**Добрый день уважаемы члены комиссии!**

**Квалификационную работу на тему:**

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ   
ДЛЯ ПАРСИНГА ТЕКСТОВ   
И ГЕНЕРАЦИИ UML МОДЕЛЕЙ

**Цель работы -** Разработка алгоритмического и программного обеспечения для получения структурированных представлений знаний из коротких семантически насыщенных текстов на естественном (английском) языке.

**Задачи**

* разработка эвристики и алгоритма преобразования текстов на естественном языке в модель представления данных на языке UML диаграммы классов
* разработать ПО реализующее эти преобразования в репрезентацию на языке XMI

**Эвристический алгоритм** (эвристика) — [алгоритм](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC) решения задачи, включающий практический метод, не являющийся гарантированно точным или оптимальным, но достаточный для решения поставленной задачи. Позволяет ускорить решение задачи в тех случаях, когда точное решение не может быть найдено.

**Практическое значение** этой дипломной работы составляет, получения и использования UML диаграмм в дальнейшем анализе и редактировании текста

**SLIDE 3**

**Java**

**Maven –** менеджер зависимостей

**Stanford Core NLP**

**JGraphT**

**JAXB**

**XMI, XML, UML**

**SLIDE 4 / 5**

**Раздел1** CoreNLP, XMI, XML, UML

SLIDE 6

Если сравнить с предыдущие решения реализации парсинга то можно увидеть что зачастую использовали привязку к зависимостям

**SLIDE 7**

|  |  |
| --- | --- |
| **Части речи** | **UML сущности** |
| **NN, NNP, PRP, NNS** | Классы |
| **JJ, CD, RB** | Атрибуты классов |
| **VBP, VBN, VBG, IN, TO, VBZ, ADVP, VB** | Зависимости между классами (ассоциация, агрегация, генерализация) |
| **ADJP, PP, SBAR** | Дополнительная информация для зависимостей между классами, которая влияет на их последующую конвертацию |
| **IN** | Указывает на агрегацию или генерализацию в зависимости контекста |
| **СС** | Соединение одинаковых по типу зависимостей |

**SLIDE 8**

Детальное представление применения правил преобразования

Часы - это временный инструмент для создания экземпляров временной меры.

**SLIDE 9**

Анализ и маппинг соответствия

**SLIDE 10**

Данный подход не будет работать я большим объемами текстов

**SLIDE 11 ВЫВОДЫ**

* разработаны правила преобразования текста в модель представления данных на языке UML диаграммы классов **раздел 2 пункт 2.2.2, слайд 7;**
* разработан алгоритм трансформации текста в UML диаграммы **раздел 2 пункт 2.2.1**;
* разработано ПО выполняющее базовые функции преобразования текста на естественном языке в UML диаграммы.

**UML классы**

**Табличные** – те что мы использаем

**Конечные** - для пользователя

**Управленческие** - методы

**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)

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"};

[**Ассоциация**](https://ru.wikipedia.org/wiki/%D0%90%D1%81%D1%81%D0%BE%D1%86%D0%B8%D0%B0%D1%86%D0%B8%D1%8F_%28%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%29) показывает, что объекты одной сущности (класса) связаны с объектами другой

[**Агрегация**](https://ru.wikipedia.org/wiki/%D0%90%D0%B3%D1%80%D0%B5%D0%B3%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%28%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%29)— это разновидность ассоциации при отношении между целым и его частями. **Профессор +------- студенты**

[**Композиция**](https://ru.wikipedia.org/w/index.php?title=%D0%9A%D0%BE%D0%BC%D0%BF%D0%BE%D0%B7%D0%B8%D1%86%D0%B8%D1%8F_%28%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%29&action=edit&redlink=1)— более строгий вариант агрегации. Закрашенный квадрат

**Комната является частью квартиры**, следовательно здесь подходит **композиция**, потому что комната без квартиры существовать не может. А, например, **мебель** не является неотъемлемой частью **квартиры**, но в то же время, квартира содержит мебель, поэтому следует использовать агрегацию.

**Сема́нтика**— [раздел](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B7%D0%B4%D0%B5%D0%BB) [лингвистики](https://ru.wikipedia.org/wiki/%D0%9B%D0%B8%D0%BD%D0%B3%D0%B2%D0%B8%D1%81%D1%82%D0%B8%D0%BA%D0%B0), изучающий [смысловое](https://ru.wikipedia.org/wiki/%D0%A1%D0%BC%D1%8B%D1%81%D0%BB) [значение](https://ru.wikipedia.org/wiki/%D0%97%D0%BD%D0%B0%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) [единиц языка](https://ru.wikipedia.org/wiki/%D0%95%D0%B4%D0%B8%D0%BD%D0%B8%D1%86%D0%B0_%D1%8F%D0%B7%D1%8B%D0%BA%D0%B0).

**XMI** (XML Metadata Interchange) стандарт [OMG](https://ru.wikipedia.org/wiki/Object_Management_Group) для обмена метаданными с помощью языка [XML](https://ru.wikipedia.org/wiki/XML). Может использоваться для любых метаданных, если их метамодель может быть выражена с помощью [MOF](https://ru.wikipedia.org/wiki/Meta-Object_Facility) (Meta-Object Facility).[[1]](https://ru.wikipedia.org/wiki/XMI#cite_note-omg-1) Наиболее часто XMI применяется как формат обмена [UML](https://ru.wikipedia.org/wiki/UML)-моделями.

**XML** (*e****X****tensible* ***M****arkup* ***L****anguage*) — расширяемый [язык разметки](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D1%80%D0%B0%D0%B7%D0%BC%D0%B5%D1%82%D0%BA%D0%B8). Рекомендован [Консорциумом Всемирной паутины](https://ru.wikipedia.org/wiki/W3C) (W3C). Спецификация XML описывает XML-документы и частично описывает поведение XML-процессоров

**Unified Modeling Language (UML)**— [язык](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F) [графического](https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D0%B7%D1%83%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F) описания для [объектного моделирования](https://ru.wikipedia.org/w/index.php?title=%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE%D0%B5_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5&action=edit&redlink=1) в области [разработки программного обеспечения](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F), [моделирования бизнес-процессов](https://ru.wikipedia.org/w/index.php?title=%D0%91%D0%B8%D0%B7%D0%BD%D0%B5%D1%81-%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5&action=edit&redlink=1), [системного проектирования](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) и отображения [организационных структур](https://ru.wikipedia.org/wiki/%D0%9E%D1%80%D0%B3%D0%B0%D0%BD%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0).

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

**Имя существительное - именник**

**Прилагательное -** прикметник

**Глагол -** дієслово

**Наречие** – присливник

**Местоимение** – займенник

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.

підходи семантичного аналізу тексту: ***cтатистичний*** *підхід*– базується на підрахунку кількості слів у текстах і не потребує глибоких теоретичних знань в лінгвістиці, використовуються усереднені дані і виявлені закономірності; ***кібернетичний*** – об’єднуює множину різноманітних математичних підходів, пов’язуючи комп’ютерну математику та штучний інтелект; ***лінгвістичний*** *підхід*до обробки природної мови складається з графоматичного розбиття тексту та виділення розділів, абзаців, речень, морфологічного – визначення характеристик та атрибутів окремого слова, синтаксичного – визначення характеристик підрядного зв'язку слова в словосполученнях, семантичного – пов’язаний із сурядним словосполученням тексту та його сенсу; ***cинтактико****-****семантичний*** *аналіз*–завданням блоку синтаксичного аналізу є представлення кожного речення заданого природномовного тексту у вигляді синтаксичного дерева (лексеми у реченні з синтактико-семантичними відношеннями між ними); ***морфологічний******аналіз*** – сутність процедури морфологічного аналізу полягає у приписуванні кожній мовній лексемі вхідного природномовного об’єкту відповідної змістової інформації та їх структуризацію у морфологічній таблиці; символічний підхід – базується на явному представленні знань, що здійснюється шляхом використання добре досліджених схем та алгоритмів (словники, формули, правила); ***коннективістський*** *підхід* – відповідає за обробку загальних моделей з використанням конкретних прикладів мовних явищ**; *метод допоміжних векторів*** – диференційний метод машинного навчання, що допомагає провести класифікацію слів за категоріями; ***прихована******марковська*** *модель* – це така графічна система, у якій кожна вершина представляє собою випадкову змінну, що може набувати будь-якого значення (з певними ймовірностями) між кількома станами, породжуючи при цьому один з декількох можливих вихідних символів з переходом для кожного з них. Множина всіх можливих станів та унікальних символів може бути дуже великою. Ми можемо бачити вихідні дані, проте початкові стани системи є прихованими**; *умовні випадкові*** *поля* – ця модель використовується для передбачення стану змінної та базується на спостереженнях за цією змінною; ***n-грамні моделі*** – використовуються в основному для прогнозу. В основі моделі лежіть теорія ймовірності. *N*-грамна модель обчислює вірогідність останнього слова *N*-грами, за умовою знання всіх попередніх.