О.П. МОСАЛОВ, И.А. ИВАНОВ

ГБОУ ВО МО «Технологический университет», г. Королёв Московской области

mosalov.op@ut-mo.ru, ivanov.ia@ut-mo.ru

ИСПОЛЬЗОВАНИЕ АВТОКОДИРОВЩИКА ДЛЯ

ФИЛЬТРАЦИИ В ЗАДАЧЕ ИЗВЛЕЧЕНИЯ КЛЮЧЕВЫХ СЛОВ ИЗ ТЕКСТА

В статье рассматривается задача построение фильтра для оценки качества ключевых слов, извлечённых из текста научной статьи. В качестве такого фильтра используется искусственная нейронная сеть прямого распространения, реализующая архитектуру автокодировщика. Рассматриваются и сравниваются различные варианты внутренней структуры искусственной нейронной сети. Приводится описание вычислительных экспериментов, и обсуждаются их результаты.

**Ключевые слова:** *рекомендательная система, векторные представления слов, извлечение информации из текста, автокодировщик, обучение без учителя.*

Введение

Разработка рекомендательных систем является важным направлением в области искусственного интеллекта [1,2]. Эффективные алгоритмы и современные вычислительные мощности позволяют создавать инструменты, существенно упрощающие решение ряда задач.

Одной из важных предметных областей, которой, тем не менее, уделяется не так много внимания, является образование. Большое количество процессов в данной сфере всё ещё выполняется с помощью ручного труда, автоматизация используется только в наиболее рутинных задачах. Одним из объяснения подобной ситуации является то, что в образовании не всегда возможно введение и применения общих формальных метрик, так как многие результаты, например, выпускные квалификационные работы или научные статьи, написанные обучающимися, могут существенно отличаться друг от друга по форме, содержанию, уровню проработки и тому подобное, но при этом выполнять свою роль – подтверждать наличие самостоятельно проделанной работы.

Довольно известным успешным примеров формальной метрики качества научных статей является «Антиплагиат» [3], система, которая оценивают процент заимствований в заданном тексте.

Авторы предлагают к рассмотрению другую формальную метрику – актуальность текста статьи, рассчитываемую на основании соответствия тем, упоминаемых в тексте, текущим актуальным трендам [4].

В данной статье подробно рассматривается один из логических блоков информационной системы для определения актуальности статей – блок фильтрации ключевых слов, извлечённых из текста статьи.

Общая постановка задачи

Предположим, есть набор текстов научных статей на русском языке $T=\left\{t\_{1},…,t\_{N\_{T}}\right\}$, в каждой из которых автор указал ключевые слова, которые далее по тексту будем назвать «авторские ключевые слова»: $w\_{ij}\in W\_{t\_{i}}$, где $i=1,…,N\_{T}$.

Необходимо построить алгоритм $A$, который на основании произвольного текста $t$, в том числе без авторских ключевых слов, сможет сформировать значение актуальности для этой статьи:

|  |  |
| --- | --- |
| $$A:t\rightarrow a$$ | (1) |

Авторами был применён подход разбиения искомого алгоритма на ряд более простых задач и решения этих задач различными методами: как с помощью детерминированных алгоритмов, так и с помощью инструментов машинного обучения, использующих подходы обучения с учителем и без учителя.

Все подзадачи могут быть разделены на два типа: подготовительные и основные. Подготовительные подзадачи должны быть решены как минимум один раз до начала использования алгоритма $A$ в операционном режиме, однако могут быть решены повторно при необходимости и наличии данных. Основные подзадачи решаются каждый раз при использовании алгоритма $A$.

К подготовительным подзадачам относятся следующие:

1. выбор функции для формирования векторных представлений ключевых слов и фраз:

|  |  |
| --- | --- |
| $P\_{1}:\left\{w\_{ij}\right\}\rightarrow \left\{s\_{ij}\right\}⊂R^{N\_{emb}}$; | (2) |

1. кластеризация векторных представлений авторских ключевых слов для нахождения центроидов кластеров, которые интерпретируются как темы в выбранной предметной области:

|  |  |
| --- | --- |
| $P\_{2}:\left\{s\_{ij}\right\}\rightarrow \left\{C\_{k}\right\}$, | (3) |

где $k=1,…,N\_{C}$;

1. определение текстовой метки для каждого центроида кластера:

|  |  |
| --- | --- |
| $P\_{3}:C\_{i}\rightarrow L\_{i}$, | (4) |

где $i=1,…,N\_{C}$;

1. определение актуальности для каждой метки кластера:

|  |  |
| --- | --- |
| $P\_{4}:L\_{i}\rightarrow R\_{i}$, | (5) |

где $i=1,…,N\_{C}$.

К основным подзадачам относятся:

1. генерирования из произвольной заданной русскоязычной статьи $t$ кандидатов в ключевые слова и фразы:

|  |  |
| --- | --- |
| $T\_{1}:t\rightarrow \{v\_{i}\}$; | (6) |

1. фильтрация кандидатов в ключевые слова и фразы и формирование сгенерированных ключевых слов и фраз:

|  |  |
| --- | --- |
| $T\_{2}: \{v\_{i}\}\rightarrow \left\{w\_{j}\right\}$, | (7) |

где $\left\{w\_{j}\right\}⊂\left\{v\_{i}\right\}$;

1. определение актуальности каждого сгенерированного слова и фразы на основании расстояний между ними и центроидами кластеров в пространстве векторных представлений, и актуальностей кластеров:

|  |  |
| --- | --- |
| $T\_{3}:w\_{i},ρ\left(w\_{i}, C\_{j}\right),R\_{j}\rightarrow a\_{i}$; | (8) |

1. определение итоговой актуальности статьи на основании актуальностей сгенерированных из неё ключевых слов и фраз:

|  |  |
| --- | --- |
| $T\_{4}:a\_{i}\rightarrow a$. | (9) |

В данной работе основное внимание уделено подзадаче $T\_{2}$, однако ниже в данном разделе приводится базовая информация по решению остальных подзадач.

Для решения задачи $P\_{1}$ использованы готовые векторные представления, реализованные с помощью библиотеки navec [5].

Задача $P\_{2}$ решалась двумя способами: простым алгоритмом *k* средних [6] и более сложным алгоритмом нечёткой кластеризации [7]. На данный момент эксперименты для окончательного выбора подхода ещё не завершены.

При решении задачи $P\_{3}$ использовалась ручная разметка.

Для решения задачи $P\_{4}$ применён поисковый API, который по строке возвращает количество её упоминаний в сети Интернет.

В качестве инструмента для решения подзадачи $T\_{1}$ используется библиотека YAKE [8], функционал которой рассчитывает значение важности для каждого набора последовательных слов, длиной не более заранее заданного числа, и возвращает такие наборы с наибольшими значениями важности. На величину важности влияют следующие критерии:

1. отсутствие в слове или фразе заглавных букв, кроме случая начала предложения;
2. как часто слово или фраза встречается ближе к началу текста;
3. как часто слово или фраза встречается в тексте;
4. количество разных слов, встречающихся в окрестности рассматриваемого слова или фразы;
5. количество различных предложений, содержащих рассматриваемое слово или фразу.

Для решения подзадачи $T\_{3}$ применяется аналитическая формула, имеющая следующий вид:

|  |  |
| --- | --- |
| $$a\_{i}=\frac{1}{N\_{C}}\sum\_{j=1}^{N\_{C}}\left(\hat{R}\_{j}\left(1-\hat{d}\_{ij}\right)\right)$$ | (10) |

|  |  |
| --- | --- |
| $$\hat{R}\_{j}=\frac{e^{R\_{j}}}{\sum\_{i}^{}e^{R\_{i}}}$$ | (11) |

|  |  |
| --- | --- |
| $$\hat{d}\_{ij}=\frac{ρ\left(w\_{i},C\_{j}\right)}{\max\_{m,n}ρ\left(w\_{m},C\_{n}\right)}$$ | (12) |

Подзадача $T\_{4}$ решается с помощью простой формулы:

|  |  |
| --- | --- |
| $$a=\sum\_{i=1}^{\left‖\left\{v\right\}\right‖}γ^{i-1}a\_{i}$$ | (13) |

Подробное описание алгоритма

Подзадача $T\_{2}$: фильтрация кандидатов в ключевые слова и фразы и формирование сгенерированных ключевых слов и фраз – может быть сведена к задаче бинарной классификации, то есть к построению функции $f\left(v\_{i}\right)$, на основании значения которой может быть принято решение о квалификации или дисквалификации рассматриваемого кандидата в ключевые слова $v\_{i}$.

Однако для решения такой задачи необходим достаточно большой набор данных размеченных данных, которым авторы не располагали – в нашем распоряжении было около 300 научных статей.

Другой проблемой является то, что даже разметив вручную большой набор текстов, возможно получить только объекты одного класса, что приведёт к сильно несбалансированному набору данных.

Приняв во внимание описанные проблемы, авторы решили применить для решения данной подзадачи искусственную нейронную сеть с архитектурой автокодировщика [9]. Такой подход позволил минимизировать влияние проблем и достичь приемлемого результата.

Архитектура автокодировщика для искусственной нейронной сети означает выполнение двух условий:

1. структура искусственной нейронной сети должны быть симметричной, то есть, если $i$‑ый слой содержит $n\_{i}$ нейронов и общее число слоёв равно $N$, то $n\_{i}=n\_{N-i+1}$,
2. число нейронов уменьшается от входного и выходного слоёв к центру, то есть $n\_{i}>n\_{j}$ при $i<j<0,5N$.

Такую архитектуру можно использовать без размеченных данных, обучая искусственную нейронную сеть так, чтобы выходной вектор, то есть вектор значений на выходном слое, был как можно ближе к входному вектору, то есть к вектору значений на входном слое. Для вычисления близости векторов можно использовать любую функцию расстояния.

Из-за того, что количество нейронов в скрытых слоях меньше количества нейронов в входном и выходном слоях, автокодировщик не может просто запомнить набор данных, на котором он обучается. Искусственная нейронная сеть ищет зависимости в данных и строит своего рода фильтр, позволяющий выделять векторы, использованные при её обучении, от случайным образом выбранных векторов из того же пространства.

Таким образом, если в пространстве $R^{N}$ есть некое подмножество $M$ и из этого подмножества выделено его подмножество $\tilde{M}$ такое, что распределение элементов $\tilde{M}$ приблизительно совпадает с распределением элементов $M$, то, обучив автокодировщик $X$ на множестве элементов $\tilde{M}$, можно рассчитывать, что ошибка работы $X$ на $M$ не будет превышать некоторого порогового значения.

Используя описанное выше предположение, авторы выделили из имеющихся статей около 900 авторских ключевых слов, рассчитали векторные представления этих ключевых слов и обучили автокодировщик на получившемся наборе данных.

Обученный автокодировщик далее использовался как решение подзадачи $T\_{2}$: для каждого кандидата в сгенерированный ключевые слова вычислялось векторное представление, оно подавалось на вход автокодировщика и вычислялась ошибка работы автокодировщика, в данном случае – евклидово расстояние между входным и выходным векторами. Если получившийся результат был ниже установленного порогового значения, то кандидат считался сгенерированным ключевым словом, иначе – отбрасывался.

Вычислительные эксперименты

При вычислении векторных представлений ключевых слов, как авторских, так и кандидатов с генерированные, получались вектора длинной 300. Поэтому размеры входного и выходного слоёв используемой искусственной нейронной сети также были равны 300.

Был проведён ряд вычислительных экспериментов для поиска эффективной внутренней структуры автокодировщика. Во всех этих экспериментах использовался один скрытый слой. Размер этого слоя менялся от 1 до 299 нейронов.

В качестве активационной функции нейронов скрытого слоя использовался линейный выпрямитель (ReLU) [10]. На выходном слое активационная функция была линейной.

В качестве оптимизируемой функции ошибки, как уже было отмечено выше, использовалось евклидово расстояние между входным и выходным векторами, то есть среднеквадратичная ошибка. Как алгоритм оптимизации использовался Adam [11].

При обучении искусственной нейронной сети использовались следующие параметры: количество эпох обучения было равно 40, размер батча был равен 16.

Для того, чтобы оценивать качество обученных автокодировщиков и сравнивать их между собой, из имеющегося набора данных было выделено 20% в качестве тестовой выборки, и метрикой каждой обученной модели считалось значение среднеквадратичной ошибки на этой тестовой выборке.

На рис. 1 приведены графики изменения величины ошибки в зависимости от эпохи обучения для скрытого слоя размеров в 100 нейронов. Сплошная линия – это ошибка на обучающей выборке, пунктирная линия – ошибка на тестовой выборке.



Рис. 1. Ошибки на обучающей и тестовой выборках в зависимости от номера эпохи обучения для 100 нейронов в скрытом слое.

Видно, что предложенная модель справляется с поставленной задачей, то есть ошибки и на обучающей, и на тестовой выборках уменьшаются, однако необходимо определить критерий для выбора количества нейронов в скрытом слое автокодировщика, который в итоге будет использован как блок рекомендательной системы.

Для этого на одном и тот же разбиении на обучающую и тестовую выборки было проведено обучение для всех возможных значений размера скрытого слоя (от 1 до 299) и обнаружены зависимости итоговых ошибок, то есть ошибок после 40 эпох обучения, для обеих выборок. Полученный результат приведёт на рис. 2. Верхняя кривая соответствует ошибке на тестовой выборке, нижняя кривая – ошибке на обучающей выборке.



Рис. 2. Величины итоговой ошибки в зависимости от размера скрытого слоя.

Действительно, с ростом числа нейронов в скрытом слое эффективность обучения повышается, то есть абсолютные значения ошибок уменьшаются, однако при больших размерах скрытого слоя наступает явление переобучения: искусственная нейронная сеть склонна запоминать элементы обучающей выборки вместо того, чтобы выявлять в них зависимости и шаблоны.

При переобучении искусственная нейронная сеть работает с новыми, ранее неизвестными, данными существенно хуже, чем с данными, входящими в обучающую выборку. Для того, чтобы оценить размер эффекта переобучения рассмотрена зависимость отношения ошибки на тестовой выборке к ошибке на обучающей выборке в зависимости от количества нейронов в скрытом слое. В идеальном случае эта величина должна быть близка к единице, а её рост означает, что модель работает с новыми данными хуже и хуже. Результат вычислительных экспериментов представлен на рис. 3.



Рис. 3. Зависимость отношения ошибок от количества нейронов в скрытом слое.

Таким образом, увеличение размера скрытого слоя автокодировщика приводит к усилению эффекта переобучения.

Для того, чтобы получить достаточно небольшую величину ошибки по итогам обучения и при этом избежать слишком большого переобучения, в итоговом автокодировщике было использовано среднее значение количества нейронов в скрытом слое – 100, таким образом в целом структура автокодировщика имеет вид «300-100-300» и является достаточной для получения необходимого результата. Автокодировщик именно такой структуры был обучен и используется в качестве решения задачи фильтрации кандидатов в сгенерированные ключевые слова.

Заключение

В данной работе предложено использование искусственной нейронной сети с архитектурой автокодировщика для решения задачи бинарной классификации при наличии данных только одного класса. На примере конкретной задачи – разработки блока рекомендательной системы – показана реализация предложенного подхода.

Описанный подход на данный момент реализован в виде программного кода и является частью информационной системы, развёрнутой в ГБОУ ВО МО «Технологический университет» и находящейся в процессе опытной эксплуатации.

*Список литературы*

1. Fayyaz Z., Ebrahimian M., Nawara D., Ibrahim A., Kashef R. Recommendation Systems: Algorithms, Challenges, Metrics, and Business Opportunities // Applied Sciences. 2020. Vol. 10. P. 7748.
2. Zhang Q., Lu J., Jin Y. Artificial intelligence in recommender systems // Complex & Intelligent Systems. 2021. Vol. 7. P. 439‑457.
3. Программная система для обнаружения текстовых заимствований в учебных и научных работах «Антиплагиат.ВУЗ» [Электронный ресурс] // URL: https://www.antiplagiat.ru (дата обращения: 01.03.2021).
4. Мосалов О.П. Определение актуальности публикаций с использованием методов машинного обучения // Большие данные в образовании: анализ данных как основание принятия управленческих решений. Сборник научных статей I Международной конференции. 15 октября 2020 г., Москва / под общ. ред. О.А. Фиофановой. – М. : Издательский дом «Дело» РАНХиГС. 2020. С. 172‑182.
5. Compact high quality word embeddings for Russian language [Электронный ресурс] // URL: https://github.com/natasha/navec (дата обращения: 20.04.2021).
6. MacQueen J. Some methods for classification and analysis of multivariate observations // Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, University of California Press. 1967. Vol. 1. P. 281‑297.
7. Bezdek J.C., Ehrlich R., Full W. FCM: The fuzzy c-means clustering algorithm // Computers & Geosciences. 1984. Vol. 10. No. 2‑3. P. 191‑203.
8. Campos R., Mangaravite V., Pasquali A., Jatowt A., Jorge A., Nunes C., Jatowt A. YAKE! Keyword Extraction from Single Documents using Multiple Local Features // Information Sciences Journal. Elsevier. 2020. Vol. 509. P. 257‑289.
9. Goodfellow I., Bengio Y., Courville A. Deep Learning // MIT Press. 2016.
10. Nair V., Hinton G.E. Rectified linear units improve restricted boltzmann machines // Proceedings of the 27th International Conference on International Conference on Machine Learning (ICML'10). Omnipress, Madison, WI, USA. 2010. P. 807‑814.
11. Diederik K., Ba J.L. ADAM: a method for stochastic optimization // AIP Conference Proceedings 1631. 2014. P. 58‑62.