Обнаружение мультиавторности в тексте

А.Фаттахов, Р.Кузнецова, В.Стрижов

Московский Физико-Тенический Институт

Abstract

В работе рассматривается задача улучшения качества обнаружения мультиавторности в тексте. Проблема обнаружения мультиавторности является актуальной в случае, когда необходимо проверить текст на наличие заимствований, но при этом нет доступа к внешним корпусам. При этом предполагается, что авторы имеют свой уникальный стиль написания, и смена стиля может быть сщуественным сигналом к подозрению на заимствование. Предлагается применение композиции базовых алгоритмов. В качестве базовых алгоритмов используются: 1) модели на основе текстовых статистик 2) рекуррентные сети LSTM на основе векторных представлений слов 3) сверточные сети с векторными представлениями символов в качестве признаков. Вычислительный эксперимент проводится на выборке с соревнования PAN-2018.

1. Введение

Рассматривается задача поиска заимствований в тексте [1]. Внешний анализ заимствований (external plagiarism detection) [2] заключается в попарном сравнении подозрительного текста с определенной коллекцией внешних текстов. Задача поиска внутренних заимствований (internal plagiarism detection) [3] состоит в обнаружении подозрительного текста без использования дополнительной коллекции, по которой ведется поиск. При этом алгоритмы анализа внутренних заимствований учитывают стиль письма и выявляют признаки в тексте, свойственные данному автору. Задача поиска внутренних заимствований имеет набор подзадач. В PAN2011 [4] необходимо найти участки в тексте, на которых происходит смена автора. В PAN2016 [5] задача расширяется на авторскую диаризацию (Author Diarization) [6], решение победителя [7]. Изложенные выше варианты задачи поиска заимствований сейчас решаются с помощью текстовых статистик. В [8] предложен статистический подход к решению задачи, основанный на признаках tf-idf: словесные n-граммы (в работе рассматриватеся только $n = 1$ и $n = 3$), пунктуации, части речи с использованием PST Tagbank Penn Treebank [9]. В [10] построена модель с использованием подобных признаков, но в постановке задачи обучения без учителя. Кроме того, в [10] применяется комбинация лексических, синтаксических

Email addresses: fattahov.ao@phystech.edu (А.Фаттахов), rita.kuznetsova@phystech.edu (Р.Кузнецова), strijov@phystech.edu (В.Стрижов)

Preprint submitted to кпукпукп *23 апреля 2018 г.*

признаков, которые зависят от содержания, но не обязательно от стиля автора. В [11] описан алгоритм отображения предложений в многомерное пространство, с последующим определением стилевой функции. Помимо статистических подходов применяются многообразия [12] и нейронные сети [13].

В PAN2018 [14] ставится новая задача. Требуется определить, написан текст одним автором или несколькими. Решение подразумевает использование только стилевых особенностей и игнорирование самого смысла текста.

В данной работе предлагается использование ансамбля алгоритмови и исследуется вклад каждого базового алгоритма в ансамбль. В качестве базовых алгоритмов используются: 1) модели на основе текстовых статистик 2) рекуррентные сети LSTM на основе векторных представлений слов 3) сверточные сети с векторными представлениями символов в качестве признаков. Работа алгоритма тестируется на данных с конкурса PAN2018 [14]. Для оценки используются Accuracy.

$$\mathsf{A} = \frac{1}{|D_k|} \sum_{\mathrm{k}} I(y_k = f(\mathbf{d}_k)) \tag{1}$$

2. Постановка задачи

Поставим формально задачу классификации текста. Дана выборка $D = \{(d_k, y_k)\}$, где d_k — текстовый документ, а $y_k \in Y = \{0, 1\}$ - является ли текст мультиавторным. Выборка разбита на обучающую D_t и контрольную D_v. Принята функция ошибки S. Требуется построить отображение $g : d_k \mapsto \mathbf{d}_k$ и модель классификации $f : \mathbf{R}^n \mapsto Y$, минимизирующую функцию ошибки S на контрольной выборке:

$$f = \frac{1}{|D_v|} \sum_{1}^{|D_v|} \arg\min_{f} S(f(\mathbf{d}_k), y_k | D_t) \tag{2}$$

$$S = -\frac{1}{|D_v|} \sum_{1}^{|D_v|} y_k \log f(\mathbf{d}_k) + (1 - y_k) \log (1 - f(\mathbf{d}_k)) \tag{3}$$

В разных базовых алгоритмах отображение $g : d_k \mapsto \mathbf{d}_k$ осуществляется разными способами. В 3.1 и 3.2 текст $\mathbf{d}_k$ разбивается на сегменты $\mathbf{s}_{ki}$: $\mathbf{d}_k = \bigcup_i \mathbf{s}_{ki}$, где сегмент - это несколько предложений. В 3.3 $\mathbf{d}_k = \bigcup_i \mathbf{w}_{ki}$, где $\mathbf{w}_{ki}$ - это векторное представление слова текста. В 3.3 и 3.5 $\mathbf{d}_k = \bigcup_i \mathbf{c}_{ki}$, где $\mathbf{c}_{ki}$ - это векторное представление символа текста.

3. Описание базовых алгоритмов

3.1. Алгоритм на основе текстовых статистик

3.1.1. Tf-idf слов

Для уникальных слов d'_{ki} текста считаются их частоты в данном тексте. Но так как мощность словаря слишком велика, то вместо частот слов используются частоты групп

слов $u_{kj} = \bigcup_i d'_{kij}$. Частоты групп определяются по следующей формуле:

$$g'(u_{kj}) = \sum_{\mathrm{i}\,=\,1}^{|d_k|} I(d_{ki} \in u_{kj}) \tag{4}$$

Функция, отображающая текст d_k в его векторное представление $\mathbf{d'}_k$ - это конкатенация всех $g'(u_{kj})$. Искомое отображение $g : d_k \mapsto \mathbf{d}_k$ определяется как $g(d_k) = \text{tf-idf}(\mathbf{d'}_k)$, где tf-idf [15] - это статистическая мера, используемая для оценки важности слова в контексте документа, являющегося частью коллекции документов. Кроме того, в эксперименте используются n-граммы слов.

3.1.2. Tf-idf символов

Базовый алгоритм аналогичен 3.1.1, но вместо слов w_{ki} используются симолы c_{ki} и их n-граммы.

3.2. Алгоритм на основе функции стиля

Предлагаемый алгоритм [11] работает с частотными признаками описания текста. В качестве таких признаков выбраны частоты встречаемости слов. Исходный текст подвергается предобработке: удаляются служебные символы, все буквы переводятся в нижний регистр. Также из текста удаляются стоп-слова. Текст разбивается на предложения. Затем формируется разбиение текста на сегменты s_{ki}: если длина очередного предложения меньше минимальной длины сегмента, к этому предложению добавляется следующее за ним — процесс повторяется, пока длина сегмента s_{ki} не превысит заданную минимальную длину. Минимальная длина сегмента является настраиваемым параметром алгоритма. Для каждого сегмента s_{ki} текста строится вектор признаков $\mathbf{s}_{ki}$. Затем строится статистика $\sigma(\mathbf{s}_{ki})$, которая сглаживается скользящим средним:

$$\sigma(\mathbf{s}_{ki}) = ||\mathbf{s}_{ki} - \mathbf{s}_{kave}|| \tag{5}$$

$$\mathbf{s}_{kave} = \frac{1}{n} \sum_{\mathrm{i}\,=\,1}^{n} \mathbf{s}_{ki} \tag{6}$$

$$\sigma'(\mathbf{s}_i) = \frac{1}{2n+1} \sum_{\mathrm{k}\,=\,\mathrm{i}\,-\,\mathrm{n}}^{i+n} \sigma(s_i) \tag{7}$$

Значения $\sigma'(s_i)$ исследуются на выбросы. С некоторого порога сегментов-выбросов текст классифицируется как мультиавторский.

$$f(\mathbf{d}_k) = I(\max_{i,j} |\sigma'(\mathbf{s}_{ki}) - \sigma'(\mathbf{s}_{kj})| > threshold) \tag{8}$$

3.3. Алгоритм на основе LSTM

Используется рекуррентная нейронная сеть на основе векторных представлений $\mathbf{w}_{ki}$ слов w_{ki}. В качестве отображения $e : w_{ki} \mapsto \mathbf{w}_{ki}$ возьмем представления слов, полученные с помощью модели word2vec [16]. $\mathbf{h}_i = LSTM(\mathbf{h}_{i-1}, \mathbf{w}_i)$ - внутреннее состояние сети на слове $\mathbf{w}_i$. После двух LSTM слоев конечное состояние передается на трёхслойный перцептрон с сигмоид активацией на последнем слое из одного нейрона. Архитектура сети приведена на (Рис. 1).

3.4. Улучшеннаый алгоритм на основе LSTM

Поскольку длина предложений в текстах из корпуса может составлять до 2000 слов, то наивное использование LSTM как в 3.3 дает не очень хорошие результаты. Поскольку внутреннее состояние ячейки LSTM не в состоянии фиксировать настолько длинные взаимодействия, то текст разбивается на сегменты s_{ki}, аналогичные 3.2. К каждому сегменту s_{ki} применяется своя LSTM_i, после чего внутренние состояния конкатенируются в один вектор. Полученный вектор подается на вход трехслойному перцептрону, аналогичному 3.3. Архитектура сети приведена на (Рис. 2)

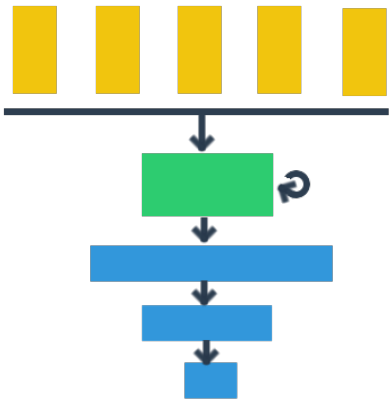

Рис. 1: Нейронная сеть на основе LSTM.

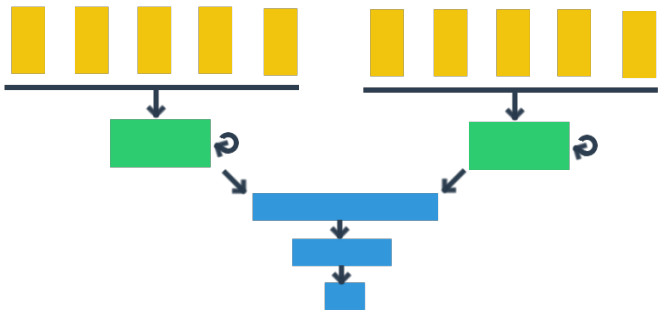

Рис. 2: Улучшенная архитектура на основ LSTM.

3.5. Сверточная сеть на основе символьных представлений текстов

Каждому уникальному символу c'_{ki} текста d_k ставится в соответствие one-hot-encoding вектор $\mathbf{c'}_{ki}$. Последовательность $\mathbf{c'}_{ki}$ подается на вход одномерной сверточной сети [17]. На выходе сети получется векторное представление $\mathbf{d_k}$ текста текста d_k. Полученное представление проходит через перцептрон, аналогичный 3.3, 3.4. Схема сети показана на (Рис. 3)

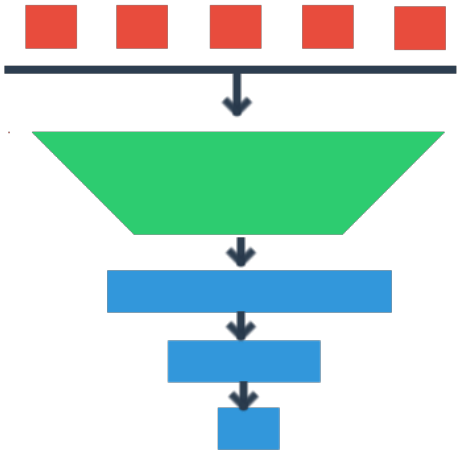

Рис. 3: Схема сети на основе char-CNN

4. Описание выборки

В работе используется набор текстовых документов с соревнования PAN-2018 [14], собранных с различных сайтов сети StackExchange [18]. Тематики и авторы у текстов различные. При этом у каждого текста может быть как один автор, так и несколько. Обучающая выборка состоит из 2980 текстов, при этом к каждому из них прилагается файл с разметкой. Несмотря на то, что стоит задача классификации, в разметке присутсвуют сами границы смены автора текста, что может помочь при разработке алгоритмов. Как видно из Рис. 4, количество заимствований линейно убывает, а большинство текстов имеет длину от двух до семи тысяч(Рис. 5).

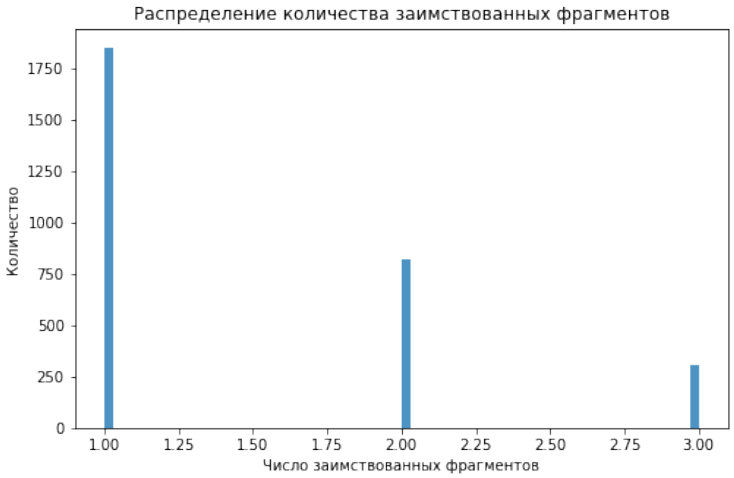

Рис. 4: Гистограмма распределения числа заимствований

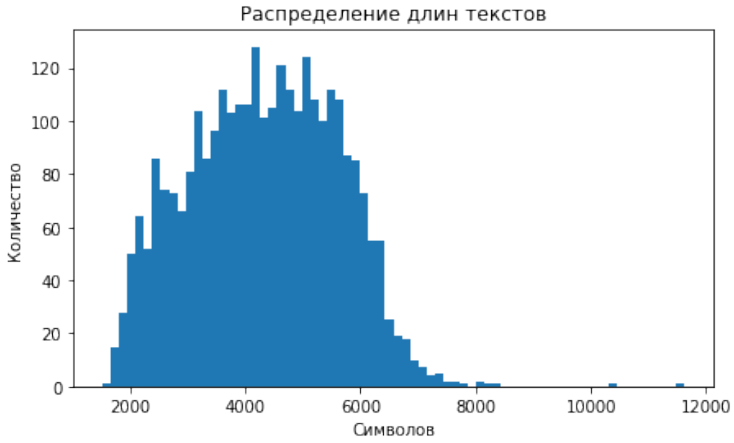

Рис. 5: Гистограмма распределения длины текстов

5. Композиция алгоритмов

В качестве композиции алгоритов предлагается использование стекинга [19]. Выборка документов D разбивается на k фолдов F_i: $D = \bigcup_{i=1}^{k} F_i$, при этом разбиение выбирается таким образом, чтобы распределения классов y_i в фолдах совпадали. Параметры каждого из алгоритмов 3.1-3.5 настраиваются на каждом из всевозможных объединений $k-1$ фолдов и делается предсказание на оставшийся фолд. После чего все предсказания объединяются в новую выборку D'_k, где i-й столбец выборки - это предсказания $f_i(d)$ соответствующей модели. Полученная выборка рзделяется на k аналогичных фолдов и обучается мета-модель $f' : D' \mapsto Y$. В данной работе в качестве метамодели используется логистическая регрессия. Качество базовых моделей и его стандартное отклонение, посчитанное по k фолдам приведено в таблице 1. На Рис. 6 показана матрица корелляций базовых алгоритов. Видно, что статистически подходы сильно отличаются от нейросетевых, при том что они дают примерно одинаковое качество. Это хорошо, потому что увеличивает разнообразность ансамбля. В таблице 2 приведено качество метамодели. На Рис. 7 показаны модули коэффициентов логистической регрессии, что можно интерпретировать как важность базовых алгоритмов в ансамбле.

Таблица 1: Качество базовых алгоритмов

	алгоритм1	алгоритм2	алгоритм3	алгоритм4	алгоритм5
logloss					
std					
Accuracy					
std					

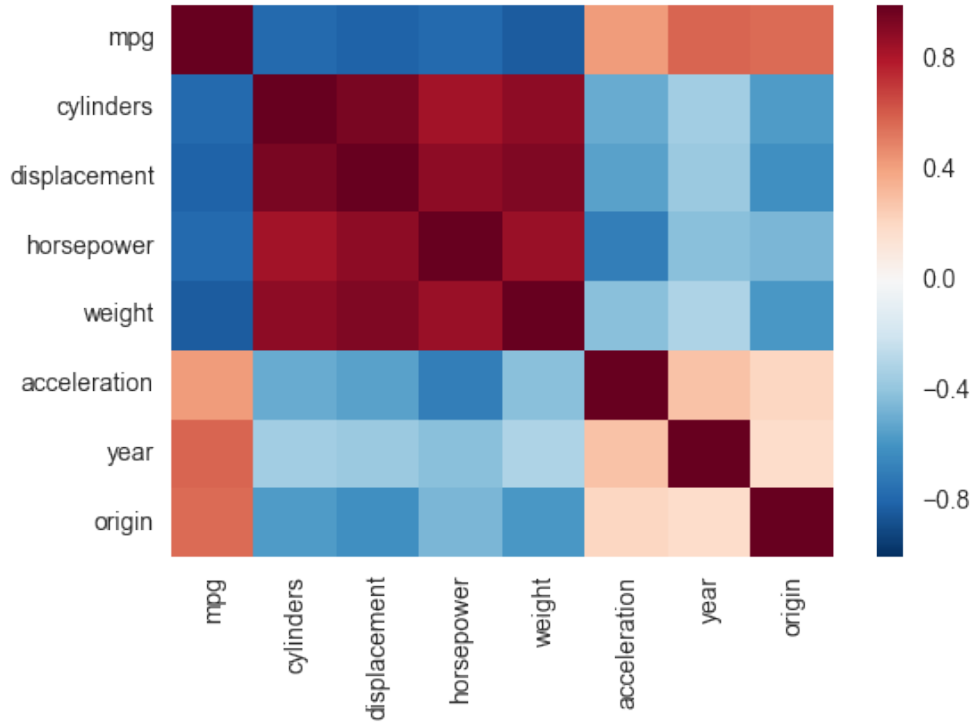

Рис. 6: Матрица корелляций базовых алгоритмов

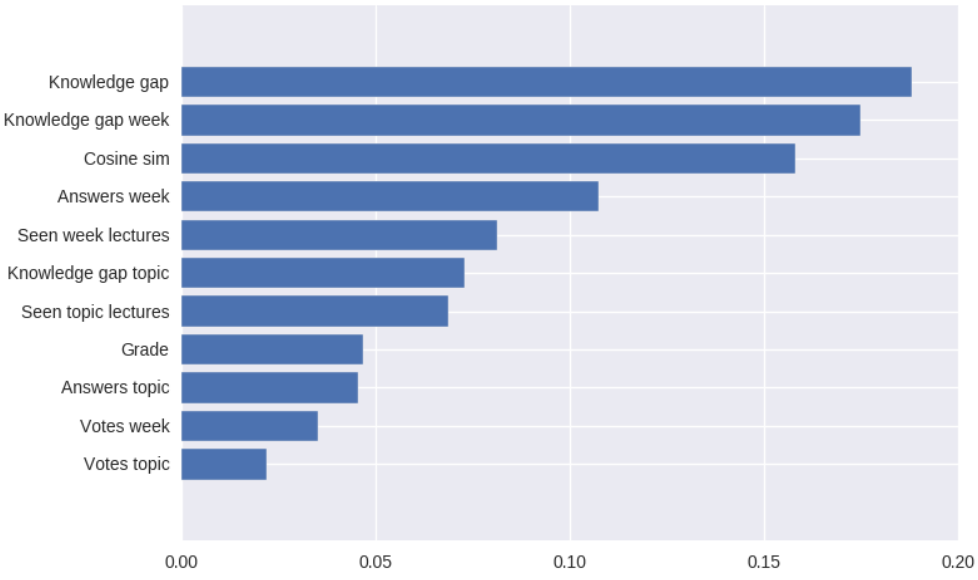

Рис. 7: Важности базовых алгоритмов в ансамбле

Таблица 2: Качество метамодели

	Метамодель
Accuracy	
std	

Список литературы

[1] S. W. P. G. D. o. C. S. C. Daria Sorokina, Johannes Gehrke, I. N. U. Information Science, Department of Physics Cornell University, Plagiarism detection, arXiv.
URL https://arxiv.org/pdf/cs/0702012.pdf

[2] V. S. R. Sobha Lalitha Devi, Pattabhi R K Rao, A. Akilandeswari, External plagiarism detection, CLEF 2010.
URL http://ceur-ws.org/Vol-1176/CLEF2010wn-PAN-DeviEt2010.pdf

[3] H. A. Chowdhury, D. K. Bhattacharyya, Plagiarism: Taxonomy, tools and detection techniques, arXiv.
URL https://arxiv.org/pdf/1801.06323.pdf

[4] Pan 2011.
URL http://pan.webis.de/clef11/pan11-web/

[5] Pan 2016.
URL http://pan.webis.de/clef16/pan16-web/

[6] H. R. I. Abdul Sittar, R. M. A. Nawab, Author diarization using cluster-distance approach, CLEF 2016.
URL https://pdfs.semanticscholar.org/1158/6f70a6bf976bf3bfc56c51c4e13e3fdd0168.pdf

[7] M. Kuznetsov, A. Motrenko, R. Kuznetsova, V. Strijov, Methods for Intrinsic Plagiarism Detection and Author Diarization—Notebook for PAN at CLEF 2016, in: K. Balog, L. Cappellato, N. Ferro, C. Macdonald (Eds.), CLEF 2016 Evaluation Labs and Workshop – Working Notes Papers, 5-8 September, Évora, Portugal, CEUR-WS.org, 2016.
URL http://ceur-ws.org/Vol-1609/

[8] D. Karaś, M. Śpiewak, P. Sobecki, OPI-JSA at CLEF 2017: Author Clustering and Style Breach Detection— Notebook for PAN at CLEF 2017.
URL http://ceur-ws.org/Vol-1866/

[9] Pst tagbank penn treebank.
URL https://www.ling.upenn.edu/courses/Fall$_2$003/$ling$001/$penn_treebank_pos.html$

[10] J. Khan, Style Breach Detection: An Unsupervised Detection Model—Notebook for PAN at CLEF 2017, in: L. Cappellato, N. Ferro, L. Goeuriot, T. Mandl (Eds.), CLEF 2017 Evaluation Labs and Workshop – Working Notes Papers, 11-14 September, Dublin, Ireland, CEUR-WS.org, 2017.
URL http://ceur-ws.org/Vol-1866/

[11] K. Safin, R. Kuznetsova, Style Breach Detection with Neural Sentence Embeddings—Notebook for PAN at CLEF 2017, in: L. Cappellato, N. Ferro, L. Goeuriot, T. Mandl (Eds.), CLEF 2017 Evaluation Labs and Workshop – Working Notes Papers, 11-14 September, Dublin, Ireland, CEUR-WS.org, 2017.
URL http://ceur-ws.org/Vol-1866/

[12] V. I.Molybog, A.Motrenko, IMPROVING CLASSIFICATION QUALITY FOR THE TASK OF FINDING INTRINSIC PLAGIARISM, Federal Investigation Centre for Information and Control, 2017. doi:10.14357/19922264170307.
URL https://doi.org/10.14357/19922264170307

[13] R. K. Kamil Safin, Style Breach Detection with Neural Sentence Embeddings, CLEF 2017, 2017.
URL https://pdfs.semanticscholar.org/c70e/7f8fbc561520accda7eea2f9bbf254edb255.pdf

[14] Pan 2018.
URL http://pan.webis.de/clef18/pan18-web/

[15] J. Ramos, Using tf-idf to determine word relevance in document queries, 2002.
URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.121.1424rep=rep1type=pdf

[16] G. C. J. D. Tomas Mikolov, Kai Chen, Efficient Estimation of Word Representations in Vector Space,

2013.
URL https://arxiv.org/abs/1301.3781
[17] Y. L. Xiang Zhang, Junbo Zhao, Character-level convolutional networks for text classification, 2015.
URL https://ru.sharelatex.com/project/5adc83a9405d7136e106ebfa
[18] Stackexchange network.
URL https://github.com/Harrix/Math-Harrix-Library
[19] Гущин, Методы ансамблирования обучающихся алгоритмов, 2015.
URL http://www.machinelearning.ru/wiki/images/5/56/Guschin2015Stacking.pdf