Сравнительный анализ статистических алгоритмов синтаксического анализа на основе деревьев зависимостей

A comparative analysis of machine learning dependency tree-based parsing algorithms

Казенников А. О. (kzn@iitp.ru)

ИППИ им. А. А. Харкевича РАН

В работе сравниваются различные подходы к построению синтаксической структуры на основе деревьев зависимостей. Рассматриваются два статистических подхода построения деревьев зависимостей: как задача минимальных остовных деревьев, и как задача разбора с автоматом с очередью. Оба подхода в качестве базового алгоритма используют SVM. Кроме того, производится сравнение эффективности этих подходов относительно системы на основе правил ЭТАП-3.

1. Введение

Традиционно сложная задача обработки текста на естественном языке разбивается на несколько уровней. Обычно, анализ на каждом уровне производится независимо от остальных. В частности, такими отдельными задачами являются морфологический анализ, снятие частеречной омонимии, построение синтаксической и семантической структуры. Поэтому, естественно предполагать, что при таком подходе в качестве исходных данных для задачи синтаксического анализа предстают слова с однозначно определенной частью речи.

Однако это не единственно возможный подход. Например, система ЭТАП-3 [1] проектировалась как лингвистический процессор. Его задачей является анализ текста, начиная с самого первого уровня. При разработке системы ЭТАП-3 было принято решение о том, что омонимия может быть разрешена в ходе синтаксического анализа.

В настоящей работе синтаксический анализ рассматривается как задача построения дерева зависимостей предложения при условии снятой частеречной омонимии.

2. Постановка задачи

Формально задача синтаксического анализа формулируется следующим образом. Дано предложение $S = \{w_i\}, i \in \{1 \dots n\}$, где w_i — i-слово предложения, n — число слов в предложении. В начало предложения добавляется фиктивное слово w_0, которое обозначает вершину синтаксической структуры предложения. Необходимо построить ориентированное дерево $G = (V, A)$, где V — вершины (слова), A — дуги (синтаксические связи). В дереве должна быть одна вершина и не должно быть циклов. Поскольку основным элементом анализа является связь, то необходимо определить несколько ее характеристик:

- хозяин связи — слово, из которого связь выходит,
- слуга — слово, в которое связь приходит,
- левое и правое слова связи —слова связи относительно их порядка в предложении,
- направление связи — в какую сторону (влево или вправо) от хозяина идет связь,
- имя связи — имя синтаксического отношения, связывающего слова между собой.

Эти характеристики являются «внутренними» — они относятся только к связи. Однако есть и очень важная «внешняя» характеристика связи, которая относится к структуре в целом и существенно влияет на алгоритмы синтаксического анализа: это проективность.

Неформально проективность связи можно определить графически — если построить дерево синтаксической структуры, то проективные связи между собой не пересекаются. Если какие-то две связи пересекаются между собой, то одна из них непроективна (см. рис 1). Более формально свойство проективности можно определить следующим образом — из хозяина связи доступны (по связям) все слова, которые эта связь перекрывает.

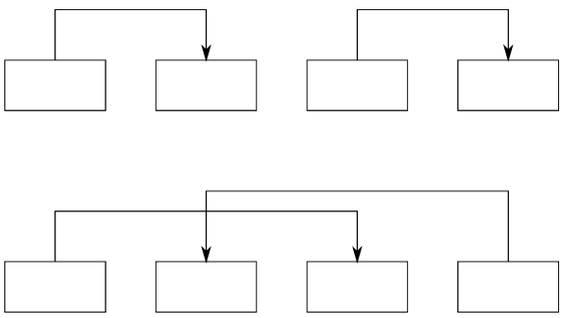

Рис. 1. Проективные и непроективные связи

3. Обзор существующих подходов

В работе рассматривалось несколько подходов к задаче синтаксического анализа:

- Облегченная система ЭТАП,
- Алгоритмы на основе максимальных остовных деревьев,
- Алгоритмы на основе системы переходов.

Система ЭТАП-3 является системой на основе правил. Она в значительной степени отталкивается от лингвистической теории «Смысл ↔ Текст» [2]. Правила для системы пишутся экспертами-лингвистами. На первом этапе синтаксического анализа строится матрица потенциальных связей в предложении, затем из этих связей формируется дерево. Часто существует возможность построения нескольких вариантов структуры для одного предложения. В таком случае по умолчанию выбирается первая построенная структура. Упрощенно говоря, задачей лингвиста является составление таких правил, при которых первое построенное дерево было бы оптимальным с лингвистической точки зрения. Таким образом, лингвистическая модель задается в явном виде с помощью правил. Основным недостатком такого подхода является необходимость больших трудозатрат для построения качественной системы (система ЭТАП-3 разрабатывается более 20 лет).

Принципиально другим способом представления модели языка является неявное представление в виде большого размеченного корпуса. Для практического применения такой модели используется подход на основе машинного обучения [3,4]. Тогда структура строится на основе закономерностей, выведенных алгоритмом из корпуса. Существенным недостатком этого подхода является сложность лингвистической интерпретации полученной модели, а так же необходимость в достаточно большом корпусе.

Важной особенностью систем на основе машинного обучения является полная зависимость от качества решения поставленной задачи машинного обучения. Т. е, если задача машинного обучения решена плохо, то соответственно будут плохими и результаты работы такой системы, независимо от используемого подхода.

В настоящем разделе представлен обзор подходов по их схеме работы, а в следующем разделе представлены подробная постановка и решение задачи машинного обучения.

В работах [3,4] был проведен сравнительный анализ статистических алгоритмов и было показано, что наиболее эффективными подходами является представление синтаксического анализа как задачи выделения максимального остовного дерева, а так же представление его как задачи поиска оптимальной последовательности действий.

Подход на основе максимальных остовных деревьев [3] рассматривает задачу синтаксического анализа как задачу нахождения максимального остовного дерева(MST) на графе возможных связей. Предполагается существование функции оценки связи $s(i,j) = \mathbf{w} \cdot \mathbf{f}(i,j)$. где $\mathbf{f}(i,j)$ — признаки, на основе которых принимается решение о проведении связи, w — модель, полученная с помощью машинного обучения.

Алгоритм выбирает такое дерево, сумма оценок связей которого будет максимальна:

$$s(\mathbf{x},\mathbf{y}) = \sum_{(i,j)\in\mathbf{y}} s(i,j)$$

$$\max_y s(\mathbf{x},\mathbf{y})$$

Задача машинного обучения заключается в получении такой функции оценки связи, которая бы позволяла построить правильную структуру, для наибольшего числа предложений из корпуса

Алгоритм построения дерева на основе функции оценки сильно зависит от необходимости построения непроективных связей. В классическом случае, когда *f(i, j)* зависит только от характеристик оцениваемой связи, алгоритм с возможностью построения непроективных связей проще, чем тот, который строит только проективные связи. Однако допущение непроективных связей сильно ограничивает возможность использования дополнительных параметров связи.

Другой подход — подход на основе системы переходов(TS) [4]. Парсер на основе такого подхода состоит из 3 компонентов:

1. Конфигурации — состояния процесса разбора в каждый конкретный момент.
2. Действия, изменяющие конфигурацию.
3. Начальное и конечное состояние конфигурации.

Задача синтаксического анализа сводится к поиску цепочки действий, которая бы переводила начальную конфигурацию в конечную. Для выбора каждого следующего действия используется оракул. Его задача — на основе текущей конфигурации выбрать следующее действие. Задача машинного обучения состоит в моделировании оракула. Параметры парсера очень сильно зависят от выбора систе-

мы действий и выбора параметров конфигурации, на основе которых принимается решение о следующем действии.

4. Обзор реализации подходов

Хотя система ЭТАП-3 разрабатывалась как система машинного перевода, ее архитектура позволяет работать в качестве синтаксического анализатора. В таком случае обработку входного предложения можно с некоторой степенью условности разделить на два этапа: морфологический и синтаксический. Важной особенностью системы ЭТАП-3 является взаимосвязь между модулями, и на вход синтаксического модуля подаются слова с неразрешенной частеречной омонимией. Поэтому, для сравнения системы ЭТАП с другими подходами ее надо уравнять в возможностях с другими системами. В проведенных экспериментах морфологический компонент не участвовал, а входными данными являлись слова со снятой частеречной омонимией.

Для подхода на основе минимальных остовных деревьев необходимо сформулировать принцип построения функции оценки связи. Фактически, задача построения такой функции есть задача ранжирования — необходимо, чтобы эталонная связь (на этапе обучения) получала большую оценку, чем остальные потенциальные связи:

$$s(i,j) > s(k,j), \forall k \neq i$$

$$s(i,j) > s(i,m), \forall m \neq j$$

Для применения такого алгоритма необходимо определить правила для построения рангов. Для деревьев зависимостей их можно определить на основе следующих фактов:

- Для каждого слова правильна только одна входящая связь
- Для данного слова потенциальными хозяевами могут могут быть все остальные слова предложения и вершина.

Другим важным вопросом является выбор модели признаков. Модель признаков определяет преобразование информации о связи и ее участниках в числовой вектор — вектор признаков. Классический вариант алгоритмов максимальный остовных деревьев предполагает независимость ребер графа. Это означает, что каждая синтаксическая связь проводится независимо от уже существующих.

Для подхода на основе системы переходов необходимо определить структуру конфигурации и набор возможных действий над конфигурацией. В работе [4] представлены несколько конфигураций и систем действий на их основе, однако наиболее интересной является конфигурация на основе списков, поскольку с помощью нее возможно построение непроективных структур (остальные системы переходов могут построить только строго проективные структуры).

Конфигурация этой системы состоит из трех списков. Под списком подразумевается линейная структура данных, обладающая вершиной, обозначаемой как head|list, и определенной операцией склейки l_1+l_2. Эти списки можно интерпретировать следующим образом. В списке b хранятся необработанные слова предложения, а вершина списка является потенциальным правым словом связи. Этот список упорядочен в соответствии с порядком следования слов в предложении. В списке l_1 хранятся возможные левые слова связи. Список l_1 расположен в обратном порядке относительно порядка слов в предложении. Список l_2 формируется из слов в промежутке между l_1 и b, порядок слов в нем так же противоположен порядку слов в предложении. Построение связи в какой-либо конфигурации возможно только между вершиной b и вершиной l_1.

При инициализации конфигурации в l_1 помещается фиктивное слово вершины предложения, а в b — слова предложения, l_2 — пуст. Признаком конечного состояния системы(конца разбора) является опустошение списка b.

Система состоит из четырех действий:

1. Left-Arc — проведение левой связи

   $(\lambda_1|i, \lambda_2, j|\beta) \Rightarrow (\lambda_1, i|\lambda_2, j|\beta)$

   связать (j,i)

2. Right-Arc — проведение правой связи

   $(\lambda_1|i, \lambda_2, j|\beta) \Rightarrow (\lambda_1, i|\lambda_2, j|\beta)$

   связать (i,j)

3. No-Arc — пропуск текущего левого слова

   $(\lambda_1|i, \lambda_2, j|\beta) \Rightarrow (\lambda_1, i|\lambda_2, j|\beta)$

4. Shift — пропуск текущего правого слова

   $(\lambda_1, \lambda_2, i|\beta) \Rightarrow (\lambda_1 + \lambda_2|i, [], \beta)$

В работе [4] показано, что сложность разбора на основе такой системы действий равна $O(n^2)$. Можно, однако, показать, что в среднем требуется количество действий, линейно зависящее от числа слов в предложении и средней длины связи. Из определений действий следует, что только операция Shift изменяет список b. Следовательно, для разбора предложения потребуется n действий Shift. В предложении всегда n связей(у каждого слова должен быть хозяин), следовательно, для разбора необходимо n действий Left-Arc или Right-Arc. После операции Shift список l_2 пуст, а потенциальная длина связи равна единице. Каждое действие Left-Arc, Right-Arc

или No-Arc увеличивает длину потенциальной связи на единицу, следовательно, максимальное число действий No-Arc пропорционально средней длине связи предложения. Таким образом, сложность алгоритма такой системы переходов O(n+k*n), где k — средняя длина связи в предложении.

Для создания модели оракула необходим алгоритм преобразования эталонной структуры в последовательность действий. Его можно построить на основе аналогичных соображений. Для преобразования эталонной структуры необходимо:

1. Отсортировать эталонные связи:
   1. По правому слову.
   2. По длине связи.
2. Два индекса: индекс вершины b, индекс вершины l_1.
3. Для каждой эталонной связи:
   1. Пока индекс вершины b не равен индексу правого слова, выполнять Shift.
   2. Пока индекс вершины l_1 не равен индексу левого слова, выполнять No-Arc.
   3. Провести эталонную связь с помощью действий Right-Arc, и Left-Arc
4. Добавить n действий Shift, где n — число оставшихся слов в b.

Обучение производится на основе пар текущая конфигурация → действие. Такая задача является задачей разбиения на несколько классов. В качестве алгоритма обучения используется алгоритм на базе SVM[6].

Одним из существенных отличий описываемой экспериментальной системы от представленной в [4] является модель признаков, извлекаемая из конфигурации. В качестве базовых признаков используются:

1. b[0] — потенциальный участник связи
2. b[1,2,3] — контекст справа
3. l_1[0] — потенциальный участник связи
4. l_1[1,2] — контекст слева
5. hd(l_1[0]) — хозяин l_1[0]
6. ld(l_1[0]) — левое зависимое слово l_1[0]
7. rd(l_1[0]) — правое зависимое слово l_1[0]
8. l_2[0,n] — контекст между потенциальными участниками связи

Кроме того, были добавлены дополнительные признаки, которые могли бы повлиять на качество разбора.

1. ngc(b[0]) — число-род-падеж b[0]
2. ngc(l_1[0]) — число-род-падеж l_1[0]
3. {ngc(b[0]), ngc(l_1[0])} кортеж из b[0] и l_1[0]
4. interior — полный контекст между b[0] и l_1[0]

Кроме того, в модель была неявно включена длина связи. В дополнение к простым признакам связи добавлялись кортежи этих признаков и длины связи.

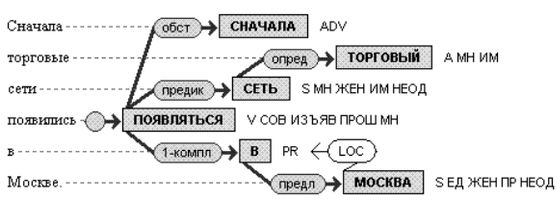

Рис 2. Синтаксическая структура предложения «Сначала торговые сети появились в Москве»

Таблица 1. Иллюстрация разбора с помощью системы действий

Начальная конфигурация	Действие	Построенные связи
[ROOT] [] [Сначала, торговые, сети, появились, в, Москве]	Shift	[]
[ROOT, Сначала] [] [торговые, сети, появились, в, Москве]	Shift	[]
[ROOT, Сначала, торговые] [] [сети, появились, в, Москве]	LeftArc(ОПРЕД)	[$(3\rightarrow2)_{\text{ОПРЕД}}$]
[ROOT, Сначала] [торговые] [сети, появились, в, Москве]	Shift	[$(3\rightarrow2)_{\text{ОПРЕД}}$]
[ROOT, Сначала, торговые, сети] [] [появились, в, Москве]	LeftArc(ПРЕДИК)	[$(3\rightarrow2)_{\text{ОПРЕД}}$,$(4\rightarrow3)_{\text{ПРЕДИК}}$]
[ROOT, Сначала, торговые] [сети] [появились, в, Москве]	NoArc	[$(3\rightarrow2)_{\text{ОПРЕД}}$,$(4\rightarrow3)_{\text{ПРЕДИК}}$]
[ROOT, Сначала] [торговые, сети] [появились, в, Москве]	LeftArc(ОБСТ)	[$(3\rightarrow2)_{\text{ОПРЕД}}$,$(4\rightarrow3)_{\text{ПРЕДИК}}$, $(4\rightarrow1)_{\text{ОБСТ}}$]

Начальная конфигурация	Действие	Построенные связи
[ROOT] [Сначала, торговые, сети] [появились, в, Москве]	RightArc(ROOT)	$[(3\rightarrow2)_{\text{ОПРЕД}},(4\rightarrow3)_{\text{ПРЕДИК}},$ $(4\rightarrow1)_{\text{ОБСТ}},(0\rightarrow4)_{\text{ROOT}}]$
[] [ROOT, Сначала, торговые, сети] [появились, в, Москве]	Shift	$[(3\rightarrow2)_{\text{ОПРЕД}},(4\rightarrow3)_{\text{ПРЕДИК}},$ $(4\rightarrow1)_{\text{ОБСТ}},(0\rightarrow4)_{\text{ROOT}}]$
[ROOT, Сначала, торговые, сети, появились] [] [в, Москве]	RightArc(1-КОМПЛ)	$[(3\rightarrow2)_{\text{ОПРЕД}},(4\rightarrow3)_{\text{ПРЕ-ДИК}},(4\rightarrow1)_{\text{ОБСТ}},(0\rightarrow4)_{\text{ROOT}},$ $(4\rightarrow5)_{\text{1-КОМПЛ}}]$
[ROOT, Сначала, торговые, сети] [появились] [в, Москве]	Shift	$[(3\rightarrow2)_{\text{ОПРЕД}},(4\rightarrow3)_{\text{ПРЕ-ДИК}},(4\rightarrow1)_{\text{ОБСТ}},(0\rightarrow4)_{\text{ROOT}},$ $(4\rightarrow5)_{\text{1-КОМПЛ}}]$
[ROOT, Сначала, торговые, сети, появились, в] [] [Москве]	RightArc(ПРЕДЛ)	$[(3\rightarrow2)_{\text{ОПРЕД}},(4\rightarrow3)_{\text{ПРЕ-ДИК}},(4\rightarrow1)_{\text{ОБСТ}},(0\rightarrow4)_{\text{ROOT}},$ $(4\rightarrow5)_{\text{1-КОМПЛ}},(5\rightarrow6)_{\text{ПРЕДЛ}}]$
[ROOT, Сначала, торговые, сети, появились] [в] [Москве]	Shift	$[(3\rightarrow2)_{\text{ОПРЕД}},(4\rightarrow3)_{\text{ПРЕ-ДИК}},(4\rightarrow1)_{\text{ОБСТ}},(0\rightarrow4)_{\text{ROOT}},$ $(4\rightarrow5)_{\text{1-КОМПЛ}},(5\rightarrow6)_{\text{ПРЕДЛ}}]$
[ROOT, Сначала, торговые, сети, появились, в, Москве] [] []		$[(3\rightarrow2)_{\text{ОПРЕД}},(4\rightarrow3)_{\text{ПРЕ-ДИК}},(4\rightarrow1)_{\text{ОБСТ}},(0\rightarrow4)_{\text{ROOT}},$ $(4\rightarrow5)_{\text{1-КОМПЛ}},(5\rightarrow6)_{\text{ПРЕДЛ}}]$

5. Эксперименты

Для оценки рассматриваемых подходов были проведены эксперименты. В качестве материала использовался корпус СинТагРус [8]. На настоящий момент в корпусе около 40 тысяч предложений, около 600 тысяч слов. В текстах используется порядка 32 тысяч лемм.

Эксперименты проводились следующим образом. Корпус делился на две части — часть для обучения, а часть для оценки полученной модели. В результате эксперимента предполагалось оценить эффективность каждого из подходов на основе машинного обучения по отношению к системе ЭТАП. Поскольку алгоритмы построения минимального остовного дерева сильно зависят от возможности построения непроективных связей, то системы сравнивались по промежуточному параметру — количеству правильно построенных связей.

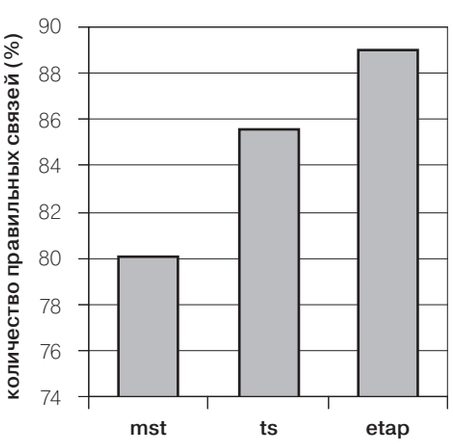

Рис. 3. Процент правильных связей

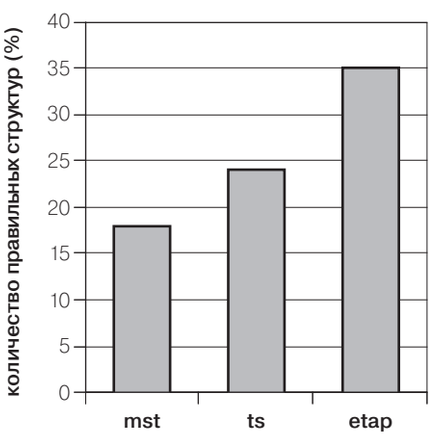

Рис. 4. Процент правильных структур

Для обучения использовалось 10 тысяч предложений, для тестирования — 5 тысяч. Для подхода на основе максимальных остовных деревьев строилось две системы принятия решений: для оценки возможности связи(проведение связи без имени), и восстановление имени связи. Такое разделение было необходимо из-за вычислительной сложности подхода.

На рис. 3 представлены данные по точности построения связей при каждом из рассмотренных подходов(mst — подход на основе максимальных остовных деревьев, ts — на основе системы переходов, etap — система ЭТАП-3). Из этих данных следует, что в абсолютных величинах разница между подходом на основе системы переходов и эталонной системой ЭТАП-3 не такая большая(около 4 %). Однако, такая разница дает более 10 % разницы в количестве построенных структур.

6. Результаты и перспективы

В работе приведена экспериментальная реализация подхода к синтаксическому анализу на основе системы переходов. Несмотря на то, что система на основе машинного обучения не показала лучшие результаты, она является довольно конкурентоспособной. В настоящей работе использовался линейный вариант SVM, в результате чего возник небольшой проигрыш относительно результатов, представленных в [4]. Однако, этот факт компенсируется значительным увеличением скорости работы парсера.

В настоящей работе модель признаков была богаче, чем в [4], поскольку представленная система ориентирована на использование совместно с корпусом СинТагРус, где представлены более богатые лингвистические характеристики.

Поэтому представляется перспективным продолжить работы в области создания статистического парсера, одновременно с этим изучая возможности создания гибридной системы.

Литература

1. *Апресян Ю. Д., Богуславский И. М., Иомдин Л. Л.* и др. Лингвистическое обеспечение системы ЭТАП-2. М., Наука, 1992
2. *Мельчук И. А.* Опыт теории лингвистических моделей «Смысл ↔ Текст». М., Наука,1974.
3. *McDonald R., Crammer K., Pereira F.* Spanning Tree Methods for Discriminative Training of Dependency Parsers. // ACL-06
4. *Nivre J.* Algorithms for Deterministic Incremental Dependency Parsing // Comp. Linguistics Vol 34, No 4.
5. *McDonald R., Satta G.* On the Complexity of Non-Projective Data-Driven Dependency Parsing. // IWPT-2007
6. *Crammer K., Singer Y.* Ultraconservative Algorithms for Multiclass Problems. // JMLR '01
7. *Апресян Ю. Д.* Идеи и методы современной структурной лингвистики. // М. Просвещение, 1966.
8. *Апресян Ю. Д., Богуславский И. М., Иомдин Б. Л., Иомдин Л. Л., Санников А. В., Санников В. З., Сизов В. Г., Цинман Л. Л.* Синтаксически и семантически аннотированный корпус русского языка: современное состояние и перспективы // Национальный корпус русского языка 2003–2005 г. (результаты и перспективы). М: «Индрик», 2005. С. 193–214