К вопросу о классификации зашумленных текстов*

В.А. Малых[I], В.А. Лялин[I]

[I] Лаборатория нейронных систем и глубокого обучения Московского физико-технического института (Государственный университет), г. Москва, Россия

Аннотация. Классическая задача классификации текстов была освещена во множестве работ, но существующие подходы в основном сосредоточены на улучшении качества классификации для так называемых чистых коллекций, не содержащих опечаток. В этой работе авторы приводят результаты исследования современных популярных моделей текстовой классфикации на предмет устойчивости к опечаткам для корпусов на русском и английском языках.

Ключевые слова: *нейронные сети; классификация текстов; устойчивость к шуму.*
DOI: 10.14357/20790279180520

Введение

Множество приложений для классификации текста, таких как анализ тональности или распознавание намерений, связано с данными, создаваемыми пользователем, где не может быть гарантирована правильная орфография или грамматика.

Классический подход к векторизации текста, такой как кодирование слов one-hot или TF-IDF-кодирование, сталкивается с проблемой отсутствия словарного запаса, учитывая огромное разнообразие орфографических ошибок. Хотя успешные приложения для задач с небольшим шумом на общеупотребимых наборах данных [8, 9] существуют, не все модели показывают хорошее качество на реальных данных, таких как комментарии или твиты.

В этой работе мы рассматриваем устойчивость моделей классификации текстов к искусственным шумам, призванным моделировать собой естественные шумы, а именно ошибки при наборе на клавиатуре.

Эта работа организована следующим образом. В разделе 1 описываются предыдущие работы по этой теме и базовые блоки, которые использовались в наших моделях, в подрразделе 1.2 описываются модели, о которых идет речь. Разделы 2 и 2.3 рассматривают постановку эксперимента и полученные результаты с последующей интерпретацией.

1. Модель классификации

В классической работе [18] рассматриваются популярные на тот момент методы классифкации тектов, например, SVM, но что более важно, в работе не рассматриваются методы устойчивости к шуму, а методы коррекции шума. То есть задача борьбы с шумом выносится на другой уровень рассмотрения. В работе [3], где векторное представление слов основано на векторном представлении символов, показаны современные подходы к классификации. В работе [17] авторы исследуют нормализацию медицинских концептов, которая на самом деле является классификацией многих возможных классов. Они используют сверточные сети с вниманием на уровне символов, и их модель шума (точнее 4 подобных и связанных модели) тесно связана с той, которая используется в этой работе.

Многие работы по классификации текстов в наши дни опираются на векторные представления слов, так что полезно рассмотреть также этот аспект. Например, модель FastText [1], где слово вложения для неизвестного слова генерируются ‘на лету’ из вложений составляющего символа n-грамм. Эта модель имеет ограниченную устойчивость к шуму в силу своего построения: в случае если в пришедшем слове нет известных модели n-грамм, то она не может сгенерировать векторное представление. В работе [2], с другой стороны представление слов генерируется на основе мешка букв, она не обладает описанным недостатком.

Для ширины перспективы следует упомянуть ненейросетевой подход, например [16], где авторы используют тематическое моделирование для улучшения классификации тональности.

До сих пор проблеме изучения влияния шума на добротность классификации было уделено мало внимания. В этой работе мы приводим результаты исследования этой проблемы и надеемся, что она простимулирует дальнейшие исследования в области.

* Исследования и разработки выполнены при поддержке Фонда поддержки проектов Национальной технологической инициативы и ПАО "Сбербанк". Идентификатор проекта 0000000007417F630002.

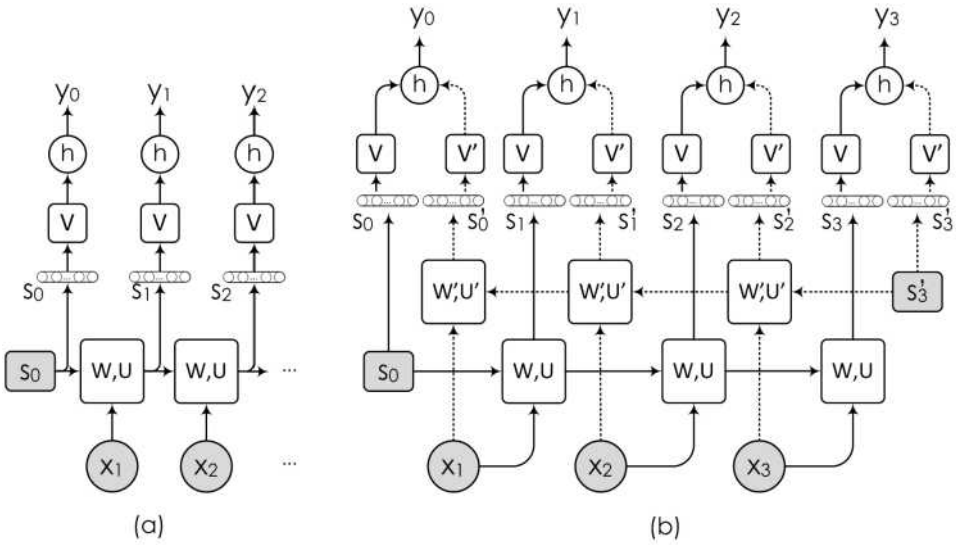

Рис. 1. Рекуррентные нейронные сети: (a) – обычная RNN; (b) – двунаправленная RNN

1.1. Архитектура

В этом разделе мы приводим примеры общих нейронных архитектур для классификации текста. Прежде всего, мы вводим базовые строительные блоки для современных нейронных архитектур: RNN и CNN. После этого в следующем разделе будут описаны фактические тестируемые архитектуры.

1.1.1. Рекуррентные нейронные сети

В то время, как полносвязные нейронные сети являются простейшим и в некотором роде наиболее общим видом нейронных сетей, существуют также более специфичные типы вычислительных графов, полезные в отдельных задачах. Одним из таких специфических типов вычислительных графов являются рекуррентные нейронные сети (RNN) [20]. RNN применяются для обработки данных, в которых есть выраженная последовательность, таких как временные ряды или последовательности слов. Ключевой особенностью этой архитектуры является обмен информацией между шагами по времени. В популярном варианте RNN, о которых мы будем говорить в подразделе 1.1.2, существует понятие состояния RNN, которое переходит на следующий временной шаг с текущего. Описанный механизм представлен на рис. 1.

1.1.2. Ячейка Gated Recurrent Unit

Gated Recurrent Unit (GRU) представляет собой общую архитектуру RNN, которая запоминает состояние между временными метками, тем самым смягчая проблему исчезновения градиентов. Он был введен в [13].

Формально он описывается в уравнениях (3). Обозначая через x_t входной вектор в момент времени *t;* через h_t, вектор скрытого состояния в момент времени *t;* по W_x (с разными вторыми индексами), матрицы весов, применяемые к входу; по W_h, матрицам весов в рекуррентных связях и b векторам смещения, получаем следующее формальное определение:

$$\begin{aligned} u_t &= \sigma(W_{xu}x_t + W_{hu}h_{t-1} + b_u), \\ r_t &= \sigma(W_{xr}x_t + W_{hr}h_{t-1} + b_r), \\ \mathrm{h}'_t &= \tanh(W_{xh'}x_t + W_{hh'}(r_t \odot h_{t-1})), \\ h_t &= (1-u_t)\odot \mathrm{h}'_t + u_t \odot h_{t-1}. \end{aligned} \quad (1)$$

1.1.3. Механизм внимания

У рекуррентных нейронных сетей есть широко известный недостаток: они быстро забывают информаую с предыдущих шагов по времени (см. например [15]). Чтобы уменьшить этот негативный эффект был предложен механизм внимания, который сам по себе является некоторым типом памяти. Ставший уже классическим первый подход был предложен в работе [14]. Идея, лежащая в основе механизма внимания, может быть представлена, как выбор «наиболее интересной» части входной последовательности для генерации выхода на текущем шаге по времени. Модель мягкого выравнивания выдает веса $\boldsymbol{a}_{ti}$, которые управляют тем, насколько каждое входное слово влияет на текущее выходное слово. Вес $\boldsymbol{a}$ показывает, должна ли сеть сосредоточиться на текущем слове прямо сейчас. v – это текстовый вектор, который собирает в себе всю информацию из входных слов. Модель тренируется end-to-end, так как внимание является мяг-

ким (действительно-значной функцией), градиенты могут протекать через сеть. Мягкое внимание существенно улучшает перевод и классификацию для длинных предложений и на сегодняшний день является подходом по умолчанию. Более формально оно описано в уравнениях (2) и представлено на рис. 2.

$$\upsilon_t = \tanh(W_\omega \left[\overrightarrow{h_t}, \overleftarrow{h_t}\right] + b_\omega)$$
$$\alpha_t = \frac{\exp(\upsilon_t^T \mathbf{u}_i)}{\sum_{j=1}^{T} \exp(\upsilon_j^T \mathbf{u}_i)} \quad (2)$$
$$\upsilon = \sum_{t=1}^{T} \alpha_t \left[\overrightarrow{h_t}, \overleftarrow{h_t}\right]$$

где W_w и b_w – параметры преобразования скрытых представлений; u_i – некоторый внешний вектор, в случае LSTM это может быть вектор состояния ячейки памяти c_i после обработки всей последовательности. u_i можно рассматривать как вектор контекста с том плане, что вектора, которые близки по контексту, должны иметь больший вес:

$$u_t = \sigma(W_{xu} x_t + W_{hu} h_{t-1} + b_u),$$
$$r_t = \sigma(W_{xr} x_t + W_{hr} h_{t-1} + b_r),$$
$$\mathrm{h}'_t = \tanh(W_{xh'} x_t + W_{hh'}(r_t \odot h_{t-1})) \quad (3)$$
$$\mathrm{h}_t = (1 - u_t) \odot \mathrm{h}'_t + u_t \odot h_{t-1},$$

где W_w и b_w – параметры линейного преобразования скрытых состояний; u_i – некоторый внешний вектор, в случае GRU это может быть скрытое состояние h_i в конце последовательности. u_i можно рассматривать как вектор контекста, что означает, что векторы, которые ближе к контексту, должны иметь больший вес.

1.1.4. Процедура Дропаут

Дропаут (dropout) – это специальное преобразование, накладываемое на входные веса любого слоя нейронной сети. Изначально эта процедура была описана в работе [19]. Для матрицы весов W размером $\boldsymbol{\kappa}$ x l, где $\boldsymbol{\kappa}$ – количество нейронов в слое, а l – количество входных весов для каждого из нейронов в слое, с помощью испытаний Бернулли генерируется бинарная матрица D аналогичного размера $\boldsymbol{\kappa}$ x l. Матрица W перемножается по Адамару с матрицей D и полученные значения используются вместо матрицы W. Матрица D может генерироваться для каждого объекта из обучающей выборки, но как правило, она фиксируется для всех объектов, входящих в один мини-батч.

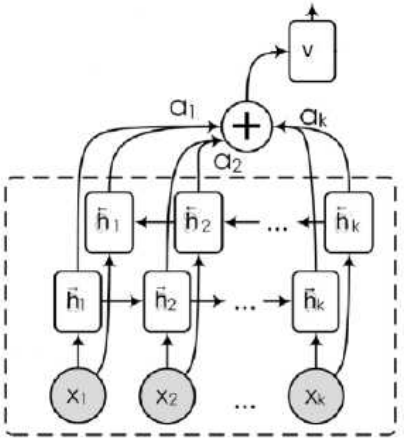

Рис. 2. Двунаправленная RNN с вниманием

Дропаут используется в процессе обучения нейронной сети для избежания так называемой ко-адаптации нейронов. В тестирования нейронной сети эта процедура не применяется.

1.2. Исследуемая модель

В наших экспериментах мы сравниваем производительность следующих моделей.

1.2.1. Архитектура CharCNN

Текст представлен в виде последовательности символов. Эта модель состоит из слоя векторного представления символов, слоя свертки с 256 фильтрами; размер ядра равен 15, а шаг – 2, за ним следует максимальное объединение с размером ядра 64 и шагом 32. После объединения мы применяем дропаут и преобразование 256-мерного скрытого вектора к 2 измерениям с помощью полносвязного слоя. Архитектура представлена на рис. 3.

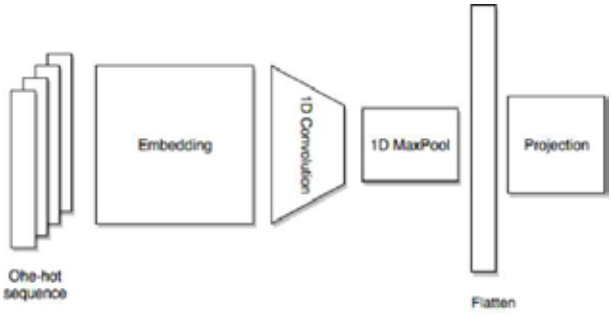

Рис. 3. CharCNN

1.2.2. Архитектура FastText-GRU

Текст представлен как последовательность 300-мерных векторов, построенных с использованием предварительно подготовленной модели FastText. Мы вводим эту последовательность в слой GRU с размерностью скрытого состояния 256. Затем дропаут применяется к последнему скрытому состоянию, а полученный вектор проецируется в двухмерное пространство.

1.2.3. Архитектура CharCNN-WordRNN

Эта модельная архитектура очень похожа на [3], за исключением отсутствия слоя highway layer. Каждое слово представляется в виде последовательности символов, а текст – как последовательность словесных представлений. Слова получают векторное представление с помощью сверточного слоя с размером ядра 5 и субдискретизируюещго слоя с максимумом по времени. Векторные представления поступают в слой GRU размерности 128, слои дропаут и проецирования – аналогично предыдущему разделу.

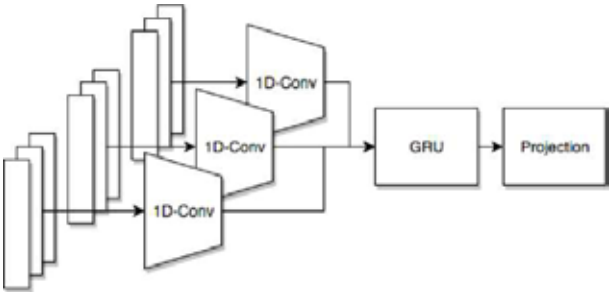

Рис. 4. CharCNN-WordRNN

1.2.4. Архитектура CharCNN-WordRNN с вниманием

Модель CharCNN-WordRNN, за исключением механизма внимания [14], применяется к скрытым состояниям GRU сразу после слоя GRU. Мы используем механизм внимания, аналогичный [10]. Поскольку в наших экспериментах мы используем одну так называемую «голову», это эквивалентно классическому вниманию, описанному в разделе 1.1.3.

2. Эксперименты и результаты

2.1. Наборы данных

Мы провели эксперименты на двух наборах данных: набор данных с отзывами о фильмах на английском языке и набор эмоционально окрашенных твитов на общие темы для русского языка.

Набор данных Movie Review описан в [4], состоит из 25 000 позитивных и 25 000 негативных обзоров фильмов из базы данных IMDB. Набор данных разделен поровну на тренировочную и тестовую выборки; это разделение опубликовано.

Набор данных Russian Twitter Sentiment был описан в работе [5]. Общедоступная версия этого набора данных состоит из 114 991 позитивных твитов и 111 923 негативных твитов, а также большого количества нейтральных твитов. В нашей работе мы используем только два класса – позитивный и негативный.

Поскольку для этого набора данных нет опубликованного разбиения, мы задали свое. Мы объединили позитивный и негативный наборы данных, перетасовали объединенный набор данных и использовали 70% в качестве тренировочной выборки, затем 15% в качестве набора вали- дационной, а последние 15% рассматриваются как тестовая выборка в наших экспериментах.[1]

Для этого набора данных следует упомянуть одну важную особенность: поскольку она автоматически собиралась фильтрованием, содержащим эмоджи (положительные и отрицательные соответственно), эти признаки легко запоминаются для любой модели, предсказывая упомянутые классы. Эмоджи в тексте – короткие последовательности символов – от одного до трех символов подряд. Такие последовательности не так легко повреждаются шумом, как сами слова, так как типичная длина слова в английском языке составляют 5,1 буквы, а для русского языка – 5,28 буквы [6]. Поэтому, чтобы усложнить задачу и исследовать свойства устойчивости, мы решили игнорировать знаки пунктуации, связанные с эмоджи, а именно: ‘)’ и ‘(’.

Следует упомянуть, что, кроме описанных, никаких трансформаций с данными не производилось.

2.2. Модель шума

Чтобы продемонстрировать устойчивость к шуму, была выполнена проверка орфографии[2] описанных наборов данных и искусственно введен в них шум. Шум моделируется следующим образом:

- вероятность вставки буквы после текущей,
- вероятность того, что буква будет пропущена

для каждой буквы входного алфавита, для каждой задачи.

На каждом символе из входной строки мы выполняем случайные вставки и удаления с помощью заданных вероятностей. А именно: для текущего символа с заданной вероятностью происходит (или не происходит) вставка символа после него. Для вставки символ берется равновероятно из алфавита текста. Для другого типа шума, с заданной вероятностью происходит удаление текущего символа.

Оба типа шума добавляются одновременно. Мы тестируем модели с различными уровнями шума от 0 (без шума) до 20%. Согласно [7], реальный уровень шума в пользовательских текстах составляет 10-15%.

[1] Авторы собираются опубликовать созданное разбиение для воспроизводимости.

[2] Для правки орфографии использовался общедоступный промышленный механизм проверки орфографии Yandex.Speller.

Табл. 1

Результаты экспериментов по оригинальному набору данных. F_1 на тестовом наборе

Модель	Movie Review	Twitter Sentiment
CharCNN	0.74	0.77
FastTextGRU	0.84	0.76
CharCNN-WordRNN	0.80	0.81
Attention	0.68	0.81

Были проведены три типа экспериментов:

- данные обрабатываются системой проверки орфографии, а после в них добавляется описанный искусственный шум; описанная процедура применяется как тренировочной, так и к тестовой выборке;
- берутся оригинальные данные;
- данные обрабатываются системой проверки орфографии, а после в них добавляется описанный искусственный шум; описанная процедура применяется только к тренировочной выборке, тестовая выборка остается без изменений.

Эти эксперименты призваны продемонстрировать добротность тестируемых архитектур для искусственного и естественного шума. Под добротностью авторы понимают скорость снижения качества модели с введением шумов различного уровня вероятности. Таким образом, более добротной полагается та архитектура, качество которой снижается меньше с ростом уровня шума. В качестве меры качества выбрана мера F_1 для классификации.

2.3. Результаты

Все модели обучаются с размером батча 32 оптимизатором Adam и дропаутом для последнего полносвязного слоя с вероятностью 0,5. Функция потерь – перекрестная энтропия. CNN инициализируются с помощью инициализации Хавьера [21] с нормальным распределением.

Ниже представлены результаты обучения и тестирования моделей без проверки орфографии или искусственного шума.

В табл. 1 представлены данные для сравнения со следующими результатами в шумной среде.

2.3.1. Набор данных Movie Review

На рис. 5 представлены результаты в следующей среде: обучающая выборка очищается от естественного шума и вводится искусственный; тестовая выборка также является предметом описанной трансформации. Мы можем видеть, что модель FastText-GRU – лучшая без представленного шума, но она не настолько добротна. Самая добротная модель на этом наборе данных модель – CharCNN-WordRNN. Неожиданно, что модель Attention является наихудшей в этом эксперименте.

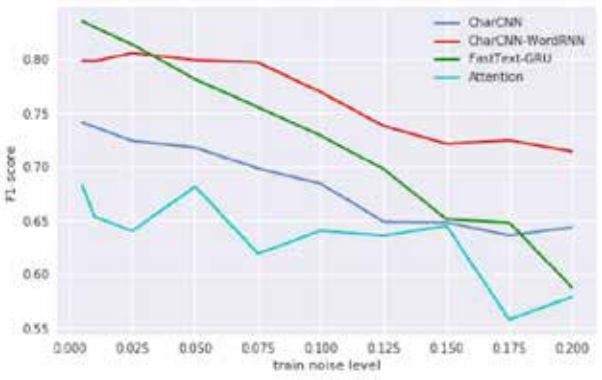

Рис. 5. Набор данных Movie Review. Тренировка на данных с исправленными естественными опечатками и добавленными искусственными, тест на данных с исправленными естественными опечатками и добавленными искусственными с тем же уровнем шума, что и на тренировочной выборке

На рис. 6 представлены результаты в следующем окружении: обучающая выборка очищается от естественного шума и вводится искусственный; набор данных для тестирования остается неизменным. Поведение всех моделей примерно одинаково, но модель Attention демонстрирует более высокую добротность на реальных данных теста.

2.3.2. Набор данных Russian Twitter Sentiment

На рис. 7 представлены результаты в следующем окружении: обучающая выборка очищается от естественного шума и вводит искусственный; тестовая выборка также является предметом описанной трансформации. Мы можем заметить, что порядок тестируемых моделей остается неизмен-

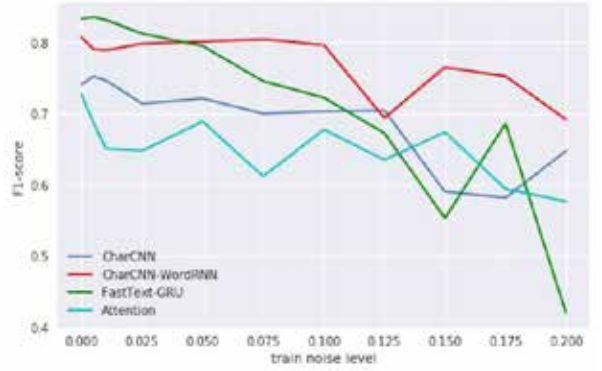

Рис. 6. Набор данных Moview Review. Тренировка на данных с исправленными естественными опечатками и добавленными искусственными, тест на оригинальных данных

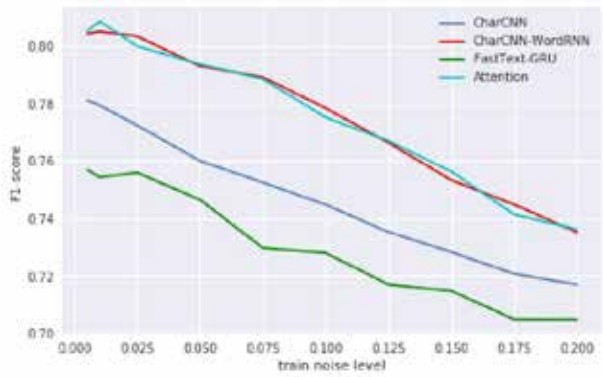

Рис. 7 Набор данных Twitter Sentiment. Тренировка на данных с исправленными естественными опечатками и добавленными искусственными, тест на данных с исправленными естественными опечатками и добавленными искусственными с тем же уровнем шума, что и на тренировочной выборке.

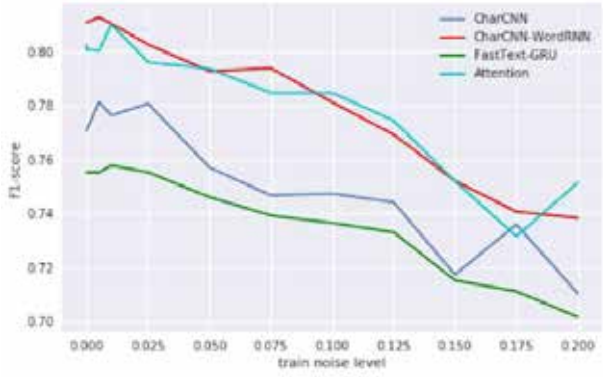

Рис. 8 Набор данных Twitter Sentiment. Набор данных Movie Review. Тренировка на данных с исправленными естественными опечатками и добавленными искусственными, тест на оригинальных данных.

ным. Модели Attention и CharRNN-WordRNN показывают близкие результаты.

На рис. 8 представлены результаты в следующем окружении: обучающая выборка очищается от естественного шума и вводится искусственный; набор данных для тестирования остается неизменным. Как можно заметить, продемонстрированное поведение свойств устойчивости теперь более сложное. Модели CharCNN и Attention показывают лечшие результты с увеличением уровня шума.

Заключение

Мы продемонстрировали добротность распространенных современных архитектур для классификации текста. И, кроме того, предложенный искусственный шум является адекватной заменой естественного шума в данных, что подтверждается тестированием на оригинальных данных при обучении на искусственно зашумленных. Интересно, что некоторые модели работают лучше на сильно зашумленных данных для обучения, что можно объяснить их внутренней адаптацией к предлагаемому шуму. Наиболее добротной моделью во всех экспериментах является комбинация CNN над символами и RNN над словами. Этот факт можно объяснить тем, что CNN нечувствителен к небольшим изменениям ввода, в то время RNN способен запоминать значение всей последовательности на основе векторных представлений из CNN.

Авторы рассматривают будущие улучшения и расширения этой работы в трех основных направлениях: внедрение других типов шума, тестирование более совершенных архитектур для классификации текста и эксперименты с более сложным набором данных, содержащие классификацию с несколькими классами и/или несколькими метками.

Литература

1. *Joulin Armand, Grave Edouard, Bojanowski Piotr and Mikolov Tomas.* 2016. Bag of Tricks for Efficient Text Classification. arXiv preprint arXiv:1607.01759.
2. *Malykh Valentin.* 2018. Robust Word Vectors: Embeddings for Noisy Texts.
3. *Kim Yoon, Jernite Yacine, Sontag David and Rush Alexander M.* 2016. Character-Aware Neural Language Models. In AAAI, pages 2741-2749.
4. *Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts.* 2011. Learning Word Vectors for Sentiment Analysis In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 142-150.
5. *Rubtsova Yuliya.* 2014. Automatic Term Extraction for Sentiment Classification of Dynamically Updated Text Collections into Three Classes In Knowledge Engineering and the Semantic Web, pp140-149, Springer
6. *Bochkarev V.V., Shevlyakova A.V., and Solovyev V.D.* 2015. The average word length dynamics as an indicator of cultural changes in society. Social Evolution & History, 14(2), 153-175.
7. *Cucerzan S. and Brill E*. 2004. Spelling correction as an iterative process that exploits the collective knowledge of web users. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing.
8. *Joulin A., Grave E., Bojanowski P. and Mikolov T.* 2016. Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759.

9. *Howard J. and Ruder S.* 2018. Fine-tuned Language Models for Text Classification. arXiv preprint arXiv:1801.06146.
10. *Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A.N., Kaiser L. and Polosukhin I.* 2017. Attention is all you need. In Advances in Neural Information Processing Systems (pp. 6000-6010).
11. *Xiang Zhang, Junbo Jake Zhao and Yann LeCun.* 2017. Character- level Convolutional Networks for Text Classification. arXiv preprint arXiv:1509.01626
12. *Yoon Kim.* 2014. Convolutional Neural Networks for Sentence Classification. arXiv preprint arXiv:1408.5882
13. *KyungHyun Cho, Bart van Merrienboer, Dzmitry Bahdanau and Yoshua Bengio.* 2014. On the Properties of Neural Machine Translation: Encoder- Decoder Approaches arXiv preprint arXiv:1409.1259
14. *Dzmitry Bahdanau, Kyunghyun Cho and Yoshua Bengio.* 2014. Neural Machine Translation by Jointly Learning to Align and Translate arXiv preprint arXiv:1409.0473
15. *Bengio Y., Simard P. and Frasconi P.* 1994. Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks, 5(2), pp.157-166.
16. *Tutubalina Elena, and Nikolenko Sergey*. 2015. Inferring sentiment- based priors in topic models. In Mexican International Conference on Artificial Intelligence, pp. 92-104.
17. *Niu J., Yang Y., Zhang S., Sun Z. and Zhang W.* 2018. Multi-task Character-Level Attentional Networks for Medical Concept Normalization. Neural Processing Letters, pp.1-18.
18. *Vinciarelli A*. Noisy text categorization, 2005. IEEE Transactions on Pattern Analysis and Machine Intelligence. Dec;27(12):1882-95.
19. *Srivastava N., Hinton G., Krizhevsky A., Sutskever I. and Salakhutdinov R.* 2014. Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15(1), pp.1929-1958.
20. *Pineda F.J.* 1987. Generalization of back-propagation to recurrent neural networks. Physical review letters, 59(19), p.2229.
21. *Glorot X. and Bengio Y.* 2010, March. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the thirteenth international conference on artificial intelligence and statistics (pp. 249256).

Малых Валентин Андреевич. Лаборатория нейронных систем и глубокого обучения Московского физико-технического института, г. Москва, Россия. Исследователь. Количество печатных работ: 12. Область научных интересов: обработка естественного языка. E-mail: valentin.malykh@phystech.edu

Лялин Вячеслав Андреевич. Лаборатория нейронных систем и глубокого обучения Московского физико-технического института, г. Москва, Россия. Младший исследователь. Количество печатных работ: 1. Область научных интересов: обработка естественного языка. E-mail: lyalin@phystech.edu

On Classification of Noisy Texts

V.A. Malykh[1], V.A. Lyalin[1]

[1] Neural Systems and Deep Learning Laboratory, Moscow Institute of Physics and Technology, Moscow, Russia

Abstract. A classic task of text classification was studied in many works, but current approaches mostly devoted to improvement of classification quality for what we call clean corpora, not containing typos. In this work we present results of modern classification models testing in the presence of noise for two languages – English and Russian.

Keywords: *neural networks; text classification; noise robustness.*
DOI: 10.14357/20790279180520

References

1. *Armand Joulin, Edouard Grave, Piotr Bojanowski and Tomas Mikolov.* 2016. Bag of Tricks for Efficient Text Classification. arXiv preprint arXiv:1607.01759.
2. *Valentin Malykh.* 2018. Robust Word Vectors: Embeddings for Noisy Texts.
3. *Yoon Kim, Yacine Jernite, David Sontag and Alexander M*. Rush 2016. Character-Aware Neural Language Models. In AAAI, pages 2741–2749.
4. *Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts.* 2011. Learning Word Vectors for Sentiment Analysis In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 142–150.
5. *Rubtsova Yuliya.* 2014. Automatic Term Extraction for Sentiment Classification of Dynamically Updated Text Collections into Three Classes In Knowledge Engineering and the Semantic Web, pp140-149, Springer
6. *Bochkarev, V. V., Shevlyakova, A. V., and Solovyev, V. D*. 2015. The average word length dynamics as an indicator of cultural changes in society. Social Evolution & History, 14(2), 153-175.
7. *Cucerzan, S. and Brill, E*. 2004. Spelling correction as an iterative process that exploits the collective knowledge of web users. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing.
8. *Joulin, A., Grave, E., Bojanowski, P. and Mikolov, T.* 2016. Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759.
9. *Howard, J. and Ruder, S*. 2018. Fine-tuned Language Models for Text Classification. arXiv preprint arXiv:1801.06146.
10. *Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I*. 2017. Attention is all you need. In Advances in Neural Information Processing Systems (pp. 6000-6010).
11. *Xiang Zhang, Junbo Jake Zhao and Yann LeCun.* 2017. Character-level Convolutional Networks for Text Classification. arXiv preprint arXiv:1509.01626
12. *Yoon Kim*. 2014. Convolutional Neural Networks for Sentence Classification. arXiv preprint arXiv:1408.5882
13. *KyungHyun Cho, Bart van Merrienboer, Dzmitry Bahdanau and Yoshua Bengio*. 2014. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches arXiv preprint arXiv:1409.1259
14. *Dzmitry Bahdanau, Kyunghyun Cho and Yoshua Bengio.* 2014. Neural Machine Translation by Jointly Learning to Align and Translate arXiv preprint arXiv:1409.0473
15. *Bengio, Y., Simard, P. and Frasconi, P*. 1994. Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks, 5(2), pp.157-166.
16. *Tutubalina, Elena, and Nikolenko, Sergey*. 2015. Inferring sentiment-based priors in topic models. In Mexican International Conference on Artificial Intelligence, pp. 92-104.
17. *Niu, J., Yang, Y., Zhang, S., Sun, Z. and Zhang, W.* 2018. Multi-task Character-Level Attentional Networks for Medical Concept Normalization. Neural Processing Letters, pp.1-18.
18. *Vinciarelli A*. Noisy text categorization, 2005. IEEE Transactions on Pattern Analysis and Machine Intelligence. Dec;27(12):1882-95.
19. *Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I.* and Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15(1), pp.1929-1958.
20. *Pineda, F.J.* 1987. Generalization of back-propagation to recurrent neural networks. Physical review letters, 59(19), p.2229.
21. *Glorot, X. and Bengio, Y.* 2010, March. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the thirteenth international conference on artificial intelligence and statistics (pp. 249-256).

V.A. Malykh. Neural Systems and Deep Learning Laboratory, Moscow Institute of Physics and Technology, Moscow, Russia. Research Scientist. Number of publications: 12 papers. Area of scientific interests: natural language processing. E-mail: valentin.malykh@phystech.edu

V.A. Lyalin. Neural Systems and Deep Learning Laboratory, Moscow Institute of Physics and Technology, Moscow, Russia. Junior Research Scientist. Number of publications: 1 papers. Area of scientific interests: natural language processing. E-mail: lyalin@phystech.edu