Я.В. Соловьева, А.С. Некрасова

ИССЛЕДОВАНИЕ МЕТОДОВ КЛАССИФИКАЦИИ ТЕКСТОВ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ

(Самарский университет)

Классификация текстов является одной из основных задач компьютерной лингвистики, поскольку к ней сводится ряд других задач: определение тематической принадлежности текстов, автора текста, эмоциональной окраски высказывания и др.

Естественный язык – язык, используемый для общения людей и не созданный целенаправленно. Примерами естественных языков являются русский, английский, китайский, казахский и др.

Отсутствие возможности получить наиболее актуальную и полную информацию по конкретной теме делает бесполезной большую часть накопленных ресурсов. Поскольку исследование конкретной задачи требует все больших трудозатрат на непосредственный поиск и анализ информации по теме, многие решения принимаются на основе неполного представления о проблеме. Таким образом, встает проблема построения классификатора текстов, позволяющего сократить трудозатраты на поиск нужной информации, представленной электронными текстами.

Классификация текстов — задача компьютерной лингвистики, заключающаяся в отнесении документа к одной из нескольких категорий на основании содержания документ. Для каждой категории отбираются текстовые массивы, которые используются системой классификации в режиме обучения. После завершения обучения система с помощью специальных алгоритмов должна распределять входные потоки текстовой информации по классам [1]. Различные решения данной задачи находят свое практическое применение в таких областях, как составление тематических каталогов, фильтрация спама, классификация сайтов по тематическим каталогам, обработка документооборота и т.д.

Перспективным направлением на сегодняшний день также считается использование метода опорных векторов в качестве основы подобного рода классификатора. Основным преимуществом данного метода является возможность выявления зависимостей, не поддающихся обнаружению при использовании других подходов обработки информации. Методы опорных векторов в анализе текстовой информации обладают достаточным быстродействием, не зависят от языка предметной области и дают хорошие результаты при обработке текстов.

Наиболее популярными моделями и техниками, основанными на вышеописанном методе классификации, являются Word2Vec, Lime, GloVe и TensorFlow. Метод опорных векторов в совокупности с методом на основе деревьев решений даёт возможность эффективно разделять входные объекты на классы, при этом расширяя возможности классификации до способности разбиения на несколько различных классов.

Метод деревьев решений для задачи классификации состоит в том, чтобы осуществлять процесс деления исходных данные на группы, пока не будут получены однородные их множества. Совокупность правил, которые дают такое разбиение, позволяют затем делать прогноз (т.е. определять наиболее вероятный номер класса) для новых данных. Метод деревьев решений применим для решения задач классификации, возникающих в самых разных областях, и считается одним из самых эффективных [2]. В листьях разрешающего дерева размещаются значения целевой функции, в прочих узлах – условия перехода, определяющие направление движения вдоль ребер дерева. Для классификации каждого примера алгоритму необходимо пройти все дерево от корня до одного из листов. Тем самым получить значение целевой функции $\Phi : D \times C \to \{0.1\}$. Для построения дерева на каждом внутреннем узле необходимо найти такое условие (проверку), которое бы разбивало множество, ассоциированное с этим узлом на подмножества [3]. Разбив множество примеров на основе значений некоторого признака X на подмножества $S_1,S_2,\ldots,S_n$, мы можем вычислить $Info(S)$ как взвешенное среднее информации, необходимой для установления принадлежности примера определенному классу в каждом подмножестве:

$$Info(X,T) = \sum_{i=1}^{n} \frac{|S_i|}{|S|} * Info(S_i). \quad (1)$$

Величина

$$Gain(X, S) = Info(S) - Info(X, S) \quad (2)$$

показывает количество информации, которое мы получаем, благодаря признаку X. Данная величина используется как критерий оценки информативности признака при построении решающих деревьев. Это позволяет получать деревья минимального размера.

Метод опорных векторов решает задачу классификации при $Y=\{-1,+1\}$, т.е. предполагается, что все объекты исходного множества принадлежат одному из классов. Если классов больше, появляется задача мульти-классификации. Два наиболее популярных метода её решения для SVM это «Один-против-всех» и «Каждый-против-каждого». Метод один-против-всех состоит в обучении одной SVM на каждый из классов. Каждая такая SVM способна отличать объекты своего класса от остальных. Для классификации произвольного объекта нужно выбрать SVM с максимальным результатом [4]. Метод каждый-против-каждого состоит в обучении одной SVM на каждую пару классов (если всего классов n – обучаем $n(n-1)/2$ SVM). Каждая такая SVM способна отличать объекты одного класса от объектов другого. Для классификации произвольного объекта все SVM голосуют за один из классов и, затем, выбирается класс с наибольшим числом голосов.

Наиболее распространённая метрика оценки качества классификации включает в себя оценку двух характеристик классификатора – точность и полноту. Точность системы в пределах класса – это доля документов, действительно принадлежащих данному классу относительно всех документов, которые си-

стема отнесла к этому классу. Полнота системы – это доля найденных классификатором документов, принадлежащих классу относительно всех документов этого класса в тестовой выборке. На основе вышеописанных величин производится подсчет метрик, которые оценивают качество работы классификатора и позволяют произвести сравнение классификаторов между собой.

Полная точность или аккуратность: $accuracy = \frac{TP + TN}{TP + TN + FP + FP}$,

Точность: $precision = \frac{TP}{TP + FP}$ или $precision = \frac{TN}{TN + FN}$

Полнота: $recall = \frac{TP}{TP + FN}$ $recall = \frac{TN}{FP + TN}$

F-мера: $F = \frac{2 * precision * recall}{precision + recall}$. Данная мера объединяет оценки точности и полноты в одну.

Для более эффективной обработки входных данных их производится предобработка методом векторизации TF*IDF. Этот метод не выбрасывает часто употребляемые слова из словаря, но уменьшает их вес в вектор-признаке. Для этого для всех слов словаря вычисляется коэффициент обратной частоты. В модели представления *TF-IDF* каждому терму *t* документа *k* ставится в соответствие величина:

$$TF * IDF = TF(k,t) \cdot \log_2\left(\frac{N}{kf(t)}\right), \quad (3)$$

где *tf(k,t)* – это частота терма *t* в документе *k*, *N* – число документов в корпусе, *kf(t)* – количество документов в которых встречается терм *t*.

В сложных естественных языках одно и то же слово может принимать разные формы (падежи), и в словарь частотного анализа могут попадать все словоформы, отличающиеся предлогами и окончаниями, что может повлиять на увеличение словаря и размера набора данных для обучения. Увеличение словаря и размера набора данных для обучения в свою очередь могут вызывать падение производительности системы и ухудшить обобщающие способности классификатора, то есть привести к переобучению. Существует несколько способов решения этой задачи: лематизация и стеминг. Лематизация – все слова в тексте приводятся к нормальной форме (единственное число, именительного падежа). Стеминг – выделение основы слов путём отбрасывания приставок и окончаний. Этот способ нормализации текста работает гораздо быстрее, чем лематизация. Он менее качественный, но для частотного анализа его вполне достаточно.

Для проведения исследования классификационных возможностей метода опорных векторов и деревьев решений были использованы текстовые данные, сформированные из русскоязычных новостных сайтов Интернета.

В таблице 4.1 показана зависимость точности классификации от способа предварительной обработки исходных тестовых данных. Первоначально сравнивались два метода – частотное представление текста и метод *TF*IDF*. Метод

*TF*IDF* показал лучшие результаты, поэтому дальнейшие исследования были проведены, взяв за основу данный метод. В данном исследовании для алгоритмов деревьев решения максимальная глубина дерева была равна 400, а для метода опорных векторов было выбрано линейное ядро, критерий остановки был равен 0,001, предельный параметр ошибки равен 1.

Таблица 4.1 – зависимость точности классификации от модели предобработки исходных данных

Модель предобработки входных данных	Деревья решений		Метод опорных векторов
	ID3	C4.5	
Частотный анализ	60%	63%	74%
*TF*IDF*	68%	71%	79%
*TF*IDF*+стоп-слова	72%	75%	84%
*TF*IDF*+стоп-слова+стеммер	79%	84%	93%
*TF*IDF*+стоп-слова+лемматизация	76%	80%	91%

Из таблицы 4.1 видно, что лучшие результаты классификации при использовании и метода опорных векторов, и деревьев решений достигаются при предварительной обработке входных текстовых данных методом *TF*IDF* при использовании стеммера Портера с удалением стоп-слов.

В таблице 4.2 показаны результаты исследования зависимости точности классификации методом деревьев решений от предельного параметра ошибки C. Из полученных результатов можно сделать вывод, что оптимальным значением предельного параметра ошибки является 1.

Таблица 4.2 – зависимость точности классификации методом опорных векторов от предельного параметра ошибки.

Параметр ошибки	Точность классификации, %
0,1	71
1	94
10	91
100	91

Из данной таблицы видно, что большинство текстов были верно распределены в соответствии с их тематиками.

Полученные результаты позволяют сделать вывод, что метод опорных векторов показал лучшие результаты, чем метод деревьев решений.

Также по результатам исследований было установлено, что один и тот же уровень точности классификации достигается за счет соблюдения баланса между величиной обучающих текстов и их количеством. Чем меньше по размеру

обучающие тексты, тем больше их должно быть в обучающем множестве, соответственно, чем фрагменты больше – тем меньшим их количеством можно обойтись.

Литература

1. Леонтьева, Н. Н. Автоматическое понимание текстов: системы, модели, ресурсы [Текст]: учеб. пособие для вузов/ Н. Н. Леонтьева – М.: Издательский центр «Академия», 2006. – 304 с.
2. Machine Learning in Automated Text Categorization [Электронный ресурс] Автоматическая классификация текстов — http://www.math.unipd.it/~fabseb60/Publications/ACMCS02.pdf (дата обращения 14.06.2022).
3. Sebastiani, F.: Machine learning in automated text categorization, ACM Computing Surveys Computing Surveys, vol. 34, pp. 1–47, 2002.
4. Деревья решений – CART математический аппарат [Электронный ресурс]. – https://basegroup.ru/community/articles/math-cart-part1 (дата обращения 15.03.2022).

А.А. Столбова, А.А. Малышев

РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ ВЫЯВЛЕНИЯ ФАЛЬСИФИЦИРОВАННЫХ ФРАГМЕНТОВ НА ФОТОИЗОБРАЖЕНИЯХ

(Самарский национальный исследовательский университет
имени академика С.П. Королёва)

В данной работе предлагается разработанная автоматизированная система выявления фальсифицированных фрагментов на фотоизображениях. Система обеспечивает многоэтапную обработку изображений с целью нахождения на них фрагментов, выполненных после создания изображений, в том числе, средствами современных нейронных сетей [1]. Данное решение развивает тему, затронутую в [2], в части развития интеллектуальных возможностей анализа изображений и локализации конкретного фрагмента, подвергшегося редактированию.

Поскольку метаданные цифровых фотографий (EXIF) содержат в себе достаточно исчерпывающую информацию о времени, месте, и устройстве, на которое они были сделаны [3], а сами фотографии сохраняются форматах со сжатием с потерями, разрабатываемая автоматизированная система выявления фальсифицированных фрагментов предполагает к реализации три этапа анализа изображения:

1. Анализ метаданных изображения;
2. Анализ изображения с помощью ELA;
3. Нейросетевой анализ изображения.

На первом этапе система анализирует следующие EXIF-метаданные: