Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра интеллектуальных информационных технологий

Лабораторная работа №2 по курсу «ЕЯзИИС» на тему:
«Представление результатов синтаксического анализа в памяти интеллектуальной системы»
Вариант 5

Выполнили студенты группы 021701: Кулак П.О., Седеневский А.М., Малаев А.А

Проверил: Крапивин Ю.Б.

МИНСК
2023

Цель работы – закрепить знания синтаксического анализа текста.

Основные **задачи** работы следующие:

1. Изучить теоретический материал, необходимый для решения задачи автоматического синтаксического анализа текста естественного языка.
2. Закрепить навыки программирования естественно-языковых систем и вспомогательных прикладных программ на одном из языков программирования.

Вариант словаря:

№ варианта	Язык текста	Формат документа
5	Русский	pdf

Ход выполнения:

Для выполнения лабораторной работы был выбраны модули pymorphy2 (Pymorphy2 написан на языке Python (работает под 2.7 и 3.5+)) и nltk.

Pymorphy2 умеет:

1. приводить слово к нормальной форме (например, “люди -> человек”, или “гулял -> гулять”).
2. ставить слово в нужную форму. Например, ставить слово во множественное число, менять падеж слова и т.д.
3. возвращать грамматическую информацию о слове (число, род, падеж, часть речи и т.д.)

При работе используется словарь OpenCorpora; для незнакомых слов строятся гипотезы. Библиотека достаточно быстрая: в настоящий момент скорость работы - от нескольких тыс слов/сек до > 100тыс слов/сек (в зависимости от выполняемой операции, интерпретатора и установленных пакетов); потребление памяти - 10…20Мб; полностью поддерживается буква ё.

NLTK

NLTK - ведущая платформа для создания программ Python для работы с данными на человеческом языке. Он предоставляет простые в использовании интерфейсы для более чем 50 корпусных и лексических ресурсов, таких как WordNet, а также набор библиотек обработки текста для классификации, токенизации, стемминга, тегирования, синтаксического анализа и семантического анализа, оболочки для промышленных библиотек NLP, и активный дискуссионный форум.

Благодаря практическому руководству, представляющему основы программирования наряду с темами вычислительной лингвистики, а также исчерпывающей документации по API, NLTK подходит как для лингвистов, инженеров, студентов, преподавателей, исследователей, так и для пользователей отрасли. NLTK доступен для Windows, Mac OS X и Linux. Лучше всего то, что NLTK - это бесплатный проект с открытым исходным кодом, управляемый сообществом.

NLTK был назван «прекрасным инструментом для обучения и работы в области компьютерной лингвистики с использованием Python» и «потрясающей библиотекой для игры с естественным языком».

Для работы был выбран текст с описанием первого кинетоскопа.

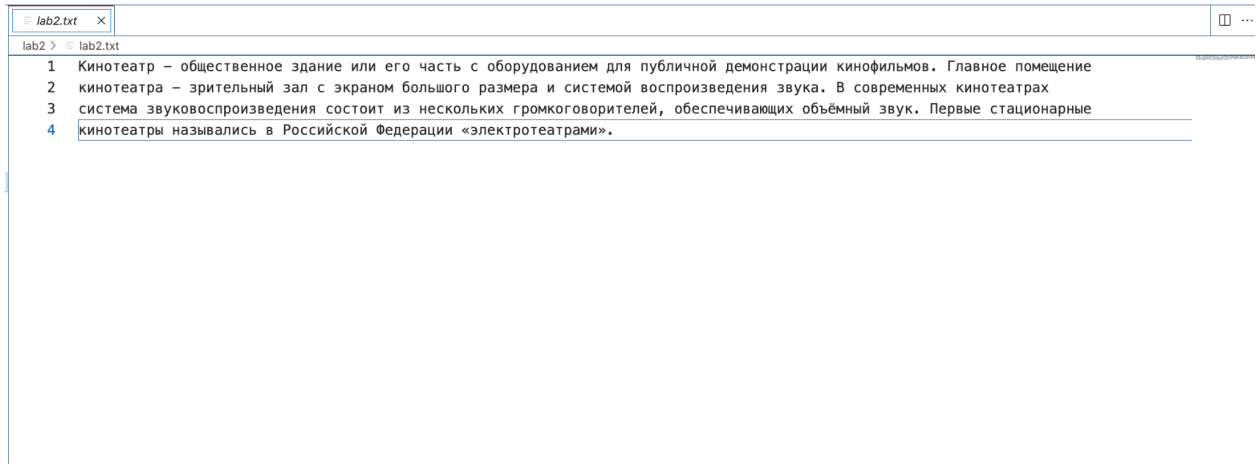

Рис 1. Исходный текст

Был спроектирован внешний интерфейс автоматизированной системы формирования синтаксического дерева предложения.

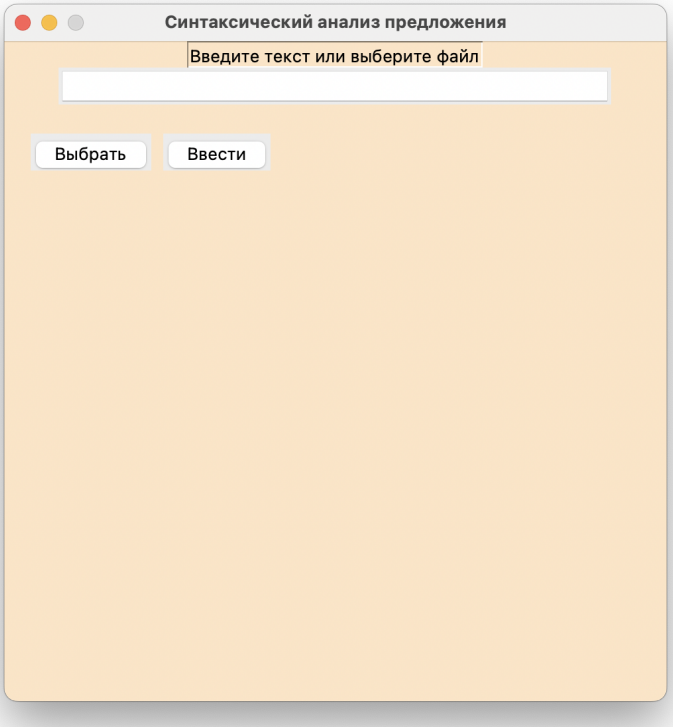

Рис 2. Внешний интерфейс

Данная программа позволяет как выбрать текстовый файл, так и ввести предложения и программа сохранит результат в текстовом файле sentence_tree.txt.
Интерфейс предельно прост и интуитивно понятен любому пользователю.

Для формирования дерева были использованы следующие методы:

```
>>> p.tag.POS          # Part of Speech, часть речи
```

Метод библиотеки pymorphy2 для определения части речи.

```
>>> grammar.productions()
```

Метод библиотеки nltk, которая и строит синтаксическое дерево.

В результате создаётся txt-файл с деревьями текста с тем же путём, что и исходный файл с содержимым текстом.

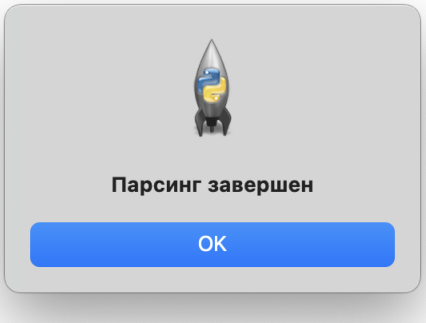

Рис. 3 Окно успешного завершения файла

```
  Кинотеатр  общественное здание или его часть с оборудованием для публичной демонстрации кинофильмов
Предложение -> Именная группа Глагольная группа
  Глагольная группа -> Глагол Именная группа
  Глагольная группа -> Глагол Именная группа Предложная группа
  Предложная группа  -> Предлог Именная группа
  Глагол ->
  Именная группа -> Детерминатор Существительное
  Именная группа -> Детерминатор Существительное Предложная группа
  Детерминатор -> с
  Детерминатор -> для
  Существительное -> Кинотеатр
  Существительное -> здание
  Существительное -> его
  Существительное -> часть
  Существительное -> оборудованием
  Существительное -> демонстрации
  Существительное -> кинофильмов
  Предлог -> общественное
  Предлог -> или
  Предлог -> публичной
  Главное помещениекинотеатра  зрительный зал с экраном большого размера и системой воспроизведения звука
Предложение -> Именная группа Глагольная группа
  Глагольная группа -> Глагол Именная группа
  Глагольная группа -> Глагол Именная группа Предложная группа
  Предложная группа  -> Предлог Именная группа
  Глагол ->
  Именная группа -> Детерминатор Существительное
  Именная группа -> Детерминатор Существительное Предложная группа
  Детерминатор -> с
  Существительное -> Главное
  Существительное -> помещениекинотеатра
  Существительное -> зал
  Существительное -> экраном
  Существительное -> размера
  Существительное -> системой
  Существительное -> воспроизведения
```

Рис 4. Результат работы

Вывод:

В ходе данной лабораторной работы была реализована программа создания дерева предложения. Данная программа позволяет создать дерево предложения соответсвенно. Время выполнения программы, обрабатывающая файл, состоящий из 4 предложений и 45 слов, составляет 0.11771225929260254 мс..

5. Автоматизация обработки текста: этап синтаксического анализа.
Синтаксический анализ является одним из важных этапов в автоматизации обработки текста. Он заключается в анализе структуры предложения и определении отношений между словами в предложении.
На этапе синтаксического анализа происходит разбор предложения на составляющие его части, то есть определение, какие слова являются подлежащими, сказуемыми, дополнениями и т.д. Это делается с помощью синтаксических анализаторов - программных компонентов, которые на основе грамматических правил определяют структуру предложения.
Синтаксический анализ необходим для решения многих задач обработки текста, таких как автоматический перевод, извлечение информации, анализ тональности и многих других. Он позволяет понимать смысл предложения, а не только отдельных слов, что делает его очень важным этапом в обработке текста.
Одним из способов синтаксического анализа является метод деревьев составляющих (constituency parsing), который заключается в построении дерева, отображающего структуру предложения. В этом дереве каждому слову присваивается свой узел, а связи между узлами отображают отношения между словами в предложении. Другой способ - метод зависимостей (dependency parsing), в котором каждому слову присваивается зависимое и главное слово, отображающее связь между ними.
Синтаксический анализ может быть достаточно сложным, особенно в случае с большими и сложными предложениями. Но современные методы и технологии могут справляться с этой задачей достаточно эффективно и точно.

6. Способы описания синтаксической структуры предложения. Достоинства и недостатки.
Существует несколько способов описания синтаксической структуры предложения. Рассмотрим некоторые из них:

1. Диаграммы дерева составляющих (constituency parse trees). В такой диаграмме каждому слову в предложении соответствует вершина дерева, а связи между вершинами отражают синтаксические связи между словами в предложении. Достоинством такого описания является его наглядность, так как дерево составляющих позволяет увидеть все синтаксические связи между словами в предложении. Однако недостатком является сложность построения такой диаграммы, особенно в случае с большими и сложными предложениями.
2. Описание синтаксической структуры в виде зависимостей между словами (dependency parsing). В таком описании каждому слову в предложении соответствует узел графа, а связи между узлами отражают зависимости между словами в предложении. Достоинством такого описания является его относительная простота и наглядность, а также возможность использования графического представления. Однако недостатком является то, что в таком описании не учитываются составляющие предложения.
3. Синтаксические правила и грамматики. В таком описании синтаксической структуры предложения используются формальные правила, которые определяют возможные комбинации слов в предложении. Достоинством такого описания является его точность и формальность, а также возможность использования для автоматической обработки текста. Однако недостатком является сложность создания таких правил и грамматик, особенно для языков с большим количеством исключений и нестандартных конструкций.

Каждый из этих способов имеет свои достоинства и недостатки, и выбор конкретного способа зависит от конкретной задачи и требований к описанию синтаксической структуры предложения.

7. Системы составляющих.

Системы составляющих (constituency parsing) являются методом автоматического анализа синтаксической структуры предложений естественного языка. Они позволяют разбить предложение на составляющие (конституенты) и определить синтаксические связи между ними.

Системы составляющих используют грамматики, которые описывают правила комбинирования слов в предложениях. Грамматика может быть контекстно-свободной (CFG), расширенной контекстно-свободной (ECFG) или зависимостной (DG). Задача состоит в том, чтобы найти дерево разбора, которое соответствует заданному предложению и которое является корректным с точки зрения грамматики.

Системы составляющих могут использоваться в различных приложениях, таких как машинный перевод, оптическое распознавание символов, анализ тональности и других. Они могут быть использованы для извлечения семантических свойств предложения, таких как субъект, объект, действие и т.д.

Одним из известных алгоритмов для решения задачи составления составляющих является алгоритм CYK (Cocke-Younger-Kasami), который работает со словарем слов и грамматикой. Алгоритм CYK имеет полиномиальную сложность от длины входного предложения.

Недостатком систем составляющих является то, что они не учитывают зависимости между словами в предложении и не могут определить порядок слов в предложении. Эти недостатки могут быть устранены с помощью других методов анализа, таких как синтаксический анализ зависимостей.

8. Деревья подчинения.

Деревья подчинения (dependency trees) представляют собой структуру данных, которая используется для описания синтаксической структуры предложения на естественном языке. Они моделируют отношения между словами в предложении, указывая, как каждое слово зависит от других слов в предложении.

В дереве подчинения каждое слово представлено узлом, а связи между словами представлены направленными дугами. Каждая дуга указывает, какое слово является главным (зависимым) и какое слово является зависимым (подчиненным). Например, в предложении "Кот ловит мышь", слово "ловит" является главным (зависимым) для слова "кот", а слово "мышь" является главным (зависимым) для слова "ловит".

Деревья подчинения могут быть построены с помощью алгоритмов анализа зависимостей. Они могут использоваться в различных приложениях, таких как машинный перевод, анализ тональности, информационный поиск и другие.

Достоинством деревьев подчинения является то, что они учитывают зависимости между словами в предложении и могут определять порядок слов в предложении. Они также обладают более простой структурой, чем системы составляющих, что облегчает их использование в различных приложениях.

Недостатком деревьев подчинения является то, что они не могут описать некоторые сложные синтаксические конструкции, такие как вложенные предложения. Они также могут иметь проблемы с описанием семантических свойств предложения, таких как субъект и объект.

2. Задачи создания естественно-языкового интерфейса включают в себя: 1. Понимание естественного языка

2. Генерация естественного языка

3. Анализ намерений и контекста

4. Повышение точности

5. Интеграция с другими системами

6. Тестирование и оптимизация

3. Отличия естественно-языкового интерфейса от других видов интерфейсов: - позволяет взаимодействовать с компьютером при помощи естественного языка - более доступен для людей с ограниченными возможностями или языковыми барьерами - не требует знания программирования

- более гибок и адаптивен к новым запросам пользователей и контексту

- может предоставлять пользователю более широкий диапазон возможностей и функциональности