Применение BERT для классификации сообщений в службу поддержки SAP

С. С. Масленникова, email: sveta.maslennikova@gmail.com
В. В. Коротков, email: chasecrunk@gmail.com

Воронежский государственный университет

Аннотация. *В статье рассматривается применение нейросетевой языковой модели BERT для решения задачи классификации сообщений на русском языке. Проводится сравнительный анализ эффективности нескольких основных реализаций модели на примере классификации запросов, поступающих в русскоязычную службу поддержки SAP.*

Ключевые слова: *BERT, Transformer, классификация, машинное обучение, обработка естественного языка.*

Введение

SAP SE – это немецкая компания, производящая программное обеспечение по управлению такими внутренними процессами предприятия, как бухгалтерский учет, производство, финансы, торговля, планирование, управление складскими запасами и человеческим капиталом и т.д. Самым известным продуктом компании является SAP R/3 – ERP-система, ориентированная на крупные и средние предприятия. В связи с ее популярностью в России региональная служба поддержки пользователей SAP вынуждена обрабатывать большое число поступающих запросов. Для оптимизации этого процесса может быть использован подход автоматической классификации сообщений с целью их перенаправления профильным специалистам. Это бы в значительной степени сократило среднее время обработки запроса пользователя.

Для решения задач обработки естественного языка успешно применяются различные нейросетевые модели, среди которых одной из самых перспективных является нейронная сеть BERT от корпорации Google [1]. Она использует векторные представления слов, в которые вставляются специальные последовательности символов – токены. В основе BERT лежит архитектура Transformer. Обученная на больших объёмах данных модель способна понимать связь между фразами и предложениями текста. Затем ее возможно дообучить на размеченных данных, специфичных для необходимой задачи. Добавлением слоев для

классификации может быть получен эффективный инструмент автоматической категоризации текстов [2].

1. Анализ данных

Перед применением моделей машинного обучения следует ознакомиться с исходными данными. Часто это помогает глубже понять задачу и улучшить качество модели.

В данном исследовании использовался датасет обращений в службу поддержку SAP, содержащий две колонки: сообщение и категорию, к которой оно относится. Всего набор содержит 15449 записей и 77 разных категорий. В среднем на каждый класс приходится 200 сообщений. На рис. 1 представлено распределение записей по категориям. По горизонтальной оси отложены категории, по вертикальной оси – количество сообщений, относящихся к этому классу.

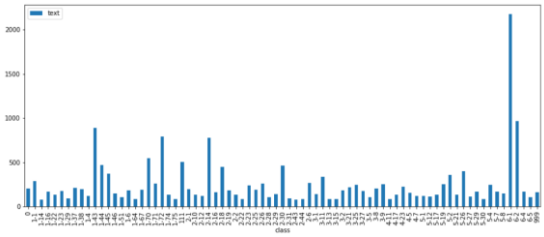

Рис. 1. Распределение сообщений по категориям

Из графика видно, что набор данных не сбалансирован – в каждой категории представлено разное число объектов с сильным разбросом по количеству сообщений. Очевидно, что при распознавании сообщений, относящихся к категориям с малым количеством данных, могут возникать трудности.

2. Анализ реализаций BERT на русском языке

В данной работе использовалась библиотека на языке Python под названием *transformers* [3]. Она содержит множество предобученных моделей для различных задач, в том числе для классификации текстов.

Для сравнительного тестирования были отобраны несколько моделей, обученных на данных разного размера. Приведём их основные характеристики.

– *bert-base-multilingual-cased*. Полноценная нейронная сеть BERT, обученная на большом корпусе текстов из Википедии и поддерживающая 104 языка, включая русский. Модель можно

использовать как для извлечения признаков, так и для решения каких-либо задач. Также эта модель чувствительна к регистру слов [4].

– *DeepPavlov/rubert-base-cased-sentence*. Модель, основанная на BERT и обученная специально для русского языка. Может взаимодействовать с предложениями и чувствительна к регистру [5].

– *sentence-transformers/LaBSE*. Модель, основанная на векторном представлении предложений и BERT от TensorFlow. Поддерживает 109 языков, включая русский [6].

– *sberbank-ai/sbert_large_nlu_ru*. Реализация BERT, обученная на расширенном наборе данных, предназначена для получения векторных представлений предложений. Модель не чувствительна к регистру [7].

3. Валидация результатов

Исходный набор данных необходимо разделить на обучающую и валидационную выборку. В данном исследовании разбиение производилось таким образом, чтобы сохранить исходное соотношение категорий, наблюдавшееся в изначальном наборе данных. 90% выборки было отнесено к обучающему набору, а 10% – к валидационному.

В силу несбалансированности исходных данных использование доли правильных результатов (*accuracy*) в качестве метрики эффективности модели может дать необъективный результат, поэтому для этой цели применялась так называемая F_1-мера (*f1-score*) [8]. Она представляет собой среднее гармоническое точности и полноты:

$$F1 = 2\frac{Precision * Recall}{Precision + Recall} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

Точность (2) – это доля меток, действительно принадлежащих этому классу, ко всем меткам, которые модель посчитала принадлежащими к данному классу, а полнота (3) – это доля объектов, определенных как принадлежащие к данному классу, ко всем объектам этого класса из выборки. F_1-мера хороша тем, что близка к 1, когда точность и полнота близки к единице, и близка к 0, если один из аргументов близок к нулю.

Т.к. рассматривается задача многоклассовой классификации, итоговая метрика складывается на основе метрик отдельных классов:

$$F1_{class1} * W_1 + F1_{class2} * W_2 + \ldots + F1_{classN} * W_N \quad (4)$$

Здесь w_i – нормированное количество меток, действительно принадлежащих i -тому классу.

$$F1_{class1} + F1_{class2} + \ldots + F1_{classN} \quad (5)$$

4. Подготовка данных и обучение сети

Первый этап подготовки данных – преобразование меток классов. В исходном наборе они представлены строками, поэтому необходимо заменить их числовыми значениями.

Второй этап – разбиение датасета на обучающий и валидационный. Размер валидационного набора данных составил 10% от данных для обучения. Из-за небольшого объема данных в ряде категорий модели увеличение размеров валидационной выборки привело бы к снижению эффективности распознавания.

Третий этап – создание векторных представлений сообщений и добавление токенов. Как было сказано ранее, BERT использует векторное представление текста, в которое вставлены специальные последовательности символов. Для подготовки данных использовался класс *BertTokenizer* из библиотеки *transformers*.

Четвертый этап – выбор алгоритма оптимизации, используемого при обучении нейронной сети, и создание соответствующего объекта. В данной работе в качестве такового применялся алгоритм *Adam* [9].

Пятый этап – установка *random seed* для воспроизводимости результатов. Если пропустить этот этап, то каждый раз результат будет случайным, так как алгоритмы глубокого обучения имеют стохастическую природу.

Шестой этап – загрузка подготовленной модели и ее дообучение. Обучение модели происходит за 4 эпохи: так как набор данных небольшой, при большем количестве эпох модель может переобучиться, а при меньшем можно не увидеть динамики обучения. После каждой эпохи модель сохраняется, чтобы была возможность проанализировать результаты, полученные на разных эпохах.

5. Анализ результатов

Результаты тестирования каждой модели представлены в табл. 1-4. Указаны значения основных метрик на каждой из четырех эпох.

Таблица 1

bert-base-multilingual-cased

Эпоха	Training loss	Validation loss	F_1 Score
1	2.71157	1.78050	0.53735
2	1.45097	1.25363	0.67771

3	1.01519	1.14616	0.70543
4	0.95474	1.05798	0.75609

Видно, что после четвертой эпохи модель переобучается. Лучшие результаты были получены на четвертой эпохе, f_1-*score* составил 0.756.

Таблица 2

DeepPavlov/rubert-base-cased-sentence

Эпоха	Training loss	Validation loss	F1 Score
1	2.51496	1.60243	0.57140
2	1.25237	1.11871	0.71879
3	0.75998	1.04339	0.75325
4	0.51782	1.04175	0.78429

Видно, что после второй эпохи модель переобучается. Лучшие результаты были получены на второй эпохе, f_1-*score* составил 0.718.

Таблица 3

sentence-transformers/LaBSE

Эпоха	Training loss	Validation loss	F_1 Score
1	2.09881	1.18195	0.69625
2	0.92560	0.87460	0.77579
3	0.64006	0.90777	0.79061
4	0.48680	0.97134	0.79374

Видно, что после второй эпохи модель переобучается. Лучшие результаты были получены на второй эпохе, f_1-*score* составил 0.775.

Таблица 4

sberbank-ai/sbert_large_nlu_ru

Эпоха	Training loss	Validation loss	F_1 Score
1	2.12313	1.13220	0.71142
2	0.82872	0.92813	0.78970
3	0.43810	0.87423	0.79303
4	0.25218	0.82134	0.80885

Видно, что после второй эпохи модель переобучается. Лучшие результаты были получены на второй эпохе, f_1-*score* составил 0.789.

Сводные результаты моделей вынесены в табл. 5. Как видно, лучше всего с задачей справилась реализация BERT, предобученная на расширенном наборе данных. Благодаря расширенному набору данных модели было проще выделить закономерности, характерные для конкретной задачи.

Таблица 5

Анализ результатов моделей

Модель	F_1 Score
bert-base-multilingual-cased	0.75609
DeepPavlov/rubert-base-cased-sentence	0.71879
sentence-transformers/LaBSE	0.77579
sberbank-ai/sbert_large_nlu_ru	0.78970

Заключение

Таким образом, было рассмотрено несколько разных реализаций модели BERT, способных работать с русским языком. Они были дообучены и протестированы на задаче классификации текстов. Модель на основе *sberbank-ai/sbert_large_nlu_ru* показала лучший результат и может быть использована в дальнейшем для решения задачи классификации сообщений, поступающих в службу поддержки SAP. Подход, описанный в статье, также может быть использован для решения схожих задач.

Список литературы

1. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [Электронный ресурс] : архив. – Режим доступа : https://arxiv.org/abs/1810.04805

2. Sun, C. How to Fine-Tune BERT for Text Classification? / C. Sun [et al.] // The proceedings of the 18th China National Conference on Computational Linguistics (Kunming, October 18–20, 2019). – Kunming, 2019. – P. 194-206.

3. Библиотека transformers [Электронный ресурс] : документация. – Режим доступа : https://huggingface.co/docs/transformers/index

4. Документация модели bert-base-multilingual-cased [Электронный ресурс] : документация. – Режим доступа : https://huggingface.co/bert-base-multilingual-cased

5. Документация модели DeepPavlov/rubert-base-cased-sentence [Электронный ресурс] : документация. – Режим доступа : https://huggingface.co/DeepPavlov/rubert-base-cased-sentence

6. Документация модели sentence-transformers/LaBSE [Электронный ресурс] : документация. – Режим доступа : https://huggingface.co/sentence-transformers/LaBSE

7. Документация модели sberbank-ai/sbert_large_nlu_ru [Электронный ресурс] : документация. – Режим доступа : https://huggingface.co/sberbank-ai/sbert_large_nlu_ru

8. Tharwat, A. Classification assessment methods / A. Tharwat // Applied Computing and Informatics. – 2021. – Vol. 17 – P. 168-192.

9. Kingma, D. P. Adam: A Method for Stochastic Optimization / D. P. Kingma, J. Ba // 3rd International Conference on Learning Representations (San Diego, May 7-9, 2015). – San Diego, 2015.