**Введение**

Вопрос обработки информации с каждым годом становится всё более и более актуальным, потому что вследствие цифровизации и человеческого развития количество информации в Интернете беспрерывно растёт.

Совсем недавно анализу подвергались лишь заранее собранные данные, и на их основе люди делали какие-то выводы и гипотезы. Серьёзными ограничением на первых порах были скромные вычислительные мощности (по сравнению с тем, что мы имеем сейчас) и высокий порог входа в индустрию - нужно было не только обладать знаниями в области языка, но и иметь навыки владения используемыми технологиями, чтобы писать эффективные программы.

Сегодня же мы анализируем данные в реальном времени. Практически не задумываемся об оптимизации, потому что имеем в своём распоряжении эффективные высокоуровневые инструменты. И самое главное - стремимся достичь наиболее эффективных результатов в анализе.

В отношении русского языка эта тема особенно актуальна, потому что все ведущие разработки в этой области ведутся на английском языке. Технологии получают поддержку других языков в случае успешности или наличием энтузиастов, которые эту поддержку и реализуют.

Так же существуют локальные разработки в рамках того или иного языка, но они редко попадают в свет, потому что разрабатываются и используются внутри корпораций, крупных компаний или банков.

В работе используются открытые технологии, которые имеют поддержку русского языка. С их помощью будет проведен частотный анализ и анализ на основе веса слов.

*Цель работы: провести семантический анализ текста новостных сообщений и сделать вывод относительно используемых методов.*

*Задачи:*

* *Провести обзор выбранных методов*
* *Собрать и подготовить данные для анализа*
* *Визуализировать результаты*
* *Сравнить полученные данные и выявить их недостатки/преимущества.*

1. **Сематнтический анализ**

Семантический анализ является подзадачей (или же частным случаем) NLP, которому прежде всего нужно дать определение.

NLP (Natural Language Processing - “обработка естественного языка”) — это направление в машинном обучении, посвященное распознаванию, генерации и обработке устной и письменной человеческой речи. Находится на стыке дисциплин искусственного интеллекта и лингвистики.

NLP включает в себя ряд основных подходов:

* Предобработка текста (переводит текст в удобный для машины вид)
* Стемминг (приводит слово к его основной форме)
* Лемматизация (альтернатива стемминга, которая приводит слово к словарной форме - Именительный падеж, единственное число, мужской род, инфинитив)
* Векторизация (отображение текста в векторном пространстве)
* Дедубликация (удаление дубликатов из текста, который представлен в векторном пространстве)
* Семантический анализ (смысловой анализ текста)
* Распознование именованных сущеностей и извлечение отношений (именованные сущности - объекты из текста, которые могут быть отнесены к одной из заранее заявленных категорий, например - организации, личности, адреса).
* И др.

Подходы пересекаются между собой и в рамках задачи их всегда используется несколько.

**Семантический анализ** — это последовательность действий алгоритма автоматического понимания текстов, которая выделяет смысловые отношения в тексте.

В общем случае семантическое представление является графом, семантической сетью, отражающим бинарные отношения между двумя узлами — смысловыми единицами текста. Глубина семантического анализа может быть разной, а в реальных системах чаще всего строится только лишь синтаксико-семантическое представление текста или отдельных предложений.

Перед анализом документ чаще всего проходит предварительную очистку, в ходе которой удаляются все служебные символы, цифры представляются в виде текста, а слова приводятся к своей начально форме. Так же возможно удаление стоп-слов — это слова, которые не вносят никакой дополнительной информации в текст и только добавляют “шум” в данные.

В ходе анализа текст проходит через несколько этапов обработки: токенизация для идентификации словоформ, морфологический, синтаксический анализ.

*Токенизация — это процесс разбиения текстового документа на отдельные слова или предложения. Разбиение происходит по пунктуационному разделителю: в случае предложения точке, слова - пробелу.*

Последним этапом идет вторичный семантический анализ (первичный анализ в основном происходит параллельно морфологическому), в ходе которого устанавливаются взаимосвязи между сущностями, происходит извлечение мнений и анализ тональности текста. Основной целью анализа тональности является не только определение настроений, но также уровень объективности высказывания.

Семантический анализ активно применяется для создания онлайн-консультантов или чат-ботов, поисковых и рекомендательных системах и анализе тональности текста.

В данной работе будут рассмотрены частотный анализ и анализ на основе веса слов.

Частотный анализ представляет собой подсчёте количества вхождений слова в текст и формировании выводов на основе самых часто встречающихся слов. Чаще всего этот метод применяется в криптоанализе и работает с буквами.

Вес слова — это его значимость в рамках данного текста. Для его вычисления подсчитывается число вхождений каждого слова не только в конкретном документе, но и в похожих текстах. Такой подход называется «Частота термина — обратная частота документа» (term frequency — inverse document frequency, TF-IDF).

Так, повторяющиеся в тексте слова «вознаграждаются», но также «штрафуются», если часто употребляются в коллекции схожих текстов, которую включают в алгоритм. Так системе NLP проще найти уникальные слова и оценить их важность в тексте.

Похожие текста, которые используются при сравнении, представляют собой “дата-сет”, на котором модель была обучена.

**2. Сбор и анализ данных**

**1.1. Инструментарий**

**Когда речь заходит об анализе данных, машинном обучении, Big-Data, нейронных сетях и в целом науке, то сразу приходит на ум Python, который является лидером в этой области. Простой синтаксис, большое комьюнити, высокая скорость прототипирования и большое количество библиотек и пакетов — всё это выводит его на первый план.**

**Стоит отметить, что сам по себе язык интерпретируемый, и как следствие, серьёзно проигрывает по скорости компилируемым языкам вроде C++ или Rust. А в рамках большого количества данных скорость и оптимизация стоят чуть ли не на первом месте.**

**Но почему же тогда большинство людей использует именно его? А потом что большинство популярных библиотек, особенно тех, которые требуют скорости, написаны на компилируемых языках, Python же выступает в роли “обёртки”, которая позволяет использовать быстрый низкоуровневый код, написав буквально пару строчек.**

**Для сбора данных будут использоваться библиотеки: selenium и bs4.**

**Для анализа данных: nltk (c помощью него будет выполняться токенизация и частотный анализ) и gensim (с пощью инструментов этой библиотеки будет выполняться частотный анализ).**

**Все модели, которые будут использованы, изначально обучены на “лучшем” дата-сете. Лучшим является тот дата-сет, который впоследствии даёт наилучшие результаты.**

**Так как большинство моделей являются регресионными, вопрос качества данных для обучения стоит чуть ли не на первом месте. В свободном доступе есть огромное количество датасетов, в том числе от компании Google.**

**1.2 Парсинг данных**

В качестве платформы, с которой мы будем брать новостные сообщения был выбран Habr. Он интересен нам не только из-за своей популярности, но и из-за разнообразия тем публикаций, которые пишутся на самые разные тематики.

Так как веб-страница формируется при помощи JavaScript, “спарсить” её не так просто - вместо содержания статьи мы получаем код, который нужно “прогрузить”. Для этого мы воспользуемся библиотекой selenium, которая эмулирует работу браузера, тем самым прогружает весь фронтенд приложения.

Затем при помощи библиотеки bs4 мы “достаём” из страницы блок с текстом, который и необходим анализа.

Так как мы имеем дело с полноценной html-страницей, то прежде всего нам нужно очистить её от тегов и лишних пробелов. Для этого воспользуемся силой регулярных выражений.

res = re.sub(r'<[^>]\*\>', '', article).strip()

res = re.sub(r' +', ' ', res)

Полученный “чистый текст” записывается в файл и дальше мы переходим к подготовке текста к анализу.

2.2. Частотный анализ

Подготовка текста к частотному анализу и анализу на основе веса слов немного отличается, поэтому мы рассмотрим их отдельно в рамках каждого метода.

Для частотного анализа текст проходит следующие этапы обработки:

1. Приведение слов к нижнему регистру
2. Удаление пунктуационных символов (в том числе перенос строки \n и таб \t) и пробелов
3. Удаление английских букв.

Чаще всего английские буквы и термины формируют только “шум”, который в дальнейшем даёт некорректные результаты.

1. Токенизация

Выполняется она при помощи функции из библиотеки nltk и выполняет следующие преобразования:

(тут фотография из Jupiter)

1. Удаление стоп-слов

Стоп-слова, как и английские символы, создают сильный “шум”, особенно при частотном анализе. Любое злоупотребление союзами или вводными словами автоматически приводит к не самым лучшим результатам.

Стоп слова достаются из библиотки nltk и отдельной бибиотеки stop\_words.

Берутся слова из двух библиотек, потому что наборы между собой отличаются. Наша же цель - убрать как можно больше стоп-слов, поэтому мы стремимся взять в наш словарь их как можно больше.