**ДОКЛАД**

**Цель работы -** разработка правил конвертации текста на естественном языке в формат для обмена данными UML, Создание программного продукта для автоматизированной конвертации текста с последующим сохранением в формате XMI.

**Практическое значение** этой дипломной работы составляет, получения и использования UML диаграмм в дальнейшем анализе и редактировании текста, с целью конвертации в OWL формат (язык онтологий) - дипломная работа Александра Василейко

Данный дипломный проект сосредотачивает в себе работу с семантически насыщенным коротким текстом, с использованием библиотеки Stanford Core NLP, разработки правил конвертирования текста из одного виде в другой, а также создания UML диаграмм на основе этого текста.

**Слайд 3**

**Состоит из 3-х глав**

1. **ОБЗОР ОБРАБОТКИ ЕСТЕСТВЕННОГО ЯЗЫКА СРЕДСТВАМИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

* Семантический анализ текста как основа для выполнения парсинга
* Обзор существующего инструментария Stanford Core NLP
* Средства хранения и визуального представления текста
* Вектор развития данного направления и его перспективы

В этом разделе я рассмотрела основные технологии, языка, библиотеки и инструментарий для работы с Natural Language Processing. Выполнила их сравнение с учетом недостатков и преимуществ.

**Слайд 4**

Ключевым етапом етого раздела било изучить библиотеку **Stanford Core NLP** и возможности которие она предоставляет

**Слайд 5**

Результатом роботы библиотеки CoreNLP является дерево в виде NP, VP зависимостей и проаннатированные **частями речи**

Этот этап является неотъемлемой частью так как он позволяет адекватно произвести оценку и предусмотреть приблизительные результаты, учитывая недостатки и преимущества того или иного подхода.  
Именно в следующем разделе я собираюсь использовать полученные данные для дальнейшего внедрения функционала касательно дипломного проекта.

**2.РЕАЛИЗАЦИЯ РАСПОЗНАВАНИЕ ТЕКСТА И СОЗДАНИЕ UML ДИАГРАММ**

**Слайд 6**

* Специфика и особенности реализации
* Разработка правил конвертации текста
* Механизм построения UML диаграмм
* Робота з форматом XMI

В этом разделе описана специфика и особенности реализации программы парсера, технологии, языка и среды разработки которые были использованы в процессе разработки. Был рассмотрен механизм построения UML диаграмм, описаны алгоритмы для извлечения классов из дерева зависимостей которое предоставляет Core NLP. Также в етом разделе были разработаны правила для конвертации частей речи в UML сущности. В разделе 2.3. рассматривается работа с форматом XMI, внедрения и использования пакета javax.xml для записи XMI сущностей в файл. В разделе 2.4. описана логика работы программы и основные архитектурные особенности парсеру, также в этом разделе описано внедрение библиотеки Core NLP и ее использования. Также была приведена логика построения UML диаграммы путем преобразования элементов промежуточного графу в UML элементы.

**Слайд 7**

Сразу хотела сделать акцент на том что в этом дипломном проекте основной структурой данных является ГРАФ

Как видите ми получаем зависимости NP, VP

Поетому основной идеей парсинга есть представить

NP – узлы

VP – ребра

Которие также в себе деревья

Так били разработани правила конвертации полученного дерева CoreNLP в промежуточный граф

**Слайд 8**

Результат конфертации дерева в промежуточний граф

**Слайд 9**

Следующими правилами било разработать правила преобразования промежуточного графа в UML граф **ТАБЛИЦА**

Основываясь на детальном парсинге каждого дерева

public static final String *singleNounSet*[] = new String[] {"NN", "NNP", "PRP"};  
 public static final String *pluralNounSet*[] = new String[] {"NNS"};  
  
 public static final String *adjectiveSet*[] = new String[] {"JJ", "CD", "RB"};  
 public static final String *verbSet*[] = new String[] {"VBP", "VBN", "VBG", "IN", "TO", "VBZ", "ADVP", "VB"};  
 public static final String *joinVerbSet*[] = new String[] {"ADJP", "PP", "SBAR"};  
 public static final String *conjVerbSet*[] = new String[] {"CC", ","};  
  
 public static final String *aggregationSet*[] = new String[] {"IN"}; //with  
 public static final String *generalizationSet*[] = new String[] {"IN"}; // of  
  
 public static final String *generalizationSetWords*[] = new String[] {"of"};

**Слайд 10**

Результат конвертации промежуточного графа в UML граф

**узлы** - жирным шрифтом виделение

**ребра** - серым

**Слайд 11**

**Диаграмма генерации UML моделей**на следующем этапе происходит финальная конвератция в список UML едементов и сохранение UML модели в файл .xmi

**Слайд 12**

полученние результаты

**Слайд 13**

Подитоживание

**Преимущества**

* Применение библиотеки Stanford CoreNLP дает широкие возможности выполнения различных типов парсинга.
* Обработка полученного дерева через выделение основных узлов NP и VP без четкой привязки к каждому типу зависимости.
* Использование JgraphT графу как основной структуры данных.  
  Быстрая обработка небольших объемов текстов.
* Использование менеджера зависимостей Maven. Этот менеджер позволяет подключать любые библиотеки которые находятся в открытом доступе.

**Недостатки**

* Stanford CoreNLP может показывать непредсказуемые результаты парсинга, это зависит от многих факторов например, ошибки в тексте, неправильно поставленные знаки препинания и другие.
* Так как обработка текстов может занимать достаточно много времени. В будущем необходимо провести оптимизацию парсинга большого объема текстов.
* Текущий подход к конвертации дерева зависимостей между частями речи хоть и является универсальным, но не является достаточно надежным. В отдельных случаях мы можем получить достаточно непредсказуемые результаты, которые могут очень сильно отличаться от ожидаемых.
* В настоящее время до сих пор не существует четких правил конвертации текста, поэтому довольно трудно вывести успешную формулу для конвертации естественного языка в UML диаграммы.
* Очень большая зависимость от результатов парсинга Stanfrod Core NLP. В будущем можно рассмотреть альтернативное или комплексное решение (например, использование дополнительных библиотек или оптимизацию Core NLP)

**3.ЭКСПЕРИМЕНТАЛЬНАЯ ПРОВЕРКА РЕЗУЛЬТАТОВ РАБОТЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

**Слайд 14**

* План і проведення експерименту
* Перевірка конвертації текстових сутностей в UML елементи
* Аналіз та оцінка проведеного експерименту

**Слайд 15 – 19**

Сравнения

**Слайд 20**

**Вывод**

Программа парсер выполняет базовые функции конвертации текста на естественном языке в UML диаграммы согласно правилам, но успешность конвертации зависит от многих факторов:

* объем текста;
* его корректность с точки зрения орфографии и семантики;
* результаты парсинга Stanford Core NLP;

- корректность правил конвертации.

**XMI** (XML Metadata Interchange) является стандартом OMG для использования языка разметки (XML), который предназначен для обеспечения стандартного способа обмена метаданными [5]. В частности, XMI помогает программистам используя Unified Modeling Language (UML) обмениваться моделями данных. Кроме того, XMI может использоваться для обмена информацией в хранилищах данных. Фактически, формат XMI стандартизирует любой набор метаданных и требует одинакового отображения этих данных в различных областях использования.

Extensible Markup Language (**XML**) используется для описания данных. Стандарт XML представляет собой гибкий способ для создания информационных форматов в электронном виде для обмена структурированными данными через общедоступные и корпоративные сети такие как Интернет

**Unified Modeling Language (UML)** содержит в себе стандартизированные визуальные обозначения, которые могут быть использованы для представления нужной информации в надлежащем виде.

**Tokenize, split, pos, lemma, pasre**

**Имя существительное - именник**

**Прилагательное -** прикметник

**Глагол -** дієслово

**Наречие** – присливник

**Метсоимение** – займенник

Диаграммы классов

Табличние

Ассоцияция

Аггрегация

Генерализация

Композиция

Whose woods these are I think I know.

His house is in the village though;

He will not see me stopping here

To watch his woods fill up with snow.

My little horse must think it queer

To stop without a farmhouse near

Between the woods and frozen lake

The darkest evening of the year.

He gives his harness bells a shake

To ask if there is some mistake.

The only other sound’s the sweep

Of easy wind and downy flake.

The woods are lovely, dark and deep,

But I have promises to keep,

And miles to go before I sleep,

And miles to go before I sleep.

Чей этот бор известно мне,  
Я видел дом его в селе –  
Меня он не заметит здесь,  
Разглядывая снежный лес.  
  
Моей лошадке не понять  
Зачем мы стали здесь опять  
Среди дубравы, льда покров  
В один из темных вечеров  
  
Трясет упряжкой, вопрошает -  
Ошибка вышла ли какая.  
Один лишь ветра ровный свист  
И шорох хлопьев тут повис.  
  
Бор этот темен и глубок  
Влечет меня пучком дорог  
Но мне еще обет сдержать  
И сотни миль идти - не спать.

**3 Били взяты разние тексти**

**- одни спец подготовлени и отформатированы и отформатированы**

**- а другие нет**

* **Создание различных как положительных так и отрицательных условий для работы программного обеспечения**
* **Тестирование разработанного программного обеспечения моделируя различные события развития**
* **Анализ и оценка работы отдельного модуля**
* **Определение узких мест и возможности их оптимизации**

Підсумовуючи проведену роботу, можна стверджувати, що основна мета була досягнута, але це ніяк не означає що отриманий результат є ідеальним. Наступним у подальшому розвитку даного ПЗ буде, розширеня правил конвертації тектсу у UML діаграму, удосконалення та оптимізація алгоритму перетворення тексту у UML, а також розширення бібліотеки Stanford Core NLP.

**1**

Цей дипломний проект зосереджує у собі роботу з семантично насиченим коротким текстом, з використанням бібліотеки Stanford Core NLP, а також створення UML діаграм на основі цього тексту.

Практичне значення цієї дипломної роботи становить, отримання та використання UML діаграм у подальшому аналізі та редагуванні тексту, з метою конвертації у OWL формат (мова онтологій) — дипломна робота [2].

**2**

В первой главе работы рассмотрены краткие теоретические ведомости направления использования NLP и Обзор существующего инструментария

Одним из важнейших направлений использования NLP является разработка способности  
компьютерной программы "понимать" естественный язык используя классификацию  
  различных лингвистических единиц (текстов, слов, словосочетаний, предложений), реализуемой  
практически во всех приложениях лингвистического процессора, информационном поиске, машинном переводе,  
автоматическом реферирования и др.

**3.** Расммотреть та Огляд існуючого інструментарію   
Stanford Core NLP

Складність роботи з Natural Language Processing і недоліки існуючих бібліотек

**4** в частности были рассмотрена библиотека **Stanford Core NLP**

Стенфорд CoreNLP надає широкий набір інструментів для аналізу природної мови. Вона може надати базові форми слів, частини мови, абревіатури, власні назви, нормалізовані дати, час і числові величини. Також ця бібліотека може розпізнати структуру речення у термінах фраз та залежності слів. Вказує яка фраза відноситься до тієї ж сутності, вказує настрій, а також розуміє цитати [4].

Вона містіть такі модулі аналізу тексту (Анататори) Додаток Д, які підтримують наступні мови таблиця

**5** Аннататори с помощью них и происходи парсинг текста

я использовала такие такие чтоб получить такие результати

дерево вида

У цьому розділі я розглянула основні технології, мови, бібліотеки та інструментарій для роботи з Natural Language Processing. Виконала їх порівняння з урахуванням недоліків та переваг.

Цей етап є невід’ємною частиною для подальшого розвитку дипломного проекту. Адже він дозволяє адекватно зробити оцінку та передбачити приблизні результати, враховуючи недоліки та переваги одного чи іншого підходу.

Саме у слідуючому розділі я збираюсь використовувати отримані дані з метою подальшого впровадження функціоналу у дипомному проекті.

В первой главе работы была рассмотрена следующая задача ...»,

«Получены следующие результаты решения задачи ...