

Детекция эмоций. Сравнение и анализ классических методов машинного обучения и методов обучения с трансформерами

Панин Никита
Научный руководитель: К.В. Воронцов

МГУ

Dec 8, 2023

Введение

Задача: Классификация эмоций на основе датасета из твитов

Проблема: Размеченных данных мало, нужна хорошая модель

Актуальность:

- ▶ В социальных сетях распознавание агрессивных эмоций помогает выявить потенциальных преступников или террористов
- ▶ Мониторинг эмоций в реальном времени на основе данных социальных сетей может помочь в профилактике самоубийств
- ▶ В маркетинге анализ предпочтений потребителей помогает улучшить бизнес-стратегии

Цель: Сравнительный анализ классических моделей машинного обучения (wKNN, Multinomial Bayes Classifier, Logistic Regression, SVM) и моделей глубокого обучения (в качестве предобученной модели брались BERT, RoBERTa, BERTweet и их large-версии)

Постановка задачи

1. Каждый твит d_i векторизовался в вектор $x_i \in \mathbb{R}^h$, описывающий структуру d_i , где h - размерность вектора x_i .
2. Полученный эмбединг x_i поступал на вход одному из перечисленных выше алгоритмов классификации \mathcal{A} и на выходе получали вектор вероятностей эмоций $\mathcal{A}(x_i) = p_i$, $\sum_{k=1}^5 p_{ik} = 1$.
3. Итоговым ответом алгоритма была эмоция, соответствующая наибольшей вероятности. В качестве эмоций рассматривалось 5 типов: нейтральная эмоция, грусть, страх, радость, гнев.

Описание данных

Данные частично получены из набора данных WASSA-2017. Было выбрано по 1500 твитов для каждой из четырех эмоций: страх, грусть, радость и гнев. К этим четырем категориям был добавлен еще один класс из 1500 нейтральных твитов с CrowdFlower.

На обучение 80% данных
Для теста и валидации по 10%.

Используемые метрики

Для каждого класса $y \in Y$:

- ▶ TP_y – верные положительные
- ▶ FP_y – ложные положительные
- ▶ FN_y – ложные отрицательные

Точность, полнота и F1 мера с **макроусреднением**:

$$Precision_{macro} = \frac{1}{|Y|} \sum_y \frac{TP_y}{TP_y + FP_y}$$

$$Recall_{macro} = \frac{1}{|Y|} \sum_y \frac{TP_y}{TP_y + FN_y}$$

$$F1_{macro} = \frac{2 * Precision_{macro} * Recall_{macro}}{Precision_{macro} + Recall_{macro}}$$

Классическое машинное обучение.

Предобработка данных.

- ▶ Эмодзи были переведены в текстовый формат.
- ▶ Все хэштеги были заменены словами без символа решетки. Например, #hello заменяется на hello.
- ▶ Пользователи часто упоминают других пользователей в своих твитах через @username. В текстах все упоминания пользователей убирались.
- ▶ URL-адреса в твитах были удалены.
- ▶ Стоп-слова удалялись. Стемминг также использовался и это еще один метод повышения производительности модели путем сведения производных слов к грамматическому корню, называемому "stem".

Классическое машинное обучение.

Векторизация и модели.

Векторизация: tf-idf

Модели:

- ▶ weighted KNN с *cosine_similarity*(x_i, x_j)
- ▶ logistic regression с правдоподобием (log-loss) и регуляризацией:

$$L(w) = \sum_{i=1}^I \log(P(y_i|x_i, w)) - \frac{\tau}{2} \sum_{y \in Y} \|w_y\|^2 \rightarrow \max_w$$

$$P(y|x, w) = \frac{\exp(\langle w_y, x \rangle)}{\sum_{z \in Y} \exp(\langle w_z, x \rangle)} = \text{SoftMax}_{y \in Y}(\langle w_y, x \rangle)$$

- ▶ multinomial naive bayes
- ▶ SVC с линейным, полиномиальным и RBF ядрами

Трансформеры. Предобработка и модели.

В работе рассмотрены три основные модели: BERT, RoBERTa, BERTweet, а также их large-аналоги, в которых больше слоев и больше обучаемых параметров.

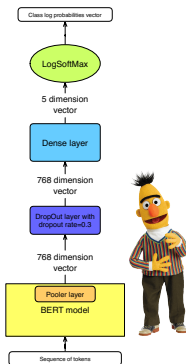


Рис.: Предложенная модель

Классическое машинное обучение. Результаты.

Лучший результат был получен с помощью SVM с rbf-ядром благодаря способности данного метода находить нелинейные зависимости сложной структуры

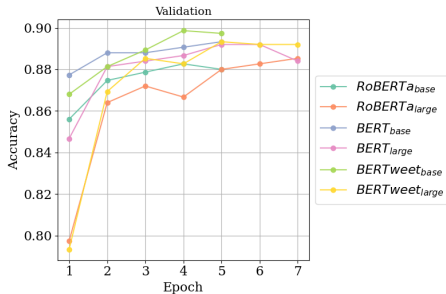
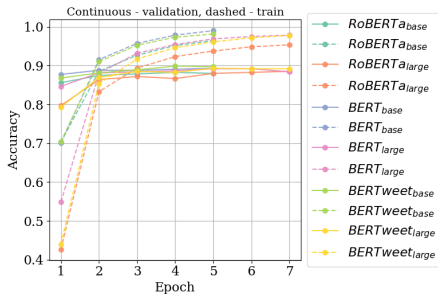
	precision	recall	F1-score
<i>wKNN</i>	0.701	0.691	0.693
<i>MNB</i>	0.778	0.78	0.776
<i>LR</i>	0.834	0.832	0.833
<i>SVM</i>	0.873	0.871	0.871

Таблица: Сравнение метрик качества моделей на валидации.

Трансформеры. Результаты.

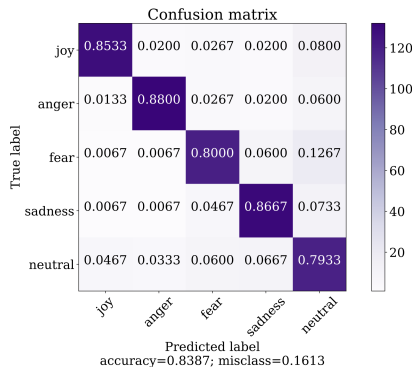
Из всех моделей самое высокое значение $accuracy = 0.8987$ было получено у модели *BERTweet*, что объясняется двумя факторами. Во-первых, *BERTweet* была предобучена по той же схеме, что и *RoBERTa*, а она, в свою очередь, является улучшенной версией *BERT*, которая превзошла *BERT* во многих задачах. Во-вторых, *BERTweet* предобучена на 850M твитов и как раз ориентирована на наш датасет.

Training history

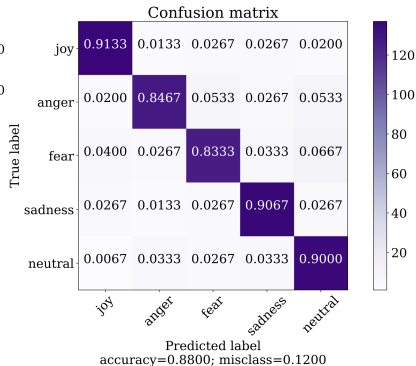


Сравнение SVC и BERTweet.

Новые возможности: ансамбли



(a) Матрица ошибок для SVC



(b) Матрица ошибок для BERTweet

Выводы

- ▶ SVM с rbf-ядром и TF-IDF при правильной обработке данных дает неплохие результаты ($accuracy = 0.8387$) при том, что обучение занимает меньше времени чем у трансформеров.
- ▶ Модель с BERTweet позволяет получить наилучшее качество для данных из твитов с $accuracy = 0.88$.
- ▶ BERT версии с большим количеством параметров, так называемые, large-модели, не показали в данной задаче заметного улучшения в сравнении с base-версиями (меньшим числом параметров) и не смогли превзойти BERTweet из исходной статьи.

Дальнейшие исследования

- ▶ использовать более умные эмбединги для классических моделей машинного обучения, которые не просто собирают статистики, а могут запоминать контекст слов (GloVe, FastText, ELMO)
- ▶ проверить гипотезу о возможном улучшении классификации в случае добавления в эмбединги информации о наиболее часто встречаемой эмоции для этого слова, собранной из словарей, в которых каждому слову сопоставлена эмоция

Thank You