УДК 004.912 **DOI** 10.15622/ia.22.6.3

А.Ю. Полетаев, И.В. Парамонов, Е.И. Бойчук

АЛГОРИТМ ПОСТРОЕНИЯ ДЕРЕВА СИНТАКСИЧЕСКИХ ЕДИНИЦ РУССКОЯЗЫЧНОГО ПРЕДЛОЖЕНИЯ ПО ДЕРЕВУ СИНТАКСИЧЕСКИХ СВЯЗЕЙ

Полетаев А.Ю., Парамонов И.В., Бойчук Е.И. **Алгоритм построения дерева синтаксических единиц русскоязычного предложения по дереву синтаксических связей.**

Аннотация. Автоматический синтаксический анализ предложения – одна из важных задач компьютерной лингвистики. В настоящее время для русского языка отсутствуют общедоступные и пригодные для практического применения анализаторы синтаксической структуры. Создание таких анализаторов «с нуля» требует составления корпуса деревьев, размеченного в соответствии с заданной формальной грамматикой, что представляет собой крайне трудоёмкую задачу. Однако, поскольку для русского языка существует несколько анализаторов синтаксических связей, представляется полезным использовать результаты их работы для анализа синтаксической структуры предложений. В настоящей работе предлагается алгоритм, позволяющий построить дерево синтаксических единиц русскоязычного предложения по данному дереву синтаксических связей. Алгоритм использует грамматику, сформулированную в соответствии с классическим справочником Д.Э. Розенталя. Приведены результаты экспериментов по оценке качества работы предложенного алгоритма на корпусе из 300 предложений на русском языке. 200 предложений были выбраны из вышеупомянутого справочника и 100 из открытого корпуса публицистических текстов OpenCorpora. В ходе экспериментов предложения подавались на вход анализаторов из состава библиотек Stanza, SpaCy и Natasha, после чего полученные деревья синтаксических связей обрабатывались предложенным алгоритмом. Полученные в результате обработки деревья синтаксических единиц сравнивались с размеченными вручную экспертами-филологами. Наилучшее качество было получено при использовании анализатора синтаксических связей из библиотеки Stanza: F_1-мера построения синтаксических единиц составила 0.85, а точность определения членов предложения – 0.93, чего должно быть достаточно для решения многих практических задач в таких областях, как извлечение событий, информационный поиск, анализ тональности.

Ключевые слова: компьютерная лингвистика, обработка естественного языка, синтаксический анализ, дерево синтаксических единиц, дерево синтаксических связей, формальная грамматика.

1. Введение.

Для решения некоторых задач обработки естественного языка, например, извлечения событий, информационного

поиска, анализа тональности, требуются методы автоматического анализа предложений. Одна из важнейших задач анализа предложения – синтаксический анализ, включающий в себя как анализ синтаксических связей (dependency parsing), так и анализ синтаксической структуры (constituency parsing) [1].

Синтаксические связи между словами предложения могут быть представлены в виде дерева синтаксических связей (дерева зависимостей, дерева подчинения, dependency tree) [2]. Пример дерева синтаксических связей для предложения на русском языке приведён на рисунке 1.

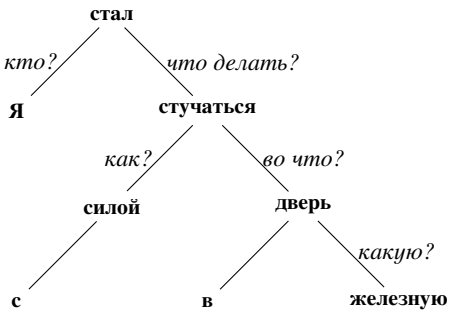

Рис. 1. Дерево синтаксических связей предложения
Я стал с силой стучаться в железную дверь

Синтаксическая структура предложения может быть представлена в виде дерева синтаксических единиц (системы синтаксических групп, дерева непосредственно составляющих, constituency tree), каждый узел которого – синтаксическая единица [3, 4], а потомки этого узла – формирующие её меньшие синтаксические единицы. Корнем дерева синтаксических единиц является всё предложение в целом как наиболее крупная синтаксическая единица, а листьями – отдельные слова. Пример дерева синтаксических единиц для предложения на русском языке приведён на рисунке 2.

Поскольку дерево синтаксических единиц описывает не только отдельные слова, но и то, как они объединяются в более крупные единицы синтаксиса, а это объединение в естественном языке происходит в соответствии с его грамматикой, для построения дерева синтаксических единиц необходима формализация этой грамматики [1]. Нужно отметить, что для одного и того же предложения по разным формальным грамматикам будут построены разные деревья синтаксических единиц [2].

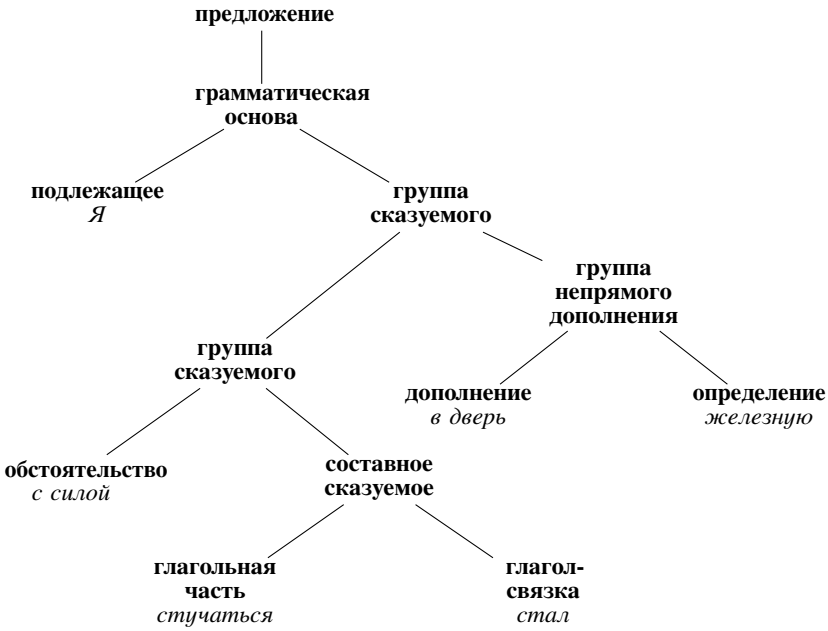

Рис. 2. Дерево синтаксических единиц предложения
Я стал с силой стучаться в железную дверь

Лингвисты изучают естественный язык с помощью анализа синтаксических единиц (immediate constituent analysis) как минимум с середины XX века [5]. Данный метод используется как в исследованиях, объектом которых является естественный язык сам по себе, так и в исследованиях, использующих тексты на естественном языке как источник информации о других процессах и явлениях [6 – 10]. Также информация о синтаксической структуре предложения может не анализироваться исследователями напрямую, а использоваться как источник данных для алгоритмов машинного обучения [11].

Важно отметить, что применимость дерева синтаксических единиц для решениях прикладных задач связана с формальной грамматикой, в соответствии с которой оно построено: например, если грамматика не разделяет сложносочинённые и сложноподчинённые предложения, то исследование, для которого это разделение имеет значение, провести просто не получится. С другой стороны, чем сложнее грамматика, тем сложнее на её основе проводить автоматический анализ и построение деревьев синтаксических единиц.

В настоящий момент для русского языка отсутствуют общедоступные и пригодные для практического применения анализаторы синтаксической структуры. Создание алгоритмов для такого анализа требует составления корпуса деревьев (treebank), размеченного в соответствии с формальной грамматикой, что представляет собой крайне трудоёмкую задачу – например, создание Penn Treebank потребовало 8 лет работы [12]. Однако, поскольку для русского языка доступно несколько парсеров синтаксических связей, представляется полезным использовать результаты анализа синтаксических связей слов в предложении для анализа его синтаксической структуры.

Цель данной работы – предложить алгоритм, позволяющий построить дерево синтаксических единиц русскоязычного предложения, соответствующее заданной грамматике, по данному дереву синтаксических связей, а также оценить эффективность предложенного алгоритма. Формальная грамматика для работы была сформулирована в соответствии с классической работой Д.Э. Розенталя, И.Б. Голуб и М.А. Теленковой [13] с некоторыми упрощениями. Эта модель языка с одной стороны достаточно точна, с другой – проста, но может при необходимости быть уточнена и дополнена для решения различных прикладных задач.

Оставшаяся часть работы организована следующим образом. В разделе 2 приведён обзор связанных работ. Раздел 3 посвящён используемой в работе формальной грамматике. В разделе 4 описан предлагаемый алгоритм построения деревьев синтаксических единиц. Раздел 5 содержит результаты экспериментов по оценке качества построения деревьев синтаксических единиц на размеченных наборах предложений при использовании различных анализаторов синтаксических связей. В заключении подведены итоги работы.

2. Связанные работы. Использование деревьев синтаксических единиц для представления синтаксической структуры предложения было предложено Н. Хомским [14, 15]. Им также была обоснована значимость используемой для синтаксического анализа грамматики естественного языка и сформулированы основные принципы построения синтаксических деревьев.

Большая часть исследований деревьев синтаксических единиц выполнены для английского языка, в частности существуют достаточно подробно проработанные грамматики, например, HPSG (Head-Driven Phrase Structure Grammar), которая учитывает, кроме синтаксиса, некоторые элементы семантики предложений [16]. Однако из-за своей сложности такие грамматики достаточно сложны для использования,

поэтому, например, крупнейший набор деревьев синтаксических единиц Penn Treebank размечен в соответствии с достаточно сильно упрощённой грамматикой [17, 18].

В работах [19 – 21] на примере английского языка рассматриваются основные современные подходы к автоматическому построению деревьев синтаксических единиц. При использовании различных нейронных сетей на некоторых корпусах, например, Penn Treebank, возможно достижение крайне высокого качества построения деревьев синтаксических единиц – F_1-мера выше 0.95. Однако в данных работах также показано, что качество синтаксического анализа достаточно сильно зависит от используемого корпуса, и, например, модель BERT, с помощью которой на Penn Treebank была достигнута F_1-мера 0.96, при построении деревьев синтаксических единиц на корпусе предложений из художественных текстов показала гораздо более низкое качество, с F_1-мерой всего 0.86 [21]. В работе [20] также отмечается, что качество построения деревьев синтаксических единиц может достаточно сильно отличаться в зависимости от языка: для китайского языка F_1-мера в схожих условиях оказывается на 5–7 %% ниже, чем для английского.

Первые попытки применить наработки Хомского для исследования русского языка относятся ещё к 60-м годам XX века [22], наиболее развитая грамматика синтаксических единиц, опирающаяся на накопленные лингвистами знания, была разработана и описана в работах А.В. Гладкого [23, 24]. Однако, из-за несовершенства вычислительной техники того периода и отсутствия достаточных лингвистических ресурсов, а также сложности самой грамматики, не было разработано автоматических синтаксических анализаторов, строящих деревья синтаксических единиц в соответствии с этой грамматикой.

В работах [25, 26] описывается созданный И.А. Кагировым и А.Б. Леонтьевой анализатор, строящий деревья синтаксических единиц на основе частеречной разметки, морфологической информации и анализа синтаксических связей. К достоинствам этих работ следует отнести предложенную формальную грамматику составляющих, достаточно хорошо соответствующую теории синтаксиса русского языка, но, вместе с тем, достаточно простую для автоматического анализа. Главным недостатком является то, что эксперименты проводились только для простых предложений.

Также необходимо отметить проект «Диалинг» и связанные с ним исследования [27]. Несмотря на то, что предлагаемая его авторами система клауз относится скорее к анализу семантики, опыт построения систем автоматического анализа на её основе показывает, что в настоящий момент

качество доступных инструментов автоматического анализа отдельных слов и связей между ними уже достаточно высоко, чтобы результаты их работы можно было использовать в качестве входных данных для автоматического анализа сложных предложений [28, 29].

3. Используемая грамматика. Крайне важен выбор формальной грамматики, описывающей синтаксическую структуру предложений. С одной стороны, она должна быть достаточно подробной, чтобы дерево синтаксических единиц предоставляло как можно больше информации. С другой стороны, чем сложнее грамматика, тем сложнее точно построить дерево на её основе, возрастает число ошибок и снижается надёжность предоставляемой деревом информации.

Для целей данного исследования была формализована грамматика русскоязычных предложений из классической работы [13]. Далее приводится описание полученного результата, а также упрощений, допущенных для лучшей формализации.

Для того, чтобы привести в рамках данной статьи полное описание полученной формальной грамматики, используется нескольких принципов. Во-первых, поскольку в русском языке нет строгого порядка слов, и, например, определение может находиться как перед определяемым словом, так и после, большая часть правил должна содержать последовательности как для прямого, так и для обратного порядка слов. Чтобы сократить запись, условимся, что порядок символов в последовательностях может быть произвольным, если в описании соответствующего правила не указано иное. Во-вторых, чтобы не вводить дополнительные фиктивные типы символов для обеспечения порядка применения продукций, необходимого для обеспечения однозначности разбора, условимся, что продукции приводятся в порядке приоритета их применения при нисходящем синтаксическом анализе, от наиболее приоритетного к наименее приоритетному.

Чтобы обеспечить однозначность разбора синтаксической структуры вне зависимости от порядка слов, было принято, что синтаксические единицы, обозначающие свойства явлений, в дереве должны находиться ближе к единицам, называющим эти явления, чем единицы, называющие другие явления. Например, вначале в одну синтаксическую единицу должны объединяться подлежащее и связанное с ним определение, и только затем будет выделена синтаксическая единица дополнения, связанного с подлежащим:

(((Лучший студент) группы) (сдал экзамен)).

3.1. Общая структура предложения. Предложение в целом состоит из грамматической основы и знака препинания, стоящего после неё. Знак препинания не обязателен, поскольку он часто опускается в сетевых текстах [30]:

предложение ::= грамматическая-основа [знак-препинания]

Грамматическая основа предложения состоит из группы подлежащего и группы сказуемого, если предложение двусоставное; только из группы подлежащего или группы сказуемого, если предложение односоставное. Грамматическая основа сложносочинённого предложения выражается специальной синтаксической единицей, объединяющей две и более грамматических основы с помощью сочинительного союза. В расчленённом сложноподчинённом предложении придаточное предложение относится ко всему главному предложению в целом, поэтому грамматическая основа такого предложения также будет состоять из двух отдельных грамматических основ, связанных подчинительным союзом. Любая грамматическая основа может предваряться вводной конструкцией, выраженной отдельной синтаксической единицей.

Для упрощения грамматики не выделяется отдельный тип бессоюзных сложных предложений, так как, согласно [13], все бессоюзные сложные предложения могут быть соотнесены либо со сложносочинёнными, либо со сложноподчинёнными предложениями, а знаки препинания в них, соответственно, с сочинительными либо подчинительными союзами.

Описанное выше выражаются следующими формальными правилами:

грамматическая-основа ::= вводная-конструкция грамматическая-основа
| сложносочинённое-предложение
| расчленённое-сложноподчинённое-предложение
| группа-подлежащего группа-сказуемого
| группа-сказуемого
| группа-подлежащего

сложносочинённое-предложение ::= грамматическая-основа
грамматическая-основа {грамматическая-основа} сочинительный-союз

расчленённое-сложноподчинённое-предложение ::= грамматическая-основа
подчинительный-союз грамматическая-основа

3.2. Группа подлежащего. Группа подлежащего может включать в себя, кроме самого подлежащего, определение, поясняющее называемый подлежащим предмет, например, *хороший ученик*; непрямое дополнение,

называющее предмет, связанный с подлежащим, например, *ученик школы*; а также прямое дополнение и обстоятельство, если подлежащее называет действие, например, *быстро сдать экзамен оказалось сложно* или *покраска забора затянулась*. Смысл подлежащего может уточняться присоединённым к нему придаточным предложением. Как и синтаксические группы любых других членов предложения, группа подлежащего может включать в себя частицу, предваряться вводной конструкцией, а также быть заключённой в кавычки. Структура группы подлежащего формально описывается следующим правилом:

```
группа-подлежащего ::= вводная-конструкция группа-подлежащего
    | однородные-подлежащие
    | группа-подлежащего подчинительный-союз грамматическая-основа
    | группа-подлежащего группа-непрямого-дополнения
    | группа-подлежащего группа-обстоятельства
    | группа-подлежащего группа-определения
    | группа-подлежащего группа-прямого-дополнения
    | группа-подлежащего частица
    | кавычка группа-подлежащего кавычка
    | подлежащее
```

Однородные подлежащие, как и другие однородные члены предложения, формируют синтаксическую группу, включающую в себя несколько групп подлежащих и соединяющий их сочинительный союз. Как и в случае со сложносочинёнными предложениями, для упрощения грамматики знаки препинания, замещающие сочинительные союзы, приравниваются к сочинительным союзам: *стол, стул и кровать = стол, и стул, и кровать*. Далее в тексте опускаются правила для синтаксических групп однородных членов, строящиеся аналогично группе однородных подлежащих. Структура синтаксической группы однородных подлежащих формально описывается следующим правилом:

```
однородные-подлежащие ::= группа-подлежащего группа-подлежащего
    {группа-подлежащего} сочинительный-союз
```

3.3. Группа сказуемого. Сказуемое может быть как простым, так и составным, состоящим из главной части – глагольной или именной – и глагола-связки. В случае глагольной главной части говорят о составном глагольном сказуемом: *начать рисовать*; в случае именной – о составном именно сказуемом: *быть студентом*. Как глагольная, так и именная части могут состоять из нескольких однородных членов: *начали петь и танцевать*, *был важен и уместен*. Глагольная главная часть может быть составной: *могу согласиться уйти*.

Для более формального отделения составных сказуемых от других синтаксических единиц был ограничен список глаголов, которые могут быть связками.

– Модальные глаголы: *быть*, *стать*, *оказаться*, *начать*, *продолжить*, *мочь*, *позволить*, *сметь*, *устать*, *перестать*, *являться*, *пытаться*.

– Глаголы чувственного восприятия и близкие к ним по семантике: *любить*, *видеть*, *смотреть*, *слышать*, *чувствовать*, *нюхать*, *трогать*, *считать*, *представлять*.

Фразы с глаголами, не входящими в данный список, в рамках данной грамматики не являются составными сказуемыми. Например, схожая по строению с составным сказуемым фраза *пожимать плечами* считается сказуемым *пожимать* и дополнением *плечами*. Связка может быть нулевой, когда на её месте подразумевается соответствующий модальный глагол: *Я (есть) студент*. Глагол-связка также может быть составной: *хочу быть генералом*.

Группа сказуемого может, кроме сказуемого, включать прямые дополнения, как правило, называющие объект действия, такие как *сдаёт экзамен*; обстоятельства, характеризующие свойства действия, например, *сдаёт легко*; непрямые дополнения, называющие объекты, связанные с действием, например, *вышел из дома*. Также в состав группы сказуемого может входить определение, сообщающее о свойстве именной части составного сказуемого, такое как *был лучшим учеником*.

Структура группы сказуемого описывается следующим набором формальных правил:

```
группа-сказуемого ::= вводная-конструкция группа-сказуемого
        | однородные-сказуемые
        | группа-сказуемого подчинительный-союз грамматическая-основа
        | группа-сказуемого группа-непрямого-дополнения
        | группа-сказуемого группа-определения
        | группа-сказуемого группа-обстоятельства
        | группа-сказуемого группа-прямого-дополнения
        | группа-сказуемого частица
        | составное-сказуемое
        | сказуемое
составное-сказуемое ::= ( глагольная-часть | именная-часть ) [ глагол-связка ]
глагольная-часть ::= однородные-глагольные-части
        | составное-сказуемое
        | глагол-начальной-формы
именная-часть ::= однородные-именные-части
        | предлог именная-часть
```

| именная-часть частица
| кавычка именная-часть кавычка
| именная-часть-речи
глагол-связка ::= составное-сказуемое
| модальный-глагол

Важное отличие синтаксической группы однородных сказуемых от синтаксических групп других однородных членов – к однородным сказуемым может относиться общее непрямое дополнение, как в предложении *Я живу и работаю в Ярославле*. Формально это описывается правилом:

однородные-сказуемые ::= группа-непрямого-дополнения однородные-сказуемые
| группа-сказуемого группа-сказуемого
{группа-сказуемого} сочинительный-союз

3.4. Группа дополнения. Синтаксические группы прямых и непрямых дополнений грамматикой описываются одинаково. Поскольку дополнение, как и подлежащее, называет явление, то и синтаксическая группа дополнения описывается почти так же, как и группа подлежащего. Отличий всего два: во-первых, в составе группы дополнения может быть предлог, во-вторых, дополнение может быть выражено синтаксической группой прямой речи. Формально это выражается следующими правилами:

группа-прямого-дополнения ::= группа-дополнения
группа-непрямого-дополнения ::= группа-дополнения
группа-дополнения ::= вводная-конструкция группа-дополнения
| прямая-речь
| однородные-дополнения
| группа-дополнения подчинительный-союз грамматическая-основа
| группа-дополнения группа-непрямого-дополнения
| предлог группа-дополнения
| группа-дополнения группа-обстоятельства
| группа-дополнения группа-определения
| группа-дополнения группа-прямого-дополнения
| группа-дополнения частица
| кавычка группа-дополнения кавычка
| дополнение

Важная особенность синтаксической группы однородных дополнений: к однородным дополнениям может относиться общий предлог, например, *рыбу ловят в озёрах и прудах*. Формально это можно

описать правилом:

однородные-дополнения ::= предлог однородные-дополнения
| группа-дополнения группа-дополнения
{группа-дополнения} сочинительный-союз

Прямая речь всегда находится в группе дополнения, поскольку она может быть заменена на соответствующее изъяснительное придаточное. Её синтаксическая группа состоит из отдельного предложения прямой речи и отделяющих его от главного предложения знаков препинания. Формально её структура описывается следующим правилом:

прямая-речь ::= предложение знаки-препинания

3.5. Группа определения. В состав группы определения, кроме самого определения, могут входить другие определения: *самый лучший*; прямые и непрямые дополнения: *читавший Пушкина*, *сбежавший из дома*; обстоятельства: *сдающие сейчас*. Структура синтаксической группы определения описывается следующим формальным правилом:

группа-определения ::= вводная-конструкция группа-определения
| однородные-определения
| группа-определения подчинительный-союз грамматическая-основа
| группа-определения группа-непрямого-дополнения
| предлог группа-определения
| группа-определения группа-обстоятельства
| группа-определения группа-определения
| группа-определения группа-прямого-дополнения
| группа-определения частица
| определение

В рамках данной грамматики приложения не выделяются в отдельный вид членов предложения, а относятся к определениям. В словосочетаниях, в которых грань между главным и зависимым словом размыта, например, *инженер-конструктор*, *река Волга*, главным словом для упрощения всегда считается идущее первым: *инженер-конструктор*, *река Волга*, *Волга-река*.

Для отделения определений от определительных дополнений используется упрощённый критерий: слово считается определением, если оно может быть заменено на прилагательное без потери смысла. Например, *двор школы* = *школьный двор*, *колесо велосипеда* = *велосипедное колесо*, но *шип дорожки* ≠ *дорожный шип*.

3.6. Группа обстоятельства. В состав синтаксической группы обстоятельства, кроме непосредственно обстоятельства, могут входить прямые и непрямые дополнения: *слушая музыку*, *мешая преподавателю*; определения: *работал целый вечер*; другие обстоятельства: *слушая невнимательно*. Формально структура синтаксической группы обстоятельства описывается правилом:

```
группа-обстоятельства ::= вводная-конструкция группа-обстоятельства
    | однородные-обстоятельства
    | группа-обстоятельства подчинительный-союз грамматическая-основа
    | группа-обстоятельства группа-непрямого-дополнения
    | предлог группа-обстоятельства
    | группа-обстоятельства группа-обстоятельства
    | группа-обстоятельства группа-прямого-дополнения
    | группа-обстоятельства частица
    | группа-обстоятельства группа-определения
    | обстоятельство
```

Для отделения обстоятельств от обстоятельственных дополнений используется несколько упрощённых правил:

– Обстоятельством считается любое уточнение, относящееся к вопросам времени: *бегать по утрам*, *уехать после сессии*.

– Обстоятельством считается любая группа с одним характерных для обстоятельств предлогов: *вопреки*, *вплоть до*, *в случае*, *в качестве*, *в связи с*, *за счёт*, *из-за*, *назло*, *несмотря на*, *после*, *вследствие*, *с помощью*, *при*.

– Обстоятельством считается любое уточнение, относящееся к абстрактным понятиям, отвечающим на вопрос «как?»: *ходить боком*, *жить отшельником*.

– Во всех остальных случаях уточняющее слово считается дополнением.

4. Алгоритм. Предлагаемый алгоритм позволяет построить дерево синтаксических единиц русскоязычного предложения, соответствующее описанной выше грамматике, на основе его дерева синтаксических связей.

На вход алгоритма подаётся дерево синтаксических связей, узлами которого являются отдельные слова и знаки препинания предложения (токены), а дугами – синтаксические связи между ними. Для каждой дуги известен её тип связи, размеченный в соответствии с Universal Dependencies [31, 32]. Результатом работы алгоритма является дерево синтаксических единиц, корень которого имеет тип *предложение*.

4.1. Общая схема. Пусть $D = (V_D, E_D)$ обозначает дерево синтаксических связей; v_D – его узел; $C(v_D)$ – не включённые в дерево

синтаксических единиц прямые потомки узла v_D, отсортированные от наиболее близких в тексте предложения к v_D к наиболее удалённым; $e(v_i, v_j)$ – метка дуги между узлами v_i и v_j. Также $R_t = (r_{t1}, \ldots, r_{tn})$ – набор продукций для синтаксических единиц типа t в порядке приоритета их применения (раздел 3); $N = (V_N, E_N)$ – дерево синтаксических единиц.

Общая схема построения дерева синтаксических единиц по данному дереву синтаксических связей описывается алгоритмом 1. Для построения дерева синтаксических единиц русскоязычного предложения по его дереву синтаксических связей $D_s = (V_{D_s}, E_{D_s})$ используется вызов *CreateConstituencyTree*$(D_s, \textit{предложение})$.

Алгоритм 1. *CreateConstituencyTree* – создаёт дерево синтаксических единиц с корнем заданного типа по данному дереву синтаксических связей

Input: $D = (V_D, E_D)$ – дерево синтаксических связей, v_r – его корень; t – тип синтаксической единицы.

Output: дерево синтаксических единиц $N = (V_N, E_N)$, корень которого имеет тип t.

```
if t является терминалом then
    /* создать синтаксическую единицу типа t из токена v_r */
    return ({v_r}, ∅)
end if
/* выбор продукции для применения */
for r ∈ R_t do
    /* попытка применить r к D */
    (success, N) := ApplyProduction(r, D)
    if success then
        /* если попытка была успешна, то её результат – искомый N */
        return N
    end if
end for
```

Для выбора продукции для построения синтаксических единиц используется вспомогательный алгоритм 2, который, в свою очередь, рекурсивно вызывает основной алгоритм. Рекурсия конечна, так как на каждом её шаге уменьшается мощность множества $C(v_D)$.

Алгоритм 2. *ApplyProduction* – пытается построить дерево синтаксических единиц по данным продукции и дереву синтаксических связей

Input: $D = (V_D, E_D)$ – дерево синтаксических связей, v_r – его корень; $r = (t_1 \dots t_n)$ – продукция, каждый её элемент – тип синтаксической единицы.

Output: *success* – успешность построения (**true** или **false**);
$N = (V_N, E_N)$ – дерево синтаксических единиц, построенное в соответствии с продукцией r.

```
W = ()
/* поиск токена, подходящего каждому элементу продукции */
for t ∈ r do
    for w ∈ v_r ∪ C(v_r) do
        /* каждому элементу продукции должен соответствовать свой токен */
        if w ∈ W then
            continue
        else if w соответствует критерию для t then
            /* токен для t найден */
            W := W + w
            break
        end if
    end for
end for
if |W| ≠ |r| then
    /* для какого-то из элементов продукции не удалось найти токен */
    return (false, (∅, ∅))
end if
/* найдены токены для всех элементов продукции */
/* включим все найденные токены в дерево синтаксических связей */
/* сформируем искомое дерево синтаксических единиц */
N = (∅, ∅)
for i := 1 . . . |r| do
    /* создадим дерево синтаксических единиц для каждой пары из элемента r
и соответствующего ему токена */
    D_i := (V_i, E_i) – поддерево D для W_i
    N_i := CreateConstituencyTree(D_i, r_i)
    добавить N_i к N
end for
return (true, N)
```

Нужно отметить, что продукция r, в соответствии с которой можно построить v_N, существует всегда, поскольку в предложении на

русском языке всегда есть хотя бы один из главных членов предложения, среди наборов продукций для синтаксических групп каждого члена предложения есть продукция, состоящая из одного элемента – самого члена предложения, а в случае построения синтаксических групп однородных членов предложения и группы прямой речи заранее производится проверка возможности построения такой группы.

Далее описаны критерии для различных типов синтаксических единиц грамматики, используемые алгоритмом 2 (условие «w соответствует критерию для t»).

4.2. Критерии для элементов предложения в целом и его грамматической основы. В данной работе считается, что синтаксическая единица предложения в целом состоит из грамматической основы и, возможно, знака препинания. Токен, на основе которого создаётся синтаксическая единица знака препинания, должен находиться в конце предложения и быть знаком препинания. Синтаксическая единица грамматической основы предложения может быть создана на основе любого токена.

Синтаксическая единица вводной конструкции, предваряющей грамматическую основу, может быть построена на основе токена v из $C(v_r)$, находящегося ближе всего к началу предложения, такого, что выполняется хотя бы одно из условий:

– Метка синтаксической связи $e(v_r, v) = parataxis$. Большинство вводных конструкций имеет именно такой тип синтаксической связи: *Безусловно, программист знает математику*. *Кроме того, река замёрзла*.

– Метка синтаксической связи $e(v_r, v) = advcl$, среди $C(v)$ есть союз «как» и между v и v_r в предложении есть запятая. Это условие необходимо, чтобы обнаружить вводные конструкции, схожие с обстоятельствами, например: *Как выражаются моряки, ветер крепчает*.

При этом v не должно быть заключено в скобки или кавычки. Этот же критерий в данной работе используется и для построения вводных конструкций, входящих во все остальные синтаксические единицы.

Синтаксическая единица сложносочинённого предложения может быть построена на основе токена v_r, если среди $C(v_r)$ есть хотя бы один токен v, для которого выполняются следующие условия:

– метка синтаксической связи $e(v_r, v) = conj$;

– среди $C(v)$ есть хотя бы один токен, соответствующий критерию для группы подлежащего.

Первая грамматическая основа сложносочинённого предложения строится на основе самого v_r, вторая и последующие – на основе всех $v \in C(v_r)$, для которых выполняются приведённые выше условия.

Синтаксическая единица расчленённого сложноподчинённого предложения может быть построена на основе токена v_r, если среди $C(v_r)$ есть токен v, для которого выполняется любой из критериев грамматической основы подчинённого предложения:

– v присоединяется к v_r подчинительным союзом, а метка синтаксической связи $e(v_r, v)$ – *parataxis*, *advcl* или *conj*;

– метка синтаксической связи $e(v_r, v)$ – *parataxis* либо *advcl*, а также либо среди $C(v)$ есть хотя бы один токен, соответствующий критерию для группы подлежащего, либо v и v_r не являются глаголами одного времени.

При этом v не должно быть заключено в кавычки или относиться к прямой речи.

Синтаксическая единица группы подлежащего может быть построена на основе токена v_r, если выполняется одно из условий:

– v_r – именная часть речи, и либо $e(v_r, v) \in \{nsubj, nsubj{:}pass\}$, либо v_r – корень дерева синтаксических связей D;

– v_r – глагол в начальной форме, и либо $e(v_r, v) \in \{csubj, csubj{:}pass\}$, либо v_r – корень дерева синтаксических связей D.

Синтаксическая единица группы сказуемого может быть построена на основе токена v_r, если он является корнем дерева синтаксических связей D, и выполняется любое из следующих условий:

– v_r – глагол;

– v_r удовлетворяет критериям для главной части составного именного сказуемого: v_r – это именная часть речи, а среди $C(v_r)$ есть хотя бы один токен, удовлетворяющий критерию для глагола-связки или группы подлежащего. Также главными частями составного именного сказуемого всегда считаются слова-категории состояния и краткие прилагательные.

4.3. Критерии для элементов групп подлежащего, сказуемого, определения, дополнения и обстоятельства. Синтаксическая единица однородных подлежащих может быть построена на основе токена v_r, если среди $C(v_r)$ есть хотя бы один токен v, для которого $e(v_r, v) = conj$. Первое из однородных подлежащих строится на основе самого v_r, второе и последующие – на основе каждого из $v \in C(v_r)$ с меткой синтаксической связи $e(v_r, v) = conj$. Такой же критерий используется и для всех остальных синтаксических единиц однородных членов предложения, если не указано иное.

Синтаксическая единица грамматической основы нерасчленённого придаточного предложения может быть построена на основе токена v_r, если выполняется хотя бы одно из условий:

– v_r отделено от v_d – корня дерева синтаксических связей D – знаком препинания, при этом либо синтаксическая связь $e(v_r, v_d)$ должна иметь метку *acl*, *acl:relcl* или *ccomp*}, либо v_r должно присоединяться к v_d подчинительным союзом;

– v_r отделено от v_d двоеточием и метка синтаксической связи $e(v_r, v_d) = appos$.

При этом синтаксическая единица подчинительного союза строится на основе токена $v \in C(v_r)$, являющегося подчинительным союзом, находящегося в предложении между v_r и v_d и такого, что $e(v, v_r) = mark$. Этот же критерий для нерасчленённого придаточного предложения используется в данной работе и при построении синтаксических единиц для остальных членов предложения.

Синтаксическая единица группы непрямого дополнения может быть построена на основе токена v_r такого, что выполняется хотя бы одно из условий:

– тип синтаксической связи между v_r и корнем дерева синтаксических связей $e(v_r, v_d)$ – *iobj*, *nmod*, *obl* или *xcomp*;

– при v_r есть предлог, и $e(v_r, v_d) = nummod:gov$.

При этом v_r не должно быть характерного для обстоятельств предлога, а среди $v_r \cup C(v_r)$ не должно быть слов, называющих время суток, день, время года или меру времени (раздел 3.6).

Синтаксическая единица группы обстоятельства может быть построена на основе токена v_r, если выполняется хотя бы одно из условий:

– тип синтаксической связи между v_r и корнем дерева синтаксических связей $e(v_r, v_d) = advmod$;

– v_r – наречие или прилагательное, и $e(v_r, v_d)$ – *advcl* или *obl*;

– при v_r есть характерный для обстоятельств предлог (раздел 3.6;

– среди $v_r \cup C(v_r)$ есть хотя бы одно слово, называющие время суток, день недели, время года или меру времени.

Синтаксическая единица группы определения может быть построена на основе токена v_r, если выполняется хотя бы одно из условий:

– тип синтаксической связи между v_r и корнем дерева синтаксических связей $e(v_r, v_d)$ – *amod*, *det* или *appos*;

– v_r – полное прилагательное, и $e(v_r, v_d) = acl$;

– v_r – полное причастие.

Синтаксическая единица группы прямого дополнения может быть построена на основе токена v_r, если выполняется хотя бы одно из условий:

– тип синтаксической связи между v_r и корнем дерева синтаксических связей $e(v_r, v_d) = obj$;

– при v_r нет предлога, и $e(v_r, v_d) = nummod:gov$;

– v_r – именная часть речи в родительном падеже, и v_d – отглагольное существительное.

При построении синтаксической единицы группы однородных сказуемых на основе токенов $V = v_1, \ldots, v_n$ синтаксическая единица их общего непрямого дополнения может быть построена при условии того, что V соединены сочинительным союзом, а не только знаками препинания, на основе токена $v_r \in \bigcup_{i=1}^{n} C(v_i)$ такого, что v_r находится в предложении либо сразу перед первым токеном из V, либо сразу после последнего из них.

Синтаксическая единица составного именного сказуемого может быть построена на основе токена v_r, являющегося корнем дерева синтаксических связей, если выполняется одно из условий:

– v_r входит в список глаголов-связок (раздел 3.3), а также существует $v \in C(v_r)$, являющееся именной частью речи, кратким причастием или категорией состояния. В этом случае синтаксическая единица именной части строится на основе v, а синтаксическая единица глагола-связки – на основе v_r.

– v_r является именной частью речи, кратким причастием или категорией состояния. В этом случае синтаксическая единица именной части строится на основе v_r. Если при этом существует $v \in C(v_r)$, входящий в список глаголов-связок, то на основе v строится синтаксическая единица глагола-связки; если же такого v не существует, то связка считается нулевой и синтаксическая единица для неё не строится.

Синтаксическая единица составного глагольного сказуемого может быть построена на основе токена v_r, являющегося корнем дерева синтаксических связей, если выполняется одно из условий:

– v_r входит в список глаголов-связок, и существует токен $v \in C(v_r)$ такой, что $e(v_r, v)$ – *csubj* или *xcomp*. В этом случае синтаксическая единица глагольной части строится на основе v, а синтаксическая единица глагола-связки – на основе v_r.

– v_r – глагол начальной формы, и существует токен $v \in C(v_r)$, являющийся глаголом-связкой. В этом случае синтаксическая единица

глагольной части строится на основе v_r, а синтаксическая единица глагола-связки – на основе v.

Синтаксическая единица группы прямой речи может быть построена на основе токена v_r такого, что выполняются все условия:

– первый по порядку в предложении элемент $C(v_r)$ является открывающей кавычкой, а последний – закрывающей;

– между v_r и корнем синтаксического дерева v_d есть тире, если v_r в предложении предшествует v_d, и двоеточие, если v_r находится после v_d.

5. Эксперименты

5.1. Корпус. При построении корпуса для экспериментов авторы работы исходили из следующих соображений. Во-первых, необходимо оценить способность предложенного алгоритма качественно строить деревья синтаксических единиц предложений, в которых присутствуют разнообразные синтаксические конструкции русского языка. Во-вторых, необходимо оценить пригодность предложенного алгоритма к использованию в рамках решения прикладных задач компьютерной лингвистики, для чего требуется оценить его способность качественно строить деревья синтаксических единиц предложений, используемых в таких задачах.

Для оценки способности предложенного алгоритма строить деревья синтаксических единиц предложений, в которых встречаются разнообразные синтаксические конструкции русского языка, был использован набор из 200 предложений, выбранных из раздела «Синтаксис» классической работы [13]. Предложения выбирались так, чтобы включать примеры применения как можно более разнообразных конструкций, относящихся к различным разделам синтаксиса, например, устройству простых предложений, различным видам сложных предложений, различным способам выражения членов предложения, последовательностям однородных членов. Разделы синтаксиса и количество подобранных для них предложений приведены в таблице 1.

Для оценки способности предложенного алгоритма строить деревья синтаксических единиц предложений, используемых в прикладных задачах компьютерной лингвистики, например, анализе тональности, было выбрано 100 случайных предложений из OpenCorpora – открытого корпуса русского языка, широко использующегося для решения разнообразных прикладных задач [33 – 35].

Экспертами-лингвистами была выполнена разметка всех 300 предложений по синтаксической структуре.

Таблица 1. Разделы синтаксиса и количество предложений, выбранных для них в качестве примеров

Раздел синтаксиса	Количество предложений
Простое двусоставное предложение	20
Односоставное предложение	10
Составное сказуемое	10
Определение	25
Приложение	5
Дополнение	25
Обстоятельство	25
Однородные члены предложения	10
Вводные конструкции	5
Обособленные члены предложения	10
Сложносочинённое предложение	20
Сложноподчинённое предложение	20
Бессоюзное сложное предложение	10
Прямая речь	5
Итого	200

5.2. Метрики. При проведении экспериментов для оценки качества построения деревьев синтаксических единиц используется несколько метрик.

Первая из них – доля предложений, для которых деревья синтаксических единиц были построены правильно, то есть дерево, полученное в результате работы алгоритма, в точности совпадает с деревом, построенным экспертами-лингвистами:

$$FullyCorrect = \frac{число\ правильно\ построенных\ деревьев}{объём\ корпуса}.$$

Существенный недостаток использования такой метрики – она не учитывает различия между полностью неверно построенными деревьями синтаксических единиц и деревьями синтаксических единиц, построенными с незначительными ошибками. Однако, например, различие между ошибкой, заключающейся в построении для двусоставного предложения дерева синтаксических единиц без группы подлежащего, и ошибкой, связанной с определением прямого дополнения как непрямого, достаточно существенна.

Для компенсации этого недостатка были использованы метрики точности, полноты и F_1-меры построения деревьев синтаксических

единиц [36]. Эти метрики схожи с классическими, используемыми в задачах классификации, однако рассчитываются для всего множества входящих в деревья синтаксических единиц.

– Точность построения деревьев синтаксических единиц – отношение числа синтаксических единиц, входящих и в деревья, построенные алгоритмом, и в деревья, построенные экспертами-лингвистами, к общему числу синтаксических единиц в деревьях, построенных алгоритмом. Точность показывает, какая доля синтаксических единиц, построенных алгоритмом, существует в предложениях на самом деле.

– Полнота построения деревьев синтаксических единиц – отношение числа синтаксических единиц, входящих и в деревья, построенные алгоритмом, и в деревья, построенные экспертами-лингвистами, к общему числу синтаксических единиц в деревьях, построенных экспертами-лингвистами. Полнота характеризует способность алгоритма отыскивать реально существующие в предложении синтаксические единицы.

– F_1-мера – среднее гармоническое точности и полноты.

Обозначив G_{TP} – синтаксические единицы, входящие и в деревья, построенные экспертами, и в деревья, построенные алгоритмом, G_{FP} – синтаксические единицы, входящие в деревья, построенные алгоритмом, но не в деревья, построенные экспертами, G_{FN} – синтаксические единицы, входящие в деревья, построенные экспертами, но не в деревья, построенные алгоритмом, можно записать формулы для точности, полноты и F_1-меры построения деревьев синтаксических единиц:

$$Precision = \frac{G_{TP}}{G_{TP} + G_{FP}}; \quad Recall = \frac{G_{TP}}{G_{TP} + G_{FN}}; \quad F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}.$$

Помимо информации о внутренней синтаксической структуре предложения, дерево синтаксических единиц также содержит информацию о том, каким членом предложения является каждое из слов (листьев дерева). Для того, чтобы отдельно оценить точность этой информации, используется дополнительная метрика точности определения членов предложения, рассчитываемая по формуле:

$$Tagging = \frac{\textit{число верно определённых членов предложений}}{\textit{число слов в предложениях}}.$$

Поскольку деревья синтаксических связей редко бывают входными данными в практических задачах, авторам работы представляется

разумным при проведении экспериментов использовать в качестве входных данных сами предложения на естественном языке: текст предложения преобразуется анализатором синтаксических связей в дерево синтаксических связей; на основе этого дерева предложенный алгоритм строит дерево синтаксических единиц, которое сравнивается с эталоном, размеченным экспертами-лингвистами. При этом результаты работы анализатора синтаксических связей не валидируются – если для какого-то предложения он допустил ошибку, то это предложение не будет удалено из рассмотрения, и ошибки, допущенные из-за неверной работы анализатора синтаксических связей, не отделяются от ошибок, допущенных из-за неверной работы предложенного алгоритма. Конечно, такая оценка эффективности предложенного алгоритма может оказаться значительно ниже его реальной эффективности, однако, она позволяет сделать наиболее точный вывод о применимости предложенного алгоритма для решения практических задач.

Безусловно, при наличии достаточных ресурсов можно валидировать результаты работы анализатора синтаксических связей, и оценивать точность работы предложенного алгоритма только на тех предложениях, дерево синтаксических связей для которых было построено полностью верно. Однако, нужно отметить, что, валидация деревьев синтаксических связей – достаточно трудоёмкий процесс, и, при наличии достаточных для него ресурсов, возможно, целесообразнее будет потратить их напрямую на разметку банка деревьев для создания алгоритма анализа синтаксической структуры предложения, не использующего анализатор синтаксических связей.

5.3. Результаты. Для проведения экспериментов предложенный алгоритм был реализован на языке программирования Python; программный код реализации доступен в репозитории https://github.com/yarfruct/constituency-from-dependency. Для построения деревьев синтаксических связей, размеченных в соответствии с системой Universal Dependencies, использовались анализаторы из состава трёх библиотек: Stanza (обученный на наборе данных SynTagRus), SpaCy (версии ru_core_news_lg) и Natasha. Полученные в результате экспериментов метрики качества приведены в таблице 2.

На 200 предложениях из справочника Д.Э. Розенталя наилучшие результаты получаются при использовании анализатора синтаксических связей из состава библиотеки Stanza – для почти трёх четвертей предложений деревья синтаксических единиц строятся полностью верно; верно определяются практически все (95 %) члены предложения; точность, полнота и F_1-мера также достаточно высоки и составляют

почти 0.9 каждая. Отсутствие существенной разницы между точностью и полнотой построения деревьев синтаксических единиц говорит о том, что алгоритм не склонен ни строить «лишние» синтаксические единицы, отсутствующие в построенных экспертами деревьях, ни упускать присутствующие в предложении синтаксические единицы. При использовании анализаторов синтаксических связей из состава библиотек SpaCy и Natasha качество значительно хуже: доля предложений, для которых деревья синтаксических единиц строятся полностью корректно, ниже на 0.16 для SpaCy и на 0.28 для Natasha; F_1-мера – на 0.08 для SpaCy и на 0.15 для Natasha; точность определения членов предложения – на 0.05 для SpaCy и на 0.15 для Natasha. Нужно отметить, что для некоторых предложений анализатор деревьев синтаксических связей из Natasha строил деревья синтаксических связей, содержащие циклы, то есть заведомо некорректные.

Таблица 2. Метрики качества построения деревьев синтаксических единиц с помощью различных анализаторов синтаксических связей

Анализатор	*FullyCorrect*	*Precision*	*Recall*	F_1	*Tagging*
Для 200 предложений из справочника Д. Э. Розенталя					
Stanza	0.73	0.88	0.89	0.89	0.95
SpaCy	0.57	0.81	0.81	0.81	0.90
Natasha	0.45	0.78	0.70	0.74	0.80
Для 100 предложений из OpenCorpora					
Stanza	0.46	0.83	0.83	0.83	0.92
SpaCy	0.23	0.70	0.69	0.70	0.84
Natasha	0.12	0.68	0.49	0.57	0.62
Для всех 300 предложений					
Stanza	0.64	0.85	0.85	0.85	0.93
SpaCy	0.45	0.75	0.75	0.75	0.87
Natasha	0.34	0.73	0.58	0.65	0.70

На 100 предложениях из OpenCorpora наилучшие результаты также получаются при использовании анализатора из состава Stanza. При этом метрики качества ниже, чем для предложений из справочника – доля предложений с полностью правильно построенными деревьями снизилась существеннее всего, на 0.27, до 0.46; снижение F_1-меры и точности определения членов предложения оказалось не таким существенным, всего на 0.06 и 0.03 соответственно. Для двух других анализаторов синтаксических связей снижение качества оказалось ещё более существенным – для анализатора из состава SpaCy доля полностью правильно построенных деревьев снизилась до 0.23, F_1-мера – до 0.70,

а точность определения членов предложения – до 0.84; для анализатора из состава Natasha эти показатели составили 0.12, 0.57 и 0.62 соответственно, кроме того, по разнице между точностью, составившей 0.68 и полнотой, составившей 0.49, при использовании этого анализатора алгоритм склонен включать в дерево меньше синтаксических единиц, чем реально существует в предложении.

Скорее всего, такое снижение качества при использовании всех анализаторов связано с тем, что предложения из OpenCorpora в среднем более сложные и содержат в себе больше комбинаций различных синтаксических конструкций, что может затруднять работу анализаторов синтаксических связей. В пользу этой гипотезы говорит то, что для наиболее точного из анализаторов – из состава библиотеки Stanza – показатели качества снизились меньше, чем для анализаторов из состава SpaCy и Natasha.

Итоговые метрики качества, рассчитанные на всех 300 предложениях, позволяют говорить о том, что при использовании анализатора синтаксических связей из состава библиотеки Stanza предлагаемый алгоритм включает в деревья практически все синтаксические единицы, реально существующие в предложениях, и достаточно редко включает несуществующие (F_1-мера – 0.85), а также достаточно точно определяет, каким членом предложения является то или иное слово (точность определения – 0.93), однако примерно в двух третях построенных деревьев синтаксических единиц присутствовали хотя бы минимальные ошибки. При использовании других синтаксических анализаторов построение деревьев синтаксических единиц происходит гораздо менее точно, с существенным снижением метрик качества. Тем не менее, поскольку во многих прикладных задачах, например, анализа тональности и извлечения информации, не требуется абсолютная точность синтаксического анализа, можно предположить, что предложенный алгоритм может успешно применяться на практике.

Заключение. В работе предложен алгоритм преобразования деревьев синтаксических связей в деревья синтаксических единиц для русскоязычных предложений. Он основан на рекурсивном применении к узлам дерева синтаксических связей продукций формальной грамматики, построенной на основе классического справочника Д. Э. Розенталя. Для оценки качества работы алгоритма был собран корпус из 300 предложений на русском языке, из которых 200 были выбраны из вышеупомянутого справочника, и 100 из открытого корпуса публицистических текстов OpenCorpora.

В ходе экспериментов предложения подавались на вход анализаторов синтаксических связей из состава библиотек Stanza, SpaCy и Natasha, после чего полученные деревья синтаксических связей обрабатывались предложенным алгоритмом. Полученные в результате обработки деревья синтаксических единиц сравнивались с размеченными вручную экспертами-филологами. Наилучшее качество было получено при использовании анализатора синтаксических связей из библиотеки Stanza: доля полностью правильно построенных деревьев синтаксических единиц составила 0.64, точность определения членов предложения составила 0.93, а F_1-мера построения синтаксических единиц составила 0.85, чего должно быть достаточно для решения многих практических задач.

Дальнейшее повышение качества работы алгоритма может осуществляться с помощью расширения используемой формальной грамматики. Доработка грамматики и алгоритма, безусловно, потребуют проведения анализа ошибок и привлечения экспертов-лингвистов для уточнения используемых продукций.

Литература

1. Jurafsky D., Martin J.H. Speech and Language Processing. 2nd Edition. USA: Prentice-Hall, Inc., 2009. 1024 p.
2. Батура Т.В., Чаринцева М.В. Основы обработки текстовой информации: Учебное пособие. Новосибирск: Институт систем информатики им. А.П. Ершова СО РАН, 2016. 45 с.
3. Андреева С.В. Типология конструктивно-синтаксических единиц в русской речи // Вопросы языкознания. 2004. № 5. С. 32–45.
4. Онипенко Н.К. Об основаниях классификации синтаксических единиц // Труды института русского языка им. В.В. Виноградова. 2019. Т. 20. С. 189–201.
5. Percival W.K. On the historical source of immediate constituent analysis // Notes from the linguistics underground. 1976. pp. 229–242.
6. Waziri Z.Y., Safana M.I. Contrastive analysis of English and Hausa sentence structures and its pedagogical implications // Voices: A Journal of English Studies. 2021. vol. 5. pp. 15–27.
7. Dewi N.M.P., Putra I.G.W.N., Winarta I.B.G.N. Imperative Sentence in «The Guidance iPhone Support Website» // Elysian Journal: English Literature, Linguistics and Translation Studies. 2021. vol. 1. pp. 81–92.
8. Nguyen H.V., Tan N., Quan N.H., Huong T.T., Phat N.H. Building a Chatbot System to Analyze Opinions of English Comments // Informatics and Automation. 2023. vol. 22. no. 2. pp. 289–315.
9. Matchin W., Hickok G. The cortical organization of syntax // Cerebral Cortex. 2020. vol. 30. no. 3. pp. 1481–1498.
10. Ениколопов С.Н., Кузнецова Ю.М., Осипов С.Г., Смирнов И.В., Чудова Н.В. Метод реляционно-ситуационного анализа текста в психологических исследованиях // Психология. Журнал Высшей школы экономики. 2021. Т. 18. № 4. С. 748–769.

11. Zhang Y., Zhang Y. Tree communication models for sentiment analysis // Proceedings of the 57th annual meeting of the association for computational linguistics. 2019. pp. 3518–3527. DOI: 10.18653/v1/P19-1342.
12. Marcus M., Santorini B., Marcinkewicz M.A. Building a large annotated corpus of English: The Penn Treebank // Computational Linguistics. 1993. vol. 19 no. 2. pp. 313–330.
13. Розенталь Д.Э., Голуб И.Б., Теленкова М.А. Современный русский язык. 16-е изд. М.: АЙРИС-пресс, 2018. 448 с.
14. Chomsky N. On certain formal properties of grammars // Information and control. 1959. vol. 2. no. 2. pp. 137–167.
15. Chomsky N. Some Puzzling Foundational Issues: the Reading Program // Catalan journal of linguistics. 2019. pp. 263–285. DOI: 10.5565/rev/catjl.287.
16. Muller S. Grammatical theory: From transformational grammar to constraint-based approaches. Fifth revised and extended edition. Berlin: Language Science Press, 2023. 889 p. DOI: 10.17169/langsci.b25.167.
17. Taylor A., Marcus M., Santorini B. The Penn Treebank: an overview // Treebanks: Building and using parsed corpora. Dordrecht: Springer Netherlands, 2003. 407 p. DOI: 10.1007/978-94-010-0201-1.
18. Zhou J., Zhao H. Head-Driven Phrase Structure Grammar Parsing on Penn Treebank // Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019. pp. 2396–2408.
19. Gaddy D., Stern M., Klein D. What's Going On in Neural Constituency Parsers? An Analysis // Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2018. vol. 1. pp. 999–1010.
20. Zhang M.S. A survey of syntactic-semantic parsing based on constituent and dependency structures // Science China Technological Sciences. 2020. vol. 63. no. 10. pp. 1898–1920.
21. Yang S., Cui L., Ning R., Wu D., Zhang Y. Challenges to open-domain constituency parsing // Findings of the Association for Computational Linguistics: ACL 2022. 2022. pp. 112–127.
22. Гладкий А.В., Мельчук И.А. Элементы математической лингвистики. М.: Наука, 1969. 192 с.
23. Гладкий А.В. Синтаксические структуры естественного языка. Изд. 2-е. М.: УРСС, 2007. 146 с.
24. Коротаев Н.А. Синтаксические группы А.В Гладкого: анализ конструкций с сочинением // Вестник РГГУ. Серия: Литературоведение. Языкознание. Культурология. 2013. № 8(109). С. 16–36.
25. Кагиров И.А., Леонтьева А.Б. Модуль синтаксического анализа для литературного русского языка // Труды СПИИРАН. 2008. Т. 6. С. 171–183.
26. Leontyeva A., Kagirov I. The module of morphological and syntactic analysis SMART // Text, Speech and Dialogue: 11th International Conference, TSD 2008. 2008. pp. 373–380.
27. Леонтьева Н.Н., Ермаков М.В., Крылов С.А., Семенова С.Ю., Соколова Е.Г. Прикладной семантический словарь РУСЛАН: основная концепция и обновленный подход // Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной международной конференции «Диалог». 2020. С. 1049–1064.
28. Москвина А.Д., Орлова Д., Паничева П.В., Митрофанова О.А. Разработка ядра синтаксического анализатора для русского языка на основе библиотек NLTK // Сборник научных статей. Труды XIX Международной объединённой научной

конференции «Интернет и современное общество». Санкт-Петербург: Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики. 2016. С. 44–54.

29. Shelmanov A., Pisarevskaya D., Chistova E., Toldova S., Kobozeva M., Smirnov I. Towards the data-driven system for rhetorical parsing of Russian texts // Proceedings of the Workshop on Discourse Relation Parsing and Treebanking. 2019. pp. 82–87.
30. Гаврилов Д.А Сопоставительное изучение пунктуации в сетевом газетном заголовке: к постановке проблемы // Вестник Чувашского государственного педагогического университета им. И.Я. Яковлева. 2021. № 3(112). С. 3–8.
31. De Marneffe M.C, Manning C.D., Nivre J., Zeman D. Universal Dependencies // Computational Linguistics. 2021. vol. 47. no. 2. pp. 255–308.
32. Lyashevskaya O., Bocharov V., Sorokin A., Shavrina T., Granovsky D., Alexeeva S. Text collections for evaluation of Russian morphological taggers // Journal of Linguistics / Jazykovedny Casopis. 2017. vol. 68. no. 2. pp. 258–267.
33. Kirillovich A., Loukachevitch N., Kulaev M., Bolshina A., Ilvovsky D. Sense-Annotated Corpus for Russian // Proceedings of the 5th International Conference on Computational Linguistics in Bulgaria (CLIB 2022). 2022. pp. 130–136.
34. Volkova L., Bocharov V. An approach to inter-annotation agreement evaluation for the named entities annotation task at OpenCorpora // Communications in Computer and Information Science. 2019. vol. 1119. pp. 33–44.
35. Lagutina K. Topical Text Classification of Russian News: a comparison of BERT and Standard Models // 31st Conference of Open Innovations Association FRUCT. 2022. pp. 160–166.
36. Yang S., Tu K. Bottom-up constituency parsing and nested named entity recognition with pointer networks // Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics. 2022. vol. 1. pp. 2403–2416.

Полетаев Анатолий Юрьевич – ассистент, кафедра компьютерных сетей, факультет информатики и вычислительной техники, Ярославский государственный университет им. П.Г. Демидова. Область научных интересов: анализ и моделирование естественного языка, математическая статистика. Число научных публикаций — 14. anatoliy-poletaev@mail.ru; улица Советская, 14, 150003, Ярославль, Россия; р.т.: +7(910)819-8325.

Парамонов Илья Вячеславович – канд. физ.-мат. наук, доцент, кафедра компьютерных сетей, факультет информатики и вычислительной техники, Ярославский государственный университет им. П.Г. Демидова; руководитель лаборатории, ярославская лаборатория Ассоциации открытых инноваций FRUCT. Область научных интересов: компьютерная лингвистика, искусственные нейронные сети, методология разработки программного обеспечения. Число научных публикаций — 55. ilya.paramonov@fruct.org; улица Советская, 14, 150003, Ярославль, Россия; р.т.: +7(905)633-3993.

Бойчук Елена Игоревна – д-р филол. наук, доцент, старший научный сотрудник, отдел управления наукой и инновациями, Ярославский государственный университет им. П.Г. Демидова. Область научных интересов: компьютерная лингвистика, филология, литературоведение, лингвистика текста, функциональная грамматика, теория языка и сравнительно-сопоставительные исследования. Число научных публикаций — 108. elena-boychouk@rambler.ru; улица Советская, 14, 150003, Ярославль, Россия; р.т.: +7(903)824-1797.

Поддержка исследований. Исследование выполнено за счет гранта Российского научного фонда № 23-21-00495 (https://rscf.ru/project/23-21-00495/).

DOI 10.15622/ia.22.6.3

A. POLETAEV, I. PARAMONOV, E. BOYCHUK

ALGORITHM OF CONSTITUENCY TREE FROM DEPENDENCY TREE CONSTRUCTION FOR A RUSSIAN-LANGUAGE SENTENCE

Poletaev A., Paramonov I., Boychuk E.. **Algorithm of Constituency Tree from Dependency Tree Construction for a Russian-Language Sentence.**

Abstract. Automatic syntactic analysis of a sentence is an important computational linguistics task. At present, there are no syntactic structure parsers for Russian that are publicly available and suitable for practical applications. Ground-up creation of such parsers requires building of a treebank annotated according to a given formal grammar, which is quite a cumbersome task. However, since there are several syntactic dependency parsers for Russian, it seems reasonable to employ dependency parsing results for syntactic structure analysis. The article introduces an algorithm that allows to construct the constituency tree of a Russian sentence by a syntactic dependency tree. The formal grammar used by the algorithm is based on the D.E. Rosenthal's classic reference. The algorithm was evaluated on 300 Russian-language sentences. 200 of them were selected from the aforementioned reference, and 100 from OpenCorpora, an open corpus of sentences extracted from Russian news and periodicals. During the evaluation, the sentences were passed to syntactic dependency parsers from Stanza, SpaCy, and Natasha packages, then the resulted dependency trees were processed by the proposed algorithm. The obtained constituency trees were compared with the trees manually annotated by experts in linguistics. The best performance was achieved using the Stanza parser: the constituency parsing F_1-score was 0.85, and the sentence parts tagging accuracy was 0.93, that would be sufficient for many practical applications, such as event extraction, information retrieval and sentiment analysis.

Keywords: computational linguistics, natural language processing, syntactic parsing, constituency tree, dependency tree, formal grammar.

References

1. Jurafsky D., Martin J.H. Speech and Language Processing. 2nd Edition. USA: Prentice-Hall, Inc., 2009. 1024 p.
2. Batura T V., Charinceva M.V. Osnovy obrabotki tekstovoj informacii: Uchebnoe posobie. [Basics of textual information processing: Study Guide]. Novosibirsk: Institut sistem informatiki im. A.P. Ershova SO RAN, 2016. 45 p. (in Russ.).
3. Andrejeva S.V. [Typology of constructive-syntactic units in Russian speech]. Voprosy yazykoznaniya – Problems of linguistics. 2004. no. 5. pp. 32–45. (in Russ.).
4. Onipenko N.K. [About the grounds for the classification of syntactic units]. Trudy Instituta Russkogo Iazyka im. V.V. Vinogradova – Proceedings of the V.V. Vinogradov Russian Language Institute. 2019. vol. 20. pp. 189–201. (in Russ.).
5. Percival W.K. On the historical source of immediate constituent analysis. Notes from the linguistics underground. 1976. pp. 229–242.
6. Waziri Z.Y., Safana M.I. Contrastive analysis of English and Hausa sentence structures and its pedagogical implications. Voices: A Journal of English Studies. 2021. vol. 5. pp. 15–27.
7. Dewi N.M.P., Putra I.G.W.N., Winarta I.B.G.N. Imperative Sentence in «The Guidance iPhone Support Website». Elysian Journal: English Literature, Linguistics and Translation Studies. 2021. vol. 1. pp. 81–92.

8. Nguyen H.V., Tan N., Quan N.H., Huong T.T., Phat N.H. Building a Chatbot System to Analyze Opinions of English Comments. Informatics and Automation. 2023. vol. 22. no. 2. pp. 289–315.
9. Matchin W., Hickok G. The cortical organization of syntax. Cerebral Cortex. 2020. vol. 30. no. 3. pp. 1481–1498.
10. Enikolopov S.N., Kuznetsova Y.M., Osipov G.S., Smirnov I.V., Chudova N.V. [The Method of Relational-Situational Analysis of Text in Psychological Research]. Psihologiya. Zhurnal vysshej shkoly ekonomiki – Psychology. Journal of the Higher School of Economics]. 2021. vol. 18. no. 4. pp. 748–769. (in Russ.).
11. Zhang Y., Zhang Y. Tree communication models for sentiment analysis. Proceedings of the 57th annual meeting of the association for computational linguistics. 2019. pp. 3518–3527. DOI: 10.18653/v1/P19-1342.
12. Marcus M., Santorini B., Marcinkewicz M.A. Building a large annotated corpus of English: The Penn Treebank. Computational Linguistics. 1993. vol. 19 no. 2. pp. 313–330.
13. Rozental D.E., Golub I.B., Telenkova M.A. Sovremennyj russkij jazyk [Modern Russian language (16th Edition)]. Moscow: AJRIS-press, 2018. 448 p. (in Russ.).
14. Chomsky N. On certain formal properties of grammars. Information and control. 1959. vol. 2. no. 2. pp. 137–167.
15. Chomsky N. Some Puzzling Foundational Issues: the Reading Program. Catalan journal of linguistics. 2019. pp. 263–285. DOI: 10.5565/rev/catjl.287.
16. Muller S. Grammatical theory: From transformational grammar to constraint-based approaches. Fifth revised and extended edition. Berlin: Language Science Press, 2023. 889 p. DOI: 10.17169/langsci.b25.167.
17. Taylor A., Marcus M., Santorini B. The Penn Treebank: an overview. Treebanks: Building and using parsed corpora. Dordrecht: Springer Netherlands, 2003. 407 p. DOI: 10.1007/978-94-010-0201-1.
18. Zhou J., Zhao H. Head-Driven Phrase Structure Grammar Parsing on Penn Treebank. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019. pp. 2396–2408.
19. Gaddy D., Stern M., Klein D. What's Going On in Neural Constituency Parsers? An Analysis. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2018. vol. 1. pp. 999–1010.
20. Zhang M.S. A survey of syntactic-semantic parsing based on constituent and dependency structures. Science China Technological Sciences. 2020. vol. 63. no. 10. pp. 1898–1920.
21. Yang S., Cui L., Ning R., Wu D., Zhang Y. Challenges to open-domain constituency parsing. Findings of the Association for Computational Linguistics: ACL 2022. 2022. pp. 112–127.
22. Gladkij A.V., Melchuk I.A. Jelementy matematicheskoj lingvistiki [Elements of mathematical linguistics]. Moscow: Nauka, 1969. 192 p. (in Russ.).
23. Gladkij A.V. Sintaksicheskie struktury estestvennogo jazyka [Syntactic structures of natural language (2nd Edition)]. Moscow: URSS, 2007. 146 p. (in Russ.).
24. Korotaev N.A. [A.V. Gladkij syntactic groups: analysis of compound constructions]. Vestnik RGGU. Seriya: Literaturovedenie. Yazykoznanie. Kulturologiya – RSUH Bulletin. «Literary theory. Linguistics. Cultural Studies.» Series.]. 2013. no. 8(109). pp. 16–36. (in Russ.).
25. Kagirov I.A., Leontyeva A.B. [Module for syntax parsing of the literary Russian language]. Trudy SPIIRAN – SPIIRAS Proceedings. 2008. vol. 6. pp. 171–183. (in Russ.).
26. Leontyeva A., Kagirov I. The module of morphological and syntactic analysis SMART. Text, Speech and Dialogue: 11th International Conference, TSD 2008. 2008. pp. 373–380.

27. Leontyeva N.N., Ermakov M.V., Krylov S.A., Semenova S.Yu., Sokolova E.G. [On traditional conception and upgrading of one applied semantic dictionary]. Kompyuternaya lingvistika i intellektualnye tekhnologii: Po materialam ezhegodnoj mezhdunarodnoj konferencii «Dialog» – Computational linguistics and intellectual technologies: Papers from the annual international conference «Dialogue». 2020. pp. 1049–1064. (in Russ.).
28. Moskvina A.D., Orlova D. Panicheva P.V., Mitrofanova O.A. Razrabotka yadra sintaksicheskogo analizatora dlya russkogo yazyka na osnove bibliotek NLTK [Development of the Core for Syntactic Parser for Russian based on NLTK libraries] Kompyuternaya lingvistika i vychislitelnye ontologii: Sbornik nauchnyh statej. Trudy XIX Mezhdunarodnoj ob'edinyonnoj nauchnoj konferencii [Computational linguistics and computational ontologies: Collection of scientific articles. Proceedings of the XIXth International joint scientific conference]. St. Petersburg: Sankt-Peterburgskij nacional'nyj issledovatel'skij universitet informacionnyh tekhnologij, mekhaniki i optiki, 2016. pp. 44–54. (in Russ.).
29. Shelmanov A., Pisarevskaya D., Chistova E., Toldova S., Kobozeva M., Smirnov I. Towards the data-driven system for rhetorical parsing of Russian texts. Proceedings of the Workshop on Discourse Relation Parsing and Treebanking. 2019. pp. 82–87.
30. Gavrilov D.A. [Comparative study of punctuation in an online newspaper headline: statement of the problem]. Vestnik Chuvashskogo Gosudarstvennogo Pedagogicheskogo Universiteta im. I. Y. Yakovleva – I. Yakovlev Chuvash State Pedagogical University Bulletin. 2021. no. 3(112). pp. 3–8. (in Russ.).
31. De Marneffe M.C, Manning C.D., Nivre J., Zeman D. Universal Dependencies. Computational Linguistics. 2021. vol. 47. no. 2. pp. 255–308.
32. Lyashevskaya O., Bocharov V., Sorokin A., Shavrina T., Granovsky D., Alexeeva S. Text collections for evaluation of Russian morphological taggers. Journal of Linguistics / Jazykovedny Casopis. 2017. vol. 68. no. 2. pp. 258–267.
33. Kirillovich A., Loukachevitch N., Kulaev M., Bolshina A., Ilvovsky D. Sense-Annotated Corpus for Russian. Proceedings of the 5th International Conference on Computational Linguistics in Bulgaria (CLIB 2022). 2022. pp. 130–136.
34. Volkova L., Bocharov V. An approach to inter-annotation agreement evaluation for the named entities annotation task at OpenCorpora. Communications in Computer and Information Science. 2019. vol. 1119. pp. 33–44.
35. Lagutina K. Topical Text Classification of Russian News: a comparison of BERT and Standard Models. 31st Conference of Open Innovations Association FRUCT. 2022. pp. 160–166.
36. Yang S., Tu K. Bottom-up constituency parsing and nested named entity recognition with pointer networks. Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics. 2022. vol. 1. pp. 2403–2416.

Poletaev Anatoliy – Assistant, Chair of computer networks, faculty of computer science, P.G. Demidov Yaroslavl State University. Research interests: natural language analysis and modeling, mathematical statistics. The number of publications — 14. anatoliy-poletaev@mail.ru; 14, Sovetskaya St., 150003, Yaroslavl, Russia; office phone: +7(910)819-8325.

Paramonov Ilya – Ph.D., Associate professor, Chair of computer networks, faculty of computer science, P.G. Demidov Yaroslavl State University; Head of the laboratory, Yaroslavl Laboratory of Open Innovations Association FRUCT. Research interests: computational linguistics, artificial neural networks, software development methods. The number of publications — 55. ilya.paramonov@fruct.org; 14, Sovetskaya St., 150003, Yaroslavl, Russia; office phone: +7(905)633-3993.

Boychuk Elena – Ph.D., Dr.Sci., Associate Professor, Senior researcher, Research and development department, P.G. Demidov Yaroslavl State University. Research interests: computational linguistics, philology, literary criticism, text linguistics, functional grammar, language theory and comparative studies. The number of publications — 108. elena-boychouk@rambler.ru; 14, Sovetskaya St., 150003, Yaroslavl, Russia; office phone: +7(903)824-1797.

Acknowledgements. The reported study was funded by the grant of Russian Science Foundation No. 23-21-00495 (https://rscf.ru/en/project/23-21-00495/).