Оценка методов авторской атрибуции для русскоязычных текстов

Солонин Михаил Андреевич (Solonin37@mail.ru)
Abbyy, Россия, Москва

In this paper we attempt to consider some basic methods of multiclass classification for solving the problem of author attribution. We also present the dataset of Russian literary works to estimate the quality of modern architectures and understanding the ability to solve a problem of classification for specific data. In addition to the above, this work contains a description of the competition in which participants were asked to build an optimal method for classifying the works of Russian authors. Their results, along with those obtained by us, are described and analyzed in detail.

Keywords: Authorship attribution, authorship identification, text classification.

В данной работе предпринимается попытка рассмотреть основные методы многоклассовой классификации текстов для решения задачи авторской атрибуции. Мы также представляем датасет русскоязычных литературных произведений для оценки качества работы моделей и понимания возможностей современных архитектур к решению задачи классификации для специфичных данных. Помимо вышеизложенного, эта работа содержит описание соревнования, в котором участникам было предложено построить оптимальный метод классификации трудов русскоязычных авторов. Их результаты, вкупе с теми, что получены нами, детально описаны и проанализированы.

Ключевые слова: Авторская Атрибуция, Идентификация Автора, Классификация Текстов

1 Введение

Вопрос установления авторства играет все более возрастающую роль в современном мире, где Интернет-коммуникация становится превалирующим способом общения. Возможность деанонимизации текстов позволит снизить угрозу экстремизма и кибербуллинга. Кроме этого, идентификация автора как самостоятельная задача имеет большое значение с филологической и исторической точек зрения.

Авторская атрибуция - одна из классических задач компьютерной лингвистики (Juola 2008 [1]; Koppel et al. 2009 [2]; Alharthi et al. 2018 [3]). Существует множество подходов для решения данной проблемы, однако одни и тех же методы, примененные к различным наборам данных, могут показывать разное качество из-за специфики текстов. В данной работе мы проводим исследование, чтобы оценить, насколько качественно можно предсказать автора русскоязычного литературного произведения современными средствами.

2 Данные

Все данные были собраны из коллекции Генерального интернет-корпуса русского языка. [4] [5]. Многие тексты включали в себя фамилии авторов, ссылки на интернет ресурсы с работами писателей или некоторый html-код. Все такие вхождения были удалены, для того чтобы анонимизировать тексты и избежать обучения на избыточных признаках. Мы выбрали 75 авторов с наибольшим количеством текстов для того, чтобы уменьшить фактор нехватки данных для обучения модели и репрезентативного представления её качества. Помимо этого тексты длины 150 символов и меньше были отброшены. Средняя длина текстов - 24000 символов или 3500 токенов. Среднее общее количество текстов на одного известного автора - 78 для первой задачи и 140 для второй (см. пункт Задачи). Среди жанров работ присутствуют критика (Евгений Ермолин, Ольга Балла), стихи (Юрий Казарин, Евгений Степанов), проза (Борис Хазанов, Александр Мелихов), нон-фикшн (Кирилл Кобрин, Лев Бердников). Многие из этих авторов имеют произведения нескольких жанров: так, например, Кирилл Кобрин пишет и в жанре нон-фикшн, и в жанре критика. Общая концепция, впрочем, следующая: многие авторы имеют произведения как в разделе Fiction, так и в разделе Non-Fiction.

Отметим, что для задачи №2 (см. пункт Задачи) мы взяли первых 15 авторов по количеству текстов, и дополнили выборку текстами, принадлежащими не одному, а многим авторам, не входящим в число «известных». Такие тексты стоит воспринимать как «тексты неизвестного авторства». Их количество равняется 8202.

В дополнении к нашим данным, мы протестировали решения на Reuter_50_50 Data Set [6], так как данные в нём имеют схожие структуру и особенности. Отметим однако, что если количество авторов (50) и количество текстов на одного автора (100) близко к тому, что имеем мы, то средняя длина текста (3100 символов или 500 токенов) значительно ниже. Тексты также отличаются стилистически: если у нас это литературные произведения, то в датасете Reuter_50_50 это тексты статей в массмедиа, имеющие публицистическую направленность. Несмотря на некоторые различия в особенностях датасетов, Reuter_50_50 - один из наиболее близких к нашему набор данных, поэтому оценка качества моделей на нём также интересна.

3 Задачи

Мы исследовали задачу авторской атрибуции в двух постановках, каждая из которых моделирует определенный процесс установления происхождения текста.

3.1 Постановка задачи №1

Описание:

Задача состоит в следующем: дана выборка, состоящая из $60^{\text{ти}}$ текстов, для каждого текста из выборки необходимо определить одного из $60^{\text{ти}}$ «известных»

авторов. Действовать нужно в предположении, что существует биекция, то есть каждому тексту из выборки соответствует ровно 1 автор из числа «известных».

Метрика качества: accuracy. Тестовый набор состоит из нескольких (N) выборок (в каждой 60 текстов разных авторов), для каждой из которых нужно предсказать 60 писателей. Итоговой оценкой является усредненное по N accuracy.

Таким образом моделируется ситуация, когда для группы из N «известных» авторов нужно распределить N текстов, каждый из которых гарантировано написан одним из этих авторов.

3.2 Постановка задачи №2

Описание:

Задача состоит в следующем: из набора текстов нужно выбрать те, которые написаны одним из «известных» 15 авторов и указать, кому именно они принадлежат, остальные пометить как «текст неизвестного авторства».

Метрика качества: F1-score (macro average).

Таким образом моделируется ситуация, когда мы «знаем» стили нескольких писателей и должны научиться выделять тексты этих авторов из входного потока, а остальные тексты пропускать.

4 Исследования

4.1 Хакатон

Вышеуказанные задачи были предложены участникам хакатона «Я – профессионал» [7] в феврале 2019 года. Соревнующиеся команды представили широкий спектр решений, включающий использование RNN, CNN, fastText, и другие. Несмотря на то что многими конкурсантами были опробованы современные методы анализа текста, лидирующие позиции заняли команды, использовавшие TF-IDF в качестве способа векторизации документов и какой-либо классификатор для предсказания автора текстов. Максимальный суммарный балл, впрочем, набрала команда, существенным образом использовавшая условие биекции в первой задаче.

4.2 Наши эксперементы

Первым делом мы проверили решения команд-участниц хакатона для тщательной оценки качества и возможности более гибкого анализа результатов в дальнейшем.

Далее перечислим основные подходы к решению, которые мы проверили и оценили.

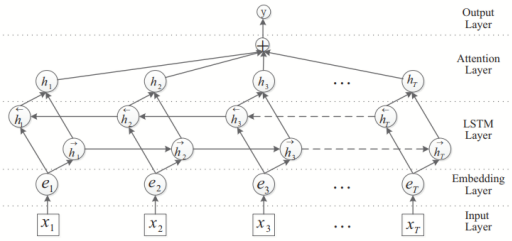
1. fastText классификатор [8]. Метод, при котором сначала каждому токену входного документа ставится в соответствие эмбеддинг этого токена, после этого векторное представление документа рассчитывается как усреднение

всех эмбеддингов токенов, в него входящих. Для предсказания автора текста обучается логистическая регрессия.

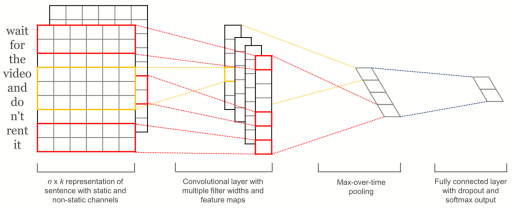
2. TF-IDF + classifier. В данном методе векторизация происходит с помощью TF-IDF Vectorizer, после чего на полученной матрице обучается какой-либо классификатор, например, XGBoost. После этого предсказания классификатора (то есть автор с максимальной вероятностью, предсказанной классификатором, для данного входного текста, векторизованного с помощью TF-IDF) выдаются в качестве ответов.

3. TF-IDF + classifier + optimizer. Метод, похожий на первый, однако вместо шага выбора автора с наибольшей вероятностью, мы используем метод, существенно опирающийся на условие биекции. Строится матрица 60×60, в которой хранятся вероятности того, что каждый конкретный текст принадлежит каждому конкретному автору. Затем к этой матрице применяется венгерский алгоритм* для поиска оптимального набора авторов, соответствующих текстам.

4. BLSTM + attention. Идея этой модели взята из Zhou et al (2016) [9]. Каждый входной токен преобразуется в эмбеддинг, который затем подается на вход BLSTM. Выходные векторы по каждому направлению конкатенируются на каждом шаге, после чего соединенные векторы подаются в слой attention. Этот слой выдает вектор, из которого, используя $Softmax$ получаем предсказание авторства текста.



5. CNN. Идея этой модели взята из Yoon Kim (2014) [10]. Архитектура модели изображена на рисунке. Основная идея: применение свёрток поверх словоформенных эмбеддингов, а также использование пуллингов, полносвязного слоя и $Softmax$ для предсказания авторства текста.



$\star$ Венгерский алгоритм позволяет найти минимальное по весу паросочетание среди максимальных по размеру. Используется для поиска биекции, максимизирующей вероятность правильных ответов.

5 Эксперементы и анализ результатов

В приведенной ниже таблице представлены результаты моделей для данных из Журнального Зала, проверенных на задачах №1 и №2, а также для данных Reuter_50_50 Data Set, проверенных на задаче №1.

Отметим, что все тексты были предобработаны перед тем, как быть поданными на вход классификаторам: пунктуаторы были отделены от слов, все табуляторы и непрерывные последовательльности одинаковых табуляторов переведены в пробелы.

	Reuter 50 50 Задача №1 (метрика Accuracy)	*Журзал* Задача №1 (метрика Accuracy)	*Журзал* Задача №2 (Метрика F1-macro)
FastText[0]	0.8	0.6	0.2
TF-IDF + classifier$^{\diamond}$	0.72	0.92	0.89
TF-IDF + classifier$^{\diamond}$ + optimizer	0.83	0.99	-
BLSTM + Attention[1]	0.68	0.36	0.64
CNN[1]	0.26	0.13	0.24

Таблица 1: Результаты экспериментов
$\diamond$ - здесь имеется в виду XGBoost Classifier

Как следует из Таблицы 1, наилучшим методом для классификации текстов для задачи в постановке №1 является комбинация TF-IDF, классификатора и оптимизатора. На задаче №2 модель TF-IDF + classifier показывает f1-score-macro в 0.89.

Отдельно обсудим результаты тестов, полученных на датасете Reuter_50_50. В работах Jagadeesh Patchala (2016) [13] и Smita Nirkhi (2014) [14] оценка качества моделей тоже производится на этом датасете. Авторы статей используют методы идентификации автора текстов, отличные от наших, однако также существенно опирающиеся на частоты слов. Полученные нами оценки соответствуют их результатам.

Хотя и утверждается [15], что для некоторых типов текстов нейросетевые методы дают лучшее представление, наш опыт показал, что наиболее качественный метод классификации «больших» текстов особой специфики - использование TF-IDF в сочетании с каким-либо классификатором. Подтверждение этому можно видеть в Таблице 1, где для первой задачи в обоих датасетах, Reuter_50_50 и нашем, использование данных методов с дополнительной оптимизацией показало результат, значительно превосходящий по качеству другие подходы.

Исходные коды архитектур:
0 - `https://github.com/facebookresearch/fastText` [11],
1 - `https://github.com/TobiasLee/Text-Classification` [12]

Из полученных результатов можно сделать вывод: в наборе «больших» текстов можно опираться на векторизацию документа с помощью TF-IDF для классификации текста, так как в пространстве признаков, полученных из TF-IDF Vectorizer, тексты различных авторов будут «хорошо» разделимы.

Заметим, что архитектура CNN могла показать невысокий результат из-за нескольких факторов:

1. Мы не использовани предобученные эмбеддинги для токенов, как это сделано, например, в Hughes et al. (2017) [16].
2. Мы сокращали тексты до длины в 128 токенов⋆, если исходный текст имел длину, большую данного числа. Возможно, имея такую малую часть оригинального текста модели «трудно» предсказать автора корректно.
3. Как утверждается в Yin et al. (2017) [17], такого рода нейронные сети чувствительны к изменениям гиперпараметров. Возможно, правильный их выбор смог бы улучшить представление модели CNN.

Из Таблицы 1 также следует, что нейросетевые модели дают слабые результаты для наших данных в постановке задачи в форме №1 относительно качества на датасете Reuter_50_50. Объяснений этому факту может быть масса, однако наиболее правдоподобным кажется существенная разница в длинах текстов. Как отмчалось выше, некоторые тексты были сокращены для возможности обработки нейронными сетями. По-видимому, эта процедура критично сказывается на более длинных текстах.

6 Дальнейшаяя работа

В будущих работах хотелось бы исследовать влияние стилистических или жанровых особенностей текстов на качествно классификации литературных произведений. Помимо этого стоит провести дифференцированное исследование проблемы авторской атрибуции для более коротких текстов или для авторов, имеющих сравнительно малое количество образцов письма. В дальнейших исследованиях мы также хотим сконцентрировать внимание на лингвистических характеристиках произведений для идентификации авторов.

7 Заключение

В данной работе мы проанализировали современные методы атрибуции авторов и пришли к выводу, что для достаточно «больших» текстов традиционные методы, такие как, например, сочетание векторизации TF-IDF и классификатора, могут дать результаты высокого качества, в то время как нейросетевые методы показывают гораздо более слабое представление.

Помимо исследований различных архитектур, мы представили и детально описали датасет русскоязычных произведений литературы, основанный на данных из Генерального интернет-корпуса русского языка [4] [5]. Этот датасет может быть использован в будущих работах на тему авторской атрибуции и в других областях компьютерной лингвистики.

⋆ При работе с BLSTM порог отсечения был равен 512

Список литературы

[1] Patrick Juola. *Authorship attribution.* Foundations and Trends® in Information Retrieval, M., 2008.

[2] Shlomo Argamon Moshe Koppel, Jonathan Schler. *Computational methods in authorship attribution.* Wiley Subscription Services, Inc., M., 2009.

[3] Stan Szpakowicz Haifa Alharthi, Diana Inkpen. *Authorship Identification for Literary Book Recommendations.* Proceedings of the 27th International Conference on Computational Linguistics, M., 2018.

[4] ГИКРЯ. Журнальный Зал. `http://magazines.russ.ru/`, May 2019.

[5] Belikov V. Kopylov N. Piperski A. Selegey V. Sharoff S. *Corpus as language: from scalability to register variation.* Компьютерная лингвистика и интеллектуальные технологии. По материалам ежегодной Международной конференции «Диалог» (2013), M., 2013.

[6] UCI. Corpus. `https://archive.ics.uci.edu/ml/datasets/Reuter_50_50`, May 2019.

[7] Yandex. Competition. `https://yandex.ru/profi/`, May 2019.

[8] Piotr Bojanowski Tomas Mikolov Armand Joulin, Edouard Grave. *Bag of Tricks for Efficient Text Classification.* https://aclweb.org/anthology/E17-2068, M., 2016.

[9] Jun Tian Zhenyu Qi Bingchen Li Hongwei Hao Bo Xu Peng Zhou, Wei Shi. *Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification.* Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, M., 2016.

[10] Yoon Kim. *Convolutional Neural Networks for Sentence Classification.* arXiv:1408.5882v2, M., 2014.

[11] Facebook. Code. `https://github.com/facebookresearch/fastText`, May 2019.

[12] Tobias Lee. Code. `https://github.com/TobiasLee/Text-Classification`, May 2019.

[13] Jagadeesh Patchala. *Data Mining Algorithms for Discovering Patterns in Text Collections.* University of Cincinnati, M., 2016.

[14] Dr.V.M.Thakre Ms.Smita Nirkhi, Dr.R.V.Dharaskar. *Stylometric Approach For Author Identification of Online Messages.* International Journal of Computer Science and Information Technologies, M., 2014.

[15] Kaggle. Competition. `https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge`, May 2019.

[16] Mark Hughes, Irene Li, Spyros Kotoulas, and Toyotaro Suzumura. Medical text classification using convolutional neural networks. *Studies in Health Technology and Informatics*, 235, 04 2017.

[17] Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. Comparative study of cnn and rnn for natural language processing. 02 2017.