ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования

Национальный исследовательский университет «Высшая школа экономики»

Факультет гуманитарных наук
Образовательная программа
«Фундаментальная и компьютерная лингвистика»

Феоктистова Эмма Александровна

ОТ НЕЙРОСЕТЕВОГО АНАЛИЗА К СЛОВАРНОМУ: МЕТОДЫ РАЗРАБОТКИ МОРФОЛОГИЧЕСКИХ АНАЛИЗАТОРОВ НА ОСНОВЕ ДАННЫХ, РАЗМЕЧЕННЫХ НЕЙРОСЕТЬЮ

Выпускная квалификационная работа студента 4 курса бакалавриата группы БКЛ191

Академический руководитель образовательной программы
канд. филологических наук, доц.
Ю.А. Ландер

_____

« » _____ 2023 г.

Научный руководитель

Профессор
О. Н. Ляшевская

_____

Научный консультант
Ученая степень, звание.
И.О. Фамилия

_____

Москва 2023

ОГЛАВЛЕНИЕ

1. Введение

Проблема морфологического анализа в области обработки естественного языка заключается в том, что естественные языки имеют сложную структуру, которая может варьироваться в зависимости от контекста. В морфологическом анализе необходимо разбирать слова на составные части, такие как корень, приставка, суффикс и окончание, для определения их грамматических характеристик, таких как часть речи, число, падеж, время и т.д. Это важно для многих задач обработки естественного языка, таких как автоматический перевод, распознавание речи, анализ тональности и других.

Однако традиционные методы морфологического анализа имеют ряд ограничений, такие как сложность в описании морфологических правил для каждого языка и его диалектов, необходимость ручной разметки текстов и сложность в масштабировании на большие наборы данных.

В свою очередь, нейросетевые методы морфологического анализа могут быть более точными и универсальными в сравнении с традиционными методами. Нейросетевые модели могут автоматически извлекать признаки из текста и определять грамматические характеристики слов, основываясь на больших наборах размеченных данных. Кроме того, такие модели могут быть масштабированы для обработки больших объемов текста.

Таким образом, разработка методов морфологического анализа на основе нейросетевых подходов является актуальной и важной задачей в области обработки естественного языка.

Традиционные методы морфологического анализа являются классическими и используются уже давно. Они основаны на правилах и шаблонах, которые определяют морфологические характеристики слов. Традиционные методы могут быть реализованы в виде морфологических словарей или морфологических анализаторов.

Морфологические словари содержат информацию о всех известных словах и их характеристиках. Однако, такие словари не могут распознавать неизвестные слова или нестандартные формы слов, которые не указаны в словаре.

Морфологические анализаторы, в свою очередь, используют правила и шаблоны для определения морфологических характеристик слов. Такие

анализаторы могут работать с неизвестными словами и даже с нестандартными формами слов. Однако, создание правил для каждого языка и диалекта является трудоемкой задачей, и требуется ручная разметка текстов.

Нейросетевые методы морфологического анализа используют машинное обучение и обработку естественного языка для определения характеристик слов. Нейросетевые модели могут использоваться для автоматического извлечения признаков из текста и определения грамматических характеристик слов. Нейросетевые методы могут обрабатывать большие объемы данных и работать с нестандартными формами слов, что делает их более универсальными в сравнении с традиционными методами.

Одним из примеров нейросетевых методов является использование рекуррентных нейронных сетей (RNN) и сверточных нейронных сетей (CNN) для морфологического анализа. Рекуррентные нейронные сети позволяют учитывать контекст слова при определении его морфологических характеристик, а сверточные нейронные сети позволяют быстро и эффективно извлекать признаки из текста.

Целью данного исследования является разработка модуля python архитектуры uniparser для русского языка на основе методов кластеризации/классификации парадигм.

Задачи диплома:

1. Изучение существующих методов и инструментов для морфологического анализа русского языка;
2. Изучение теории кластеризации и классификации данных в контексте морфологического анализа;
3. Сбор и предобработка данных для обучения и тестирования модуля;
4. Разработка модели на основе выбранных методов кластеризации/классификации парадигм;
5. Реализация и тестирование модуля на выбранном наборе данных;
6. Оценка производительности и точности модуля и сравнение с другими существующими инструментами для морфологического анализа русского языка;
7. Документирование результатов исследования.

2. Обзор существующей литературы

Методы разработки морфологических анализаторов на основе данных, размеченных нейронной сетью, являются одним из активно развивающихся направлений в области обработки естественного языка. В этой области существует множество статей, посвященных различным аспектам морфологических анализаторов на основе нейронных сетей.

Кузнецов Ю. и Лесин А. (2018) написали одну из первых работ в этой области. В данной работе предложена архитектура морфологического анализатора на основе нейронной сети, которая позволяет эффективно обрабатывать слова на русском языке.

Авторы обсудили несколько недостатков традиционных подходов, основанных на правилах и словарях, при обработке сложных морфологических явлений. Затем они описали архитектуру своего морфологического анализатора на основе нейронной сети, который состоит из двунаправленной сети с долгой кратковременной памятью (LSTM) для кодирования на уровне символов, сети с прямой связью для предсказания морфологических признаков и слоя условного случайного поля (CRF) для маркировки последовательности. Авторы также обсудили процесс обучения сети, который включает в себя использование набора данных аннотированных слов и их морфологических признаков.

Юревич Э. и Кузнецов Ю. (2019) представили продолжение своей предыдущей работы над морфологическим анализатором на основе нейронной сети, включив дополнительное обучение, позволяющее добавлять новые данные для повышения точности модели.

Авторы представили процесс обучения с добавлением новых аннотированных данных в существующий набор данных и обновлением весов модели для включения новой информации. Результаты оценки показывают, что метод поэтапного обучения повысил точность модели.

Белоногов А. и Константинова Н. (2021) представили новый подход к построению морфологического анализатора русского языка с использованием нейронных сетей. Авторы описали предложенную ими модель, в которой используется комбинация сверточных и рекуррентных нейронных сетей для

выполнения кодирования на уровне символов и прогнозирования морфологических признаков.

Одной из сильных сторон этой статьи является внимание к русскому языку, который является сложным флективным языком, который создает проблемы для традиционных морфологических анализаторов, основанных на правилах. Предлагаемый подход способен обрабатывать сложную морфологию русского языка, используя возможности нейронных сетей.

Еще одной сильной стороной статьи является подробное описание предлагаемой модели и экспериментов, проведенных для оценки ее производительности.

Кессикбаева Г., Чичекли И. (2014) представили основанный на правилах морфологический анализатор казахского языка. Авторы начинают с обсуждения важности морфологического анализа в обработке естественного языка, особенно в языках со сложной морфологией, таких как казахский. Они отмечают, что, хотя существуют морфологические анализаторы для казахского языка, они имеют такие ограничения, как неполный охват данных и отсутствие гибкости. Предлагаемый анализатор состоит из набора правил анализа различных морфологических признаков казахских слов.

Фариди, А.З., и Тайерс, Ф.М. (2009) разработали морфологический анализатор с открытым исходным кодом для бенгальского языка. Авторы начинают с обсуждения важности морфологического анализа в обработке естественного языка и проблем, связанных со сложной флективной и деривационной системой бенгальского языка.

Затем авторы описали процесс разработки анализатора, который включал в себя составление исчерпывающего лексикона бенгальских слов и их морфологических особенностей, а также создание набора правил для проведения морфологического анализа новых слов.

Дереза О. В., Каютенко Д. А., Феногенова А. С. (2016) представили сравнение десяти систем автоматического морфологического анализа: TreeTagger, Mystem, Pymorhy, Stanford POS tagger и других. Авторы сравнили производительность анализаторов с точки зрения точности и скорости обработки, а также их способность обрабатывать различные примеры.

Авторы представили четыре анализатора, выбранных для сравнительного исследования: TnT, TreeTagger, HunPos и Citar. Каждый из этих анализаторов описан с точки зрения лежащей в их основе методологии, сильных и слабых сторон.

Сегалович И. (2003) разработал модуль Mystem, который изначально был создан для поисковой системы Яндекс. Автор начал с обсуждения проблем морфологического анализа для поисковых систем, в частности проблемы неизвестных слов, которых нет в стандартных словарях. Затем он представил свой предложенный алгоритм, который использует комбинацию методов на основе словаря и правил для выполнения морфологического анализа и угадывания неизвестных слов.

Алгоритм сначала использует поиск в словаре для идентификации известных слов и их флективных форм. Затем он применяет набор правил для создания возможных форм неизвестных слов на основе их суффиксов и других лингвистических особенностей.

Коробов М. (2015) создал модуль pymorphy2, морфологический анализатор и генератор для русского и украинского языков. Автор описывает конструкцию системы, в основе которой лежит сочетание основанных на правилах и статистических методов.

Эта система состоит из нескольких компонентов, включая морфологический словарь, морфологический анализатор, лемматизатор и морфологический генератор. Морфологический словарь основан на большом корпусе русских и украинских текстов.

Сорокин А. и соавт. (2017) изучили результаты MorphoRuEval-2017, призванного стимулировать развитие технологий автоматической морфологической обработки русского языка. Они сравнивают методы, используемые участниками для решения задачи морфологического анализа.

В этом исследовании авторы описали дизайн трека оценки, который состоял из двух заданий: морфологической маркировки и лемматизации. Задача морфологической маркировки требовала от участников присвоения морфологических меток каждому слову в заданном предложении. Оценка

основывалась на корпусе текстов на русском языке, состоящем из новостных и художественных статей. В данной статье также рассматривается проблема унификации различных существующих обучающих сборников по русскому языку.

Панченко А., Константинова Н. и Лукашевич Н. (2021) представили новый подход к созданию морфологического анализатора, который может включать новую лингвистическую информацию в дополнение к стандартному вводу на уровне символов.

Авторы проводят эксперименты на российских и турецких наборах данных и сравнивают производительность своей модели с дополнительной лингвистической информацией и без нее. Результаты показывают, что включение данной информации повышает точность модели, особенно в тех случаях, когда морфологический анализ зависит от контекста, выходящего за рамки самого входного слова.

3. Обзор методов кластеризации / классификации парадигм

Для разработки модуля Python архитектуры uniparser для русского языка на основе методов кластеризации/классификации парадигм можно использовать следующие методы исследования:

1. Кластеризация слов по формам. Можно использовать алгоритмы кластеризации, такие как k-means, DBSCAN, Agglomerative Clustering, чтобы разбить слова на группы по их формам. Например, можно сгруппировать все слова с одинаковым окончанием.
2. Классификация слов по парадигмам. Можно использовать алгоритмы машинного обучения, такие как Decision Tree, Random Forest, Naive Bayes, чтобы классифицировать слова по их парадигмам. Например, можно обучить модель, которая будет определять, к какой парадигме относится слово по его морфологическим признакам.
3. Использование статистических методов. Можно использовать статистические методы для анализа частотности различных форм слов. Например, можно использовать методы частотного анализа, такие как TF-IDF, для определения наиболее важных форм слова.
4. Анализ контекста. Можно анализировать контекст, в котором используются слова, для определения их форм и парадигм. Например, можно использовать алгоритмы NLP, такие как Named Entity Recognition, для извлечения информации о морфологических признаках слова из текста.
5. Обучение с учителем и без учителя. Можно использовать как методы обучения с учителем, такие как классификация, так и методы без учителя, такие как кластеризация, для анализа и классификации парадигм слов.

В целом, для разработки модуля Python на основе методов кластеризации/классификации парадигм можно использовать комбинацию различных методов исследования, чтобы достичь наилучших результатов.

Рассмотрим подробнее несколько существующих методов. **K-means** - это один из наиболее популярных методов кластеризации. Он основан на разбиении множества объектов на заранее заданное количество кластеров, где каждый кластер представляет собой группу объектов, близких по своим характеристикам.

Данный алгоритм работает следующим образом:

1. Инициализация: задается количество кластеров K и случайным образом выбираются K начальных центров кластеров;
2. Присвоение объектов кластерам: каждый объект из множества данных относится к ближайшему центру кластера;
3. Пересчет центров кластеров: на основе присвоения объектов кластерам пересчитываются центры кластеров;
4. Повторение шагов 2 и 3 до тех пор, пока центры кластеров не перестанут изменяться или не будет достигнуто максимальное число итераций.

На каждой итерации алгоритма происходит пересчет расстояний между объектами и центрами кластеров, что может быть вычислительно затратно при большом количестве данных и/или большом числе кластеров.

K-means является эффективным методом кластеризации, но его результаты могут зависеть от начальных значений центров кластеров и числа кластеров.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) - это метод кластеризации, который основывается на плотности объектов в пространстве. Он позволяет автоматически определять количество кластеров и обнаруживать выбросы (шум).

Процесс происходит таким образом:

1. Задается радиус *Eps* и минимальное количество объектов *Min_samples* для определения плотных областей;
2. Выбирается случайный необработанный объект и проверяется, есть ли в его окрестности другие объекты, находящиеся на расстоянии не более *Eps*. Если количество таких объектов больше или равно *Min_samples*, то создается новый кластер и все объекты в его окрестности добавляются в него. Если количество объектов меньше *Min_samples*, то объект помечается как шум;
3. Для каждого объекта в кластере проверяется, есть ли в его окрестности другие объекты, которые еще не были обработаны. Если есть, то эти объекты добавляются в кластер;
4. Повторяются шаги 2 и 3 до тех пор, пока все объекты не будут обработаны.

DBSCAN позволяет определять кластеры произвольной формы и устойчив к шумовым данным. Однако, при большом количестве данных и/или большой размерности пространства расчеты также могут быть вычислительно затратными. Выбор оптимальных значений *Eps* и *Min_samples* может быть нетривиальным и требует экспериментов.

Agglomerative Clustering - это метод иерархической кластеризации, который начинается с того, что каждый объект считается отдельным кластером, а затем объединяет близлежащие кластеры, пока не будет достигнуто определенное количество или пока все объекты не будут объединены в один.

Расстояние между кластерами может быть измерено различными способами, например, евклидовым расстоянием между центрами кластеров или минимальным расстоянием между объектами в разных кластерах. При этом получается дерево кластеров (дендрограмма), которое можно использовать для визуализации результатов.

Agglomerative Clustering имеет ряд преимуществ, таких как возможность работы с любым типом данных, возможность определения оптимального количества кластеров с помощью дендрограммы, а также возможность использования различных метрик расстояния для определения близости между кластерами.

Однако метод также имеет некоторые недостатки, такие как высокая вычислительная сложность при большом количестве объектов и необходимость определения оптимального значения параметра расстояния.

Naive Bayes - это метод машинного обучения, который используется для классификации объектов на основе вероятностных моделей. Он основан на теореме Байеса, которая позволяет вычислять вероятность того, что объект относится к определенному классу на основе его характеристик.

В методе Naive Bayes каждый объект представляется в виде набора признаков, которые описывают его свойства. Затем для каждого класса вычисляются вероятности появления каждого из признаков и общая вероятность появления объектов этого класса. При классификации нового объекта вычисляются вероятности его принадлежности к каждому из классов на основе

его признаков, и объект относится к тому классу, для которого эта вероятность максимальна.

Метод Naive Bayes называется "наивным", потому что он предполагает, что все признаки объекта независимы друг от друга, что не всегда является правдой в реальных данных. Однако этот метод все еще дает хорошие результаты во многих задачах классификации, таких как определение спама в электронной почте или распознавание рукописных цифр.

Одним из главных преимуществ метода Naive Bayes является его скорость работы и низкие требования к вычислительным ресурсам. Кроме того, он хорошо работает с большими объемами данных и может использоваться для классификации объектов с любым типом признаков.

Однако метод Naive Bayes также имеет некоторые недостатки, такие как чувствительность к выбросам в данных и неспособность учитывать сложные взаимодействия между признаками. Кроме того, он не всегда дает лучшие результаты по сравнению с другими методами машинного обучения, такими как Decision Tree или Random Forest.

Random Forest - это метод машинного обучения, который используется для кластеризации объектов на основе ансамбля решающих деревьев. Он основан на идее комбинирования нескольких деревьев решений для улучшения точности и стабильности кластеризации.

В методе Random Forest каждый объект представляется в виде набора признаков, которые описывают его свойства. Затем создается несколько решающих деревьев, каждое из которых обучается на подмножестве объектов и признаков. Каждое дерево принимает решение о кластеризации объекта на основе его признаков, и объект относится к тому кластеру, который получил большинство голосов от всех деревьев.

Данный метод использует случайный выбор подмножества объектов и признаков для обучения каждого дерева. Это позволяет уменьшить вероятность переобучения и улучшить обобщающую способность модели.

Преимуществом метода Random Forest является его высокая точность и стабильность кластеризации. Он также хорошо работает с большими объемами данных и может учитывать объекты с любыми типами признаков.

Недостатки данного метода заключаются в высокой сложности модели и длительном времени обучения. Кроме того, он не способен учитывать сложные взаимодействия между признаками. В целом, метод Random Forest является мощным инструментом для кластеризации объектов, который может быть эффективно использован во многих задачах анализа данных.

4. Описание архитектуры uniparser и ее особенностей

Uniparser - это инструмент для автоматического разбора естественного языка на основе грамматических правил. Он использует набор грамматических правил, написанных на языке Python, для анализа входных данных и выдачи структурированной информации о частях речи, зависимостях между словами и других лингвистических аспектах. Эта библиотека является модульной и может быть использована на различных языках программирования, таких как Python, Java и C++.

Архитектура Uniparser имеет несколько особенностей:

1. Модульность: данный инструмент имеет модульную архитектуру, которая позволяет использовать различные модули для морфологического анализа и синтаксического разбора. Это означает, что пользователи могут выбрать те модули, которые наиболее подходят для конкретных задач;
2. Простота использования: Uniparser имеет простой и интуитивно понятный интерфейс, что делает его легким в использовании для пользователей с различными уровнями опыта в обработке естественного языка;
3. Лексический анализатор: он разбивает текст на отдельные слова и определяет их части речи и другие морфологические характеристики слова;
4. Синтаксический анализатор: он строит дерево зависимостей между словами и определяет их роли в предложении;
5. Семантический анализатор: он определяет значения слов и их связей на основе семантических правил.
6. Открытый исходный код: Uniparser распространяется под лицензией Apache 2.0, что позволяет свободно использовать и модифицировать его исходный код. Это также способствует развитию сообщества пользователей и разработчиков Uniparser.

Еще одной особенностью Uniparser является его способность работать с различными языками благодаря использованию универсальной модели грамматических правил. Это позволяет создавать грамматические правила для новых языков без необходимости переписывания всего инструмента.

Uniparser также предоставляет возможность настройки параметров анализа, таких как уровень детализации вывода и выбор используемых грамматических правил. Это делает его гибким инструментом для различных задач в области обработки естественного языка.

5. Рассмотрение примеров использования uniparser

Конкретные примеры использования Uniparser зависят от задачи и языка, на котором необходимо проводить морфологический анализ текста. Рассмотрим несколько примеров использования данной библиотеки:

- Автоматический анализ текста:
  Uniparser может быть использован для автоматического анализа текста на естественном языке. Например, он может быть использован для анализа новостных статей и выделения ключевых слов и фраз, определения смысла предложений и выявления связей между ними.
- Разработка приложений для обработки естественного языка:
  Uniparser может быть использован для разработки приложений для обработки естественного языка, таких как чат-боты или системы распознавания речи. Он может быть интегрирован в приложение, чтобы обеспечить автоматический анализ вводимого пользователем текста и создание соответствующего ответа.
- Исследовательские работы в области лингвистики:
  Uniparser может быть использован для исследовательских работ в области лингвистики, таких как анализ структуры языка или разработка новых грамматических правил для определенного языка. Он может быть использован для автоматического анализа текстов на различных языках и создания соответствующих грамматических правил.

Конкретно в области обработки естественного языка Uniparser может использоваться для таких задач, как:

- Морфологический анализ:
  Uniparser может разбирать слова на составляющие (лемму, часть речи, падеж, число и т.д.), что может быть полезно для анализа текста на естественном языке.
- Синтаксический анализ:
  Uniparser может использоваться для построения синтаксических деревьев, которые показывают связи между словами в предложении. Это может быть

полезно для автоматического извлечения информации из текста или для машинного перевода.

- Автоматический перевод:

  Uniparser может использоваться для автоматического перевода текста на другие языки, используя полученную информацию о морфологии и синтаксисе исходного текста.

- Классификация текста:

  Uniparser может использоваться для классификации текста на основе его содержания, используя информацию о морфологии и синтаксисе.

6. Особенности морфологии русского языка

Одной из особенностей морфологии русского языка является его богатство и сложность. Русский язык имеет более 15 падежей, что делает его одним из самых падежных языков в мире. Кроме того, русский язык имеет богатый глагольный аспект, который отражает не только время действия, но и его продолжительность, завершенность и другие характеристики.

Также русский язык имеет развитую систему склонения и спряжения, которая позволяет выражать различные оттенки значений слов. Например, слово "дом" может иметь различные формы в зависимости от падежа и числа: *дом, дома, дому, домов, домами, домах*.

Еще одной особенностью морфологии русского языка является наличие сложных словообразовательных процессов, таких как приставки, суффиксы и корневые изменения. Эти процессы позволяют образовывать новые слова и выражать сложные понятия.

Кроме того, русский язык имеет множество исключений и нестандартных форм, что усложняет его изучение и применение в речи. Например, некоторые слова имеют несколько различных форм в одном и том же падеже, а некоторые слова не склоняются вообще.

Также стоит отметить, что морфология русского языка подвержена изменениям и эволюции в соответствии с изменением языковых норм и общественных потребностей. В последнее время наблюдается упрощение склонения и спряжения в речи молодежи.

Морфология русского языка имеет многочисленные грамматические категории, такие как род, число, падеж, время, лицо, наклонение и другие, которые требуют от говорящего не только знания правил, но и умения правильно применять их в речи.

Однако, несмотря на все сложности, морфология русского языка является неотъемлемой частью его красоты и выразительности. Благодаря ей можно создавать разнообразные формы слов и выражать различные оттенки значения.

Таким образом, морфология русского языка - это одна из его главных особенностей, которая делает его уникальным и богатым. Несмотря на сложность,

она позволяет говорящим выразить свои мысли более точно и точнее передать свои эмоции.

Морфология русского языка отличается от морфологии других языков. Например, в английском языке меньше грамматических категорий, чем в русском, и многие формы слов не изменяются. В некоторых других языках, например, в китайском, нет грамматических падежей и окончаний слов, а смысл выражается контекстом и порядком слов в предложении.

Также морфология русского языка имеет свои особенности в сравнении с другими славянскими языками. Например, в польском языке есть больше падежей и различных форм глаголов, чем в русском. В украинском языке есть свои уникальные формы глаголов и прилагательных.

Тем не менее, морфология всех языков имеет свою сложность и требует от говорящего знания правил и умения правильно применять их в речи. Каждый язык имеет свои особенности, которые делают его уникальным и интересным для изучения.

7. Выделение особенностей для кластеризации / классификации парадигм

Можно выделить несколько характеристик слов, которые смогут помочь более точно кластеризовать или классифицировать используемый набор данных:

1. Падеж. В русском языке шесть падежей (именительный, родительный, дательный, винительный, творительный и предложный), которые используются для обозначения грамматической функции существительного или местоимения в предложении;
2. Число. Существует два числа (единственное и множественное число), которые используются для обозначения того, относится ли существительное или местоимение к одному или нескольким лицам, животным, предметам и т. д.;
3. Род. В русском языке есть три рода (мужской, женский и средний), которые используются для обозначения грамматического рода существительного или местоимения;
4. Спряжение глаголов. Русские глаголы спрягаются по времени, виду и наклонению. Время указывает на время действия (прошедшее, настоящее или будущее), вид указывает на завершенность или незавершенность действия, а наклонение указывает на отношение говорящего к действию;
5. Склонение прилагательных. Прилагательные согласуются в роде, числе и падеже с изменяемым существительным;
6. Склонение местоимений. Местоимения также согласуются в роде, числе и падеже с замещаемым ими существительным;
7. Части речи. В русском языке есть различные части речи, в том числе существительные, глаголы, прилагательные, наречия, местоимения, предлоги, союзы и междометия. Каждая часть речи имеет свои морфологические признаки, отличающие ее от других;
8. Изменения основы. Некоторые русские слова претерпевают изменения основы при изменении формы. Например, основа слова «дом» изменяется на «домов» в родительном падеже множественного числа;
9. Неправильные формы. Некоторые русские слова имеют неправильные формы, не соответствующие обычным правилам словоизменения.

Например, слово «человек» имеет неправильную форму родительного падежа множественного числа («людей» вместо «человеков»).

8. Выбор представительного набора данных для обучения и тестирования

Для разработки модуля python для русского языка необходимо выбрать представительный набор данных для обучения и тестирования.

Этот набор данных должен содержать достаточное количество примеров различных парадигм русского языка, чтобы обеспечить эффективное обучение модели. Также необходимо убедиться, что в наборе данных представлены различные типы слов (существительные, глаголы, прилагательные и т.д.), чтобы модель могла обрабатывать разнообразный текст.

Один из возможных способов выбора представительного набора данных - использование корпуса текстов на русском языке, содержащего большое количество различных слов и парадигм.

Для выбора конкретного набора данных можно использовать методы кластеризации и классификации, чтобы выделить наиболее представительные примеры различных парадигм и типов слов. Также можно использовать методы обработки естественного языка, такие как POS-теггинг и лемматизация, чтобы автоматически выделить различные формы слов и их грамматические характеристики.

Мы выделили несколько представительных наборов данных:

- **НКРЯ** (Национальный корпус русского языка): это крупнейший корпус текстов на русском языке, содержащий тексты из различных источников, включая прозу, поэзию, газеты и т.д. НКРЯ также содержит морфологически размеченные тексты, которые могут использоваться для обучения и тестирования модуля;
- **Taiga**: это открытый набор размеченных данных из разных источников. Содержит информации о морфологических характеристиках слов и синтаксических связях между ними;
- **Universal Dependencies**: это проект, который содержит размеченные корпусы текстов на различных языках, включая русский. Universal Dependencies содержит морфологическую и синтаксическую разметку для каждого слова в тексте, что может быть полезно для разработки модуля;

- **SynTagRus**: это корпус текстов на русском языке, содержащий синтаксическую разметку для каждого предложения. SynTagRus может использоваться для обучения модуля на основе синтаксических правил;
- **OpenCorpora**: это корпус текстов на русском языке с морфологической разметкой, который содержит более 3 миллионов словоформ.

При выборе набора данных следует учитывать его размер, качество разметки и доступность для использования в разработке модуля. Лучше всего будет использовать наборы данных, которые содержат разметку для различных морфологических категорий, таких как часть речи, падеж, число и т.д., чтобы обучить модуль на широком диапазоне языковых особенностей.

9. Описание наборов данных

Набор морфологически размеченных данных **НКРЯ** (Национальный корпус русского языка) включает в себя большой объем текстов на русском языке, которые были размечены по частям речи, падежам, числам, временам и другим морфологическим признакам.

Набор данных НКРЯ содержит тексты из различных источников, таких как художественная литература, научные статьи, газетные и журнальные статьи, разговорная речь и т.д.

Эти данные используются для проведения лингвистических исследований, анализа текстов, создания компьютерных программ для обработки естественного языка и других целей.

База данных **Taiga** - это корпус текстов на русском языке, который содержит более 700 миллионов слов и фраз, собранных из различных источников, включая художественную литературу, публицистику, научные статьи и другие жанры.

Каждое слово в тексте имеет свой морфологический разбор, который включает информацию о его части речи, падеже, числе, роде, времени и других характеристиках. Корпус включает в себя информацию о синтаксических связях между словами в предложении, а также о стилистических особенностях текстов.

Набор данных **Universal Dependencies** (UD) - это международный проект, который предоставляет открытые и доступные данные о различных языках мира, размеченные по универсальным морфологическим признакам.

Каждое слово в тексте имеет свой морфологический разбор, который включает информацию о его части речи, падеже, числе, роде, времени и других характеристиках.

Набор данных UD включает в себя тексты на более чем 100 языках мира, в том числе на таких экзотических языках, как хакасский, карачаево-балкарский, нивхский и другие.

Данные в наборе UD используют универсальные теги для обозначения морфологических признаков, что позволяет проводить сравнительный анализ между различными языками и создавать компьютерные программы для обработки естественного языка на разных языках.

Набор данных UD используется для проведения лингвистических исследований, анализа текстов, создания компьютерных программ для обработки естественного языка и других целей.

SynTagRus - это набор морфологически размеченных данных, используемых в исследованиях по синтаксису и морфологии русского языка. Этот набор данных разработан на основе Корпуса русского языка, который включает в себя различные тексты на русском языке, такие как проза, поэзия, научные тексты и другие.

Каждому слову в предложении приписаны морфологические тэги и грамматические атрибуты. Такая разметка позволяет исследователям изучать различные языковые явления, такие как грамматические конструкции, зависимости между словами, синтаксические структуры предложений и многое другое.

Набор данных SynTagRus содержит более 500 000 словоформ, которые разделены на предложения и размечены с использованием морфологических тэгов. Каждое слово в предложении имеет свой уникальный идентификатор, морфологическую информацию (такую как падеж, род, число, время и т.д.) и связи с другими словами в предложении.

Этот набор данных позволяет проводить различные эксперименты, тренировать модели и создавать приложения, основанные на анализе и синтаксической обработке текстов на русском языке.

Набор данных **OpenCorpora** - это корпус текстов на русском языке, размеченных по морфологическим признакам.

Каждое слово в тексте имеет свой морфологический разбор. Кроме того, данные в наборе OpenCorpora включают информацию о грамматических формах слов, например, о склонении глаголов и прилагательных, и включают уникальные теги для обозначения морфологических признаков, что позволяет проводить более точный анализ текстов на русском языке.

Набор данных OpenCorpora включает в себя большое количество текстов различных жанров, таких как художественная литература, научные статьи, газетные материалы и другие.

10. Предобработка данных для подготовки их к вводу в нейросеть

Перед обучением модели необходимо провести предобработку данных, чтобы подготовить их к вводу в нейросеть. Важными этапами предобработки данных являются:

1. Токенизация - разбиение текста на отдельные слова или токены. Это необходимо для того, чтобы нейросеть могла обрабатывать каждое слово отдельно;
2. Лемматизация - приведение слов к их базовым формам (леммам). Это позволяет сократить количество уникальных слов в наборе данных и уменьшить размерность входных данных для нейросети;
3. POS-теггинг - определение грамматических характеристик каждого слова (часть речи, падеж, число и т.д.). Эта информация может быть полезна для определения парадигм и правил склонения/спряжения;
4. Удаление стоп-слов - удаление часто встречающихся слов, которые не несут смысловой нагрузки (например, предлогов, союзов и т.д.). Это позволяет уменьшить размерность входных данных и ускорить обучение модели;
5. Векторизация - преобразование слов в числовые векторы. Для этого можно использовать различные методы, такие как Bag-of-Words, TF-IDF или Word2Vec. Это позволяет представить слова в виде числовых значений, которые могут быть обработаны нейросетью.

После проведения предобработки данных и получения числовых векторов для каждого слова, можно приступать к обучению модели на выбранном наборе данных. В нашем исследовании мы используем уже размеченные корпуса текстов, содержащие всю необходимую информацию о каждом из токенов. Комбинация различных датасетов, использование текстов с различными тематиками, размер обучающих данных - все это может повлиять на результаты работы нейронной сети.

11. Выбор подходящей модели

Открытые датасеты GSD, SynTagRus, Taiga представлены в формате conllu (Conference on Computational Natural Language Learning), содержащий в себе всю необходимую информацию о предложении и словах, морфологии, синтаксисе и семантике.

В представленных данных используется универсальный набор тегов для обозначения различных частей речи:

- ADJ – имя прилагательное
- ADP – предлог
- ADV – наречие
- AUX – вспомогательный глагол
- CCONJ – соединительный союз
- DET – определяющее слово
- INTJ – междометие
- NOUN – имя существительное
- NUM – числительное
- PART – частица
- PRON – местоимение
- PROPN – имя собственное
- PUNCT – пунктуация
- SCONJ – подчинительный союз
- SYM – символ
- VERB – глагол
- X – иностранные слова и т. д.

Данные, выбранные нами для задачи морфологического анализа текста, представлены в разных объемах, но в схожих форматах. Каждый датасет включает в себя следующую информацию: id предложения, предложение, id слова, словоформу, лемму, часть речи, id главного слова (от которого зависит текущий токен), отношение между главным и зависимым в формате UD Relations, список вторичных зависимостей и другие аннотации.

У каждой части речи также есть свои особенности, перечисленные для каждого слова: одушевленность / неодушевленность, совершенный / несовершенный вид, падеж (именительный, родительный, дательный, винительный, творительный, партитивный, местный, звательный), степень (положительная, сравнительная), иностранное / не иностранное слово, род (мужской, женский, средний), единственное / множественное число, лицо (первое, второе, третье), отрицательная полярность, время (прошедшее, настоящее, будущее), краткость, форма глагола (начальная, инфинитив, причастие, деепричастие), залог (активный, пассивный).

Таблица 1. Число токенов в используемых датасетах

Датасет	Обучающие данные (кол-во токенов)	Валидационные данные (кол-во токенов)	Тестовые данные (кол-во токенов)
GSD	74900	11710	11385
SynTagRus	1206300	153590	157990
Taiga	176630	10095	10275

В нашем эксперименте будут использованы основные характеристики слова, а именно начальная форма, часть речи и ее особенности. Для каждой словоформы будет представлен свой набор признаков.

Например, для слова “наследство” будет список “наследство,NOUN,Inan,Acc,Neut,Sing”, указывающий на то, что слово является неодушевленным существительным среднего рода, стоящим в винительном падеже в единственном числе. В случаях с пунктуацией характеристика будет описана в виде “PUNCT,None”.

Рисунок 1. Пример полученной базы данных размеченных слов

sent_id	text	id	form	lemma	upos	xpos	feats	head	deprel	deps	misc
train-s1	Во время битвы между силами Магнето	1	Во	во	ADP	IN		2	case		
train-s1	Во время битвы между силами Магнето	2	время	время	NOUN	NN	{'Animacy': 'Inan', 'Case': 'Acc', 'Gender': 'Neut', 'Number': 'Sing'}	18	obl		
train-s1	Во время битвы между силами Магнето	3	битвы	битва	NOUN	NN	{'Animacy': 'Inan', 'Case': 'Gen', 'Gender': 'Fem', 'Number': 'Sing'}	2	nmod		
train-s1	Во время битвы между силами Магнето	4	между	между	ADP	IN		5	case		
train-s1	Во время битвы между силами Магнето	5	силами	сила	NOUN	NN	{'Animacy': 'Inan', 'Case': 'Ins', 'Gender': 'Fem', 'Number': 'Plur'}	3	nmod		
train-s1	Во время битвы между силами Магнето	6	Магнето	Магнето	PROPN	NNP	{'Animacy': 'Anim', 'Case': 'Gen', 'Gender': 'Masc', 'Number': 'Sing'}	5	nmod		
train-s1	Во время битвы между силами Магнето	7	и	и	CCONJ	CC		8	cc		
train-s1	Во время битвы между силами Магнето	8	героями	герой	NOUN	NN	{'Animacy': 'Anim', 'Case': 'Ins', 'Gender': 'Masc', 'Number': 'Plur'}	5	conj		{'SpaceAfter': 'No'}
train-s1	Во время битвы между силами Магнето	9	,	,	PUNCT	,		11	punct		
train-s1	Во время битвы между силами Магнето	10	кто	кто	PRON	WP	{'Animacy': 'Anim', 'Case': 'Nom', 'Gender': 'Masc', 'Number': 'Sing'}	11	nsubj		
train-s1	Во время битвы между силами Магнето	11	восстановил	восстановить	VERB	VBC	{'Aspect': 'Perf', 'Gender': 'Masc', 'Mood': 'Ind', 'Number': 'Sing', 'Ten	8	acl:relcl		
train-s1	Во время битвы между силами Магнето	12	свои	свой	DET	PRP$	{'Animacy': 'Inan', 'Case': 'Acc', 'Number': 'Plur'}	13	det		
train-s1	Во время битвы между силами Магнето	13	воспоминания	воспоминание	NOUN	NN	{'Animacy': 'Inan', 'Case': 'Acc', 'Gender': 'Neut', 'Number': 'Plur'}	11	obj		{'SpaceAfter': 'No'}

Рисунок 2. Пример подготовленной к обучению базы данных размеченных слов

form	data
Начальный	начальный,ADJ,Nom,Pos,Masc,Sing
ролик	ролик,NOUN,Inan,Nom,Masc,Sing
,	,,PUNCT,None
или	или,CCONJ,None
опенинг	опенинг,NOUN,Inan,Nom,Masc,Sing
(	(,PUNCT,None
от	от,ADP,None
,	,,PUNCT,None
сокр.	сокращенно,ADV,Pos

После приведения данных в единый формат, мы приступаем к задаче кластеризации слов, которая может помочь в задачах классификации и категоризации текстов. Например, если у нас есть большой корпус новостных статей, мы можем использовать кластеризацию, чтобы выделить группы статей по тематике. Это позволит нам создать более точные модели классификации, которые будут учитывать не только отдельные слова, но и контекст, в котором они используются.

Аналогично кластеризация может быть полезна в задачах анализа тональности текстов. Если мы имеем дело с большим количеством отзывов на продукт или услугу, мы можем использовать кластеризацию, чтобы выделить группы отзывов с разной тональностью (положительные, отрицательные, нейтральные).

В нашем исследовании данная процедура поможет в обучении нейронной сети, так как она позволит сократить количество уникальных слов в обучающем корпусе и преобразовать их в числовые значения. Это позволит уменьшить размерность входных данных для нейронной сети и ускорить ее обучение. Кроме того, кластеризация может помочь выделить смысловые группы слов, что может улучшить качество предсказаний нейронной сети.

Мы применили метод K-means на наших данных. Одной из главных сложностей является выбор количества необходимых кластеров. В рамках данного эксперимента мы установили количество, равное одной десятой от количества токенов в датасете (например, для обучающего датасета GSD данное число равно 7490). Полученные кластеры довольно разнообразны по своей длине и наполнению. Количество токенов в одном кластере может достигать 19000

единиц, в то время как другой кластер содержит всего 2 единицы. При этом средняя длина кластера составляет 11 единиц.

Рисунок 3. Пример образованных кластеров с помощью метода K-means

Cluster 0	Cluster 1	Cluster 2	Cluster 3
пустое	права	кровь	Сериал
пусть	права	крови	сериалах
Капустин	права	акров	сериале
Пустошь	правах	крови	сериала
Пустоте	права	крови	Сериал
устья	Справа	сокровищах	минисериала
устья	правах	крови	сериала
устья	правах		сериале
	права		

В дальнейшем обучении данным будет присваиваться номер кластера, что должно как ускорить обучение модели нейронной сети, так и улучшить предсказание для новых входных данных.

12. Описание архитектуры разработанного модуля

Архитектура нейронной сети:

- Входной слой: Принимает последовательность слов или символов, которые требуют морфологического анализа. Эти данные могут быть представлены в виде векторных представлений слов или символов.
- Скрытые слои: Содержат набор скрытых узлов, которые выполняют вычисления и обрабатывают информацию из входного слоя.
- Выходной слой: Содержит набор узлов, которые предсказывают морфологические тэги для каждого слова во входной последовательности.

Список литературы

Belonogov, A., & Konstantinova, N. (2021). Neural network-based morphological analyzer for Russian language. In: Proceedings of the 2021 International Conference on Computational Linguistics and Natural Language Processing (pp. 42-48). ACM.

Dereza O. V., Kayutenko D. A., Fenogenova A. S. (2016). Automatic morphological analysis for Russian: a comparative study. In: Proceedings of the International Conference Dialogue 2016. Computational linguistics and intellectual technologies.

Faridee, A.Z., & Tyers, F.M. (2009). Development of a morphological analyser for Bengali. FREEOPMT.

Kessikbayeva G., Cicekli I. (June 2014). Rule based morphological analyzer of Kazakh language. In: Proceedings of the 2014 Joint Meeting of SIGMORPHON and SIGFSM, Baltimore, Mary-land, Association for Computational Linguistics

Korobov, M. (2015). Morphological Analyzer and Generator for Russian and Ukrainian Languages. In: Khachay, M., Konstantinova, N., Panchenko, A., Ignatov, D., Labunets, V. (eds) Analysis of Images, Social Networks and Texts. AIST 2015. Communications in Computer and Information Science, vol 542.

Kuznetsov, Y., & Lesin, A. (2018). Neural network-based morphological analyzer. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (pp. 5084-5094). Association for Computational Linguistics.

Panchenko, A., Konstantinova, N., & Loukachevitch, N. (2021). Neural network-based morphological analyzer with the possibility of embedding additional linguistic information. In: Proceedings of the 2021 International Conference on Information Technology and Systems (pp. 1-8). IEEE.

Segalovich I. (2003). A Fast Morphological Algorithm with Unknown Word Guessing Induced by a Dictionary for a Web Search Engine. MLMTA, pp. 273-280.

Sorokin, A., Shavrina, T., Lyashevskaya, O., Bocharov, B., Alexeeva, S., Droganova, K., ... & Granovsky, D. (2017). MorphoRuEval-2017: an evaluation track for the automatic morphological analysis methods for Russian. In: Proceedings of the International Conference “Dialogue 2017”.

Yurevich, E., & Kuznetsov, Y. (2019). Neural network-based morphological analyzer with incremental training. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (pp. 4274-4279). Association for Computational Linguistics.

Приложение

Ссылка на github с материалами работы:

https://github.com/Mefeoss/Diploma_MA_with_neural_networks/tree/main