Компьютерная лингвистика и интеллектуальные технологии:
по материалам международной конференции «Диалог 2017»

Москва, 31 мая — 3 июня 2017

СИНТАКСИЧЕСКИЙ ПАРСЕР РУССКОГО ЯЗЫКА LPARUS КОМПАНИИ MEGAPUTER INTELLIGENCE

Киселёв М. В. (mkiselev@megaputer.ru)
Компания Megaputer Intelligence, Москва, Россия

Федосеева Д. В. (fedoseeva@megaputer.ru)
Компания Megaputer Intelligence/
МГУ имени М. В. Ломоносова, Москва, Россия

В данной статье описываются основные принципы работы синтаксического парсера русского языка LPaRus, разработанного на основе лингвистических технологий компании Megaputer Intelligence. Синтаксический анализатор LPaRus использует для своего анализа формализм грамматики зависимостей. Алгоритм, лежащий в его основе, базируется на правилах соединения направленных коннекторов, приписанных словам, в некоторый ациклический граф, имеющий определенную цену. В результате у каждого полученного варианта синтаксического разбора имеется определенная «стоимость», что позволяет выбрать наиболее правильный и полный анализ из возможных. В последней части данной статьи обсуждается эффективность парсера LPaRus, оценка результатов его работы, а также основные виды ошибок, допускаемых данным алгоритмом в его текущей реализации.

Ключевые слова: грамматика зависимостей, синтаксис русского языка, синтаксический парсер, обработка естественного языка

RUSSIAN SYNTACTIC PARSER LPARUS OF COMPANY MEGAPUTER INTELLIGENCE

Kiselev M. V. (mkiselev@megaputer.ru)
Megaputer Intelligence, Moscow, Russia

Fedoseeva D. V. (fedoseeva@megaputer.ru)
Megaputer Intelligence/Moscow State University, Moscow, Russia

In this paper we describe underlying mechanisms of Russian syntactic parser LPaRus based on linguistic technologies developed by Megaputer Intelligence company. The parser is rule-based and uses Dependency Grammar formalism. On the first step of the analysis algorithm operates with linking requirements for each word called connectors. Based on syntactic rules, two connectors can be bound into a syntactic link and each link has its own "price". The "price" depends on three basic factors: link's type, its length and frequency. As a result of analysis, the algorithm outputs an acyclic graph that spans all words in the sentence, connected by links. In case when there can be more than one possible graph for a sentence (so more than one variant of syntactic analysis), the "cheapest" candidate is preferred. To choose the candidate with the minimum "price", the algorithm evaluates graph's "cost" which includes, among other factors, "price" of each link and "penalties" for violations of basic syntactic principles, for example, planarity. On this step the parser chooses the "cheapest" variant with "cheapest" links and without major syntactic violations.

In the last part of this paper we discuss quality evaluation aspects of LPaRus, its efficiency, precision and recall. We also describe main problems and limitations of this version of LPaRus as well as possible ways to improve the algorithm.

Keywords: natural language parsing, dependency grammar, Russian syntax, syntactic parser, natural language processing

В данной работе описан синтаксический парсер русского языка LPaRus, основанный на лингвистических разработках компании Megaputer Intelligence. Для определения синтаксической структуры предложений на русском языке нами был выбран метод зависимостей как наиболее подходящий для языков со свободным порядком слов и богатой морфологией. Наш подход основан на правилах, так как данная стратегия, в отличие от машинного обучения, позволяет быстро и эффективно внести изменения в синтаксический анализатор, подстроив его под определенный вид текстов: пресса, художественная литература, форум, отзывы на продукты и т. д.

Одним из основных синтаксических парсеров русского языка является синтаксический анализатор системы ЭТАП-3, в основе которого лежит лингвистическая теория Мельчука и Жолковского «Смысл-Текст» [Mel'čuk 1974, 1999].

При помощи данного анализатора был размечен синтаксический корпус русского языка СинТагРус. Анализатор ЭТАП-3 основан на правилах (синтагмах), связывающих два слова предложения некоторой связью (подробнее см. [Iomdin et al. 2012]), и использует толково-комбинаторный словарь, описывающий синтаксические, семантические и сочетаемостные свойства слов. К недостаткам данного парсера относятся невозможность его использования для анализа разговорной речи и текстов, отклоняющихся от литературной нормы, а также некорректный анализ длинных предложений (от 60 слов).

К числу наиболее эффективных парсеров также относится универсальный языконезависимый MaltParser. Для его использования не требуется большого объема аннотированных данных, а также не производится настройка параметров под определенный язык. Точность разметки данного парсера составляет, согласно [Nivre et al. 2007], около 80–90% для различных языков. Этот алгоритм был применен к русскому языку при помощи обучения на синтаксическом корпусе СинТагРус ([Sharoff, Nivre 2011]), при этом его точность составила 82,3. Полученный анализатор активно применяется в различных работах, в частности, в [Droganova 2015]. Однако он основан на машинном обучении, а значит, обладает недостатками парсеров данного вида, в частности, он не позволяет быстро скорректировать полученные результаты, для этого необходимо переобучение на новых размеченных данных.

Синтаксический анализатор компании Yandex SyntAutom [Antonova, Misyurev 2012] использует для построения анализа правила и морфологический словарь, кроме того, в анализе задействованы конечные автоматы и частотность синтаксических связей. В систему Abbyy Compreno также встроен синтаксический парсер, запускающийся одновременно с семантическим анализатором и основывающийся на синтаксических парадигмах (подробнее в [Anisimovich at al. 2012]), описывающих синтаксическую и отчасти семантическую сочетаемость лексемы. Однако данные системы являются закрытыми и предназначены исключительно для внутреннего использования.

В состав открытого парсера Tomita компании Яндекс [Tomita Parser] входит контестно-свободная грамматика, позволяющая описывать синтаксические конструкции. Однако в данной программе не представлено полноценного синтаксического парсера, его компоненты присутствуют исключительно в качестве вспомогательных элементов, позволяющих выделить факты.

Наш подход вдохновлен синтаксическим парсером английского языка Link Grammar Parser, который строит дерево зависимостей для некоторого предложения на английском языке, а также представляет его в виде структуры составляющих. В отличие от системы ЭТАП-3, Link Parser способен пропустить части предложения, для которых он не способен предоставить разбор, и приписать связи оставшимся [Sleator, Temperley 1991]. Каждое из слов имеет определенные синтаксические сочетаемостные требования, которые приписаны ему в словаре и называются в данной работе ***коннекторами***. Соединение двух направленных коннекторов различных слов между собой означает проведение связи между этими словами и производится при помощи применения различных правил. У некоторых связей может присутствовать несколько возможных

коннекторов, в таком случае должен быть реализован хотя бы один из них. Link Parser, согласно [Sleator, Temperley 1991], затрачивает на обработку среднего предложения несколько секунд и способен применять ряд эвристик для ускорения данного процесса. Большинство ошибок данного парсера происходит из-за того, что в нём описаны не все типы возможных конструкций, в частности, любые расхождения со стандартной грамматикой провоцируют ошибочный разбор. Как и синтаксический парсер ЭЛАН-3, Link Parser плохо справляется с конструкциями, содержащими прямую или косвенную речь.

Большинство синтаксических парсеров испытывают затруднения при разборе предложений, содержащих грамматические либо орфографические ошибки. В системе PolyAnalyst, частью которой является парсер LPaRus, перед синтаксическим анализом возможно провести опциональную проверку правописания (SpellCheck node) и проверку грамматичности предложения (GrammarCheck node) и исправить ошибки исходного текста.

Как и в системе LinkParser, описанной выше, в нашем парсере каждому токену приписывается некоторый набор валентностей или коннекторов, которые могут быть специфически левыми или правыми. Используя правила совместимости коннекторов, а также некоторые дополнительные условия, например, требование ***проективности***, LPaRus строит для предложения все возможные конфигурации связей, охватывая, по возможности, все слова предложения. Для выбора наиболее подходящего синтаксического разбора, каждому полученному графу зависимостей по определенным правилам приписывается некоторая «цена», при этом предпочтительным является самый «дешевый» граф зависимостей. Отдельные этапы данной процедуры будут рассмотрены ниже.

1. Входное представление предложения

Входными данными для модуля LPaRus являются отдельные предложения, разбитые на токены при помощи алгоритмов токенизации. Как правило, один токен соответствует одному слову или знаку препинания, однако возможны и более сложные токены. В частности, предполагается, что перед применением синтаксического анализатора LPaRus в предложении были выделены именованные сущности (например, имена людей, названия организаций, географические названия) и метаязыковые объекты (интернет-адреса, адреса электронной почты и т. д.). Каждый из этих объектов представляется одним токеном, при этом для многословных конструкций при помощи различных эвристик выделяется вершина, по которой происходит согласование. Например, для объекта *агентства “Интерфакс”* вершиной будет слово ***агентства***.

2. Препроцессинг и лемматизация

На втором этапе выделенные токены подвергаются лемматизации, в результате которой каждому токену сопоставляется набор пар *<нормальная форма*, грамматические атрибуты>, соответствующий данной словоформе. После этого происходит процедура первичного снятия грамматической

неоднозначности при помощи набора правил, описывающих, какой из разборов является наиболее предпочтительным. Например, если для некоторой словоформы один из разборов представляет собой лемму с категорией <предлог>, остальные возможные разборы исключаются из рассмотрения: так, для токена ***для*** выбирается лемма <*для*, предлог>, но не <*длить*, глагол>.

3. Предварительное выделение частей сложносочиненных и сложноподчиненных предложений

На третьем этапе алгоритма на основе союзов, союзных слов и знаков препинания выделяются части сложносочиненных и сложноподчиненных предложений, например, при помощи конструкций [, потому что], [, который], а также происходит анализ синтаксических конструкций, содержащих в себе скобки и двоеточия. Этот этап алгоритма позволяет сузить пространство поиска возможных зависимостей между словами и повысить общую эффективность и скорость анализа. В дальнейшем мы будем рассматривать процедуру построения дерева зависимостей для простого предложения.

4. Определение подлежащего и сказуемого предложения

В LpaRus используются различные способы для нахождения связей между главными членами предложения и для определения остальных связей. Это вызвано тем, что нередко связи между главными членами предложения образуют не граф, а некоторую более общую структуру — гиперграф, как, например, в примере 1.

(1) *Маша и Витя пошли в лес и увидели там ежа*.

Для второстепенных членов эта ситуация встречается существенно реже, например, в примере 2, поэтому в текущей версии парсера мы ею пренебрегаем.

(2) *Кузнец ковал и закалял мечи и кинжалы*.

Для нахождения подлежащего и сказуемого предложения используется следующий механизм. Вначале находятся все слова, претендующие на роль подлежащего, в частности, все существительные, допускающие их интерпретацию в именительном падеже. Для полученных кандидатов строятся все комбинации их трактовок в качестве подлежащего или второстепенных членов.

После этого для каждого возможного варианта подлежащего находятся все потенциальные сказуемые, например, финитные глаголы, определенные классы кратких прилагательных и причастий, эллиптические конструкции с дефисом и/или существительным в именительном падеже и др. Из всех полученных пар <подлежащее — сказуемое> исключаются несогласованные пары, остальным вариантам разбора присваивается различная «цена». Понятие «цены» будет раскрыто в следующем разделе. Для всех полученных пар воспроизводятся шаги, описанные ниже, и, как уже отмечалось, из множества получаемых графов связей выбирается самый «дешевый».

5. Построение дерева зависимостей

После разбиения сложного предложения на части и определения его главных членов алгоритм переходит к основному этапу анализа.

Перед построением деревьев зависимостей производится несколько подготовительных шагов. Прежде всего, для каждого слова определяется диапазон позиций в предложении, на которых могут находиться его зависимые. Данный диапазон ограничивается, прежде всего, требованием проективности, например, наличие подчинения наречия ***невероятно*** прилагательному ***пушистая*** в примере 3 делает невозможными инверсии 4 и 5, нарушающие свойство проективности.

(3) *Во дворе гуляла **невероятно пушистая** собака.*

(4) *Во дворе **невероятно** гуляла **пушистая** собака.*

(5) *Во дворе гуляла **пушистая** собака **невероятно**.*

На подготовительном шаге анализа для каждой пары слов, выбранных с учетом полученных на предыдущем шаге диапазонов возможных зависимостей, определятся возможность их интерпретации как главное — зависимое с некоторым видом связи. Используемая в LPaRus система типов связей между словами частично соответствует системе синтаксических отношений, принятой в корпусе СинТагРус (http://ruscorpora.ru/instruction-syntax.html). Однако присутствуют определенные различия, связанные с нашим подходом к задаче построения дерева зависимостей, основанным на системе «цен», приписанных различным компонентам и свойствам дерева. В частности, свой вклад в «цену» разбора предложения дает каждая связь, при этом его размер отражает вероятность, распространенность связи. Например, в нашем подходе различаются правая (*AN_R*) и левая (*AN_L*) связи между прилагательным (причастием) и существительным, так как они обладают существенными различиями в частотности: левая связь *AN_L* ***белого*** ← ***света*** встречается очень часто, а правая *AN_R* ***света*** → ***белого***, наоборот, весьма редка. Данная закономерность отображается в том, что «цена» связи *AN_R* в 10 раз больше «цены» связи *AN_L*. При этом в корпусе СинТагРус обе связи попадают в категорию *определительных синтаксических отношений*.

В целом, система синтаксических связей LPaRus более дробная по сравнению с принятой в СинТагРусе. Это связано с необходимостью ограничения множества рассматриваемых связей для сокращения перебора возможных вариантов. Например, связи между словами ***приказал*** → ***солдату*** и ***приказал*** → ***петь*** согласно системе СинТагРус входят в одно синтаксическое отношение — *первое комплетивное*, в нашей же системе они принадлежат к различным типам: *NDATV* и *INFINITIVEV*, так как, если в сочетании ***приказал солдату петь*** связи слова ***приказал*** с обоими зависимыми словами были бы одного типа, то ***солдату*** и ***петь*** считались бы однородными членами. Отсутствие между ними союза или запятой оценивалось бы как сомнительный вариант разбора предложения, таким образом, маркировка данных связей различными типами позволяет обойти эту проблему. Необходимо отметить, что в некоторых аспектах система связей слов СинТагРуса значительно подробнее, так как в большей

степени учитывает семантические аспекты. Например, в корпусе СинТагРус в сочетаниях ***привезли ящик*** и ***привезли ящик книг*** выделяются различные связи: *первое комплетивное* в первом случае и *количественно-копредикативное* во втором, в нашей же системе в обоих случаях устанавливается связь *NACCV*.

Существенное отличие между СинТагРус и LPaRus состоит также в наличии в нашей системе значительного количества технических видов связей, введенных для того, чтобы распространить строгий формализм парных связей слов на устойчивые многословные нелокальные конструкции. При этом были введены особые связи между компонентами этих конструкций, учитывающие их структуру и связи с остальными словами. Например, корректное связывание в сочетании ***добавивший такие средства, как спирт*** требует введения специальной непроективной связи *ASSUCH_R* между словами ***такие*** и ***как***. Полное рассмотрение нашей системы синтаксических отношений между словами в данной статье невозможно по соображениям ее объема.

Таким образом, система связей СинТагРуса более стройна и логична теоретически, в то время как система LPaRus оптимизирована для применяемого в нём подхода к нахождению зависимостей.

Следующий шаг анализа заключается в определении «цены» каждой из найденных связей. «Цена» связи складывается из трех параметров:

1) «цена» типа связи;
2) длина связи в токенах;
3) частотность данного сочетания слов.

Для вычисления третьего параметра нами была построена статистика сочетаемости слов на больших русскоязычных корпусах.

На следующем этапе список потенциальных связей сортируется в порядке возрастания их цен. Сама процедура построения дерева имеет вид алгоритма перебора. В процессе перебора мы движемся от начала списка связей к его концу, перебирая состояния связей («включена» — «выключена»). Если какая-то связь «включается», то в хвосте этого списка принудительно «выключаются» все связи, несовместимые с ней. Причины этой несовместимости могут быть различны. Во-первых, слово не может быть зависимым одновременно у двух главных слов, так, связь ***отзыв → на*** в ***послал отзыв на книгу*** исключает связь ***послал → на***. Во-вторых, при фиксации связи происходит разрешение грамматической неоднозначности: в сочетании ***лишенные прав совершенно*** связь ***лишенные → прав*** <сущ.> исключает связь ***прав*** <прил.> → ***совершенно*** к омонимичному прилагательному ***прав***. В-третьих, несовместимость связей зачастую обусловлена требованием проективности. В сочетании ***бесконечного возвращения алгоритма к началу*** связь ***возвращения → к*** исключает ***алгоритма → бесконечного***, так как в таком случае нарушается требование проективности. Описанные выше ограничения приводят к тому, что перебор вместо теоретических 2^N вариантов (N — количество связей) осуществляется лишь по их малому подмножеству. При этом для небольших предложений, состоящих из нескольких слов, зачастую остается только один возможный вариант разбора.

На следующем шаге производится предварительная проверка правильности полученных возможных синтаксических разборов, которая оценивается по следующим критериям:

1) **Полнота**: деревья должны включать все смысловые слова предложения.
2) **Достижимость из главных членов**: корнями всех деревьев должны быть главные члены предложения.
3) **Функциональность сочинительных союзов и запятых, разделяющих однородные члены**: наличие сочинительных союзов, а также, в определенных случаях, запятых должно соответствовать наличию однородных членов.

После определения корректности всех возможных синтаксических разборов предложения, если найдена хотя бы одна корректная система, все некорректные решения исключаются из рассмотрения.

На последнем этапе происходит оценка всех корректных деревьев зависимостей, полученных для определенного предложения. «Цена» варианта синтаксического разбора складывается из суммы «цен» выявленных связей, «штрафов» за нарушение структурных особенностей системы, например, за отсутствие первого актанта у переходного глагола, и «штрафов» за разбор, не соответствующий пунктуации. Как уже было отмечено выше, в качестве окончательной принимается система деревьев зависимостей, получившая наименьшую «цену».

В том случае, если построить корректное дерево зависимостей для данного предложения невозможно, применяются эвристики, позволяющие производить анализ, основанный на локальном связывании слов, без анализа корректности всей системы связей в целом.

Таким образом, в конце разбора мы получаем синтаксический разбор предложения в виде дерева зависимостей (пример реализации в интерфейсе программы см. на Схеме 1).

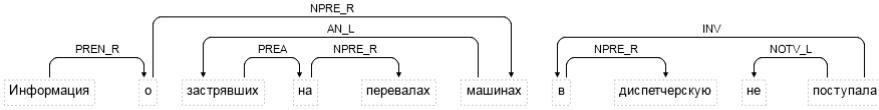

Схема 1: Пример дерева зависимостей, построенного при помощи LPaRus[1]

[1] На данной схеме не отображена связь между предикатом ***поступала*** и субъектом ***информация***.

6. Оптимизация и меры, направленные на повышение устойчивости анализа

Одним из важнейших параметров эффективности работы парсера LPaRus является скорость синтаксического анализа. Переборные алгоритмы являются вычислительно-интенсивными по их сути, и поэтому требуют специальных усилий по их оптимизации. В отношении описываемого алгоритма существуют два фактора, радикально влияющих на его скорость, — это длина предложения и наличие в нем несловарных слов.

Рост количества потенциальных связей в зависимости от длины предложения может быть весьма значительным, в сочетании со вторым фактором — почти экспоненциальным. Для решения данной проблемы было введено несколько защитных механизмов, однако их явно не достаточно. Главный из них — это переход в специальный «панический» режим перебора при достижении длительности анализа предложения определенного значения. В «паническом» режиме производится насильственная фиксация некоторого непротиворечивого набора наиболее дешевых связей с ограничением количества просматриваемых вариантов других связей. Более радикальными экспериментальными мерами являются жесткое ограничение длины связи и принудительное разбиение предложения на части.

В том случае, если в предложении присутствует несловарное слово, допускающее большое количество потенциальных вариантов лемматизации, синтаксическому парсеру приходится рассматривать гораздо больше потенциальных связей, нежели для предложения с известными словами. Наличие нескольких несловарных слов в предложении приводит к появлению сотен потенциальных связей, именно поэтому иноязычные вкрапления в тексте на русском языке существенно снижают скорость работы и эффективность алгоритма. Основными методами борьбы с данной проблемой могут быть более точные алгоритмы морфологического анализа, разумная стратегия фильтрации «мусора» и оптимизация системы типов связей.

Отдельным механизмом повышения эффективности и устойчивости анализа является включенная в LPaRus возможность задания ожидаемого уровня грамотности текста:

1 уровень. Грамотный текст с верной пунктуацией;
2 уровень. Текст с ошибками в пунктуации;
3 уровень. Текст без пунктуации и большим количеством неграмотно написанных и/или намеренно искаженных слов.

В данной статье мы рассматриваем алгоритм, предназначенный для анализа грамотных текстов, относящихся к первому уровню. Модификации алгоритма для второго и третьего уровня на основе данной априорной оценки выходят за рамки данной статьи.

Необходимо отметить, что парсер LpaRus обладает специфическим набором связей, в частности, включает в себя разнообразные подтипы стандартных связей, например, левые и правые, поэтому машинное обучение с учителем при помощи находящихся в открытом доступе размеченных корпусов на данном

этапе не представляется возможным. Кроме того, предложенные нами типы связей затрудняют сравнение параметров точности и полноты парсера с другими синтаксическими анализаторами.

7. Тестирование синтаксического анализатора LPaRus

Эффективность работы синтаксического парсера LPaRus оценивалась в терминах точности и полноты. Тестирование работы LPaRus проводилось для режима первого уровня грамотности: на грамотных текстах с верной пунктуацией. Для этого был создан подкорпус новостей с сайтов новостных агентств «Интерфакс» и «ИТАР-ТАСС» величиной 800 предложений, общее число слов подкорпуса — 15 572, среднее число слов в предложении — 20.

Ввиду специфики нашего подхода отдельно были протестированы точность и полнота определения субъекта и предиката, результаты тестирования представлены в Таблице 1.

Таблица 1. Результаты тестирования точности и полноты выделения субъекта и предиката предложения

	Точность	**Полнота**	**F-мера**
Субъект	80,48	91,52	85,64
Предикат	86,81	89,73	88,23

Основные ошибки при выделении субъекта происходят в предложениях с прямым объектом, омонимичным номинативу, например, «мастера украсили входы на станцию» — в качестве субъекта ошибочно выделяется существительное ***входы***. Кроме того, зачастую субъектом становятся слова в номинативе, обозначающие название объекта, например, ***зодчество*** в конструкции «XXI международного фестиваля „Зодчество“». Ошибки при определении предиката предложения связаны в основном с определением именной части сказуемого, а также с определением предиката в предложениях с опущенной глагольной связкой.

Точность и полнота определения других связей в предложении представлены в таблице 2[2]:

[2] В данной таблице представлены наиболее частотные связи

Таблица 2. Результаты тестирования связей парсера. L — левая связь; R — правая связь

Обозначение связи		Комментарий (тип зависимого объекта)	Число связей	Точность	Полнота	F-мера
ADVERBPARTICIPLEV		Глагольный модификатор — деепричастие	27	76,00	90,48	82,61
AN_L		Адъективный модификатор существительного	1180	97,33	95,04	96,17
AN_R			84	82,61	79,17	80,85
AVAUX		От глагола «быть» к краткому причастию или прилагательному	69	90,77	93,65	92,19
CONJ_THAT_V_R		Подчинительный союз	27	100,00	74,07	85,11
COOCON		Сочинительный союз	110	91,46	72,82	81,08
DATEPRE_R		Дата	75	98,67	100,00	99,33
ENTITYNSAMECASE		Именованная сущность	154	80,00	81,25	80,62
INFINITEVEV		Инфинитив	166	83,33	88,65	85,91
NACCV		Прямой объект глагола	477	64,07	76,35	69,67
NN	*N_GEN_N*	Существительное в различных падежах при именной группе	1320	86,38	88,09	87,23
	N_DAT_N		15	92,86	92,86	92,86
	N_INST_N		116	81,13	89,58	85,15
NV	*N_GEN_V*	Непрямой беспредложный объект глагола	33	67,74	91,30	77,78
	N_DAT_V		26	92,86	92,86	92,86
	N_INST_V		116	81,13	89,58	85,15
NNUMBER_R		От существительного к количественному числительному	88	82,26	66,23	73,38
NPRE_R		От предлога к существительному	1482	88,68	91,59	90,26
ORDINALN_L		Порядковое числительное	27	84,62	95,65	89,80
PP_V		Связь от глагольной группы к предложной	862	78,69	91,85	84,53
PREN_R		Связь от именной группы к предложной	664	65,43	79,27	71,69
PRONPOSESSN		Притяжательное местоимение	67	91,07	82,26	86,44
RV_L		Глагольный модификатор (наречие)	150	82,79	78,29	80,48
RV_R			75	65,52	69,09	67,26
NOT		Отрицание	69	100,00	95,65	97,78

Итак, нами было размечено 8000 связей, результаты тестирования показали, что точность анализа LPaRus составляет 81,66, полнота — 85,73, F-мера — 82,76. Таким образом, LpaRus по ряду показателей уступает некоторым ведущим парсерам зависимостей для русского языка (результаты соревнования синтаксических парсеров представлены в [Toldova et al. 2012, 356]), однако необходимо учитывать, что наш парсер располагает большим количеством потенциальных связей, поэтому вероятность ошибочного выбора между несколькими потенциальными связями выше, чем у других анализаторов.

Основные проблемы анализатора возникают при обработке длинных сложносочиненных и сложноподчиненных предложений, так как алгоритм

не всегда справляется с выбором из большого разнообразия полученных связей и в ряде случаев насильственно прерывается для быстроты анализа. Ещё одной проблемой для нашего парсера является неспособность грамотно анализировать кавычки и скобки в таких конструкциях, как ***космический корабль «Прогресс М-27М»***, что ведет к неверному анализу предложения. Как уже было отмечено выше, присутствие иноязычных вкраплений и других несловарных лемм провоцирует ошибки, например, в сочетании ***лауреат премии BBC Awards***. В текущей реализации LPaRus в ряде случаев не справляется с эллиптическими конструкциями, вводными словами и инверсией. При этом конструкции, содержащие прямую речь, в большинстве случаев анализируются корректно.

Таким образом, парсеру LPaRus не удается в полной мере избежать ошибок, характерных для синтаксических анализаторов, основанных на правилах и статистике. Однако ведется работа над улучшением существующих механизмов, при этом данный парсер уже сейчас может быть использован для решения прикладных задач, таких как извлечение отношений между именованными сущностями и анализ тональности текста.

Коллектив авторов выражает благодарность Азерковичу Илье, Антошину Денису, Ермаковой Екатерине, Кокоповой Виктории, Петуховой Елене и Тюриной Людмиле за помощь в разметке корпусов для тестирования парсера.

Литература

1. *Anisimovich K. V., Druzhkin K. Ju., Minlos F. R., Petrova M. A., Selegey V. P., Zuev K. A.* (2012), Syntactic and semantic parser based on ABBYY Compreno linguistic technologies. // Computational Linguistics and Intellectual Technologies: Papers from the Annual International Conference “Dialogue” (2012). — Issue 11 (18). — Vol. 2 — Moscow, RGGU, 2012. pp. 91–104.
2. *Antonova A. A., Misyurev A. V.* (2012), Russian dependency parser SyntAutom at the DIALOGUE-2012 parser evaluation task. // Computational Linguistics and Intellectual Technologies: Papers from the Annual International Conference “Dialogue” (2012). — Issue 11 (18). — Vol. 2 — Moscow, RGGU, 2012. pp. 104–119.
3. *Droganova K.* (2015), Building a Dependency Parsing Model for Russian with MaltParser and MyStem Tagset // Proceedings of the AINL-ISMW FRUCT, Saint-Petersburg, Russia, 9–14 November 2015, ITMO University, FRUCT Oy, Finland.
4. *Iomdin L. L., Petrochenkov V. V., Sizov V. G., Tsinman L. L.* (2012), ETAP parser: state of the art. // Computational Linguistics and Intellectual Technologies: Papers from the Annual International Conference “Dialogue” (2012). — Issue 11 (18). — Moscow, RGGU, 2012. pp. 119–132.
5. Link Parser, available at: http://www.link.cs.cmu.edu/link/index.html
6. *Mel’čuk I.* (1974/1999), Opyt teorii lingvisticheskih modelej Smysl ⇔ Tekst. [The theory of linguistic models ‘Meaning ⇔ Text’]. Moscow, Nauka; Jazyki russkoj kultury.
7. *Nivre, J., Hall J., Nilsson J., Chanev A., Eryigit G., Kübler S., Marinov S., Marsi E.* (2007) MaltParser: A language independent system for data-driven dependency parsing. // Natural Language Engineering, 13, pp. 95–135.

8. *Sharoff S., Nivre J.* (2011) The proper place of men and machines in language technology. Processing russian without any linguistic knowledge. // Computational Linguistics and Intelligent Technologies: Proceedings of the International Conference Workshop Dialogue 2011). Vol. 10 (17), 2011. Moscow: RGGU, pp. 657–670.
9. *Sleator D., Temperley D.* (1991), Parsing English with a Link Grammar. Carnegie Mellon University Computer Science technical report CMU-CS-91–196.
10. *Toldova S., Gareyshina A., Ionov M., Lyashevskaya O., Privoznov D., Sokolova E.* (2012), Ru-eval-2012: Evaluating dependency parsers for russian. // Proceedings of COLING 2012: Posters. IIT Bombay, Mumbai, India, 2012. pp. 349–360.
11. Tomita Parser, available at: https://tech.yandex.ru/tomita/