УДК 004.853

Л.Э. Чалая [1], С.Г. Удовенко [2], Е.В. Кушвид [1]

[1] ХНУРЭ, г. Харьков, Украина, larysa.chala@nure.ua;;

[2] ХНЭУ, г. Харьков, Украина, serhii.udovenko@nure.ua;

[1] ХНУРЭ, г. Харьков, Украина, eugenii.kushvid@nure.ua

МЕТОД ДВУХЭТАПНОЙ КЛАССИФИКАЦИИ ЭЛЕКТРОННЫХ ТЕКСТОВ

Предложен метод классификации электронных, основанный на комбинированном применении полиномиальной модели классификатора Байеса, лингвистических дескрипторов и модифицированного метода Гинзбурга. Метод содержит два этапа: этап предварительной фильтрации псевдоспама и этап классификации документов основного массива . Результаты тестирования подтверждают эффективность применения метода для обработки и классификации документов в больших политематических текстовых массивах .

КЛАССИФИКАЦИЯ ТЕКСТОВ, КЛАССИФИКАТОР БАЙЕСА, ЛИНГВИСТИЧЕСКИЙ ДЕСКРИПТОР, ПРОГРАММНЫЙ МОДУЛЬ

Введение

В области автоматической обработки электронных текстов сложился ряд относительно самостоятельных направлений: извлечение объектов и признаков, реферирование, классификация, кластеризация, интеллектуальный поиск, семантический анализ и т.п. [1, 2]. В частности, задача организации эффективного доступа к неструктурированной тематической информации непосредственно связана с задачей классификации электронных текстов, извлекаемых из ресурсов сети Интернет или электронных библиотек. Для решения последней разработано множество эффективных методов, некоторые из которых характеризуются качеством классификации, сравнимым с результатами классификации, выполняемой квалифицированными экспертами [3]. К наиболее используемым методам относятся, в частности, метод опорных векторов, метод логистической линейной регрессии, метод k-ближайших соседей, нейронные классифицирующие сети и т.п. В то же время следует отметить, что данные методы характеризуются высокой сложностью разработки и реализации классифицирующих алгоритмов, высокую вычислительной сложностью, сложностью коррекции функций обучения.

В последнее время получили распространение алгоритмы классификации объектов, лишенные подобных недостатков, например, наивный классификатор Байеса (НКБ) [4, 5]. Этот классификатор обладает простой и реализуемой в реальном времени обучающей процедурой, которую легко модифицировать под особенности конкретной решаемой задачи. Однако базовый классификатор Байеса основан на предположении об условной независимости переменных, что во многих случаях может оказаться неприемлемым, так как гипотеза полной независимости результата классификации от сочетания признаков оказывает существенное влияние на качество классифицирующей процедуры во многих реальных задачах обработки электронных текстов (в частности, текстов научно-технического содержания). Когда же допущение о независимости выполняется, НКБ, как правило, превосходит другие алгоритмы классификации (в том числе, и многоклассовой) и при этом использует меньший объем обучающих данных.

Одним из подходов, позволяющим учесть совокупность классификационных признаков в анализируемых текстах, является нечеткая классификация [6]. При такой классификации фрагменты данных могут принадлежать нескольким классам, связанным с каждым элементом набором степеней принадлежности. Они указывают силу ассоциации между элементом данных и определенным классом. Нечеткая классификация - процесс определения этих степеней принадлежности и использования их, чтобы присвоить элементы данных двум или более классам. В реальных случаях не может быть никаких резких границ между классами, и тогда применение методов нечеткой логики может оказаться перспективным направлением повышения эффективности процедур автоматической классификации текстов. Однако практическая реализация нечетких классификаторов предполагает необходимость использования ряда эвристических назначений (в частности, задания функций принадлежности и интервалов дискретизации значений лингвистических переменных), что зачастую не дает возможности гарантировать точность.

Другим направлением повышения эффективности решения задач, связанных с автоматической классификацией политематических текстовых ресурсов, является гибридное применение байесовской классификации и методов, связанных с возможностью учета в схеме классификатора операций выделения из текстов дескрипторов и ключевых слов. В общем случае, при необходимости классификации большого объема разнородных классов с целью получения конечного множества массивов, содержащих документы заданной тематики, целесообразно разработать простую в вычислительном отношении процедуру построения соответствующего классификатора.

Целью настоящей работы является разработка и тестирование метода двухэтапной классификации политематических электронных текстов, основанной на ком-

бинированном применении НКБ и модифицированной процедуры байесовской классификации с предварительной фильтрацией исходного массива текстовых документов и выделением лингвистических дескрипторов.

1. Структура и описание предлагаемого метода классификации электронных текстов

Рассмотрим задачу классификации по тематическим рубрикам большого массива документов, имеющего политематический и составной характер. Таким массивом может быть, например, электронный сборник материалов конференции, содержащих аннотации и основной текст; массив документов большой электронной библиотеки, посвященных некоторому общему научному направлению, и т.п. Проблемами, возникающими при реализации такой классификации, являются: наличие служебных элементов и посторонних блоков текста, не относящихся к основной тематике документа; наличие в обучающем массиве аномальных документов (пустых, в неизвестных кодировках и т.п.); сложность автоматического формирования решающих правил для рубрик из-за негативного влияния посторонней информации; снижение качества классификации из-за наложения нескольких рубрик друг на друга; сложность интерпретации результатов классификации из-за неопределенности расположения в тексте информации, релевантной рубрике, и т.п.

Предлагаемая схема решения этой задачи приведена на рис. 1.

Рис.1. Схема двухэтапного метода классификации

Схема предполагает последовательную реализацию процедур двухэтапной обработки исходного текстового массива документов М1.

Предполагается, что потенциальный пользователь определил рубрики, в соответствии с которыми должен быть сформирован классифицированный массив документов, содержащий N классов.

Первый этап метода содержит процедуры фильтрации исходного массива М1 с целью удаления документов, относящихся к *псевдоспаму*. Под *псевдоспамом* будем понимать все документы массива, не представляющие тематический интерес для потенциального пользователя, вероятность отнесения которых к одному из N классов результирующего массива очень низка.

Выделение псевдоспама будем осуществлять с использованием минимального словаря дескрипторов, формируемых по тематической направленности рубрик, интересующих потенциального пользователя. Дескриптор – лексическая единица (слово, словосочетание, аббревиатура), служащая для описания основного смыслового содержания документа или формулировки запроса при поиске документа в информационно-поисковой или классифицирующей системе [7]. Дескриптор однозначно ставится в соответствие группе ключевых слов естественного языка, отобранных из текста, относящегося к определенной области знаний. Это позволяет создать уникальные словари дескрипторов для учета их в классификаторе с целью повышения точности классификации. Процедура выделения псевдоспама может быть основана на применении классического подхода, использующего машинное обучение с учителем [5]. В этом подходе требуется наличие обучающей коллекции содержащих дескрипторные термины текстов, на базе которой строится статистический или вероятностный классификатор.

На первом этапе предлагаемого метода для обработки исходного массива документов с целью выделения псевдоспама используется полиномиальная модель метода Naive Bayes (NB), на основании которой реализуется наивный бинарный классификатор Байеса NB1.

При этом для характеристики анализируемых текстов используется векторная модель представления документов d исходного массива М1:

$$d = (t_1, t_2, ..., t_n), \qquad (1)$$

где $t_1, t_2, ..., t_n$ – дескрипторные термины (признаки) текстов; n – общее количество учитываемых терминов. Компонентам вектора (1) могут быть поставлены в соответствие бинарная или частотная функции взвешивания. Бинарные векторы представляются как последовательность нулей и единиц: если конкретный термин из словаря выборки встречается в тексте – вес термина будет равен 1, в противном случае – 0. Частотные векторы формируются на основе количества вхождений определенного термина в классе документов.

Для классификатора псевдоспама NB1 выберем бинарную функцию взвешивания, полагая, что при реше-

нии задачи фильтрации псевдоспама наличие термина в документе важнее, чем его частота.

Полиномиальная модель классификатора NB1, как и базовая модель байесовской классификации, базируется на понятии условной вероятности принадлежности документа d классу c. В соответствии с теоремой Байеса, такая вероятность определяется следующим образом:

$$P(c|d) = \frac{P(c) \cdot P(d|c)}{P(d)}. \quad (2)$$

Наиболее вероятным классом (по оценке апостериорного максимума), к которому принадлежит документ d, в классификации по методу NB1 является тот, при котором условная вероятность принадлежности документа d классу c максимальна:

$$c_{map} = \arg\max_c P(t_1, t_2, ..., t_n|c) * P(c). \quad (3)$$

Отметим, что в (3) не учитывается знаменатель из отношения (2), так как для одного и того же документа d вероятность $P(d)$ будет одинаковой.

Предположим, что позиция термина в предложении документа не важна. Тогда условную вероятность для признаков $t_1, t_2, ..., t_n$ можно представить следующим образом:

$$P(t_1|c) * P(t_2|c) * ... * P(t_n|c) = \prod_i P(t_i|c). \quad (4)$$

Для нахождения наиболее вероятного класса для документа с помощью классификатора NB1 необходимо определить условные вероятности принадлежности документа d для каждого из представленных классов (в задаче классификации псевдоспама рассматриваются два возможных класса: массив M3 (ham) документов, отбираемых для дальнейшего анализа, и массив M2 (spam) документов псевдоспама) отдельно и выбрать класс, имеющий максимальную вероятность:

$$c_{map} = \arg\max_c \left[P(c) * \prod_i P(t_i|c) \right]. \quad (5)$$

В уравнении (5) умножается множество условных вероятностей, что может привести к потере значимости при выполнении операций с плавающей точкой для относительно больших текстов. Поэтому целесообразно применять логарифмическое представление уравнения (5):

$$c_{map} = \arg\max_c \left[\log P(c) + \sum_i \log P(t_i|c) \right]. \quad (6)$$

Вероятности классов $P(c)$ в (5) и (6) оцениваются как отношение количества документов класса в обучающей выборке к общему количеству документов в выборке:

$$P(c) = \frac{N_c}{N}. \quad (7)$$

Условные вероятности для признаков (терминов) t_i оцениваются как отношение количества терминов t_i в классе к общему количеству терминов в этом классе:

$$P(t_i|c) = \frac{count(t_i|c)}{\sum_t count(t,c)}, \quad (8)$$

где $t \in V$; V – словарь обучающей выборки.

Здесь принимается предположение о том, что позиция слова в тексте не влияет на вероятность его появления. В этом случае условные вероятности слова, занимающего две разные позиции $k1$ и $k2$ в разных местах документа, будет одинаковой:

$$P(t_{k_1}|c) = P(t_{k_2}|c). \quad (9).$$

При вычислении вероятностей возможна ситуация, когда какое-либо слово из текста для классификации ни разу не присутствовало в обучающей выборке какого-либо класса, тогда вероятность данного слова в классе и полная вероятность соответствия документа данному классу будут равны нулю. Для устранения таких выбросов используем аддитивное сглаживание Лапласа (Laplace smoothing). Принцип такого сглаживания состоит в том, что к частотам появления всех терминов из словаря искусственно добавляется единица. При этом термины, которые не присутствовали в документах обучающей выборки, получают незначительную, но отличную от нуля вероятность появления и дают возможность отнести документ к одному из формируемых классов. Приэтом формула (8) трансформируется следующим образом:

$$P(t_i|c) = \frac{count(t_i|c)+1}{\sum_t (count(t,c)+1)} = \frac{count(t_i|c)+1}{(\sum_t count(t;c)) + |v|}, \quad (9)$$

где в знаменатель правой части добавляется количество слов в словаре обучающей выборки V.

Альтернативой полиномиальной модели классификатора NB1 может быть многофакторная модель или модель Бернулли метода NB. Модель Бернулли оценивает условные вероятности для признаков (терминов) t_i как часть документов класса c, которые содержат термин t_i по отношению ко всем документам класса c. При классификации тестового документа модель Бернулли игнорирует количество вхождений слова в документ, в то время как полиномиальная модель отслеживает все вхождения терма. В результате этого модель Бернулли обычно делает много ошибок при классификации длинных документов. Условная вероятность для признаков (терминов) t_i в методе Бернулли рассчитывается следующим образом:

$$P(t_i|c) = \frac{N_{ct_i}+1}{N_c+2}. \quad (10)$$

Для оценки эффективности классификатора NB1 используем простую метрику эффективности. Пусть в результате классификации документов тестовой выборки, к классу M3 правильно отнесены T3 документов, неправильно – F3, а к классу M2 правильно были отнесены T2 документов, неправильно – F2.

Тогда точность классификации на первом этапе с помощью NB1 определится следующим образом:

$$Prec\ NB1 = (T3 + T2)/(T3+T2+F3+F2)*100\%. \quad (11)$$

Второй этап метода содержит процедуры обработки основного массива М3, сформированного после фильтрации псевдоспама, с целью формирования результирующего классифицированного массива документов, содержащего N классов (по количеству тематических рубрик, интересующих пользователя). Отметим, что удаление псевдоспама из исходного массива позволяет существенно сократить количество анализируемых в дальнейшем документов, что дает возможность применения на втором этапе более сложных и точных процедур классификации.

Для решения задачи классификации на втором этапе предлагается использовать модифицированный классификатор Байеса (ВМ)). Этот классификатор, в отличие от классификатора NB1, имеет следующие особенности: применение локальных словарей дескрипторов для каждого из формируемых классов; возможность учета связей между дескрипторными терминами при реализации процедуры классификации.

При обучении классификатора ВМ для каждого встреченного в текстах слова рассчитывается и сохраняется его вес – оценка вероятности того, что текст с этим словом принадлежит к одному из возможных классов (одна из возможных аппроксимаций такой оценки состоит в замене бинарной функции взвешивания терминов t_i в векторной модели (1) частотной функцией, ставящей в соответствие этим терминам частоту β_i их появления в анализируемом документе). Обучающая выборка, используемая при настройке ВМ, содержит документы, гарантированно принадлежащие хотя бы к одному из заданных тематических классов. Классификация таких документов осуществляется с помощью локальных словарей дескрипторов для каждого из формируемых классов.

На основе данных о классификации лингвистических дескрипторов при составлении локальных словарей производится расчет целесообразности выбора тех или иных его вариантов с учетом требований к системе, по каждому выбранному классу дескрипторов в различных классификациях. При этом учитывается общее соотношение слов, распределенных для каждой из категорий выбранных подклассов дескрипторов. Для формирования локальных словарей дескрипторов для каждого из формируемых на втором этапе классов рассматриваются такие виды классификации, как «Части речи» и «Частотность». Анализ научно-технических текстов позволяет оценить общее процентное разделение по категориям дескрипторов каждого вида классификации: классификация «Части речи»: существительные (60%), глаголы (13,75%), наречия (6,25%), прилагательные (16,25%), причастия (3,75%); классификация «Частотность»: высокочастотные дескрипторы (60%); среднечастотные дескрипторы (22,5%); низкочастотные дескрипторы (17,5%).

На основе данной классификации, соотношения числа дескрипторов и возможностей добычи их из неструктурированной текстовой информации можно утверждать, что для повышения точности классификации: рациональней всего использовать существительные; по частотной встречаемости видов дескрипторов: целесообразно использовать высокочастотные и низкочастотные дескрипторы. Наиболее соответствующей этим требованиям структурой являются аббревиатуры. Аббревиатуры – это существительные, состоящие из усеченных слов, входящих в исходное словосочетание, или из усеченных частей исходного сложного слова, а также из названий начальных букв этих слов (или их частей). Аббревиатуры являются низкочастотными дескрипторами, которые легко получить из текстов (особенно из их аннотаций). В случае, если в текстах отсутствуют аббревиатуры, целесообразно использовать высокочастотные дескрипторы для максимального охвата всего массива текстовой информации. Для качественной классификации текстов на основе выделенных дескрипторов необходимо прибегнуть к предварительной его очистке от шумов, под которыми понимается категория лексем, мешающих адекватной классификации данных (служебных слов, стоп-слов и т.п.). На втором этапе рассматриваемого метода предусмотрена предварительная обработка текста документов из массива М3: приведение слов в начальную форму, удаление служебных слов, вычисление веса для целых фраз, транслитерация.. К стоп-словам будем относить: союзы и союзные слова; местоимения; предлоги; частицы; междометия; указательные слова; цифры; знаки препинания; отдельно стоящие буквы алфавита; вводные слова. При классификации текстов стоит обращать внимание на наличие стоп-слов из вышеперечисленных категорий и их соотношение с общей массой слов и дескрипторов. Общие шумовые слова часто не учитываются классификатором ВМ, однако, они заменяются специальным маркером. Данное обстоятельство имеет практическое значение при составлении классификатора и оценки плотности ключевых слов разного рода, так как игнорирование стоп-слов влияет на некоторые показатели, которые в свою очередь влияют на точность классификации текстовой информации.

В классификаторе ВМ предусмотрена также возможность учета связей между дескрипторными терминами при реализации процедуры классификации на втором этапе. Для этого реализуется специальная процедура поиска слов, вероятность использования которых в качестве связей между выделенными дескрипторными терминами для каждого из классов высока, с последующим выбором дескрипторов, которые может соединить в будущем локальном словаре дескрипторов данная связь. Это позволяет дополнить начальную совокупность дескрипторов, используемых классификатором ВМ, связными тройками дескрипторов вида «де-

скриптор1-связка12-дескриптор2», наиболее характерными для документов рассматриваемого класса.

Учет в классификаторе таких связных дескрипторов позволяет снизить влияние на качество классифицирующей процедуры электронных текстов гипотезы полной независимости результата классификации от сочетания признаков, которая является одним из наиболее существенных недостатков применения метода Naive Bayes. Формирование совокупности связных дескрипторов осуществляется в предлагаемом методе с применением модифицированного метода Гинзбурга, этапы которого описаны в работе [8]. Процедуры (6)-(11), используемые на первом этапе для определения параметров вероятностного классификатора и оценки качества классификации, дополняются на втором этапе процедурами, реализующими описанные дополнительные функции классификатора BM.

Итоговым результатом классификации является формирование классифицированного массива M3K, содержащего совокупность документов КД31, КД32,…, КД3N, распределенных по соответствующим классам.

2. Программная реализация и тестирование предлагаемого метода классификации

Для программной реализации классификаторов NB1 и BM, а таже алгоритмов выделения наиболее значимых атрибутов были использованы объектно-ориентированный язык программирования Java и среды программной разработки (IDE), в частности, Eclipce и NetBeans.

Разработанный программный модуль Booker V позволяет осуществлять классификацию больших объемов текстовой информации по пользовательским категориям, учитывая возможность распределенного хранения массивов информации.

Общая модель архитектуры модуля BookerV приведена на рис. 2.

Рис. 2. Общая структура модуля BookerV

Разделение модуля на четыре основных блока позволит вносить изменения в систему в любом из блоков, улучшая его работу, при сохраняющемся интерфейсе взаимодействия с пользователем. В блоке «Оболочка». происходит взаимодействие с пользователем с целью получения от него ссылки на библиотеку классифицируемых данных и выдачи пользователю данных о принадлежности к определенному классу некоторой текстовой информации. Схема передачи данных в модуле BookerV приведена на рис. 3.

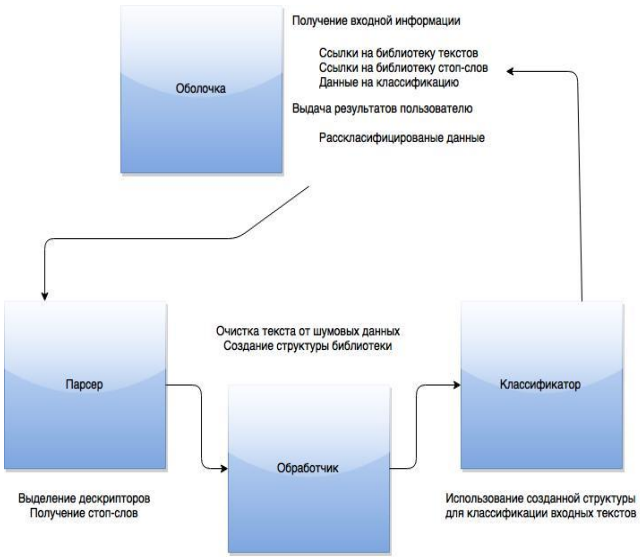

Рис. 3. Передача данных в модуле

Для запуска программы необходимо инициировать исполняемый файл BookerV.jar.

Интерфейс модуля построен с помощью визуальных средств проектирования в среде разработки NetBeans с помощью технологии jxml.

Визуальная оболочка программы состоит из четырех основных структурных блоков: блок загрузки библиотеки; блок системных настроек; блок ввода информации на классификацию; блок вывода информации.

Загрузка входных данных из библиотеки производится через один шлюз, с возможностью дополнительных настроек.

В качестве библиотеки может выступать файл или директория операционной системы. При этом есть возможность подгрузки библиотеки из десериализованного файла библиотеки. При выборе файла (LIB_NAME.data / struct.cvs) или директории необходимо сделать правильные преднастройки для анализа библиотеки. Пример настроек приведен на рис. 4.

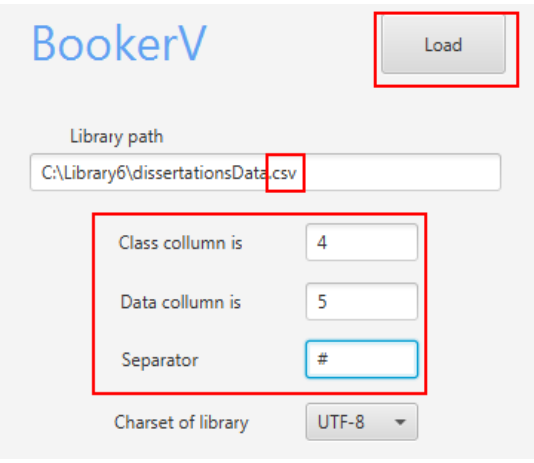

Рис. 4. Настройки модуля (для cvs подобных структур)

Для упрощения взаимодействия с системой был реализован алгоритм определения типа входящей информации перед ее классификацией, что позволяет вводить информацию для классификации следующим образом: ссылки на файлы в расширениях .txt и .cvs; списки ссылок, воспринимаемые построчно; текст в одну строку; текст многострочный; текстовые cvs-подобные структуры.

В блоке вывода информации отображаются результаты классификации документов массива M3. Структура библиотеки сохраняется самостоятельно по окончании сканирования указанного источника. Для выбора пользовательского имени десериализованного файла библиотеки необходимо ввести его заранее, используя полный путь к желаемому создаваемому файлу. При отсутствии пользовательского файла файл создается в корне хранимого источника библиотеки.

Разработанное приложение создано с целью обработки большого количества текстовой информации, его можно использовать на серверных платформах, в электронных библиотеках, а также в организациях с большим объемом документооборота.

Структура части программного модуля Booker V, реализующей функции классификатора NB1, состоит из двух основных частей (рис.5 и рис.6): первая содержит исходный код модуля, интерфейсы взаимодействия с данными и реализацию классификатора, вторая содержит визуальную оболочку для используемого модуля. Каждый файл с расширением .java реализует различные функции программы: обучение на различных видах выборки; классификацию текстовых документов на элементы M3 (ham), отбираемые для дальнейшего анализа, и документы массива M2 (pseudospam); анализ производительности реализованных методов на тестовых выборках. Визуальный интерфейс (вторая часть модуля) реализован с помощью технологии jxml для качественного отображения информации и простоты использования на различных платформах, включая OS-X и Android.

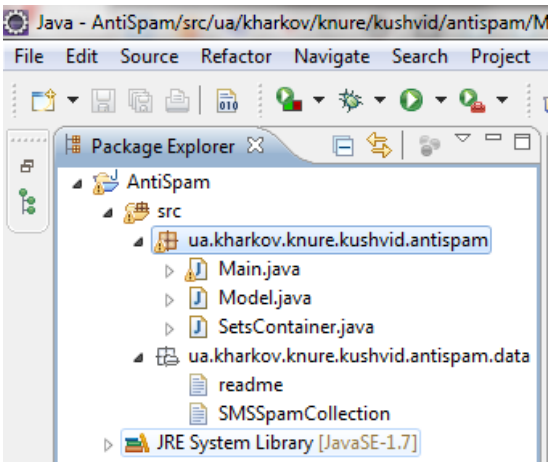

Рис.5 Структура программного модуля Booker V (1)

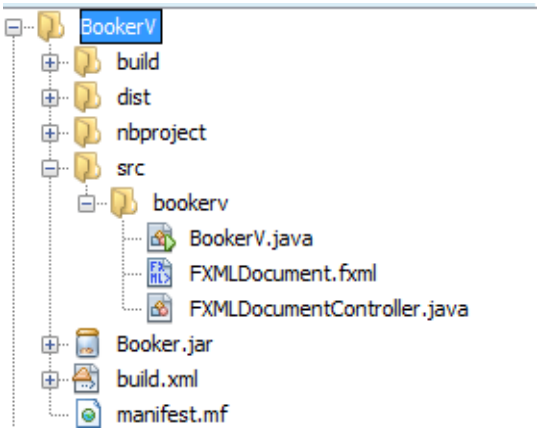

Рис.6 Структура программного модуля Booker V (2)

Фрагмент программы, реализующей основные функции классификатора NB1, приведен на рис. 7

```
public static String classification(String string, Model model,
        boolean preprocess) {
    boolean isSpam = true;
    TreeMap<String, Integer> hamMap = model.hamMap;
    TreeMap<String, Integer> spamMap = model.spamMap;
    double laplaceFactor = model.laplaceFactor;

    HashSet<String> uniqueWordSet = model.uniqueWordSet;

    double ham, spam;

    ham = Math.log((double) hamMap.size()
            / (hamMap.size() + spamMap.size()));
    spam = Math.log((double) spamMap.size()
            / (hamMap.size() + spamMap.size()));

    if (preprocess) {
        string = reformation(string);
    }

    String[] words = string.split(" +");

    for (String word : words) {

        if (hamMap.containsKey(word))
            ham += Math
                    .log((hamMap.get(word) + laplaceFactor)
                            / (hamMap.size() + laplaceFactor
                                    * uniqueWordSet.size()));
        else
            ham += Math
                    .log((laplaceFactor)
                            / (hamMap.size() + laplaceFactor
                                    * uniqueWordSet.size()));

        if (spamMap.containsKey(word))
            spam += Math.log((spamMap.get(word) + laplaceFactor)
                    / (spamMap.size() + laplaceFactor
                            * uniqueWordSet.size()));
        else
            ham += Math.log((laplaceFactor)
                    / (spamMap.size() + laplaceFactor
                            * uniqueWordSet.size()));
    }

    if (ham >= spam)
        return "ham";
    else
        return "spam";
}
```

Рис.7 Программная реализация основных функций NB1

Структура части программного модуля BookerV, реализующей функции классификатора BM для документов из массива M3, фрагмент которой приведен на рис.8, содержит файлы, инициирующие реализацию функций предварительной очистки документов от шумов; применения локальных словарей дескрипторов для каждого из формируемых классов; выделения и учета связей между дескрипторными терминами при реализации процедуры классификации

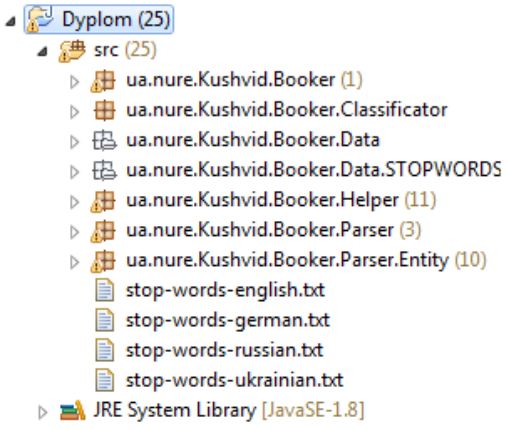

Рис.8 Структура программного модуля Booker V (3)

Система тестировалась на основе бесплатных электронных библиотек, тестовой выборки из массива «Reuters-21578», а также свободно распространяемых списков стоп-слов от компании Google.

При тестировании полиномиальной модели классификатора NB1 использована часть архива выборки электронных документов, приведенного в ресурсе http://archive.ics.uci.edu/ml/dataset/SMS+Spam+Collectio.

По результатам тестирования (с применением сглаживания по Лапласу) точность классификации (Prec NB1) достигала для большинства текстовых выборок 97%.

При тестировании классификатора BM использована часть электронной библиотека «ITAR» объемом 21 ГБ и коллекция аннотаций авторефератов диссертационных работ объемом 2 ГБ.

Поскольку в классификаторе BM реализована возможность пользовательского создания или редактирования личных списков стоп-слов путем указания пути хранения текстовых файлов с их наборами, каждый отдельный файл считался отдельной группой и проход по группам пользовательских стоп-слов происходил в порядке их хранения в директории.

На разных выборках библиотек формировались персональные наборы дескрипторов трех видов: аббревиатуры (например, HDL, IP, JPEG, MIMO, БА, БИС, БПИС, БТС, ВА, ВЗ, ВКАС и т.п.), связные дескрипторы (например, «использование в конкретной функциональной задаче - позволяет расширить - функции администрирования», «модель актуализации оперативных данных - позволяет автоматизировать - функция обработки документов» и т.п.) и общие ключевые слова (например, центральный процессор, оценка быстродействия, программное обеспечение и т.п.).

При исследовании возможностей классификатора BM были использованы два метода тестирования выборок, программно реализованные в модуле BookerV: testModel и crossValidation. Данные функции вызываются после обучения классификатора на обучающей выборке во время его проверки на тестовых данных. Это позволяет: сократить вычислительное время благодаря сокращению числа слов, учитываемых в каждом классе; уменьшить возможность переобучения; снизить влияние выбросов и шумов на результат классификации.

В процессе тестирования осуществлялся подбор наилучшего значения коэффициента размытия по Лапласу (KPM). Исследовано влияние этого коэффициента на точность классификации. Анализ был проведен на примере построения динамических выборок с разными коэффициентами KPM. Кроме того, исследовался алгоритм выбора вида дескрипторов в зависимости от характера выборки для дополнительного повышения качества классификации. Точность двухэтапной классификации текстовых документов с помощью модуля BookerV, для разных серий тестовых экспериментов находилась в диапазоне от 84% до 99,26%.

Выводы

Применение рассмотренного метода классификации является перспективным для классификации по тематическим рубрикам больших массивов документов, имеющего политематический и составной характер. Такими массивами могут быть, например, электронные сборники материалов конференции, содержащих аннотации и основной текст; массив документов большой электронной библиотеки, посвященных некоторому общему научному направлению, и т.п. Первый этап предложенного метода, реализованного в виде программного модуля BookerV, осуществляет фильтрацию исходного массива с целью удаления документов, не представляющие тематический интерес для потенциального пользователя, вероятность отнесения которых к одному из классов результирующего массива очень низка. На втором этапе осуществляется процедура основной классификацации, использующая модифицированный полиномиальный байесовский классификатор. Модификация состоит в применении связных дескрипторов, что позволяет снизить влияние на качество классифицирующей процедуры электронных текстов гипотезы полной независимости результата классификации от сочетания признаков, которая является одним из наиболее существенных недостатков применения метода Naive Bayes.

Результаты тестирования предложенного метода подтверждают целесообразность его использования при решении широкого класса задач политематической классификации электронных текстов. Перспективным развитием метода является проведение экспериментов по усовершенствованию классификаторов NB1 и BM для работы с тематическими документами, характеризующимися наличием неравномощных подтем, а также выбор наиболее эффективных комбинированных критериев оценки качества классификации.

Список литературы: **1** *Sebastiani F.* Machine learning in automated text categorization/ *Sebastiani F.* // ACM Computing Surveys, 34(1), 2002. – pp. 1-47. **2.** Автоматическая обработка текстов на естественном языке и компьютерная лингвистика : учеб. пособие / Большакова Е.И., Клышинский Э.С., Ландэ Д.В., Носков А.А., Пескова О.В., Ягунова Е.В./ – М.: МИЭМ, 2011. – 272 с. **3.** Епрев А.С. Автоматическая классификация текстовых документов / А.С. Епрев // Математические структуры и моделирование. – 2010 – Вып.21. – С. 65-81. **4.** Domingos P. On the optimality of the simple Bayesian lassifier under zero-oneloss / P..Domingos, M.Pazzani // Machine Learning. - 1997. - № 29. - pp. 103-137. **5.** *Петровский М.И.* Алгоритмы машинного обучения для задачи анализа и рубрикации электронных документов/ М.И. Петровский, В.В. Глазкова // Вычислительные методы и программирование – 2007. – Т. 8– № 2. – С. 57-69. **6.** *Рыжов.А.П.* О качестве классификации объектов на основе нечетких правил / А.П. Рыжов// Интеллектуальные системы – 2005 – Т.9. – С. 253-264. **7.** Чалая, Л. Э. Оценивание пертинентности лингвистических дескрипторов в системах информационного поиска документов [Текст] / Л.Э. Чалая, Ю.Ю. Харитонова// Восточно-европейский журнал передовых технологий. – 2015. – № 1/9(73). – С. 46–53 **8.** Чалая, Л. Э. Метод поиска пертинентных связей между концептами проектируемых онтологий [Текст]/ Л.Э. Чалая, А.В. Чижевский, Е.Б. Волощук// Комп'ютерно-інтегровані технології: освіта, наука, виробництво. – 2016. – №22. – С.50–56.

Поступила в редколлегию 23.10.2016

УДК 004.853

Метод двохетапної класифікації електронних текстів. /Л.Е. Чала, С.Г. Удовенко, Є.В. Кушвід // Біоника інтелекту: наук.-техн. журнал. 2016. № 2 (87). Р. 00-00.

У статті розглянуто метод двохетапної класифікації електронних документів. На першому етапі використовується поліноміальна модель байєсівського класифікатора для фільтрації псевдоспаму. На другому етапі здійснюється тематична класифікація документів основного масиву з використанням зв'язних лінгвистичних дескрипторів.

Іл. 8. Бібліогр.: 8 назв.

UDK 004.853

Method of a twostage electronic text classification / L/E/ Chala, S.G. Udovenko, Ye.V. Kushvid // Bionica Intellecta: Sci. Mag. 2016. № 2 (87). P. 00-00.

The method of a twostage classification of electronic documents is considered in article. On the first stage the поліноміальна model of Bayesian classifier is used for filtration of pseudospam On the second stage thematic classification of documents of basic document array is realised with the use of coherent linguistic descriptors.

Fig. 8. Ref.: 8 items.

Рецензент: професор кафедри штучного інтелекту ХНУРЕ, д.т.н., проф. Бодянський Є.В.