Схема моделей и их описание для МТС.


Опросы
Морфологическая и лексикографическая обработка, очистка.
Исправление орфографии.
Отсев неинформативных и пустых комментариев.
Разметка данных.
Анализ тональностей.
Выделение сущностей, создание абстрактов.
Классификация текстов.
Создание новых категорий.
Формирование рекомендаций.
Визуализация и отчёты.
Сохранение в базу.

Опрос.

- Данные опросов: оценки, комментарии, ответы на вопросы, данные о пользователях.
- Размер датасета: не менее 10000 единиц (для достижения точности многоклассовой классификации в 95%). Это порядка 300 опросов (считаем, что если два человека отвечают на одни и те же вопросы, то это разный опрос).

Морфологичесткая и лексикографическая обработка. Очистка.

- Делаем очистку и восстановление пропущенных значений стандартными методами.
- Удаляем стоп-слова.
- Используем natasha или аналоги для морфологической обработки.
- Используем scikit learn или аналоги для лексикографической обработки.

Исправление орфографии.

- Смотрим на редкие слова, используем строковые метрики из textdistance или аналогов для сравнения со словарём и делаем замены.
- Сохраняем исправления для последующей проверки.
- Использования словая ручных исправлений.

Отсев неинформативных и пустых комментариев.

- Комментарии со словами не из корпуса.
- Использование информации полученной при ручном просмотре.
- Комментарий только со служебными словами.
- Сохраняем такие комментарии для последующей проверки.
- В сложном случае отдельная классификационная модель.

Выделение сущностей, создание абстрактов.

- Выделение сущностей производится с помощью natasha или аналогов.
- Создание абстрактов (суммаризация) делается с помощью spacy или аналогов.
- LDA с помощью gensim.

Разметка данных.

- Основные клиенты: классификация текстов, sentiment analysis.
- Размер сэмпла определяется, исходя из размера полученного датасета и возможностей команды.

Анализ тональностей.

- Используем spacy или аналоги для анализа тональностей.
- Формируем датасет для обучения, несколько категорий, тренируем модель классификации текстов.

Классификация текстов.

- Naive Bayes.
- SVM.
- spacy
- Нейронные сети (pytorch или keras).
- fasttext.
- Использование готовых embedings (natasha), сравнение их.

Создание новых категорий.

- Группировка на основе результатов суммаризации, выделения сущностей.
- Кластеризация embeddings (с помощью hdbscan).
- Группировка по результатам LDA.

Сохранение в базу.

- От базы в основном требуется только сохранение на данном этапе.
- Я бы использовал MongoDB. Могут возникнуть подгруппы в процессе.

Визуализация и отчёты.

- plotly для интерактивных графиков.
- flask для создания веб-приложения.
- Не очень большой кусок, но важный для пользователя.

Формирование рекомендаций.

- Не очень большой кусок, но важный для пользователя.
- Выдаём наиболее частые темы.
- Если для данного человека есть критические темы, то мы их выдаём.
- Согласовываем список критических тем.

Библиотеки.

- scikit learn, содержит классическую обработку и классические алгоритмы
- textdistance, содержит около 30 алгоритмов сравнения строк
- natasha, так как она заточена именно под русский язык, содержит несколько разных пакетов, хорошая производительность
- spacy, так как заточен под production, поддерживает в том числе и русский язык, можно решать разные задачи nlp
- gensim, так как это один из способов делать LDA
- pytorch или keras, если будем писать нейронные сети для классификации сами
- fasttext, один из способов создания embedings, поддерживает русский язык, решает классификацию текстов
- hdbscan, один из способов кластеризации, число кластеров определяется в ходе работы, минимальный размер кластера —- один из параметров.
- plotly, самый распространённый вариант создания интерактивных графиков.
- flask, самый простой способ создания веб-приложений