Отчет о выполненной практики

**Задача практики:**

Сделать базу данных. Из чего она будет состоять: вопрос по языку Python, и, соответственно, ответ на него в виде некоторого программного текста. Первоначально из стандартного PythonQA. Ответ может быть как в форме программы/фрагмента программы на языке Python, так и на естественном языке. В первую очередь необходимо загрузить в базу различных QA на русском языке. Всякие FAQ и т.п. Имеем два типа базы данных: первый - первоначальный, второй тип - дополнив из StackExchange на русском языке. Далее попытаться реализовать хоть как-то вопросно-ответную систему. Реализовать классическую поисковую систему. Т.е. есть запрос (некий вопрос) - и мы ищем наиболее близкий вопрос из БД. Как это делается - поисковым образом. Именно обработка естественного языка сводится к тому, что мы делаем некоторый набор слов - который является образом вопроса. В вопросе ищем косинусовую меру, т.е. ищем наиболее близкий вопрос из того, что есть в базе данных - и выдаем ответ на него.

BERT - огромная универсальная модель. Необходима будет реализация поисковой системы с попыткой ее улучшить, использовав механизм типа BERT - это новизна!

База данных не просто фрагмент текста - ответ, но и индексация. Потом индексируя запрос и используя меру по типу косинусовой близости. Нужно сделать, чтобы соответствующий модуль индексации можно было хорошо менять. Т.е. поисковый образ документа состоит в наборе ключевых слов, которые приведены к нормальной форме. Это классическая схема поиска и такой подход старый. Когда мы индексируем документы, создаем поисковый образ, и потом используя различные метрики idf (чтобы снижать роль незначащих слов). Что тогда должно входить в образ - word embedding (вектор признаков, по модели). Это векторное представление слов (то, что использует Bert). Как это работает: у нас есть текст, мы выкалываем произвольное слово. Есть некоторая последовательность слов. И какова вероятность для слов того, что оно появится на этом месте. Приятно то, что для этого не нужны какие-то дата-сеты, потому что любой текст является дата-сетом. Чтобы набрать статистику слов - этих текстов должно быть очень много. Почему это смогли сделать Google и Facebook - у них были огромные тексты на разных языках (именно на естественном языке), из которых они набрали соответствующую статистику. Соответствующим образом обученная сеть - она каким-то образом сама выбирает набор параметров, которые соответствуют каждому слову. Что из себя представляют эти параметры - никто не знает. Далее они подаются на вход этим моделям и они выдают, что на такой вопрос надо ответить вот так. Мы должны хорошо представлять механизм индексации в том плане, что если нужно, то чтобы мы могли быстро сменить на любой другой более сложный. Или дополнить другим более сложным, например word embedding.

**Результаты практики:**

Найдены соответствующие данные StackExchange на русском языке. Произведен детальный анализ и установка логических взаимосвязей. Далее был реализован и применен парсер, который конвертирует полученные данные в JSON. Из этих данных была сформирована PostgreSQL БД с интересующими нас таблицами и данными, опираясь на реализованный движок. Благодаря этому engine в дальнейшем возможна удобная доработка более сложных модулей индексации. Индексация реализована через GiST (generalized search tree в PostgreSQL), а также попытка реализации различных других индексов через CREATE INDEX.