УДК 004.89:004.03

А.В. Чеусов
Белорусский государственный университет, г. Минск, Беларусь
vle@gmx.net

Разработка лингвистических процессоров промышленной обработки текстовых документов

В настоящей работе изложена архитектура и некоторые принципы разработки промышленного лингвистического процессора, который лег в основу известного продукта инженерии знаний.

Введение

В эпоху интенсивной информатизации практически всех сфер человеческой деятельности задачи интеллектуализации информационного поиска и автоматизации инженерии знаний являются важнейшими задачами, требующими эффективного решения. Универсальным средством описания действительности и коммуникации с вычислительной системой является естественный язык (ЕЯ), поэтому указанные задачи фактически сводятся к задаче автоматизации его обработки на всех уровнях глубины ЕЯ, что требует разработки сложного программно-информационного комплекса. Такой комплекс, как сумму автоматизированных средств переработки текстовой информации на естественном языке, в том числе и не рассчитанных на работу с ЕЯ в полном объеме, принято называть лингвистическим процессором (ЛП) [1].

Автоматическая обработка ЕЯ или в более широком смысле – разработка ЕЯ-систем – не молодая область исследований, интерес к ней существовал уже в 50-х годах. К этому же периоду можно отнести и первые попытки решить ряд важнейших задач этой области, таких, например, как лексико-грамматическое аннотирование текста, генерация речи, машинный перевод текста с одного ЕЯ на другой и т.д. И многие из этих проблем все еще остаются открытыми. Это связано с тем, что в настоящее время к ЛП предъявляются совершенно новые требования как по качеству лингвистической обработки ЕЯ, так и, что особенно важно, по скорости работы всех модулей ЛП. Во-первых, речь идет о разработке систем, пригодных для достаточно точной обработки именно произвольных текстов ЕЯ, причем на всех уровнях его глубины. Во-вторых, объемы этих текстов чрезвычайно велики, нередко речь идет об объеме документов в десятки терабайт, не говоря уже о сети Интернет, где объемы значительно выше и информационное наполнение которой не только не прекращается, но неуклонно ускоряется из года в год. И в этих условиях ЛП на выполнение своей функциональности должен использовать минимальный временной ресурс, что требует серьезных исследований по оптимизации алгоритмического обеспечения ЛП.

Отметим, что глубокий анализ ЕЯ является очень сложной задачей, которую на практике, т.е. в промышленном варианте, крайне трудно решить в полном объеме, что особенно заметно на синтаксическом и тем более на семантическом уровне

обработки ЕЯ. В то же время очевидно, что требования к скорости и качеству обработки ЕЯ противоречивы: чем более глубокий анализ производит ЛП, тем сложнее становится система, тем сложнее становятся алгоритмы, лежащие в ее основе, и тем сложнее обеспечить необходимую производительность такой системы. Таким образом, перед разработчиками лингвистических процессоров стоит сложная задача нахождения компромисса, выбора наиболее приемлемого подхода в данных конкретных условиях. Тем не менее, как оказалось, эффективное решение задачи возможно. Разработанная нами технология обеспечивает высокие показатели промышленных ЛП, которые уже реализованы для целого ряда языков.

Архитектура лингвистического процессора и принципы его разработки

Классическая структурно-функциональная схема лингвистического процессора включает, прежде всего, следующие этапы обработки ЕЯ: лексический анализ, лексико-грамматический, синтаксический и семантический анализ.

Задача лексического анализа заключается в распознавании границ слов в тексте. Эта, на первый взгляд простая задача, в действительности ставит перед разработчиками ряд серьезных проблем. Они связаны, например, с наличием математических и химических формул в текстах патентов и научных статей, которые в лингвистическом анализе имеет смысл рассматривать как одно слово. Определенную трудность представляет собой наличие в текстах указателей на интернет-ресурсы, дат, адресов и т.п. несоблюдение единых правил написания знаков препинания и др. В разработанном нами ЛП для решения этих и многих других проблем была применена техника, описанная в [2] и основанная на использовании конечных автоматов (КА) [3], которые достаточно широко используются при обработке ЕЯ [4].

Задачей лексико-грамматического анализа (ЛГА) является разрешение лексико-грамматической омонимии у слов входного текста. В нашем случае эта задача решается для каждого предложения текста в отдельности. Существует огромное количество методов решения данной проблемы и среди них можно выделить два диаметрально противоположных подхода: вероятностный (probabilistic approach), основанный на использовании скрытых цепей Маркова [5], [6], и подход, основанный на правилах (rule-based approach) [5]. Для вероятностного подхода характерна более высокая скорость аннотирования, в том время как для подхода, основанного на правилах, – большая гибкость, управляемость алгоритма, и, по мнению некоторых исследователей, более высокое качество аннотирования [7]. В разработанной нами системе используется комбинированный подход, сочетающий в себе признаки и достоинства обоих указанных выше методов [8]. Так же, как и многие другие модули описываемого ЛП, модуль ЛГА основан на разработанном нами эффективном формализме расширенных регулярных выражений WRE [9], что придает ему значительную гибкость и простоту в использовании. Вероятностная составляющая алгоритма ЛГА использует цепи Маркова второго порядка и алгоритмы сглаживания Back-off и Witten-Bell [5].

При анализе произвольного текста важной проблемой является корректная обработка слов текста, отсутствующих в лексико-грамматическом словаре системы. Под обработкой в данном случае понимается, во-первых, распознавание лексико-грамматических свойств каждого такого слова, во-вторых, определенные морфологические преобразования, например получение начальной формы слова или построение его полной парадигмы по одной из его словоформ. Для решения первой задачи используется метод, предложенный в [10], что значительно улучшает

качество работы модуля лексико-грамматического анализа в целом. В процессе решения второй задачи был разработан алгоритм, описанный в [11], который успешно используется для целого ряда языков, поддерживаемых нашей системой.

Синтаксический анализ в описываемой системе производится в два этапа. На первом, более простом, этапе выделяются именные словосочетания, поскольку их выделение играет особенно важную роль как в самом анализе, так и в различных приложениях ЛП. На втором этапе синтаксического анализа выделяются определенные отношения между различными именными словосочетаниями и словами, не вошедшими ни в одно из них, например выделяются связи «подлежащее – сказуемое – дополнение», «глагол – обстоятельство» и т.п. Эта задача в нашем случае реализуется через построение дерева синтаксического вывода и так же, как и выделение именных групп, решается с использованием аппарата WRE. На этапе семантического анализа полученное дерево синтаксического вывода приводится к определенной унифицированной форме, далее происходит семантическая детализация выделенных связей, например выделение так называемых семантический падежей [12] и выделение новых связей, никак не выраженных синтаксической структурой предложения. Среди выделяемых связей в нашем случае наиболее важными являются отношения типа «субъект – акция – объект», «причина – следствие», «объект – параметр – процесс» и подобные, поскольку они являются определяющими с точки зрения понимания текстового документа и автоматического извлечения из него классических элементов знаний. Решение данной задачи основано на множестве паттернов, оперирующих элементами различных уровней глубины языка: лексическими единицами, лексико-грамматическими, семантическими и т.д. классами.

Также в описываемой нами системе реализован модуль разрешения местоименной анафоры [13]. Автоматическое установление анафорических связей необходимо для извлечения информации, которую невозможно получить на основе, например, только синтаксического анализа. Идентификация анафорически связанных элементов текста позволяет повысить полноту и точность систем информационного поиска, инженерии знаний, открывает новые возможности при решении других важных прикладных задач, таких, как машинный перевод, автоматическое аннотирование и реферирование текста, его синтез.

Важной составной частью работ по созданию промышленных ЛП является тестирование его отдельных модулей, а на конечной стадии работы самого ЛП в целом. С этой целью был создан аннотированный корпус текстов достаточно большого объема, отображающий «основное лингвистическое поведение» в большинстве предметных областей. Так, например, в состав корпуса включены тексты художественной и публицистической литературы, тексты американских, европейских и японских патентов, научные статьи, новости, отдельный корпус вопросительных предложений и многое другое. Аннотированный лексико-грамматическими тэгами корпус, включающий более миллиона слов для всех поддерживаемых системой языков, также произведено частичное синтаксическое и семантическое аннотирование, т.е. в нашем случае были выделены различные виды отношений между именными словосочетаниями. Такой корпус текстов, как составная часть базы знаний ЛП, использовался для оценки качества его работы на всех указанных этапах анализа текста и как основной источник знаний о самом ЕЯ, а также использовался для решения различных прикладных задач. Качество работы каждого модуля ЛП оценивается по общепринятым двум показателям: точность и полнота, – впервые примененным еще для оценки качества работы информационно-поисковых систем. Аннотирование осуществлялось экспертами с максимальной автоматизацией процесса.

Отметим также, что над реализацией промышленных ЛП работают, как правило, целые коллективы разработчиков, прежде всего лингвистов и программистов, что делает необходимым условием для успешной разработки полное разделение

программного кода и лингвистических ресурсов, таких, как словари, классификаторы, различные грамматики и т.п. Таким образом, появляется возможность полностью менять поведение ЛП, изменяя лингвистические ресурсы системы, не изменяя ее программный код. Это делает систему в целом максимально открытой и гибкой. Это качество становится особенно важным при разработке многоязычного ЛП и делает адаптацию уже существующей системы на обработку других языков предельно простой.

Кроме полного отделения программного кода от лингвистических ресурсов для нашей системы характерна модульность на всех уровнях и этапах обработки ЕЯ. В процессе разработки ЛП, мы пришли к пониманию того, что значительную роль в успехе нашей работы играет этап прототипирования, т.е. этап быстрой разработки прототипа нового модуля, пригодного для оценки правильности заложенных в него концепций, однако не обладающего пока требуемыми скоростными характеристиками и показателями точности. Необходимость такого этапа разработки продиктована несколькими основными причинами. Во-первых, разработка прототипов ведется, как правило, на скриптовых языках высокого уровня, таких, например, как ruby, pike, awk, shell с привлечением утилит, традиционных для операционной системы UNIX, таких, как m4, make, grep и другие, что позволяет реализовать прототип максимально быстро, завершающий же этап разработки модуля, пригодного для эксплуатации в промышленных условиях и требующего длительной работы по его оптимизации, начинается только после того, как разработанный прототип продемонстрировал ожидаемые от него характеристики и показатели. Во-вторых, модули, реализуемые, как правило, в виде набора исполняемых файлов и файлов лингвистических ресурсов, разрабатываются таким образом, чтобы из них можно было с помощью языка сценариев shell создать более сложные модули, включающие его в качестве составной части. Такая схема позволяет легко заменить любой модуль на любом уровне обработки ЕЯ на новый модуль без длительного этапа перепрограммирования системы. Таким образом, наша система в некотором смысле является АРМ-ом эксперта-лингвиста, позволяющим легко «конструировать» или «моделировать» новую систему на основе уже готовых блоков. Понятно, что для этого эксперт должен обладать элементарными навыками программирования и базовыми знаниями об операционной системе UNIX, поскольку система не предусматривает никаких визуальных средств.

Как было отмечено выше, лингвистические ресурсы системы, составляющие ее лингвистическую базу знаний (ЛБЗ), отделены от программного кода и, таким образом, определяют работу лингвистического процессора. В состав ЛБЗ входит около сотни различных типов лингвистических ресурсов: лексико-грамматический и семантический словари объемом порядка 250 тысяч слов, классификаторы, различные корпуса текстов общим объемом в несколько десятков миллионов слов, различного рода грамматики, несколько десятков тысяч правил анализа и преобразования текста на различных уровнях и т.п. Наполнение ЛБЗ производится экспертами-лингвистами вручную или полуавтоматически с использованием корпусов текстов и различного рода прототипов, упомянутых выше, при этом любое изменение ЛБЗ сопровождается тестированием ЛП с использованием как аннотированных, так и неаннотированных корпусов текстов.

С целью обеспечения необходимой производительности все лингвистические ресурсы перед их использованием в промышленном варианте ЛП компилируются в определенные структуры данных, эффективные для работы с ними, например, конечные автоматы, деревья, графы, хеш-таблицы и т.п. Учитывая довольно большое количество типов лингвистических ресурсов, одним из наиболее важных этапов работы по созданию промышленного ЛП является, по возможности, максимальная унификация различных по назначению инструментов, разработка языков представления правил, грамматик, их семантики и, в конечном итоге, их эффективная реализация. Среди многих разработанных с этой целью средств можно отметить, например,

разработанный нами аппарат расширенных регулярных выражений WRE, упомянутый выше, который без преувеличения можно назвать проблемно-ориентированным языком высокого уровня, на котором, так или иначе, основано большинство модулей как базового ЛП, так и его многочисленных приложений.

Заключение

Описанная в настоящей работе технологическая схема использовалась для разработки многоязычного промышленного ЛП, обрабатывающего текстовые документы на английском, японском, французском и немецком языках, который является базовым модулем известной системы автоматизации инженерии и управления знаниями Goldfire [14]. Что касается показателей качества работы ЛП, то, при явно более высоких временных показателях, показатели точности обработки текста также оказались выше наиболее известных систем подобного типа. Так, например, для английского языка точность ЛГА составила порядка 97 %, точность распознавания именных словосочетаний – 90 % (при идеальном лексико-грамматическом анализе – 98 %), распознавание отношений типа «субъект – акция – объект» – 78 %, при этом скорость обработки текста составила порядка 100 – 200 Кб/с. (в зависимости от входного языка). Без всякого сомнения, приведенные оценки качества и скорости работы лингвистического процессора могут быть улучшены, а изложенная схема и методы представляются нам весьма продуктивными.

Литература

1. Апресян Ю., Богуславский И., Иомдин Л. Лингвистический процессор для сложных информационных систем. – М.: Наука, 1992.
2. Чеусов А.В. Лексический анализ текста в промышленных системах его обработки // Вестник белорусского государственного университета: Серия 1. – 2005. – № 1. – С. 104-108.
3. Брауер В. Введение в теорию конечных автоматов. – М.: Радио и связь, 1987.
4. Kornai A. Extended Finite State Models Of Language. – United Kingdom: Cambridge University Press, 1999.
5. Jurafsky D., Martin J.H. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. – New Jersey: Prentice Hall PTR, 2000.
6. Manning C., Schütze H. Foundations of Statistical Natural Language Processing. – Cambridge (Massachusetts): The MIT Press, 1999.
7. Tapanainen P., Voutilainen A. Tagging accurately: don't guess if you know // Proceedings of the 4th Conference on Applied Natural Language Processing. – Morgan Kaufmann Publishers Inc., 1994. – P. 47-52.
8. Cheusov A. A part-of-speech tagging based on subtraction of word-based regular languages // Proceedings of the International Conference On Modelling And Simulation. – 2004. – P. 135-138.
9. Cheusov A. The word-based regular expressions in computational linguistics // Proceedings of 7th International Conference «Pattern Recognition and Information Processing» (PRIP-2003). – Minsk, 2003. – Vol. 1. – P. 208-212.
10. Mikheev A. Automatic rule induction for unknown-word guessing // Computational Linguistics. – 1997. – Vol. 23, № 3. – P. 405-423.
11. Чеусов А.В. Алгоритм автоматического порождения правил морфологического преобразования слов // Искусственный интеллект. – 2004. – № 4. – С. 673-678.
12. Fillmore C. The case for the case // Universals in Linguistic Theory. – 1968.
13. Поцепня В.Н. Контекстуальные, семантические и логические ограничения т преференции при разрешении местоименной анафоры // Искусственный интеллект. – 2005. – № 4. – С. 634-639.
14. Goldfire innovator // http://www.invention-machine.com/prodserv/GFIN.cfm.

A.V. Cheusov

Development of Industrial Natural Language Processing Systems

The present work describes architecture and some techniques of NLP systems development.

Статья поступила в редакцию 29.06.2006.