Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ **ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ**

КАФЕДРА **КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.01 Информатика и вычислительная техника**

О Т Ч Е Т

по лабораторной работе № 1

Название: Построение семантической сети документов с помощью программы Complearn.

Дисциплина: Методы и алгоритмы сжатия данных..

Студент	ИУ6-62Б		С.В. Астахов, М.Э Хабаров, Д.И. Насибуллин, А.Е. Медведев
	(Группа)	(Подпись, дата)	(И.О. Фамилия)
Преподаватель			
		(Подпись, дата)	(И.О. Фамилия)

Москва, 2022

Цель работы: знакомство с основными принципами построения семантической сети. Освоение навыков анализа семантической сети документов.

Ход работы.

Текст 1.

6 упущений в курсе науки о данных

Наука о данных — обширная тема. Для ее понимания требуется много времени, не говоря уже о полном овладении. Неудивительно, что многие учебные заведения, в отличие от традиционных курсов, разрабатывают специализированные программы по науке о данных. Эти программы, как правило, находятся на стыке информатики, математики и статистики. Студентов учат понимать и решать задачи с использованием данных и статистики. Однако программированию и практическому использованию решений уделяется недостаточно внимания.

Другими словами, углубленное изучение математики и статистики происходит в ущерб таким дисциплинам, как создание программного обеспечения, передовые методы программирования, разработка пользовательского интерфейса и базы данных. Кроме того, повышенное внимание к нейронным сетям отвлекает новичков от более простых и порой более эффективных технологий, таких как деревья решений и бустеры.

В этой статье освещаются аспекты, которые часто упускаются из виду в стандартных образовательных программах и которые, при должном внимании к ним, позволяют приобрести важнейшие компетенции в науке о данных. Конечно, не обязательно становится исключительным знатоком во всех областях Data Science. Тем не менее приобретение определенных навыков существенно повысит ваш профессиональный уровень в целом.
#1. Архитектура программного обеспечения

Инженеры ненавидят ноутбуки Jupyter по веской причине: они являются полной противоположностью “модульному подходу”.

Хороший дизайн программного обеспечения базируется на трех основополагающих принципах: высокая согласованность, низкая связанность и низкая избыточность. Другими словами, каждый из модулей специализируется на одной проблеме, они независимы друг от друга и в них практически нет дублирования кода. Так, код, загружающий набор данных, не должен выполнять ничего другого (например, очистку данных) или зависеть от какого-либо другого модуля (например, модуля расширения данных). Он должен быть единственным местом в кодовой базе, предназначенным для загрузки данных.

В большинстве учебников по науке о данных все помещается в один ноутбук, что с инженерной точки зрения совершенно недопустимо. Файл “все в одном” означает, что загрузка, очистка и подготовка набора данных происходит вместе с кодом, который его обслуживает и использует. Результирующий файл имеет множество пересекающихся задач, и, скорее всего, несколько ячеек, пришедших из других ноутбуков.

Текст 2.

Нюансы распознавания речи. Восстанавливаем пунктуацию, числа и заглавные буквы

градиент обреченный

В задачах распознаваниях речи при переводе аудио в текст есть дополнительные этапы, делающие этот текст более человекочитаемым. Например, предложение "привет хабр сегодня мы сделаем двадцать шесть моделей по распознаванию голоса" будет выглядеть лучше в таком виде: "Привет, хабр. Сегодня мы сделаем 26 моделей по распознаванию голоса". Другими словами, сегодня мы поговорим про то, как автоматически восстановить пунктуацию и капитализацию (сделать нужные буквы заглавными). Также упомянем

денормализацию текста (при этом числа обретут свою цифровую форму обратно, эту задачу еще называют inverse text normalization).

Пунктуация и капитализация

После непродолжительного поиска выяснится, что пара решений для русского языка в этом направлении уже есть (например, модели от vosk и silero). Мы же копнем чуть глубже и разберемся как самому натренировать такую модель. Это даст нам возможность выбора знаков препинания, нужного языка (например, башкирского или чувашского) и подбора соответствующего нашему домену корпуса текстов.

Важно понимать, что так как мы работаем исключительно с текстами, то предложения типа "казнить нельзя помиловать" модель будет разрешать на основании данных, на которых она обучалась. Эти данные никак не связаны с аудио (я встречал идеи о том, как можно использовать паузы из аудио в виде признаков, но это не очень надежно, особенно в разговорной речи) и неточности обязательно будут. Нашей задачей здесь является лишь сделать текст более легким для восприятия.

Данные

Первым делом нам нужно раздобыть тренировочный корпус. Это должен быть текстовый документ с одним предложением на строку. В предложениях должны быть знаки препинания, которые мы хотим восстанавливать. Подготовить такой файл, имея на руках обычные тексты, довольно просто.

Текст 3.

Прогнозирование временных рядов можно представить как контролируемую проблему обучения.

Это переформирование ваших данных временных рядов позволяет вам получить доступ к набору стандартных линейных и нелинейных алгоритмов машинного обучения для вашей задачи.

В этом посте вы узнаете, как можно переформулировать проблему временных рядов как проблему контролируемого обучения для машинного обучения. Прочитав этот пост, вы узнаете:

Что такое контролируемое обучение и как оно является основой для всех алгоритмов интеллектуального машинного обучения с прогностическим моделированием.
Метод скользящего окна для формирования набора данных временного ряда и как его использовать.
Как использовать скользящее окно для многомерных данных и многошагового прогнозирования.

Давайте начнем.

Контролируемое машинное обучение

Большая часть практического машинного обучения использует контролируемое обучение.

Контролируемое обучение - это когда у вас есть входные переменные (Икс) и выходная переменная (Y) и вы используете алгоритм, чтобы узнать функцию отображения от входа к выходу.

$Y = f(X)$

Цель состоит в том, чтобы аппроксимировать реальное базовое отображение настолько хорошо, чтобы при появлении новых входных данных (Икс), вы можете предсказать выходные переменные (Y) для этих данных.

Фрагмент 3 текста для построения дерева (переведен на английский, так как программа некорректно воспринимает кириллицу).

This is called supervised learning, because the process of learning an algorithm from a set of training data can be considered as a teacher controlling the learning process.

We know the correct answers; the algorithm iteratively makes predictions on the training data and is corrected by making updates. Training stops when the algorithm reaches an acceptable level of performance.

Supervised learning problems can be further grouped into regression and classification problems.

Classification: The classification problem is that the output variable is a category, for example: red" and "blue" or "disease" and "no disease".

regression: A regression problem where the output variable is a real value, such as "dollars" or "weight. The above example is a regression problem.

Выполнение.

Часть 1. Сравнение текстов 1-3

Попарно сравниваем тексты командой ncd. Сравнение текстов и результаты приведены на рисунке 1.

```
C:\Users\Professional\Desktop\complearn-qsearch-win-1.0.8-bin>ncd texts/1.txt  texts/2.txt
 0,906207

C:\Users\Professional\Desktop\complearn-qsearch-win-1.0.8-bin>ncd texts/1.txt  texts/3.txt
 0,915737

C:\Users\Professional\Desktop\complearn-qsearch-win-1.0.8-bin>ncd texts/2.txt  texts/3.txt
 0,905325
```

Рисунок 1 – результаты сравнения текстов

Часть 4. Построение дерева

В результате работы команд ncd и maketree была получена матрица, фрагмент которой приведен на рисунке 2 и дерево (рисунки 3-4).

```
This 0,000000 0,333333 0,428571 0,555556 0,529412 0,466667 0,333333 0,466667 0,333333
0,500000 0,333333 0,529412 0,333333 0,333333 0,333333 0,333333 0,500000 0,333333 0,333333
0,333333 0,555556 0,333333 0,333333 0,466667 0,578947 0,333333 0,500000 0,500000 0,333333
0,333333 0,333333 0,466667 0,500000 0,333333 0,529412 0,578947 0,384615 0,578947 0,333333
0,333333 0,500000 0,333333 0,333333 0,333333 0,529412 0,333333 0,428571 0,500000 0,500000
0,384615 0,333333 0,333333 0,529412 0,466667 0,333333 0,555556 0,384615 0,333333 0,600000
0,555556 0,500000 0,500000 0,333333 0,333333 0,466667 0,466667 0,333333 0,555556 0,333333
0,636364 0,529412 0,652174 0,333333 0,636364 0,466667 0,333333 0,333333 0,333333 0,428571
0,500000 0,333333 0,333333 0,529412 0,333333 0,500000 0,333333 0,333333 0,428571 0,333333
0,529412 0,333333 0,333333 0,529412 0,578947 0,333333 0,555556 0,466667 0,384615 0,333333
0,428571 0,500000 0,333333 0,333333 0,333333 0,428571 0,333333 0,333333 0,529412 0,333333
0,500000 0,333333 0,384615 0,466667 0,333333 0,333333 0,555556 0,500000 0,466667 0,428571
0,333333 0,333333 0,428571 0,333333
is 0,333333 0,000000 0,428571 0,555556 0,529412 0,466667 0,272727 0,466667 0,200000
0,500000 0,200000 0,529412 0,333333 0,200000 0,272727 0,200000 0,500000 0,333333 0,272727
0,200000 0,555556 0,200000 0,200000 0,466667 0,578947 0,272727 0,500000 0,500000 0,200000
0,333333 0,272727 0,466667 0,500000 0,272727 0,529412 0,578947 0,384615 0,578947 0,200000
0,272727 0,500000 0,333333 0,272727 0,000000 0,529412 0,200000 0,428571 0,500000 0,500000
0,384615 0,333333 0,272727 0,529412 0,466667 0,200000 0,555556 0,384615 0,200000 0,600000
0,555556 0,500000 0,500000 0,272727 0,200000 0,466667 0,466667 0,333333 0,555556 0,272727
0,636364 0,529412 0,652174 0,272727 0,636364 0,466667 0,000000 0,333333 0,272727 0,428571
0,500000 0,000000 0,200000 0,529412 0,272727 0,500000 0,333333 0,272727 0,428571 0,200000
0,529412 0,272727 0,272727 0,529412 0,578947 0,200000 0,555556 0,466667 0,384615 0,272727
0,428571 0,500000 0,000000 0,200000 0,333333 0,428571 0,333333 0,200000 0,529412 0,200000
0,500000 0,272727 0,384615 0,466667 0,000000 0,200000 0,555556 0,500000 0,466667 0,428571
0,272727 0,333333 0,428571 0,333333
```

Рисунок 2 – результат работы команды ncd (просмотр файла в sublime text)

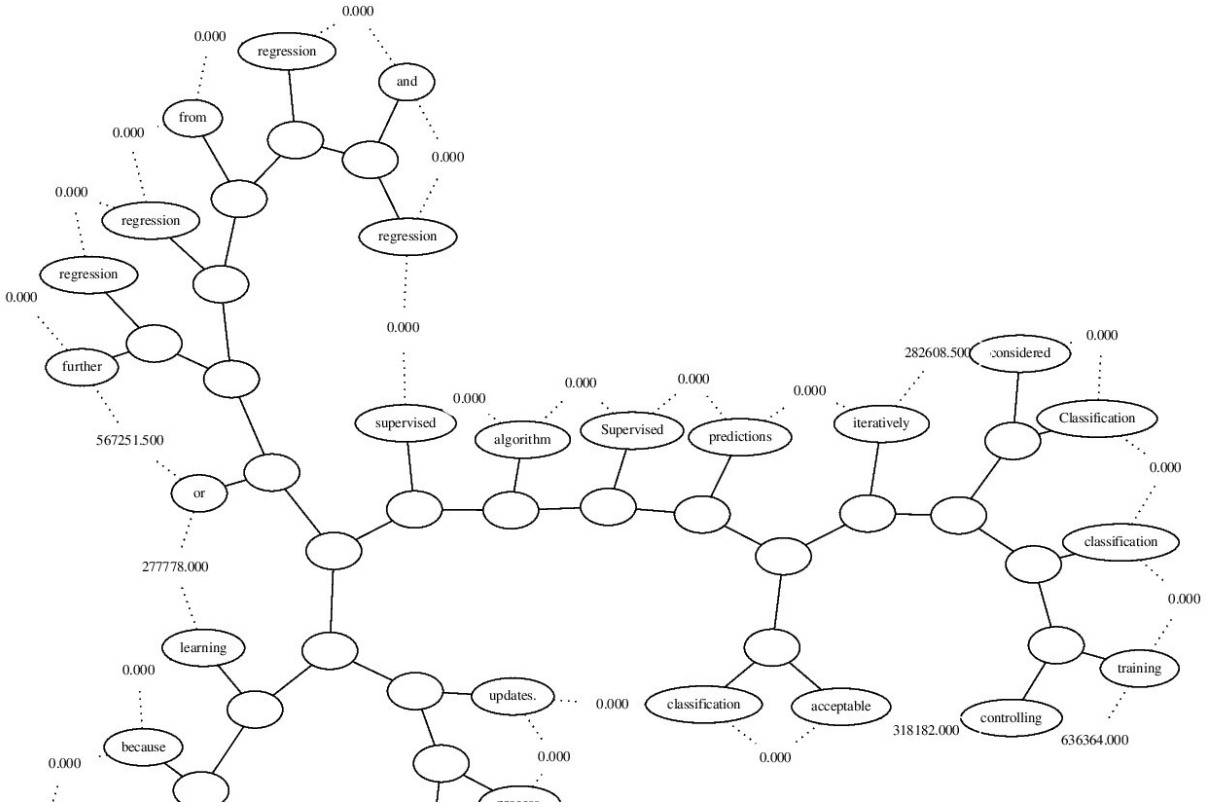

Рисунок 3 – фрагмент полученного дерева

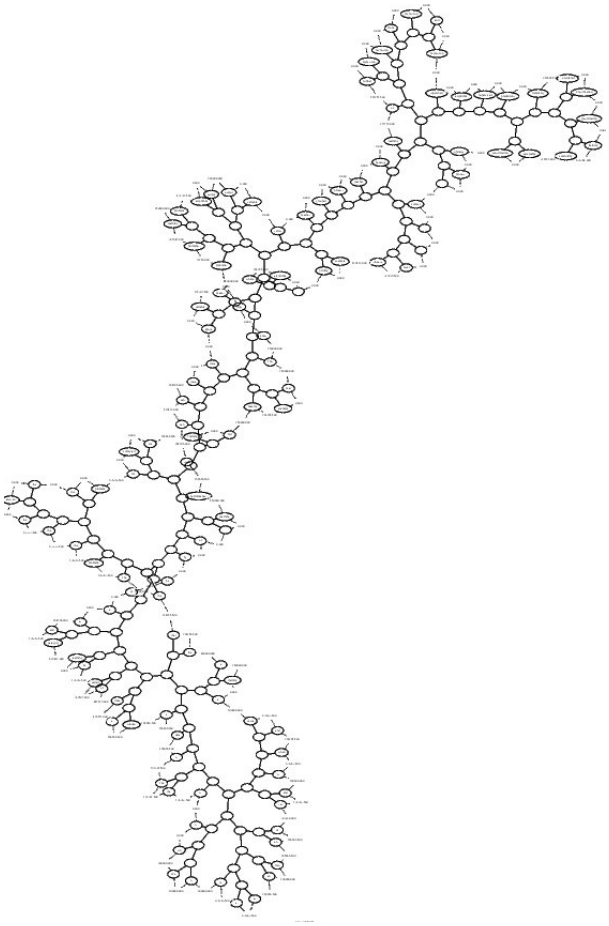

Рисунок 4 – общий вид полученного дерева

Вывод: В результате выполнения лабораторной работы была изучена программа CompLearn, с помощью которой были получены значения нормализованного расстояния сжатия между текстами, а также были получены навыки использования данной программы для построения соответствующих деревьев.