МЕТОД АННОТИРОВАННОГО СУФФИКСНОГО ДЕРЕВА ДЛЯ ОЦЕНКИ СТЕПЕНИ ВХОЖДЕНИЯ СТРОК В ТЕКСТОВЫЕ ДОКУМЕНТЫ

Б.Г. Миркин,
доктор технических наук, профессор кафедры анализа данных и искусственного интеллекта Национального исследовательского университета «Высшая школа экономики»

Е.Л. Черняк,
студент магистерской программы «Математическое моделирование» Национального исследовательского университета «Высшая школа экономики»

О.Н. Чугунова,
студент магистерской программы «Математическое моделирование» Национального исследовательского университета «Высшая школа экономики»

E-mail: bmirkin@hse.ru, ktr.che@gmail.com, olya.chug@gmail.com
Адрес: г. Москва, Покровский бульвар, д. 11

Излагается модификация метода аннотированного суффиксного дерева (АСД), разработанного с участием одного из авторов, которая ориентирована на то, чтобы, во-первых, убрать априорное ограничение на глубину конструируемого дерева, во-вторых, сделать более адекватной оценку степени вхождения последовательности букв в текст, и, в-третьих, рассмотреть другие приложения метода. На конкретных примерах описываются методы разработки и использования АСД для двух классов задач анализа текстовой информации: (а) связь корпуса текстов и совокупности ключевых словосочетаний; (б) связь корпуса текстов с таксономией предметной области.

Ключевые слова: анализ текстов, аннотированное суффиксное дерево, интерпретация, концептуальные кластеры.

Введение

Основные работы по автоматизации обработки и анализа текстов идут в разрезе представления текстов как совокупностей слов, как это делается в наиболее популярных методиках «модели мешка слов» и «обработки естественного языка». Значительно реже применяются методики, основанные на представлении текстов как последовательностей символов. Между тем, последние имеют то значительное преимущество, что они не требуют предварительной обработки текстов, на-

пример, выделения таких видов слов как «ключевые слова», «стоп-слова», и пр. В данной работе представлена методика анализа текстов, основанная на понятии суффиксного дерева. Суффиксное дерево – одна из основных структур данных для хранения и поиска фрагментов символических последовательностей – текстов, биологических последовательностей, и т.п. [1]. В последнее время ее все чаще используют для кластеризации текстовых объектов, см. например, [2]. Мы используем суффиксные деревья, аннотированные частотами вхождения фрагментов текстов (АСД), иногда также называемые обобщенными суффиксными деревьями. Этот инструмент позволяет эффективно решать задачи анализа текстов, не связанные с их грамматическими описаниями и не зависящие от языка, на котором они написаны. В частности, в [3] этот аппарат использовался для выделения так называемого «спама» в составе электронной почты. В данной работе метод АСД модифицирован: во-первых, убрано априорное ограничение на глубину конструируемого суффиксного дерева, во-вторых, предложена новая формула оценки степени вхождений, позволяющая убрать ее зависимость от длины анализируемых текстов и, в-третьих, рассматриваются приложения, основанные на предварительном построении так называемой ПС таблицы (см. раздел 3.1). Это: (а) анализ связи данного корпуса текстов и совокупности ключевых словосочетаний путем анализа структуры либо множества текстов, либо множества словосочетаний; (б) анализ связи текстов с таксономией своей предметной области.

Далее в разделе 2 описаны методы построения аннотированного суффиксного дерева (АСД) и наложения строки на АСД. Раздел 3 описывает методики и примеры решения задач типа (а), а раздел 4 посвящен примерам задач типа (б). Раздел 5 заключает статью.

1. Индексирование фрагментов текста с помощью аннотированного суффиксного дерева (АСД)

1.1. Понятие аннотированного суффиксного дерева

Мы рассматриваем текст как последовательность символов. Для уменьшения объема вычислений текст разбивается на короткие фрагменты, в дальнейшем называемые строками. При этом текст рассматривается не как единое целое, а неупорядоченная совокупность строк. Любые фрагменты, содержащие семантически законченные отрывки текста, могут быть использованы в качестве строк. Использование «строкового» представления теряет семантические связи между отрывками, но сохраняет связи внутри них. Однако теряемые нами «дальние» связи обычно теряются и при других подходах к анализу текстов, как, например, при подходе «обработки естественного языка» [4]. Вместе с тем сохраняются «локальные» связи, что существенно отличает данный подход от другого популярного подхода, использующего модель так называемого «мешка слов» [3-5].

АСД организовано как корневое дерево, в котором каждый узел, кроме корня, помечен одним из символов строки [3]. Путь от корня АСД до любого узла кодирует один из фрагментов строки, а путь из корня до листа дерева – один из суффиксов строки. Помимо символьной метки, каждый узел обладает числовой меткой, соответствующей частоте соответствующего фрагмента.

Рис. 1 представляет АСД для строки ‘XABXAC’ (в [1] дается другое, обычное, суффиксное дерево для этой строки).

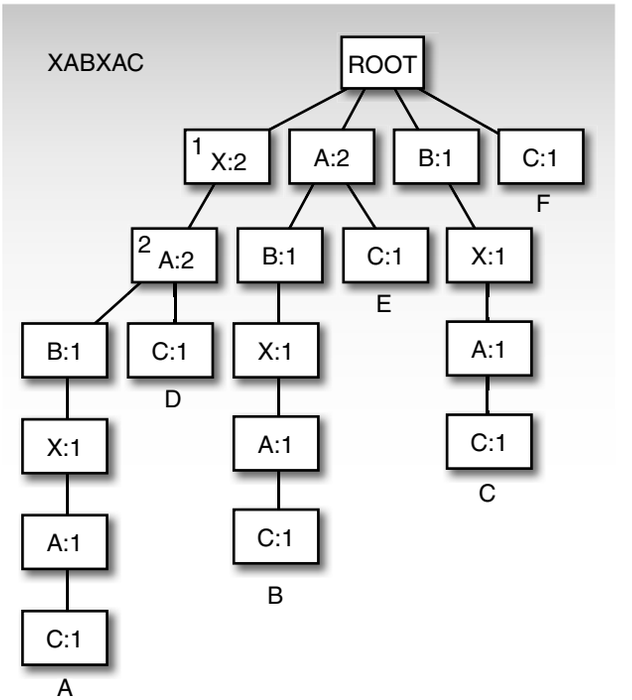

Рис. 1. Аннотированное суффиксное дерево для строки ‘XABXAC’

У строки s = ‘XABXAC’ шесть суффиксов: s[1:] = ‘XABXAC’, s[2:] = ‘ABXAC’, s[3:] = ‘BXAC’, s[4:] = ‘XAC’, s[5:] = ‘AC’, s[6:] = ‘C’. В АСД на Рис. 1 их кодируют пути от корня ROOT до листьев A, B, C, D, E, F, соответственно. Сочетание ‘XA’ дважды

встречается в строке s: это фрагменты s[1:2] = s[4:5] = ‘XA’. На *рис. 1* эти символы представлены узлами, помеченными номерами 1 и 2. Узел 2 имеет двух потомков с частотами 1, соответствующих символам s[3] = ‘B’ и s[6] = ‘C’. Другими словами, ‘XA’ – префикс обоих суффиксов A и D.

АСД для двух и более строк не имеет принципиальных отличий от АСД для одной строки. Оно получается добавлением информации строк к уже построенному дереву и так же представляет все фрагменты коллекции строк и их частоты. Рассмотрим АСД для двух строк: s = ‘XABXAC’ и t = ‘BABXAC’ (*рис. 2*). Строки s и t различаются только первыми символами (s[1] = ‘X’, t[1] = ‘B’). Их суффиксы, начиная со вторых, полностью совпадают. Поэтому АСД на *рис.* 2 отличается от АСД на *рис. 1* только тем, что, во-первых, в нем - новая цепочка узлов G, соответствующая суффиксу t[1:] = ‘BABXAC’, и, во-вторых, изменились частоты, приписанные узлам. Теперь фрагмент ‘XA’ встречается три раза (s[1:2] = s[4:5] = t[4:5] = ‘XA’), поэтому в новом дереве узлам 1 и 2 приписаны частоты 3. Узел 3, помеченный символом ‘B’, является префиксом двух суффиксов: s[3:] = t[3:] = ‘BXAC’, t[1:] = ‘BABXAC’, причем частота первого из них равна 2, второго – 1. Поэтому узлу 3 приписана частота 3.

Важное свойство АСД: частота любого узла равна сумме частот его узлов-детей, так как родительский узел соответствует префиксу нескольких суффиксов и его частота складывается из частот этих суффиксов. Отсюда же следует, что частота родительского узла равна сумме частот листьев, которые он покрывает.

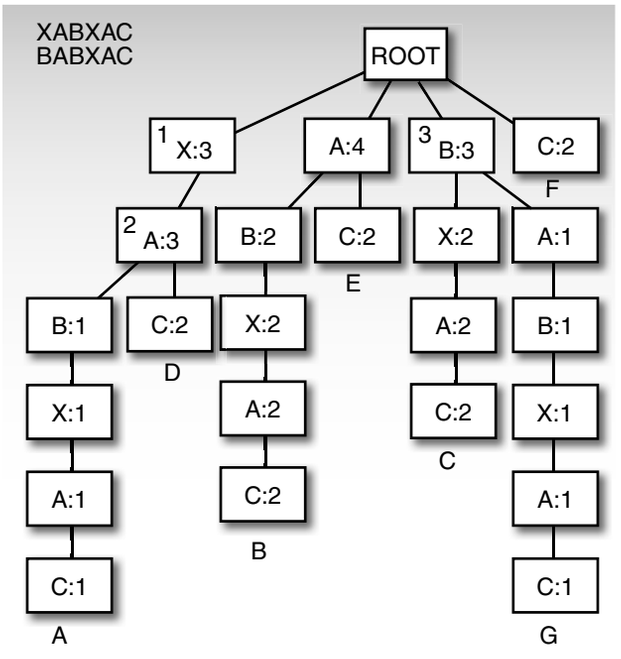

Рис. 2. Аннотированное суффиксное дерево для строк ‘XABXAC’ и ‘BABXAC’

1.2 Алгоритм построения аннотированного суффиксного дерева

Алгоритм построения АСД для коллекции строк основан на последовательном переборе всех суффиксов всех строк из коллекции. Приведем точный алгоритм, модифицированный по сравнению с [3] в связи со строковым представлением текста и измененной мерой качества вхождения.

Построение дерева для коллекции строк $s_1,..., s_N$

Инициализация: создаем пустую структуру, в которой будет храниться АСД. Обозначим ее ast. Далее итеративно будем добавлять в ast подструктуры, соответствующие строкам из входной коллекции.

На i-той итерации алгоритма, i=1, 2, ...,N, для строки $s=s_i$ длины l:

Находим все суффиксы $s[j:]$, где j=1, 2, ..., l.

Для каждого суффикса $s[j:]$ ищем в ast совпадение – путь от корня, совпадающий с максимальным начальным отрезком суффикса $s[j:]$.

Пусть найдено совпадение $m = s[j:k]$, где $k \le l$. Для узлов, попавших в совпадение, увеличиваем частоты на 1.

Если $k < l$, требуется создать новые узлы для фрагмента строки $s[k+1:l]$. Для этого создаем у последнего узла в найденном совпадении нового потомка, помечаем его символом $s[k+1]$ и приписываем ему частоту 1. Таким же образом последовательно создаем узлы для всех оставшихся символов в фрагменте. В результате будет создана новая цепочка узлов, кодирующая текущий суффикс. Если $k = l$, новые узлы не создаются.

Как пример, построим АСД для строки s = ‘XABXAC’ (*рис. 1, 2*). Для первых трех суффиксов, XABXAC, ABXAC, BXAC, совпадений не будет найдено. Поэтому в дерево будут добавлены соответствующие цепочки к узлам A, B, C, у которых частота каждого узла определяется как равная 1. При добавлении следующего суффикса $s[4:]$ = ‘XAC’ будет найдено непустое совпадение ‘XA’, поэтому частота узлов 1 и 2 из совпадения будет увеличена на 1, а для узла 2 - создан новый потомок с меткой ‘C’ и частотой 1. Аналогично, при добавлении суффикса $s[5:]$ = ‘AC’ будет найдено совпадение из одного узла с меткой ‘A’. Следуя алгоритму, частота узла

будет увеличена на 1 и у него будет создан новый потомок с меткой ‘C’ и частотой 1.

Если к уже построенному для строки s=‘XABXAC’ АСД требуется добавить строку t = ‘BABXAC’, то для первого суффикса $t[1:]$ = ‘BABXAC’ будет найдено совпадение из одного узла с меткой ‘B’. Увеличим его частоту и добавим к нему путь из всех остальных символов суффикса. Получим цепочку узлов G на *рис. 2.* Для всех остальных суффиксов строки t будут найдены совпадения, полностью покрывающие суффиксы, поэтому у всех узлов в дереве частоты будут увеличены, но новых узлов создано не будет.

1.3 Процедура наложения строки на АСД

Использование АСД позволяет оценить степень вхождения строки, как последовательности символов, в данный текст.

Введем обозначения: *ast*– АСД, построенное для коллекции строк, $ROOT$ – корень АСД, u – узел АСД, $f(u)$ – его частота, s – строка, степень вхождения которой в коллекцию требуется оценить.

Условной вероятностью узла u назовем величину

$$\hat{p} = \frac{f(u)}{f(parent(u))},$$

где $f(u)$ –частота узла u, $f(parent(u))$– частота узла-родителя u.

Условную вероятность узла на первом уровне АСД считаем равной

$$\frac{f(u)}{\sum_{n:parent(n)=ROOT} f(n)},$$

где n:*parent*$(n)=ROOT$ – множество всех узлов n на первом уровне дерева.

Оценка вхождения строки s в дерево ast получается путем усреднения оценок всех суффиксов строки s. Для каждого суффикса строки s найдем в дереве ast совпадение $m = m_1 \ldots m_k$. Оценка совпадения m – это сумма условных вероятностей узлов, принадлежащих совпадению:

$$score(m) = \sum_{i=0}^{k} \hat{p}(m_i) \qquad (1)$$

Это позволяет агрегировать оценки всех совпадений по формуле:

$$SCORE(s,ast) = \frac{\sum_{i=1}^{l} s([i:]score,ast)}{l}, \qquad (2)$$

где l – длина строки s, $s[i:]$– ее i-тый суффикс. Полная оценка (2) строки s – это средняя оценка совпадений ее суффиксов с последовательностями в дереве.

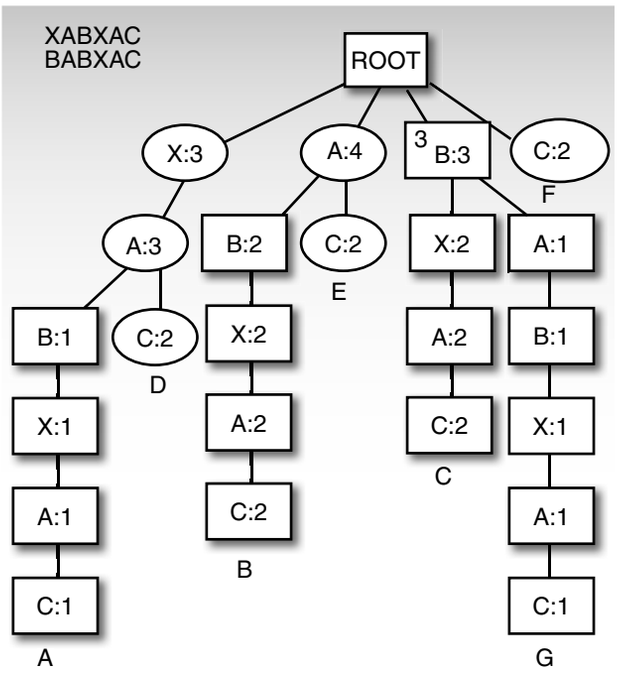

Рис. 3. Наложение строки ‘VXACA’ на АСД коллекции строк {‘XABXAC’,‘BABXAC’}; концы совпадений отображены овалами

Таблица 1.

Оценка результатов наложения всех суффиксов строки ‘VXACA’ на АСД, построенное для коллекции строк {‘XABXAC’,‘BABXAC’}

Суффикс	Совпадение	Score
‘VXACA’	Нет	0
‘XACA’	‘X’->’A’->’C’	3/12 + 3/3 + 2/3=1 11/12
‘ACA’	‘A’->’C’	4/12 + 2/4=5/6
‘CA’	‘C’	2/12
‘A’	A’	4/12

Пример: оценка степени вхождения строки ‘VXACA’ в коллекцию {‘XABXAC’, ‘BABXAC’} (*рис. 3*).

Для того, чтобы найти условную вероятность узлов на первом уровне АСД, будем считать, что частота корня равна сумме частот всех узлов на первом уровне: 3+4+3+2 = 12. Таким образом, для суффикса ‘XAC’ (совпадение по ветви D) получается сумма трех слагаемых, 3/12 для ‘X’, 3/3 для ‘A”, и 2/3 для ‘C’. В *табл. 1* приведены расчеты для всех суффиксов строки. Суммирование этих оценок ‘C’ последующим усреднением дает SCORE = 3.25/ 5 = 0.65.

1.4. Нормировка оценки при сравнении строк с различными АСД

Часто возникает потребность сравнить оценки сходства строк с двумя или более АСД. Получаемые оценки могут сильно зависеть от размеров АСД. Чем больше узлов в АСД, тем больше разброс оценок, получаемых при сличении строк с этим деревом. Для того, чтобы сделать оценки по разным деревьям сравнимыми между собой, модифицируем формулы (1) и (2) так, чтобы нормировать результаты по длине фактических совпадений:

$$score(m) = \frac{\sum_{i=0}^{k} \hat{p}(m_i)}{k}, \qquad (1^*)$$

где k – длина найденного совпадения $m = m_1 ... m_k$. Общая оценка

$$SCORE(s, ast) = \frac{\sum_{i=1}^{l} score(s[i:], ast)}{l} \qquad (2^*)$$

имеет смысл условной вероятности, приходящейся на одну букву суффикса в совпадениях.

Это делает оценки сравнимыми как по документам, так и по словосочетаниям.

1.5 Другие направления использования суффиксных деревьев

Обычные суффиксные деревья часто называют моделью представления текста, альтернативной модели «мешка слов». Мешок слов – пожалуй, самый популярный способ представления текста в компьютере – представляет собой вектор, компоненты которого соответствуют отдельным словам и равны их частотам. Эта модель обладает рядом недостатков и ограничений. Во-первых, в такой модели теряются связи между словами из словосочетаний. Во-вторых, такая модель не всегда удобна. В работе [2], например, утверждается, что в задаче кластеризации текстовых документов модель «мешок слов» не эффективна из-за чрезмерно большой размерности и разреженности векторов частот, представляющих отдельные тексты. С точки зрения авторов этой работы, в задаче кластеризации текстовых документов использование суффиксных деревьев более обосновано. Заметим, что в и в той работе, и в множестве других, суффиксное дерево понимается как иерархическая структура слов, а не символов. Такой поход к представлению суффиксных деревьев впервые предложен в [5] и с алгоритмической точки зрения не отличается от традиционного, изложенного в [1].

Наше представление суффиксных деревьев несколько отличается: во-первых, аннотированные суффиксные деревья имеют другую структуру, во-вторых, мы используем их с другими целями. В рассматриваемых в данной статье задачах аннотированное суффиксное дерево используется в первую очередь для характеристики связей между фрагментами текстов и коллекцией текстов. Эта задача отличается от задачи кластеризации текстовых документов и требует, очевидно, анализа текста не на уровне слов, а на уровне символов.

2. Использование метода АСД для анализа текстов по словосочетаниям

В этом разделе будут описаны два подхода к анализу пары «корпус текстов – совокупность ключевых словосочетаний». Один подход связан с исследованием структуры корпуса в разрезе словосочетаний; другой – с исследованием структуры связей между словосочетаниями согласно данному корпусу.

2.1. ПС таблица

Метод АСД может использоваться для анализа структуры корпуса текстов в разрезе определенных словосочетаний, связанных с этим корпусом. Рассмотрим какой-нибудь корпус текстов, например, набор публикаций о бизнес-процессе в после-кризисной России (2009-2010 годы). Словосочетания могут характеризовать различные типовые события:

1. Изменение организационно-правовой формы
2. Изменение уровня концентрации собственности
3. Повышение эффективности управления затратами
4. Смена генерального директора
5. Участие в судебных разбирательствах
6. Присвоение кредитного рейтинга
7. Выход на международный рынок
8. Публикация финансовой отчетности
9. Реструктуризация кредита
10. Первичное размещение на зарубежной бирже и др.

С помощью АСД метода построим таблицу «публикация-словосочетание» (ПС таблица), в которой строки соответствуют отдельным публикациям (текстам), столбцы – отдельным словосочетаниям, а элементы – оценки степени вхождения строк-словосочетаний в АСД, построенное для соответствующей публикации. Для каждой публикации

строим свое АСД, а затем, с помощью процедуры наложения вычислим оценки степеней вхождения каждого словосочетания публикацию (см. *табл. 2*, представляющую фрагмент одной из наших ПС таблиц). Мы экспериментировали с различными методами разбиения статьи на строки. Хорошие результаты достигаются при разбиении публикации на тройки слов, по-видимому, из-за того, что большинство рассматриваемых словосочетаний тоже состоит из трех слов, так что глубина АСД получается близкой к длине словосочетаний, с ним сличаемых. Построенную ПС таблицу можно использовать как для анализа структуры корпуса публикаций, так и для анализа взаимосвязей между словосочетаниями.

2.2 Анализ структуры корпуса публикаций путем иерархической группировки

ПС таблица позволяет использовать словосочетания как количественные признаки, значения которых она содержит. С ее помощью можно построить иерархическую классификацию публикаций. Мы используем метод иерархической концептуальной кластеризации [6] в модификации Миркина [7]. Концептуальный кластер-анализ отличается от остальных методов кластер-анализа тем, что разделение кластеров в нем осуществляется не по многомерному расстоянию, комбинирующему в себе действие всех рассматриваемых признаков, а по только одному из признаков. Если признак x количественный, то две части, на которые разбивается кластер, отвечают предикатам “x>a” и “x≤a” для некоторого значения a. Если признак – категоризованный, то две части разделения отвечают предикатам “x=a” и “x≠a” для некоторой категории a. В процессе вычислений осуществляется полный перебор всех кандидатов для разбиения – по всем признакам и всем их значениям a – их на самом деле очень немного, линейная функция от числа признаков, и выбирается то разделение, для которого суммарная ассоциация с существующими признаками максимальна. При этом максимально и многомерное расстояние Уорда между центрами разделенных частей [7]. Получаемое «концептуальное» дерево имеет простую интерпретацию и, кроме того, выступает в качестве инструмента отбора информативных признаков – тех, которые действительно участвуют в разделениях. Степень ассоциации иерархического разбиения с признаками измеряется так называемым корреляционным отношением в случае количественных признаков, или коэффициентами, основанными на таблице сопряженности между искомым разбиением и признаками, в случае категоризованных признаков. Оказывается, в последнем случае некоторые известные коэффициенты ассоциации, популярные в построении решающих деревьев, такие как индекс Джини и коэффициент сопряженности Пирсона, эквивалентны специальным случаям критерия квадратичной ошибки в методе к-средних, при условии, что отдельные категории представлены как числовые 1/0 признаки и подходящим образом нормированы [7].

Благодаря специфике метода, каждый кластер в полученной иерархии может быть легко интерпретирован предикатами на пути от корня дерева до листа, соответствующего рассматриваемому кластеру. В силу своей дихотомической структуры, метод может на разных этапах построении дерева иерархии использовать различные значения одного и того же признака, что многократно происходило при расчетах. При этом возникает отдельная задача согласования соответствующих бинарных предикатов — формирование интервалов оценки степени вхождения, соответствующих тому или иному кластеру. Например, кластер {2,4,5} на *рис. 4* описыва-

Таблица 2.

Фрагмент ПС таблицы*

	Доклад Всемирного Банка об экономике России	Международные стандарты финансовой отчетности	Если генеральный директор иностранец
1. Изменение организационно-правовой формы	0.3145	0.3616	0.3644
2. Изменение уровня концентрации собственности	0.5016	0.3148	0.2706
3. Повышение эффективности управления затратами	0.4433	0.2809	0.2445
4. Смена генерального директора	0.2264	0.2351	0.5947

* Столбцы соответствуют публикациям, а строки – словосочетаниям, значения в клетках – степени вхождения словосочетаний в публикации

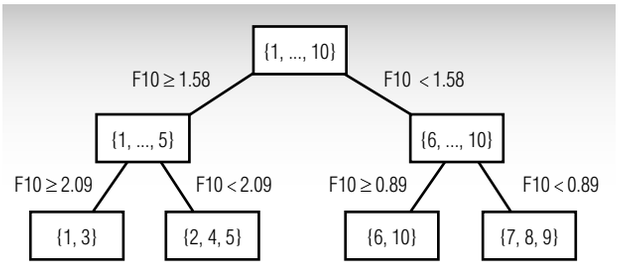

Рис. 4. Пример иерархической классификации

ется условием: степень вхождения словосочетания F10 находится в пределах между 1.58 и 2.09.

В одном из расчетов таким способом было получено дерево иерархии, имеющее 7 уровней и 51 узел, 26 из которых являются листьями (они и есть искомые кластеры публикаций). При этом из нескольких десятков рассматривавшихся словосочетаний в построенном дереве были использованы только те десять, что перечислены в начале раздела 3.1. Этот список характеризует существенные стороны выживания и развития компаний, особенно в после-кризисный период. Он оказался довольно устойчивым относительно различных методов разделения текстов на строки, а также преобразования данных путем обнуления малых значений оценки вхождения словосочетаний в тексты.

2.3. Анализ связей между ключевыми словосочетаниями

Ту же ПС таблицу можно использовать для анализа связей между входящими в нее словосочетаниями. Все публикации разделяются на три типа: (1) статьи, в которых явно выражено только одно словосочетание, так что его оценка по методу АСД значительно превосходит оценки всех остальных словосочетаний; (2) статьи с высокими оценками вхождения двух и более словосочетаний; (3) статьи, где нет ни одного словосочетания с высокой оценкой. Поэтому для каждого словосочетания определены множества публикаций, составляющие его моно- и мульти-ядра, т.е. множества публикаций только типа (1) (моноядро) и типа (2) (мультиядро). Объединение этих двух типов образует множество всех публикаций F(A), в которых оценка вхождения соответствующего словосочетания A превышает заданный порог.

Будем считать, что словосочетание A влечет словосочетание B согласно данному корпусу публикаций, если доля множества F(B) в F(A) составляет не менее 60%. Это правило, соответствующее традиционным логико-статистическим построениям, напоминает известный аппарат построения ассоциаций в так называемом майнинге данных [8]. Действительно, в обоих случаях имеется в виду, что одно множество объектов содержит другое, с точностью до определенной ошибки, конечно. Однако имеется и существенная разница. В майнинге данных импликации (ассоциативные правила) ищутся по всему массиву данных, без привязки к каким-либо специфическим утверждениям, что требует задания дополнительного порога на уровень «поддержки» импликации. В нашем случае импликации привязаны к заранее заданным словосочетаниям, и не нуждаются в проверке на уровень поддержки. Данный подход ближе к так называемому детерминационному анализу [9]. Но в детерминационном анализе главное – выявление системы категорий, комбинация которых приводит к максимальной точности импликации. Для нас же главное – выявление структуры связей между заданными словосочетаниями. В частности, в расчете по публикациям о бизнес-процессе в России с использованием нескольких десятков словосочетаний был получен граф, представленный на *рис. 5.* Словосочетания, соответствующие его вершинам:

1. Ввод автоматизированного производства.
2. Выпуск пресс-релизов (с положительными или отрицательными новостями).
3. Изменение размера пакета акций, принадлежащего институциональному инвестору.
4. Изменение уровня концентрации собственности.
5. Повышение квалификации персонала.
6. Проведение вертикального слияния.
7. Проведение операций купли-продажи бренда.
8. Выход на международный рынок.
9. Изменение организационно-правовой формы.
10. Повышение эффективности управления затратами.
11. Публикация финансовой отчетности.
12. Смена финансового директора.

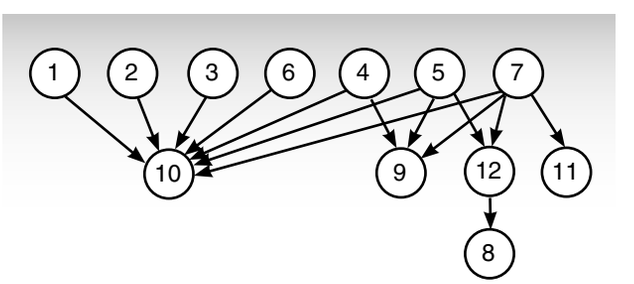

Рис. 5. Граф значимых (на уровне 60%) связей между словосочетаниями по массиву публикаций. Расшифровка номеров приведена в тексте.

Обращает на себя внимание то, что в графе нет контуров и всего два основных уровня, естественным образом интерпретируемых как источники и цели. С содержательной точки зрения, граф допускает разумную интерпретацию, связанную с содержанием процессов развития бизнес-процесса в России в 2009-2010 гг. Показанные в нем цели: уменьшение издержек (10), изменение организационно-правовой формы (9), повышение прозрачности (11), выход на мировые рынки (8) действительно могут помочь бизнесу выжить и развиваться; событие (12) оказывается шагом, ведущим к выходу на мировые рынки. Согласно графу, основными факторами этого процесса являются: купля-продажа брендов (развитие сетевых структур), автоматизация производства, повышение квалификации персонала, а также передача государственных активов в частные руки.

3. Использование метода АСД для анализа связи текстов с таксономией предметной области

В этом разделе будет рассмотрена возможность отображения текстов на таксономию соответствующей предметной области. Под таксономией, или иерархической онтологией, понимается иерархическое представление основных понятий в разрезе таких отношений как «A состоит из B1, B2, ...» или «B — это частный случай A». В настоящее время иерархические онтологии — одно из основных направлений автоматизации хранения, использования и накопления знаний [10-12]. В разделе описаны наши попытки использования метода АСД для отображения текстов в таксономиях математики и информатики, одна – англоязычная (информатика), вторая – русскоязычная (математика).

3.1. Индексирование научных статей таксономическими единицами классификации информатики ACM-CCS

Одно из возможных приложений метода АСД – индексирование научных публикаций по существующим научным классификациям. Например, в журналах международной Ассоциации Вычислительной Техники (Association for Computing Machinery) используется классификационная система вычислительной техники, разработанная этой организацией [13] (ACM-CCS), для индексирования (рубрикации) статей. Авторы вручную приписывают своим статьям две-три таксономические единицы ACM-CCS, наилучшим образом отвечающие им по содержанию. Нас интересует возможность использования метода АСД для автоматизации такого индексирования.

Чтобы реализовать эту идею, для каждой рассматриваемой научной публикации следует:

- ✦ выделить ее ключевые фрагменты, включая заголовок, список ключевых слов и аннотацию (abstract);
- ✦ построить АСД по выделенным фрагментам публикации (по всем или частично);
- ✦ оценить степень вхождения каждой листовой таксономической единицы ACM-CCS в построенное АСД, т.е. построить профиль статьи;
- ✦ выбрать таксономические единицы с максимальными оценками в качестве искомой индексации.

Рассмотрим один из журналов, Journal of the ACM, издаваемый в электронном формате и находящийся в свободном доступе. Для ускорения расчетов использованы только текст аннотации и ключевые слова статей. В *табл. 3* приведены аннотации, таксономические единицы, приписанные авторами, и списки ключевых слов для публикаций [14] и [15]. Это сделано для того, чтобы читатель смог оценить сам, насколько аннотации, полученные с использованием метода АСД и приведенные далее в *табл. 4* и *5*, соответствуют содержанию. Кроме того, приведенные тексты могут быть использованы как данные для тестирования других методов анализа текстов.

Следует иметь в виду, что, несмотря на внешнее сходство с задачей распознавания образов, в данной проблеме нет внешнего учителя. Поэтому не существует объективного измерителя степени успешности работы автоматического индексатора, по крайней мере, в настоящее время.

В левой части *таблиц 4* и *5* представлены АСД-профили, а в правой – авторские аннотации, а также места, которые авторские таксономические единицы заняли в АСД-профиле.

Профиль *табл. 4* представляется вполне удачным, поскольку таксономические единицы из авторского индекса статьи занимают 3 и 4 место в АСД-профиле. Напротив, профиль *табл. 5* – неудачный: авторский индекс статьи содержит таксономические единицы, крайне низко оцененные АСД-профилем. Дело в том, что авторские индексации содержат таксономические единицы, формулировки которых не отражены в тексте аннотации – они передаются другими, синонимичными словами.

Приведенный пример показывает трудности, связанные с методом АСД:

✦ неоправданно высокая оценка общих слов и фраз. Эту проблему отчасти можно решить путем введения списка стоп-слов, включающего все подобные слова, «вручную»;

✦ таксономическая единица получает низкую оценку, если автор предпочитает использовать другое, хотя и близкое по смыслу, понятие.

Эта проблема может быть решена, если с каждой таксономической единицей связать список синонимичных понятий.

3.2. Анализ учебных программ математического цикла НИУ ВШЭ с использованием таксономии РЖ «Математика»

На сайте НИУ ВШЭ имеются в свободном доступе файлы программ различных курсов, относящихся к математическим дисциплинам и читаемым студентам различных специализаций. Естественный вопрос – как эти программы отражают современную математику – может трактоваться как возможность выделения основных кластеров математического знания, содержащихся в этих программах, и их отображения на иерархическую классификацию математики. Поскольку учебные программы написаны на русском языке, в качестве классификации математики мы взяли русскоязычный иерархический рубрикатор реферативного журнала РЖ «Математика» (в настоящее время — на сайте [14]; мы использовали более ранний вариант рубрикатора, доступный в 2010 г., когда проводились расчеты).

Приведем некоторые из полученных результатов. В *табл. 6* представлены таксономические единицы, получившие максимальные оценки вхождения в программу курса «Дискретная математика». Часть таксономических единиц, получивших высокие оценки по методу АСД, оказались не адекватными содержанию учебной программы. Три из представленных в *табл. 6* неадекватных таксономических единиц содержат слово «алгебраический». Это слово или однокоренные с ним часто употребляются в программе «Дискретная математика», поэтому данные таксономические единицы и получили высокие оценки.

Хороший профиль оказался у программы «Дифференциальные уравнения»: он включает в себя большую часть соответствующего раздела таксономии и почти полностью покрывает содержание программы.

Таблица 3.

Аннотации, индексные таксономические единицы и ключевые слова двух публикаций журнала ACM

M. Bojanczyk, A. Muscholl, T. Schwentick, L. Segoufin	M. Grohe, A. Henrich, N. Schweikardt
Two variable logic on data trees and XML reasoning, Journal of ACM, 2009, 56(3), 58 p.	Lower bounds for processing data with few random accesses to external memory, Journal of ACM, 2009, 56(3), 48 p.
Motivated by reasoning tasks for XML languages, the satisfiability problem of logics on data trees is investigated. The nodes of a data tree have a label from a finite set and a data value from a possibly infinite set. It is shown that satisfiability for two-variable first-order logic is decidable if the tree structure can be accessed only through the child and the next sibling predicates and the access to data values is restricted to equality tests. From this main result, decidability of satisfiability and containment for a data-aware fragment of XPath and of the implication problem for unary key and inclusion constraints is concluded.	We consider a scenario where we want to query a large dataset that is stored in external memory and does not fit into main memory. The most constrained resources in such a situation are the size of the main memory and the number of random accesses to external memory. We note that sequentially streaming data from external memory through main memory is much less prohibitive. We propose an abstract model of this scenario in which we restrict the size of the main memory and the number of random accesses to external memory, but admit arbitrary sequential access. A distinguishing feature of our model is that it allows the usage of unlimited external memory for storing intermediate results, such as several hard disks that can be accessed in parallel. In this model, we prove lower bounds for the problem of sorting a sequence of strings (or numbers), the problem of deciding whether two given sets of strings are equal, and two closely related decision problems. Intuitively, our results say that there is no algorithm for the problems that uses internal memory space bounded by $N^{1-\varepsilon}$ and at most $o(\log N)$ random accesses to external memory, but unlimited «streaming access», both for writing to and reading from external memory. (Here N denotes the size of the input and ε is an arbitrary constant greater than 0.) We even permit randomized algorithms with one-sided bounded error. We also consider the problem of evaluating database queries and prove similar lower bounds for evaluating relational algebra queries against relational databases and XQuery and XPath queries against XML-databases.
Primary Classification F.4.1[Mathematical logic and formal languages]:Mathematical logic Additional Classification: F.2.3[Database management]: Languages–Query languages	Primary Classification F.1.1 [Computation by Abstract Devices]: Models of Computation—bounded-action devices; F.1.3 [Computation by Abstract Devices]: Complexity Measures and Classes—relations among complexity classes; relations among complexity measures; Additional Classification: H.2.4 [Database Management]: Systems—query processing; relational databases
General Terms: Theory Key Words and Phrases: First-order logic, data trees, decidability	General Terms: Theory, Languages Key Words and Phrases: complexity, data streams, real-time data, query processing, query optimization, semi-structured data, XML

Таблица 4.

Пример «удачного» АСД профиля

Статья: Bojanczyk M. et al. Two variable logic on data trees and XML reasoning, Journal of the ACM, 2009, Vol. 56(3). 2-48.					
АСД Профиль			Индексационные таксономические единицы (ручное аннотирование)		
ID	TE	ACM–CCS единица	ID	#	ACM–CCS категории
I.6.2	6.1969	Simulation Languages	F.4.3	3	Formal Languages
I.1.3	6.1415	Languages and Systems	H.2.3	4	Languages
F.4.3	6.0796	Formal Languages	H.2.1	13	Logical Design
H.2.3	4.0757	Languages	F.4.1	28	Mathematical Logic
D.4.5	3.7387	Reliability	I.7.2	53	Document Preparation

Таблица 5.

Пример «неудачного» АСД профиля

Статья: Grohe M., et al. Lower bounds for processing data with few random accesses to external memory. Journal of the ACM, 2009, Vol. 56(3), 1-58.					
AST found profile			ACM-CCS index terms (manual annotation)		
ID	TE	ACM–CCS класс	ID	#	ACM–CCS класс
J.1	9.5991	ADMINISTRATIVE DATA PROCESSING	F.1.3	161	Complexity Measures and Classes
I.2.7	7.3757	Natural Language Processing	H.2.4	166	Systems
H.2.5	5.0704	Heterogeneous Databases	F.1.1	220	Models of Computation
H.2.8	4.4196	Database Applications			
C.5.1	4.0146	Large and Medium Computers			

В целом, результаты этого приложения вызывают больше вопросов, чем дают ответов. Это связано, на наш взгляд, не только с вышеотмеченными недостатками метода АСД, но и особенностями использованной таксономии. Использованная версия таксономии РЖ «Математика» (2009) — это дерево неравномерной глубины (от 3 до 8 уровней в разных разделах), содержащее разделы, не сбалансированные между собой по объему и содержанию. Кроме того, в таксономии отсутствуют некоторые современные темы такие как «Дискретная математика» или «Математическая экономика». В разделах, относящихся к современным частям математики, часто опущены важные понятия. Например, категория «Теория игр» содержит таксономические единицы, перечисляющие виды игр, но понятие «равновесие» здесь не представлено. Напротив, имеются разделы, связанные с относительно небольшими частями математики, особенно с точки зрения учебных программ, которым соответствуют глубокие и кустистые поддеревья. Кроме того, в таксономии усложнена система навигации; есть ссылки на удаленные после последней модификации таксономические единицы. Это делает актуальной задачу разработки более адекватной классификации математики, включая, вероятно, информатику и прикладную математику.

Заключение

Методы анализа текстов, основанные на аннотированных суффиксных деревьях (АСД), удобны тем, что не связаны с необходимостью грамматического разбора фраз, т.е. с особенностями того или иного языка, и, более того, вообще позволяют свести до минимума предварительную обработку текстов, составляющую важную и иногда трудоемкую часть других подходов. Путем разбивки текста на строки нам удалось значительно снизить вычислительную трудоемкость метода. Рассмотренные примеры, с одной стороны, показывают эффективность метода в правильно выбранных приложениях и, с другой стороны, показывают пути его дальнейшего совершенствования. Главный недостаток метода – существенная привязка к буквенному содержанию фрагментов текста, от чего, вероятно,

Таблица 6.

Часть профиля программы «Дискретная математика»**

Программа по курсу «ДИСКРЕТНАЯ МАТЕМАТИКА»		
Оценка	Код	Таксономическая единица
25.53	517.986.9	Другие алгебраические структуры в функциональном анализе
18.096	512.562	Частично упорядоченные множества
17.205	512.664.8	Деформации алгебраических структур
16.204	512.732.1	Общие свойства алгебраических пучков
16.025	510.63	Классические логические теории
15.307	512.545.6	Частично упорядоченные группы
12.076	510.6	Математическая логика
11.267	510.64	Неклассические логики
11.253	519.171.2	Представления графов
11.049	519.876.3	Сетевое планирование
10.96	519.712.4	Сложность алгоритмов
10.96	510.52	Сложность алгоритмов
8.31	510.633	Логика высказываний
7.36	519.681.4	Сложность вычислений

можно освободиться путем добавления информации о шумовых словах и словах-синонимах.

Работа частично выполнялась в рамках научно-исследовательского проекта «Учитель-Ученики: 11-04-0019 Разработка и адаптация методов кластер-анализа для автоматизации анализа текстовой информации с использованием онтологии предметной области», поддержанного Программой «Научный фонд НИУ-ВШЭ» в 2011-2012 гг., а также Международной лаборатории по выбору и анализу решений НИУ-ВШЭ. Авторы благодарят руководителя лаборатории Ф.Т. Алескерова за советы и помощь в работе. Часть проекта осуществлялась при финансовой поддержке Министерства образования и науки Российской Федерации в рамках договора N 13.G25.31.0033 от 7 сентября 2010 г., заключенного между Министерством образования и науки Российской Федерации и ЗАО «Авикомп Сервисез». Авторы благодарны анонимному рецензенту, работа над замечаниями которого позволила улучшить изложение. ■

Литература

1. Gusfield D. Algorithms on Strings, Trees, and Sequences. – Cambridge University Press, 1997.
2. Huang C., Yin J., Hou F. Text clustering using a suffix tree measure //Journal of Computers. – 2011. – Vol 10(6). – P. 2180-2186.
3. Pampapathi R., Mirkin B., Levene M. A suffix tree approach to anti-spam email filtering // Machine Learning. – 2006. – Vol. 65(1). – P. 309-338.
4. Manning C., Schütze H. Foundations of Statistical Natural Language Processing. – MIT Press, 1999.
5. Zamir O, Etzioni. O. – Web document clustering: A feasibility demonstration – Proceedings of SIGIR'98, University of Washington, Seattle, USA, 1998.
6. Stepp R., Michalski R.S. Conceptual clustering of structured objects: A goal-oriented approach// AI Journal. – 1986. – Vol. 28. – P. 43-69.
7. Mirkin B. Clustering for Data Mining: A Data Recovery Approach. – Chapman & Hall/CRC, 2005.
8. Han J., Kamber M., Pei J. Data Mining: Concepts and Techniques. – Third Edition. – Morgan Kaufmann, 2011.
9. Чесноков С.В. Детерминационный анализ социально-экономических данных. – 2-е изд. – М.: URSS, 2009.
10. Robinson P.N., Bauer, S. Introduction to Bio-Ontologies. – Chapman & Hall/CRC, 2011.
11. Лукашевич Н.В. Тезаурусы в задачах информационного поиска. – М.: Изд-во Московского университета, 2011.
12. Онтологическое моделирование экономики предприятий и отраслей современной России: Российские исследования и разработки в области онтологического инжиниринга и бизнес-онтологий. – Препринт WP7/2011/08 [Текст] / Ефименко И.В., Хорошевский В.Ф. – Нац. исслед. ун-т «Высшая школа экономики». – М.: Изд. дом Высшей школы экономики, 2011.
13. ACM Computing Classification System, 1998. URL: http://www.acm.org/about/class/1998 (дата обращения: 09.09.2010).
14. Рубрикатор РЖ «Математика». URL: http://www.viniti.ru/russian/math/files/271.htm (дата обращения 15.05. 2012).

** Серым фоном выделены неадекватные таксономические единицы