Решение задачи распознавания блокируемых объявлений с помощью методов обработки естественных языков

А. С. Бессалов, А. П. Рыжов

В данной статье речь пойдёт о решении задачи распознавания нелегального контента в объявлениях на сайте Avito [1], опубликованной на популярном сервисе по решению задач машинного обучения Kaggle [2]. Нами был рассмотрен алгоритм, основанный на преобразовании текстовых полей методами обработки естественных языков [3] и нахождении правил из слов и словосочетаний, идентифицирующих объявление спамом. В работе были изучены основные варианты обработки текстовой информации, а также на основе экспериментов доказано, что сложная обработка, такая как построение биграмм и стемминг, не приводит к увеличению точности.

Ключевые слова: большие данные, анализ текста, обработка естественных языков, интеллектуальный анализ данных, ассоциативные правила.

Введение

Спецификой данной задачи является большой объём входных данных (около 4 млн объявлений), а также наличие текстовой информации (заголовок, описание), поэтому для её решения необходимо использовать масштабируемые алгоритмы, способные обучаться в ограниченном объёме оперативной памяти, примеры которых вы можете найти в [6]. Многие участники соревнования для решения данной задачи использовали метод онлайн-обучения Vowpal Wabbit [7] и логистическую регрессию [8], а затем объединяли построенные классификаторы в ансамбли [9], такие как бустинг или случайный лес. Однако, в данной работе мы не будем применять такие подходы. На-

ми будет построены алгоритмы, работающие на выявлении правил из текстовой информации.

Мотивация применения простых правил заключается в том, что сложные методы, основанные на ансамблях моделей, хоть и дают высокую точность, но их реализация на практике, как правило, также очень сложна, и поэтому часто не доходит до своего финального завершения. Примером тому может послужить знаменитый конкурс по рекомендательным системам от компании Netflix [10]: лучший алгоритм так и не был внедрён из-за его сложности.

В первой главе будет приведён краткий обзор входных данных. С помощью визуализации будут выявлены некоторые закономерности, на основе которых можно делать выводы о распределении объявлений. Во второй главе мы опишем основные методы преобразования текстовых полей, которые будут использованы для построения правил. В третьей главе будет описана метрика, по которой определялась лучшая модель. Далее, будет описан алгоритм построения правил и способ их прогонки на новых объявлениях. В заключении, мы протестируем алгоритм с различными входными параметрами и выберем из них лучший.

1. Исследование данных

На первом этапе анализа основной целью является ознакомление с данными. В таблице 1 приведён список входных атрибутов с их описанием и примером значений. Выявляем, что в данных примерно 7% заблокированных объявлений (is_blocked = 1). Посмотрим, как распределены разные атрибуты в зависимости от статуса блокировки. По категориальным атрибутам строим [11] гистограммы (bar graph), а по числовым — box plot.

Можно заметить, что распределения по категориям сильно отличаются (рис. 1). Наиболее часто объявления блокируются в категории «Услуги». Почти не блокируются — в категории «Недвижимость». Также, из рис. 2 видно, что в блокируемых объявлениях, как правило, указывается более низкая цена.

Название атрибута	**Описание**	**Пример**
itemid	Идентификатор объявления	10000369
category	Категория	Услуги
subcategory	Подкатегория	Предложения услуг
title	Заголовок	Установка дверей, монтаж пвх окон
description	Описание товара/услуги	гарантия,качество!
attrs	Атрибуты товара/услуги	{«Вид услуги»: «Ремонт, строительство»}
price	Цена товара/услуги	0
is_proved	Идентификатор того, что объявление закрыто опытным модератором	NA
is_blocked	Идентификатор блокировки объявления (выход для нашей модели)	0
phones_cnt	Количество номеров телефонов	0
emails_cnt	Количество адресов почт	0
urls_cnt	Количество url	0
close_hours	В течение какого времени объявление было закрыто	0.16

Таблица 1. Пример входных данных.

2. Обработка текстовых полей

В прошлой главе мы провели анализ распределений различных атрибутов — категориальных и числовых. Как же быть с текстовыми полями? Для их анализа, нам необходимо провести их предварительную обработку [12]. В таблице 2 приведён пример, какие были сделаны преобразования.

Хотелось бы отметить, что в данной задаче не стоит создавать Document-Term матрицу [12], поскольку данные сильно разреженные, и для её хранения потребуется огромное количество оперативной памяти. Мы будем преобразовывать данные в формат, представленный в таблице 3.

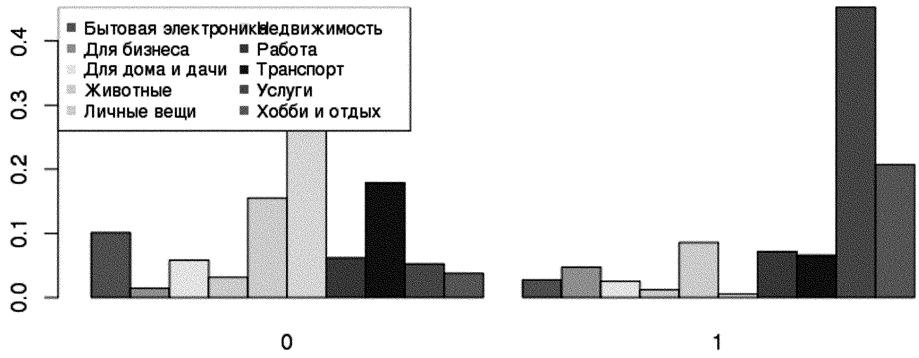

Рис. 1. Распределение категорий в зависимости от статуса блокировки.

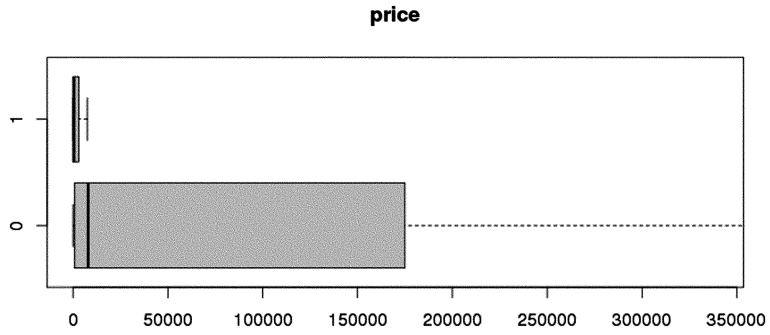

Рис. 2. Распределение цены товара/услуги в зависимости от статуса блокировки.

3. Метрика качества модели

Прежде чем строить алгоритм классификации объявлений давайте познакомимся с метрикой, по которой оценивалось качество модели. Организаторы выбрали [14] метрику MAP@k, причём в качестве k было выбрано значение 65 000, что примерно составляет 5% от всех тестовых данных, то есть примерно общее количество всех заблокированных объявлений.

Для подсчёта значения метрики MAP@k, мы должны отсортировать прогнозы в порядке убывания вероятности блокировки объявления. Далее, на каждом прогнозе высчитываем потери точности, в

Преобразование	**Строка до преобразования**	**Строка после преобразования**
Убираем цифры из текста	установка дверей, монтаж пвх окон	«установка дверей, монтаж пвх окон»
Переводим все буквы в прописные	установка дверей, монтаж пвх окон	«установка дверей, монтаж пвх окон»
Делаем стемминг русских слов по алгоритму Портера [13]	установка дверей, монтаж пвх окон	«установк двер, монтаж пвх окон»
Разбиваем текст на слова	установк двер, монтаж пвх окон	«установк», «двер», «монтаж», «пвх», «окон»
Разбиваем текст на биграммы	установк двер, монтаж пвх окон	«установк двер», «двер монтаж», «монтаж пвх», «пвх окон»

Таблица 2. Методы обработки текстовых полей.

itemid	**title_words**
10000369	установк
10000369	двер
10000369	окон

Таблица 3. Формат преобразования текстовых полей.

зависимости от количества прошлых ошибок. Затем все эти потери усредняются и получается значение метрики.

На рис. 3 изображен график, который показывает зависимость значения этой метрики от доли первых правильно распознанных заблокированных объявлений. Как видно, если мы правильно отсортируем, например, первые 25% заблокированных объявлений, а остальные 75% будут неверны, то точность нашего алгоритма будет 25%, а вот значение метрики MAP@k будет значительно выше, и равным примерно 0,6.

4. Построение модели классификации статуса блокировки

Для построения правил используем следующий алгоритм:

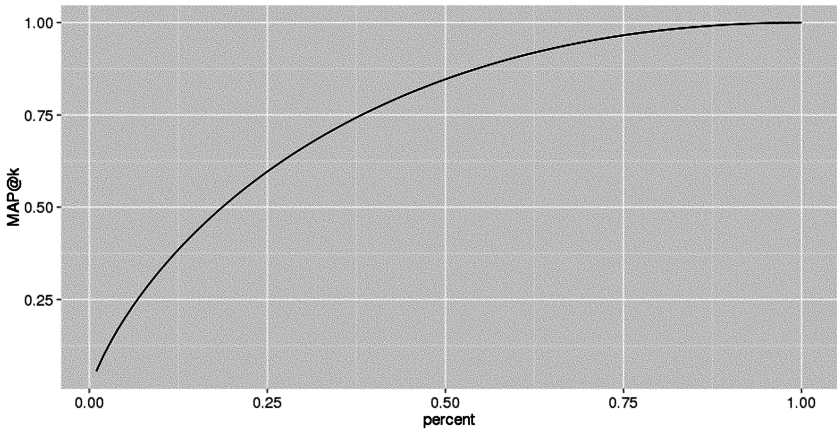

Рис. 3. Зависимость значения метрики MAP@k от доли правильно распознанных заблокированных объявлений.

- разбиваем исходное множество на обучающее (3 млн), на котором будем строить правила [15], и валидационное (1 млн), на котором будем тестировать значение метрики MAP@k.
- преобразовываем текстовые поля (title и description) как описано выше. Мы будем тестировать алгоритм отдельно со стеммингом и без него. Разбиение по словам и по биграммам будем также тестировать отдельно.
- по всем словам высчитываем условные вероятности $\mathbf{P}(\text{is\_blocked} = 1 \mid \text{word} = x)$, а также, поскольку категория сильно влияет на статус блокировки, то высчитаем и $\mathbf{P}(\text{is\_blocked} = 1 \mid \text{category} = y \ \& \ \text{word} = x)$. В терминологии ассоциативных правил [6, 16, 17] считаем у правил поддержку (support) и достоверность (confidence).
- сортируем правила по убыванию достоверности, а затем по убыванию поддержки.

Для получения прогнозов по построенным правилам, мы последовательно пробегаем по отсортированным правилам и находим объявления, для которых оно выполнено. Найденным объявлениям выставляем значения поддержки и достоверности правила и удаляем из дальнейшей прогонки. Повторяем данную процедуру пока не распознаем 5% объявлений.

На рис. 4 изображено облако слов [12] с наибольшей достоверностью. Размер слова указывает на количество объявлений, покрытых этим словом. Как видим, наиболее популярные слова в блокируемых объявлениях связаны с кредитованием, оружием и медикаментами, что логично, поскольку все эти товары запрещены правилами размещения объявлений на сайте Avito. Также, может показаться, что в облаке слов присутствует много одинаковых слов, хотя на самом деле это не так — просто эти слова написаны с использованием латинских букв. Так обычно поступают те люди, которые хотят обойти спам-фильтры.

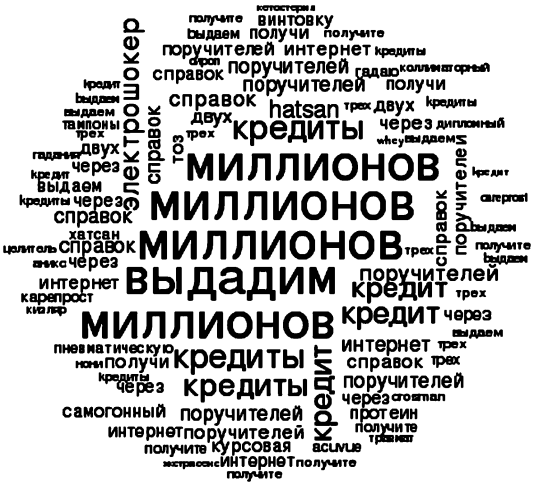

Рис. 4. Облако наиболее значимых слов в правилах.

5. Оценка точности моделей

Как уже отмечалось выше, на валидационном множестве мы будем тестировать значения метрики MAP@k в зависимости от параметров модели. К числу таких параметров относится текстовый атрибут (заголовок, описание), преобразования, которые мы будем делать над этим атрибутом, а также минимальная поддержка правил, используемых для прогонки алгоритма.

На рис. 5 изображена зависимость значений метрики MAP@k от преобразований над атрибутом title и от минимальной поддержки. Из этого графика можно сделать следующие выводы:

- если значение минимальной поддержки слишком низкое, то модель переобучена (overfitting) [15], так как присутствует много лишних правил, а если значение минимальной поддержки высокое, то наоборот, правил не хватает, и модель недообучена (underfitting). Оптимальное значение минимальной поддержки составляет примерно 15–20 объявлений.
- со стеммингом модель работает хуже.
- с биграммами модель работает хуже, чем с одиночными словами.
- наилучшая модель получается при построении правил с группировкой по категориям и без дополнительной обработки текста (без стемминга, биграмм).

Проделав такие же исследования для поля description, можно сделать аналогичные выводы.

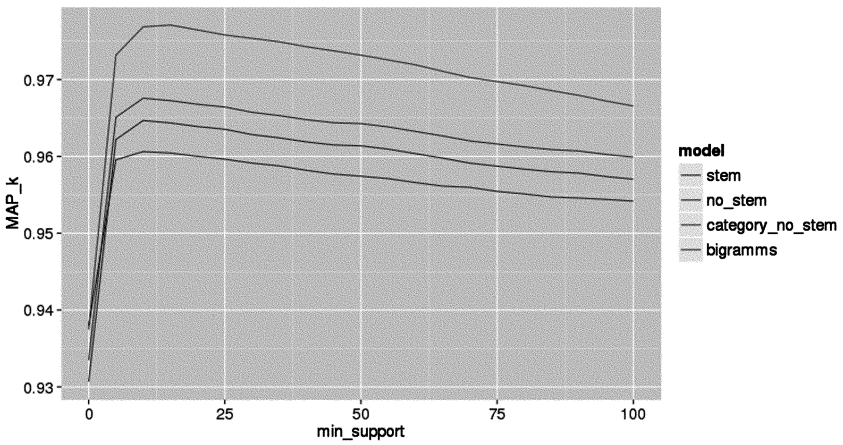

Рис. 5. График значения метрики MAP@k разных моделей в зависимости от минимальной поддержки.

После того, как мы настроили лучшие модели на валидационном множестве, применим их к тестовому множеству для оценки результатов на Kaggle. В таблице 4 приведены результаты. Мы видим, что

результаты по атрибуту title гораздо лучше результатов по атрибуту description, также, при объединении моделей мы не получили дополнительной точности, а только потеряли.

Атрибут	**MAP@k Validation set (best)**	**MAP@k Test set**
Title	0,977	0,945
Description	0,880	0,793
Title+description	0,971	0,931

Таблица 4. Результаты лучших моделей по текстовым полям на тестовом и валидационном множествах.

Заключение

В данной статье были построены модели прогнозирования заблокированных объявлений на основе построения правил из слов в текстовых полях. Лучшая модель получилась на основе преобразования текста из одного лишь заголовка, при этом, мы выяснили, что никаких дополнительных обработок слов (стемминг, биграммы) проводить не надо, поскольку они не добавляют точности. Учитывая простоту реализации данной модели на практике, значение метрики MAP@k на новых данных достаточно высокое — 0,945. Для сравнения, модели, основанные на комбинации классификаторов, дают точность $\sim 0,985$. Помимо прогнозирующей составляющей, в ходе анализа данных были выявлены закономерности, которые можно использовать для дальнейшего обучения более точного алгоритма.

Список литературы

[1] Сайт бесплатных объявлений № 1 в России: [Электронный ресурс]. URL: http://www.avito.ru/ (Дата обращения: 16.09.2014).

[2] The Hunt for Prohibited Content: [Электронный ресурс]. URL: http://www.kaggle.com/c/avito-prohibited-content/ (Дата обращения: 16.09.2014).

[3] Kao A., Poteet S. R. Natural Language Processing and Text Mining: [Текст]. — Springer, 2007.

[4] The R Project for Statistical Computing: [Электронный ресурс]. URL: http://www.r-project.org/ (Дата обращения: 16.09.2014).

[5] MySQL: [Электронный ресурс]. URL: http://www.mysql.com/ (Дата обращения: 16.09.2014).

[6] Rajaraman A., Ullman J. D. Mining of Massive Datasets [Текст]. — 2010, 2011.

[7] Vowpal Wabbit: [Электронный ресурс]. URL: http://en.wikipedia.org/wiki/Vowpal_Wabbit/ (Дата обращения: 16.09.2014).

[8] Logistic regression: [Электронный ресурс]. URL: http://en.wikipedia.org/wiki/Logistic_regression/ (Дата обращения: 16.09.2014).

[9] Ансамбль моделей: [Электронный ресурс]. URL: http://www.basegroup.ru/glossary/definitions/ensemble_models/ (Дата обращения: 16.09.2014).

[10] What the Failed $1M Netflix Prize Says About Business Advice: [Электронный ресурс]. URL: http://www.forbes.com/sites/ryanholiday/2012/04/16/what-the-failed-1m-netflix-prize-tells-us-about-business-advice/ (Дата обращения: 16.09.2014).

[11] Chang W. R Graphics Cookbook [Текст]. — O'Reily, 2012.

[12] Sanchez G. Handling and Processing Strings in R [Текст]. — 2014.

[13] Stemming: [Электронный ресурс]. URL: http://en.wikipedia.org/wiki/Stemming/ (Дата обращения: 16.09.2014)

[14] Mean Average Precision: [Электронный ресурс]. URL: https://www.kaggle.com/wiki/MeanAveragePrecision/ (Дата обращения: 16.09.2014).

[15] Lantz B. Machine Learning with R [Текст]. — PACKT, 2013.

[16] R and Data Mining: [Электронный ресурс]. URL: http://www.rdatamining.com/home/ (Дата обращения: 16.09.2014).

[17] Zhao Y., Cen Y. Data Mining Applications with R [Текст]. — ELSEVIER, 2014.