Использование нейронных сетей для нечёткого сопоставления текстов

А. Н. Матвеев
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)
andrewsamsunguser@gmail.com

***Аннотация*. В работе рассматривается задача нечеткого сопоставления текстов, в частности рабочих программ и текстовых описаний проектов, с целью выдачи рекомендаций студентам. Создан бенчмарк для оценки качества сопоставления с помощью F1-score и методов кросс-валидации. Бенчмарк представляет собой набор пар вида: “рабочая программа – проект”. Для решения задачи рассмотрена и оценена на бенчмарке модель, использующая архитектуру "трансформер" двумя методами оценки производительности.**

Ключевые слова: трансформер; метрика F1-score; максимизация

I. Введение

Студенту в рамках его образовательной траектории доступен выбор из большого количества активностей (рабочих программ, проектов и т. п.), и они определяются документами на естественном языке. Для предоставления рекомендаций в данном выборе рассматривается задача нечеткого сопоставления текстов. Например, для оценки соответствия проекта студенту можно рассмотреть соответствие рабочих программ дисциплин, изученных студентом, и текстового описания этого проекта.

Целью исследования является тестирование предобученной модели, использующей архитектуру «трансформер».

Таким образом, в качестве рассматриваемой проблемы выступает сложность сопоставления текстов на естественном языке в контексте рекомендаций индивидуальной образовательной траектории студента. Объектом исследования являются тексты на естественном языке. Предметом исследования является нейросетевая модель, предназначенная для сопоставления текстов на естественном языке. Результатом сравнения текстов является число от 0 до 1, характеризующее семантические сходства двух сравниваемых предложений. Таким образом, для достижения поставленной цели предстоит решить следующие задачи:

1. описать исследуемую модель;
2. описать метрику производительности F1-score;
3. описать способ максимизации метрики F1-score;
4. описать метод кросс-валидации модели с разбиением на train и test части и предоставить результаты его применения;
5. описать метод кросс-валидации модели Leave One Out и предоставить результаты его применения.

II. Описание исследуемой модели

Наиболее подходящая архитектура для задачи нечеткого сопоставления текстов – трансформер, благодаря тому, что данная архитектура предполагает глубокое обучение, что позволяет выявлять зависимости между элементами, которые находятся на большом расстоянии друг от друга, то есть позволяет анализировать контекст больших размеров. При этом использование данной архитектуры допускает параллельные вычисления, в отличие от рекуррентных нейросетей, которые так же предполагают глубокое обучение, но при этом имеют проблемы с распараллеливанием.

В качестве исследуемой модели была позаимствована модель [1]. Данная модель инициализирована RuBERT и настроена на SNLI [2] с google-переводом на русский и на русскоязычной части Википедии на SNLI dev set [3]. Представления предложений являются векторными представлениями токенов (эмбеддингами токенов), которые были усреднены операцией mean pooling, аналогично тому, как это происходит в Sentence-BERT [4].

Имеется набор пар текстов, которые необходимо сравнить, в частности, текст рабочей программы и текст проекта и на основании числа сходства (вещественное число от 0 до 1) предоставлять наиболее подходящие под описание рабочей программы проекты (чем ближе число к 1, тем более тексты семантически совпадают). Для того, чтобы оценить качество работы модели, нужно оценить F1-score.

III. Метрика производительности F1-score

Для начала введем следующие понятия:

- TP – истинно позитивное предсказание. В данном случае предполагается, что моделью было предсказано семантическое совпадение текстов рабочей программы и проекта, и предполагалось, что тексты действительно должны были сопоставляться, как подходящие друг другу;
- TN – истинно отрицательное предсказание. В данном случае предполагается, что моделью было предсказано семантическое несовпадение текстов, и предполагалось, что тексты действительно не должны были сопоставляться;
- FP – ложное положительное предсказание. Моделью было предсказано семантическое совпадение текстов, однако предполагалось, что тексты не должны были сопоставляться;

- FN – ложно отрицательное предсказание. Предполагалось, что тексты семантически схожи, однако модель предсказала их несовпадение.
- Precision – точность – мера того, сколько из сделанных положительных предсказаний верны;

$$Precision = \frac{TP}{TP + FP}$$

- Recall – полнота – мера того, сколько положительных случаев модель предсказала верно среди всех положительных случаев в данных;

$$Recall = \frac{TP}{TP + FN}$$

На практике невозможно одновременно максимизировать точность и полноту, поэтому необходима метрика, соединяющая информацию о точности и полноте. Такая метрика называется F1-score и является средним гармоническим точности и полноты.

$$F1 = \frac{2 * Recall * Precision}{Recall + Precision}$$

F1 позволяет работать с данными даже в тех случаях, когда они несбалансированы. Таким образом, F1-score включает в себя множество аспектов и позволяет увидеть общую картину качества работы модели.

IV. Максимизация F1-score

Необходимо подобрать такой порог числа сходства (cutoff), чтобы F1-score был максимален. Наиболее простой способ для этого – перебрать все значения числа сходства от 0 до 1 с некоторым шагом (например, 0.02) и выбрать то число, на котором F1 будет максимальным (пример результатов работы такого метода на рис. 1).

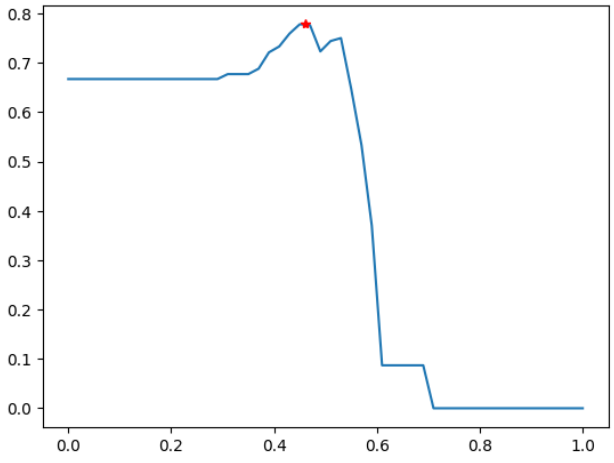

Рис. 1. Максимизация F1-score

В наборе данных текстовых описаний было рассмотрено 44 пары текстов, 22 из которых должны быть сопоставлены, а 22 не должны. Согласно данному графику, максимальный F1-score – 0.778, а порог сходства текстов – 0.46.

V. Расчет метрики для исследуемой модели

A. *Описание набора данных текстовых описаний*

Для этого были рассмотрены пары “рабочая программа – проект”, и о каждой паре доступна информация, должны ли рабочая программа и проект сопоставляться или нет. Для того, чтобы определить, сопоставляются ли они, необходимо, чтобы число сходства было не меньше значения определенного порога. Этот порог необходимо подобрать. Следует подбирать его таким образом, чтобы F1-score была максимально возможной. Наиболее простой способ выполнить данный перебор – перебрать все возможные значения порогов от 0 до 1, при этом на каждой итерации вычислять F1-score. Значение порога, на котором F1-score будет максимальным, будет принято в качестве искомого.

B. *Предобработка текстов*

Также стоит поговорить о том, как предобрабатываются тексты перед формированием векторных представлений. Если при работе с неглубокими нейросетями текст разбивают на токены, то в случае с трансформером, который выполняет построение векторных представлений для целых предложений, необходимо разбивать текст на предложения. Сначала для каждого предложения проекта строятся эмбеддинги, после чего для каждого предложения рабочей программы строятся эмбеддинги. Далее, в список cosine similarity (список чисел сходства) записываются попарные числа сходства проекта и рабочей программы. В список максимальных чисел сходства записывается только одно максимальное число сходства одного предложения рабочей программы всех предложений текста проекта. Иными словами, для каждого предложения рабочей программы в список чисел сходства записывается максимальное число сходства для всех проектов. В качестве результирующего числа сходства двух текстов возвращается усредненное число сходства из максимальных чисел сходства рабочих программ и проектов.

C. *Описание метода кросс-валидации модели с разбиением на train и test части*

1) Исходный набор данных текстовых описаний случайным образом разбивается на train и test части в пропорции 50/50;

2) На train-части максимизируется F1-score и запоминается cutoff;

3) На test-части вычисляется F1-score при числе сходства, полученном при максимизации на train-части. Чем ближе результаты F1-score на train и test части, тем выше качество работы модели.

Применим данный метод. На рис. 2 приведена максимизация F1-score на train-части набора данных. Как видно из табл. 1, F1-score значительно отличается на train и test части.

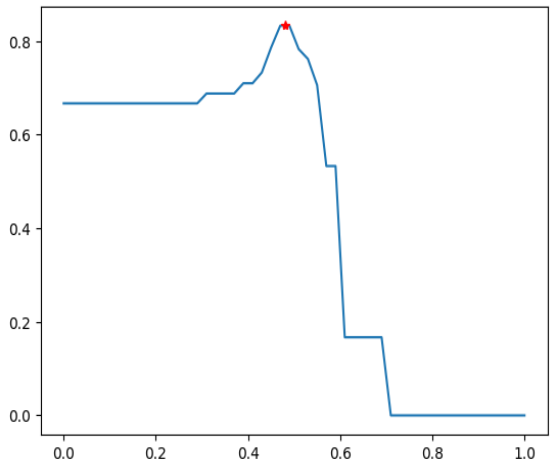

Рис. 2. Максимизация F1-score на train-части набора данных

ТАБЛИЦА I.

	Максимизированный F1-score	**Cutoff**
train	0.83	0.48
test	0.72	0.48

Cutoff – порог числа сходства

Проблема этого метода состоит в том, что с учетом перемешивания текстов при формировании train и test части F1-score и cutoff каждый раз немного изменяются, что негативно свидетельствует о применимости такого метода на небольших наборах данных (исследуемый набор данных невелик и имеет 44 пары текстов), рис. 3.

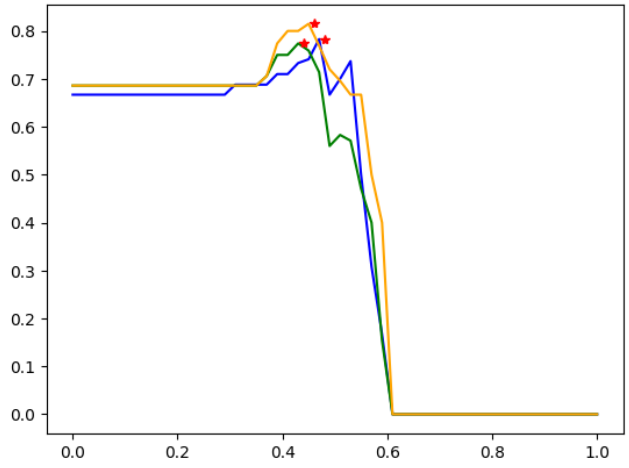

Рис. 3. Проблема нестабильности оценки F1-score

D. Описание метода кросс-валидации модели Leave One Out

Данный метод имеет название Leave One Out [5].

1. Рассматривается весь набор данных – все имеющиеся пары.
2. Сначала пары текстов кодируются в эмбеддинги.
3. Далее производится N итераций, где каждая итерация соответствует одной паре.
4. На каждой i-ой итерации строится набор данных без учёта i-ой пары.
5. Для него максимизируется F1-score описанным ранее способом.
6. Если в списке чисел сходства соответствующее число больше или равно полученному порогу i-ой итерации, то в список предсказаний добавляется истинное значение, иначе – ложное.
7. На конечной стадии вычисляется F1-score по списку предсказаний.

Данный метод предполагается использовать на небольших наборах данных.

Псевдокод данного метода приведен ниже.

```
predictions = список предсказаний (изначально пустой)
Для всех пар:
  current_df = набор данных без i-ой пары
  current_sim = список чисел сходства (sim) без i-ой пары
      steps, thresholds, max_, cutoff = max_f1_score(current_sim, current_df)
  Если sim[i] >= cutoff: (если число сходства не меньше полученного на текущей итерации порога)
    Predictions добавить (True)
  Иначе:
    Predictions добавить (False)
calculate_f1_score (predictions, df)
```

При применении Leave One Out F1-score был ниже, однако результат не изменялся вне зависимости от перемешивания пар текстов и составил 0.731.

VI. Заключение

В результате исследования все задачи выполнены и поставленная цель достигнута. Описана метрика производительности F1-score, а также исследуемая модель, рассмотрен способ максимизации метрики F1-score, рассмотрены и применены на практике оба метода кросс-валидации модели, а также был проведен расчет метрики F1-score для исследуемой модели. Установлено, что метод, связанный с вычислением F1-score на train-выборке и валидацией на тестовой, нестабилен для наборов данных небольшого размера и чувствителен к тому, какие тексты будут находиться в train-части, а какие в test-части. При работе с небольшими наборами данных более корректно применять Leave One Out для оценки качества модели. Для дальнейших исследований, связанных с анализом текстов относительно семантической схожести, можно рассмотреть методы, использующие неглубокие нейронные сети (word2vec/fastText [6]).

Список литературы

[1] DeepPavlov/rubert-base-cased-sentence (дата обращения: 20.03. 2023)

[2] Bowman S.R., Angeli G., Potts C., Manning C.D. (2015) A large annotated corpus for learning natural language inference. arXiv preprint arXiv:1508.05326 (дата обращения: 23.03. 2023)

[3] Williams A., Bowman S. (2018) XNLI: Evaluating Cross-lingual Sentence Representations. arXiv preprint arXiv:1809.05053 (дата обращения: 26.03. 2023)

[4] Reimers N., Gurevych I. (2019) Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. arXiv preprint arXiv:1908.10084 (дата обращения: 28.03. 2023)

[5] Emily Black, Matt Fredrikson (2021) Leave-one-out Unfairness arXiv preprint arXiv:2107.10171 (дата обращения: 29.03. 2023)

[6] Luc R., Bajger A. Evaluation of the stability of word embeddings from FastText and Word2Vec. – 2022 (дата обращения: 04.04.2023)