СИНТАКСИЧЕСКИЙ АНАЛИЗАТОР ЕСТЕСТВЕННОГО ЯЗЫКА DICTASCOPE SYNTAX

DICTASCOPE SYNTAX: THE NATURAL LANGUAGE SYNTAX PARSER

Скатов Д.С. (ds@dictum.ru), **Окатьев В.В.** (oka@dictum.ru),
Ратанова Т.Е. (ratanova@dictum.ru)

ООО «Диктум», Нижний Новгород, Россия

Ерехинская Т.Н. (erekhinskaya@gmail.com)

The University of Texas at Dallas

Ключевые слова: синтаксический анализ, соревнование синтаксических парсеров, деревья зависимостей, исправление опечаток,

В статье даны краткие сведения о моделях и методах, положенных в основу функционирования синтаксического анализатора естественного языка DictaScope Syntax. Дан обзор методов анализа на основе грамматик зависимостей, показаны преимущества и особенности применения этого подхода в представленном анализаторе. Статья подготовлена в рамках участия анализатора в соревновании синтаксических парсеров.

1. Введение

Синтаксический анализатор естественного языка DictaScope Syntax разрабатывается специалистами компании «Диктум» с 2002-го года. За этот период был пройден путь от простых эвристического подхода, основанного на формализме деревьев зависимостей и обладающего кубической сложностью анализа от длины входа. К достоинствам подхода можно отнести следующие его свойства:

- Разрешение морфологической омонимии путём прямого учёта в алгоритме анализа [5];
- Учёт информации о пунктуации непосредственно в процессе разбора [7, 6];
- Учёт опечаток и выбор верного исправления в процессе анализа [8];
- Возможность получения частичного разбора в случае ошибок в тексте;
- Учёт в процессе анализа как лингвистической составляющей (возможность и невозможность существования определённых конструкций), так и статистической (оценивание вероятностей конструкций для построения решения с максимальным правдоподобием).

Т.о., подход не является однополярным, а в определённой степени сочетает современные достижения в области парсинга естественного языка, занимая промежуточное место в пространстве методов на основе грамматик зависимостей и непосредственно составляющих, использования явно заданной лингвистической информации и статистического оценивания вариантов решения.

Разработаны реализации парсера DictaScope Syntax для русского и английского языков, имеются экспериментальные версии для немецкого и испанского. Парсер задействован в коммерческой системе мониторинга мнений в сети Интернет.

В данной статье даются базовые сведения о принципах работы DictaScope Syntax.

Представленный синтаксический анализатор принимал участие в соревновании синтаксических парсеров, организованного в рамках международной конференции «Диалог 2012».

2. Обзор анализа на основе деревьев зависимостей

Анализ на основе деревьев зависимостей (dependency parsing). Интенсивное развитие компьютерных технологий и формальных языков для их поддержки в 60-80 годах 20 в., а также превалирующие в то время логико-декларативные подходы к построению интеллектуальных систем, привели к созданию большого числа методов анализа естественных языков на основе формальных грамматик — явно заданных правил синтеза текстов. Сегодня существуют state-of-art парсеры на основе грамматик непосредственно составляющих для языков с жёстким порядком слов, таких как английский. Для языков с мягким порядком, напр. славянских, изолированное применение этого подхода в настоящее время представляется затруднительным в силу громоздкости результирующих правил и, как следствие, высокой трудоёмкости разработки и поддержки.

Формализм грамматик зависимостей до 90-х годов считался сугубо теоретическим. Он имеет древние корни — первые упоминания можно встретить уже в трудах древнеиндийского лингвиста Панини (около V века до н. э.), развитие получено в работах средневековых европейских лингвистов. Современные работы по этим грамматикам восходят к французскому лингвисту Тернье (50-е годы 20 в.) [4].

Можно дать следующую интерпретацию анализу на основе грамматик зависимостей: если в качестве вершин взять слова предложения и провести между ними допустимые связи, соответствующим образом взвешенные, то получение верного разбора можно попытаться свести к построению минимального остовного дерева (minimal spanning tree, MST) в полученном графе. С учётом этой аналогии, и в связи с трендом на развитие численного анализа данных, наметившегося в конце 80-х — начале 90-х годов, в грамматиках зависимостей увидели платформу для построения новых алгоритмов синтаксического анализа.

Развитие методов информационного поиска (information retrieval), обусловленное появлением сети Интернет и последующим экспоненциальным ростом объёма текстовых данных, показало, что представление результатов разбора деревьями зависимостей более удобно для их последующей обработки в приложениях. Это обусловлено тем, что структура зависимостей прозрачным образом отражает информацию о предикатах и их аргументах [4]. Ранние исследования (Eisner [2], McDonald, Pereira [3]), начиная с первой половины 90-х годов, показали также более высокую степень применимости этого метода для языков с мягким порядком слов, в первую очередь славянских и восточноевропейских.

Основные методы анализа на основе деревьев зависимостей. Сегодня в западной литературе доминирующей парадигмой в анализе на основе зависимостей является разбор, управляемый данными (data-driven parsing). Это обусловлено доступностью выверенных корпусов деревьев синтаксического разбора как для английского и китайского, так и для ряда европейких языков, напр., чешского [4]. Существует три основных подхода: (1) на основе машин состояний (transition-based parsing, Nivre), (2) взвешенных графов (graph-based parsing, Eisner, McDonald, Pereira), (3) грамматик ограничений [4]. В (1) для анализа используется стековый автомат с двумя множествами слов и множеством результирующих связей, связи образуются в результате операций сдвига и свёртки над парами элементов из каждого множества слов. Выбор операции и операндов на основе текущего состояния осуществляется по правилам, полученным по размеченному корпусу деревьев в результате процедуры обучения. Результирующим правилам, по мнению авторов, трудно дать содержательную интерпретацию и подвергнуть ручной корректировке. В (3) описываются ограничения, свойственные определённым грамматическим категориям при наличии связи, а результирующая система ограничений обычно сводится к задаче целочисленного линейного программирования (вообще говоря, NP-полной), которая далее решается приближённо.

Разбор на основе взвешенных графов. Подход (2) выбран за основу в парсере DictaScope Syntax. В нём потенциальным связям назначаются веса, и далее в графе извлекается остовное дерево с заданными свойствам и наименьшего веса.

Идея получения непроективного дерева разбора на основе MST-алгоритма со сложностью $O\ n^2$ привлекательна, но реализация имеет трудности, связанные с учётом дополнительных свойств результирующего дерева. Напр., показано [1], что учёт арности вершины в MST-алгоритме разбора — NP-полная задача, это же справедливо для учёта свойств её окрестности (т.н. горизонтальная и вертикальная марковизация). Этого недостатка лишён алгоритм Эйснера [2] получения проективного дерева, который по сути является адаптацией метода Кока-Янгера-Касами для взвешенных графов. Он имеет сложность $O\ n^3$, но лишён означенных выше проблем с учётом структуры деревьев. Метод может быть расширен для получения k лучших деревьев за счёт замедления на величину $O\ \log k$ [3].

Взвешивание рёбер обычно осуществляется следующим образом. Для каждого ребра вводятся числовые признаки $f = f_1,\ldots,f_n$, вычисляемые на основе свойств отдельных вершин, пар вершин, содержимого фрагмента между парой вершин, их окрестностей и метрических свойств связи. Далее вес связи полагается равным $\mathbf{w}\cdot\mathbf{f}$, где $\mathbf{w}$ — это числовой вектор, определяемый методами машинного обучения по размеченному корпусу так, чтобы наибольшее число минимальных деревьев было верными деревьями разбора. Уровень качества анализа по данным зарубежных исследователей — порядка 90% верных непомеченных связей (UAS) для английского, 80% для чешского [3].

Метод вычисления взвешивающего вектора путём обучения на корпусе привлекателен относительной простотой, однако пока слабо применим в чистом виде для многих национальных языков, в т.ч. для русского, в силу отсутствия широко доступных и статистически достаточных корпусов. Авторы предлагают решение этих проблем в виде т.н. синтаксических правил.

3. Расстановка и взвешивание зависимостей

В реализации парсера DictaScope Syntax авторами предложен механизм решающих правил, назначающих веса потенциальным связям. Они содержат описание свойств упорядоченной пары слов, к которой применяется правило, в виде логической формулы, возможную характеристику их окрестности и результирующие значения коэффициентов, на основе которых вычисляется вес. Для упрощения описаний используется наследование.

Продемонстрируем использование языка описания правил на примере.

Пусть требуется описать связь согласования. Базовое правило имеет вид:

(1)
```
Coordination {
Criterion:
 CaseEqual() & NumberGenderCoord() & AccAnimAgree();
Character:
 type = Agreement;
 isol = 0; ord = 0;
 num = 0; gender = 0; ...
}
```

Правило сработает, если выполнятся условия, записанные в секции `Criterion`.

На основе базового правила определим согласование существительного и прилагательного:

(2)
```
CoordNounFullAdj:_Coordination
{
 Key: (Noun, FullAdj)
 Character:
 ord = !(FatherFirst() & IsMarkBetween() | ChildFirst());
 isol = FatherFirst() & IsMarkBetween();
 num = !IsNumEqual();
 gender = !IsGenderEqual(); ...
}
```

По сути, правило `Coordination` представляет собой некоторый составной признак, заданный в секции `Criterion`. Отдельные признаки из `Character:ord` (обратный порядок слов), `isol` (способность связи образовывать обособление), `num` (рассогласование по числу), `gender` (рассогласование по роду) также являются составными, образованными комбинированием более простых признаков, обычно свойственным традиционным зарубежным реализациям парсеров зависимостей.

Такое избирательное комбинирование признаков, основанное на лингвистическом опыте, эквивалентно нелинейному преобразованию пространства простых признаков (ср. нелинейные ядра в методе SVM) и потому на практике способно к большей силе дискриминации верных решений в пространстве деревьев разбора.

Для русского языка база синтаксических правил сейчас насчитывает порядка 200 записей.

Вес, назначаемый ребру, имеет вид линейной комбинации взвешенных значений секции `Character`, определённых во время срабатывания правила. Т.о., весовые коэффициенты этой комбинации допускают оценку по корпусу всеми традиционными для dependency parsing методами.

В парсере DictaScope Syntax сочинительная связь между двумя однородными членами моделируется двумя подчинительными связями от общего предка к каждому из членов. Связи с придаточными предложениями имеют более сложную природу, их вычисление определяется правилами другого вида, включающими специальную систему штрафов. Алгоритм Эйснера был модифицирован для учёта пунктуации во время разбора [6, 7], и это на практике явилось более продуктивным путём учёта пунктуации в сравнении с изолированной оценкой присутствия знаков препинания в весах рёбер.

После расстановки связей на основе правил часть из них вычёркивается. Это существенно, например, для обработки придаточных предложений. Вычёркивание производится на основе локальных эвристик и снижает нагрузку с этапа собственно разбора.

4. Разрешение омонимии

Общая формулировка задачи синтаксического анализа. Представленная далее теоретико-графовая постановка задачи синтаксического анализа дана к.ф.-м.н. Окатьевым В.В [5].

Информация о синтаксических связях представляется в виде графа $G = <V, E>$, где V — множество вершин-омоформ, разбитое на N классов:

$$V_i : V_i \neq \varnothing, V_i \cap V_j = \varnothing, i \neq j, \bigcup_{i=1}^{N} V_i = V .$$

Класс V_i соответствует слову W_i в исходном предложении и содержит все его омоформы. В конце предложения приписан фиктивный корень, представляющий отдельный класс с единственной омоформой. E — множество связей между вершинами, строится в соответствии с правилами примыкания, согласования и управления в русском языке, а также на основании некоторых аналитических методов. В общем случае граф G деревом не является, т.к. в силу морфологической и синтаксической омонимии он будет содержать как истинные (образующие искомое дерево синтаксического разбора), так и ложные связи.

Задача анализа сводится к построению дерева $T = <V^*, E^*>$, $V^* \subseteq V$, $E^* \subseteq E$, причем из каждого класса в дерево входит строго одна омоформа:

$$\forall\, i \in \{1, \ldots, N\} \ \left| V_i \bigcap V^* \right| = 1.$$

Алгоритм Эйснера для учёта этой постановки был соответствующим образом модифицирован.

Учёт опечаток. Понятие омоформы в данной формулировке задачи можно трактовать более широко: это может быть любой вариант реализации слова (лексический, семантический и т.д.) в

данном предложении. Так, если слово отсутствует в словарях, для него можно построить гипотетические омоформы с помощью модуля исправления опечаток. Далее, с учётом весов вариантов исправления слова, эти омоформы будут участвовать в процессе синтаксического анализа, в результате будет выбираться омоформа, соответствующая наиболее вероятному исправлению.

Такой подход позволяет исправлять нетривиальные ситуации типа окончаний *«ться»-«тся»* в глаголах, а также принимать решение об отказе от исправления (напр., если была попытка исправить неизвестное имя собственное). Подробные сведения о реализации учёта опечаток в DictaScope Syntax даны в работе [8].

5. Проведенное тестирование

При тестировании парсеров актуальным является вопрос определения «канонической» структуры зависимостей, а также выделения составных лексем. Так в рамках конкурса был означен ряд случаев, в которых любое направление связей считалось допустимым. Таким образом, основная задача адаптации парсера к условиям тестирования заключалась в настройке результатов разбиения предложения на лексемы согласно разметке корпуса.

В систему DictaScope Syntax включен модуль выделения именованных сущностей, который определяет несколько классов объектов, рассматриваемых с точки зрения парсера как единая лексема. Среди них:

1. Названия организаций: *ЗАО «Красный мак»*;
2. Даты: *29 февраля 2012 года;*
3. Денежные единицы: *20$;*
4. Составные наречия: *без умолку, без устали;*
5. Составные предлоги: *в качестве, во время, в отношении, в связи с, несмотря на;*
6. Составные сочинительные и подчинительные союзы: *а также, в противном случае, как будто, коль скоро.*

Согласно такой классификации, одной лексеме может принадлежать довольно длинная цепочка слов, в то время как в корпусе, напротив, был принят принцип максимального разбиения лексем. Поэтому при формировании результирующей выдачи полученные в ходе анализа составные лексемы делились согласно разметке данной в корпусе, при этом дополнительного анализа составных лексем не проводилось – первое слово всегда считалось главным, а остальные с пометкой lexmod цеплялись к нему.

Был зафиксирован ряд предложений, содержащих формулы или другие специальные конструкции. Однако, поскольку система DictaScope Syntax ориентирована на обработку текстов на естественном языке, никакой дополнительной поддержки формул реализовано не было. Также, ввиду того, что тестирование проводилось на новостных текстах, модуль исправления опечаток был отключен.

6. Заключение

В статье изложены краткие сведения о синтаксическом анализаторе естественного языка DictaScope Syntax. Описано современное состояние научной области, связанной с деревьями зависимостей, и как её результаты расширены и реализованы в DictaScope Syntax. Показаны преимущества парсера, функционирующего на основе формализма деревьев зависимостей. Дана информация о возможностях, связанных с учётом омонимии в теоретико-графовой постановке задачи синтаксического анализа.

В настоящий момент представленный анализатор используется в коммерческой системе управления репутацией. Особенности применения парсера в этом контексте определяются спецификой

обрабатываемого материала — это в основном потребительские отзывы о товарах, услугах и брендах. Такие тексты насыщены опечатками и синтаксическими ошибками, иногда затруднительно определить даже границы предложений из-за отсутствия точек. На этом материале производительность DictaScope Syntax составляет порядка 70-80% верных связей. Обычно для оценки объектов и их характеристик достаточно хотя бы неполного разбора предложения, поэтому алгоритм восходящего разбора на основе деревьев зависимостей оказывается предпочтительным. В силу особенности текстов учёт опечаток в синтаксическом анализе также является критичным.

7. Список литературы

1. Juravsky D., Martin J. Speech and Language Processing: An introduction to natural language processing, computational linguistics, and speech recognition. 2007.
2. Eisner J. Three new probabilistic models for dependency parsing: An exploration // In: Proceedings of the 16th International Conference on Computational Linguistics (COLING), 1998, pp 340–345.
3. McDonald R., Pereira F. Online Learning of Approximate Dependency Parsing Algorithms // 11th Conference of the European Chapter of the Association for Computational Linguistics: EACL 2006, pp. 81-88.
4. Kübler S., McDonald R., Nivre J. Dependency Parsing // Synthesis Lectures on Human Language Technologies, Morgan & Claypool Publishers, 2009.
5. Окатьев В.В., Гергель В.П., Алексеев В.Е., Таланов В.А., Баркалов К.А., Скатов Д.С., Ерехинская Т.Н., Котов А.Е., Титова А.С. Отчет о выполнении НИОКР по теме: «Разработка пилотной версии системы синтаксического анализа русского языка» // М.: ВНИТЦ, 2008. Инвентарный номер ВНТИЦ 02200803750.
6. Окатьев В.В., Ерехинская Т.Н., Скатов Д.С. Модели и методы учета пунктуации при синтаксическом анализе предложения русского языка // Труды Международ-ной конференции «Диалог’2009». – М.: Наука, 2009.
7. Окатьев В.В., Ерехинская Т.Н., Ратанова Т.Е. Тайные знаки пунктуации // Труды Международной конференции «Диалог’2010». – М.: Наука, 2010.
8. Ерехинская Т.Н., Титова А.С., Окатьев В.В. Синтаксический анализ текста с орфографическими ошибками в системе Dictascope Syntax // Труды Международной конференции «Диалог’2011». – М.: Наука, 2011.