ОГЛАВЛЕНИЕ

Предисловие **11**

Вступление **12**

Благодарности **16**

Об этой книге **19**

Предполагаемая аудитория 19
Структура книги 20
Автор в сети 21

Об иллюстрации на обложке **23**

Глава 1. Готовимся к приручению текста **24**

1.1. Почему так важна задача обработки текста 25
1.2. Предварительный обзор фактографической вопросно-ответной системы 28
1.2.1. Здравствуй, доктор Франкенштейн 29
1.3. Понять смысл текста трудно 32
1.4. Прирученный текст 35
1.5. Текст и интеллектуальные приложения: поиск и не только 38
1.5.1. Поиск и сопоставление 39
1.5.2. Извлечение информации 40
1.5.3. Группировка информации 41
1.5.4. Интеллектуальное приложение 41
1.6. Резюме 42
1.7. Ресурсы 42

Глава 2. Основы приручения текста **44**

2.1. Основы лингвистики 45
2.1.1. Категории слов 46
2.1.2. Словосочетания и части предложения 48
2.1.3. Морфология 50
2.2. Популярные инструменты для обработки текста 51
2.2.1. Инструменты для манипуляций со строками 52
2.2.2. Лексемы и лексический анализ 52
2.2.3. Частеречная разметка 55

2.2.4. Стемминг 57
2.2.5. Распознавание предложений 59
2.2.6. Грамматика и грамматический анализ 61
2.2.7. Моделирование последовательности 63
2.3. Предобработка и выделение содержимого из файлов в распространенных форматах 65
2.3.1. Важность предобработки 65
2.3.2. Извлечение содержимого с помощью Apache Tika 68
2.4. Резюме 71
2.5. Ресурсы 72

Глава 3. Поиск 73

3.1. Пример фасетного поиска: Amazon.com 74
3.2. Введение в концепции поиска 77
3.2.1. Индексирование содержимого 78
3.2.2. Ввод запроса пользователем 81
3.2.3. Ранжирование документов с помощью векторной модели 85
3.2.4. Отображение результатов 89
3.3. Введение в поисковый сервер Apache Solr 92
3.3.1. Первый запуск Solr 93
3.3.2. Основные концепции Solr 95
3.4. Индексирование содержимого с помощью Apache Solr 100
3.4.1. Индексирование данных в формате XML 101
3.4.2. Извлечение и индексирование содержимого с помощью Solr и Apache Tika 103
3.5. Поиск по содержимому в Apache Solr 107
3.5.1. Параметры запроса к Solr 108
3.5.2. Построение фасетов по извлеченному содержимому 112
3.6. Факторы, влияющие на производительность поиска 115
3.6.1. Оценка качественных показателей 116
3.6.2. Оценка количественных показателей 121
3.7. Повышение производительности поиска 122
3.7.1. Совершенствование на уровне оборудования 123
3.7.2. Повышение качества анализа 124
3.7.3. Повышение качества обработки запросов 127
3.7.4. Альтернативные модели оценивания 130
3.7.5. Способы повышения производительности Solr 131
3.8. Альтернативные поисковые системы 134
3.9. Резюме 136
3.10. Ресурсы 136

Глава 4. Неточное сравнение строк 138

4.1. Различные подходы к неточному сравнению строк 140
4.1.1. Меры, основанные на множестве общих символов 141

4.1.2. Редакционные расстояния 144
4.1.3. *N*-граммное редакционное расстояние 148
4.2. Нахождение строк, неточно совпадающих с данной 150
4.2.1. Использование префиксного сравнения в Solr 151
4.2.2. Использование префиксных деревьев для префиксного сравнения 152
4.2.3. Сравнение с помощью *n*-грамм 158
4.3. Использование неточного сравнения строк в приложениях 159
4.3.1. Добавления механизма автозаполнения к поиску 160
4.3.2. Проверка орфографии запроса 164
4.3.3. Сопоставление записей 170
4.4. Резюме 177
4.5. Ресурсы 177

Глава 5. Распознавание имен людей, географических названий и других сущностей 178

5.1. Различные подходы к распознаванию именованных сущностей 180
5.1.1. Применение правил для распознавания имен и названий ... 181
5.1.2. Применение статистических классификаторов для распознавания имен и названий 182
5.2. Основы распознавания сущностей в OpenNLP 184
5.2.1. Нахождение имен и названий с помощью OpenNLP 185
5.2.2. Интерпретация имен, распознанных OpenNLP 187
5.2.3. Фильтрация имен на основе вероятности 188
5.3. Подробнее о распознавании сущностей в OpenNLP 189
5.3.1. Распознавание разнородных сущностей в OpenNLP 189
5.3.2. Под капотом: как в OpenNLP распознаются имена 193
5.4. Качество работы OpenNLP 196
5.4.1. Качество результатов 196
5.4.2. Производительность 197
5.4.3. Потребление памяти в OpenNLP 198
5.5. Настройка OpenNLP для распознавания сущностей в новой предметной области 200
5.5.1. Зачем и как обучают модель 200
5.5.2. Обучение модели OpenNLP 202
5.5.3. Изменение входных данных для модели 204
5.5.4. Другой способ моделирования имен 206
5.6. Резюме 209
5.7. Ресурсы 210

Глава 6. Кластеризация текста 211

6.1. Кластеризация документов в Google News 212

6.2. Основы кластеризации 213
6.2.1. Три типа текстов, поддающихся кластеризации 214
6.2.2. Выбор алгоритма кластеризации 216
6.2.3. Определение сходства 218
6.2.4. Пометка результатов 219
6.2.5. Как оценивать результаты кластеризации 220
6.3. Подготовка к созданию простого приложения кластеризации 222
6.4. Кластеризация результатов поиска с помощью Carrot^2 223
6.4.1. Использование Carrot^2 API 224
6.4.2. Кластеризация результатов поиска Solr с помощью Carrot^2 226
6.5. Кластеризация наборов документов с помощью Apache Mahout 229
6.5.1. Подготовка данных к кластеризации 230
6.5.2. Кластеризация методом K-средних 234
6.6. Тематическое моделирование с помощью Apache Mahout 239
6.7. Качество кластеризации 243
6.7.1. Отбор и уменьшение числа признаков 243
6.7.2. Производительность и качество Carrot2 246
6.7.3. Тесты производительности кластеризации в Mahout 247
6.8. Благодарности 254
6.9. Резюме 254
6.10. Ресурсы 255

Глава 7. Классификация, категоризация и пометка 257

7.1. Введение в классификацию и категоризацию 260
7.2. Процесс классификации 263
7.2.1. Выбор схемы классификации 265
7.2.2. Отбор признаков для категоризации 266
7.2.3. Важность обучающих данных 268
7.2.4. Оценка качества классификатора 271
7.2.5. Внедрение классификатора в эксплуатацию 274
7.3. Построение классификаторов документов с помощью Apache Lucene 276
7.3.1. Классификация текстов с помощью Lucene 276
7.3.2. Подготовка обучающих данных для классификатора MoreLikeThis 279
7.3.3. Обучение классификатора MoreLikeThis 281
7.3.4. Классификация документов с помощью классификатора MoreLikeThis 285

7.3.5. Тестирование классификатора MoreLikeThis 288
7.3.6. Классификатор MoreLikeThis в производитственной системе 291
7.4. Обучение наивного байесовского классификатора в Apache Mahout 292
7.4.1. Наивная байесовская классификация текста 293
7.4.2. Подготовка обучающих данных 294
7.4.3. Резервирование тестовых данных 298
7.4.4. Обучение классификатора 299
7.4.5. Тестирование классификатора 300
7.4.6. Усовершенствованный процесс бутстрапинга 302
7.4.7. Интеграция байесовского классификатора Mahout с Solr 304
7.5. Классификация документов с помощью OpenNLP 308
7.5.1. Регрессионные модели и классификация документов методом максимальной энтропии 309
7.5.2. Подготовка обучающих данных для классификатора документов на основе алгоритма максимальной энтропии 313
7.5.3. Обучение классификатора документов на основе алгоритма максимальной энтропии 314
7.5.4. Тестирование классификатора документов на основе алгоритма максимальной энтропии 320
7.5.5. Классификатор документов на основе алгоритма максимальной энтропии в производственной системе 322
7.6. Построение рекомендателя меток с помощью Apache Solr 323
7.6.1. Подготовка обучающих данных для рекомендателя меток 327
7.6.2. Подготовка обучающих данных 329
7.6.3. Обучение рекомендателя меток на основе Solr 330
7.6.4. Создание рекомендаций меток 332
7.6.5. Оценивание рекомендателя меток 335
7.7. Резюме 338
7.8. Ресурсы 340

Глава 8. Пример вопросно-ответной системы 341

8.1. Основы вопросно-ответной системы 343
8.2. Установка и запуск QA-системы 345
8.3. Архитектура демонстрационной вопросно-ответной системы 347
8.4. Установление смысла вопроса и порождение ответов 351
8.4.1. Обучение классификатора типов ответов 351
8.4.2. Разбиение вопроса на блоки 356
8.4.3. Вычисление типа ответа 357
8.4.4. Генерация запроса 361
8.4.5. Ранжирование фрагментов-кандидатов 362

8.5. Усовершенствование системы 365
8.6. Резюме 365
8.7. Ресурсы 366

Глава 9. Неприрученный текст: на переднем крае 367

9.1. Семантика, дискурс и прагматика: высшие уровни NLP 368
9.1.1. Семантика 369
9.1.2. Дискурс 371
9.1.3. Прагматика 373
9.2. Реферирование документов и наборов документов 375
9.3. Извлечение отношений 377
9.3.1. Обзор имеющихся подходов 379
9.3.2. Оценка 383
9.3.3. Инструменты для извлечения отношений 383
9.4. Выявление важного содержимого и людей 384
9.4.1. Глобальная важность и авторитетность 386
9.4.2. Персональная важность 386
9.4.3. Ресурсы и ссылки на тему важности 387
9.5. Распознавание эмоций с помощью анализа тональности 388
9.5.1. Исторический обзор 389
9.5.2. Инструменты и данные 391
9.5.3. Базовый алгоритм определения тональности 392
9.5.4. Дополнительные темы 394
9.5.5. Библиотеки с открытым исходным кодом для анализа тональности 396
9.6. Межъязыковой информационный поиск 397
9.7. Резюме 399
9.8. Ресурсы 400

Предметный указатель 403

ПРЕДИСЛОВИЕ

Во времена, когда спрос на высококачественные средства обработки текста растет экспоненциально, трудно назвать хотя бы одну отрасль экономики, которая не зависела бы от той или иной текстовой информации. А в связи с развитием веб-экономики эта зависимость только усиливается. И вместе с ней быстро возрастает потребность в талантливых технических специалистах. Вот в таких условиях выходит на свет отличная, практически ориентированная книга «Обработка неструктурированных текстов», в которой вы найдете проверенные на реальном опыте рекомендации и инструкции.

Грант Ингерсолл и Дрю Фэррис, два блистательных и в высшей степени квалифицированных инженера-программиста, с которыми я работала много лет, и Тим Мортон, внесший немалый вклад в обработку естественного языка (natural language processing, NLP), предлагают прагматическое руководство тем, кто хотел бы войти в избранный круг специалистов по обработке текстов,

Грант, Дрю и Том выбрали подход, который я называю «обучение на практике ради практики», и сумели сорвать покров тайны с действительно очень сложных процессов. Для этого они не пошли по длинному пути – теоретическому семестровому курсу по NLP, а сосредоточились на существующих инструментах, реализованных до конца примерах и хорошо протестированном коде.

Для инженера-программиста этих основ будет достаточно, чтобы открыть дверь в мир примеров и упоминаемых проектов с открытым исходным кодом. И гораздо быстрее, чем вам кажется, вы превратитесь в настоящего эксперта, готового к решению реальных задач.

Лиз Лидди
Декан, ISchool,
Сиракузский Университет

ВСТУПЛЕНИЕ

Жизнь полна удивительных сюрпризов, и некоторые из них оказали определяющее влияние на мою карьеру. Было это в конце 1990 годов, я тогда был молодым программистом, занимался моделированием распределения электромагнитных полей и случайно наткнулся на предложение места разработчика в небольшой компании в городе Сиракузы, штат Нью-Йорк, которая называлась TextWise. Прочитав описание работы, я подумал, что едва ли подойду, но решил все-таки попробовать и отправил резюме. Непонятно почему, меня взяли, и так началась моя карьера в области обработки естественных языков. Кто бы мог подумать, что спустя столько лет я так и буду заниматься поиском и NLP и даже напишу книгу на эту тему.

А тогда моей первой задачей стало участие в разработке межъязыковой информационно-поисковой системы (CLIR), которая позволяла пользователю вводить запросы на английском языке, а находить и автоматически переводить документы на французском, испанском или китайском. Оглядываясь назад, я понимаю, что в первой же системе, над которой я работал, встретились все те трудные проблемы работы с текстом, которые я впоследствии так полюбил: поиск, классификация, извлечение информации, машинный перевод, а также специфические правила конкретных языков, способные свести с ума любого, кто изучает грамматику. После этого проекта я работал над самыми разными системами поиска и обработки естественных языков – от классификаторов на основе правил до вопросно-ответных систем. Позже, в 2004 году, уже на новой работе в Центре обработки естественных языков я столкнулся с Apache Lucene, поисковой системой с открытым исходным кодом, которая в те дни являлась стандартом де факто. И снова мне пришлось разрабатывать CLIR-систему, только теперь для английского и арабского языков. Поскольку мне потребовались кое-какие функции, которых в Lucene не было, я начал писать дополнения и исправлять ошибки. Спустя какое-то время я стал отправлять плоды своих трудов в репозиторий исходного кода. И пошло-поехало. Я пристрастился к проектам с открытым исходным

кодом, начав с системы машинного обучения Apache Mahout вместе с Изабель Дрост (Isabel Drost) и Карлом Веттином (Karl Wettin), а потом стал сооснователем компании Lucid Imagination, которая специализировалась на задачах поиска и анализа текстов с применением Apache Lucene и Solr.

Описав полный круг, я пришел к выводу, что поиск и NLP принадлежат к числу вопросов, определяющих предмет информатики, поскольку требуют изощренных подходов как к структурам данных, так и к алгоритмам решения задач. Добавьте сюда требования к масштабируемости, необходимой для обработки гигантских объемов данных, порождаемых пользователями веб вообще и социальных сетей в частности, – и вот вам мечта любого разработчика. Эта книга призвана заполнить пустующую на тот момент рыночную нишу – текст, написанный инженерами для инженеров и посвященный, прежде всего, использованию существующих, проверенных практикой библиотек с открытым исходным кодом для решения трудных задач обработки текста. Надеюсь, что она поможет вам в повседневной работе, а также откроет мир текстовых данных – богатейшую возможность для изучения нового.

Грант Ингерсолл

Я подпал под очарование искусственного интеллекта на втором курсе вуза, а на старших курсах решил остаться в аспирантуре и сосредоточиться на обработке естественных языков. В Пенсильванском университете я очень много узнал об обработке текстов, машинном обучении, а также об алгоритмах и структурах данных вообще. У меня также была возможность работать с некоторыми из лучших специалистов в области обработки естественных языков и набираться у них ума-разума.

В аспирантуре я занимался различными NLP-системами и принимал участие в ряде финансируемых агентством DARPA исследований по кореференции, свертыванию и порождению ответов на вопросы. В ходе этой работы я познакомился с системой Lucene и движением за ПО с открытым исходным кодом в целом. Я также обратил внимание на пробел в открытых программах обработки текстов, заполнение которого могло бы обеспечить эффективную сквозную обработку. Работая над диссертацией, я активно участвовал в проекте OpenNLP, а также продолжал изучать NLP-системы, разрабатывая систему автоматизированной оценки сочинений и кратких ответов в службе образовательного тестирования (Educational Testing Services).

Тесное сотрудничество с разработчиками ПО с открытым исходным кодом научило меня коллективной работе и позволило усовершенствоваться в своей профессии. Сейчас я работаю в компании Comcast Corporation с командами программистов, которые применяют многие описанные в этой книге приемы и инструменты. Надеюсь, эта книга станет мостом между напряженно ищущими исследователями типа тех, у кого я учился в аспирантуре, и инженерами-практиками, цель которых – использовать обработку текстов для решения реальных задач в интересах обычных людей.

Томас Мортон

Я, как и Грант, получил первое представление об информационном поиске и обработке естественных языков под руководством д-ра Элизабет Лидди, Вуджина Пайка (Woojin Paik) и прочих сотрудников компании TextWise в середине 1990-х годов. В то время TextWise была в стадии превращения из исследовательской группы в новообразованную компанию, специализирующуюся на разработке приложений на основе полученных результатов в области обработки текста. Я работал в компании много лет и все это время занимался самообразованием, открывал для себя что-то новое и общался с выдающимися людьми, которые, не убоявшись трудностей, решили научить машины понимать различные аспекты человеческого языка.

Лично я подхожу к проблеме анализа текста, прежде всего, с точки зрения разработчика программного обеспечения. Мне повезло работать с блестящими учеными и превращать их идеи из экспериментов сначала в функционирующие прототипы, а затем и в массивно масштабируемые системы. По ходу дела у меня была возможность плотно заниматься тем, что теперь принято называть наукой о данных, и я глубоко и навсегда полюбил исследование больших наборов данных и методы извлечения информации из них.

Невозможно переоценить то огромное влияние, которое открытое ПО оказало на мою карьеру. Наличие под рукой исходного кода как подспорья в исследованиях, – невероятно эффективный способ изучения новых методов и подходов к анализу текста и к разработке ПО вообще. Я приветствую всякого, кто приложил усилия, чтобы поделиться своими знаниями и опытом с другими людьми, страстно желающими сотрудничать и учиться. И особо я хочу поблагодарить отличных ребят из фонда Apache Software Foundation, неустанно взращивающих динамичную экосистему, которая способствует раз-

работке открытого ПО и помогает организовывать процессы и сплачивать людей, это ПО поддерживающих.

Инструменты и методы, описанные в этой книге, своими корнями уходят глубоко в сообщество разработчиков ПО с открытым исходным кодом. Lucene, Solr, Mahout и OpenNLP – все эти проекты выросли под опекой Apache. В этой книге мы лишь скользнули по поверхности того, что умеют эти инструменты. Нашей целью было продемонстрировать базовые концепции, лежащие в основе обработки текстов, и заложить прочный фундамент под будущие исследования в этой области.

Успехов в кодировании!

Дрю Фэррис

ОБ ЭТОЙ КНИГЕ

«Обработка неструктурированных текстов» – это книга о создании программных приложений, ценность которых состоит, главным образом, в использовании и манипулировании содержимым обычных письменных текстов. Это не теоретический трактат по поиску, обработке естественного языка и машинному обучению, хотя все эти вопросы обсуждаются довольно подробно. Мы стремились избегать специальной терминологии и сложной математики, а сосредоточиться на концепциях и примерах, необходимых современным программистам, архитекторам и пользователям для реализации интеллектуальных приложений обработки текста нового поколения. Кроме того, наша твердая позиция – демонстрировать примеры из реальной практики с помощью бесплатных, широко распространенных инструментов с открытым исходным кодом – Apache Solr, Mahout, OpenNLP и других.

Предполагаемая аудитория

Будет ли эта книга полезна вам? Возможно. Мы ориентировались на программистов-практиков, не имеющих солидной теоретической подготовки в проблемах поиска, обработки естественного языка и машинного обучения. На самом деле, книга рассчитана на людей, с которыми мы встречались во многих компаниях: команду разработчиков, которой поручено добавить поиск и другие средства в уже существующее приложение при том, что мало кто из них (а то и вовсе никто) не имеет опыта работы с текстом. Им необходимо хорошее введение в суть предмета, не отягощенное ненужными деталями.

Часто мы отсылаем читателя к легко доступным источникам типа википедии и фундаментальным научным статьям, тем самым подготавливая стартовую площадку для, кто хотел бы изучить предмет более подробно. И еще – хотя большинство инструментов и примеров написаны на Java, сами идеи легко переносятся на многие другие языки программирования, поэтому пишущие на Ruby, Python или еще каком-то языке тоже получат пользу от чтения этой книги.

Эта книга определенно не подойдет тем, кому интересны объяснения математических основ описываемых систем, или тем, кто жаждет академической строгости изложения, хотя, как нам кажется, она пригодится студентам, когда перед ними встанет задача реализовать идеи, почерпнутые из лекций и других книг академической направленности.

Не рассчитана эта книга и на опытных специалистов-практиков, которые за свою карьеру написали не одно приложение для обработки текстов, хотя и они смогут найти в ней подсказку-другую о том, как работать с описываемыми пакетами с открытым исходным кодом. Не раз мы слышали от практиков, что эта книга очень помогает, когда нужно быстро обучить новых членов команды концепциям, относящимся к созданию приложений для обработки текстов.

В общем и целом, мы надеемся, что эта книга станет актуальным пособием для современного программиста, пособием, которого всем нам так не хватало, когда мы только начинали свою карьеру в области обработки текста.

Структура книги

В главе 1 объясняется, почему задача обработки текста важна и в чем ее трудность. Мы дадим предварительный обзор вопросно-ответной фактографической системы, подготовив сцену для «приручения» текста с применением открытых библиотек.

В главе 2 описываются основные элементы обработки текста: лексический анализ, разбиение на блоки, грамматический разбор и частеречная разметка. Затем мы поговорим о том, как извлекать текст из файлов в распространенных форматах с помощью проекта Apache Tika с открытым исходным кодом.

Глава 3 посвящена теории поиска и основам векторной модели. Мы познакомимся с поисковым сервером Apache Solr и покажем, как с его помощью индексировать документы. Вы узнаете о количественных и качественных оценках работы поисковой системы.

В главе 4 рассматривается неточный поиск в строке с помощью префиксов и *n*-грамм. Мы рассмотрим две характеристики близости строк – меру Жаккарда и расстояние Джаро-Винклера – и объясним, как с помощью Solr находить и ранжировать соответствия.

В главе 5 представлены основные концепции распознавания именованных сущностей. Мы покажем, как находить именованные сущности с помощью проекта OpenNLP, и обсудим некоторые аспекты его функционирования. Мы также рассмотрим вопрос о на-

стройке OpenNLP на распознавание сущностей новой предметной области.

Глава 6 посвящена кластеризации текста. Из нее вы узнаете об основах стандартных алгоритмов кластеризации текстов и увидите, как кластеризация может повысить качество приложения. Мы также объясним, как производить кластеризацию целых наборов документов с помощью Apache Mahout и как кластеризовать результаты поиска с помощью Carrot[2].

В главе 7 обсуждаются основы классификации, категоризации и грамматической разметки. Мы покажем, как категоризация применяется в приложениях для обработки текста и как можно построить, обучить и использовать классификатор с помощью открытых инструментов. Мы также воспользуемся реализацией алгоритма наивной байесовской фильтрации в проекте Mahout для построения категоризатора документов.

Графические выделения и загрузка исходного кода

В этой книге много примеров кода. Код выделяется моноширинным шрифтом, чтобы было проще отличить его от обычного текста. Элементы программы, например имена методов, классов и т. д., также выделяются моноширинным шрифтом.

Многие листинги сопровождаются аннотациями, иллюстрирующими основные идеи, и пронумерованными маркерами, на которые даются ссылки в последующих пояснениях.

Многие приведенные в книге примеры довольно близки к тем, что можно найти в сети. Но для краткости мы иногда удаляли некоторые части, например комментарии, чтобы код поместился на странице.

Исходный код к этой книге можно скачать с сайта издательства по адресу www.manning.com/TamingText.

Автор в сети

Приобретение книги «Обработка неструктурированных текстов» открывает бесплатный доступ к закрытому форуму, организованному издательством Manning Publications, где вы можете оставить свои комментарии к книге, задать технические вопросы и получить помощь от автора и других пользователей. Получить доступ к форуму и подписаться на список рассылки можно на странице www.manning.com/TamingText. Там же написано, как зайти на форум после регис-

трации, на какую помощь можно рассчитывать, и изложены правила поведения в форуме.

Издательство Manning обязуется предоставлять читателям площадку для общения с другими читателями и автором. Однако это не означает, что автор обязан как-то участвовать в обсуждениях; его присутствие на форуме остается чисто добровольным (и не оплачивается). Мы советуем задавать автору какие-нибудь хитроумные вопросы, чтобы его интерес к форуму не угасал!

Форум автора в сети и архивы будут доступны на сайте издательства до тех пор, пока книга не перестанет печататься.