ISSN 2079-3316 ПРОГРАММНЫЕ СИСТЕМЫ: ТЕОРИЯ И ПРИЛОЖЕНИЯ № 3(3), 2010, с. 3–12

УДК 519.767.6

Д. А. Кормалев

Настраиваемый подход к эффективному распознаванию текстовых ситуаций

Аннотация. В статье предложен механизм, обеспечивающий удобный способ специализированной аналитической обработки текстовой информации, не требующий от пользователя системы знания формальных языков описания правил и обеспечивающий высокую вычислительную эффективность. Подход предназначен для распознавания относительно простых контекстов, специфика которых неизвестна заранее, поэтому разработка полноценной системы правил распознавания текстовых ситуаций нецелесообразна. Высокая вычислительная эффективность обеспечивается за счет предобработки корпуса текстов и предварительной фильтрации.

Ключевые слова и фразы: обработка текстов на естественном языке, сопоставление образцу, контекстная фильтрация, конечные автоматы.

Введение

Один из важных инструментов автоматической обработки текста — средства и методы распознавания текстовых ситуаций. Распознавание текстовых ситуаций состоит в выделении фрагментов текста, описывающих объекты, и содержательных связей между этими фрагментами, основанных в той или иной мере на синтаксисе естественного языка. Можно рассматривать распознавание ситуаций как ориентированный на предметную область точный синтактико-семантический анализ, поэтому этот подход находит применение в широком спектре предметно-ориентированных технологий и приложений. Распознавание текстовых ситуаций может использоваться для извлечения информации [1, 2] (распознавание и выделение релевантной информации и представления ее в структурированной форме: поиск объектов, фактов, ситуаций), повышения точности информационного поиска, в составе вопросно-ответных систем или как компонент

Работа выполнена при поддержке РФФИ — проект 09-07-00407 «Абстрактные машины для обработки нелинейных данных».

предметно-ориентированного синтактико-семантического анализатора. Распознавание чаще всего опирается на сопоставление образцу, который задается при помощи правил на формальном языке (существуют и другие подходы к решению этой задачи [3, 4]).

Одна из проблем, возникающих при настройке средств распознавания ситуаций на предметную область — сложность написания и отладки контекстных правил. Для разработки системы точных правил, обеспечивающих высокую полноту распознавания, требуется высокая квалификация пользователя системы обработки текстов. В то же время существует ряд прикладных задач, для решения которых достаточно распознавания относительно простых контекстов, но попрежнему требуется хорошее знание языка описания правил. Кроме того, традиционный подход к обработке текстов с помощью правил подразумевает полный проход по корпусу документов, что нецелесообразно с точки зрения вычислительной эффективности. Решить эти проблемы позволяет подход, описанный в настоящей статье.

Дальнейшее изложение идет из предположения о том, что механизм шаблонов встроен в систему обработки текстовой информации, в которой

- присутствуют средства лингвистической обработки текстов (графематика, выделение границ предложений, нормализация словоформ);
- выполняется распознавание объектов предметной области с привязкой к тексту (извлечение информации);
- существует хранилище текстовых данных, в котором можно сохранять дополнительную информацию.

1. Назначение пользовательских шаблонов

Пользовательские шаблоны обработки представляют собой простой и в то же время довольно мощный механизм создания и выполнения специализированных сценариев обработки текстовой информации. Особенность таких сценариев заключается в том, что пользователь системы обработки текстовой информации получает возможность вмешаться в процесс обработки текстовой информации, оперативно выполнить дополнительную обработку по заранее неизвестным правилам, осуществить быструю проверку гипотез. Относительно простая структура шаблонов делает возможным их задание

с помощью специального программного средства (конструктора), не предполагающего знания каких-либо формальных языков.

Шаблоны позволяют решать ряд аналитических задач, возникающих при работе с корпусом текстов, например:

- выполнение сложных поисковых запросов;
- построение выборок документов из корпуса;
- распознавание текстовых ситуаций;
- построение простых объектов предметной области;
- рубрицирование документов.

Будем рассматривать два вида пользовательских шаблонов обработки текстовой информации:

- шаблоны, для которых важен факт успешного сопоставления хотя бы на одном из предложений документа (шаблоны фильтрации);
- шаблоны, которые обеспечивают относительно простую дополнительную разметку текстов и, возможно, построение объектов предметной области (шаблоны разметки).

Результатом применения шаблонов фильтрации является множество документов, шаблонов разметки — множество обобщенных термов (построенных в ходе применения). Эти два вида шаблонов обеспечивают решение задач, перечисленных выше (например, рубрицирование сводится к построению подмножеств документов из корпуса).

Шаблоны разметки являются более общими. Если для шаблонов фильтрации интересен только факт срабатывания шаблона на документе, то для шаблонов разметки кроме факта сопоставления важна дополнительная информация.

2. Структура шаблонов

Введем понятие обобщенного терма. *Обобщенный терм* — это текстовая единица, представляющая собой отдельное слово или фрагмент текста, соответствующий распознанному в тексте объекту предметной области. Обобщенные термы являются единицами, на уровне которых осуществляется сопоставление шаблонов. Для каждого обобщенного терма задана левая и правая граница, то есть можно говорить о перекрывающейся интервальной разметке документов анализируемого корпуса.

Введем определения взаимного расположения термов, подобные определениям для интервальной разметки. Пусть для обобщенного терма x известны левая и правая границы в тексте (l_x и r_x соответственно). Тогда обобщенный терм y следует за обобщенным термом x, если $r_x < l_y$ (обозначим этот факт $x \prec y$). Обобщенный терм y непосредственно следует за обобщенным термом x, если

$$x \prec y \wedge \nexists z : (x \prec z \wedge z \prec y).$$

Шаблон представляет собой обобщенный блок сопоставления, составленный при помощи операций из базовых блоков (тестов). Базовые блоки описывают обобщенные термы. Операции позволяют составлять обобщенные блоки из тестов. Для обеспечения основных аналитических потребностей требуются следующие виды тестов.

Буквальный: блок сопоставляется конкретному слову в анализируемой позиции (с точностью до словоизменения).

Альтернативный: блок сопоставляется слову из заданного синонимического множества. Этот вид блоков выделен в отдельную категорию для того, чтобы было удобно задавать синонимические группы при использовании тезаурусов (например WordNet [5]).

Свободный: блок успешно сопоставляется любому обобщенному терму в анализируемой позиции.

Описание объекта предметной области: наличие в анализируемой позиции обобщенного терма, соответствующего объекту предметной области и обладающего заданными характеристиками. В простом случае накладывается ограничение только на таксономический класс объекта.

Регулярное выражение: наличие в анализируемой позиции обобщенного терма, на которой успешно срабатывает приведенное регулярное выражение.

Каждый вид базовых блоков может быть снабжен признаком *отрицания*: инверсия признака успешного сопоставления базового блока, находящегося в области действия этой операции.

Обобщенные блоки формируются при помощи двух операций.

Следование: неявная операция, заключающаяся в последовательном сопоставлении двух (обобщенных) блоков. Блок AB успешно сопоставляется, если в некоторой позиции текста последовательно сопоставляются блоки A и B (сопоставление B начинается с термов, непосредственно следующими за последним термом, который был рассмотрен при сопоставлении A).

Квантификация: аналогично квантификаторам регулярных выражений. Запись $A\{\mathtt{m},\mathtt{n}\}$ означает явное ограничение на количество повторных сопоставлений блока снизу (m) и сверху (n). Если не задано значение m, оно полагается равным нулю; если не задано n, вместо него допускается произвольное число повторов (не менее m). Для удобства можно использовать общепринятые квантификаторы:

- `?` $\equiv$ `{0,1}`,
- `*` $\equiv$ `{0,}`,
- `+` $\equiv$ `{1,}`.

Блок Ax (здесь под x подразумевается операция квантификации) успешно сопоставляется, если из текущей позиции удается сопоставить блок A нужное количество раз (в зависимости от квантификатора). Квантификаторы могут применяться как к базовым, так и к обобщенным блокам.

Отметим важное отличие от аппарата регулярных выражений: из операций специально исключена операция альтернативы. Кроме того, в шаблоне должен быть хотя бы один базовый блок, не находящийся в области действия необязательного квантификатора, не обладающий признаком отрицания и не являющийся свободным. Эти упрощения не сильно сказываются на практической действенности шаблонов, но обеспечивают их эффективное применение, как будет показано ниже.

3. Применение шаблонов

Сопоставление шаблонов осуществляется в узком контексте (на практике в качестве такого контекста используется предложение текста на естественном языке). Перед сопоставлением происходит трансляция шаблонов в конечные преобразователи (КП) следующего вида:

$$\langle Q, i, F, T, \Sigma, \delta, \sigma \rangle ,$$

где Q — множество состояний; i — начальное состояние; F — множество конечных состояний; T — множество блоков; Σ — выходной алфавит (для фильтрующих шаблонов этот алфавит будет пустым); $\delta : Q \times T \to Q$ — функция переходов; $\sigma : Q \times T \to \Sigma$- функция выходов.

Шаблоны представляют собой разновидность средств распознавания текстовых ситуаций, используемых в системах извлечения информации [1]. Применение описанных шаблонов напрямую не будет отличаться по производительности от полного анализа текста документов с помощью традиционных средств распознавания текстовых

ситуаций [6]. Высокая вычислительная эффективность применения рассматриваемых в статье средств обеспечивается благодаря предварительной фильтрации документов и выделении их фрагментов для более детального анализа. Применение шаблонов включает в себя два этапа: выделение узких контекстов, в которых потенциально возможно сопоставление, и полная проверка сопоставления средствами КП. В некотором смысле первый этап похож на метод анализа потоков управления КП, применяемый в традиционном распознавании текстовых ситуаций [7]. Отличие состоит в том, что при анализе потоков управления необходим однократный полный проход по тексту документа (линейная сложность), а для предварительной фильтрации можно построить дополнительные структуры данных, обеспечивающие логарифмическую сложность для большинства пользовательских шаблонов.

Как отмечалось, сопоставление шаблонов происходит на уровне предложений анализируемого естественного языка, следовательно, необходимо хранить информацию о границах предложений для всех документов анализируемого корпуса. Для действенной предварительной фильтрации также необходима информация о содержимом документов: как минимум, нужна информация о границах обобщенных термов, канонические формы слов и информация о таксономических классах распознанных ранее текстовых объектов. Особенность рассматриваемых в статье шаблонов заключается в том, что в них будет большое количество буквальных или альтернативных блоков. Кроме того, структура шаблонов гарантирует, что такие блоки необходимы для успешного сопоставления шаблона (за исключением случая, когда они находятся в области действия необязательного квантификатора).

Один из вариантов предварительной фильтрации — создание полнотекстового индекса по словам документов с указанием привязки к предложениям. По мере обработки корпуса текстов сохраняется информация о границах предложений документов, строится справочник встреченных обобщенных термов (с нормализацией), происходит привязка термов к предложениям. Такой подход потребует хранения дополнительной информации пропорционально объему текстовой базы, но обеспечит быструю грубую фильтрацию предложений-кандидатов для полной обработки средствами конечных преобразователей.

Для более тонкой обработки сделаем еще один шаг и будем анализировать не только отдельные блоки (и обобщенные термы), но

и последовательные блоки (и непосредственно следующие обобщенные термы). В 1997 году Бродером и др. был предложен [8, 9] метод оценки сходства между документами, основанный на представлении документа в виде множества всевозможных последовательностей фиксированной длины k, состоящих из соседних слов. Такие последовательности были названы «шинглами».

Идею шинглов длины 2 (2-шинглов) можно использовать для эффективного решения задачи предварительной фильтрации. Будем строить индекс не по отдельным обобщенным термам, а по их последовательностям (привязка к предложениям сохраняется и в этом случае): для каждой пары непосредственно следующих термов делаем запись в индексе. Объем индекса при этом будет расти по-прежнему почти линейно, поскольку в документе, содержащем N слов, будет $N - 1$ шинглов. Незначительная нелинейность возникнет из-за того, что каждый предварительно распознанный объект обычно добавляет две дополнительных записи в индексе. Количеством дополнительных записей можно пренебречь, поскольку для практических приложений количество таких объектов будет много меньше количества слов документа, а случай примыкания распознанных объектов будет встречаться еще реже. Индекс 2-шинглов используется следующим образом. В конечном преобразователе, полученном из шаблона, выделяются обязательные последовательности длины 2. После этого осуществляется запрос к индексу для фильтрации множества предложений, содержащих 2-шинглы, соответствующие выявленным обязательным последовательностям. Такая фильтрация оказывается намного более точной, поскольку учитывается не только наличие термов, но и их непосредственное следование. Увеличение длины шинглов в индексе нецелесообразно, поскольку 2-шинглы уже обеспечивают достаточную фильтрацию, а цепочки из трех или более буквальных блоков в применимых на практике пользовательских шаблонах встречаются довольно редко.

Второй этап сопоставления шаблонов заключается в применении конечного преобразователя, полученного из шаблона, к структурам данных, соответствующим предложениям-кандидатам, которые были получены на первом этапе. Принципиально он ничем не отличается от исполнения распознающих правил языка, ориентированного на обработку интервальной разметки (например INEX PSL [10]).

Более того, возможна реализация, в которой пользовательские шаблоны транслируются в правила извлечения информации, и дальнейшее распознавание идет с применением соответствующих механизмов. При этом необходимо обеспечить также преобразование структур данных, описывающих термы, в структуры данных, обрабатываемые языком правил (например аннотации [11, 12]). Это возможно, поскольку язык шаблонов представляет собой узкий диалект языка правил, основанного на CPSL, а аннотации практически совпадают по устройству с описаниями обобщенных термов.

Результаты работы шаблонов передаются пользователю для верификации и возможного последующего сохранения. Повторная разметка или найденные документы может быть проигнорирована автоматически или с подтверждением со стороны пользователя.

Заключение

В статье предложен механизм, предоставляющий удобный для пользователя способ специализированной аналитической обработки текстовой информации. При этом не требуется знания формальных языков описания правил и обеспечивается высокая вычислительная эффективность.

Возможности пользовательских шаблонов не ограничиваются теми, что описаны в статье. Рассмотрим ряд перспективных направлений развития.

Естественным направлением развития предлагаемых идей будет создание сценариев обработки, состоящих из нескольких шаблонов. Шаблоны, входящие в сценарий, могут использовать результаты работы других шаблонов разметки, уточняя или дополняя их. Для обеспечения такой возможности результаты работы шаблонов разметки должны сохраняться в индекс. Сценарии также могут включать в себя сохранение результатов работы отдельных шаблонов с последующим применением к ним теоретико-множественных операций.

Внутренний механизм шаблонов может выступать нижним уровнем для полнотекстовых запросов (в том числе, задаваемых с применением специализированных поисковых языков).

Чтобы избежать необходимости задавать каждый раз синонимические группы явным образом, будет полезна библиотека синонимических групп, обеспечивающая их повторное использование в различных шаблонах.

Для известных заранее синонимических групп из справочников (в том числе иерархических) можно провести дополнительную предобработку текстов с построением индекса не только по конкретным терминам, но и группам в целом. Для этого необходимо сохранять в индексе информацию о принадлежности термов группам из справочников.

Предложенный подход не накладывает ограничений на дополнительную информацию, хранящуюся в индексе. Например, можно дополнительно сохранять морфологическую информацию о словах и расширенную предметную информацию о распознанных объектах. Способ применения шаблонов при этом не изменится, размер индекса будет по-прежнему пропорциональным объему корпуса, но появятся дополнительные возможности задания точных шаблонов.

Работа выполнена при поддержке РФФИ — проект 09-07-00407 «Абстрактные машины для обработки нелинейных данных»

Список литературы

[1] Appelt D. E., Israel D. J. *Introduction to Information Extraction. Tutorial* // Sixteenth Int. Joint Conf. on Artificial Intelligence IJCAI'99. — Stockholm, Sweden, 1999. ↑[], 3

[2] Ермаков А. Е. *Извлечение знаний из текста и их обработка: состояние и перспективы* // Информационные технологии, 2009, № 7. ↑[]

[3] Stevenson M., Greenwood M. A. *Comparing information extraction pattern models* // IEBeyondDoc '06: Proceedings of the Workshop on Information Extraction Beyond The Document. — Morristown, NJ, USA : Association for Computational Linguistics, 2006. ↑[]

[4] Etzioni O., Banko M., Soderland S., Weld D. S. *Open information extraction from the web* // Commun. ACM, 2008. **51**, no. 12. ↑[]

[5] WordNet: An Electronic Lexical Database / ред.Fellbaum Ch. : The MIT Press, 1998. ISBN 026206197X. ↑2

[6] Александровский Д. А., Кормалев Д. А., Кормалева М. С., Куршев Е. П., Сулейманова Е. А., Трофимов И. В. *Развитие средств аналитической обработки текста в системе ИСИДА-Т* // Тр. Десятой нац. конф. по искусственному интеллекту с междунар. участием КИИ-2006. — Москва : Физматлит, 2006. Т. **2**, с. 555–563. ↑3

[7] Кормалев Д. А. *Повышение производительности при распознавании текстовых ситуаций* // КИИ-2008. — Москва : ЛЕНАНД, 2008. Т. **2**, с. 192–200. ↑3

[8] Broder A., Glassman S., Manasse M., Zweig G. *Syntactic clustering of the Web* // 6th International World Wide Web Conference. — Santa Clara, California, United States : Elsevier Science Publishers Ltd., 1997, p. 1157–1166. ↑3

[9] Broder A. *On the Resemblance and Containment of Documents* // Proceedings of the Compression and Complexity of Sequences 1997 // SEQUENCES '97. — Washington, DC, USA : IEEE Computer Society, 1997. ↑3

[10] Кормалев Д. А., Куршев Е. П. *Развитие языка правил извлечения информации в системе ИСИДА-Т* // Тр. междунар. конф. «Программные системы: теория и приложения». — Москва : Физматлит, 2006. Т. **1**, с. 365–377. ↑3

[11] Grishman R. // TIPSTER Text Architecture Design. Version 3.1. New York, NYU, 1998. ↑3

[12] Кормалев Д. А. *Представление лингвистической и предметно-ориентированной информации о тексте при помощи аннотаций* // Четвертый российско-украинский научный семинар «Интеллектуальный анализ информации ИАИ-2004». — Киев : Просвіта, 2004, с. 120–128. ↑3

D. A. Kormalev. *An efficient customizeable technique for recognition of textual situations.*

Abstract. The paper proposes a technique for user-friendly, customizable analytical text processing. The technique is computationally efficient and does not require in-depth user knowledge of formal pattern specification languages. The approach is targeted at recognition of relatively simple contexts, which are unknown beforehand, when it is impractical to develop a full-scale rule-based system for recognition of textual situations. High efficiency is based on document collection preprocessing and preliminary filtering.

Key Words and Phrases: natural language processing, pattern matching, context filtering, finite automata.

Образец ссылки на статью:

Д. А. Кормалев. *Настраиваемый подход к эффективному распознаванию текстовых ситуаций* // Программные системы: теория и приложения : электрон. научн. журн. 2010. № 3(3), с. 3–12. URL: http://psta.psiras.ru/read/psta2010_3_3-12.pdf