МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ

ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. А. И. ГЕРЦЕНА»

**институт информационных технологий и технологического образования**

**кафедра информационных технологий и электронного обучения**

Основная профессиональная образовательная программа

Направление подготовки 09.03.01 Информатика и вычислительная техника

Направленность (профиль) «Технологии разработки программного обеспечения»

форма обучения – очная

**Курсовая работа**

по дисциплине «Технологии компьютерного моделирования»

Моделирование синтаксической структуры текста при помощи инструментов обработки естественного языка

Обучающейся 2 курса

Елкиной Г.А.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководители:

д.п.н, профессор

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Власова Е. З.

«\_\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

к.п.н, доцент

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Гончарова С. В.

«\_\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Санкт-Петербург

2020

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 3](#_Toc42248615)

[ОСНОВНАЯ ЧАСТЬ 5](#_Toc42248616)

[**1 Теоретический раздел** 5](#_Toc42248617)

[**2 Практический раздел** 12](#_Toc42248618)

[ЗАКЛЮЧЕНИЕ 16](#_Toc42248619)

[СПИСОК ЛИТЕРАТУРЫ 17](#_Toc42248620)

[ПРИЛОЖЕНИЕ А 19](#_Toc42248621)

[ПРИЛОЖЕНИЕ Б 19](#_Toc42248622)

# ВВЕДЕНИЕ

Обработка естественного языка – это часть таких научных направлений, как компьютерная лингвистика и искусственный интеллект. На данный момент проблема понимания и синтеза текстов компьютером является актуальной и широко исследуемой во всем мире.

Если понимание текста применительно к человеку характеризуется неким субъективным состоянием: человек считает, что он что-то понял, значит, так и есть, - то применительно к компьютеру такая характеристика понимания не может быть признана приемлемой: ей нужны четкие и точно фиксируемые признаки, чтобы что-либо понять. И для понимания машиной какого-либо объема текста созданы алгоритмы обработки естественного языка. Данным направлением занимаются многие компании и научные деятели, связанные с сферой искусственного интеллекта. Главная задача компьютерной лингвистики и синтеза тексов в целом состоит в том, чтобы сделать процесс общения с компьютером максимально удобным, а следовательно, и продуктивным.

Понимание текста компьютером состоит в том, чтобы правильно выявить в тексте предложения, словосочетания и слова, структурировать их и наделить смыслом, который в них изначально вложил человек. Этот алгоритм и представляет собой морфологический, синтаксический и семантический анализы текста, составляющие основу полноценного анализа текста для понимания и усвоения информации, заложенной в тексте, компьютером.

*Целью* данной курсовой работы является изучение методов и инструментария синтаксического анализа текста и составление синтаксической структуры при помощи инструментов моделирования в сфере обработки естественного языка.

*Задачами* моей курсовой работы являются:

1. выявление актуальности выбранной темы;
2. исследование общих принципов синтаксического анализа текста как со стороны лингвистики, так и со стороны компьютерных технологий;
3. изучение инструментов компьютерной лингвистики для синтаксического анализа;
4. исследование программного обеспечения, выбранного мной для выполнения цели работы;
5. моделирование синтаксической структуры простых и сложных составляющих текста при помощи выбранного программного обеспечения;
6. анализ полученных результатов моделирования и составление соответствующих выводов по работе;
7. анализ практических решений проблем, выявленных в процессе выполнения работы.

Синтаксический анализ является частью обработки естественного языка, поэтому актуальность изучения именно этой темы в том, что технологии структурирования текста, разбора его на синтаксические части и построения структуры понятной компьютеру все еще совершенствуются и при этом широко используются, например, при машинном переводе, планировании текста («text mining») и в экспертных системах. А также появляются новые прикладные задачи, требующие постоянного улучшения эффективности применяемых алгоритмов, что в свою очередь требует разработки все более совершенных подходов к анализу текстов и самообучению компьютерной системы.

# ОСНОВНАЯ ЧАСТЬ

## **1 Теоретический раздел**

Главная задача синтаксического анализа как со стороны лингвистики, так и со стороны компьютерных технологий – это обозначение роли слов в предложениях и словосочетаниях, а также определение связей между словами.

Словосочетание состоит из двух и более слов, обязательно связанных подчинительной связью. Существует три вида связи:

1) Согласование – вид связи, при котором лексическая форма подчиненного слова повторяет лексическую форму главного.

2) Управление – вид связи, при котором лексическая форма подчиненного слова не повторяет форму главного, но зависит от его значения.

3) Примыкание – вид связи, при котором зависимость слова выражается лексически, порядком слов и интонацией, без применения служебных слов и морфологического изменения.

Но минимальной единицей человеческой речи является не словосочетание, а предложение. Оно представляет из себя грамматически организованное соединение слов или слово, содержащее в себе законченную мысль. На письме конец предложения оформляется точкой, знаками восклицания или вопроса, а также реже многоточием.

Для распределения ролей слов в предложении существуют определенные правила и обозначения.

Подлежащее (от лат. «subjectum» – субъект) определяется как один из двух главных членов предложения. Традиционное определение подлежащего – «то, о чем говорится в предложении».

Сказуемое — это главный член предложения, который связан с подлежащим и отвечает на вопросы «что делает предмет (или лицо)?», «что с ним происходит?», «что он такое?», «кто он такой?» и др., т.е. сказуемое обозначает действие или состояние предметов и лиц, которые выражены подлежащим.

Определение – это второстепенный член предложения, который обозначает признак предмета. Существуют согласованные и несогласованные определения. Это выражается тем, как связано определение с главным словом. Обычно определения конкретизируют описываемый ими предмет или лицо. Поэтому они иногда несут основную смысловую нагрузку члена предложения, с которым они согласуются, и могут быть включены в его состав.

Приложение – это частный случай определения, который отвечает за другое название описываемого предмета или лица. Чаще всего выражается существительным и может дать характеристику предмета, указывать на степень родства, национальность, звание, профессию.

Дополнение является также второстепенным членом предложения, зависящим от других членов, и дополняет смысл слов, от которых зависит. Дополнения делятся на прямые и косвенные, что зависит от того, как образовано слово.

Обстоятельство – второстепенный член предложения, который зависит от сказуемого. Обстоятельства различают по задаваемым к ним вопросам на обстоятельства времени, места, образа действия, причины, цели, условия, уступки.

Существует некая зависимость между тем, какой частью речи является слово и какую роль оно может играть в предложении. Причем такую зависимость можно представить в виде таблицы, общей как минимум для всех индоевропейских языков (см. приложение А).

Для частей речи, которые могут быть разными членами предложения, синтаксическая роль может быть определена, исходя из их морфологической формы или по их месту в предложении относительно других слов.

Блок синтаксического анализа состоит, как правило, из трех частей:

1) сегментации предложения;

2) устранение морфологической омонимии;

3) установления связей между словами.

На этапе сегментации простые предложения выделяются в составе сложного. Любое простое предложение может быть разорвано деепричастными или причастными оборотами или придаточными предложениями. Важно правильно соотнести каждый сегмент к своему простому предложению.

Морфологический разбор слов обычно предшествует синтаксическому анализу, и на этом этапе важно понять, с какой из форм слова мы имеем дело. И если этого не было сделано на предыдущем этапе анализа текста, то морфологическая омонимия устраняется прежде, чем перейти к установлению связей между словами, по средствам анализа расположения слов вокруг слова с данной аномалией.

На этапе установки синтаксических связей между словами – главной задачи синтаксического анализа – происходит формальное описание синтаксиса с помощью различных правил. Существует несколько формальных моделей синтаксиса:

1. Модели, основанные на Link Grammar.

Link Grammar (LG) (грамматика ссылок) – это теория синтаксиса, разработанная Дэйвом Темперли и Дэниэлом Слитором, которая построена на отношениях между парами слов в отличии от древоподобных структур в других моделях. В то время как большинство систем используют структуры уровня именных и глагольных групп при построении дерева фразы, грамматика ссылок использует информацию о типах связей, которое каждое слово может иметь со словами, находящимися справа и слева и несколько общих грамматических правил. Ссылки вправо представляются как a+, а ссылки влево как a-. Необязательные ссылки заключаются в фигурные скобки. Неожиданные ссылки в любом количестве заключаются в квадратные скобки. Множественные ссылки соединяются посредством конъюнкции & или дизъюнкции or. Каждое правило заканчивается точкой с запятой.

Например, список базовых правил для языка SVO (подлежащее + сказуемое + объект) может иметь вид:

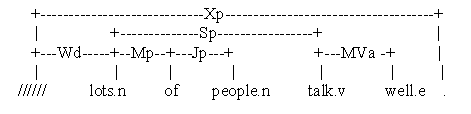
<артикль>: D+;

<существительное-субъект>: {D-}&S+;

<существительное-объект>: {D-}&O-

<сказуемое>: S-&{O+}

Тогда английское предложение «The man killed a shark.» («Человек убил акулу») должно быть представлено следующим образом (рис. 1):

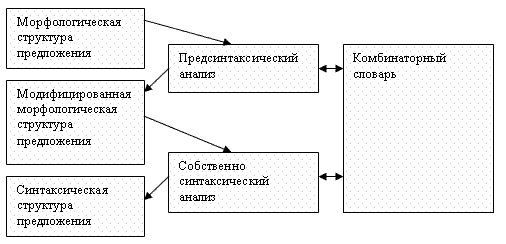
  
Рисунок 1. Схема по правилам Link Grammar

где Xp – связь между началом и концом предложения, Sp – связь между существительным и глаголом, Wd – связь между началом предложения и предложением, Mp - связь между именной группой и модифицирующей ее предложно-именной группой, Jp - связь между предлогом и относящейся к нему именной группой, MVa – связь между глаголом (прилагательным) и модификатором.

2. Лингвистический процессор Ю.Д. Апресяна, И.М. Богуславского и Л.Л. Иомдина.

Процессор основан на модели «Смысл-Текст». Он позиционируется как пригодный как минимум для всех индоевропейских языков.

Блок-схема синтаксического анализа этого процессора выглядит так (рис. 2):

  
Рисунок 2. Блок-схема синтаксического анализа в модели «Смысл-Текст»

Работа блока синтаксического анализа представляет собой синтаксическую структуру в виде дерева, вершинами которого являются слова с набором их морфологических признаков, а ветвями – типы синтаксических связей между ними.

На этапе предсинтаксического анализа устраняется морфологическая омонимия и происходит предварительная постройка синтаксических связей между словами. На этапе собственно синтаксического анализа строится окончательный вариант синтаксической структуры предложения.

Комбинаторный словарь представляет собой совокупность команд на специальном языке. Это набор правил, каждое из которых представляет некую структуру, начинающуюся с заголовка вида REG ИМЯ N, где ИМЯ – название совокупности правил, N – номер правила.

Правила бывают двух видов: элементарные и обобщенные (альтернативные). Элементарное правило состоит из двух частей: зона CHECK (проверить), содержащая условия, при выполнении которых выполняется следующая часть – зона DO, где описаны инструкции и действия, которые нужно выполнить.

Обобщенное правило состоит из зоны общих условий и нескольких элементарных подусловий. Если выполняются общие условия, то происходит проверка подусловий, являющихся альтернативными по отношению друг к другу. Если какое-то из условий выполняется, осуществляется выполнение действий из зоны DO этого подусловия.

Все условия в зоне CHECK представляют собой выражения логики предикатов. Действия из зоны DO записываются в виде операторов.

Данная структура напоминает инструкцию «if .. else» в различных языках программирования, поэтому интуитивно понятна для разработчиков в этой среде.

3. Модели, использующие структуры уровня именных и глагольных групп.

В данной модели важно правильно определить к какой группе относится слово (именной или глагольной). Далее определяются связи в группах и между группами и составляется синтаксическое дерево на основе распределенных ролей.

В этой модели имеются следующие группы и роли:

S – предложение

NP – именная группа (подлежащего)

VP – глагольная группа (сказуемого)

PP – предложная группа

N – существительное

V – глагол

AnV – аналитическая форма глагола

Aux – вспомогательный глагол

O – дополнение

Det – местоимение

A – прилагательное

Prep – предлог

Adv – наречие

По ним далее синтаксический анализатор строит синтаксическое дерево.

Существуют два вида синтаксических деревьев: 1) дерево составляющих; 2) дерево зависимостей.

1) Для данной структуры дерева единицами являются слова – составляющие. Они определяются тем, на каком месте стоят в предложении. Связь между словами определяется отношением вложенности, то есть каждая последующая часть предложения разбивается на другие отношения, что в итоге приводит к разбиению на базовые роли в предложении.

Синтаксические роли и правила построения дерева составляющих определяются с помощью КС-грамматики по Хомскому. КС-грамматика дерева может выглядеть следующим образом:

S -> NP VP NP

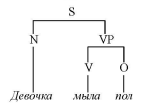
NP -> N | A N | Det N | N PP

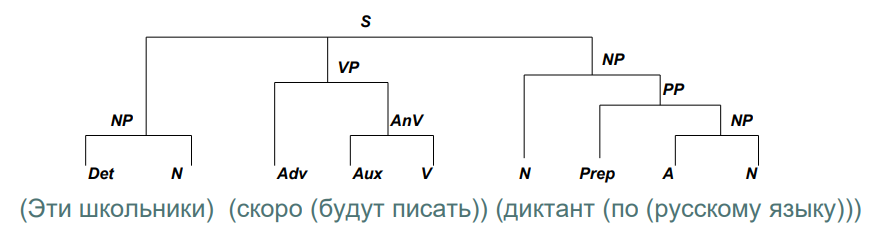
VP -> V | AnV | Adv V | Adv AnV

AnV -> Aux V

PP -> Prep NP

Заметим также, что составляющие могут вкладываться друг в друга, но не пересекаться. Это важно для составления самого дерева. Ниже представлены примеры готового дерева составляющих (рис. 3) (рис. 4).

  
Рисунок 3. Дерево составляющих для простого примера

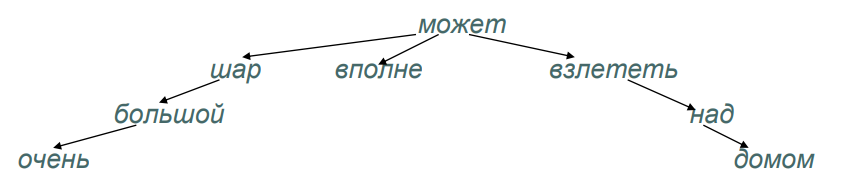
  
Рисунок 4. Дерево составляющих для более сложного примера

2) Основой связей между словами в дереве зависимостей являются подчинительные связи, основанные на грамматике естественного языка. В дереве узлами, или листьями, являются слова, а дугами, или ветвями, дерева – связи-зависимости. Корень дерева – это сказуемое (в редких случаях, подлежащее), от которого зависят остальные члены предложения.

При построении дерева зависимостей учитываются построенные связи и результат морфологического анализа, а также дополнительно может учитываться порядок слов в предложении.

Примеры дерева зависимостей приведены ниже (рис. 5) (рис. 6).

  
Рисунок 5. Дерево зависимостей для простого примера

  
Рисунок 6. Дерево зависимостей для более сложного примера

Для любого синтаксического анализатора на вход подается предложение на естественном языке и результат его морфологического анализа. При необходимости анализатор устраняет морфологическую омонимию, а затем выполняет синтаксический анализ предложения. На выходе, как мы уже видели, получается синтаксическое дерево одного из видов в зависимости от того, как работает сам анализатор.

На основе первой модели построен анализатор такой как Link Grammar Parser. Он написан на языке Си и поддерживает интерфейсы на английском и русском языках. Анализаторы второй модели основаны на синтаксической модели, которую мы рассматривали третьей. По принципам этой модели работает, например, синтаксический анализатор Natural Language Toolkit, написанный на языке программирования Python.

Далее я буду рассматривать именно последнюю модель построения синтаксического дерева, так как мне она кажется более понятной и является более распространенной.

## **2 Практический раздел**

Моя курсовая работа построена на таком методе научного исследования как моделирование.

В современной науке и технологии математическое моделирование усиливается, актуализируется проблемами, успехами других наук. Математическое моделирование реальных и нелинейных систем живой и неживой природы позволяет перекидывать мостики между нашими знаниями и реальными системами, процессами, в том числе и мыслительными.

Моделирование можно охарактеризовать как изучение объекта путем построения и исследования его модели, осуществляемое с определенной целью и состоящее в замене эксперимента с оригиналом экспериментом на модели.

Моделирование также предполагает использование специальных инструментов для создания модели. В нашем случае это будет синтаксический анализатор.

Таких систем довольно много на технологической арене, но мой выбор был сделан в пользу библиотеки Natural Language Toolkit (NLTK), написанной на языке Python и используемой вместе с ним для синтаксического анализа текстов.

NLTK широко используется для обучения студентов обработке естественного языка в сфере лингвистики или вычислительной техники. Для данной библиотеки созданы множество учебной литературы, есть детальная документация и учебник от самих создателей проекта. Все это способствует быстрому пониманию принципов работы NLTK и освоению этого инструмента на практике.

NLTK является уникальным пакетом программ, в плане предоставления всесторонних корпусов для развития вычислительного понимания языка. NLTK поддерживает токенизирование, морфологический поиск, тегирование, фрагментирование, синтаксический анализ, кластеризацию, классификацию, моделирование языка, семантическую интерпретацию, стандартизацию и другие функции.

NLTK предназначен, как говорили сами создатели проекта Лоупер и Берд, для обучения студентов, а также для создания ими проектов. Именно поэтому NLTK спроектирован так, чтобы по названию функций можно было догадаться, что именно делает эта функция. Простота и прозрачность структуры библиотеки обеспечивает большее понимание работы в сфере обработки естественного языка. NLTK используется для обучения в более чем 60 вузах в 20 странах и с помощью преподавательского опыта и опыта работы студентов развивается и совершенствуется для более удобной работы пользователей.

Чтобы начать работу с NLTK, нужно загрузить его на компьютер и далее в консоли Python выполнить следующие команды:



После этого можно начинать работу с текстом в программе.

Возьмем следующее предложение для синтаксического разбора:



С помощью функций word\_tokenize() и pos\_tag() библиотеки NLTK мы можем разделить предложение на слова и определить их синтаксические роли. Чтобы увидеть результат, воспользуемся стандартной функцией Python print().



Результат будет таковым:



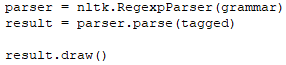
Здесь мы можем видеть, какие роли NLTK дал словам в нашем предложении.

Для того чтобы построить наше синтаксическое дерево, нам нужны синтаксические правила. NLTK предоставляет возможность самим определить синтаксические правила с помощью регулярных выражений Python, называемых грамматикой.

Для начала разобьем наше предложение на две группы: именную NP и глагольную VP. В таком случае наша грамматика будет выглядеть следующим образом:

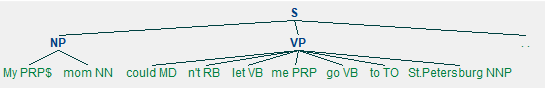


Для того, чтобы NLTK собрал и вывел нам дерево составляющих, запишем следующие строки в файл кода:



где RegexpParser() – функция-парсер регулярных выражений для составления дерева, parse() – функция составления самого дерева по группам, draw() – функция для прорисовки нашего дерева (графическое представление).

На рисунке ниже представлен результат написанного мной кода (рис. 7).

  
Рисунок 7. Дерево составляющих, именная и глагольная группа

Далее я буду изменять только строки грамматики (переменная grammar), чтобы добиться желаемого результата – полноценного синтаксического дерева с правильной структурой.

Именная группа в нашем случае далее меняться не будет, так как это ее конечное представление. Поэтому обратимся к разбору глагольной группы.

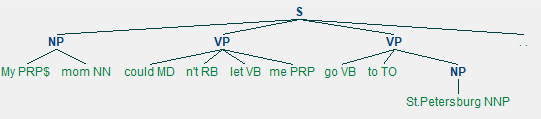
Составленное регулярное выражение отвечает только одной структуре. Ее можно модернизировать, чтобы получились более общие группы. Тогда регулярное выражение глагольной группы будет выглядеть следующим образом:



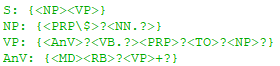
И также стоит заметить, что в именные группы могут объединиться и другие составляющие. Поэтому регулярное выражение именных групп можно записать следующим образом:



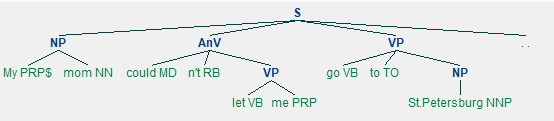
В таком случае синтаксическое дерево будет выводиться так (рис. 8):

  
Рисунок 8. Дерево составляющих, расширенные именная и глагольная группы

Первая глагольная группа на мой взгляд выглядит довольно громоздкой и незаконченной. Составляющие “could” и “n’t” связаны только общей группой, хотя в речи они считаются одной речевой единицей, поэтому перепишем нашу грамматику:



А теперь посмотрим, как выглядит наше дерево составляющих (рис. 9).

  
Рисунок 9. Дерево составляющих, окончательная синтаксическая структура рассматриваемого предложения

Теперь данное дерево больше похоже на действительно правильную структуру рассматриваемого нами предложения.

На этом мы и остановимся, так как разбор одного предложения на синтаксические группы может занять довольно много времени и ресурсов. Но мы достигли цели курсовой работы – построить синтаксическую структуру с помощью инструмента обработки естественного языка, поэтому данная структура будет окончательной. Полный текст кода на языке программирования Python приведен в приложении Б.

NLTK, как инструмент обработки естественного языка, достаточно прост для понимания, но для обработки больших текстов с его помощью понадобится довольно много временных и умственных ресурсов, чтобы синтаксический разбор был выполнен в полной мере правильно, для чего нужна командная работа лингвистов и программистов.

# ЗАКЛЮЧЕНИЕ

В ходе работы мной были изучены принципы синтаксического анализа, как со стороны лингвистики, так и со стороны компьютерных технологий. Для этого было произведено исследование в сети Интернет по поиску развернутой и полной информации по обоим направлениям, были найдены и проанализированы источники нужной информации, что способствовало более точному пониманию объекта и предмета исследования курсовой работы.

Также был произведен поиск инструментов для обработки естественного языка. Каждый инструмент был проанализирован на предмет наличия возможности синтаксического анализа предложений, чтобы выполнить цель курсовой работы. В ходе этого этапа работы была выбрана и изучена библиотека NLTK для языка Python.

Далее было смоделировано с помощью выбранного инструмента синтаксическое дерево предложения, выбранного в качестве примера, и построено графическое представление данного дерева, то есть была получена модель синтаксической структуры предложения.

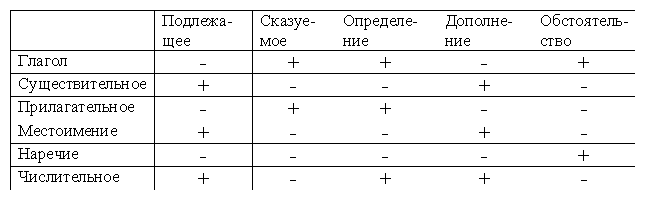
Из проделанной работы я могу сделать следующие выводы:

1. Моделирование синтаксических структур в рамках обработки естественного языка довольно сложное и обширное направление, требующее углубленных знаний как со стороны лингвистики, так и со стороны компьютерных технологий.
2. Библиотека NLTK является хорошим инструментом для понимания разбора и обработки синтаксических структур, в том числе и для графического представления данных структур, но также требует хороших познаний в сфере компьютерной лингвистики.
3. Полученная модель данных синтаксического анализа является достаточной для понимания принципов синтаксической обработки естественного языка, но при этом не является идеальной структурой в связи с недостатком знаний в сфере лингвистики.

# СПИСОК ЛИТЕРАТУРЫ

1. Автоматическая обработка текстов на естественном языке и анализ данных: учеб. пособие / Большакова Е.И., Воронцов К.В., Ефремова Н.Э., Клышинский Э.С., Лукашевич Н.В., Сапин А.С. — М.: Изд-во НИУ ВШЭ, 2017. — 269 с.
2. Информационные технологии в лингвистике: учеб. пособие / Л.Ю. Щипицина. — М.: ФЛИНТА: Наука, 2013. — 128 с.
3. Понимание и синтез текста компьютером [Электронный ресурс] / Чесебиев И.А. – 2009 – URL: http://compuling.narod.ru/index2.html
4. Автоматический анализ и синтез текста [Электронный ресурс] / Автор не известен – 2012 – URL: http://csaa.ru/vtomaticheskij-analiz-i-sintez-teksta/
5. Обработка естественного языка [Электронный ресурс] / Wikimedia Foundation, Inc. – 2020 – URL: https://ru.wikipedia.org/wiki/Обработка\_естественного\_языка
6. Link Grammar Parser [Электронный ресурс] / Усталов Д. NLPub: каталог и сообщество русских лингвистических ресурсов – 2015 – URL: https://nlpub.ru/Link\_Grammar\_Parser
7. Синтаксический анализ [Электронный ресурс] / Усталов Д. NLPub: каталог и сообщество русских лингвистических ресурсов – 2018 – URL: https://nlpub.ru/Синтаксический\_анализ#.D0.93.D1.80.D0.B0.D0.BC.D0.BC.D0.B0.D1.82.D0.B8.D0.BA.D0.B0\_.D1.81.D0.B2.D1.8F.D0.B7.D0.B5.D0.B9
8. АВТОМАТИЧЕСКАЯ ОБРАБОТКА ТЕКСТОВ: ЗАДАЧИ, ПОДХОДЫ, РЕСУРСЫ [Электронный ресурс] / Большакова Е.И. – 2017 – URL: https://www.hse.ru/data/2017/07/24/1173868389/КЛ\_Задачи\_Подходы\_Ресурсы.pdf
9. Программное обеспечение для обработки естественного языка [Электронный ресурс] / Wikimedia Foundation, Inc. – 2020 – URL: https://ru.wikipedia.org/wiki/Программное\_обеспечение\_для\_обработки\_естественного\_языка
10. NLTK 3.5 documentation [Электронный ресурс] / NLTK Project – 2020 – URL: https://www.nltk.org/
11. Natural Language Toolkit [Электронный ресурс] / Wikimedia Foundation, Inc. – 2019 – URL: https://ru.wikipedia.org/wiki/Natural\_Language\_Toolkit
12. Natural Language Processing with Python [Электронный ресурс] / Стивен Берд, Эван Кляйн, Эдвард Лоупер – 2019 – URL: http://www.nltk.org/book/
13. NLTK: перспективный синтаксический парсер или … [Электронный ресурс] / Антон Бузанов – 2020 – URL: https://hum.hse.ru/data/2020/02/09/1574924211/NLTK.pdf
14. AI с Python – пакет NLTK [Электронный ресурс] / Автор не известен – 2019 – URL: https://coderlessons.com/tutorials/python-technologies/izuchit-iskusstvennyi-intellekt-s-python/ai-s-python-paket-nltk
15. ОБРАБОТКА ЕСТЕСТВЕННОГО ЯЗЫКА (NATURAL LANGUAGE PROCESSING) ПРИ ИСПОЛЬЗОВАНИИ ТЕХНОЛОГИИ NLTK (NATURAL LANGUAGE TOOLKIT) НА БАЗЕ ЯЗЫКА ПРОГРАММИРОВАНИЯ PYTHON [Электронный ресурс] / Кривальцевич Е.В. – 2011 – URL: http://elib.bsu.by/bitstream/123456789/110358/1/Кривальцевич%20Е.В.%20ОБРАБОТКА%20ЕСТЕСТВЕННОГО%20ЯЗЫКА%20%28NATURAL%20LANGUAGE%20PROCESSING%29%20ПРИ%20ИСПОЛЬЗОВАНИИ%20ТЕХНОЛОГИИ%20NLTK%20%28NATURAL%20LANGUAGE%20TOOLKIT%29%20НА%20БАЗЕ%20ЯЗЫКА%20ПРОГРАММИРОВАНИЯ%20PYTHON.pdf
16. ФОРМАЛЬНЫЕ ЯЗЫКИ И ГРАММАТИКИ [Электронный ресурс] / Автор не известен – 2013-2020 – URL: https://studopedia.su/19\_20467\_sintaksicheskoe-derevo.html
17. Моделирование как метод исследования [Электронный ресурс] / Автор не известен – 2015-2020 – URL: https://mydocx.ru/12-50243.html

# ПРИЛОЖЕНИЕ А



# ПРИЛОЖЕНИЕ Б

