У статті розглядається можливість використання акронімів в якості лінгвістичних дескрипторів для класифікації аналізованих електронних текстів. Запропонований підхід реалізується за допомогою двоетапної процедури. На першому етапі акроніми вилучаються з декількох текстових документів розглянутої області з подальшим складанням спеціалізованих акроніміческіх словників. На другому етапі застосовується модифікована метрика DeMT, яка дозволяє визначати пертинентні визначення акроніма

Ключові слова: дескриптор, акронім, інтелектуальний аналіз, електронний текст, класифікація, семантична інформація

В статье рассматривается возможность использования акронимов в качестве лингвистических дескрипторов для классификации анализируемых электронных текстов. Предлагаемый подход реализуется с помощью двухэтапной процедуры. На первом этапе акронимы извлекаются из нескольких текстовых документов рассматриваемой области с последующим составлением специализированных акронимических словарей. На втором этапе применяется модифицированная метрика DeMT, которая позволяет определять пертинентное определение акронима

Ключевые слова: дескриптор, акроним, интеллектуальный анализ, электронный текст, классификация, семантическая информация

УДК 004.912
DOI: 10.15587/1729-4061.2015.37450

ОЦЕНИВАНИЕ ПЕРТИНЕНТНОСТИ ЛИНГВИСТИЧЕСКИХ ДЕСКРИПТОРОВ В СИСТЕМАХ ИНФОРМАЦИОННОГО ПОИСКА

Л. Э. Чалая
Кандидат технических наук, доцент*
E-mail: kovalivnich@yahoo.com
Ю. Ю. Харитонова
Аспирант*
E-mail: julie.kharitonova@gmail.com
*Кафедра искусственного интеллекта
Харьковский национальный
университет радиоэлектроники
пр. Ленина 14, г. Харьков, Украина, 61166

1. Введение

Рост массивов текстовых Веб-данных, доступных пользователю, порождает сложную проблему, связанную с их автоматической обработкой. Методы информационного поиска текстов (ИПТ) и их автоматической обработки (АО) могут отчасти соответствовать этой проблематике. Они состоят в моделировании и последующем формировании методологии, применяемой к текстовым данным, чтобы потом определить значение и получить новые знания. Большинство таких существующих методологий основано на лингвистических или статистических подходах. Процессы ИПТ и АО связаны с реализацией двух последовательных этапов. Первый этап состоит в извлечении дескрипторов (например, наиболее значимых ключевых слов). Они должны быть извлечены и обработаны, чтобы потом иметь возможность оценивать содержание текстов. После решения задачи извлечения возникает проблема эффективного использования найденных ключевых слов, связанных с исследуемой тематикой. Это составляет второй этап процесса поиска и анализа текста. Алгоритмы, используемые при этом, основаны на использовании дескрипторов и некоторых числовых текстовых характеристик (например, частоты слов в текстах). Кроме внутренней информации в текстах для этапа фильтрации можно также использовать внешние знания (словари, таксономии и т. д.). Рассмотрим разные этапы процессов ИПТ и АО с точки зрения использования лингвистических дескрипторов. Под последними будем понимать совокупность элементов (слов, акронимов, синтагм, словосочетаний и т. д.), составляющую представительный вход для алгоритмов обработки текстовых данных. Существуют различные подходы к извлечению из текстов и использованию таких дескрипторов [1, 2]. В первую очередь синтаксическая информация может извлекаться из самих дескрипторов. Такую информацию будем называть эндогенными знаниями. Например, в биомедицинской области суффиксы часто являются индикаторами патологического состояния (например, «ит» обозначает воспалительные процессы – панкреатит, аппендицит, гастрит).

Следовательно, показатели, использующие совокупность таких индикаторов, могут применяться для непосредственного получения информации по дескрипторам. Однако эндогенная информация часто бывает недостаточной, поэтому необходимо ее дополнять учетом экзогенной информации (например, синонимов или таксонимов). Следует отметить, что поиск дескрипторов часто затруднен из-за зашумленных корпусов и слабого текстового содержания (в частности, для корпусов из Веб 2.0).

2. Анализ литературных данных и постановка проблемы

В [1] приведен обзор метрик, позволяющих осуществлять поиск эндогенных дескрипторов (в том числе, и дескрипторов-акронимов) и их определений по анализу корпусов электронных текстов из сети Интернет. Однако терминологические метрики могут в этом случае иметь ограничения, связанные с возможностью присутствия в тексте полисемических элементов (элементов с несколькими значениями) или синонимичных (семантически близких) элементов. Анализ подобия структурных элементов текста является основной проблемой во многих прикладных областях построения и коррекции баз данных, таких как интеграция данных, электронный бизнес, хранилищ данных и семантической обработки запросов. Возможность запрашивать разнородные и семантически связанные источники данных зависит от способности поисковой системы находить соответствия между их структурой и/или их содержанием. Следует отметить, что большинство из инструментов, используемых в настоящее время для выявления таких соответствий, являются ручными или полуавтоматическими. В работе [2] приводится метод автоматического оценивания близости между двумя элементами текста.

В большинстве существующих подходов к поиску дескрипторов определение близости полисемических и синонимичных элементов текста, как правило, также выполняется вручную, что приводит к существенному снижению оперативности и качества поиска. В [3, 4] проводится обзор подходов к автоматическому определению такой близости и приводится классификация, которая охватывает наиболее перспективные направления к поиску дескрипторов и их определений. Использование аббревиатур в современных текстах и связанные с этим задачи в области NLP (Neuro-linguistic programming) рассматриваются в работах [5, 6]. Здесь показаны возможности применения NLP-инструментов и инженерии знаний для облегчения обработки текстовой электронной информации.

В работе [7] авторы предлагают онлайн-алгоритм, основанный на определении обобщенной меры подобия косинуса классификации k-NN, и построении соответствующей матрицы билинейной формы. В отличие от стандартной меры косинуса, осуществляемая здесь нормализация не позволяет непосредственно использовать алгоритмы, разработанные для расстояния Махаланобиса и основанные на положительных полуопределенных (PSD) матрицах. При автоматической классификации документов могут быть использованы два основных типа реляционной информации: отношения между терминами (онтологии) и отношения между документами (Web-ссылки или цитаты в статьях). В работе [8] предлагается модель, в которой традиционный тип классификатора bag-of-words постепенно расширяется для использования этих типов информации при выявлении акронимических дескрипторов.

Проведенный анализ свидетельствует о целесообразности разработки обобщенного подхода к поиску лингвистических акронимических дескрипторов, который бы учитывал особенности исследуемой области с использованием модифицированных критериев оценивания пертинентности.

3. Цели и задачи исследования

Рассмотрим задачу применения слов и акронимов в качестве эндогенных дескрипторов при анализе корпусов электронных текстов. Слово или акроним будем рассматривать, как последовательность символов, которая может содержать значимую внутреннюю семантическую информацию.

Целью данной работы является разработка и тестирование модифицированных версий критериев оценивания пертинентности лингвистических дескрипторов (на примере дескрипторов-акронимов), которые учитывали бы тематический контекст и устраняли возможную неоднозначность акронимических определений.

В соответствии с этой целью необходимо решить следующие задачи:

– провести анализ существующих методов выявления пертинентных дескрипторов;

– предложить модифицированные версии метрик для оценивания пертинентности дескрипторов-акронимов с учетом контекста;

– провести тестирование модифицированных метрик.

4. Общая характеристика задачи выявления пертинентных дескрипторов

Для оценивания семантической близости между двумя текстовыми элементами можно использовать различные меры. Ниже будут рассмотрены две классические терминологические меры. Эти меры формируют значения в интервале [0, 1]: значение 1 означает полную близость между элементами, а 0 – полное отсутствие такой близости. Техника n-грамм используется для определения количества n символов в последовательных цепочках символов. В общем случае, это количество колеблется от 2 до 5 (чаще всего выбирается равным 3). Например, триграммы для цепочек символов «терм» и «термин» соответственно такой вид: tr(терм)={тер,ерм} и tr(термин)={тер,ерм,мин}. В этом примере присутствуют две общих триграммы: «тер» и «ерм». Для расчета коэффициента близости используем следующую формулу:

$$\mathrm{tri}(e1,e2)=\frac{1}{1+|\mathrm{tr}(e1)|+|\mathrm{tr}(e2)|-2\times|\mathrm{tr}(e1)\cap\mathrm{tr}(e2)|}\in[0,1],$$

где $|\mathrm{tri}(e1)\cap\mathrm{tri}(e2)|=2$;

tri(терм)={тер,ерм)→|tri(терм)|=2;

$$\mathrm{tri}(\text{термин})=\{\text{тер},\text{ерм},\text{рми},\text{мин}\}\rightarrow|\mathrm{tri}(\text{термин})|=4;$$

$$\mathrm{tri}(\text{терм},\text{термин})=\frac{1}{1+2+4-2\times2}=0{,}33.$$

Таким образом, для рассмотренного примера получаем две операции исключения (символы «и», «н»). Кроме измерения триграмм может использоваться другая метрика, именуемая «String Matching». Эта

метрика основана на измерении расстояния *E*, которое соответствует минимальной сумме стоимости операций, осуществляемых для преобразования двух цепочек символов. При этом рассматриваются операции включения, исключения и замены символов. Например, можно осуществить две операции исключения (символы «и» и «н») между цепочками символов «терм» и «термин». При этом E(терм,термин)=2. «String Matching» (обозначим его, как Str) определится здесь формулой:

$$\mathrm{Str}(e1,e2)=\max\left\{0,\frac{\min\left\{|e1|,|e2|-\mathrm{E}(e1,e2)\right\}}{\min\left\{|e1|,|e2|\right\}}\right\}\in[0,1].$$

В рассмотренном примере получаем:

$$\mathrm{Str}(e1,e2)=\max\left\{0,\frac{4-2}{4}\right\}=0,5\,.$$

Метрики, предложенные в [1], основаны на простом вычислении количества символов, различных для двух цепочек (расстояние Хемминга) или поиске самой большой подпоследовательности. Рассмотрим теперь некоторые лексические метрики для оценивания семантической близости элементов текста.

Возможность обращения к источникам семантически близких данных зависит только от возможностей системы находить соответствие между их структурами (схемами) и/или их содержанием [3]. Отметим, что подходы, основанные на измерении семантической близости между элементами деревьев, часто используются при автоматическом формировании онтологий [5]. В проекте Forum [9] предложены методы приведения в соответствие схем для очень специализированных данных без наличия лексических метрик и семантик. Подход Bmatch основан на комбинации терминологических метрик и контекстной информации для установления соответствия между схемами. При этом здесь не используются ни словари, ни специфические знания языков. В подходе Bmatch предлагается построение контекста для каждого принимаемого во внимание элемента. При этом формируется вектор с именами близких элементов (ярлыков). Количество рассматриваемых элементов составляет один из параметров метрики. В подходе Bmatch предусматривается реализация двух основных этапов. Первый этап состоит в замене одним и тем же термином лексически близких элементов формируемого вектора. Если два термина достаточно близки, то они будут заменены тем термином, размер которого меньше, и который содержит более активную генерирующую форму. Для измерения лексической близости здесь используются «String Maching» и метрика, основанная на триграммах. Параметры этих метрик могут быть определены экспериментально. Выбор порога лексической близости осуществляется для дальнейшей замены терминов [2].

Следующий этап состоит в использовании CM-метрики, позволяющей оценить близость двух сформированных контекстных векторов. Для этого можно применить измерение косинуса, получившее широкое распространение в процедурах информационного поиска. Соответствующая метрика позволяет измерить угол между двумя формируемыми векторами. Косинус определяется как деление скалярного произведения векторов на произведение их норм. Если два вектора имеют множество общих дескрипторов, то величина, соответствующая такому косинусу, будет близка к 1 [7].

Общий принцип, используемый в проекте Bmatch, состоит в комбинировании метрики CM, основанной на контексте элементов дерева, и подхода SM, который измеряет лексическую близость между элементами [4].

Кроме рассмотрения метрик лексической близости, в последнее уделяется внимание применению экзогенных данных для определения семантических связей между элементами анализируемого текста. Например, применение семантической информации, содержащейся в онтологиях и таксономиях, позволяет расширить возможности методов информационного поиска [8].

Рассмотрим функции ранжирования, принимающие во внимание экзогенные факторы, связанные с анализом Веб-информации. Предлагаемые метрики, относящиеся к Веб-анализу (Web-Mining) используют экзогенную информацию двух типов. Во-первых, могут быть учтено присутствие ассоциаций слов в Веб-текстах. Однако информация о факте наличия соответствующих ассоциаций не всегда является достаточной. Среди индексированных Web-страниц следует учитывать наличие зашумленных документов (количество индексированных Web-страниц в Google составляет 20000 миллиардов документов [10]).

Таким образом, необходимо также принимать во внимание интенсивность этой ассоциации, которая определяется количеством страниц, выявленных поисковой системой, имеющих искомые ассоциации. При этом должна учитываться контекстная информация и параметры (например, типы операторов). Более того, для измерения уровня ассоциации возможно использование статистических критериев.

Целесообразно рассмотреть метрику (функцию ранжирования), которая классифицирует возможные ассоциации для задачи обработки акронимов/определений. Для этого, на основании списка возможных расширений акронимов, такая метрика может быть использована для отбора наиболее пертинентных расширениий для пар «акроним/определение».

Существующие метрики подобного типа основываются на алгоритме PMI-IR (Pointwise Mutual Information and Information Retrieval [6]. Этот алгоритм использует поисковый Web-навигатор AltaVista для определения соответствующих синонимов. В соответствии с заданным словом PMI-IR выбирает синоним из заданного списка. Алгоритм PMI-IR использует различные метрики, основанные на пропорциях документов, в которых присутствуют два терма. Наиболее применяемой является следующая метрика:

$$\mathrm{score}\left(\mathrm{choice}_i\right)=\frac{\mathrm{nb}\left(\mathrm{word\,NEAR\,choice}_i\right)}{\mathrm{nb}\left(\mathrm{choice}_i\right)},$$

где nb(x) – подсчитанное количество документов, содержащих слово x ; NEAR – оператор, который используется для уточнения, если два слова присутствуют вместе в окне из 10 слов.

Отметим, что информация, используемая в рассмотренном критерии ранжирования, может оказаться более весомой в сочетании со статистическими критериями.

Многие метрики качества используются для осуществления классификации в таких задачах, как поиск ассоциативных правил или извлечение терминологии [11]. Одной из метрик, используемых для расчета определенного вида зависимости каждым из слов, составляющих зависимую взаимно совокупность (взаимные вхождения), является взаимная информация:

$$I(x,y)=\log_2\frac{P(x,y)}{P(x)P(y)}, \quad (1)$$

где $P(x,y)$ – оценка вероятности появления пары слов (x,y).

Формулу (1) можно привести к следующему виду:

$$IM(x,y)=\log_2\frac{nb(x,y)}{nb(x)nb(y)}, \quad (2)$$

где nb – количество вхождений слов и пар слов.

Эта метрика может быть также расширена для вхождений из n слов:

$$IM(x_1,\ldots,x_n)=\log_2\frac{nb(x_1,\ldots,x_n)}{nb(x_1)\times\ldots\times nb(x_n)}. \quad (3)$$

Рассмотрим метрику качества, именуемую коэффициентом Dice [12]:

$$D(x,y)=\frac{2\times P(x,y)}{P(x)+P(y)}. \quad (4)$$

Коэффициент Dice позволяет учитывать редкие и часто несущественные взаимные вхождения. Формула (1) может быть приведена к виду, основанному на учете числа вхождений nb слов и пар слов:

$$Dice(x,y)=\frac{2\times nb(x,y)}{nb(x)+nb(y)}. \quad (5)$$

Очевидно, что расширение формулы (5) на n элементов имеет следующий вид:

$$Dice(x_1,\ldots,x_n)=\frac{2\times nb(x_1,\ldots,x_n)}{nb(x_1)+nb(x_n)}. \quad (6)$$

Рассматриваемые в данной статье метрики качества основаны на поиске в Web-ресурсах. В этом случае, функция nb, используемая в рассмотренных выше метриках, представляет количество страниц, выделенных поисковой системой (например, Google).

На основании формулы (6) можно получить базовую метрику DeMT, учитывающую зависимость между словами терма:

$$DeMT_{Dice}(a^j)=\frac{\left|\left\{a_i^j;a_i^j\notin M_{outils}\right\}_{j\in[1,n]}\right|\times nb\left(\bigcap_{i=1}^n a_i^j\right)}{\sum_{i=1}^n nb\left(a_i^j;a_i^j\notin M_{outils}\right)}, \quad (7)$$

где $\bigcap_{i=1}^n a_i^j$ – последовательность слов $a_i^j\,(i\in[1,n])$, которая рассматривается как цепочка символов ($n\geq 2$); M_{outils} – список слов-инструментов (предлогов, артиклей и т. д.); $|.|$ – количество слов в наборе.

В поисковой системе Google используется Web-метрика следующего вида:

$$NCG(x,y)=\frac{\max\left\{\log(nb(x)),\log(nb(y))\right\}-\log(nb(x,y))}{\log(N)-\min\left\{\log(nb(x)),\log(nb(y))\right\}}.$$

По сравнению с этой метрикой преимущество метрики Dice состоит в том, что в ней не принимается во внимание общее число N индексированных поисковой системой Web-страниц. Применение этой величины для выявления пертинентных дескрипторов может оказаться проблематичным, так как она зависит от поисковых систем и является часто аппроксимативной.

5. Модификация метрики DeMT для дескрипторов-акронимов

Базовая метрика DeMT не учитывает контекст. Однако в случае работы с акронимами целесообразно его учитывать для осуществления выбора более пертинентного определения (определим контекст, как характеристические слова в странице, где присутствует определяемый акроним). При этом могут использоваться различные контексты C: n наиболее часто встречающихся слов (кроме служебных); n наиболее часто встречающихся имен собственных; применение грамматической информации (имен, глаголов и т. д.) и/или терминологии. Целесообразно рассмотреть также комбинацию этих контекстов. Особенно важно это для контекстов, представляемых словами, наиболее часто встречающимися в текстах, в которых должна быть устранена неоднозначность акронимов. Добавление концептуальной информации в метрику DeMT (формула (7)) позволяет получить метрику DeMTC, учитывающую зависимость между словами концептуализированного терма:

$$DeMTC_{Dice}(a^j)=\frac{\left|\left\{a_i^j+C;a_i^j\notin M_{outils}\right\}_{j\in[1,n]}\right|\times nb\left(\left(\bigcap_{i=1}^n a_i^j\right)+C\right)}{\sum_{i=1}^n nb\left(a_i^j+C;a_i^j\notin M_{outils}\right)}.$$

Особенность этой метрики состоит в применении статистического подхода к набору, относящемуся непосредственно к исследуемой области. Зависимость слов из определения акронима может быть здесь вычислена на основании страниц, содержащих близкий контекст. В метрике DeMTC сумма a_i^j+C представ-

ляет слово a_i^j со всеми словами из контекста C. Величина $nb(a_i^j + C)$ соответствует количеству страниц, полученных с помощью поисковой системы по запросу $a_i^j + C$. Для примера рассмотрим акроним $a = ПО$, который содержит, как минимум, два возможных определения: «программное обеспечение» (ПО1) и «производственный отдел» (ПО2). Анализ представительного корпуса технических текстов с использованием базовой метрики DeMT показывает, что более предпочтительным является определение ПО1: $DeMT(ПО1) = 0.0075$ и $DeMT(ПО2) = 0.0053$.

Рассмотрим контекст $C = (предприятие)$. В этом случае получаем такие значения модифицированной метрики: $DeMTC(ПО1) = 0.016$ и $DeMT(ПО2) = 0.153$. Этот пример для ПО1 приводит к генерированию трех запросов: «программное обеспечение» AND «предприятие»; «программное» AND «предприятие»; «обеспечение» AND «предприятие».

Метрика DeMTC , принимающая во внимание контекст $C = (предприятие)$, позволяет сделать привилегированным определение «производственный отдел» в ассоциации с акронимом ПО. Эта метрика предполагает вычисление коэффициента Dice только по страницам, содержащим слово «предприятие».

Используя же при анализе корпуса текстов контекст $C = (алгоритм)$, получаем такие значения: $DeMTC(ПО1) = 0.035$ и $DeMTC(ПО2) = 0.009$. В этом случае является привилегированным определение «программное обеспечение». Повышение насыщенности контекста (при учете большего числа слов) позволяет акцентировать различия в значениях метрик. Например, для C =(алгоритм, подпрограмма), получаем: $DeMTC(ПО1) = 0.084$ и $DeMTC(ПО2) = 0.003$. Отметим также, что метрика DeMTC не зависит от языка анализируемых текстов.

Базовую метрику DeMTC можно модифицировать с учетом взаимной информации и взаимной информации в кубе соответственно:

$$DeMTC_{IM} = (a^j) = \frac{nb\left(\left(\bigcap_{i=1}^{n} a_i^j\right) + C\right)}{\prod_{i=1}^{n} nb\left(a_i^j + C; a_i^j \notin M_{outils}\right)}, \quad (8)$$

$$DeMTC_{IM3} = (a^j) = \frac{nb\left(\left(\bigcap_{i=1}^{n} a_i^j\right) + C\right)^3}{\prod_{i=1}^{n} nb\left(a_i^j + C; a_i^j \notin M_{outils}\right)}. \quad (9)$$

Набор метрик DeMTC можно, в частности, использовать для снятия неоднозначности акронимов, используемых в качестве лингвистических дескрипторов.

6. Устранение неоднозначности акронимов

Рассмотрим подход, позволяющий с применением метрик DeMTC снизить неоднозначность акронимов/определений. Общий процесс обработки акронимов/определений описан в [12]. В рамках этого процесса расширим набор метрик снятия неоднозначности. Следует учитывать специфику применения акронимов, которая особенно важна для задач устранения многозначности.

Акроним – это аббревиатура сформированной группы слов, в общем случае состоящая из начальных букв (инициалов) этих слов. Различия существуют между аббревиатурами, где каждая буква произносится (например, ЦВМ) и акронимами, произносимыми как классические слова (например, ТЭЦ). Однако, как правило, можно использовать один и тот же термин «акроним» для обозначения этих двух ситуаций, которые могут вызвать трудности при автоматическом распознавании. В одном заголовке со словами могут также присутствовать акронимы, имеющие несколько смыслов. Например, как уже было отмечено, акроним «ПО» может быть ассоциирован с определениями «программное обеспечение» (ПО1) или «производственный отдел» (ПО2). Существуют специализированные ресурсы, предлагающие возможные определения для одного и того же акронима (например, такой список содержится в [13]).

Проблемы касаются текстов, для которых отсутствуют любые определения акронимов. Трудность здесь состоит в автоматическом выборе более адаптированного определения. Обозначим как a заданный акроним (например, $a = ПО$). Для каждого a , определение которого не представлено в документе d , предположим наличие списка из n возможных определений $a: a_1, a_2, \ldots a_k$ (например, $a_1 =$ программное обеспечение ; $a_2 =$ производственный отдел). Задачей рассматриваемого подхода является определение такого k (для n возможных определений), чтобы было наиболее пертинентным определением для документа d . Для осуществления такого выбора используем метрику качества DeMT , которая основывается на Web-ресурсах. На рис. 1 представлена общая схема выбора определений для акронимов, содержащихся в анализируемом тексте. В этой схеме для устранения неоднозначностей в определениях акронимов используется специализированный модуль DefAcro.

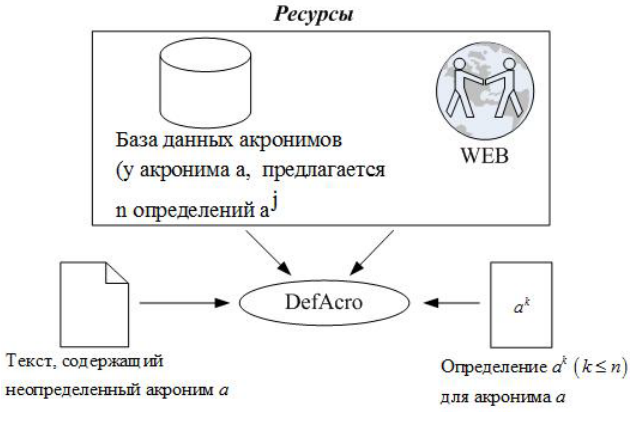

Рис. 1. Выбор определений для акронимов

В предлагаемой процедуре устранения неоднозначностей в определениях акронимов предусмотрена реализация двух следующих этапов:

– акронимы и их определения вначале извлекаются из нескольких текстовых документов (в специализированных областях, для разных языков). Это позволяет создать или расширить специализированные словари. При этом необходимо осуществить извлечение акронимов-кандидатов и их фильтрацию;

– на основании сформированных словарей применяются статистические метрики, которые позволяют определять пертинентное определение акронима, присутствующего в документе. В этих документах, как правило, не присутствуют определения акронимов, что вызывает трудности при обработке текстов. В этом случае необходимо создать (с применением метрики DeMT) адаптированный словарь, поддерживающий первую фазу рассматриваемого процесса.

При нахождении акронимов в электронном тексте предлагается использовать специальные маркеры (скобки, точки и т. д.) для деления фраз на фрагменты. Далее каждое слово из текущего фрагмента сравнивается со словами предыдущих фрагментов. Акронимы-кандидаты отбираются, если буквы акронимов соответствуют первым буквам потенциальных определений. В приведенном примере пара «ПО/программное обеспечение» является акронимом-кандидатом. Последний этап использует специфические эвристики для отбора пертинентных кандидатов. Эти эвристики основаны на том, что размер акронимов меньше размеров их определений, что они представлены прописными буквами, что определения акронимов значительной длины могут содержать служебные слова (например, артикли и предлоги), и т. д. В нашем примере пара «ПО /программное обеспечение», на которой верифицируются эти эвристики, может рассматриваться как акроним. Для расширения начального корпуса Web-страниц, используемых для составления специализированных словарей акронимов, на основе первичного списка акронимов, подаются запросы к поисковой системе. При отборе кандидатов-акронимов/определений с использованием маркеров можно использовать как для акронимов, так и для определений, необходимо учитывать два варианта представления акронимов в тексте. Первый вариант: акроним находится перед определением, которое находится между маркерами-скобками (например: «...ГИС (географические информационные системы)...»). Второй вариант: определение находится перед акронимом, расположенным между маркерами (например: «...географические информационные системы (ГИС)...»). В этом случае размер определения не является определимым в текущий момент. Поэтом необходимо случайным образом определить функцию количества букв, составляющих акроним. Такой выбор должен учитывать слова, которые могли бы попасть в отбор, не представляя для нас интереса (например, предлоги или артикли). В предлагаемом подходе этот размер зафиксирован как утроенное число букв, составляющих акроним. Например, потенциально выбираемое определение для примера «ГИС» будет состоять из девяти новых слов, которые предшествуют этому акрониму. Этап извлечения всех пертинентных кандидатов-акронимов/определений может усложняться значительным уровнем шума, потому что он основан лишь на маркерах при идентификации потенциальных кандидатов. Перед фильтрацией пар «акронимы/определения» среди выбранного списка осуществляется их сортировка, которая позваляет: исключить наиболее непертинентные пары «акронимы/определения»; уточнить определения, присутствующие в потенциальных парах-кандидатах (последние могут быть слишком длинными, так как ограничиваются случайным образом во время второго случая отбора кандидатов).

Чтобы перейти к такой фильтрации, осуществим выравнивание букв, содержащихся в акронимах, со словами определения. Такое выравнивание состоит в проверке соответствия между буквами акронимов с первыми буквами каждого из слов определений. В предлагаемом методе, если первый символ слов определения кандидата не может быть выравненным, то рассматриваются последующие символы группы слов.

Например, этод метод позволяет распознать «Кременчугский автомобильный завод» как определение акронима КРАЗ, в котором буква «Р» могла быть расширена. Отметим также, что служебные слова (предлоги, артикли и т.д.) могут рассматриваться как несколько слов без специфической обработки.

В табл. 1 представлены оценки процедуры выравнивания акронимов с кандидатами в определения. Для этого оценивания были использованы данные источника [13], где приведены более 25000 акронимов и их определений на 15 языках. Оценивание состоит в случайном извлечении из этого сайта акронимов с 2, 3 и 4 символами и оценке коэффициента успешного расширения (количества расширенных акронимов с определениями сайта на основе использования текущей версии нашего программного обеспечения).

Таблица 1

Выравнивание пар «акронимы/определения»

Количество букв	Количество акронимов	Количество определений	Расширения, %
2	102	417	95
3	48	120	82
4	23	33	68

Табл. 1 представляет результаты для 620 случаев установления соответствия, которые не всегда являются удовлетворительными (коэффициенты успешного расширения от 68 % до 95 %). Очевидно, что длинные акронимы являются более трудными для расширения. Это вызвано наличием некоторых нетипичных случаев обработки, пока не рассматриваемых программой процедуры: в частности, расширением смешанных цифровых/нецифровых символов (например, «3D/трехмерный»; «ПИ2/пропорциональный с двойным интегрированием»); присутствием в акронимах выделенных прописных букв и т. д. В специализированном модуле DefAcro используются функциональные метрики DeMT и DeMTC, адаптированные к проблематике устранения неоднозначности акронимов при выборе пертинентных акронимических дескрипторов. Помимо этих метрик, при решении задачи выбора акронимических дескрипторов целесообразно использовать метрику, отражающую степень зависимости между акронимом и его возможным расширением (в более общем случае между двумя термами). В качестве такой метрики предлагается метрика DeT (зависимость между термами) следующего вида:

$$DeT_{IM}^{And}\left(a^j\right)=\frac{nb\left(a\,And\bigcap_{i=1}^{n}a_i^j\right)}{nb\left(\bigcap_{i=1}^{n}a_i^j\right)}, \quad (10)$$

где a и a j – акроним и его определение соответственно.

Например, для a=ПО и a_j=программное обеспечение в числителе рассчитывается количество страниц, отобранных по запросу ПО AND «программное обеспечение», где указанные термы представлены на одной и той же странице.

Можно также добавить контекст C в метрику DeT и получить метрику *DeTC* (по аналогии с метрикой DeMTC).

7. Результаты тестирования модифицированных метрик оценивания пертинентности дескрипторов-акронимов с учетом контекста

Эксперименты были проведены с контекстом, формируемым из одного слова (наиболее часто встречающегося слова в каждом документе). Эксперименты были проведены в специальной области, при этом запросы с большим количеством слов часто дают нулевые значения, когда мы используем общую поисковую систему (например, Google).

В табл. 2 приведены результаты экспериментов, проведенных для метрик *DeMTC*, *DeMTC* и *DeTC*. Первый столбец соответствует количеству случаев, где правильное определение занимает первую позицию; второй столбец соответствует количеству случаев, где правильное определение занимает первую или вторую позицию; третий столбец соответствует количеству случаев, где правильное определение занимает одну из первых трех позиций. Для каждой пары отбирались резюме статей из библиографической базы данных Medline [14], содержащие акронимы и их расширения.

Результаты показывают, что при решении задачи устранения неоднозначностей акронимов/определений, предпочтительнее рассчитывать зависимость между акронимом и расширением (с применением метрики *DeTC*), чем зависимость между словами определений.

Таблица 2

Оценки пертинентности определений

Ранг	1	1 или 2	1, 2 или 3
DeMT(7)	72 (34 %)	124 (60 %)	158 (75 %)
DeMTC(8)	59 (29 %)	106 (52 %)	145 (70 %)
DeMTC(9)	71 (32 %)	112 (56 %)	162 (79 %)
DeTC(10)	101 (51 %)	129 (66 %)	159 (81 %)

Отметим, что метрика *DeTC* позволяет находить пертинентное определение с рангом 1 в 51 % случаев. В той же ситуации, случайная предикция позволяет получить лишь 22 % пертинентных определений.

8. Выводы

Анализ показывает, что эндогенная информация позволяет определить важные семантические знания. С другой стороны, экзогенные знания являются важными для улучшения подходов по анализу текстов. Для осуществления процессов обработки Веб-текстов (в данном случае, для устранения неоднозначности акронимов/определений), часто необходимо иметь дополнительную информацию, связанную со спецификой контекста. В общем случае принятие во внимание особенностей контекста при выборе акронимических лингвистических дескрипторов позволяет улучшить интерпретацию результатов. Модифицированная метрика DeMT , учитывающая контексты, может быть успешно адаптирована к задаче оценивания пертинентности лингвистических дескрипторов (в частности, акронимов и пар «акроним/определение»). Результаты тестирования подтверждают работоспособность предложенного подхода.

Перспективным представляется расширение предложенного подхода для адаптации рассмотренных модифицированных метрик при выборе и оценивании различных типов лингвистических дескрипторов (например, слов и синтагм) и для анализа эффективности их использования в системах инфромационного Веб-поиска.

Литература

1. Navarro, G. A guided tour to approximate string matching [Text] / G. Navarro // ACM Computing Surveys. – 2001. – Vol. 33, Isuse 1. – P. 31–88. doi: 10.1145/375360.375365
2. Duchateau, F. A context-based measure for discovering approximate semantic matching between schema elements [Text] / F. Duchateau, Z. Bellahsene, M. Roche // In Proceedings of IEEE Research Challenges in Information Science (RCIS), 2007. – P. 9–20.
3. Rahm, E. A survey of approaches to automatic schema matching [Text] / E. Rahm, P.A. Bernstein // VLDB Journal: Very Large Data Bases. – 2001. – Vol. 10, Issue 4. – P. 334–350. doi: 10.1007/s007780100057
4. Duchateau, F. Improving quality and performance of schema matching in large scale [Text] / F. Duchateau, Z. Bellahsene, M. Roche // Ingénierie des Systèmes d'Information (ISI). – 2008. – Vol. 13, Issue 5. – P. 59–82. doi: 10.3166/isi.13.5.59-82
5. Aussenac-Gilles, N. Construction d'ontologies à partir de textes [Text] / N. Aussenac-Gilles, D. Bourigault // In Actes de Traitement Automatique des Langues Naturelles (TALN). – 2003. – Vol. 2. – P. 27–47.
6. Turney, P. Mining the Web for synonyms: PMI–IR versus LSA on TOEFL [Text] / P. Turney // Proceedings of the 12th European Conference on Machine Learning (ECML), LNCS. 2001. – P. 491–502. doi: 10.1007/3-540-44795-4_42
7. Qamar, A. Online and batch learning of generalized cosine similarities [Text] / A. Qamar, E. Gaussier // In Proceedings of International Conference on Data Mining (ICDM), 2009. – P. 926–931. doi: 10.1109/icdm.2009.114
8. Nyberg, K. Document classification utilising ontologies and relations between documents [Text] / K. Nyberg, T. Raiko, E. Hyvönen, T. Tiinanen // In Proceedings of the Eighth Workshop on Mining and Learning with Graphs (MLG), 2010. – P. 86–93. doi: 10.1145/1830252.1830264

9. Bellahsene, Z. Forum: a flexible data integration system based on data semantics [Text] / Z. Bellahsene, S. Benbernou, H. Jaudoin, F. Pinet, O. Pivert, F. Toumani, S. Bernard, P. Colomb, R. Coletta, E. Coquery, F. De Marchi, F. Duchateau, M.-S. Hacid, A. HadjAli, M. Roche // SIGMOD Record. – 2010. – Vol. 39, Issue 2. – P. 11–18.
10. Roche, M. AcroDef: A quality measure for discriminating expansions of ambiguous acronyms [Text] / M. Roche, V. Prince // Modeling and Using Context. Springer-Verlag Berlin Heidelberg, 2007. – P. 411–424. doi: 10.1007/978-3-540-74255-5_31
11. Roche, M. Intégration de la construction de la terminologie de domaines spécialisés dans un processus global de fouille de textes [Text]: PhD thesis / M. Roche. – Paris, 2004.
12. Smadja, F. Translating collocations for bilingual lexicons: A statistical approach [Text] / F. Smadja, K.R. McKeown, V. Hatzivassiloglou // Computational Linguistics. – 1996. – Vol. 22, Issue 1. – P. 1–38.
13. Dictionnaire de sigles et acronyms [Electronic resource] / G. Blandin. – Asankyeya, 2005. – Available at: http://www.sigles.net
14. Medline [Electronic resource] / R. Pike. – USA, 2004. – Available at: http://www.ncbi.nlm.nih.gov/PubMed

У множинній лінійній регресії, коли провісники сильно корельовані, оцінки найменших квадратів (LSE), як правило, дають неточні прогнози. Гребенева регресія, яка грунтується на мінімізації квадратичної функції втрат, чутлива до викидів. Розглянуто дві гладко знижені ψ-функції, засновані на принципі Вінзора, які призводять до асимптотично ефективних оцінок

Ключові слова: М-оцінки, принцип Вінзора, робастні гребеневі оцінки

В множественной линейной регрессии, когда предсказатели сильно коррелированы, оценки наименьших квадратов (LSE), как правило, дают неточные прогнозы. Гребневая регрессия, основываясь на минимизации квадратичной функции потерь, чувствительна к выбросам. Рассмотрены две сглаженно сниженные ψ-функции, основанные на принципе Винзора, которые приводят к асимптотически эффективным оценкам

Ключевые слова: М-оценки, принцип Винзора, робастные гребневые оценки

УДК 519.6

DOI: 10.15587/1729-4061.2015.37316

УЛУЧШЕННЫЕ РОБАСТНЫЕ ГРЕБНЕВЫЕ ОЦЕНКИ РЕГРЕССИИ

В. И. Грицюк

Кандидат технических наук, доцент

Кафедра проектирования и эксплуатации электронных аппаратов

Харьковский национальный университет радиоэлектроники

пр. Ленина, 14, г. Харьков, Украина, 61166

E-mail: astra_kk12@mail.ru

1. Введение

Гребневая регрессия чувствительна к выбросам. Гребневая регрессия и робастная регрессия были предложены для решения этой проблемы мультиколлинеарности и выбросов в классической линейной регрессионной модели соответственно. Эта статья предлагает робастную и гребневую регрессии для одновременного решения проблемы мультиколлинеарности и определения выбросов в классической линейной регрессионной модели.

Когда предикторные переменные мультиколлинеарны, оценки наименьших квадратов могут быть слишком большими по абсолютной величине и дисперсии, могут стать очень большими.

2. Анализ литературных данных и постановка проблемы

В этой статье мы рассмотрим гребневые оценки. В множественной линейной регрессии, когда предсказатели тесно связаны, оценки наименьших квадратов (LSE) дают неточные прогнозы. В попытке исправить это, Hoerl и Kennard предложили гребневую регрессию [1, 2]. Они добавили штраф, который создаёт небольшое смещение для того, чтобы одновременно уменьшить оценку и уменьшить дисперсию, что приводит к повышению общей точности прогнозирования. Многие проблемы регрессии состоят как в мультиколлинеарности, так и в ненормальности в той или иной степени. Известно, что метод НК ведет себя плохо, когда распределение ошибок не является нормальным, особенно, когда ошибки являются тяжелыми хвостами, то есть, если существуют отдаленные наблюдения. Эта чувствительность МНК к выбросам результатов приводит к очень обманчивым результатам. Чтобы справиться с этой проблемой была разработана методика робастной регрессии. Наиболее распространенным является метод робастной регрессии М-оценки, введенный Хьюбером [5–3]. Холланд изучал совокупную проблему, и предложил использовать взвешенную гребневую регрессию с робастным выбором весов. В статье представлен подход, основанный на сочетании математических формулировок программирования как гребневой регрессии, так и робастной регрессии. Искомые коэффициенты регрессии могут быть легко вычислены путем итератив-