Ещё раз о способах снятия структурной омонимии: выбор единственной структуры в парсере Hurma

А.М. Попов[1], Е.В. Протопопова[2], Г.Т. Букия [3]

[1] СПбГУ, проект Хурма, hedgeonline@gmail.com

[2] СПбГУ, проект Хурма, protoev@gmail.com

[3] СПбГУ, gregorybookia@yandex.ru

Аннотация

Целью проекта Хурма является создание исследовательской платформы для анализа текстов на естественном языке, в первую очередь на русском. Данная платформа предполагает анализ текстов на различных языковых уровнях, в том числе и на синтаксическом.

Возможность выбора единственной синтаксической структуры в результате синтаксического анализа востребована в большинстве прикладных задач, использующих синтаксическую информацию: для создания размеченных корпусов, извлечения информации из текста (фактов и именованных сущностей), машинного перевода и т.д.

Наше исследование посвящено способам выбора наиболее правильной синтаксической структуры и выделению критериев, по которым такой выбор становится возможным. Мы предлагаем гибридный подход к взвешиванию синтаксических структур, использующий как критерии, задаваемые вручную, так и статистическую информацию о лексической сочетаемости синтаксически связываемых слов.

Ключевые слова: **синтаксический анализ, структуры составляющих, структуры зависимостей, взвешенное дерево, лексическая сочетаемость, алгоритм CYK**

1. Введение

Синтаксический анализ является одной из наиболее спорных и трудоёмких задач автоматической обработки текста. С одной стороны, это связано с большим количеством синтаксических теорий, многие из которых имеют лишь ограниченное практическое применение. С другой стороны, модули синтаксического анализа востребованы в таких приложениях, как системы извлечения фактов, вопросно-ответные системы, машинный перевод.

Что касается русского языка, то вследствие богатой морфологии задача синтаксического анализа оказывается более сложной по сравнению с английским, для которого сформировался ряд общепризнанных решений (Stanford Parser etc.). Подробный анализ существующих решений для русского языка был проведён в 2011-12 годах в рамках форума «Оценка методов автоматического анализа текстов» [1], который был посвящён синтаксическому анализу. В рамках соревнования был подготовлен золотой стандарт разметки, который до сих пор остаётся единственным эталоном разбора текстов на русском языке. Среди участников были как коммерческие системы – ABBYY Compreno [2], SyntAutom [3], – так и академические разработки (ЭТАП-3 [4]).

Результаты форума и последующие работы показывают, что системы синтаксического анализа (парсеры) используют как подходы, основанные на правилах, так и статистические и гибридные подходы. Кроме того, существующие парсеры предполагают различные виды синтаксического представления – от традиционных грамматик зависимостей и составляющих до, например, грамматики связей (Link Grammar Parser – http://slashzone.ru/parser/). Однако, немногие среди известных решения представляют собой системы, доступ к которым открыт исследователя. Следует упомянуть MaltParser, обученная модель для которого распространяется свободно в том числе и для русского языка, и которую используются большинство некоммерческих разработок [5], [6].

Сборник научных статей XIX Объединенной конференции «Интернет и современное общество» IMS-2015, Санкт-Петербург, 22 - 24 июня 2016 года г.

2. Архитектура синтаксического парсера Hurma

2.1. Реализация и алгоритм

В силу того, что к самому лингвистическому анализатору предъявляются максимально широкие требования для возможности решения широкого спектра прикладных и исследовательских задач, к модулю синтаксического анализа были выдвинуты следующие требования:

1. Синтаксический анализ должен проводиться без предварительного снятия морфологической неоднозначности;
2. Синтаксический анализатор должен строить все варианты синтаксических структур;
3. Алгоритмическая сложность анализа не должна быть экспоненциальной;
4. В результате синтаксического анализа должна получаться единственная структура и при том наиболее вероятная среди всех;
5. Алгоритм должен быть адаптирован к неполноте грамматики и обеспечивать возможность выбора наиболее крупных фрагментов «развалившейся структуры»;
6. Тип выходной структуры – дерево зависимостей.

Требование (1) является необходимым для выполнения требования (2), т.к. при удалении некоторых морфологических вариантов ещё до синтаксического анализа можно потерять «хорошие» кандидаты структур. Требование (2) является необходимым для выполнения требования (4), т.к. если не перебирать все варианты структур, то нельзя гарантировать, что выдаваемая структура будет иметь наивысший вес. Требования (3) необходимо для того, чтобы гарантировать конечное время синтаксического анализа при условии (2). Последние требования (5) и (6) обусловлены тем, что структуры зависимостей лучше подходят для непосредственного использования в последующих уровнях анализа, таких как, например, семантический, а вероятность частичных разборов на ранних стадиях разработки грамматики весьма высока, поэтому должна быть возможность извлечь максимум информации для последующих уровней анализа даже в том случае, если разбор «развалился».

Проанализировав вышеописанные требования мы пришли к выводу, что наиболее удобной базой для нашего анализатора является алгоритм Кока-Янгера-Касами (CYK). Тому есть ряд причин.

Во-первых, алгоритм CYK позволяет без каких-либо доработок использовать морфологически неоднозначный вход – мы просто создаём несколько нетерминальных символов в ячейке таблицы, соответствующей каждому многозначному токену, таким образом, обеспечивая выполнение (1).

Во-вторых, алгоритм CYK является алгоритмом использующий принцип динамического программирования, что позволяет нам перебирать все варианты синтаксической структуры и при этом в явном виде их не перечислять, обеспечивая полиномиальную сложность анализа, и, соответственно, выполнение (2) и (3).

В-третьих, хотя на выполнение (4) выбор алгоритма непосредственно никак не влияет, стоит отметить, что CYK, являясь подвидом табличного парсера, позволяет максимально быстро определять наличие целого разбора, путём просмотра содержимого угловой ячейки треугольной таблицы или, в случае отсутствия такового, путём постепенного спуска выбирать наиболее вероятные фрагменты развалившегося разбора, обеспечивая выполнение (5).

Разумеется, использование алгоритма CYK приводит к определённым особенностям, которые можно было бы счесть недостатками, но в нашей реализации мы попытались превратить их в преимущества. Во-первых, алгоритм CYK предполагает использование грамматики составляющих, что в принципе не очень хорошо ложится требование (6). Во-вторых, грамматика составляющих должна быть представлена в нормальной форме Хомского, т.е. она должна быть бинарной.

Объединяя эти два аспекта в единую концепцию, мы получаем возможность в каждом бинарном правиле проставить синтаксическое отношение между его компонентами (отмечая каждую из двух составляющих каждого правила как вершину или зависимое), что даёт нам простой и эффективный способ получать структуру зависимостей автоматически по структуре составляющих. Кроме того, менее очевидным преимуществом такого подхода является возможность для каждой составляющей определять такой параметр как «длину связи», который в дальнейшем играет немаловажную роль в нашей модели для выбора правильной струкруты.

2.2. Особенности реализации таблицы CYK

Для того чтобы в полной мере воспользоваться преимуществами алгоритма CYK в плане динамического программирования и предотвращения комбинаторного взрыва необходимо разработать специальную модель представления составляющих, которая бы позволяла объединять «одинаковые» составляющие, попавшие в одну ячейку таблицы в разное время и путём применения разных правил в разном порядке. Данная модель будет накладывать определённые ограничения на возможности описания составляющих в правилах. Это связано с тем, что мы в нашем подходе отказываемся от различения внутренних структур составляющих при их сравнении. В нашем понимании, составляющие могут считать идентичными, если они:

1. Покрывают один и тот же диапазон токенов;
2. Имеют один и тот же тип (например VP или NP);

3. Имеют один и тот же набор граммем и помет;
4. Имеют один и тот же токен в качестве вершины.

За выполнение первого условия отвечает сам алгоритм CYK, т.к. координаты ячейки таблицы определяют собственно начало и длину диапазона токенов, таким образом, попавшие в одну ячейку составляющие всегда будут совпадать по диапазону, а попавшие в разные – никогда не совпадут. Соответственно, получается, что если в одну ячейку таблицы попадают составляющие, совпадающие по (2) и (3), то они по определению будут порождать в дальнейшем идентичные синтаксические структуры. Это, в свою очередь означает, что их можно объединить в одну (что, в свою очередь, как раз и обеспечивает полиномиальность алгоритма). Под объединением мы в данном случае понимаем закрепление за одним сочетанием свойств (2) и (3) нескольких возможных разбиений данной составляющей на две других. Кроме того, условие (4) необходимо для того, чтобы гарантировать однозначность зависимого слова для отношения, входящего в составляющую имеющую различные разбитения.

Кроме того, ещё одна важная особенность алгоритма CYK состоит в том, что стандартная схема обхода таблицы для объединения составляющих по правилам предусматривает поочерёдный обход составляющих по увеличению длины. Это гарантирует, что когда составляющая включается в состав родительской, то все возможные парные сочетания её дочерних составляющих уже включены в её состав. Важность данного свойства будет показана в следующем разделе.

2.3. Взвешивание составляющих

В конце предыдущего раздела была высказана фундаментальная мысль, позволяющая нам в дальнейшем эффективно выбирать единственную правильную структуру: к моменту, когда составляющая включается в состав родительской, она уже содержит в себе все возможные разбиения на две других (если, конечно, она не является атомарной). Это означает, что вес составляющей *c* может быть определён как вес максимального её разбиения *p*.

$$w(c) = \max_i w(p_i)$$

Строго говоря, это означает, что мы для каждой составляющей можем игнорировать все разбиения, которые имеют вес меньше максимального (т.е. все кроме одной). Получается, что после завершения анализа, выбирая составляющую с максимальным покрытием и максимальным весом, путём обхода структуры сверху вниз и выбирая при обходе каждый раз разбиения с максимальным весом, мы выбираем из множества всевозможных структур одну единственную, и притом с гарантированно максимальным весом. Таким образом, нам остаётся решить два задачи:

1. Как определять вес разбиения;
2. Как определять вес атомарной составляющей (полученной из терминального символа).

В нашей модели мы выбрали следующий подход. Для разбиений, вес считается как сумма трёх компонентов: веса, соответственно, двух составляющих и вес синтаксической связи между ними:

$$w(p) = w(c_1) + w(c_2) + w(c_1 \rightarrow c_2)$$

Таким образом, главная задача сводится к тому, по каким критериям и как вычислять вес синтаксической связи.

Что касается взвешивания атомарных составляющих, то, строго говоря, методика принципиального значения не имеет: все атомарные составляющие могут получать или один и тот же вес в начале анализа, или получать индивидуальные веса, соответствующие вероятности появления в тексте леммы, словоформы или морфологического тега того токена, который является основой данной составляющей.

2.4. Взвешивание синтаксических связей

Мы плавно перешли в нашем вопросе о выборе единственной правильной структуре от алгоритма и взвешивания составляющих к взвешиванию синтаксических связей, т.е. фактически пришли к той модели, которую очень часто используют в синтаксических анализаторах, использующие грамматики зависимостей [3], [5]. Мы получили такую модель, в которой нам нужно только лишь правильно взвесить синтаксические связи, а выбор единственной верной структуры (в соответствии с весами) будет произведён автоматически. Нам остаётся лишь описать критерии и механизмы преобразования этих критериев в компоненты веса синтаксических связей. На сегодняшний день мы используем следующие критерии:

1. Длина синтаксической связи;
2. Тип синтаксической связи;
3. Лексическая сочетаемость связываемых слов.

По поводу критерия (1) существует обратная зависимость между длиной между словами и вероятностью наличия синтаксической связи между ними [7] – чем меньше расстояние между словами, тем больше вероятность, что ни синтаксически связаны. Критерий (2) позволяет управлять процессом выбора структуры путём присвоения веса каждому типа связи (путём изменения этих весов, добавления или удаления типов связей). Критерий (3) позволяет использовать статистику о лексической сочетаемости слов для того чтобы определить, насколько вероятна связь между этими словами. Таким образом, итоговый вес связи может быть вычислен как:

$$w_r = k_d \times v_d + k_t \times v_t + k_l \times v_l$$

где w_r – это итоговый вес связи, v – это «сырое» значение по критерию, а k – это коэффициент влияния критерия, для критериев d (distance – величина обратная расстоянию между словами), t (type – вес типа синтаксического отношения) и l (lexical – вес лексической сочетаемости связываемых слов).

Получается, что критерий (1) условно можно назвать автоматическим (им нельзя явно управлять), критерий (2) можно назвать экспертным (за счёт него возможна тонкая настройка), а критерий (3) можно назвать статистическим. При этом, тонкая настройка всей системы возможно при помощи экспертного управлениями коэффициентами k, раздельно по каждому критерию.

Именно поэтому мы называем наш подход гибридным, подразумевая, что вес синтаксической структуры определяется и на основе экспертных и на основе статистических данных.

2.5. Оценка лексической сочетаемости

Оценка лексической сочетаемости производится согласно алгоритму, описанному в статье [8]. Для каждой пары слов (x_1, x_2) вычисляется базовая мера сочетаемости f (в нашем случае – по критерию взаимной информации, который показал наилучшие результаты в указанной статье). Затем используется понятие взаимозаменяемости $g\{x_1 \sim x_2\}$, которое оказывается особенно удобным для пар, которые не встречались в обучающем корпусе: грубо говоря, две лексемы взаимозаменяемы настолько, насколько совпадают их контексты. Итоговое значение, характеризующее сочетаемость, описывается через взвешенное среднее:

$$F(x,y) = \frac{f(x,y) + \sum_{y_j \in c(x)} f(x,y_i) g(y_i,y) + \sum_{x_i \in c(y)} f(x_i,y) g(x_i,x)}{1 + \sum_{y_i \in c(x)} g(y_i,y) + \sum_{x_i \in c(y)} g(x_i,x)}.$$

3. Оценка метода

3.1. Особенности тестирования

В силу того, что наш синтаксический анализатор находится в стадии активной разработки, а наша грамматика далека от того, чтобы называться хотя бы относительно полной, мы решили подойти к составлению тестовой коллекции с точки зрения того, что наш синтаксический анализатор уже умеет правильно анализировать. Для данного исследования мы решили ограничиться только простыми предложениями, ограниченными по длине 150 символами и не содержащими никаких пунктуационных знаков, кроме кавычек и знаков конца предложения. По данным критериям был сформирован предварительный корпус объёмом 1000 предложений, выбранных из Открытого Корпуса (ссылка на ОК). Далее эти предложения были обработаны нашим анализатором и из них были отобраны те, которые имели полный разбор, таких оказалось 241 предложение. Далее мы вручную просматривали полученные структуры и экспертно отбирали правильные. Таковых оказалось 74 штуки, из которых и был сформирован наш тестовый корпус. Это означает, что при текущих настройках (включено взвешивание по расстоянию и типам связей) наш анализатор на данном корпусе обеспечивает 100% точность выбора вариантов структуры.

3.2. Наш baseline

Теперь, отключив все механизмы взвешивания синтаксических структур, мы можем определить наш baseline по точности выбора правильной структуры. Мы определяем, какой процент структур и синтаксических связей наш анализатор способен правильно выбрать в качестве единственного варианта не имея никаких критериев, т.е. произвольно. Мы получили для произвольного выбора следующие показатели:

- 5,4% точности при выборе целых структур;
- 76,6% точности при выборе отдельных связей.

Первый показатель означает, что при произвольном выборе одной целой структуры из предложенных грамматикой вариантов полное совпадение со структурой из эталона происходит чуть более чем в пяти случаях из ста. Второй показатель означает, что более трёх синтаксических связей из четырёх, содержащихся в выбранной произвольно синтаксической структуре, являются правильными.

3.3. Оценка различных методов взвешивания

Мы провели серию экспериментов, нацеленных на то, чтобы выяснить, насколько вырастает наша точность относительно baseline'а при активации тех или иных критериев взвешивания синтаксических структур при выборе варианта.

Таблица 1. Точность различных критериев взвешивания

Эксперимент	Точность структур	Точность связей	Прирост	
baseline	0,054	0,766	**Структуры**	**Связи**
Длина связи	0,108	0,838	0,054	0,072

Вес связи	0,837	0,979	0,783	0,213
Лекс. соч. по методу [8]	0,081	0,791	0,027	0,025
Лекс. соч. по w2v	0,122	0,806	0,068	0,040

Точность по целым структурам у baseline'а крайне низкая, чуть выше 5%, что означает, что шансов на произвольный выбор правильной структуры практически нет и для того, чтобы выбирать правильную структуру обязательно нужно подключать дополнительные критерии. Такой, казалось бы логичный критерий, как длина синтаксической связи, на практике, показывает себя крайне малоэффективным при самостоятельном использовании – точность с ним не доходит до 11%, прибавляя к baseline'у всего лишь 5,4%. Вес синтаксической связи, напротив, показывает себя как наиболее эффективный критерий, обеспечивая точность почти в 84%, что не удивительно, т.к. это основной критерий, которым может управлять разработчик грамматики для максимизации точности выбора синтаксической структуры. Критерий лексической сочетаемости проверялся на различных биграммных моделях, по методу, описанному в статье [8] и при помощи инструмента Word2vec. Тем не менее, модель, полученная при помощи Word2Vec, по приросту превысила качество модели на длинах синтаксических связей. Более того, причины такого результата следует анализировать относительно того, насколько в принципе модель лексической сочетаемости могла быть применима для исправления тех ошибок выбора синтаксической структуры, которые имели место в корпусе.

3.4. Выводы

В рамках нашего исследования мы провели ручную верификацию почти двух с половиной сотен предложений. Учитывая, что лишь треть из них расценивалась экспертами как полностью правильные, остальные приходилось анализировать с точки зрения ошибок синтаксических структур. Мы обнаружили несколько основных типов ошибок:

1. Правильный вариант синтаксической связи не описан в грамматике – таких случаев примерно треть от всех встретившихся ошибок;
2. Ошибки при присоединении предложных групп – самая многочисленная категория, примерно половина всех встретившихся у нас ошибок;
3. Ошибки в выборе типа синтаксического отношения, не приводящие к модификации структуры – например, когда прямой объект маркируется как подлежащее, а подлежащие как прямой объект;
4. Ошибки в выборе типа синтаксического отношения, приводящие к изменению структуры – например, в предложении «банкноты прошлого года» проставляются два генитивных отношения (слово «прошлое» считается существительным) вместо атрибутивного и генитивного (слово «прошлое» -- прилагательное, определение к слову «год»).

На основе подобранного материала мы провели исследование возможного использования различных критериев для снятия структурной омонимии. Следует отметить, что никакой критерий не может нивелировать ошибки типа (1) – их можно устранить только путём повышения полноты грамматики. Ошибки типа (3) и (4) могут быть исправлены как при помощи изменения модели весов синтаксических связей, так и при помощи семантического анализа, что выходит за рамки данного исследования. Наиболее же частотные ошибки типа (2) напрямую не могут быть исправлены при помощи критерия лексической сочетаемости, т.к. под лексической сочетаемостью мы понимали бинарное отношение, а сочетания с предложными группами состоят из трёх слов, что требует адаптации статистической модели. Это планируется в будущем.

Литература

[1] Толдова, С.Ю. Оценка методов автоматического анализа текста 2011–2012: синтаксические парсеры русского языка / Толдова С. Ю., Соколова Е. Г., Астафьева И., Гарейшина А., Королева А., Привознов Д., Сидорова Е., Тупикина Л., Ляшевская О. Н. // Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialog 2012" [Komp'yuternaya Lingvistika i Intellektual'nye Tekhnologii: Trudy Mezhdunarodnoy Konferentsii "Dialog 2012"]. – M., 2012.

[2] Anisimovich, K.V. Syntactic and semantic parser based on ABBYY Compreno linguistic technologies / Anisimovich K. V., Druzhkin K. Ju., Minlos F. R., Petrova M. A., Selegey V. P., Zuev K. A. // Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialog 2012" [Komp'yuternaya Lingvistika i Intellektual'nye Tekhnologii: Trudy Mezhdunarodnoy Konferentsii "Dialog 2012"]. – M., 2012.

[3] Iomdin L.L. ETAP parser: state of the art / Iomdin L., Petrochenkov V., Sizov V., Tsinman L. // Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialog 2012" [Komp'yuternaya Lingvistika i Intellektual'nye Tekhnologii: Trudy Mezhdunarodnoy Konferentsii "Dialog 2012"]. – M., 2012.

[4] Antonova, A., Misyurev, A. Russian dependency parser SyntAutom at the DIALOGUE-2012 parser evaluation task // Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference

"Dialog 2012" [Komp'yuternaya Lingvistika i Intellektual'nye Tekhnologii: Trudy Mezhdunarodnoy Konferentsii "Dialog 2012"]. – M., 2012.
[5] Droganova K. Building Dependency Parsing Model for Russian with MaltParser and MyStem Tagset // Proceedings of the AINL-ISMW FRUCT 2015. – СПб, 2015.
[6] Shelmanov A. O., Smirnov I. V. Methods for semantic role labeling of Russian texts // Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialog 2014" [Komp'yuternaya Lingvistika i Intellektual'nye Tekhnologii: Trudy Mezhdunarodnoy Konferentsii "Dialog 2014"]. – M., 2014.
[7] Skatov D. Parsing Russian: a hybrid approach / Skatov D., Liverko S., Okatiev V., Strebkov D. // Proceedings of the 4th Biennial International Workshop on Balto-Slavic Natural Language Processing. – Sofia, Bulgaria, 2013.
[8] Букия, Г.Т. Корпусная оценка степени близости единиц в лексических конструкциях / Букия, Г.Т., Протопопова, Е.В., Митрофанова, О.А. // Структурная и прикладная лингвистика. Межвузовский сборник. №11. Под. ред. А.С. Герда. – СПб, 2015.

Yet another Time on Approaches to Structural Disambiguation: Single Structure Selection in Hurma Parser

A. Popov [1], E. Protopopova [2], G. Bukia [3]

[1] SPSU, Hurma Project

[2] SPSU, Hurma Project

[3] SPSU,

The goal of our Hurma Project is to develop a research platform for NLP, primarily for Russian. Our platform implements a level-by-level natural text analysis including syntactic level.

The capability of selecting a single syntactic structure in the end of parsing is required by many applied tasks, relying on syntactic information: Treebank creation, Information Extraction, Machine Translation and so on.

In our research we evaluate different criteria, which provide means for successful selection of single syntactic structure. We suggest a hybrid approach to weighting syntactic structures that incorporates manually set information and statistically gathered information about lexical collocation of syntactically connected words.

Keywords: syntax parsing; phrase structure; dependency structure; weighted graph; lexical collocation; CYK-algorithm;

Метаданные к статье

Название стати рус. (вставляется из текста статьи)	Ещё раз о способах снятия структурной омонимии: выбор единственной структуры в парсере Hurma
Название стати англ. (вставляется из текста статьи)	Yet another Time on Approaches to Structural Disambiguation: Single Structure Selection in Hurma Parser
Фамилия первого автора рус (вставляется из текста статьи).	Попов
Имя первого автора рус. (заполняется автором вручную)	Андрей

Отчество первого автор рус. (заполняется автором вручную)	Михайлович
Инициалы первого автора рус. (вставляется из текста статьи)	А.М.
Фамилия первого автора лат. (вставляется из текста статьи)	Popov
Имя первого автора лат.	Andrey
Отчество первого автора лат.	Michailovich
Инициалы первого автора лат. (вставляется из текста статьи)	A.
Ученая степень первого автора рус. (заполняется автором вручную)	Место для ввода текста.
Ученая степень первого автора лат. (заполняется автором вручную)	Место для ввода текста.
Ученое звание первого автора рус. (заполняется автором вручную)	Место для ввода текста.
Ученое звание первого автора лат. (заполняется автором вручную)	Место для ввода текста.
Организация 1 первого автора рус. (вставляется из текста статьи)	СПбГУ
Организация 2 первого автора рус. (вставляется из текста статьи)	проект Хурма
Организация 1 первого автора лат. (вставляется из текста статьи)	SPSU
Организация 2 первого автора лат. (вставляется из текста статьи)	Hurma Project
E-mail первого автора (вставляется из текста статьи)	hedgeonline@gmail.com
Профиль РИНЦ (заполняется автором вручную)	Профиль РИНЦ
РИНЦ ID (заполняется автором вручную)	РИНЦ ID
РИНЦ SPIN (заполняется автором вручную)	РИНЦ SPIN
Страна рус. (заполняется автором вручную)	Российская Федерация
Страна лат. (заполняется автором вручную)	Russian Federation
Фамилия второго автора рус. (вставляется из текста статьи)	Протопопова
Имя второго автора рус. (заполняется автором вручную)	Екатерина
Отчество второго автора рус. (заполняется автором вручную)	Владимировна
Инициалы второго автора рус. (вставляется из текста статьи)	Е.В.
Фамилия второго автора лат (вставляется из текста статьи)	Protopopova
Имя второго автора лат (заполняется автором вручную)	Ekaterina
Отчество второго автора лат. (заполняется автором вручную)	Vladimirovna

Инициалы второго автора лат. (вставляется из текста статьи)	E.V.
Ученая степень второго автора рус. (заполняется автором вручную)	Место для ввода текста.
Ученая степень второго автора лат. (заполняется автором вручную)	Место для ввода текста.
Ученое звание второго автора рус. (заполняется автором вручную)	Место для ввода текста.
Ученое звание второго автора лат. (заполняется автором вручную)	Место для ввода текста.
Организация 1 второго автора рус. (вставляется из текста статьи)	СПбГУ
Организация 2 второго автора рус. (вставляется из текста статьи)	проект Хурма
Организация 1 второго автора лат. (вставляется из текста статьи)	SPSU
Организация 2 второго автора лат. (вставляется из текста статьи)	Hurma Project
e-mail второго автора (вставляется из текста статьи)	protoev@gmail.com
Профиль РИНЦ (заполняется автором вручную)	Профиль РИНЦ
РИНЦ ID (заполняется автором вручную)	РИНЦ ID
РИНЦ SPIN (заполняется автором вручную)	РИНЦ SPIN
Страна рус. (заполняется автором	Российская Федерация
Страна лат. (заполняется автором вручную)	Russian Federation
Фамилия третьего автора рус. (вставляется из текста статьи)	**Ошибка! Источник ссылки не найден.**
Имя третьего автора рус. (заполняется автором вручную)	Григорий
Отчество третьего автора рус. (заполняется автором вручную)	Теймуразович
Инициалы третьего автора рус. (вставляется из текста статьи)	Г.Т.
Фамилия третьего автора лат. (вставляется из текста статьи)	Bukia
Имя третьего автора лат. (заполняется автором вручную)	Grigoriy
Отчество третьего автора лат. (заполняется автором вручную)	Teymurazovich
Инициалы третьего автора лат. (вставляется из текста статьи)	G.T.
Ученая степень третьего автора рус. (заполняется автором вручную)	Место для ввода текста.
Ученая степень третьего автора лат. (заполняется автором вручную)	Место для ввода текста.
Ученое звание третьего автора рус. (заполняется автором вручную)	Место для ввода текста.
Ученое звание третьего автора лат. (заполняется автором вручную)	Место для ввода текста.

Организация 1 третьего автора рус. (вставляется из текста статьи)	СПбГУ
Организация 2 третьего автора рус. (вставляется из текста статьи)	
Организация 1 третьего автора лат. (вставляется из текста статьи)	SPSU
Организация 2 третьего автора лат. (вставляется из текста статьи)	
e-mail третьего автора. (вставляется из текста статьи)	gregorybookia@yandex.ru
Профиль РИНЦ (заполняется автором вручную)	Профиль РИНЦ
РИНЦ ID (заполняется автором вручную)	РИНЦ ID
РИНЦ SPIN (заполняется автором вручную)	РИНЦ SPIN
Страна рус (заполняется автором вручную)	Российская Федерация
Страна лат (заполняется автором вручную)	Russian Federation
УДК (заполняется автором вручную)	УДК
ББК (заполняется автором вручную)	ББК
ГРНТИ (заполняется автором вручную)	ГРНТИ
Секция (заполняется автором вручную)	Выберите секцию.
Аннотация рус. (вставляется из текста статьи)	Целью проекта Хурма является создание исследовательской платформы для анализа текстов на естественном языке, в первую очередь на русском. Данная платформа предполагает анализ текстов на различных языковых уровнях, в том числе и на синтаксическом. Возможность выбора единственной синтаксической структуры в результате
Аннотация лат. (вставляется из текста статьи)	The goal of our Hurma Project is to develop a research platform for NLP, primarily for Russian. Our platform implements a level-by-level natural text analysis including syntactic level. The capability of selecting a single syntactic structure in the end of parsing is required by many applied tasks, relying on syntactic information: Treebank creation, Information Extraction, Machine Translation and so on. In our research we evaluate different criteria, which provide means for successful selection of single syntactic
Ключевые слова рус, разделенные точкой с запятой (вставляется из текста статьи)	синтаксический анализ, структуры составляющих, структуры зависимостей, взвешенное дерево, лексическая
Ключевые слова лат, разделенные точкой с запятой (вставляется из текста статьи)	syntax parsing; phrase structure; dependency structure; weighted graph; lexical collocation; CYK-algorithm;

Список литературы (вставляется из текста статьи)	[1] Толдова, С.Ю. Оценка методов автоматического анализа текста 2011–2012: синтаксические парсеры русского языка / Толдова С. Ю., Соколова Е. Г., Астафьева И., Гарейшина А., Королева А., Привознов Д., Сидорова Е., Тупикина Л., Ляшевская О. Н. // Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference “Dialog 2012” [Komp’yuternaya Lingvistika i Intellektual’nye Tekhnologii: Trudy Mezhdunarodnoy Konferentsii “Dialog 2012”]. – M., 2012. [2] Anisimovich, K.V. Syntactic and semantic parser based on ABBYY Compreno linguistic technologies / Anisimovich K. V., Druzhkin K. Ju., Minlos F. R., Petrova M. A., Selegey V. P., Zuev K. A. // Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference “Dialog 2012” [Komp’yuternaya
Номера страниц (заполняется редакцией)	Страницы
Ссылка на текст статьи в openbooks (заполняется редакцией)	Ссылка на текст статьи (в openbooks) (заполняется редакцией)