Содержание

Об авторе

Ричард Риз (Richard M Reese) имеет опыт работы не только в промышленности, но и в науке. В течение 17 лет он работал в отрасли телефонной связи и аэрокосмической индустрии, выполняя разные обязанности, включая исследовательскую и конструкторскую деятельность, разработку программного обеспечения, руководство группами разработчиков и преподавание. В настоящее время Ричард преподает в Тарлтонском государственном университете (Tarleton State University), где применяет свой богатый практический опыт для усовершенствования курсов обучения.

Ричард написал несколько книг по языкам программирования Java и C. Он излагает материал в лаконичном и понятном стиле. Среди его книг можно отметить «EJB 3.1 Cookbook», книги о новых функциональных возможностях Java 7 и 8, книгу о сертификации разработчика на языке Java, книгу о jMonkey Engine, а также книгу об использовании указателей в языке C.

«Я благодарен моей дочери Дженнифер за многочисленные замечания, поправки и дополнения. Ее вклад в создание данной книги невозможно переоценить».

О рецензентах

Сурьяпракаш С. В. (Suryaprakash C. V.) занимается задачами обработки естественного языка с 2009 года. По окончании учебного заведения получил степень бакалавра по физике, затем в аспирантуре специализировался в области разработки компьютерных приложений. Позже получил возможность продолжить карьеру в наиболее привлекательной для него области – обработке естественного языка.

В настоящее время Сурьяпракаш является ведущим исследователем в компании Senseforth Technologies.

> «Я благодарен коллегам за постоянную поддержку во всем. Их помощь была весьма существенной и при составлении данной рецензии».

Эван Демпси (Evan Dempsey) – программист из Уотерфорда (Ирландия). В редкие часы, когда он отвлекается от языка Python (который использует не только для заработка, но и с большим удовольствием), Эван любит посидеть с кружкой знаменитого ирландского пива, заняться программированием на Common Lisp и почитать о новейших достижениях в области машинного обучения (machine learning). Он также участвует в нескольких проектах с открытым исходным кодом.

Анил Оманвар (Anil Omanwar) – весьма энергичная личность с огромным интересом к новейшим направлениям в технологиях и исследованиях. Обладает более чем 8-летним опытом исследовательской работы в области когнитивных (интеллектуальных) вычислений. Обработка естественного языка, машинное обучение, визуальное представление информации и интеллектуальный анализ текстов – главные сферы его исследовательского интереса.

Анил является экспертом в анализе эмоциональной окраски текста (sentiment analysis), составлении и анализе опросных листов, кластеризации текстов (документов) и извлечении информации (связных фраз) в разнообразных сферах знаний, таких как науки о жизни (биология, медицина, антропология и т. п.), производство, розничная торговля, электронная коммерция, представительство и работа с клиентами, банковское дело, социальные медиакоммуникации.

В настоящее время Анил активно сотрудничает с лабораториями обработки естественного языка и IBM Watson в корпорации IBM. Тема его исследований – автоматизация важнейших ручных операций и помощь экспертов в соответствующих областях знаний для оптимизации функциональных возможностей систем «человек–машина».

Свое свободное время Анил посвящает общественной работе, пешеходному туризму, фотографии и путешествиям. Он всегда готов заняться решением любой, самой сложной технической задачи.

Амитабх Шарма (Amitabh Sharma) – профессиональный программист. В его активе множество промышленных приложений в сферах телекоммуникации и бизнес-аналитики. Профессиональные интересы Амитабха – сервис-ориентированные архитектуры, хранение данных и такие языки программирования, как Java, Python и др.

Предисловие

Обработка текстов на естественных языках (Natural Language Processing, NLP) используется для решения обширного класса задач, включающего поддержку механизмов поиска, аннотирования и классификации текстов на веб-страницах, а также для внедрения методов машинного обучения с целью решения таких нетривиальных задач, как распознавание речи и анализ запросов. Технология обработки естественного языка наиболее эффективна при работе с документами, содержащими полезную информацию.

Обработка текстов на естественных языках применяется для расширения функциональных возможностей приложений, например для упрощения ввода пользователем исходных данных и преобразования текста в более удобные формы. Суть технологии NLP состоит в обработке текстов на естественном языке, взятых из самых разнообразных источников. При этом используется последовательность ключевых операций обработки естественного языка для преобразования исходного текста или извлечения из него полезной информации.

В этой книге подробно рассматриваются ключевые операции NLP, которые с большой вероятностью можно встретить в NLP-приложениях. Для каждой операции, рассматриваемой в книге, сначала дается описание задачи и области ее применения. Далее перечисляются все нюансы, определяющие сложность задачи, чтобы читатель смог лучше понять данную задачу в целом. Затем следуют многочисленные примеры решений на языке Java, а также примеры использования прикладных программных интерфейсов (API) поддержки обработки естественного языка.

Краткий обзор содержания книги

Глава 1 «Основы обработки естественного языка» описывает области применения обработки естественного языка и важное значение этой технологии. Методики NLP, используемые в этой главе, объясняются на простых практических примерах.

Глава 2 «Поиск фрагментов текста» в основном посвящена операциям разделения потока текста на смысловые фрагменты (фразы, слова, символы и т. п.)[1]. Это первый этап подготовки к решению более

[1] Такие операции обозначаются термином токенизация (tokenization). – *Прим. перев.*

сложных задач NLP. Также описываются программные интерфейсы Java для разделения текста на фрагменты и поиска по образцам.

Глава 3 «Поиск предложений» демонстрирует важную роль еще одной задачи обработки естественного языка – определение границ предложений. Этот этап обработки предшествует многим другим NLP-задачам, в которых текстовые элементы не должны выходить за границы предложений, включая гарантии вхождения всех фраз в одно предложение, а также частичную поддержку распознавания и анализа речи.

Глава 4 «Поиск людей и именованных объектов» посвящена тому, что в широком смысле обычно обозначают термином «распознавание и идентификация именованных объектов» (named-entity recognition). Это задача идентификации людей, местоположений и прочих подобных объектов в тексте. Данная методика является подготовительным этапом для обработки запросов и операций поиска.

Глава 5 «Определение частей речи» рассказывает, как определяются части речи грамматических элементов текста, такие как существительные и глаголы. Идентификация этих элементов является важным этапом в процессе определения общего смыслового значения исследуемого текста и установления смысловых связей внутри текста.

Глава 6 «Классификация текстов и документов» наглядно показывает необходимость классификации текста для таких задач, как выявление спама и анализ эмоциональной окраски текста (sentiment analysis). Подробно описывает методики NLP, которые обеспечивают поддержку классификации текстов.

Глава 7 «Использование синтаксического анализатора (парсера) для выделения взаимосвязей» рассматривает деревья синтаксического анализа. Дерево синтаксического анализа используется для многих целей, в том числе для извлечения информации. Оно содержит данные об отношениях между своими элементами. Для наглядной демонстрации процесса в главе приведен пример реализации простого запроса.

Глава 8 «Комплексные методики» рассматривает способы извлечения данных из документов различных типов, таких как PDF и файлы, созданные в MS Word. Здесь показано, как объединить NLP-методики, описанные в предыдущих главах, в своеобразный «конвейер» для решения более крупномасштабных задач.

Что нужно для чтения этой книги

Для демонстрации работы методик обработки естественного языка используется Java SDK 7. Потребуются также разнообразные NLP

API, которые нетрудно будет найти и скачать. Наличие интегрированной среды разработки (IDE) не обязательно, но желательно.

Для кого эта книга

Книга будет полезна опытным разработчикам на Java, интересующимся технологиями обработки естественного языка. Для ее чтения не требуется предварительное знакомство с NLP.

Соглашения, принятые в книге

В книге вы обнаружите несколько стилей текста, которые позволяют выделять различные виды информации. Ниже приведены примеры этих стилей и описание их смысла.

Ключевые слова языка программирования, имена классов, переменных, методов и т. п. оформляются моноширинным шрифтом: «Метод `keyset` возвращает набор всех ключей аннотации, в текущий момент содержащихся в объекте `Annotation`».

Имена таблиц баз данных, каталогов, файлов, расширения файлов, новые термины и важные замечания оформлены курсивом: «Для демонстрации применения POI воспользуемся файлом с именем *TestDocument.pdf*».

Блоки программного кода и листинги оформлены моноширинным шрифтом:

```
for (int index = 0; index < sentences.length; index++) {
    String tokens[] = tokenizer.tokenize(sentences[index]);
    Span nameSpans[] = nameFinder.find(tokens);
    for(Span span : nameSpans) {
        list.add("Sentence: " + index
            + " Span: " + span.toString() + " Entity: "
            + tokens[span.getStart()]);
    }
}
```

Моноширинным шрифтом также оформляются ввод пользователя и вывод результатов работы программ на экран:

```
Sentence: 0 Span: [0..1) person Entity: Joe
Sentence: 0 Span: [7..9) person Entity: Fred
Sentence: 2 Span: [0..1) person Entity: Joe
```

Так оформляются предупреждения или важные примечания.

Так оформляются советы и рекомендации.

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв прямо на нашем сайте www.dmkpress.com, зайдя на страницу книги и оставив комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу dmkpress@gmail.com, при этом напишите название книги в теме письма.

Если есть тема, в которой вы квалифицированы, и вы заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу http://dmkpress.com/authors/publish_book/ или напишите в издательство по адресу dmkpress@gmail.com.

Скачивание исходного кода примеров

Скачать файлы с дополнительной информацией для книг издательства «ДМК Пресс» можно на сайте www.dmkpress.com или www.дмк.рф в разделе «Читателям – Файлы к книгам».

Список опечаток

Хотя мы приняли все возможные меры для того, чтобы удостовериться в качестве наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг – возможно, ошибку в тексте или в коде, – мы будем очень благодарны, если вы сообщите нам о ней. Сделав это, вы избавите других читателей от расстройств и поможете нам улучшить последующие версии этой книги.

Если вы найдете какие-либо ошибки в коде, пожалуйста, сообщите о них главному редактору по адресу dmkpress@gmail.com, и мы исправим это в следующих тиражах.

Нарушение авторских прав

Пиратство в Интернете по-прежнему остается насущной проблемой. Издательства «ДМК Пресс» и «Packt» очень серьезно относятся к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в Интернете с незаконно выполненной копией любой нашей кни-

ги, пожалуйста, сообщите нам адрес копии или веб-сайта, чтобы мы могли принять меры.

Пожалуйста, свяжитесь с нами по адресу электронной почты dmk-press@gmail.com со ссылкой на подозрительные материалы.

Мы высоко ценим любую помощь по защите наших авторов, помогающую нам предоставлять вам качественные материалы.

Вопросы

Вы можете присылать любые вопросы, касающиеся данной книги, по адресу dm@dmk-press.ru или questions@packtpub.com. Мы постараемся разрешить возникшие проблемы.

Глава 1

Основы обработки естественного языка

Обработка естественного языка (*Natural Language Processing, NLP*) – это обширная область ИТ, связанная с использованием компьютеров для анализа естественных языков, к которой относятся такие дисциплины, как распознавание и обработка речи, выделение смысловых отношений, категоризация документов, а также реферирование и аннотирование текста. Но все эти виды анализа основаны на относительно небольшой группе базовых методик: разделение потока текста на фрагменты – *токенизация* (*tokenization*), определение границ предложений, классификация и выделение отношений (между элементами текста). Именно эти основополагающие методики и являются главными темами данной книги. Мы начнем с подробного рассмотрения обработки естественного языка в целом, попробуем разобраться, почему она столь важна, и определим области ее применения.

Для поддержки задач обработки естественного языка существует множество доступных инструментальных средств. Мы сосредоточим внимание на использовании языка программирования Java и различных *прикладных программных интерфейсов* (*Application Programming Interface, API*) на языке Java поддержки NLP. В этой главе будут кратко описаны основные программные интерфейсы, включая OpenNLP (Apache), библиотеки Stanford NLP, LingPipe и GATE.

Затем будут обсуждаться основные методики обработки естественного языка, рассматриваемые в данной книге. С помощью одного из прикладных программных интерфейсов будут представлены и продемонстрированы сущность и примеры использования этих методик. Многие из этих методик используют так называемые модели. Модели похожи на наборы правил, применяемых для выполнения, например, такой задачи, как разделение текста на фрагменты, и обычно представлены в виде классов, экземпляры которых хранятся в файлах.

Завершается глава кратким описанием процесса подготовки данных для поддержки задач NLP.

В целом проблему обработки текстов на естественных языках нельзя назвать простой. Несмотря на то что некоторые ее задачи могут быть решены относительно легко, многие другие задачи требуют применения изощренных методик. В этой книге сделана попытка предоставить всю необходимую фундаментальную информацию, чтобы читатель смог лучше понять, какие методики доступны и применимы для каждой конкретной поставленной задачи.

Обработка естественного языка представляет собой весьма обширную и сложную область ИТ. В данной книге рассматривается лишь небольшая ее часть. Мы сосредоточимся на ключевых задачах NLP, которые могут быть реализованы на языке программирования Java. На протяжении всей книги будет демонстрироваться практическое применение методик NLP с использованием Java SE SDK и других библиотек, таких как OpenNLP и Stanford NLP. Для работы с конкретными библиотеками в проект должны быть включены специальные JAR-файлы. Описание библиотек можно найти в разделе «Обзор инструментальных средств обработки естественного языка», где также приведены ссылки на эти библиотеки. Примеры, приводимые в книге, были разработаны с помощью IDE NetBeans 8.0.2. Перед сборкой проектов необходимо добавить требуемые JAR-файлы в категорию **Библиотеки** (Libraries) в диалоговом окне **Свойства проектов** (Projects Properties).

Что такое обработка естественного языка

Формальное определение обработки текстов на естественных языках часто представлено в таком изложении: область исследований, использующая информатику (computer science), искусственный интеллект и понятия формальной лингвистики для анализа естественного языка. Менее формальное определение полагает, что это набор инструментальных средств для извлечения содержательной и полезной информации из источников, сформированных на естественном языке, таких, например, как веб-страницы и текстовые документы.

Характеристики «содержательная» и «полезная» означают, что информация должна обладать некоторой коммерческой ценностью, хотя часто такая информация используется также для решения научных теоретических проблем. Наиболее очевидно это проявляется в поддержке поисковых механизмов. Пользовательский запрос обра-

батывается с применением методик обработки естественного языка, чтобы сгенерировать страницу результатов, наиболее удобную для пользователя. Современные механизмы поиска в этом отношении обладают весьма впечатляющими достижениями. Кроме того, методики обработки естественного языка нашли применение в автоматизированных системах помощи (подсказки) и в поддержке сложных систем запросов, например в известном проекте Watson корпорации IBM.

При работе с любым языком часто встречаются термины *«синтаксис»* (*syntax*) и *«семантика»* (*semantics*). Синтаксис языка определяет правила, управляющие правильной структурой предложения. В английском языке, например, общая структурная схема предложения начинается с подлежащего (существительное), за которым следует сказуемое (глагол) и далее – дополнение: «Tim hit the ball» (Тим ударил по мячу). Изменение порядка слов в предложении в английском языке (и в некоторых других) считается неправильным и не употребляется: «Hit ball Tim»[1]. Несмотря на то что синтаксические правила английского языка не столь строги, как их аналоги для компьютерных языков, мы вправе ожидать, что предложение будет соответствовать основной структурной схеме.

Семантика предложения – это его смысл. Люди, говорящие по-английски, понимают смысл предложения «Tim hit the ball». И все же как английский, так и другие естественные языки иногда могут быть неоднозначными, поэтому смысл предложения можно правильно определить только из его контекста. Далее мы увидим, как использовать различные методики машинного обучения для определения смысла текста.

В процессе чтения вы встретите множество лингвистических терминов, которые помогут лучше понимать естественные языки и сформируют небольшой специализированный словарь, полезный для описания различных методик обработки естественного языка. Вы узнаете, как разделить текст на отдельные элементы и как провести классификацию этих элементов.

Вообще говоря, рассматриваемые методики используются для улучшения приложений, то есть для того, чтобы приложения стали более полезными. Область применения обработки естественного язы-

---

[1] В этом отношении правила русского языка более свободны и зачастую позволяют произвольно менять порядок слов в предложении – например, знаменитые фразы из букваря: «Мама мыла раму» – «Раму мыла мама» и т. п. – *Прим. перев.*

ка охватывает широкий диапазон: от относительно простых случаев до «технологий завтрашнего дня». В книге показаны примеры, демонстрирующие простые методики, которых вполне достаточно для решения некоторых практических задач, а также библиотеки и классы, обладающие более развитыми функциональными свойствами, позволяющими справиться с комплексными сложными проблемами.

Для чего используется обработка естественного языка

Обработка естественного языка используется во многих научных и прикладных дисциплинах для решения разнообразных типов задач. Анализ текста выполняется и при вводе пользователем данных, состоящих из нескольких слов, и при формировании интернет-запроса для поиска многочисленных документов, которые необходимо обработать. В последние годы наблюдается лавинообразный рост объема доступных неструктурированных данных, представленных в таких формах, как блоги, твиты и прочие социальные сети. Технология обработки естественного языка является наиболее подходящим инструментом для анализа этого типа информации.

Машинное обучение и анализ текста часто используются для расширения функциональных возможностей приложений. Вот краткий список областей применения обработки естественного языка:

- *поиск*: идентификация заданных элементов текста. Может быть простым, например поиск всех вхождений заданного имени в документе, или выполняться с использованием синонимов, а также различного и даже неверного (неточного) написания для поиска объектов, более или менее совпадающих с исходной строкой-образцом;
- *машинный перевод*: обычно предполагает перевод с одного естественного языка на другой;
- *аннотирование*, или *реферирование* (*summation*): для абзацев, статей, документов или подшивок документов может потребоваться аннотирование (реферирование). NLP-технология успешно справляется с этой задачей;
- *распознавание и идентификация именованных объектов* (*Named Entity Recognition, NER*): имеется в виду извлечение из текста имен людей, названий географических и прочих объектов. Как правило, эта методика используется в сочетании с другими за-

дачами обработки естественного языка, например с обработкой запросов;

- *группирование информации*: это важный этап обработки – на основе исходных текстовых данных создается набор категорий, отображающих содержание документа. Вам наверняка встречались веб-сайты, где информация достаточно удобно организована по категориям, а список этих категорий размещен в левой части страницы;
- *морфологическая разметка* (*Parts of Speech Tagging, POST*): при решении данной задачи текст разбивается на различные грамматические элементы: существительные, глаголы и т. п. Это удобно для дальнейшего анализа текста;
- *анализ эмоциональной окраски текста* (*sentiment analysis*): с помощью этой методики можно определять, какие чувства люди испытывают и как относятся к фильмам, книгам и многим другим вещам (в том числе и к предлагаемым товарам). С коммерческой точки зрения полезно знать, как покупатели воспринимают тот или иной продукт, применяя автоматизированную методику;
- *ответы на вопросы*: этот тип обработки был наглядно продемонстрирован, когда программа Watson, разработанная корпорацией IBM, выиграла телеигру Jeopardy[1]. Но область применения этой методики вовсе не ограничивается игровыми шоу, она используется во многих областях человеческой деятельности, например в медицине;
- *распознавание речи*: человеческую речь анализировать трудно. Большинство достижений в этой области являются результатами применения технологии обработки естественного языка;
- *генерация текстов на естественном языке* (*Natural Language generation, NLG*): это процесс генерации связного текста из данных или из специализированного источника данных (знаний), например из базы данных. Методика позволяет автоматизировать создание отчетов и информационных сводок, таких как метеорологические бюллетени или результаты медицинских обследований.

В задачах обработки естественного языка часто используются методики машинного обучения. В обобщенном виде суть этого подхода такова: сначала модель обучается выполнению поставленной задачи,

[1] В России аналогичная телеигра называется «Своя игра». – *Прим. перев.*

проверяется корректность выбранной модели, затем эта модель применяется для решения реальной задачи. Более подробно мы рассмотрим этот процесс в разделе «Основы использования моделей в обработке естественного языка» ниже.

Трудности обработки естественного языка

Как отмечалось выше, проблему обработки естественного языка нельзя назвать простой. Трудности возникают по ряду объективных причин. Например, существуют сотни естественных языков, в каждом из которых имеются свои синтаксические правила. Слова могут иметь разный смысл в зависимости от контекста употребления. Ниже мы подробнее остановимся на некоторых наиболее важных факторах, затрудняющих обработку естественного языка.

Даже на уровне отдельных символов встречаются некоторые трудности. Например, всегда следует учитывать кодировку, используемую в конкретном документе. Текст может храниться в различных кодировках: ASCII, UTF-8, UTF-16 или Latin-1. Также необходимо принимать во внимание и другие факторы, например зависимость текста от регистра букв. Особые виды обработки могут потребоваться для знаков пунктуации и для чисел. Иногда приходится отдельно обрабатывать использование значков, отражающих эмоции (комбинации символов или специальные символы), гиперссылок, повторяющихся знаков пунктуации (... или ---), расширений файлов и имен пользователей, содержащих точки. Большинство этих задач решается с помощью предварительной обработки текста, которая будет обсуждаться в разделе «Подготовка данных» ниже.

Под делением текста на фрагменты или элементы обычно подразумевается представление текста в виде последовательности слов. В этом случае слова обозначаются термином «*лексический элемент*», «*лексема*», или просто «*токен*» (*token*), а процесс деления текста – «*токенизация*» (*tokenization*). Этот процесс не вызывает особых затруднений в языках, использующих пробельные символы для разделения слов, но в языках, подобных китайскому, это сделать гораздо труднее, поскольку иероглифы могут обозначать и слоги, и целые слова.

Слова и морфемы могут быть обозначены метками, идентифицирующими соответствующую часть речи. *Морфема* (*morpheme*) – это минимальная значимая часть слова (текста). Примерами морфем являются приставки, корни и суффиксы слов. При работе с отдельными

словами как с элементами текста часто приходится определять синонимы, аббревиатуры, акронимы и правильность написания.

Определение основы слова представляет собой еще одну задачу, требующую практического решения. Термин *стемминг* (*stemming*) обозначает процесс определения основы слова по его различным формам (причем основа не всегда совпадает с корнем слова). Например, для слов «замедлить», «замедленный», «замедляющий» основой является «замедл». Пример из английского языка: слова «walking», «walked», «walks» имеют основу «walk». Поисковые механизмы часто используют стемминг для улучшения качества ответов на запросы.

Со стеммингом тесно связана *лемматизация* (*lemmatization*), процесс определения начальной (основной) формы слова, обозначаемой термином «*лемма*» (*lemma*). Например, для слова «действующий» основой является «действ», но его лемма – «действовать» (в английском языке слово «operating» имеет основу «oper», но лемму «operate»). Лемматизация – это более сложный и тонкий процесс, чем стемминг. Для определения леммы используются словарь форм и морфологические методики. В некоторых случаях это позволяет выполнить более точный анализ текста.

Слова объединяются в словосочетания и предложения. Определение границ предложений тоже может быть связано с некоторыми трудностями, хотя на первый взгляд кажется, что достаточно лишь найти точку, обозначающую конец предложения. Но точки могут встречаться и внутри предложения, например после сокращенных слов г. или н. э., а также при использовании британского формата записи чисел 12.834.

Часто необходимо знать, какие слова в предложении являются существительными, а какие – глаголами. Иногда приходится определять отношения между словами. Например, установление *кореферентности* (*референциального тождества – coreference resolution*) определяет взаимоотношения между конкретными словами в одном или в нескольких предложениях. Рассмотрим следующие предложения:

«Город велик, но прекрасен. Он занимает всю долину».
(«The city is large but beautiful. It fills the entire valley».)

Здесь слово «он» («it») кореферентно (референциально тождественно) слову «город» («city»). Если слово может иметь несколько смысловых значений, для определения его смысла в данном конкретном случае может потребоваться выполнение операции *разрешения лексической многозначности* (*word sense disambiguation, WSD*). Иногда

это связано с определенными трудностями. Например, «Джон вернулся домой» («John went back home»).

Что именно здесь подразумевает слово «дом» («home»): строение, город, какое-либо место или нечто другое? В некоторых случаях смысл слова можно определить из контекста, в котором оно употребляется. Например, «Джон вернулся домой. Дом его находился в конце тупика» («John went back home. It was situated at the end of a culde-sac»).

Несмотря на все перечисленные трудности, технология обработки естественного языка в большинстве случаев способна достаточно уверенно справляться со своими задачами, поэтому весьма полезна во многих областях. Например, по твитам покупателей можно выполнить анализ эмоциональной окраски текста и получить возможность предложить недовольным клиентам варианты компенсации, в том числе какие-либо бесплатные товары. Медицинские документы легко поддаются аннотированию, что позволяет выделять наиболее важные части текста и увеличить производительность их обработки.

Аннотирование, или реферирование (summarization) – это процесс составления краткого описания различных фрагментов текста. Фрагменты могут состоять из нескольких предложений, абзацев, включать в себя целый документ или даже несколько документов. Задачей аннотирования (реферирования) может быть определение тех предложений, которые выражают основной смысл фрагмента, определение предпосылок, помогающих понять содержание фрагмента, или поиск заданных элементов во фрагментах. Зачастую для выполнения этих задач (и задачи аннотирования в целом) очень важен общий контекст обрабатываемого текста.

Обзор инструментальных средств обработки естественного языка

Существует много общедоступных инструментальных средств, поддерживающих технологию обработки естественного языка. Некоторые из них входят в комплект Java SE SDK, но их возможности ограничены достаточно узким кругом простейших задач. Другие библиотеки, например OpenNLP (Apache) и LingPipe, предоставляют более мощную и усовершенствованную поддержку методик обработки естественного языка.

Низкоуровневая поддержка в языке Java включает классы для работы со строками: `String`, `StringBuilder` и `StringBuffer`. Эти классы содержат методы, выполняющие операции поиска, сопоставления и замены текстовых фрагментов. *Регулярные выражения* (*regular expressions*) применяют специализированные методы определения со-

впадения подстрок. Java предлагает богатый выбор инструментальных средств для использования регулярных выражений.

Ранее уже отмечалось, что для разделения текста на отдельные элементы используются механизмы *токенизации*. Java поддерживает и этот вид обработки:

- метод `split()` в классе `String`;
- класс `StreamTokenizer`;
- класс `StringTokenizer`.

Кроме того, для языка Java написано множество библиотек/API для поддержки обработки естественного языка. Неполный список таких библиотек приводится в табл. 1.1. Большинство из них являются проектами с открытым исходным кодом, но существуют также и коммерческие API и библиотеки. Мы будем рассматривать API с открытыми исходными кодами.

Таблица 1.1. Список API на языке Java, поддерживающих обработку естественного языка

API	URL
Apertium	http://www.apertium.org/
General Architecture for Text Engineering	http://gate.ac.uk/
Learning Based Java	http://cogcomp.cs.illinois.edu/page/software_view/LBJ
LinguaStream	http://www.linguastream.org/
LingPipe	http://alias-i.com/lingpipe/
Mallet	http://mallet.cs.umass.edu/
MontyLingua	http://web.media.mit.edu/~hugo/montylingua/
Apache OpenNLP	http://opennlp.apach.org/
UIMA	http://uima.apache.org/
Stanford Parser	http://nlp.stanford.edu/software

Многие из упомянутых выше API обработки естественного языка позволяют объединять задачи в *конвейеры* (*pipeline*), своеобразные последовательности шагов (операций), для достижения определенных целей обработки текста. Примерами инструментов, поддерживающих конвейеры, являются GATE и Apache UIMA.

В следующих разделах мы подробнее познакомим вас с некоторыми NLP API, дадим краткий обзор их возможностей и для каждого API приведем список полезных ссылок.