Содержание

Условные обозначения

ТК — текстовая категоризация;

РМ — разрешение многозначности;

D — множество документов;

C — множество категорий;

c_i — элемент множества категорий;

d_j — элемент множества документов;

Tr — обучающее множество документов;

Te — тестовое множество документов;

a_{ij} — элемент матрицы сопоставления;

ca_{ij} — элемент матрицы корректного сопоставления;

w_{ki} — вес k-ого термина в векторном представлении i-ой категории;

k-NN — «k nearest neighbours», алгоритм «k ближайших соседей»;

CSV_i — «categorization status value», функция принадлежности документа к категории c_i .

1 Введение

Автоматическая текстовая категоризация — такая область искусственного интеллекта, с которой мы сталкиваемся практически повседневно: при поиске информации в Internet с помощью поисковых машин, при получении почты спам-фильтр отсеивает сомнительные письма. Текстовая категоризация (ТК) используется во многих прикладных задачах, от автоматической индексации документов на основе управляемого словаря до фильтрации документов, автоматической генерации метаданных, разрешения омонимии, расширения иерархических каталогов веб-ресурсов, упорядочения документов и выборочной рассылки почты. Хотя коммерческие системы ТК пока еще не широко распространены (например, системы поиска коммерческой информации), экспериментальные системы ТК давно уже отошли от стадии опытных образцов благодаря наличию огромного количества солидной литературы по экспериментальным методам. В некотором смысле, ТК можно рассматривать как точку пересечения дисциплин машинного обучения и информационного поиска, послужившего основой почти для всех отраслей, связанных с автоматической смысловой обработкой документов.

Автоматическая категоризация (или классификация) текстов на тематические категории имеет длинную историю, относящуюся к началу 1960-х годов. Вплоть до конца 1980-х годов наиболее эффективным подходом к ТК, скорее всего, было построение системы автоматической ТК «вручную» с помощью методов из области инженерии знаний. Суть этих методов заключалась в определении набора логических правил, в которых зашифровывалось экспертное знание о способе классификации документов по заданному множеству категорий. Узким местом «ручных» методов построения классификаторов является, как и в случае с экспертными системами, сложность накопления знаний.

В последнее десятилетие задачи автоматической смысловой обработки документов получили довольно высокий приоритет в области информационных систем, в значительной степени благодаря широкой доступности и увеличению числа документов к электронной форме, и, как следствие, потребности в быстром доступе к ним со стороны пользователя. Благодаря появлению больших коллекций электронных документов в 90-е годы на смену прежним методам пришло машинное обучение, заместившее подходы на основе экспертных знаний. Согласно идее машинного обучения общий индуктивный процесс автоматически строит классификатор, «изучая» множество предварительно классифицированных документов и характеристики интересующих категорий. По сравнению с подходами на основе знаний, идея машинного обучения обладает рядом преимуществ: усилия раз-

работчиков направлены не на построение классификатора, а на составление алгоритма построения классификатора. Таким образом, в случае изменения набора категорий или применения системы в совершенно другой области, потребуется лишь вновь провести автоматическое обучение классификатора на новом обучающем множестве. При этом не требуется вмешательства ни экспертов в предметной области, ни в программировании. С точки зрения эффективности классификаторы, обученные с помощью методик машинного обучения, на сегодняшний день достигли впечатляющих уровней эффективности, поэтому автоматическая категоризация качественно не уступает классификации вручную.

Широкий спектр задач, в которых возможно применение методик классификации, и доступность больших коллекций документов в электронном виде, которые можно использовать в качестве обучающих корпусов, стали причиной живого интереса к данной области со стороны многих прикладных дисциплин.

В данной работе рассматривается применение текстовой категоризации к определению значений многозначных слов на основе контекста. Обращение в задаче разрешения многозначности (РМ) именно к методикам текстовой категоризации (а не кластеризации, где множество категорий требуется определить) обусловлено тем, что набор категорий задан заранее. При определении смысла многозначных слов в качестве категорий служат значения изучаемого слова. В отличие от стандартных корпусных методов РМ, суть которых заключается в использовании локальных характеристик контекста (например, рассмотрение слов, смежных с исследуемым), предлагается учитывать также глобальные свойства контекста такие, как стиль, тема, форма, временная соотнесенность. Следовательно, в этом случае потребуется такой механизм автоматической ТК, который будет классифицировать документ сразу по нескольким «измерениям». И вполне естественно, что для каждого способа классификации будет использован специфический набор признаков и, возможно, отдельный алгоритм обучения. Данный принцип классификации документа сразу по нескольким множествам категорий был реализован на примере экспериментального приложения, определяющего значения некоторых многозначных слов на основе контекста.

2 Автоматическая классификация документов

2.1 Текстовая категоризация как задача машинного обучения

Под *текстовой категоризацией* (или текстовой классификацией) понимают автоматическое отнесение текста на естественном языке к заранее определенному набору категорий. Нужно отметить, что под «автоматической классификацией текстов» порой подразумеваются довольно различные проблемы. Следует отличать задачу автоматической разметки текстов с помощью категорий из заданного множества (собственно, текстовой категоризации) от задачи автоматического определения множества категорий по заданному набору текстов (*кластеризации*) и от задачи отнесения документов к набору категорий, которые не заданы заранее (*свободного текстового индексирования*).

Формально, текстовую категоризацию можно определить как задачу нахождения матрицы сопоставления, каждый элемент которой $a_{ij} = 1$, если документ d_j относится к категории c_i, и $a_{ij} = 0$ в противном случае:

	d_1	...	...	d_j	...	...	d_n
c_1	a_{11}	...	...	a_{1j}	...	...	a_{1n}
...	...	...	...	...	...	...	...
c_i	a_{i1}	...	...	a_{ij}	...	...	a_{in}
...	...	...	...	...	...	...	...
c_m	a_{m1}	...	...	a_{mj}	...	...	a_{mn}

Таблица 2.1

где $C = \{c_1, ..., c_m\}$ — множество заранее определенных категорий, а $D = \{d_1, ..., d_n\}$ — множество классифицируемых документов.

Более точно задачу категоризации можно определить как задачу аппроксимации функции $f : D \times C \to \{0,1\}$ (определяющей распределение документов по категориям) с помощью функции $f' : D \times C \to \{0,1\}$ (называемой классификатором, модельной или гипотетической функцией) так, чтобы f и f' были наиболее близки. Существуют различные способы определения степени близости (*эффективности*).

Относительно понятия ТК существуют следующие соглашения:

- Категории — всего лишь символические метки и никакого другого смысла в них не вкладывается. В частности, это означает, что при категоризации не используется

смысл «слова», обозначающего какую-либо категорию.

- Для ТК используется только информация, полученная из самого текста, а не из сторонних источников и метаданных (даты публикации, типа документа, издательства и проч.). То есть, классификация документов должна полностью основываться на эндогенной информации, и не использовать экзогенные знания.

Поскольку понятие семантики документа по существу является субъективным, то и центральное понятие ТК, отношение документа к категории, тоже не может быть определено однозначно. Это хорошо видно на примере известного феномена несогласованности взаимной индексации, когда два человека, классифицирующих документы, могут отнести один и тот же текст к разным категориям, что происходит довольно регулярно.

С точки зрения машинного обучения ТК представляет собой процесс индуктивного построения (обучения) классификатора для категории c_i, «изучающий» множество предварительно классифицированных документов и их характеристики. На основе этих характеристик индуктивный процесс строит правило, которое определяет, каков должен быть документ, чтобы его можно было отнести к c_i. В терминологии машинного обучения данный процесс называется «обучением с учителем», так как требует участия эксперта при составлении обучающего множества документов.

Негативным явлением в методе машинного обучения является феномен переобучения. Поскольку основное требование, предъявляемое к модельной функции f', заключается в минимизации ошибок, получающихся при ее применении к новым неисследованным экземплярам, то, конечно, имея некоторое множество обучающих данных, можно построить функцию, в точности удовлетворяющую этим данным (Рисунок 2.1). Но в случае наличия «шумов» это будет не лучший вариант, т.к. на неисследованных экземплярах такая функция будет сильно ошибаться (такое явление и называют переобучением). Основной идеей при разработке обучающего алгоритма является его нацеленность на поиск закономерностей в исследуемых явлениях (обучающих данных). В общих словах можно сказать, что мы переходим от исследованного прошлого в будущее. Мы просматриваем набор возможных моделей и ищем среди них ту, которая наилучшим образом удовлетворяет обучающим данным, но в то же время имеет самую «простую» структуру из возможных.

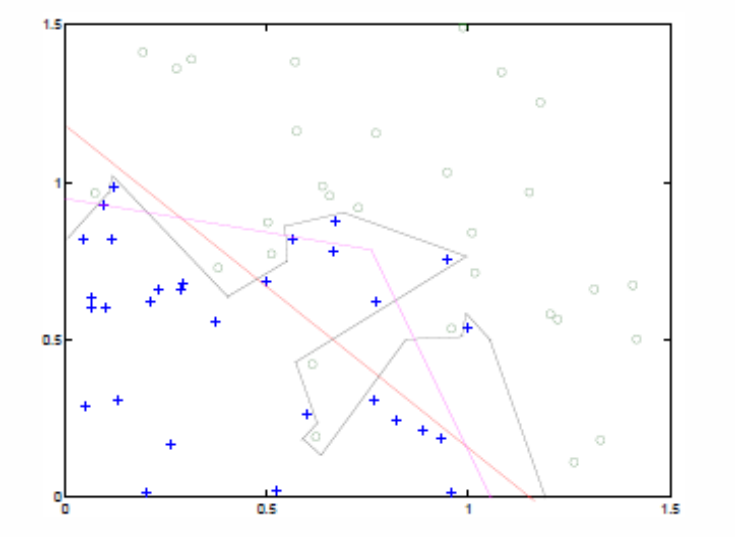


Рисунок 2.1

Тем самым возникает необходимость построения *устойчивых* алгоритмов (например, «*k* ближайших соседей» для подходящего *k*), для которых, по мере поступления новых данных, точность предсказания стремится все ближе и ближе к оптимальному значению.

Построение классификатора основывается на начальном корпусе документов $Co=\{\bar{d}_1,\dots,\bar{d}_s\}$, отнесенных к категориям из того же множества $C=\{c_1,\dots,c_m\}$, с которым будет в дальнейшем работать система. Начальный корпус обычно задается в виде матрицы корректного сопоставления:

	Training set				Test set			
	$\bar{d}_1$	…	…	$\bar{d}_g$	$\bar{d}_{g+1}$	…	…	$\bar{d}_s$
c_1	ca_{11}	…	…	ca_{1g}	$ca_{1(g+1)}$	…	…	ca_{1s}
…	…	…	…	…	…	…	…	…
c_i	ca_{i1}	…	…	ca_{ig}	$ca_{i(g+1)}$	…	…	ca_{is}
…	…	…	…	…	…	…	…	…
c_m	ca_{m1}	…	…	ca_{mg}	$ca_{m(g+1)}$	…	…	ca_{ms}

Таблица 2.2

где ca_{ij} = 1 соответствует принадлежности, с точки зрения эксперта, d_j к категории c_i. Документ d_j называется положительным экземпляром c_i, если ca_{ij} = 1, и отрицательным экземпляром c_i, если ca_{ij} = 0.

Обычно начальный корпус документов разбивают на два, не обязательно равных, подмножества:

- *обучающее множество* $Tr=\{\bar{d}_1,\dots,\bar{d}_g\}$. Изучая характеристики представителей данного множества, индуктивный процесс будет строить классификатор для раз-

личных категорий.

- *тестовое множество* $Te=\{\bar{d}_{g+1}, \ldots, \bar{d}_s\}$. Это множество будет использоваться для оценки эффективности полученного классификатора.

2.2 Индексация документов

2.2.1 Способы индексации документов

Текстовые документы в непосредственном виде не подходят для интерпретации классификатором или алгоритмом построения классификатора. Поэтому необходимо применение процедуры индексации, которая переводит текст в удобное представление. Очевидно, что для индексации обучающих и тестовых документов должна применяться один и тот же метод индексации.

Выбор представления текста зависит от того, что считать значимыми текстовыми единицами (проблема лексической семантики) и значимыми правилами естественного языка для комбинации этих единиц (проблема композиционной семантики). В традициях информационного поиска каждый документ обычно представляется в виде вектора из n взвешенных терминов. Различия в подходах заключаются:

(1) в понимании, что такое термин;

(2) в способах определения веса термина.

Обычно терминам соответствуют все слова, встречающиеся в документе. Такой подход интерпретирует текст как *набор слов*. В ряде экспериментов ([2], [14], [26]) было обнаружено, что даже более сложное представление менее эффективно. В частности, некоторые авторы пробовали использовать *именные группы* в качестве индексных терминов, но результаты экспериментов не оказались поразительными, вне зависимости от того, что подразумевалось под «группой»:

- *синтаксические группы*, т.е. фраза, построенная в соответствии с грамматикой языка;
- *статистическая группа*, т.е. набор/последовательность слов, которые с высокой долей вероятности встречаются вместе в коллекции документов.

Д. Д. Льюис [26] довольно убедительно утверждает, что скорее всего причиной неутеши-

тельных результатов является то, что методы индексирования на основе фраз обладают худшими статистическими характеристиками по отношению к методам на основе одиночных слов, хотя их семантические качества гораздо выше. Несмотря на неутешительные результаты, исследования эффективности индексации фраз до сих пор продолжаются. Особенно это характерно для статистических групп, так как для них утверждение Льюиса менее справедливо.

Не уменьшая общности, веса терминов обычно нормализуют, чтобы они варьировались от 0 до 1. Частным случаем является двоичный вектор (1 означает наличие термина в документе, а 0 – отсутствие), обычно используемый в обучаемых системах нечислового, символьного характера. Для недвоичного представления вектора чаще всего используются методики информационного поиска. Например, применяется стандартная функция *tfidf*, определяемая как

$$tfidf(t_k, d_j) = N(t_k, d_j) \cdot \log\left(\frac{|Tr|}{N(Tr(t_k))}\right)$$

где $N(t_k,d_j)$ – количество терминов t_k в документе d_j, а $N(Tr(t_k))$ – количество документов в обучающем множество *Tr*, в которых встречается термин t_k (также известная как *документная частота*). Данная функция заключает в себе интуитивную идею того, что (i) чем чаще термин встречается в документе, тем лучше он отражает его содержание, и (ii) чем в большем количестве документов встречается термин, тем менее значимым он является для классификации. На самом деле, *tfidf* скорее целый класс функций, отличающихся друг от друга способами нормализации терминов и другими корректирующими множителями. Вышеприведенная формула – лишь один из возможных примеров этого класса.

Отметим, что эта формула оценивает значимость термина только с точки зрения частоты вхождения в документ, тем самым не учитывая порядок следования терминов в документе и их синтаксическую роль; другими словами, семантика документа сводится к лексической семантике входящих в него терминов, а композиционная семантика не рассматривается.

Для того, чтобы веса находились в интервале (0, 1), а векторы документов имели равную длину, значения *tfidf* обычно *нормализуются по косинусу*:

$$w_{kj} = \frac{tfidf(t_k, d_j)}{\sqrt{\sum_{s=1}^{r} (tfidf(t_s, d_j))^2}}$$

где *r* – количество всех терминов в обучающем множестве *Tr*.

Хотя *tfidf* довольно популярна, также используются другие индексирующие функции, включая вероятные способы индексирования [16] и методики индексирования структурированных документов [25]. Иные функции индексации могут потребоваться в тех случаях, когда изначально обучающее множество не дано (например, при адаптивной фильтрации), и документную частоту не удаётся посчитать. В этих случая *tfidf* заменяют на более эмпирические функции [12].

В зависимости от приложения, могут индексироваться как весь тест, так и отдельные его части. Например, при категоризации патентов у Л. Ларки [24] берутся в рассмотрение только заголовок, аннотация, первые двадцать строк истории вопроса и раздел, описывающий новизну изобретения. Этот подход возможен при наличии знаний о структуре классифицируемого документа. Подобным же образом, при наличии у документа заголовка, образующим его словам можно придавать большую значимость. В противном же случае, определение наиболее важных разделов неструктурированных документов – довольно неочевидная задача.

2.2.2 Предварительная обработка

Перед индексацией обычно происходит удаление функциональных слов (семантически нейтральных слов, таких как артикли, союзы, предлоги). Что касается морфологической обработки, существуют противоречивые мнения относительно полезности данного шага в текстовой категоризации. Некоторые исследования ([3]) отмечают снижение эффективности при использовании морфологической обработки, хотя в основном многие прибегают к ней, поскольку это способствует значительному сокращению размерности пространства терминов и уровня стохастической зависимости между ними.

2.2.3 Дополнительная обработка

Иногда для эффективной работы классификатора может потребоваться сокращение пространства терминов. Ведь количество терминов в корпусе может достигать десятков тысяч. Если типичный алгоритм в поисковых системах (такой, как совпадение по косинусу) легко справляется с большой размерностью, то для более сложных обучающихся алгоритмов индуктивного построения классификатора это большая проблема. В этих случаях часто применяются методики сокращения размерности, сужающие векторное пространство до размерности $r' << r$.

Сокращение размерности также полезно для снижения эффекта переобучения – яв-

ления, при котором классификатор следует в большей степени случайным, ошибочным, чем важным и значимым характеристикам обучающих данных. Переобученный классификатор отлично работает на тех экземплярах, на которых он обучался, и значительно хуже на тестовых данных. Например, если классификатор для категории «Машины на продажу» был обучен на трех примерах, двое из которых относятся к продаже желтых машин, то классификатор определит случайное свойство обучающих данных («желтый») как основополагающее для данной категории. Эксперименты показали, что для того, чтобы избежать переобучения, количество обучающих примеров должно быть соразмерно числу используемых терминов. Н. Фур и К. Бакли [15] предлагали использовать 50-100 обучающих примеров на термин. Это значит, что даже при меньших размерах обучающего множества можно снизить эффект переобучения засчет сокращения размерности пространства терминов.

Предлагаются различные функции сокращения размерности, как из теории информации, так и из линейной алгебры. Их относительные достоинства обычно сравниваются экспериментально, оценивая эффективность категоризации классификаторов, использующих ту или иную функцию, примененную к пространству терминов.

По степени локализации задачи относительно категорий выделяют два вида сокращения размерности:

- *локальное*: для каждой категории c_i выбирается $r'_i \ll r$ терминов. Таким образом, каждый документ d_j обладает различным представлением для каждой категории. На практике это означает, что для классификации на m категорий понадобится использовать m векторов для одного документа. Авторы, использовавшие этот подход, сокращали размерность пространства до $10 \leq r' \leq 50$.
- *глобальное*: выбирается общее количество $r' << r$ терминов для всех категорий $C = \{c_1, \ldots, c_m\}$.

Данное разделение не касается методики сокращения размерности, поэтому довольно большое количество методов может использоваться и как для локального, и как для глобального случая.

По характеру получения результата можно выделить:

- сокращение размерности с помощью отбора терминов: из множества первоначальных r терминов выбирается подмножество из r' элементов.
- сокращение размерности с помощью синтеза терминов: новый набор терминов не

является подмножеством первоначальных, а получается из него с помощью комбинаций и трансформаций. Результат может представлять собой множество совершенно иной природы, отличной от слов.

Очевидно, что для каждого из этих случаев используются довольно различные приемы.

Сокращение размерности отбором терминов

Методы отбора терминов (или *сужения пространства терминов*) выбирают из первоначального множества $r' << r$ терминов, которые будут использоваться для индексации документа. Опубликованные результаты показывают даже небольшое возрастание эффективности после сужения пространства терминов ($\leq 5\%$, в зависимости от классификатора, *степени сокращения* $\frac{r}{r'}$ и метода сужения пространства).

И. Мулине [33] экспериментировал с так называемым контейнерным методом отбора, когда новое множество терминов получается с помощью метода обучения, используемого при построении классификатора. Новое множество терминов формируется из старого засчет добавления и удаления терминов. После формирования нового пространства терминов происходит обучение классификатора и его проверка на тестовом множестве документов. То множество терминов, на котором была достигнута наибольшая эффективность, выбирается в качестве окончательного. Преимущество этого подхода заключается в том,

- что полученное множество терминов получается подстроенным под обучающийся алгоритм;
- в случае локального отбора для каждой категории может быть выбрано различное количество терминов, в зависимости от степени отличия категории от других.

С другой стороны, конечно, этот подход заключает в себе полной перебор, при котором количество различных перебираемых множеств не позволяет использовать этот метод в стандартных приложениях текстовой классификации.

Более простым с точки зрения производительности является метод фильтрации ([21]). В соответствии со значениями определенной числовой функции, которая определяет «важность» термина для классификации, выбираются $r' << r$ терминов, на которых достигаются наибольшие значения данной функции.

Документная частота. Простая и удивительно эффективная функция отбора терминов — *документная частота* $N(Tr(t_k))$ термина t_k — была впервые использована К.

Апте [2], а затем систематично рассмотрена И. Янгом и Дж. Педерсеном ([44]). Будем называть обучающее множество документов *пространством событий*; в традициях теории вероятности документную частоту можно также обозначить $P(t_k)$, как вероятность того, что термин t_k встретится в произвольно взятом документе обучающей выборки. И. Янг и Дж. Педерсен показали, что использование данной методики позволяет сократить пространство терминов в 10 раз без потери эффективности, вне зависимости от применяемого классификатора и корпуса текстов (сокращение в 100 раз дает небольшую потерю в эффективности).

По существу, этот результат позволяет утверждать, что при категоризации наиболее ценными являются те термины, которые встречаются в коллекции чаще. На первый взгляд, это противоречит основному стереотипу информационного поиска, согласно которому наиболее информативные термины имеют низкую или среднюю документную частоту. Но, на самом деле, эти два результата не противоречат друг другу, поскольку подавляющее большинство слов, встречающихся по крайней мере один раз в корпусе, имеют очень малую документную частоту. Тем самым, при десятикратном сокращении множества терминов будут удалены только эти слова, и останутся слова с малой, средней и высокой частотой. Конечно, этот подход подразумевает предварительное удаление стоп-слов, иначе после сокращения множество терминов будет состоять только из семантически нейтральных слов.

Напоследок хочется отметить, что также применяется немного более эмпирическая форма этого метода, при которой из рассмотрения выбрасываются термины, встречающиеся менее чем в x обучающих документах (часто x=1,...,3). Такой подход используется и как самостоятельный способ сокращения размерности, и как предварительный шаг перед более сложным алгоритмом. Еще одной разновидностью данного подхода может служить удаление терминов, встречающихся менее x раз в обучающей коллекции, где x обычно принимает значения от 1 до 5.

Другие информационно-теоретические функции отбора терминов. В различных исследованиях были использованы и другие, более сложные информационно-теоретические функции (Таблица 2.3), такие как

- хи-квадрат,
- коэффициент корреляции,
- прирост информации,
- взаимная информация,

- степень расхождения,
- степень значимости,
- упрощенное хи-квадрат.

Function	Denoted by	Mathematical form
Document frequency	$\#(t_k, c_i)$	$P(t_k \mid c_i)$
Information gain	$IG(t_k, c_i)$	$P(t_k, c_i) \cdot \log \frac{P(t_k, c_i)}{P(c_i) \cdot P(t_k)} + P(\overline{t}_k, c_i) \cdot \log \frac{P(\overline{t}_k, c_i)}{P(c_i) \cdot P(\overline{t}_k)}$
Mutual information	$MI(t_k, c_i)$	$\log \frac{P(t_k, c_i)}{P(t_k) \cdot P(c_i)}$
Chi-square	$\chi^2(t_k, c_i)$	$\frac{g \cdot [P(t_k, c_i) \cdot P(\overline{t}_k, \overline{c_i}) - P(t_k, \overline{c_i}) \cdot P(\overline{t}_k, c_i)]^2}{P(t_k) \cdot P(\overline{t}_k) \cdot P(c_i) \cdot P(\overline{c_i})}$
Correlation coefficient	$CC(t_k, c_i)$	$\frac{\sqrt{g} \cdot [P(t_k, c_i) \cdot P(\overline{t}_k, \overline{c_i}) - P(t_k, \overline{c_i}) \cdot P(\overline{t}_k, c_i)]}{\sqrt{P(t_k) \cdot P(\overline{t}_k) \cdot P(c_i) \cdot P(\overline{c_i})}}$
Relevancy score	$RS(t_k, c_i)$	$\log \frac{P(t_k \mid c_i) + d}{P(\overline{t}_k \mid \overline{c_i}) + d}$
Odds Ratio	$OR(t_k, c_i)$	$\frac{P(t_k \mid c_i) \cdot (1 - P(t_k \mid \overline{c_i}))}{(1 - P(t_k \mid c_i)) \cdot P(t_k \mid \overline{c_i})}$
Simplified Chi-square	$s\chi^2(t_k, c_i)$	$P(t_k, c_i) \cdot P(\overline{t}_k, \overline{c}_i) - P(t_k, \overline{c}_i) \cdot P(\overline{t}_k, c_i)$

Таблица 2.3

Функция *прирост информации* также известна как *вероятная взаимная информация*. В формулах χ^2 и *CC* коэффициент *g* — обычно мощность обучающего множества. Вероятности имеют обычную интерпретацию для пространства документов и вычисляются напрямую на обучающем множестве. Например, $P(\overline{t}_k, c_i)$ означает вероятность того, что для произвольно взятого документа *x*, термин t_k не встречается в *x*, и *x* относится к категории c_i. Большинство данных функций реализуют идею того, что самые значимые для категории c_i термины имеют совершенно разное распределение на множествах положительных и отрицательных экземпляров.

Эти функции дают более качественный результат, чем документная частота: И. Янг и Дж. Педерсен [44] на различных классификаторах и различных обучающих корпусах показали, что такие методы, как *IG* или χ^2 могут сокращать пространство терминов в 100 раз без потери эффективности категоризации или даже с небольшим её увеличением.

К сожалению, сложность данных информационно-теоретических метрик не всегда позволяет быстро определить причину таких хороших результатов; другими словами, использование этих метрик в качестве функций отбора параметров не всегда теоретически

обосновано.

В этом отношении, наблюдения [34] показали, что использование $\chi^2(t)$ необоснованно, и объясняли они это следующим образом: значение в числителе, возводимое в квадрат, уравнивает множители, характеризующие положительную корреляцию термина и категории ($P(t, c_i)$ и $P(\bar{t}, \bar{c_i})$), с множителями, характеризующими отрицательную корреляцию ($P(t, \bar{c_i})$ и $P(\bar{t}, c_i)$). Коэффициент корреляции *CC(t)*, как квадратный корень из $\chi^2(t)$, усиливает значимость первых и уменьшает вес последних. Результаты экспериментов [34] показали превосходство *CC(t)* над $\chi^2(t)$, но нужно отметить, что эти результаты справедливы к локальному, а не к глобальному отбору терминов.

Следующий шаг в этом направлении сделал Н. Фур [16], заметивший что в $CC(t_k,c_i)$ (и в большей степени $\chi^2(t_k, c_i)$)

- множитель $\sqrt{g}$ в числителе не существен, поскольку он постоянен для всех пар (t_k,c_i);
- наличие $\sqrt{P(t_k)\cdot P(\bar{t_k})}$ в знаменателе выделяет очень редкие термины, что, как ясно было показано в [44], мало эффективно для ТК;
- наличие $\sqrt{P(c_i)\cdot P(\bar{c_i})}$ в знаменателе подчеркивает очень редкие категории, что весьма необоснованно.

Сокращение размерности с помощью синтеза терминов

Синтез терминов (или *репараметризация*) подразумевает получение из начального множества терминов нового множества $r' << r$, на котором достигается максимальная эффективность. В качестве объяснения использования синтетических, а не естественных терминов можно сказать, что при использовании обычных слов возникают проблемы, связанные с многозначностью, омонимией и синонимией, поэтому их использование в качестве независимых измерений может быть не совсем оптимально. Методы синтеза терминов направлены на решение этих задач, и получаемые в результате искусственные термины не подвержены воздействию данных негативных явлений. В области текстовой категоризации применялись в особенности два подхода: кластеризация терминов и латентно-семантическое индексирование.

Кластеризация терминов. Кластеризация терминов заключается в группировке слов с высокой степенью попарно-семантической близости в кластеры (кластерные центры), которые в последствии используются в качестве базиса векторного пространства.

Кластеризация терминов имеет совершенно иную природу, в отличие от отбора терминов. Кластеризация сокращает размерность пространства засчет излишних терминов, которые являются синонимами (или семантически близкими понятиями), а отбор терминов удаляет неинформативные слова. Любой метод кластеризации должен определять:

- способ группировки слов в кластеры
- способ перевода первоначального представления документа в пространство с новым синтетическим базисом.

Первым влияние кластеризации терминов на ТК исследовал Д. Льюис [26]. Его метод (обратная кластеризация по соседям) заключался в группировке терминов в пары в соответствии с некоторой метрикой сходства. Результаты оказались ниже тех, которые получались при простом индексировании, в следствие, возможно, плохих характеристик метода кластеризации. Сам Льюис писал: «Взаимосвязи терминов в кластерах оказались по большей мере случайными, а не обусловленными, как мы ожидали».

Другим примером этого подхода является работа И. Ли и А. Джейн [31], которые связывали термины в кластеры на основе их взаимной смежности-рассеянности в тексте. Используя эту методику с применением алгоритма иерархической кластеризации, они добились лишь небольшого увеличения эффективности; однако, малый объём их эксперимента вряд ли позволяет делать какие-либо строгие заключения.

В обоих вышеприведенных случаях использовалась кластеризация без учителя, которая не учитывала категориальную разметку обучающих документов. В свою очередь, Л. Бейкер и А. МакКаллум [3] предложили вариант кластеризации с учителем. В своем методе распределительной кластеризации они объединяли термины, которые в наивысшей степени определяли принадлежность документа к одной и той же категории (или группе категорий). Их эксперименты с использованием классификатора Байеса показали лишь 2% потерю в эффективности при степени сокращения пространства $\frac{r}{r'}=1000$, и даже некоторый прирост эффективности при меньших уровнях сжатия.

Латентно-семантическое индексирование.

Латентно-семантическое индексирование (ЛСИ) как метод сокращения пространства терминов появился в контексте задач информационного поиска при использовании слов-синонимов и многозначных слов в представлении документов и тексте запроса. В основе этого метода сжатия лежит представление базиса нового пространства в виде ли-

нейной комбинации базисных векторов первоначального пространства с использованием матриц взаимной частоты вхождения терминов. На деле, ЛСИ определяет зависимости между терминами в корпусе документов и использует эти зависимости как новые независимые измерения. При отображении изначального пространства в новое, к матрице инцидентности, составленной из векторов документов, применяется сингулярное разложение. В дальнейшем, это же линейное отображение, что было получено на обучающем множестве, применяется к тестовым документам, переводя их в пространство меньшей размерности.

Одной из характеристик ЛСИ в качестве метода сокращения размерности является отсутствие интуитивной интерпретации новых базисных векторов (в отличие от отбора терминов и кластеризации терминов). Однако, похоже что этот подход неплохо работает, извлекая «скрытые» семантические структуры из словаря обучающего корпуса. Например, Х. Шутце [38] описывал случай классификации для категории «Демографические изменения в США под влиянием экономики» с использованием нейронной сети. Как отмечают исследователи, среди положительных экземпляров данной категории был документ, содержащий довольно замысловатое предложение: «В 1980х численность населения выросла до 249.6 млн человек, т.к. большее количество граждан стало покидать промышленные и сельскохозяйственные районы Юга и Запада». Этот же документ был неправильно классифицирован при использовании функции отбора параметров χ^2, сократившей количество терминов до 200. Этот пример хорошо объясняет работу ЛСИ: в приведенном выше предложении не содержится ни одного из 200 терминов высшего ранга, отобранных функцией χ^2 для данной категории, но вполне вероятно, что эти слова могли составить один или несколько ЛСИ-терминов высшего уровня, порождающих векторное пространство для категории. Как писал по этому поводу Х. Шутце: «В тех случаях, когда в большом количестве терминов каждый несет малую часть важной информации, классификация, основанная на простых словах, становится проблематичной». К недостаткам ЛСИ можно отнести возможное уменьшение дискриминантной способности тех терминов, которые сами по себе хорошо характеризуют категорию.

Е. Винер [41] использовал ЛСИ в двух альтернативных вариантах: (i) для локального сокращения размерности, создавая несколько ЛСИ-представлений для различных категорий и (ii) для глобального сокращения размерности, создавая единственное представление для всего множества категорий. Результаты экспериментов показали, что первый вариант работает лучше, чем второй. Но, в любом случае, оба эти варианта были эффективнее простого отбора терминов на основе «степени значимости».

Х. Шутце проводил эксперименты по сравнению ЛСИ и χ^2 с использованием трех различных классификаторов (а именно, линейный дискриминантный анализ, логистическую регрессию и нейронные сети) на примере поискового приложения. Результаты на первых двух классификаторах показали значительное превосходство ЛСИ над χ^2, а в случае с нейронной сетью оба метода работали в равной степени хорошо.

2.3 Методы индуктивного построения классификаторов

Индуктивное построение классификатора для категории $c_i \in C$ заключается в определении функции принадлежности документа к категории $CSV_i : D \to [0,1]$, которая, грубо говоря, представляет собой степень отношения (от 0 до 1) документа d_j к категории c_i. Данная функция CSV_i имеет различную структуру, в зависимости от классификатора: например, в «наивном» байесовском подходе она определяется с помощью вероятностей, в то время как в методе Роккио CSV_i представляется в виде метрики в r-мерном пространстве.

2.3.1 Вероятностные классификаторы

В вероятностном подходе функция принадлежности документа к категории $CSV_i(d_j)$ рассматривается как вероятность $P(c_i|d_j)$ того, что документ $d_j = \langle w_{1j}, \ldots, w_{rj} \rangle$ попадает в категорию c_i, которая вычисляется по теореме Байеса:

$$P(c_i|d_j) = \frac{P(c_i)P(d_j|c_i)}{P(d_j)} \quad (1)$$

В этом выражении вероятности задаются на пространстве документов; $P(d_j)$ — вероятность того, что произвольно взятый документ можно представить в виде вектора d_j, $P(c_i)$ — вероятность того, что произвольно взятый документ относится к категории c_i.

Вычисление $P(d_j|c_i)$ в выражении (1) затруднительно, в следствие большого количества всевозможных векторов d_j (тоже самое относится и к $P(d_j)$, но дальше мы покажем, что она нам не помешает). Чтобы обойти эту проблему, предполагают, что любые две координаты вектора документа, рассматриваемые как случайные величины, статистически независимы друг от друга; данное *предположение о независимости* можно сформулировать как

$$P(d_j|c_i)=\prod_{k=1}^{r} P(w_{kj}|c_i) \quad (2)$$

Вероятностные классификаторы, использующие данное предположение, обычно называются «*Наивными» классификаторами Байеса*, которые являются самыми распространенными из вероятностных подходов к ТК. «Наивность» этих классификаторов обусловлена тем, что обычно в естественных корпусах документов предположение о независимости не выполняется.

Одним из наиболее известных «наивных» байесовских подходов является *бинарный независимый классификатор*, основанный на двоичном представлении вектора документа. В этом случае, если обозначить через p_{ki} вероятность $P(w_{ki}=1|c_i)$, то множители $P(w_{ki}|c_i)$ в выражении (2) примут вид:

$$P(w_{kj}|c_i)=p_{ki}^{w_{kj}}(1-p_{ki})^{1-w_{kj}}=\left(\frac{p_{ki}}{1-p_{ki}}\right)^{w_{kj}}(1-p_{ki}) \quad (3)$$

В дальнейшем будем считать, что пространство документов разбито на две категории, а именно, c_i и ее дополнение $\bar{c}_i$ такие, что $P(\bar{c}_i|d_j)=1-P(c_i|d_j)$. Если подставить выражения (2) и (3) в равенство (1) и прологарифмировать, получим:

$$\log P(c_i|d_j)=\log P(c_i)+\sum_{k=1}^{r} w_{kj}\log\frac{p_{ki}}{1-p_{ki}}+\sum_{k=1}^{r}\log(1-p_{ki})-\log P(d_j) \quad (4)$$

$$\log(1-P(c_i|d_j))=\log(1-P(c_i))+\sum_{k=1}^{r} w_{kj}\log\frac{p_{k\bar{i}}}{1-p_{k\bar{i}}}+\sum_{k=1}^{r}\log(1-p_{k\bar{i}})-\log P(d_j) \quad (5)$$

где $p_{k\bar{i}}$ обозначает $P(w_{ki}=1|\bar{c}_i)$. Сведем равенства (4), (5) к единому выражению, вычитая почленно из (4) равенство (5):

$$\log\frac{P(c_i|d_j)}{1-P(c_i|d_j)}=\log P\frac{(c_i)}{1-P(c_i)}+\sum_{k=1}^{r} w_{kj}\log\frac{p_{ki}(1-p_{k\bar{i}})}{p_{k\bar{i}}(1-p_{ki})}+\sum_{k=1}^{r}\log\frac{1-p_{ki}}{1-p_{k\bar{i}}} \quad (6)$$

Заметим, что $\frac{P(c_i|d_j)}{1-P(c_i|d_j)}$ - монотонно возрастающая функция от $P(c_i|d_j)$, поэтому может сама служить функцией принадлежности $CSV_i(d_j)$. Также заметим, что слагаемое

$\log P\frac{(c_i)}{1-P(c_i)}$ и слагаемое $\sum_{k=1}^{r}\log\frac{1-p_{ki}}{1-p_{k\bar{i}}}$ постоянны для всех документов и могут быть выпущены из рассмотрения. И при построении классификатора для категории c_i необходимо все лишь вычислить значения $2r$ параметров $\{p_{1i}, p_{1\bar{i}}, \ldots, p_{ri}, p_{r\bar{i}}\}$, что очень легко можно сделать на основе обучающих данных. Вероятностные классификаторы еще называют параметрическими классификаторами, так как в процессе их построения оцениваются вероятностные параметры обучающих данных. Заметим, что в общем случае для классификации документа не обязательно вычислять сумму r слагаемых, как в $\sum_{k=1}^{r} w_{kj}\log\frac{p_{ki}(1-p_{k\bar{i}})}{p_{k\bar{i}}(1-p_{ki})}$. Фактически, можно исключить все слагаемые, в которых $w_{kj}=0$, которые составят большинство в силу разреженности векторов документов.

В свою очередь, У. Купер [11] отмечал, что на самом деле в «наивном» байесовском классификаторе используется не предположение полной независимости, а более слабое *предположение связной зависимости*, которое можно записать следующим образом:

$$\frac{P(d_j|c_i)}{P(d_j|\bar{c}_i)}=\prod_{k=1}^{r}\frac{P(w_{kj}|c_i)}{P(w_{kj}|\bar{c}_i)}$$

Существует также множество других подходов к построению «наивного» байесовского классификатора, ключевая идея которых содержится в выражении (2). Основными направлениями развития данного метода являются:

- *ослабление ограничения на представление векторов в двоичном виде*. Довольно-таки естественно приписывать терминам веса, характеризующие их «важность» по отношению к данному документу.
- *нормализация размера документов*. Действительно, как видно из выражения (6), значение $\log\frac{P(c_i|d_j)}{1-P(c_i|d_j)}$ тем выше, чем больше документ (и тем больше в $w_{kj}=1$ его векторном представлении), вне зависимости от его отношения к c_i. Поэтому в вероятностных классификаторах различия размерах документов могут привести к снижению эффективности. Л. Бейкер и А. МакКаллум [3] предложили, в качестве одного из вариантов решения этой проблемы, рассматривать в качестве событий не документы, а термины. Если в первом случае мы имели дело с несколькими двоичными случайными величинами (соответствующими терминам), то во втором случае мы имеем дело с одной многомерной случайной величиной (соответ-

ствующей документу). При таком подходе размер документа учитывается естественным образом. Но в данном случае недостатком является предположение, что различные экземпляры одного и того же слова в тексте независимы. Это утверждение является еще более грубым, чем предположение независимости.

- *ослабление предположения независимости*. Это самый трудоемкий путь усовершенствования, поскольку он неизбежно влечет повышение вычислительной сложности и решение сложных задач оценок параметров. Ранние работы в этом направлении в рамках вероятностного информационного поиска не дали ожидаемого повышения эффективности. П. Домингос и М. Паццани [13] теоретически показали, что предположение бинарной независимости редко влияет на эффективность классификации.

Упоминание об информационном поиске в последнем абзаце не случайно. Работы по вероятностным классификаторам тесно переплетаются с исследованиями в вероятностном информационном поиске. Смысл его заключается в определении вероятности того, что документ относится к категории запроса, тем самым позволяя получать *релевантный* ответ, для чего очень часто используется обучение с учителем.

2.3.2 Правила и деревья принятия решений

Вероятностные индуктивные методы по существу имеют количественную (численную) природу, поэтому они довольно сложны для человеческого понимания, за что обычно и подвергаются критике. К алгоритмам, которые в меньшей степени подвержены таким упрекам, можно отнести символьные (не числовые) методы, среди которых нужно отметить классификаторы на основе правил и деревьев принятия решений.

Деревья принятия решений

Классификатор на основе дерева принятия решений представляет собой дерево, внутренние вершины которого помечены терминами, а исходящие из них ребра помечены проверочными весами. Вес термина в векторном представлении тестового документа сравнивается с этими проверочными весами. Листья данного дерева размечены значениями категорий (не обязательно попарно различными). Классификатор подобного типа последовательно проходит по вершинам дерева, сравнивая вес термина текущей вершины в представлении документа и определяя дальнейшее направление обхода, до тех пор пока не достигнет листа. Значение категории данного листа и присваивается документу. Большая часть такого рода классификаторов работает с бинарными векторами документов, и поэто-

му представляет из себя бинарные деревья.

Существует целый ряд стандартных пакетов для построения дерева принятия решения на основе обучающей выборки. Большинство приложений этого метода в ТК обычно используют ту или иную подобную библиотеку. Среди наиболее популярных можно назвать ID3 [17], C4.5 ([9], [10], [20], [27]) и C5 [31].

Одним из возможных алгоритмов построения дерева принятия решений для категории c_i с помощью обучающего множества может служить стратегия «разделяй и властвуй». На каждом шаге проверяется:

i. принадлежат ли все обучающие экземпляры к одной категории (либо c_i, либо $\bar{c}_i$);
ii. если нет, выбирается термин t_k, разбивающий обучающее множество на два класса, в которых вес t_k постоянный (0 или 1). Эти классы помещаются в разные поддеревья.

Этот процесс повторяется до тех пор, пока в каждом листе дерева все обучающие документы не будут принадлежать к одной категории, значение которой и присваивается данному листу. Ключевым моментом в этом процессе является определение подходящего термина t_k, по которому проходит разбиение. Выбор такого термина обычно производят, используя значение прироста информации или критерий энтропии. Однако, «разросшееся» таким образом дерево подвержено переобучению, поскольку некоторые ветви могут быть чрезмерно чувствительны к обучающим данным. Поэтому методы на основе деревьев принятия решений включают в себя не только алгоритм построения дерева, но и его усечения, т.е. удаления чрезмерно специфичных ветвей для более правильной классификации тестовых документов.

Проект AIR/X. Среди всех исследований, посвященных деревьям принятия решений, особое место занимает система AIR/X [17]. Эта система представляет собой результат одного из важнейших в истории ТК проектов, AIR/X. Продолжавшийся более десяти лет проект AIR/X создал систему для классификации коллекции научной литературы, состоящей из более чем миллиона документов. Также были получены важные теоретические результаты в области вероятностного индексирования.

Подход, использовавшийся в проекте AIR/X, известен как *Дармштадтский метод индексирования* (ДМИ). Для индексации использовались слова из управляемого словаря (в дальнейшем ДМИ был расширен для работы не только с управляемым словарем). ДМИ основан на предварительном вычислении *факторов ассоциированности*

$$z(t_k, c_i) = \frac{P(t_k, c_i)}{P(t_k)}$$

где $P(t_k, c_i)$ — количество обучающих документов, отнесенных к категории c_i и содержащих t_k, $P(t_k)$ — количество обучающих документов, содержащих t_k (во многих ранних работах по ТК факторы ассоциированности назывались *коэффициенты склеивания*). После вычисления данных факторов ассоциированности обучение дерева принятия решений проходило в два этапа. На *этапе описания* для каждого экземпляра t_{kj}^x термина t_k в документе $\bar{d}_j$ обновлялось относительное описание $rd(c_i, \bar{d}_j)$, используя при этом $z(t_k, c_i)$ и характеристики экземпляра t_{kj}^x (например, раздел $\bar{d}_j$, в котором встречается t_{kj}^x). На этапе принятия решений относительное описание $rd(c_i, \bar{d}_j)$ преобразовывалось в дискретный вектор $\vec{rd}(c_i, \bar{d}_j)$. Далее применялся алгоритм ID3, который для каждой координаты вектора (выбираемой по критерию χ^2) разбивал обучающие векторы на классы эквивалентности.

Классификация тестового документа d_l также проводилась в два этапа. Точность отнесения d_l к c_i определялась через процент обучающих векторов $\vec{rd}(c_i, \bar{d}_j)$, принадлежащих классу эквивалентности $\vec{rd}(c_i, d_l)$.

Правила принятия решений

Классификатор, построенный по методу правил принятия решений, состоит из дизъюнктивных нормальных форм (ДНФ), т.е. условных конструкций (утверждений), состоящих из посылки и заключения и соединенных логическими «И» и «ИЛИ». В посылке утверждается наличие или отсутствие термина в документе, а в заключении содержится решение о классификации документа по данной категории. Из теории машинного обучения известно, что методы на основе ДНФ эквивалентны методам на основе деревьев принятия решений. Однако, одним из преимуществ ДНФ является то, что данные классификаторы более компактны, по сравнению с деревьями.

Идея ДНФ-методов заключается в отборе из всевозможных покрывающих правил (правил, которые корректно классифицируют все обучающие документы) «наилучшее» правило с точки зрения некоторого критерия минимальности. В то время как деревья принятия решений обычно строятся сверху вниз с помощью стратегии «разделяй и властвуй», ДНФ-правила зачастую формируются снизу вверх. В начале индуктивного процесса построения классификатора для категории c_i каждый обучающий документ представляет собой утверждение $\tau_1, \ldots, \tau_n \Rightarrow \gamma_i$, где $\tau_1, \ldots, \tau_n$ - термины, содержащиеся в документе, а γ_i равняется либо c_i, либо $\bar{c}_i$. Этот набор утверждений уже является ДНФ-классифи-

катором для c_i, но очевидно, что в таком виде он ужасно подвержен эффекту переобучения. Поэтому процесс обучения включает в себя стадию генерализации, в ходе которой правила упрощаются, проходя через серию модификаций (сокращение посылок утверждений, слияние утверждений). Это делает правила более компактными и, в то же время, не нарушает свойства покрываемости классификатора. В завершении этого процесса, как и в случае с деревьями, выполняется стадия усечения, в результате которой повышается обобщающая способность классификатора.

Каждый обучающийся алгоритм на основе правил принятия решений может сильно отличаться в методах и критериях генерализации и усечения. Среди индуктивных обучающихся ДНФ-методов в области ТК можно назвать Charade [32], DL-ESC [30], Ripper ([9], [10], [8]), Scar [33] и Swap-1 [2].

Нужно отметить, что помимо упомянутых выше ДНФ-методов, работающих на уровне логики высказываний, также исследовались подходы на основе логики первого порядка с использованием *индуктивного логического программирования*. У. Кохен [8] проводил обширное сравнение обоих данных подходов в приложении к ТК (сравнивая Ripper, основанный на логике высказываний, и Flipper, основанный на логике первого порядка) и пришел к выводу, что мощный аппарат логики первого порядка дает довольно скромные преимущества.

2.3.3 Регрессивные модели

Некоторые подходы к ТК используют регрессивные модели ([19], [28], [38]). С точки зрения статистического обучения, *регрессия* — это аппроксимация действительнозначной функции f с помощью функции $\hat{f}$, удовлетворяющей обучающим данным. Примером такой модели может служить *метод наименьших квадратов* (МНК) [43]. В МНК каждому документу d_j соответствует два вектора: *исходный вектор* $I(d_j)$, т.е. стандартный вектор r весов терминов, и *результирующий вектор* $O(d_j)$, составленный из m весов, характеризующих принадлежность документа к категориям (в случае обучающего документа, $O(d_j)$ — бинарный вектор). Тогда задача классификации сводится к нахождению для тестового документа d_j результирующего вектора $O(d_j)$ по данному исходному вектору $I(d_j)$. То есть построение классификатора заключается в вычислении матрицы $\hat{M}_{m\times r}$ такой, что $I(d_j)\hat{M} = O(d_j)$. Данная матрица вычисляется с помощью МНК на основе обучающих данных, минимизируя ошибку по следующей формуле:

$$\hat{M} = arg\min_{M} \|MI - O\|_F$$

$\|V\|_F \stackrel{def}{=} \sum_{i=1}^{m}\sum_{j=1}^{n} v_{ij}$ - *норма Фробениуса* для матрицы размерами $m \times n$,

I — матрица $r \times g$, составленная из исходных векторов обучающих документов,

O — матрица $m \times g$, составленная из результирующих векторов. Матрица $\hat{M}$ обычно вычисляется с помощью сингулярного разложения векторов обучающего множества. В окончательной матрице $\hat{M}$ элементы $\hat{m}_{ik}$ характеризуют степень ассоциированности категории c_i и термина t_k.

Эксперименты показали, что МНК является одним из наиболее эффективных классификаторов, но, в то же время, вычислительная сложность при нахождении матрицы $\hat{M}$ значительно выше, чем у прочих методов ТК.

2.3.4 Линейные классификаторы

В линейном классификаторе категории представлены в виде векторов $c_i = \langle w_{1i}, \ldots, w_{ri} \rangle$ из того же r-мерного пространства, что и векторы документов. Тогда функцию принадлежности $CSV_i(d_j)$ можно вычислять на основе скалярного произведения вектора документа и вектора категории $\sum_{k=1}^{r} w_{ki} \cdot w_{kj}$. Хочется отметить, что в случае нормализованных векторов скалярное произведение сводится к нахождению косинуса угла между векторами:

$$S(c_i, d_j) = \cos(\alpha) = \frac{\sum_{k=1}^{r} w_{ki} \cdot w_{kj}}{\sqrt{\sum_{k=1}^{r} w_{ki}^2} \cdot \sqrt{\sum_{k=1}^{r} w_{kj}^2}}$$

Интересно то, что подобная же метрика используется в стандартных поисковых машинах при вычислении расстояния между запросом и документом. И, таким образом, если у нас есть построенный линейный классификатор, то можно проводить категоризацию, используя один из таких стандартных поисковых алгоритмов.

Линейные классификаторы также называют *профильными классификаторами*, так как их обучение заключается в построении *профиля* категории (идеального представителя) на основе обучающих данных. С точки зрения интерпретации подобные методы, очевид-

но, имеют преимущество, потому что такого рода профили категорий значительно проще для понимания, чем, скажем, конструкции в «наивном» байесовском подходе. Линейные классификаторы обычно разделяют на два обширных класса: групповые классификаторы и оперативные классификаторы.

Групповые методы конструируют классификатор, анализируя сразу всё обучающее множество. Примером такого подхода в ТК можно назвать *линейный дискриминантный анализ*, модерилующий стохастические зависимости между терминами на основе матриц ковариаций для различных категорий ([5], [18], [38]). Другим примером группового линейного классификатора может служить метод Роккио.

Оперативные (постепенные) методы начинают построение классификатора сразу с первого же обучающего вектора и постепенно модифицируют его по мере обхода всего обучающего множества. Такой подход может быть полезен в тех приложениях, в которых обучающее множество изначально недоступно в полном объеме, или в которых понятие категории может изменяться в ходе работы (например, при адаптивной фильтрации). Он также незаменим в тех приложениях, в которых пользователь может сам принимать решения о классификации, тем самым обучая систему в ходе эксплуатации. В качестве такого приложения можно назвать интерактивную классификацию [25].

Простым примером оперативного метода может служить алгоритм *персептрона* ([38], [41]). В начале алгоритма все координаты профильного вектора категории c_i принимают одно и то же положительное значение. При обработке очередного обучающего вектора $\bar{d}_j$ (в двоичном виде) классификатор сначала пытается определить категорию этого вектора самостоятельно, а затем сравнивает свой результат с истинным значением категории документа. При ошибочном результате координаты профильного вектора пересчитываются: если $\bar{d}_j$ относится к категории c_i, то веса w_{ki} всех «активных терминов» (для которых $w_{kj} = 1$ в векторе $\bar{d}_j$) увеличиваются на фиксированную константу $\alpha > 0$ (называемую степенью обучения), а если $\bar{d}_j$ не относится к категории c_i, все веса «активных терминов» уменьшаются на α. Отметим, что на определенном этапе, когда классификатор достигнет приемлемого уровня эффективности, можно исключить из рассмотрения все термины с очень малыми весами, поскольку они отрицательным образом влияют на классификацию. Таким образом, алгоритм персептрона (как и другие постепенные методы обучения) можно отнести к алгоритмам с автоматическим сокращением пространства терминов.

Приведенный выше алгоритм персептрона основывается на аддитивном обновлении весов. Мультипликативный вариант персептрона был реализован в алгоритме Positive Winnow [12], в котором увеличение и уменьшение весов получалось в результате умноже-

ния на константы $\alpha_1 > 0$ и $0 < \alpha_2 < 1$. Дальнейшее развитие эта идея получила в алгоритме Balanced Winnow ([12], [36]), где профиль категории имел две координаты (w_{ki}^{+} и w_{ki}^{-}) для каждого термина, а значение, используемое в скалярном произведении вычислялось как $w_{ki} = w_{ki}^{+} - w_{ki}^{-}$. В случае неправильной классификации положительного экземпляра категории c_i, веса w_{ki}^{+} увеличивались, а w_{ki}^{+} уменьшались; в противном случае уменьшались w_{ki}^{+} , а w_{ki}^{+} - увеличивались (по аналогии с Positive Winnow). Balanced Winnow допускал отрицательные значения для w_{ki}, в то время как в Positive Winnow все веса положительные.

Среди прочих примеров оперативных классификаторов нужно отметить алгоритм Widrow-Hoff, его усовершенствованный вариант Exponentiated Gradient [29] и Sleeping Experts ([10], [36]), как вариант Balanced Winnow. Основными отличительными чертами данных подходов от уже описанных являются:

i. пересчет весов в них происходит не только при ошибочной классификации;
ii. веса пересчитываются у всех координат, а не только у «активных терминов».

2.3.5 Классификатор Роккио

Классификатор Роккио основывается на формуле Роккио для определения значимости вектора в пространстве, адаптированной к ТК. Возможно, это единственный метод ТК, имеющий корни исключительно в области информационного поиска, а не машинного обучения. Такое применение было впервые предложено Д. Халлом [18]; позже метод Роккио использовался многими авторами и как объект собственных исследований, и как основной классификатор, и как член совокупности классификаторов.

Метод Роккио вычисляет идеальный представитель $(w_{1i}, \ldots, w_{ri})$ класса c_i по формуле

$$w_{ki} = \left(\frac{\beta}{|\{\bar{d}_j | ca_{ij} = 1\}|} \cdot \sum_{\{\bar{d}_j | ca_{ij} = 1\}} w_{kj} \right) - \left(\frac{\gamma}{|\{\bar{d}_j | ca_{ij} = 0\}|} \cdot \sum_{\{\bar{d}_j | ca_{ij} = 0\}} w_{kj} \right)$$

где w_{ki} вес, соответствующий термину t_k в документе $\bar{d}_j$. В этой формуле β и γ являются управляющими параметрами, позволяющими установить относительный вклад положительных и отрицательных экземпляров. Например, если β установлено в 1, а $\gamma = 0$, то идеальный представитель класса c_i можно рассматривать как центр масс его положительных обучающих векторов. В общем случае, классификатор Роккио компенсирует близость тестового документа к центру масс положительных обучающих экземпляров и удаленность

от центра масс отрицательных экземпляров. В большинстве случаев влияние отрицательных экземпляров снижают, придавая β большие значения, а γ – малые.

Этот метод достаточно легко реализовать, и такие классификаторы достаточно эффективны с вычислительной точки зрения. С точки зрения же эффективности классификации у данного метода есть один недостаток: если документы в одной категории принадлежат различным кластерам (например, часть документов категории Спорт относится и к боксу и к альпинизму), классификатор Роккио может большинство из них пропустить, поскольку центр масс этих документов не будет принадлежать этим кластерам (Рисунок 2.3, а). В более общем случае, классификатор Роккио, как все линейные классификаторы, имеет тот же недостаток, поскольку разбивает пространство на два подпространства; любой документ, попадающий в первое подпространство (или *n*-сферу, в случае Роккио), считается относящимся к категории c_i, а любой вектор, попадающий во второе подпространство, считается не относящимся к данной категории. Классификатор Роккио в сущности находит среднее значение (центроид) всех положительных экземпляров, а любое среднее значение лишь частично характеризует всё множество.

Усовершенствования метода Роккио. При использовании формулы Роккио возникает следующий вопрос: стоит ли рассматривать всё множество отрицательных экземпляров $\{\bar{d}_j \in Tr \mid ca_{ij} = 0\}$ или можно выбрать из них «хорошее» подмножество. Например, подмножество *почти положительных* экземпляров, определяемых как «наиболее положительные среди отрицательных обучающих экземпляров». В этом случае влияние второго слагаемого в формуле Роккио становится более значительным, поскольку именно почти положительные экземпляры бывает сложно отличить от подходящих документов. Почти положительные экземпляры использовал А. Сингал [39] в методе *локализации запроса*, предложенного для информационного поиска. Этот метод основывается на том наблюдении, что при использовании формулы Роккио почти положительные экземпляры используются чаще, чем чисто отрицательные, так как пользователь при вынесении решения о релевантности документов берёт во внимание только те, которые имеют наибольший ранг. В ранних приложения формулы Роккио в ТК не различались почти положительные и чисто отрицательные экземпляры, а уже Р. Шапир [37] использовал вариант запроса Роккио, противопоставляющего центр масс положительных обучающих экземпляров подмножеству отрицательных обучающих документов. Экземпляры этого подмножества, имеющие наивысший ранг, считались наиболее «близкими» к центру масс и могли быть использованы как почти положительные. В свою очередь, Н. Фур [16] определял почти положительные экземпляры категории c_i как положительные экземпляры категорий, родственных с c_i, по-

скольку сфера применения позволяла легко определить понятие «родственной категории c_i» (категоризация веб-документов в структуру иерархических каталогов).

Используя метод локализации запроса вместе с прочими усовершенствованиями (отбор терминов, статистическое выделение фраз, динамическая оптимизация результатов поиска), Шапир экспериментально показал, что классификатор Роккио способен достигать такой же эффективности, что и современные методы машинного обучения (например, «стимулирование»), при этом обучаясь в 60 раз быстрее. Эти последние результаты без сомнения послужили причиной вновь возросшего интереса к классификатору Роккио, который до этого являлся объектом нападок со стороны более изощренных обучающих методов.

2.3.6 Классификаторы на основе экземпляров

Классификаторы на основе экземпляров не пытаются найти явное декларативное представление требуемой категории, а «паразитируют» на решениях, принимаемых экспертом при категоризации обучающих документов, действуя подобным же образом на тестовых документах. Поэтому такие методы еще называют «ленивыми» обучающимися системами, так как они откладывают решение о классификации до тех пор, пока не будет рассмотрен каждый новый экземпляр.

Ярким примером подходов на основе экземпляров может служить алгоритм *k-NN* («k ближайших соседей»), реализованный И. Янгом [42] в системе ExpNet. Принимая решение об отнесении документа d_j к категории c_i, *k-NN* просматривает k наиболее похожих на $d_{j\ документов}$; если большинство из них относится к той же категории, принимается положительное решение.

Фактически, Янг использовал версию *k-NN* на основе расстояния, поскольку наиболее схожим документам из категории c_i приписывались веса, характеризующие их близость к тестовому документу. Математически, классификация документа с помощью *k-NN* сводится к вычислению

$$CSV_i(d_j) = \sum_{\bar{d}_z \in Tr_k(d_j)} RSV(d_j, \bar{d}_z) \cdot ca_{iz}$$

где $Tr_k(d_j)$ – множество k документов $\bar{d}_z$, на которых достигается максимум $RSV(d_j, \bar{d}_z)$, а ca_{iz} – значения из корректной матрицы принятия решений. В свою очередь, $RSV(d_j, \bar{d}_z)$ - это некая мера семантического сходства между тестовым докумен-

том d_j и обучающим документом $\bar{d}_z$; для этих целей может быть использована любая функция сходства, будь то вероятностная [25] или векторная мера из поисковой системы [42].

Метод *k-NN* может быть графически представлен следующим образом:

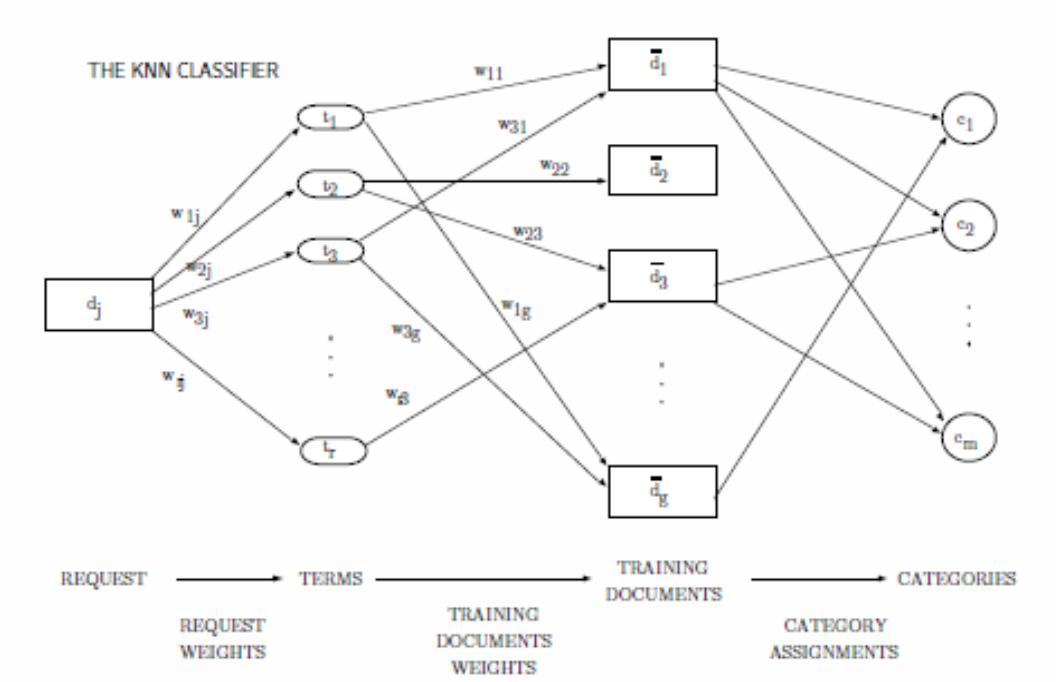


Рисунок 2.2

Из схемы видно, что данный подход естественным образом попадает под случай документно-ориентированной категоризации. Теоретически, для организации категориально-ориентированной классификации нужно лишь удалить дуги, идущие от всех обучающих документов к категориям $c_1,...,c_{i-1},c_{i+1},...,c_m$. Получившаяся сеть будет классификатором исключительно для категории c_i (в свою очередь, изначальная сеть может рассматриваться как система параллельно работающих классификаторов для всех категорий). Потребуется каждый документ пропускать через эту сеть. К сожалению, на практике это может быть очень неэффективно, поскольку всё обучающее множество должно быть обработано m раз (m – число категорий). Таким образом, документно-ориентированная категоризация – *de facto* – единственный разумный путь использования *k-NN*.

Построение *k-NN* классификатора также включает нахождение порогового значения k, определяющего количество обучающих документов, участвующих в вычислении $CSV_i(d_j)$. Этот порог обычно определяется экспериментально на тестовом множестве. Например, Ларки и Крофт [25] использовали $k = 20$, в то время как Янг [42] получил наибольшую эффективность при $30 \le k \le 45$. Так или иначе, различные эксперименты показали, что увеличение k снижает производительность незначительно.

Отметим, что в отличие от линейных классификаторов, *k-NN* не разбивает пространство документов на два подпространства, поэтому он лишен того недостатка, который был описан в методе Роккио. Эта ситуация графически показана на рисунке (Рисунок 2.3, b), с помощью которого можно оценить более «локальный» характер *k-NN* по отношению к Роккио.

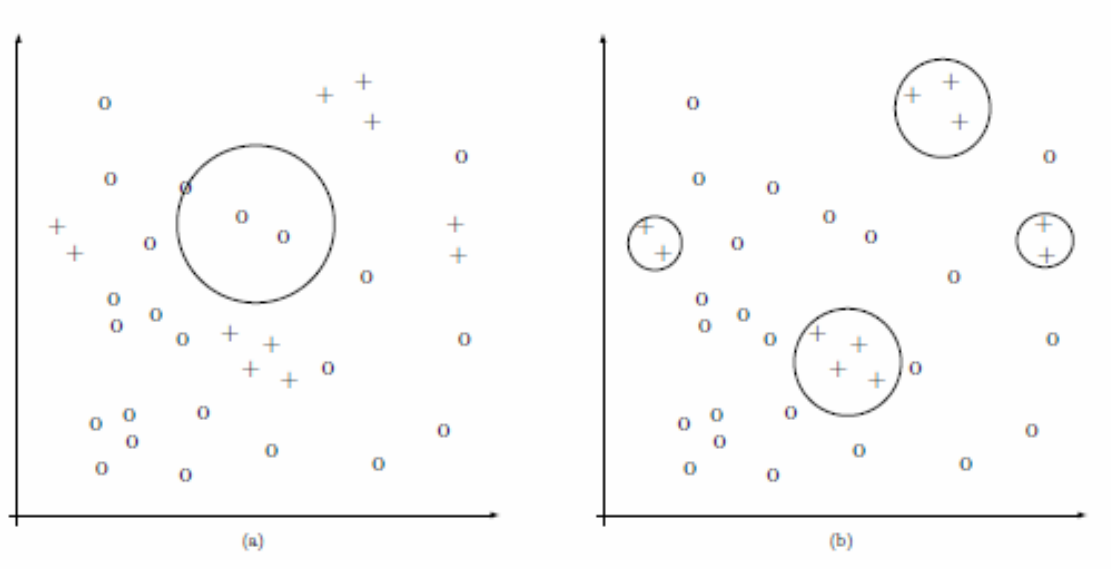


Рисунок 2.3

Помимо своей эффективности, проверенной в ряде различных экспериментов, одним из преимуществ *k-NN* является его производительность, так как классификация документа на m категорий может быть выполнена за линейное по мощности обучающего множества время. Тем не менее, можно отметить, что «ленивые» методы менее производительны, чем «активные», поскольку у них нет фазы обучения, и все вычисления производятся во время классификации.

Другие методики, основанные на экземплярах. В литературе встречаются различные методики на основе ближайших соседей.

У. Кохен и Х. Хирч [9] разработали классификатор на основе экземпляров, расширив стандартную технологию реляционных DBMS засчет «слабых объединений на основе сходства». В своей системе Whirl они использовали в качестве альтернативы стандартной формуле следующую оценочную функцию

$$CSV_i(d_j) = 1 - \prod_{\bar{d}_z \in Tr_k(d_j)} (1 - RSV(d_j, \bar{d}_z)) \cdot ca_{iz}$$

получив небольшие, но статистически значимые улучшения. Их эксперимент показал, что эта методика превосходит по производительности ряд классификаторов, таких как C4.5

(классификатор на основе дерева принятия решений) и Ripper DNF (классификатор на основе логических правил).

Интересный вариант обычного *k-NN* подхода предложил Л. Галавотти, который по новому взглянул на стандартную оценочную функцию, переопределив ca_{iz} как

$$ca_{iz}=\begin{Bmatrix} 1,\ \bar{d}_z\in c_i \\ -1,\ \bar{d}_z\notin c_i \end{Bmatrix}$$

Отличие от стандартного *k-NN* подхода заключается в том, что если обучающий документ $\bar{d}_z$, близкий к тестовому документу d_j, не принадлежит категории c_i, то эта информация учитывается как отрицательный факт, т.е. вносит негативный вклад в решение об отнесении d_j к c_i.

У. Лам и К. Хо представили комбинацию профильных и экземплярных классификаторов. В этой работе на вход алгоритма *k-NN* вместо обучающих документов подаются *обобщенные сущности*. Этот подход состоит из нескольких этапов:

- кластеризация обучающего множества, и получение множества кластеров $CL_i=\{cl_{i1},\ldots,cl_{ik_i}\}$;
- построение линейного классификатора $lc(cl_{iz})$ («обобщенная сущность») на документах, принадлежащих кластеру cl_{iz}, с помощью алгоритмов построения идеального экземпляра класса;
- применение *k-NN* к линейным классификаторам, а не к обучающим документам, т.е. вычисляя

$$\begin{aligned} CSV_i(d_j) &\stackrel{def}{=} \sum_{cl_{iz}\in CL_i} RSV(d_j, lc(cl_{iz}))\cdot\frac{|\{\bar{d}_j\in cl_{iz}|ca_{ij}=1\}|}{|\{\bar{d}_j\in cl_{iz}\}|}\cdot\frac{|\{\bar{d}_j\in cl_{iz}\}|}{|Tr|} \\ &= \sum_{cl_{iz}\in CL_i} RSV(d_j, lc(cl_{iz}))\cdot\frac{|\{\bar{d}_j\in cl_{iz}|ca_{ij}=1\}|}{|Tr|} \end{aligned}$$

где $\frac{|\{\bar{d}_j\in cl_{iz}|ca_{ij}=1\}|}{|\{\bar{d}_j\in cl_{iz}\}|}$ – степень вхождения обобщенной сущности $lc(cl_{iz})$ в категорию c_i,

$\frac{|\{\bar{d}_j\in cl_{iz}|ca_{ij}=1\}|}{|Tr|}$ – представляет её вклад в общий процесс. Такой метод обладает описанным преимуществом k-NN перед линейными классификаторами и в тоже время лишен восприимчивости k-NN к одиноко стоящим экземплярам, расположенным "далеко" от места скопления большинства экземпляров той же категории.

2.3.7 Метод опорных векторов

Главная идея этого метода заключается в том, чтобы отобразить наше первоначаль-

ное множество атрибутов в пространство признаков большой размерности, а затем построить в этом пространстве линейное классифицирующее правило («оптимальную гиперплоскость»). Таким образом, мы сможем найти максимальную границу между векторами двух классов. Векторы, задающие эту границу, называются «опорными векторами». Получается, что они могут заменить все обучающее множество экземпляров.

В общем случае рассматривается задача обучения с учителем: $\langle X, Y, \hat{y}, X^l \rangle$, где X — пространство векторов атрибутов, Y - множество категорий, $\hat{y}: X \to Y$ - целевая зависимость, значения которой известны только на объектах обучающей выборки $X^l = (x_i, y_j)_{i=1..l}$, $y_i = \hat{y}(x_i)$. Требуется построить алгоритм $a: X \to Y$, аппроксимирующий целевую зависимость на всём пространстве X. Обычно рассматривают задачу классификации на два непересекающихся класса, в которой документы описываются n-мерными вещественными векторами: $X = \mathbb{R}^n$, $Y = \{-1, +1\}$.

В результате требуется построить линейный пороговый классификатор:

$$a(x) = sign\left(\sum_{j=1}^{n} w_j x^j - w_0\right) = sign(\langle w, x \rangle - w_0) \quad (1)$$

где $x = (x^1, \ldots, x^n)$ признаковое описание документа x; вектор $w = (w^1, \ldots, w^n) \in \mathbb{R}^n$ и скалярный порог $w^0 \in \mathbb{R}^n$ являются параметрами алгоритма.

Уравнение $\langle w, x \rangle = w_0$ описывает гиперплоскость, разделяющую классы в пространстве $\mathbb{R}^n$.

Предположим, что выборка линейно разделима, то есть существуют такие значения параметров w, w_0, при которых функционал числа ошибок

$$Q(w, w_0) = \sum_{i=1}^{l} [\, y_i(\langle w, x_i \rangle - w_0) < 0 \,]$$

принимает нулевое значение. Но тогда разделяющая гиперплоскость не единственна, поскольку существуют и другие положения разделяющей гиперплоскости, реализующие то же самое разбиение выборки. Идея метода заключается в том, чтобы разумным образом распорядиться этой свободой выбора. Потребуем, чтобы разделяющая гиперплоскость максимально далеко отстояла от ближайших к ней точек обоих классов. Первоначально данный принцип классификации возник из эвристических соображений: вполне естественно полагать, что максимизация зазора между классами должна способствовать более уверенной классификации. В дальнейшем этот принцип получил мощное теоретическое обоснование.

Нормировка. Заметим, что параметры линейного порогового классификатора опре-

делены с точностью до нормировки: алгоритм $a(x)$ не изменится, если w и w_0 одновременно умножить на одну и ту же положительную константу. Удобно выбрать эту константу таким образом, чтобы для всех пограничных (т. е. ближайших к разделяющей гиперплоскости) объектов x_i из X^l выполнялись условия

$$\langle w, x_i \rangle - w_0 = y_i$$

Сделать это возможно, поскольку при оптимальном положении разделяющей гиперплоскости все пограничные векторы находятся от неё на одинаковом расстоянии. Остальные векторы находятся дальше. Таким образом, для всех $x_i \in X^l$

$$\langle w, x_i \rangle - w_0 \begin{cases} \leqslant -1, & y_i = -1 \\ \geqslant 1, & y_i = +1 \end{cases} \quad (2)$$

Условие $-1 < \langle w, x \rangle - w_0 < 1$ задаёт полосу, разделяющую классы. Ни одна из точек обучающей выборки не может лежать внутри этой полосы. Границами полосы служат две параллельные гиперплоскости с направляющим вектором w. Точки, ближайшие к разделяющей гиперплоскости, лежат в точности на границах полосы. При этом сама разделяющая гиперплоскость проходит ровно по середине полосы.

Ширина разделяющей полосы. Чтобы разделяющая гиперплоскость как можно дальше отстояла от точек выборки, ширина полосы должна быть максимальной. Пусть x_- и x_+ - две произвольные точки классов −1 и +1 соответственно, лежащие на границе полосы. Тогда ширина полосы есть

$$\left\langle (x_+ - x_-), \frac{w}{\|w\|} \right\rangle = \frac{\langle w, x_+ \rangle - \langle w, x_- \rangle}{\|w\|} = \frac{(w_0 + 1) - (w_0 - 1)}{\|w\|} = \frac{2}{\|w\|}$$

Ширина полосы максимальна, когда норма вектора w минимальна.

Итак, в случае, когда выборка линейно разделима, достаточно простые геометрические соображения приводят к следующей задаче: требуется найти такие значения параметров w и w_0, при которых норма вектора w минимальна при условии (2). Это задача квадратичного программирования.

Линейно разделимая выборка. Построение оптимальной разделяющей гиперплоскости сводится к минимизации квадратичной формы при l ограничениях-неравенствах вида (2) относительно $n + 1$ переменных w, w_0:

$$\begin{aligned} &\frac{1}{2}\langle w, w \rangle \to min \\ &y_i(\langle w, x_i \rangle - w_0) \geqslant 1, \; i = 1 \ldots l \end{aligned} \quad (3)$$

По теореме Куна-Таккера эта задача эквивалентна двойственной задаче поиска седловой

точки функции Лагранжа:

$$L(w, w_0; \lambda) = \frac{1}{2}\langle w, w\rangle - \sum_{i=1}^{l} \lambda_i (y_i(\langle w, x_i\rangle - w_0) - 1) \to \min_{w, w_0} \max_{\lambda}$$
$$\lambda_i \geqslant 0,\ i = 1 \ldots l$$
$$\lambda_i = 0,\ \textit{либо}\ \langle w, x_i\rangle - w_0 = y_i,\ i = 1 \ldots l$$

где $\lambda = (\lambda_1, \ldots, \lambda_l)$ - вектор двойственных переменных. Последнее из трёх условий называется *условием дополняющей нежёсткости*.

Необходимым условием седловой точки является равенство нулю производных Лагранжиана. Отсюда немедленно вытекают два полезных соотношения:

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{l} \lambda_i y_i x_i = 0 \ \Rightarrow\ w = \sum_{i=1}^{l} \lambda_i y_i x_i \quad (4)$$
$$\frac{\partial L}{\partial w_0} = -\sum_{i=1}^{l} \lambda_i y_i = 0 \ \Rightarrow\ \sum_{i=1}^{l} \lambda_i y_i = 0 \quad (5)$$

Из (4) следует, что искомый вектор весов w является линейной комбинацией векторов обучающей выборки, причём только тех, для которых $\lambda_i \neq 0$. Согласно условию дополняющей нежёсткости на этих векторах x_i ограничения-неравенства обращаются в равенства: $\langle w, x_i\rangle - w_0 = y_i$, следовательно, эти векторы находятся на границе разделяющей полосы. Все остальные векторы отстоят дальше от границы, для них $\lambda_i = 0$, и они не участвуют в сумме (4). Алгоритм (1) не изменился бы, если бы этих векторов вообще не было в обучающей выборке.

Определение. Если $\lambda_i > 0$ и $\langle w, x_i\rangle - w_0 = y_i$, то объект обучающей выборки x_i называется *опорным вектором*.

Подставляя (4) и (5) обратно в Лагранжиан, получим эквивалентную задачу квадратичного программирования, содержащую только двойственные переменные:

$$-L(\lambda) = -\sum_{i=1}^{l} \lambda_i + \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} \lambda_i \lambda_j y_i y_j (\langle x_i, x_j\rangle) \to \min_{\lambda}$$
$$\lambda_i \geqslant 0,\ i = 1 \ldots l \qquad (6)$$
$$\sum_{i=1}^{l} \lambda_i y_i = 0$$

Здесь минимизируется квадратичный функционал, имеющий неотрицательно определённую квадратичную форму и, следовательно, выпуклый. Область, определяемая ограничениями (неравенствами и одним равенством), также выпуклая. Следовательно, данная задача имеет единственное решение.

Допустим, мы решили эту задачу. Тогда вектор w вычисляется по формуле (4). Для определения порога w_0 достаточно взять произвольный опорный вектор x_i и выразит w_0 из

равенства $w_0 = \langle w, x_i \rangle - y_i$. На практике для повышения численной устойчивости рекомендуется брать в качестве w_0 среднее по всем опорным векторам, а ещё лучше медиану:

$$w_0 = med\{\langle w, x_i \rangle - y_i : \lambda_i > 0,\ i = 1 \ldots l\}$$

В итоге алгоритм классификации может быть записан в следующем виде:

$$a(x) = sign\left(\sum_{i=1}^{l} \lambda_i y_i \langle x_i, x \rangle - w_0\right)$$

Обратим внимание, что реально суммирование идёт не по всей выборке, а только по опорным векторам, для которых $\lambda_i \neq 0$. Именно это свойство разреженности отличает механизм опорных векторов от других линейных разделителей таких, как дискриминант Фишера, логистическая регрессия и однослойный персептрон.

Данный метод также применим в том случае, когда положительные и отрицательные экземпляры обучающей выборки линейно не разделимы. В этом случае целевую зависимость ищут в ином виде, используя вместо скалярного произведения функции-ядра из теории Гильберта-Шмидта. Также возможно добавление в Лагранжиан дополнительных «ослабляющих» неизвестных, в результате чего приходится решать более сложную задачу.

Использование этого подхода в текстовой категоризации было предложено в [20]. В качестве преимущества механизма опорных векторов называют отсутствие необходимости в отборе параметров, т.к. в случае линейно неразделимой выборки эффект переобучения удается легко обойти описанным выше способом.

2.4 Стилистическая категоризация

Существует множество различных способов классификации текстов. Тексты, описывающие «одно и то же», могут относиться к разным жанрам, с различными характеристиками и формой. Тексты делятся по различным категориям, и каждая такая классификация может быть важна для той или иной области применения, включая информационный поиск, разрешение многозначности и другие области искусственного интеллекта. Если рассматривать классификацию документов с этой точки зрения, то возникают следующие задачи: (i) определение жанров с определенной точностью и (ii) выбор критериев принадлежности текстов к одному жанру. Не следует путать эту задачу с задачей определения тематики текста и выбора параметров, позволяющих отличить одну тему от другой. Данные категориальные пространства если не ортогональны, то по крайней мере находятся в

разных измерениях. Естественно, между ними существует некоторая зависимость. Есть темы, которые могут быть описаны только в определённом стиле, также как некоторые жанры могут встретиться только в определенных темах. Но в большинстве случаев одна и та же тема может встретиться во всех жанрах.

Помимо описанной выше задачи тематической категоризации, отдельные исследования проводились в отношении стилистической классификации документов ([1], [6], [22]), и уже в них содержится первая попытка отойти от устоявшейся традиции, в рамках которой текст рассматривается как «набор слов». В качестве параметров использовались лексические, графические, морфологические, грамматические и синтаксические признаки. Например, в работе П. Браславского [1], после процедуры отбора параметров, набор характеристик составили: доля наречий, средняя длина слова, средняя длина предложения, доля местоимений первого лица, доля научных терминов, доля слов с научными приставками, доля слов-названий официальных документов. Для категоризация документов в работах [1], [22] использовался дискриминантный анализ и стандартные пакеты статистической обработки данных. По результатам данных исследований можно сказать, что самым «непростым» с точки зрения классификации является художественный стиль, так как на текстах именно этой группы процедуры дискриминантного анализа показывали наименьшую эффективность. В целом, эффективность работы классификатора в [1] достигала достаточно высоких уровней (82,28 % для художественного стиля).

3 Определение значений многозначных слов на основе контекста

При первых же попытках обработки естественного языка с помощью компьютера в 1950-х годах возникла проблема автоматического определения значения многозначных слов. Хотя данная задача является промежуточным этапом, но этот этап необходим для выполнения многих типов обработки естественного языка на различных уровнях. Очевидно, что данная задача является наиважнейшей для систем понимания текста, таких, как системы понимания сообщений, системы человеко-машинного взаимодействия и т. д.; и является, по крайней мере, полезной в тех приложениях, целью которых не является понимание естественного языка:

- *машинный перевод*: разрешение полисемии имеет очень важное значение для правильного перевода таких слов, как французское *grille*, которое, в зависимости от контекста, может быть переведено как *перила*, *ворота*, *прутoк*, *решетка*, *шкала*, *расписание* и проч.
- *информационный поиск и навигация по гипертексту*: в случае поиска по ключевым словам не плохо было бы отбросить те случаи, в которых слово или слова употреблены в неподходящем значении (Например, при поиске юридических ссылок, из всех документов со словом *court*, можно исключить исключить тексты, где данное слово имеет значение «королевский двор», а не с «суд»).
- *определение темы и содержания теста*: общий подход к анализу темы и содержания текста заключается в вычислении распределения определенных наборов слов (т.е. слов, отражающих определенную тему, идею, концепцию) на протяжении текста. Необходимость в снятии многозначности при таком анализе признается уже давно.
- *грамматический анализ*: разрешение омонимии полезно и при грамматической разметке. Например, во французском предложении *L'etagere plie sous les livres* (*Полка провисает под книгами*) необходимо разъяснить значение слова *livres*, которое в мужском роде означает «книги», а в женском «фунты». Снятие многозначности также необходимо при синтаксическом анализе, как, например, при сокращении числа возможных вариантов разбора предложения.
- *обработка речи*: снятие многозначности требуется и при автоматическом синтезе речи, а также для разбиения слов и разъяснения слов-омофонов при распознавании

речи.

- *обработка текста*: данная проблема актуальна при проверке правописания (определения расстановки диакритических знаков, выбора правильного регистра символа *He read the Times*, лексический анализ в семитских языках, где гласные не пишутся).

Задача снятия многозначности слов относится к *AI-полным* задачам, то есть может быть решена только после решения всех сложных задач искусственного интеллекта, таких как поиск универсального метода представления смысла и энциклопедических знаний. Характерная сложность снятия многозначности была центральным моментом в знаменитом трактате Бар-Хиллела по машинному переводу, в котором он утверждал, что не видит способов автоматического определения значения слова *pen* в предложении *The box is in the pen*. Данное заявление Бар-Хиллела для ALPAC послужило поводом к сворачиванию большинства исследований в области машинного перевода в начале 1960-х годов.

Однако, приблизительно в то же время был сделан значительный шаг в области представления знаний, особенно в виде семантических сетей, которые сразу были применены к разрешению многозначности слов. На протяжении следующих двух десятилетий работы в этой области велись в направлении интеллектуального понимания естественного языка. Этот же подход использовался и при анализе содержания текста, и при стилистическом и литературном анализе, и при информационном поиске. В последние 10 лет исследования в области автоматического разъяснения смыла слов (и подобных направлениях компьютерной лингвистики) стали набирать новую силу. Возрождение интереса к этой задаче обусловлено появлением больших коллекций электронных документов и разработкой статистических методов определения закономерностей в эмпирических данных. Хотя данные методы довольно успешно применяются в таких задачах, как распознавание речи и синхронизация переводов, задача разрешения многозначности занимает центральное положение и часто упоминается как одна из наиболее важных проблем обработки естественного языка на сегодняшний день.

В общих словах, *разрешение многозначности слов* заключается в сопоставлении данному слову в тексте или в речи определенного смысла, отличного от других значений, присущих этому слову. Таким образом эта задача обязательно включает:

(1) определение всевозможных значений для каждого значимого слова в рассматриваемом контексте;

(2) составление алгоритма, ставящего в соответствие каждому экземпляру слова подходящее значение.

Большинство работ по разрешению многозначности (РМ) опираются на то, что шаг (1) уже пройден, и значения слов заранее определены в виде:

- списка значений, подобных тем, что встречаются в обычных словарях;
- группы признаков, категорий или ассоциированных слов (синонимов, в случае тезауруса);
- записей в словарях перевода с одного языка на другой.

Однако, точное определение смысла является предметом немалых споров среди исследователей. Множество подходов к определению понятия «смысл», существующее в настоящее время, наряду со сложностью определения значений наводят на мысль о том, что какое-либо определенное решение этой проблемы в короткий срок не найти. Однако, с первых лет исследований РМ существует общая уверенность в том, что задачи разрешения морфологической, синтаксической и семантической омонимии можно разрешить. Например, для слов-омографов, принадлежащих различным частям речи (*play* как глагол и как существительное), разрешение морфо-синтаксической омонимии способствует снятию многозначности, и поэтому, впоследствии, исследования по РМ были сосредоточены на омографах из одной и той же синтаксической категории (особенно после разработки эффективных алгоритмов разметки частей речи).

Шаг (2), сопоставление слов и значений, проводится на основании двух основных источников информации:

- широкий *контекст* многозначного слова: он включает в себя не только информацию из текста или речи, где встречается слово, но и экстралингвистическую информацию (ситуацию и проч.);
- *внешние источники знаний*, в том числе лексические, энциклопедические, составленные вручную ресурсы, содержащие данные, полезные для сопоставления слов и значений.

Всевозможные способы РМ сводятся к сопоставлению контекста слова либо с информацией из внешних источников знаний (подход на основе *знаний*), либо с информацией из других контекстов с данным многозначным словом, взятых из корпусов текстов (подход на основе фактов или корпусов). Для нахождения соответствия между текущим контекстом и каким-нибудь из этих источников информации используется ряд методов оценки близости.

3.1 Первые исследования в области разрешения многозначности

С середины 1960-х годов большую популярность обрели методы на основе искусственного интеллекта. В числе данных подходов можно назвать методы построения семан-

тических сетей, в которых слова («токены») и понятия («типы») связаны дугами («семантическими отношениями»). Сложность составления такого рода источников знаний не позволила развиться этим системам дальше экспериментальных вариантов, которые охватывали лишь малую часть языка. Соответственно, процесс разрешения многозначности в них тестировался на очень малых объемах текста, что не позволяет говорить об их эффективности в реальных документах.

Методы на основе искусственного интеллекта, развиваемые в 1970-1980-х годах, были интересны для понимания естественного языка с теоретической точки зрения, но не с практической. С появлением обширных, широко доступных лексических ресурсов таких как словари, тезаурусы, корпусы способствовало выработке новых методик на основе баз знаний, составленных автоматически или вручную. Полученные в результате данных подходов машинные словари и тезаурусы обладали одной характерной особенностью: несмотря на то, что в них содержалось очень много информации о лексике слов, они слабо описывали взаимоотношения между понятиями. Использование семантических лексиконов для определения смысла многозначных слов развивалось по двум направлениям: перечислимые лексиконы, в которых явно описывались все значения слова, и генеративные лексиконы, в которых семантическая информация о конкретных значениях порождалась с помощью генерирующих правил. Наиболее известным перечислимым лексиконом для английского языка является *WordNet*, который и по сей день не потерял свою актуальность и используется в задачах разрешения многозначности.

3.2 Развитие, упадок и возрождение корпусных методов

Начиная с конца XIX века, благодаря ручному анализу корпусов началось исследование слов и графем, а также создание списков слов и словосочетаний для изучения способов овладения языком и обучения языку. Корпусы использовались лингвистами на протяжении первой половины XX века. Некоторые из этих работ касались значений слов, и зачастую были поразительно похожи на современные методы: например, Х. Палмер (1933) изучал словосочетания в английском языке; И. Лордж (1949) вычислял информацию о частотах значений 570 наиболее распространенных английских слов; Х. Итон (1940) сравнивала частоты значений в четырех языках; Торндайк (1948) и Ципф (1945) обнаружили, что существует положительная корреляция между частотой и количеством синонимов слова, которое характеризует его семантическое богатство (чем больше значений у слова, тем больше у него синонимов).

Корпус представляет собой банк примеров, на основе которого можно строить численные модели, т.е. использование корпусов тесно связано с эмпирическими методами. Хотя количественно-статистические метода встречались и в ранних работах по машинному переводу, в середине 1960-х годов интерес к статистической обработке языка среди лингвистов стал угасать. Произошло это в следствие новой тенденции в лингвистике, направленной на выявление формальных лингвистических правил, основанием к которой послужила теория Целлиг Харриса (1951), в особенности подкрепленная теорией трансформаций Ноама Чомски (1957). Большое внимание уделялось лингвистическому анализу: анализу предложений, а не текстов; анализу придуманных примеров и искусственно ограниченных моделей, а не естественного языка. В последующие 10-15 лет лишь немногие из лингвистов продолжали работать с корпусами, чаще в педагогических или лексикографических целях. Тем не менее, в данный период было создано несколько важных корпусов, таких как *Brown Corpus* (Kucera и Francis, 1967), *Tresor de la Langue Francaise* (Imbs, 1971), *Lancaster-Oslo-Bergen* (*LOB*) *Corpus* (Johansson, 1980) и т.д. По поводу обработки естественного языка в своем докладе ALPAC (1966) рекомендовала усилить исследования корпусов с целью создания богатых грамматических и лексических словарей, но из-за тенденции отхода от эмпиризма, продолжавшейся вплоть до 1980-х годов, в данном направлении было мало что сделано. Использование статистики для анализа языка было, в основном, характерно для литературной и гуманитарной численной обработки, информационного поиска и общественных наук. Работы над разрешением многозначности проводились и этих областях, например, в гарвардском проекте «РМ» для анализа содержания (Стоун, 1966), а также в работе Икера (1974, 1975), Чука и Дрейзина (1976), Чука и Голдберга (1979).

В свете ухода от использования корпусов и эмпирических методов особенно прогрессивными были работы Уэйсса [40], Келли и Стоуна [23] по автоматическому извлечению информации для РМ. Уэйсс продемонстрировал, что на основе размеченного вручную корпуса текстов можно обучить машину правилам снятия многозначности. Несмотря на малый масштаб его эксперимента (пять слов, обучающее множество по 20 предложений на слово, тестовое множество по 30 предложений на слово), результаты Уэйсса были довольно обнадеживающими (90% правильных значений). Исследование Келли и Стоуна, исходящее к гарвардскому проекту «РМ» для анализа содержания, проводилось в значительно больших масштабах: они извлекали устойчивые словосочетания для 1800 многозначных слов из корпуса в полмиллиона слов. Данные словосочетания служили фундаментом для ручного составления правил определения значения для каждого из 1800 слов. При тестировании данные правила (порой также довольно замысловатые) определяли информа-

цию о словосочетании, в которое входит тестовое слово, синтаксические связи с другими словами контекста и принадлежность к общим семантическим категориям. Данный подход показал лучшую эффективность, чем у Уэйсса, определив 92% правильных значений для случаев с ярко выраженными различиями.

В 1980-х годах интерес к методам на основе корпусов был возрожден. Технологические возможности позволяли создавать и хранить корпусы больших, чем прежде, размеров, что привело к разработке новых методов, зачастую использующих статистику. Эти методы были заново открыты при обработке речи, а затем применены к анализу рукописного текста.

В сфере разрешения многозначности слов Блэк [4] разработал модель на основе деревьев принятия решений. Он использовал корпус в 22 миллиона токенов, разметив вручную приблизительно 2000 словосочетаний для пяти тестовых слов. В дальнейшем *обучение с учителем* на основе размеченного корпуса использовалось многими исследователями.

Корпусные методы обучения с учителем

Корпусные методы разрешения многозначности — это класс алгоритмов построения классификатора на основе размеченного вручную набора текстов, использующих методики машинного обучения.

На первом этапе данных алгоритмов составляется множество примеров (результатов испытаний), иллюстрирующих различные возможные варианты классификации. Определяется набор параметров, характеризующих данные примеры или испытания. Далее на основе значений этих параметров выводятся правила, используемые в дальнейшей классификации новых экземпляров.

Таким образом, важными составляющими корпусных методов являются обучающая выборка текстов, словарь значений, синтаксический анализатор (парсер, леммер, стеммер и проч.). Обычно рассматривается одно многозначное слово в каждом тексте, причем одной части речи. Тем самым задача РМ сводится к задаче категоризации, где категории представлены различными значениями изучаемого слова.

«Наивный» байесовский классификатор. В данном методе предполагается *условная независимость* признаков друг от друга для данного значения слова. Хотя сама модель основывается на предположениях, параметры вычисляются по тестовым данным. Применительно к разрешению многозначности, признаками обычно являются термины, т.е.

контекст рассматривается как «набор слов». Обычно это тысячи двоичных признаков, обозначающих наличие или отсутствие слова в тексте.

$$sense = \underset{sense \in S}{argmax}\, P(F_1|S) \cdot \ldots \cdot P(F_n|S) \cdot P(S)$$

Пример:

- Если в 2,000 текстов со словом «bank», 1,500 содержат это слово в значении bank/1 («финансовая организация») и 500 в значении bank/2 («берег реки»), то вероятности будут принимать значения:
  - $P(S = 1) = 1{,}500/2000 = 0.75$
  - $P(S = 2) = 500/2{,}000 = 0.25$
- Если слово «credit» встречается 200 раз со значением bank/1 и 4 раза со значением bank/2, то:
  - $P(F_1 = \text{«credit»}) = 204/2000 = 0.102$
  - $P(F_1 = \text{«credit»} \mid S = 1) = 200/1{,}500 = 0.133$
  - $P(F_1 = \text{«credit»} \mid S = 2) = 4/500 = 0.008$
- Тогда для тестового документа, содержащего признак «credit» получим:
  - $P(S = 1 \mid F_1 = \text{«credit»}) = 0.133*0.75/0.102 = 0.978$
  - $P(S = 2 \mid F_1 = \text{«credit»}) = 0.008*0.25/0.102 = 0.020$

Правила и деревья принятия решений. Деревья принятия решений стали использоваться в области РМ довольно давно ([23], [4]). В данном подходе снятие многозначности представляется в виде последовательности условий (условий наличия признака), которые определяют значение слова. Набор параметров для данного метода меньше, чем в «наивном» байесовском классификаторе, и значительно проще для интерпретации.

Правила принятия решений для задач РМ использовал Яровский [45]. В качестве признаков он использовал словосочетания, т.е. изучаемое слово в совокупности с k словами, находящимися непосредственно справа или слева от него. Полученные правила сортировались согласно рангу, вычисляемому по формуле:

$$DLScore = \left| \log \frac{P(S = 1 | F_i = Collocation_i)}{P(S = 2 | F_i = Collocation_i)} \right|$$

Таким образом, наиболее показательные словосочетания для данного значения получали больший ранг. В вышеприведенном примере со словом «bank» ранг словосочетания «credit within bank» принимал бы значение:

$$DLScore = | \log (0.978 / 0.020) | = 3.89$$

При классификации алгоритм обходил список правил, начиная с наивысшего ранга.

Построение дерева принятия решений у Яровского [46] также основывалось на словосочетаниях. На каждом шаге определялось словосочетание, наиболее «четко» разделяющее обучающее множество документов в соответствии с различными смыслами изучаемого слова. «Четкость» определялась с помощью вычисления таких метрик, как прирост информации или отношение прироста информации. Этот процесс продолжался до тех пор, пока каждое подмножество обучающих документов не содержало исследуемое слово в одном значении.

Неполные методы обучения с учителем

Если методами обучения с учителем называют алгоритмы построения классификаторов значений на основе полностью размеченной обучающей выборки, то алгоритмы обучения на частично размеченной обучающей выборке (т.е. с минимальным участием эксперта) естественно называть неполными методами обучения с учителем.

В качестве примеров применения этих идей можно назвать следующие:

- автоматическая разметка корпуса, опирающаяся на небольшое количество документов, категоризованных экспертами;
- использование однозначных отношений или словарных определений для автоматической разметки данных;
- использование Web-агентов и активного обучения для разметки корпуса.

Развертывающиеся алгоритмы. Основной идеей данного подхода является построение классификатора на основе небольшого количества обучающих данных, что в значительно мере расширяет область применения корпусных методов РМ.

Принцип развертывающихся алгоритмов заключается в применении стандартного базового классификатора лишь к малой части обучающего множества примеров, категория которых уже определена экспертом. Значительная же часть обучающих данных не имеет подобных комментарием, и задача обучающего метода заключается в построении классификатора, более совершенного, чем базовый.

В начале работы алгоритма обучения задаются множество L размеченных обучающих экземпляров, множество U неразмеченных экземпляров, классификаторы C_i.

Алгоритм самообучения:

1. Выбор подмножества экземпляров U' (выбор P экземпляров из U, например, произвольным образом).
2. Для каждого i выполняется:
   a) обучение C_i на L и разметка U';
   b) выбор G наиболее правильных экземпляров из U' и добавление их в L (при этом сохраняя то же самое распределение в L).
   c) выбор новых экземпляров из U и добавление их в U' (сохраняя постоянный размер P для U').

Тем самым классификатор обучается на своих собственных результатах. Применяется такой метод в разметке частей речи [7], в определении взаимосвязей [35]. Основными параметрами алгоритма самообучения являются размер P выбираемого подмножества и величина приращения G. Отсутствие определенных методов подбора этих параметров является основным недостатком развертывающихся методов. В результате экспериментов, проводимых в рамках проекта SensEval-2, с различными комбинациями параметров данный подход показал максимальную эффективность 65%, при эффективности базового классификатора 53%.

Алгоритм Яровского. Данный алгоритм является разновидностью развертывающихся методов [47]. Он основывается на двух предположениях:

- в каждом словосочетании значение слова не меняется (рядом стоящие слова обуславливают устойчивость смысла изучаемого слова).
- значение одного и того же слова в документе не меняется (смысл изучаемого слова устойчив на протяжении одного документа).

Сам алгоритм использует в качестве базового классификатора правила принятия решений и на каждом шаге выполняет следующие действия:

1. составление правил принятия решений на основе небольшого обучающего множества (зачаточные словосочетания);
2. классификация остальных документов в корпусе;

3. построение нового обучающего множества, определяя экземпляры, размеченные с вероятностью, большей определенного порога;
4. обучение классификатора на новом обучающем множестве.

В первоначальной выборке значения изучаемого слова должны сильно различаться. Выбор первоначального обучающего множества может происходить различными способами: выбором различных определяющих словосочетаний для каждого смысла; используя словарные определения; разметка наиболее частых значений вручную. В экспериментах данный метод показал себя на одном уровне с обычным классификатором на основе правил принятия решений.

4 Многомерная категоризация

4.1 Идея многомерной классификации

Каждый классификатор — это функция, отображающая множество документов во множество категорий

$$f : D \to C$$

Поскольку каждый документ может быть охарактеризован с разных сторон, то можно проводить классификацию по нескольким направлениям. Каждое такое направление характеризуется своим набором категорий и набором параметров, по которым классифицируется документ. Вполне естественно использовать для каждого отдельного направления свой алгоритм построения классифицирующего правила.

Данные множества категорий можно использовать опять в качестве параметров, на которых будет проводиться категоризация более высокого уровня. Пример такой высокоуровневой категоризации можно встретить в словарях: значение слова зависит от значения вектора категорий контекста (тема, стиль, время, регион). Каждому множеству категорий соответствует свой набор параметров, наиболее подходящий для данной характеристики контекста, а значит и свое пространство признаков.

В смысловой (тематической) классификации текстов, как уже отмечалось, обычно в качестве параметров используются термины (*n*-граммы, слова, словосочетания). В классификации по стилям в большей степени применяются синтаксические, лексические и грамматические признаки.

Для каждого множества категорий естественно применять свою функцию индексации, отображающую набор признаков документа в векторно-числовое представление. Если D — множество всех документов, $d \in D$, тогда

$$\begin{gathered} I_1 : d \to d_1 \in D_1 \\ I_2 : d \to d_2 \in D_2 \\ \ldots \\ I_k : d \to d_k \in D_k \end{gathered}$$

где I_1, I_2, ..., I_k — отображения (индексирующие функции), переводящие документ d в векторные пространства признаков D_1, D_2, ..., D_k соответственно; d_1, d_2,..., d_k — векторные представления документа d в различных признаковых пространствах.

4.1.1 Случай I. Различные алгоритмы для каждого множества категорий.

Рассмотрим случай, когда для принятия решения о классификации для каждого множества категорий используется свой алгоритм:

$$f_1 : D_1 \to C_1$$
$$f_2 : D_2 \to C_2$$
$$\dots$$
$$f_k : D_k \to C_k$$

где $C_1, C_2, \dots, C_k$ - множества категорий, $D_1, D_2, \dots, D_k$ - соответствующие им векторные пространства признаков.

$$(f_1(d_1), f_2(d_2), \dots, f_k(d_k)) = (c_1, c_2, \dots, c_k) \in C_1 \times C_2 \times \dots \times C_k$$

где d_1, ..., d_k — представления документа в виде векторов, составленных из числовых значений признаков. Например, если C_i — множество всевозможных функциональных стилей текста, а набор признаков включает в себя среднюю длину слова, долю глаголов в неопределенной форме и т.д., тогда d_i — представление документа в виде вектора значений данных признаков.

$$I = (I_1, I_2, \dots, I_k)$$
$$F = (f_1, f_2, \dots, f_k)$$
$$d \overset{I}{\to} (d_1, d_2, \dots, d_k) \overset{F}{\to} (c_1, c_2, \dots, c_k)$$

Тем самым, каждому документу d был сопоставлен вектор категорий (c_1, c_2, ..., c_k), характеризующих его с различных точек зрения (форма, стиль, тема и т.д.). Множество всех таких векторов можно рассматривать как новое признаковое пространство, на основе которого происходит категоризация того же документа на более высоком уровне. Например, определение смысла изучаемого многозначного слова можно также рассматривать как задачу классификации, в которой из множества всех значений S исследуемого слова выбирается наиболее подходящее в зависимости от глобальных характеристик контекста.

$$g : C_1 \times C_2 \times \dots \times C_k \to S$$
$$g(c_1, c_2, \dots, c_k) = s \in S$$

Обучение и классификация, таким образом, будут состоять из двух ступеней.

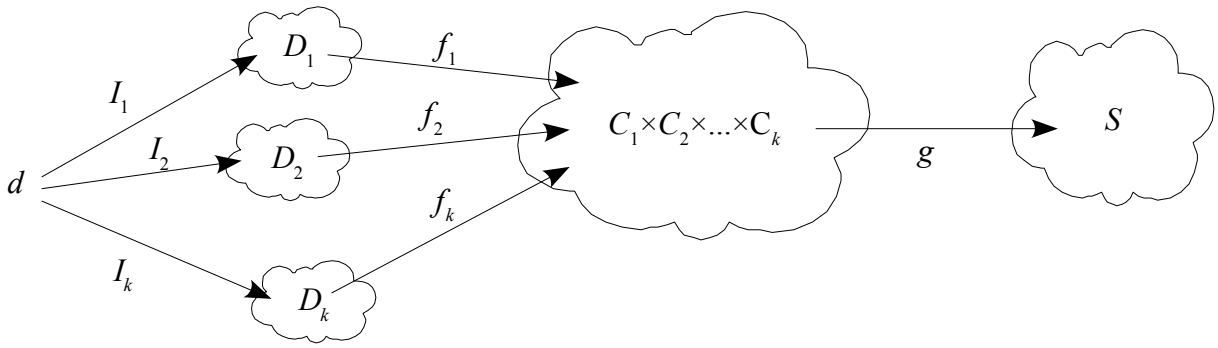
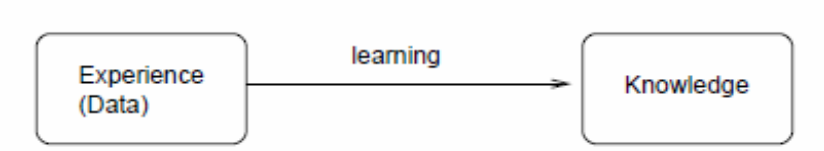


Рисунок 4.1

На первом этапе для каждого множества категорий отдельный алгоритм будет строить классифицирующее правило f_i. Совокупность данных решающих правил будет любому документу сопоставлять вектор его глобальных характеристик. На втором этапе (Рисунок 4.1), классификатор g определяет значение изучаемого многозначного слова, в зависимости от этих свойств документа.

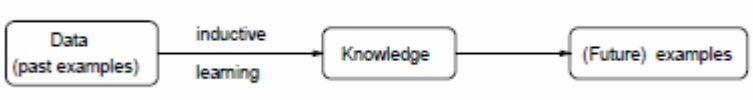
4.1.2 Трансдукция

Основная идея всех алгоритмов индуктивного построения классификаторов заключается в выведении общего приближенного «знания» (правила) из частных конкретных примеров. По подобной схеме построено любое индуктивное познание, а не только машинное обучение.



Знания, полученные в результате обучения, частично состоят из описаний явлений, которые мы уже пронаблюдали, и частично получены с помощью выводов, которые мы сделали из (прошлых) данных с целью предсказания природы (будущих) экземпляров.

Приобретенные в результате этого процесса «знания» в дальнейшем используются для описания (предсказания) будущих явлений. Таким образом, можно сказать, что мы используем «знания» чтобы предсказать «новые явления»:



«Знанием» в нашем случае является классификатор, индуцированный на обучающих данных. Построенное правило позволяет нам предопределять (положительные или отрицательные) классы новых сведений. Фактически мы можем предопределять и классифицировать даже бесконечное количество новых сведений. Однако в реальности нас интересует предвидение только конечного числа новых явлений.

Поэтому встает интересный вопрос: всегда ли нам нужно переходить от частного (прошедшие события - сведения) к общему (знания) – что обычно называется индукцией, а затем обратно к частным (будущим) сведениям – что обычно называется дедукцией? Не можем ли мы «срезать путь» и перейти сразу от частного к частному? Будем называть этот короткий путь «трансдукцией» (Рисунок 4.2). Похоже, что таким образом мы в некотором смысле «сэкономим» - вместо «общего знания» гораздо более эффективней получить «специализированное знание» о частных случаях. Итак, трансдукция – есть переход от частных (прошлых) сведений к частным (будущим) сведениям без каких-либо попыток «обобщить» наш опыт и получить «общее знание».

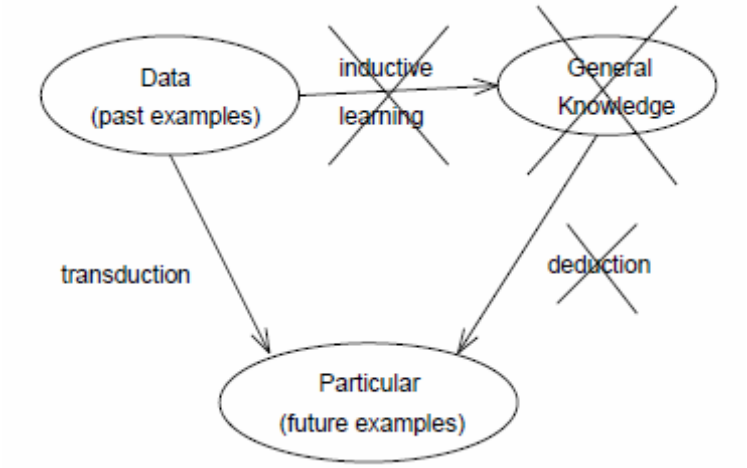


Рисунок 4.2

Хорошей иллюстрацией этой идеи могут служить классификаторы на основе экземпляров, в которых решение о классификации принимается непосредственно из обучающих данных. Покажем, как принцип трансдукции нашел свое применение в многомерной категоризации при определении смысла многозначных слов.

4.1.3 Случай II. Единый алгоритм для каждого множества категорий.

Пусть f — классифицирующее (решающее) правило, сопоставляющее каждому вектору d_i из пространства признаков D_i значение категории (степень принадлежности) $\tilde{c}_i$ из множества категорий C_i.

$$\begin{array}{l} f: d_1 \to \tilde{c}_1, \;\; d_1 \in D_1, \;\; \tilde{c}_1 \in C_1 \\ f: d_2 \to \tilde{c}_2, \;\; d_2 \in D_2, \;\; \tilde{c}_2 \in C_2 \\ \qquad\qquad ... \\ f: d_k \to \tilde{c}_k, \;\; d_k \in D_k, \;\; \tilde{c}_k \in C_k \end{array}$$

Данным решающим правилом может быть один из методов машинного обучения (распознавания образов) на основе прецедентов. Поскольку алгоритм f — общий для всех пространств признаков, то можно записать:

$$\begin{array}{l} f: D_1 \times D_2 \times ... \times D_k \to C_1 \times C_2 \times ... \times C_k \\ f(d_1, d_2, ..., d_k) = (c_1, c_2, ..., c_k) \in C_1 \times C_2 \times ... \times C_k \end{array}$$

Таким образом, мы каждому документу поставили в соответствие кортеж разнотипных категорий.

$$\begin{array}{l} I = (I_1, I_2, ..., I_k) \\ d \;\; \stackrel{I}{\to} \;\; (d_1, d_2, ..., d_k) \;\; \stackrel{f}{\to} \;\; (c_1, c_2, ..., c_k) \end{array}$$

Воспользуемся полученными кортежами как новыми признаками для определения значения многозначного слова. Новыми категориями будут служить значения слова, а признаками — значения категорий, к которым был отнесен контекст (документ) в процессе многомерной категоризации.

$$\begin{array}{l} g: C_1 \times C_2 \times ... \times C_k \to S \\ g(c_1, c_2, ..., c_k) = s \in S \end{array}$$

где S — множество значений интересующего слова, g — отображение, реализующее зависимость значения слова от категорий контекста.

Диаграмма-схема многомерной классификации:

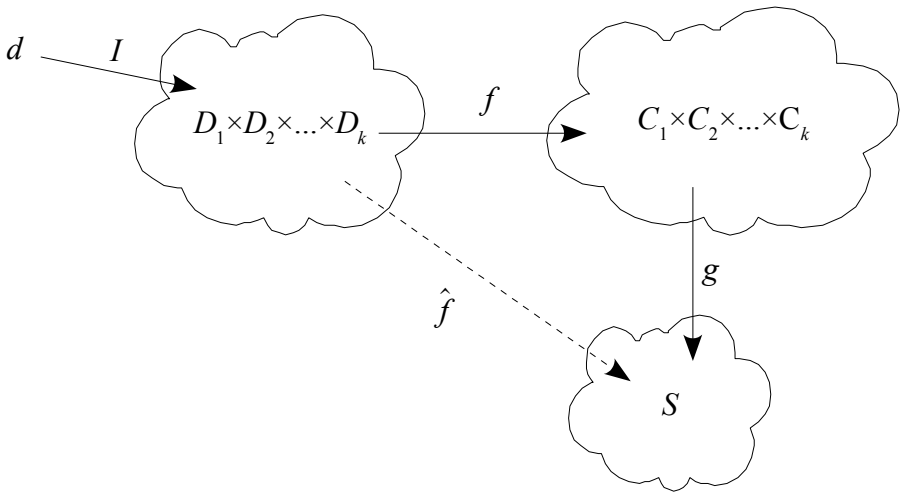


Рисунок 4.3

В результате обучения будет построено новое единственное классифицирующее правило $\hat{f} = f \circ g$ (Рисунок 4.3), которое будет отображать прямую сумму пространств признаков во множество значений слова. Именно оно и будет использоваться при дальнейшей классификации.

Таким образом, при обучении классификатора для категории $s \in S$ нам не потребуется переходить к глобальным характеристикам контекста. Мы сможем на основе пространств признаков D_i сразу относить документ к определенному значению исследуемого слова, в чем и заключается идея трансдукции.

4.2 Описание эксперимента

Данный принцип многомерной категоризации был реализован в приложении, включающем в себя следующие компоненты:

- модули индексации;
- модуль классификации;
- модуль оценки эффективности.

Сам эксперимент проходил в 4 этапа:

1. тематическая классификация;
2. стилистическая классификация;
3. определение значения слова в тестовом документе;
4. вычисление параметров оценки эффективности.

В свою очередь, тематическая и стилистическая классификации заключали в себе этап обработки обучающих документов и этап обработки тестовых документов.

Модуль индексации

Для классификации по темам и классификации по стилям использовались различные методики индексирования. Стилистические особенности теста традиционно связывают с синтаксическими, грамматическими и морфологическими признаками, в то время как семантику текста обычно определяют по набору слов (словосочетаний), входящих в него.

Параметры стилевой классификации:

- средняя длина предложения в словах
- средняя длина слова в символах
- доля длинных слов (больше 6 символов)
- доля местоимений 3 лица
- доля слова «that»
- доля слова «which»
- доля слов с апострофами
- доля ing-форм

Для тематической классификации параметрами индексирования служили термины, встречающиеся в документе. Вес термина вычислялся по методу *tf***idf*. После вычисления обратной документной частоты проводился отбор параметров (сохранялись только те термины, у которых $A \leq idf < B$). Это позволило сократить размерность пространства векторов с 9600 до 90, повысить скорость обработки векторов при обучении и классификации. Значения A, B, при которых достигалась наибольшая эффективность, были найдены экспериментально.

Обработка обучающего множества

Для тематической классификации использовалась обучающая выборка с 240 документами, а для категоризации по стилям обучение проводилось на 150 документах. Терминологическая индексация состояла из следующих этапов:

- подсчет слов в каждом документе, вычисление документной частоты (*tf*);
- вычисление обратной документной частоты (*idf*) для каждого слова из обучающей коллекции;
- сокращение размерности пространства терминов (Наибольшая эффективность была достигнута при $1 \leq idf < 2$. В результате осталось 90 терминов, каждый из которых встречался в среднем в половине обучающих документов, т.е. имел большое значение для классификации);

- вычисление для каждого документа *tf*idf* и составление векторов.

В случае со стилистической индексацией вычисление параметров и составление векторов проходило за один цикл обхода обучающей выборки.

Модуль классификации

Набор категорий для тематической категоризации был следующий: юриспруденция, военное дело, политика, телекоммуникации, развлечения, экономика, строительство, образование. Стилистическая классификация проходила по пяти функциональным стилям: литературный, официально-деловой, публицистический, научный, разговорный.

В качестве метода классификации использовался алгоритм *k-NN*. Параметр $k = 6$, при котором достигалась максимальная эффективность был найден опытным путем. Алгоритм классификации заключался в следующем:

1. для пробного вектора рассчитывались расстояния в евклидовой метрике до каждого обучающего вектора;
2. выбирались k ближайших к пробному векторов;
3. среди этих k векторов выбиралась наибольшая группа, представляющая одну категорию. (Если этих групп оказывалось несколько, выбиралась та, центроид которой был ближе к пробному вектору. Данный метод также носит название метода Роккио);
4. категория выбранной группы и присваивалась пробному вектору.

Метод оценки эффективности категоризации

Как и в случае поисковых систем, оценка эффективности классификаторов документов скорее носит экспериментальный характер, чем аналитический. Причиной тому — неформализованность и субъективность задачи ТК. Поэтому при экспериментальной оценке классификаторов обычно определяют не сложность алгоритма классификатора, а его эффективность, то есть способность правильно распределять документы по категориям.

Для оценки эффективности построенного классификатора вычисляются следующие статистические величины:

- *Точность* (*precision*) — вероятность того, что произвольный документ d_x среди всех тестовых документов, отнесенных классификатором к категории c_i, действительно

относится к этой категории. $P(ca_{ix} = 1 \mid a_{ix} = 1)$.

- *Полнота* (*recall*) - вероятность того, что произвольный документ d_x среди всех тестовых документов, относящихся к категории c_i, отнесен классификатором именно к категории c_i. $P(a_{ix} = 1 \mid ca_{ix} = 1)$.

Данные вероятности можно оценить статистически на тестовом множестве:

$$\hat{Pr}_i = \frac{TP_i}{TP_i + FP_i}$$

$$\hat{Re}_i = \frac{TP_i}{TP_i + FN_i}$$

где FP_i — неправильные положительные экземпляры (количество тестовых документов, неправильно отнесенных к c_i), TN_i — правильные отрицательные, TP_i — правильные положительные, FN_i — неправильные отрицательные экземпляры.

Для получения величин точности и полноты на всем множестве категорий применяются следующие характеристики:

1. Микроусреднение

$$\hat{Pr}^{\mu} = \frac{TP}{TP + FP} = \frac{\sum_{i=1}^{m} TP_i}{\sum_{i=1}^{m} (TP_i + FP_i)}$$

$$\hat{Re}^{\mu} = \frac{TP}{TP + FN} = \frac{\sum_{i=1}^{m} TP_i}{\sum_{i=1}^{m} (TP_i + FN_i)}$$

2. Макроусреднение

$$\hat{Pr}^{M} = \frac{\sum_{i=1}^{m} \hat{Pr}_i}{m}$$

$$\hat{Re}^{M} = \frac{\sum_{i=1}^{m} \hat{Re}_i}{m}$$

Эти две величины могут принимать довольно различные значения, особенно если категории представлены неравномерно. Например, если классификатор лучше всего работает только на категориях с большой долей положительных тестовых экземпляров, то, скорее всего, значение микроусреднения будет больше макроусреднения. Макроусреднение более устойчиво к неравномерному тестовых распределению документов по категориям, в то время как на микроусреднение большее влияние будут оказывать наиболее представительные категории. Большинство исследователей склоняются к мысли, что такая неравномер-

ность распределения категорий в тестовом множестве более естественна и более подходит для реальных приложений.

Общую эффективность классификатора определяют с помощью функции F_β для $0 \leqslant \beta < +\infty$:

$$F_\beta = \frac{(\beta^2 + 1) \cdot Pr \cdot Re}{\beta^2 \cdot Pr + Re}$$

В этой формуле β можно рассматривать как уровень соотношения *Pr* и *Re*. Если $\beta = 0$, $F_\beta = Pr$, в то время как при $\beta \to +\infty$, $F_\beta \to Re$. Обычно используется значение $\beta = 1$, соответствующее компромиссному варианту между *Pr* и *Re*.

Для тестирования эффективности реализации метода многомерной категоризации использовалась тестовая выборка SenseEval с тестами на английском языке. Эксперимент проводился над двумя словами: «paper» и «party». Каждому из них в тестовом наборе соответствовало несколько текстов, в которых данные слова были употреблены в различных значения. Таблица соответствия кортежей категорий контекста и значения слова была составлена экспертом на основе толкового словаря и смысловых значений, встречающихся в тестовых документах. Объем тестового множества составлял 240 документов для каждого многозначного слова.

Индексирование тестового множества проводилось по тем же методикам, что и для соответствующих обучающих документов. По результатам работы алгоритма классификации *k- NN*, для каждого документа составлялось отношение (стиль, тема), по которому определялось значение тестового слова, исходя из составленной заранее таблицы соответствия.

Эффективность классификации оценивалась по параметрам *Pr* и *Re* для каждого смыслового значения тестового слова. Другими словами, оценивалась точность совпадения значения, присвоенного классификатором, со предопределенным значением из корпуса SenseEval. В результате вычисления микроусреднения *Pr* и *Re* и нормы F_1, были получены следующие величины: для слова «party» $Pr \approx Re \approx F_1 \approx 0{,}77$; для слова «paper» $Pr \approx Re \approx F_1 \approx 0{,}44$. Причинами такой довольно скромной эффективности скорее всего явились сравнительно малый объем обучающей выборки, малое количество множеств категорий, использование простых методов классификации. Увеличение числа множеств категорий и применение более сложных метрик пространства признаков являются предметом дальнейших исследований.

5 Заключение

В работе были рассмотрены некоторые алгоритмы автоматической текстовой категоризации, основанные на методах распознавания образов и машинного обучения, а также различные способы индексации документов и функции сокращения размерности пространства признаков. Особое внимание было уделено одной из прикладных задач ТК — определению значений многозначных слов на основе контекста и особенностям применения описанных алгоритмов к данной проблеме.

Применительно к разрешению многозначности, была предложена идея многомерной классификации текстов, позволяющая охарактеризовать контекст с различных точек зрения. В результате такой одновременной классификации текста по нескольким независимым направлениям каждому документу ставится в соответствие кортеж из категорий. Данная идея была использована в экспериментальном приложении, реализующем алгоритмы тематической и стилистической классификации документов. Метод многомерной категоризации был применен для определения значения многозначных английских слов, где каждому кортежу категорий контекста соответствовало одно значение изучаемого слова. Предложенный подход к разрешению многозначности согласуется в значительной степени с теми тенденциями, которые наметились на данный момент как в области текстовой категоризации, так и, в частности, в сфере автоматического определения смысла многозначных слов.

В частности, в последнее время возрос интерес исследователей к использованию совокупностей классификаторов и комбинированию нескольких разнородных методов. Рассматриваются всевозможные варианты функции комбинирования классификаторов: по методу большинства, линейно-весовое комбинирование, динамический метод отбора, адаптационный метод отбора. Помимо применения совокупностей разнородных алгоритмов построения классификаторов, отдельные исследования ведутся в направлении «стимулирующих» алгоритмов обучения. Смысл данных алгоритмов заключается в пошаговом использовании одного и того же алгоритма обучения к данному обучающему множеству, с постепенным увеличением качества индуцируемого классификатора. Тем не менее, это направление еще требует дальнейших исследований.

В работе используется иная концепция понимания документа, отличная от представления контекста в виде «множества слов». Не смотря на то, что модель документа в виде «набора слов» все еще остается наиболее распространенным текстовым представле-

нием, исследователи считают, что текст может представлять собой нечто большее, чем просто набор терминов. Поэтому возникла необходимость в поиске лучшего представления для документа, и более сложных моделей.

Еще одним направлением исследований является изучение расширяемости систем текстовой классификации, т.е. в определении, будет ли система, которая показала наибольшую эффективность, оставаться такой же эффективной в случае очень большого количества категорий (например, десятки тысяч).

Последним, но не менее интересным направлением исследований является попытка решения проблемы разметки обучающего множества документов, поскольку определение категории для каждого документа из неразмеченного обучающего множества может быть очень трудоемким процессом. В результате пристальное внимание исследователей ТК привлекли к себе неполные методы машинного обучения с учителем. Данные методы могут обучаться на небольшом множестве помеченных экземпляров и переходить к еще непомеченным.

Успехи в автоматической ТК способствовали распространению ее методов и приемов также и на соседние предметные области. Методики, присущие автоматической ТК, успешно были использованы в классификации документов, представленных в несколько иной форме, например, классификация очень «зашумленных» текстов, полученных в результате оптического распознавания или классификация расшифровок речи. Таким образом, достигнутые на сегодняшний день успехи и наличие дальнейших путей развития говорят о том, что автоматическая текстовая категоризация остается перспективной областью исследований, нашедшей свое применение в решении целого ряда практических задач, и что интерес к данной области не угаснет до тех пор, пока будет актуальная машинная обработка естественного языка.

Список литературы

[1] Браславский П. И., Методы повышения эффективности поиска научной информации (на материале Internet). Екатеринбург, Уральский государственный технический университет, 2000

[2] Apte, C., Damerau, F. J., Weiss, S. M., Automated learning of decision rules for text categorization. ACM Transactions on Information Systems 12, 3, 233–251., 1994

[3] Baker, L. D., McCallum, A. K., Distributional clustering of words for text categorisation. In Proceedings of SIGIR-98, 21st ACM International Conference on Researchand Development in Information Retrieval (Melbourne, AU, 1998), pp. 96–103., 1998

[4] Black, Ezra, An Experiment in Computational Discrimination of English Word Senses, IBM Journal of Research and Development,32(2), 185-194., 1988

[5] Blosseville, M., Hebrail, G., Montell, M., Penot, N., Automatic document classification: natural langage processing and expert system techniques used together. In Proceedings of SIGIR-92, 15th ACM International Conference on Research and Developmentin Information Retrieval (Kobenhavn, DK, 1992), pp. 51–57., 1992

[6] Braslavski P., Combining Relevance and Genre-Related Rankings: an Exploratory Study. In Proc. of the International Workshop "Towards Genre-Enabled Search Engines: The Impact of NLP", Georg Rehm, Marina Santini (eds.), Borovets, Bulgaria, 2007

[7] Clark, S, Curran, J.R. and Osborne, M. , Bootstrapping POS taggers using unlabelled data. Proceedings of CoNLL , 2003

[8] Cohen W.W., Learning to classify English text with ILP methods. In L. De Raedt Ed., Advances in inductive logic programming. Amsterdam, NL: IOS Press., 1995

[9] Cohen, W. W., Hirsch, H., Joins that generalize: text classification using Whirl.In Proceedings of KDD-98, 4th International Conference on Knowledge Discovery and Data Mining (New York, US, 1998), pp. 169–173., 1998

[10] Cohen, W. W, Singer, Y., Context-sensitive learning methods for text categorization. ACM Transactions on Information Systems 17, 2, 141–173., 1999

[11] Cooper, W. S., Some inconsistencies and misnomers in probabilistic information retrieval. ACM Transactions on Information Systems 13, 1, 100–111., 1995

[12] Dagan, I., Karov, Y., Roth, D., Mistake-driven learning in text categorization. In Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing (Providence, US, 1997), pp. 55–63., 1997

[13] Domingos, P., Pazzani, M. J., On the the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning 29, 2-3, 103–130., 1997

[14] Dumais, S. T., Platt, J., Heckerman, D., Sahami, M., Inductive learning algorithms and representations for text categorization. In Proceedings of CIKM-98, 7th ACM International Conference on Information and Knowledge Management (Washington, US, 1998), pp. 148–155., 1998

[15] Fuhr, N., Buckley, C., A probabilistic learning approach for document indexing. ACM Transactions on Information Systems 9, 3, 223–248., 1991

[16] Fuhr, N., Govert, N., Lalmas, M., and Sebastiani, F., Categorisation tool: Final prototype. Deliverable 4.3, Project LE4-8303 “EUROSEARCH”, Commission of the European Communities, 1998

[17] Fuhr, N., Hartmann, S., Knorz, G., Lustig, G., Schwantner, M., Tzeras, K., AIR/X – a rule-based multistage indexing system for large subject fields. In Proceedings of RIAO-91, 3rd International Conference “Recherche d’Information Assistee par Ordinateur” (Barcelona, ES, 1991), pp. 606–623., 1991

[18] Hull, D. A., Improving text retrieval for the routing problem using latent semantic indexing. In Proceedings of SIGIR-94, 17th ACM International Conference on Researchand Development in Information Retrieval (Dublin, IE, 1994), pp. 282–289., 1994

[19] Ittner, D. J., Lewis, D. D., Ahn, D. D., Text categorization of low quality images. In Proceedings of SDAIR-95, 4th Annual Symposium on Document Analysis and Information Retrieval (Las Vegas, US, 1995), pp. 301–315., 1995

[20] Joachims, T., Text categorization with support vector machines: learning with many relevant features. In Proceedings of ECML-98, 10th European Conference on MachineLearning (Chemnitz, DE, 1998), pp. 137–142., 1998

[21] John, G., Kohavi, R., Pfleger, K., Irrelevant features and the subset selection problem. In Proceedings of ICML-94, 11th International Conference on Machine Learning(New Brunswick, US, 1994), pp. 121–129., 1994

[22] Karlgren J., Cutting D., Recognizing Text Genres with Simple Metrics Using Discriminant

Analysis, 2005

[23] Kelly Edward F., Stone Philip J., Computer Recognition of English Word Senses, North-Holland, Amsterdam., 1975

[24] Larkey, L. S., A patent search and classification system. In Proceedings of DL-99,4th ACM Conference on Digital Libraries (Berkeley, US, 1999), pp. 179–187., 1999

[25] Larkey, L. S., Croft, W. B., Combining classifiers in text categorization. In Proceedings of SIGIR-96, 19th ACM International Conference on Research and Developmentin Information Retrieval (Zurich, CH, 1996), pp. 289–297., 1996

[26] Lewis, D. D., An evaluation of phrasal and clustered representations on a text categorization task. In Proceedings of SIGIR-92, 15th ACM International Conference on Researchand Development in Information Retrieval (Kobenhavn, DK, 1992), pp. 37–50., 1992

[27] Lewis, D. D., Catlett, J., Heterogeneous uncertainty sampling for supervised learning. In Proceedings of ICML-94, 11th International Conference on Machine Learning(New Brunswick, US, 1994), pp. 148–156., 1994

[28] Lewis, D. D., Gale, W. A., A sequential algorithm for training text classifiers. In Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval (Dublin, IE, 1994), pp. 3–12., 1994

[29] Lewis, D. D., Schapire, R. E., Callan, J. P., Papka, R., Training algorithms for linear text classifiers. In Proceedings of SIGIR-96, 19th ACM International Conferenceon Research and Development in Information Retrieval (Z¨urich, CH, 1996), pp. 298–306., 1996

[30] Li, H., Yamanishi, K., Text classification using ESC-based stochastic decision lists. In Proceedings of CIKM-99, 8th ACM International Conference on Information and Knowledge Management (Kansas City, US, 1999), pp. 122–130., 1999

[31] Li, Y. H., Jain, A. K., Classification of text documents. The Computer Journal 41, 8, 537–546., 1998

[32] Moulinier, I., Ganascia, J.-G., Applying an existing machine learning algorithm to text categorization. In S. Wermter, E. Riloff, and G. Scheler Eds., Connectionist, statistical, and symbolic approaches to learning for natural language processing (Heidelberg, DE, 1996), pp. 343–354. Springer Verlag. Published in the “Lecture Notes for Computer Science” series, number 1040., 1996

[33] Moulinier, I., Raskinis, G., Ganascia, J.-G., Text categorization: a symbolic approach. In

Proceedings of SDAIR-96, 5th Annual Symposium on Document Analysis and Information Retrieval (Las Vegas, US, 1996)., 1996

[34] Ng, H. T., Goh, W. B., Low, K. L., Feature selection, perceptron learning, and a usability case study for text categorization. In Proceedings of SIGIR-97, 20th ACM International Conference on Research and Development in Information Retrieval (Philadelphia,US, 1997), pp. 67–73., 1997

[35] Ng, V., Cardie, C. , Weakly supervised natural language learning without redundant views. Proceedings of HLT-NAACL , 2003

[36] Ragas, H., Koster, C. H., Four text classification algorithms compared on a Dutch corpus. In Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval (Melbourne, AU, 1998), pp. 369–370., 1998

[37] Schapire, R. E., Singer, Y., Singhal, A., Boosting and Rocchio applied to text filtering. In Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval (Melbourne, AU, 1998), pp. 215–223., 1998

[38] Schutze, H., Hull, D. A., Pedersen, J. O., A comparison of classifiers and document representations for the routing problem. In Proceedings of SIGIR-95, 18th ACM International Conference on Research and Development in Information Retrieval (Seattle,US, 1995), pp. 229–237., 1995

[39] Singhal, A., Mitra, M., Buckley, C., Learning routing queries in a query zone. In Proceedings of SIGIR-97, 20th ACM International Conference on Research and Development in Information Retrieval (Philadelphia, US, 1997), pp. 25–32., 1997

[40] Weiss, Stephen, “Learning to disambiguate.” Information Storage and Retrieval, 9,, 1973

[41] Wiener, E., Pedersen, J. O., Weigend, A. S., A neural network approach totopic spotting. In Proceedings of SDAIR-95, 4th Annual Symposium on Document Analysis and Information Retrieval (Las Vegas, US, 1995), pp. 317–332., 1995

[42] Yang, Y., Expert network: effective and efficient learning from human decisions in text categorisation and retrieval. In Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval (Dublin, IE, 1994),pp. 13–22., 1994

[43] Yang, Y., Chute, C. G., An example-based mapping method for text categorization and retrieval. ACM Transactions on Information Systems 12, 3, 252–277., 1994

[44] Yang, Y., Pedersen, J. O., A comparative study on feature selection in text categorization. In

Proceedings of ICML-97, 14th International Conference on Machine Learning (Nashville, US, 1997), pp. 412–420., 1997

[45] Yarowsky, D., Decision lists for lexical ambiguity resolution: Application to accent restoration in Spanish and French. In Proceedings of ACL. pp. 88-95., 1994

[46] Yarowsky, D., Hierarchical decision lists for word sense disambiguation. Computers and the Humanities, 34., 2000

[47] Yarowsky, D., Unsupervised word sense disambiguation rivaling supervised methods. Proceedings of ACL , 1995