. В четвертом разделе подведены итоги работы.

2. Machine Learning as a Service

Самостоятельная настройка и эксплуатация систем машинного обучения может быть дорогостоящим и наукоемким процессом ввиду сложности программного обеспечения и большого объема требуемого оборудования. Облачная инфраструктура, построенная на основе модели обслуживания Machine Learning as a Service, предоставляет пользователям ряд облачных услуг, в которые входят инструменты машинного обучения. В основе технологии облачных вычислений лежит предоставление пользователям сетевого доступа к различным ресурсам, например, платформе, данным или приложениям, что снимает часть временных и финансовых затрат с разработчиков. Ключевая особенность MLaaS заключается в том, что исследователи могут оперативно начать

работу с машинным обучением без необходимости в установке специализированного программного обеспечения или в развертывании серверов, поскольку все задачи по инфраструктуре и администрированию системы делегированы компании провайдеру.

3. Google Cloud Prediction API

Google Cloud Prediction API – облачный сервис Google, предназначенный для обучения и дальнейшего использования моделей машинного обучения, способных решать задачи регрессии и классификации. Prediction API сопоставляет новые элементы для прогнозирования с элементами из обучающей выборки и далее определяет категорию, которая в наибольшей степени соответствует новому элементу, либо оценивает значение по ближайшим совпадениям (в зависимости от типа модели). Данный сервис работает по принципу «черного ящика», то есть при обучении Prediction API самостоятельно подбирает наиболее оптимальные алгоритмы классификации и их конфигурацию в зависимости от входных данных, при этом не афишируя описание построенной модели.

На рис. 1 приведён пример обучающей выборки для модели, которая будет использоваться для прогнозирования тональности сообщения в социальной сети Twitter в зависимости от текста сообщения, количества слов в сообщении и времени публикации в минутах. На рисунке представлено пять примеров (строк), где каждая строка соответствует признаковому описанию одного сообщения. Каждый элемент имеет следующие столбцы: «Эмоциональная окраска сообщения», «Текст сообщения», «Количество слов в сообщении», «Время дня в минутах». Текстовые признаки объектов заключаются в кавычки, в то время как числовые записываются без кавычек.

Элементы обучающей выборки	Классы	Признаки	
		Текстовые	Числовые
{	"отрицательный»",	"Чувствую, что все становится только хуже..."	6 720
	"положительный",	"Вчера была потрясающая погода!"	4 1100
	"отрицательный",	"Сегодня какой-то нехороший день у меня :-("	6 800

Рис. 1. Пример формата тренировочных данных

При обработке текстовой информации Prediction API автоматически разбивает строки по пробелам на составные слова, которые далее использует для прогнозирования без учета порядка следования. Взаимодействие с сервисом осуществляется посредством RESTful API (на момент написания статьи была актуальна версия V1.6), основные методы представлены в табл. 1.

Таблица 1

Методы Google Cloud Prediction API [13]

Метод	Описание
<i>analyze</i>	Получение информации о б обученной модели и данных, на которых осуществлялось обучение.
<i>delete</i>	Удаление обученной модели.
<i>get</i>	Проверка статуса обучаемой/обученной модели.
<i>insert</i>	Обучение модели.
<i>list</i>	Получение списка созданных моделей.
<i>predict</i>	Прогнозирование на основе обученной модели.
<i>update</i>	Добавление новых данных к обученной модели.

Работа с Prediction API состоит из нескольких шагов. Во-первых, необходимо сформировать обучающую выборку. Во-вторых, необходимо загрузить обучающие данные в Google Cloud Storage (облачное хранилище данных, с которым работает Prediction API). В-третьих, следует запустить обучение на загруженных данных, используя метод *insert*. И в заключение, можно использовать обученную модель для прогнозирования новых данных с помощью метода *predict*.

4. Эксперимент

В качестве данных для обучения были использованы представленные в табл. 2 датасеты: корпус коротких текстов на русском языке на основе постов Twitter [14], созданный Юлией Рубцовой; коллекция рецензий на фильмы с сайта КиноПоиск, сформированная автором работы самостоятельно. Все тексты прошли процедуру предварительной обработки: замена ссылок и никнеймов на общие термины, удаление повторяющихся символов, приведение к нижнему регистру.

Таблица 2

Информация о корпусах тонально-аннотированных текстов

Корпус	Количество текстов		
	Общее	+	-
Корпус коротких текстов на русском языке на основе постов Twitter	226834	114911	111923
Коллекция рецензий на фильмы с сайта КиноПоиск	1500	750	750

Обучающие данные были загружены в облачное хранилище Google Cloud Storage, после чего для каждого из наборов данных был вызван метод *predict*. Информация о размерах файлов с обучающими выборками и времени обучения представлена в табл. 3.

Таблица 3

Информация об размерах файлов и времени обучения

Данные	Размер файла	Время обучения
Корпус коротких текстов на русском языке на основе постов Twitter	29.58 MB	1579 сек
Коллекция рецензий на фильмы с сайта КиноПоиск	943.18 KB	49 сек

Baseline моделью был выбран наивный Байесовский классификатор с мультиномиальной моделью распределения, комбинацией униграмм и биграмм и функцией оценки весов TF-IDF [15]. Было использовано аддитивное сглаживание для решения проблемы неизвестных слов. Реализация байесовского классификатора была взята из Python библиотеки Scikit-learn [16].

Эффективность подходов оценивалась в критерии точность *Accuracy*, который рассчитывается как отношение количества текстов из тестовой выборки, по которым классификатор принял правильное решение, к размеру текстовой выборки.

Мультиномиальный наивный Байес и Prediction API были обучены на каждом корпусе данных, после чего была произведена оценка качества. Согласно результатам (использовалась 10-кратная кросс-валидация), представленным в табл. 4. Модель Prediction API показала более высокую точность во всех случаях.

Таблица 4

Оценка эффективности бинарной классификации тональности

Данные	Подход	Точность
Корпус коротких текстов на русском языке на основе постов Twitter	MultinomialNB	75,19%
	<i>Prediction API</i>	80,00%
Коллекция рецензий на фильмы с сайта КиноПоиск	MultinomialNB	88,32%
	<i>Prediction API</i>	91,00%

5. Заключение

Таким образом, подход на основе использования Prediction API продемонстрировал высокие показатели точности работы в сравнении с мультиномиальным наивным Байесом. Качество работы подхода было проверено на двух непересекающихся наборах данных. Рассмотренный в статье метод анализа тональности с помощью Google Cloud Prediction API является эффективным и перспективным инструментом решения задачи анализа тональности. Дальнейшее развитие работы может быть направлено на использование синтаксических признаков при классификации.

Литература

1. *Thelwall M.* Sentiment analysis and time series with twitter // *Twitter and Society*. Peter Lang Publishing. – 2014. – С. 83-96.
2. *Nguyen T. H., Shirai K.* Topic Modeling based Sentiment Analysis on Social Media for Stock Market Prediction // *ACL (1)*. – 2015. – С. 1354-1364.

3. *Nguyen T. H., Shirai K., Velcin J.* Sentiment analysis on social media for stock movement prediction // *Expert Systems with Applications*. – 2015. – Т. 42. – №. 24. – С. 9603-9611.
4. *Tunggawan E., Soelistio Y. E.* And the winner is...: Bayesian Twitter-based prediction on 2016 US presidential election // *Computer, Control, Informatics and its Applications (IC3INA), 2016 International Conference on*. – IEEE, 2016. – С. 33-37.
5. *Pak A., Paroubek P.* Twitter as a Corpus for Sentiment Analysis and Opinion Mining // *LREc*. – 2010. – Т. 10. – №. 2010.
6. *Ganeshkumar M., Ramesh V.* A Study on Digital India Programme Using Azure Cloud and Twitter Data // *International Journal of Computational Intelligence Research*. – 2017. – Т. 13. – №. 5. – С. 781-790.
7. Binary Classification: Twitter sentiment analysis. URL: <https://gallery.cortanaintelligence.com/Experiment/Binary-Classification-Twitter-sentiment-analysis-4> (дата обращения: 27.07.2017).
8. *Go A., Bhayani R., Huang L.* Twitter sentiment classification using distant supervision // *CS224N Project Report, Stanford*. – 2009. – Т. 1. – №. 2009. – С. 12.
9. Ask Watson what Twitter is telling you, Part 2: Analyze the tweet text for emotions. URL: <https://www.ibm.com/developerworks/library/cc-ask-watson-part2-bluemix-trs/cc-ask-watson-part2-bluemix-trs-pdf.pdf> (дата обращения: 27.07.2017).
10. *Perrier A.* *Effective Amazon Machine Learning*. – Packt Publishing Ltd, 2017.
11. Using the Google Prediction API to Predict the Sentiment of a Tweet. URL: <https://sookocheff.com/post/prediction-api/predicting-sentiment/> (дата обращения: 27.07.2017).
12. Google Cloud Prediction API Documentation. URL: <https://cloud.google.com/prediction/docs/> (дата обращения: 27.07.2017).
13. Google Cloud Prediction API Reference. URL: <https://cloud.google.com/prediction/docs/reference/v1.6/> (дата обращения: 27.07.2017).
14. *Рубцова Ю. В.* Построение корпуса текстов для настройки тонового классификатора // *Программные продукты и системы*, 2015, №1(109), – С.72-78.
15. *Paltoglou G., Thelwall M.* A study of information retrieval weighting schemes for sentiment analysis // *Proceedings of the 48th annual meeting of the association for computational linguistics*. – Association for Computational Linguistics, 2010. – С. 1386-1395.
16. Scikit-learn: machine learning in Python. URL: <http://scikit-learn.org/stable/> (дата обращения: 27.07.2017).

С. И. СМЕТАНИН
E-MAIL: SISMETANIN@GMAIL.COM