Ivanova A.V., Aduenko A.A., Strijov V.V. Algorithm of construction logical rules for text segmentation // Software Engineering, 2013, 6 : 41-48.

УДК 519.256

Алгоритм построения логических правил при разметке текстов[1]

А. В. Иванова, А. А. Адуенко, В. В. Стрижов

Аннотация. В работе предложен метод восстановления структуры библиографических записей BibTeX по их текстовому представлению. Структура восстанавливается с помощью логических правил, определенных на экспертно-заданном множестве регулярных выражений. Для построения набора логических правил предложен алгоритм, использзующий тупиковые покрытия. Предложенный алгоритм проиллюстрирован задачей поиска структуры библиографических записей, представленных набором текстовых строк.

Ключевые слова: библиографичесая запись, BibTeX, регулярное выражение, тупиковое покрытие, кластеризация.

1 Введение

Задан набор текстовых строк, которые составлены в соответствии с некоторыми правилами, например, определен порядок следования информации в строках и способ форматирования. Требуется решить задачу восстановления правил, по которому был построен набор строк. Для различных типов строк и правил, по которым они построены, предложены разные способы решения задачи [1, 2]. В данной работе в качестве текстовых строк рассматриваются библиографические записи, которые строятся по существующим стандартам (ГОСТ 7.82-2001, ISO 690-2:1997, MLA). Однако разные стандарты определяют разный порядок следования полей библиографической записи и разный способ форматирования полей. Более того, библиографическая запись может содержать ошибки.

В работе рассматривается задача выделения сегментов

[1] Работа выполнена при поддержке Министерства образования и науки РФ в рамках Государственного контракта 07.524.11.4002.

библиографических записей и их разметка – определение соответствия между текстовыми сегментами библиографической записи и полями структуры BibTeX. В работе рассматриваются следующие поля структуры BibTeX: автор или авторы; название; журнал; страницы; том; номер; год; город; издательство; редакторы; ссылка в Интернет.

Кроме того, наряду с неразмеченной коллекцией — набором библиографических записей, которые требуют разделения на сегменты и дальнейшей разметки этих сегментов, имеется размеченная коллекция библиографических записей, для которой сегментация и разметка уже произведены. Таким образом, требуется провести выделение сегментов и разметку библиографических записей, то есть классификацию полученных сегментов на классы, соответствующие полям структуры BibTeX, располагая уже размеченной коллекцией. В работе эта задача рассмотрена как задача кластеризации [3]. При этом каждый класс, соответствующий полю структуры BibTeX, описывается несколькими кластерами. При наличии размеченной коллекции применимы методы частичного обучения [4], однако в данной работе размеченная коллекция используется лишь для интерпретации получившихся кластеров, то есть определения, какой класс описывается рассматриваемым кластером.

Для проведения сегментации и кластеризации сегментов [5] порождались признаки объектов [6]. В качестве объектов использовались позиции в тексте библиографической записи. В качестве признаков использовались регулярные выражения [7], задающие шаблоны текстовых строк. Значение призака позиции равняется единице, если библиографическая запись, начиная с этой позиции, соответствует шаблону. В противному случае это значение равняется нулю. Таким образом была порождена бинарная матрица объектов – признаков [6].

В алгоритме кластеризации пошагово строятся кластеры сходных объектов. Для определения меры сходства [8] используются тупиковые

покрытия [9]. Для их построения применяется алгоритм, изложенный в [9]. После кластеризации каждому кластеру ставится в соответствие класс — поле библиографической записи, которое он описывает. Соответствие устанавливается с помощью размеченной коллекции. Кластер относится к тому классу, объектов которого в нем наибольшее число из размеченной коллекции.

2 Постановка задачи и алгоритм кластеризации

Библиографические записи представлены текстовой строкой T, где $t = |T|$ — число символов в строке. В качестве объектов используются позиции в строке, а для построения признаков используется экспертно заданное множество Θ регулярных выражений мощности $\theta = |\Theta|$. Построим матрицу $\mathbf{D}$ размеров $t \times \theta$ следующим образом:

$$\mathbf{D}(i, j) = \begin{cases} 1, & \text{если на позиции } i \text{ сработало регулярное выражение } j, \\ 0, & \text{в противном случае.} \end{cases} \quad (1)$$

Обозначим p – ый столбец единичной матрицы размеров $r \times r$ как $\mathbf{e}_p^r$.

Для описания алгоритма кластеризации требуется ввести понятия покрытия матрицы и тупикового покрытия.

Определение. Пусть $\mathbf{L}$ – матрица размеров $m \times n$. Множество H, состоящее из r столбцов матрицы $\mathbf{L}$, $r \in \mathbb{N}, r \le n$, назовем *покрытием* матрицы $\mathbf{L}$, если матрица, образованная столбцами из H, не содержит нулевую строку. Покрытие H матрицы $\mathbf{L}$ назовем *тупиковым*, если для каждого $p \in \{1,\ldots,r\}$ найдется столбец $\mathbf{h} \in H$, который содержит $\mathbf{e}_p^r$, то есть $h_1 = \ldots = h_{p-1} = h_{p+1} = \ldots = h_r = 0, h_p = 1$.

Заметим, что если рассмотреть построенную согласно (1) матрицу $\mathbf{D}$, то в силу того, что каждый сегмент библиографической записи содержит значительное число позиций и лишь некоторые из них характеризуют этот сегмент и отличают его от остальных, в матрице $\mathbf{D}$ есть множество нулевых

строк, соответствующих таким неинформативным позициям в тексте записи. Отсюда получим, что покрытия у исходной матрицы $\mathbf{D}$ не существует. Поэтому будем рассматривать подматрицу $\hat{\mathbf{D}}$ матрицы $\mathbf{D}$, полученную из нее исключением всех нулевых строк.

Введем следующие обозначения:

$\mathbf{\Omega}$ — некоторая подматрица матрицы $\mathbf{D}$,

$\boldsymbol{\sigma}^T(\mathbf{\Omega})$ — множество тупиковых покрытий матрицы $\mathbf{\Omega}$.

Для двух подматриц $\mathbf{\Omega}_f$ и $\mathbf{\Omega}_e$ матрицы $\mathbf{D}$ с одинаковым числом столбцов определим подматрицу $\boldsymbol{\sigma}^T(\mathbf{\Omega}_f \,|\, \mathbf{\Omega}_e)$ — множество тупиковых покрытий матрицы $\mathbf{\Omega}_f$, которые также являются тупиковыми и для соединенной матрицы $\begin{bmatrix}\mathbf{\Omega}_f \\ \mathbf{\Omega}_e\end{bmatrix}$.

Создадим матрицу $\mathbf{\Omega} = \hat{\mathbf{D}}$, полученную из $\mathbf{D}$ исключением всех нулевых строк. Для описания алгоритма кластеризации требуется ввести меру сходства. Предлагаемый алгоритм является пошаговым, причем на i-м шаге рассматривается очередной объект $\mathbf{x}_i$, соответствующий некоторой строке $\mathbf{\Omega}$. Он либо относится к рассматриваемому на i-м шаге кластеру M_j, либо образует новый кластер M_{j+1}. Поэтому требуется определить меру сходства не между объектами, а между объектом и кластером объектов, то есть набором объектов. Для этого воспользуемся понятием тупикового покрытия и в качестве меры сходства или близости очередного объекта $\mathbf{x}_i$ кластеру M_j будем рассматривать функцию

$$P(\mathbf{\Omega}_f, \mathbf{x}_i) = \frac{|\sigma^T(\mathbf{\Omega}_f \,|\, \mathbf{x}_i)|}{|\sigma^T(\mathbf{\Omega}_f)|}. \qquad (2)$$

Здесь $\mathbf{\Omega}_f$ – матрица, составленная из признаковых описаний объектов, входящих в рассматриваемый кластер M_j. Заметим, что $P(\mathbf{\Omega}_f, \mathbf{x}_i) \le 1$ и эта величина определяет сходство объекта $\mathbf{x}_i$ с кластером M_j. Чем $P(\mathbf{\Omega}_f, \mathbf{x}_i)$ больше,

тем ближе объект $\mathbf{x}_i$ к кластеру M_j. В числителе формулы (2) находится мощность множества тупиковых покрытий соединенной матрицы $\begin{pmatrix} \mathbf{\Omega}_f \\ \mathbf{x}_i \end{pmatrix}$, а в знаменателе мощность множества тупиковых покрытий матрицы $\mathbf{\Omega}_f$, составленной из признаковых описаний объектов, входящих в кластер M_j. Опишем алгоритм кластеризации подробнее. Для удобства обозначим Ω множество объектов, признаковое описание которых есть строки матрицы $\mathbf{\Omega}$. Зададим некоторый порог $p \in [0,1]$.

Шаг 1. Положим, что кластер M_1 включает единственный элемент $\mathbf{x}_1$, то есть $M_1 = \{\mathbf{x}_1\}$, где $\mathbf{x}_1$ – объект, описанный первой строкой $\mathbf{\Omega}$. Если в матрице $\mathbf{\Omega}$ ровно одна строка, то закончить процедуру, иначе перейти к шагу 2.

Шаг 2. Пусть построены классы $M_1,\ldots M_{j-1}, j \geq 2$. Обозначим множество

$$\tilde{\Omega}_{j-1} = \Omega \setminus \bigcup_{k=1}^{j-1} M_k.$$

Если это множество пусто, то все объекты из множества Ω уже кластеризованы, поэтому заканчиваем работу. Иначе определяем объект $\mathbf{x}' \in \tilde{\Omega}_{j-1}$ такой, что

$$\mathbf{x}' = \underset{\mathbf{x} \in \tilde{\Omega}_{j-1}}{\arg\max}\, P(M_{j-1}, \mathbf{x}). \qquad (3)$$

То есть находим элемент, наиболее близкий к текущему рассматриваемому кластеру. Если $P(M_{j-1}, \mathbf{x}') \geq p$, то $M_{j-1} = M_{j-1} \bigcup \{\mathbf{x}'\}$, где $\mathbf{x}'$ определен (3), иначе начинаем добавлять объекты к новому классу M_j :$M_j = \{\mathbf{x}'\}$.

Алгоритм заканчивает работу, когда все объекты из множества Ω будут распределены по кластерам.

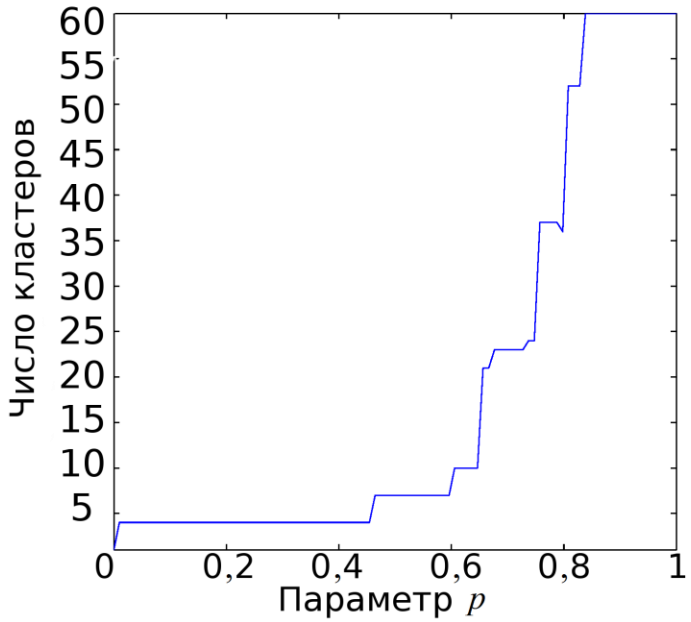

Рис. 1: Зависимость числа кластеров от значения параметра p.

Интерпретация кластеров. Полученные кластеры требуется проинтерпретировать, то есть указать для каждого из них класс, который он описывает, иначе – указать, к какому полю библиографической записи отнести элементы, в нем находящиеся. Для решения этой задачи используется размеченная коллекция. Кластер относится к тому классу, объектов которого в кластере из размеченной коллекции больше, чем в прочих кластерах.

3 Вычислительный эксперимент

Проиллюстрируем предложенный алгоритм на коллекции библиографических записей, представленных в виде текстовых строк. В коллекции 1000 библиографических записей, из которых 100 размечены, то есть каждому предварительно выделенному сегменту из этих 100 строк поставлена в соответствие метка типа записи. Каждый сегмент принадлежит одному из 11 типов: автор или авторы, название, журнал, страницы, том, номер, год, город, издательство, редакторы, ссылка в интернет. Требуется, пользуясь уже

размеченной коллекцией, спрогнозировать разметку оставшейся ее части. Размеченная коллекция использовалась лишь для интерпретации полученных при разметке кластеров сегментов записей.

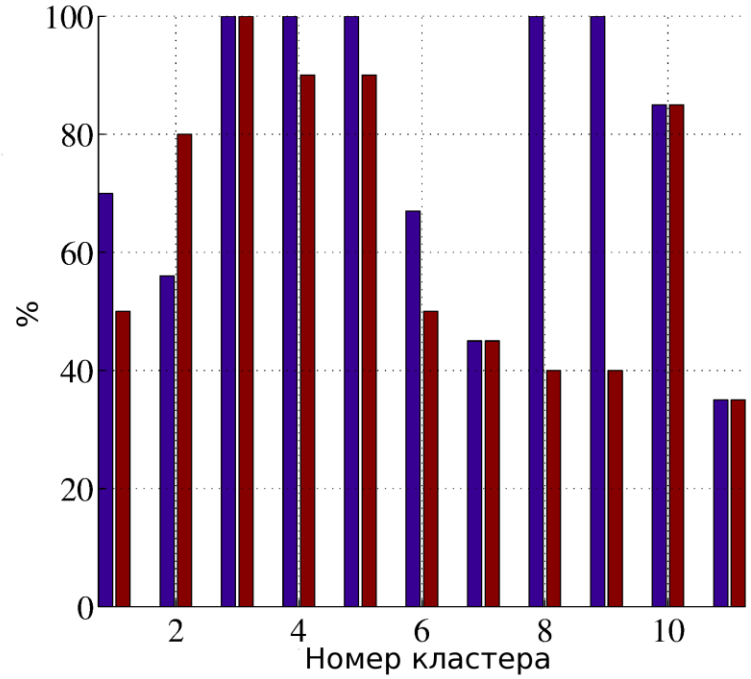

Рис. 2: Интерпретация кластеров при $p = 0.83$.

Для всей коллекции, включая ее размеченную часть, была составлена матрица $\mathbf{D}$ (1). Приведем примеры регулярных выражений, которые использовались для построения матрицы $\mathbf{D}$: наличие четырех цифр подряд, начиная с рассматриваемой позиции, наличие запятой, заглавной буквы, заглавной буквы и точки за ней и т.д. На всей коллекции применялся алгоритм кластеризации с параметром p из отрезка $[0,1]$. График зависимости числа выделяемых кластеров от значения этого параметра приведен на рис. 1. При малых $p = 1$ число выделяемых кластеров очень мало, напротив — при $p \approx 1$ число кластеров стабилизируется на 60 кластерах. То, что число кластеров при $p \approx 1$ не равно полному числу сегментов, то есть не исчисляется тысячами,

говорит о том, что набор использовавшихся признаков не является избыточным, то есть позволяет указать и очень близкие объекты.

При $p = 0.83$ число кластеров равно 35. Из них 24 являются шумовыми, то есть малочисленными и не имеющими элементов из размеченной коллекции. Оставшиеся 11 кластеров поддаются интерпретации с помощью размеченной коллекции. Каждый кластер отнесен к тому типу поля библиографической записи, элементов которого в нем наибольшее число среди записей размеченной коллекции. Кластеры 1-2 соответствуют полю авторы, кластеры 3-4 соответствуют инициалам автора, кластер 5 соответствует ссылкам в интернет, кластеры 6-7 соответствуют названиям журналов, кластеры 8-9 соответствуют названиям статей, а кластеры 10-11 — годам выпуска статей или книг. На рис. 2 приведены свойства этих 11 кластеров. Каждому кластеру на диаграмме соответствует два столбца. Высота левого столбца показывает долю (в процентах) записей из размеченной коллекции, попавших в кластер и относящихся к тому же типу поля библиографической записи, к которому отнесен и сам кластер. Высота правого столбца показывает ту же долю для неразмеченной коллекции.

Опишем некоторые получившиеся кластеры подробнее. Кластер 7 хотя и отнесен к полю названия журналов содержит почти одинаковое количество названий журналов и названий статей. Поскольку одно от другого отделить, пользуясь только самим написанием, зачастую невозможно, результат вполне закономерен. Возможно, что для улучшения работы алгоритма можно использовать некоторый словарь, содержащий список названий журналов. Кластер 11 отнесен к полю библиографической записи «год», однако туда попало значительное число полей «том», «номер журнала», «страницы». Это так же объясняется тем, что различить указанные поля не представляется возможным. Кластер 10, соответствующий тому же полю «год» отличается от кластера 11 тем, что за номером года в сегментах, отнесенных к этому кластеру,

есть запятая, а количество прочих числовых полей с запятой после номера значительно меньше. Значительное отличие результатов в кластерах 8 и 9 для размеченной и неразмеченной коллекции объясняется тем, что в размеченной коллекции внешние знаки препинания были исключены, а потому в ней не встречались сегменты вида «автор,», «город,» или «издательство,», что часто встречается в неразмеченной коллекции и мешает выделению названий статей, содержащих запятые и прочие знаки препинания. В табл. 1. приведены примеры верно и неверно отнесенных к некоторым классам сегментов по кластерам.

Таблица 1: Верно и неверно классифицированные сегменты текста для различных кластеров.

Номер и класс кластера	Классифицированные сегменты	
	Верно	Неверно
1: авторы	Hand, D.J. D.N.Potts F.X. Le Dimet	Comput. Syst. U.K. Shalev-Shwartz (Eds.)
2: авторы	Shawe-Taylor Kai-Min Chung MacKay	Prentice-Hall Springer-Verlag TD-Gammon
7: названия журналов	SIAM Journal on computing Neural computation ACM Transactions on Mathematical Software	Sparse matrix test problems Interior-point method for convex programming A trust region algorithm for nonlinearly constrained optimisation
8: названия	A QP-free globally convergent,	Tibshirani,

статей	locally superlinearly convergent algorithm for inequality constrained Learning from noisy examples The hardness of approximate optima in lattices, codes and systems of linear equations	USA MIT Press

Значительное число ошибок алгоритма объясняется малым размером набора регулярных выражений и тем, что этот набор не учитывает специфику задачи разметки библиографических записей.

Далее расширим набор регулярных выражений с учетом специфики задачи. В набор будут добавлены регулярные выражения, улучшающие выделение отдельных полей библиографических записей (см. табл. 2). Далее C обозначаем некоторое число, а d-цифру.

Таблица 2: Добавленные регулярные выражения для разных полей структуры BibTeX.

Поле структуры BibTeX	Добавленные регулярные выражения
Том	“Vol”; ”vol”; C(C); C:C
Номер	“No.”; “no.”; C(C); C:C
Страницы	“Pp”; “pp.”; “pages”; “p.”; “page”; C-C
Год	20dd; 19dd
Журнал	“Journal”; “journal’
Издательство	“Publish”; “publish”; “Press”; “press”
Редакторы	“Eds”; “eds”; “ed”; “editor”; “editors”; “edited”; “Editor”; “Editors”

Результат работы предложенного алгоритма с расширенным множеством регулярных выражений на той же коллекции библиографических строк в текстовом представлении сравнивался с результатом работы онлайн-программы восстановления структуры Brown University (http://freecite.library.brown.edu/). Каждому полю библиографической записи на диаграмме на рис. 3 соответствует три столбца.

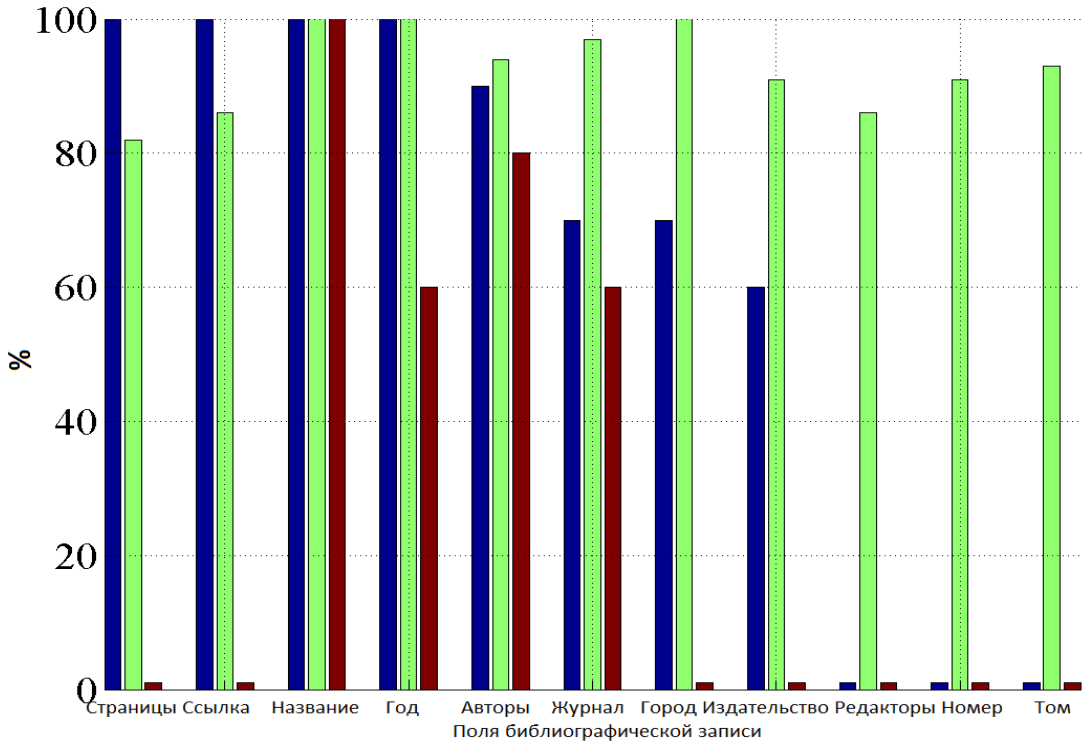

Рис. 3 Доля верно кластеризованных строк для каждого типа библиографической записи для разных алгоритмов

Высота левого столбца определяет долю записей из размеченной коллекции, попавших в кластер и относящихся к тому же типу поля библиографической записи, к которому отнесен и сам кластер для предложенного в работе алгоритма. Высота среднего и правого столбца определяет то же для онлайн-программы, с которой происходит сравнение, и для алгоритма, предложенного в работе, при использовании исходного

множества регулярных выражений.

Диаграмма демонстрирует значительное улучшение качества работы алгоритма после добавления новых регулярных выражений. Так строки, соответствующие полям «страницы» и «год», кластеризованы без ошибок. Улучшение качества для поля «город» можно объяснить улучшением общего качества кластеризации строк, соответствующим тем полям, с которыми и происходило смешивание строк, соответствующим полю город: издательство, название журнала. Отсутствие кластера, соответствующего редакторам объясняется малочисленностью записей, содержащих это поле. Ошибки в полях «том» и «номер» объясняются наличием регулярных выражений, которые срабатывают и для поля «страницы» и для этих двух полей. Более аккуратный выбор регулярных выражений для указанных трех типов полей библиографической записи позволит решить и эту проблему. Полученные результаты сопоставимы с результатами онлайн-программы, что свидетельствует в пользу использовавшегося подхода.

4 Заключение

В данной работе решается задача разметки библиографических данных. Используется метод метрической кластеризации, при котором расстояние вводится с помощью тупиковых матриц. В качестве признаков исходных объектов использовалось срабатывание регулярных выражений. В вычислительном эксперименте предложенный алгоритм тестировался на данных библиографических записей, часть которых была размечена. Полученные результаты говорят о применимости метода и указывают на возможные его модификации для улучшения разметки библиографических записей.

Список литературы

[1] *Borkar V., Deshmukh K., Saravagi S.* Automatic segmentation of text into structured records. // Proceedings of the 2001 ACM SIGMOD international conference on management of data. New York: ACM, 2001. Vol. 30. No. 2. Pp. 175–186.

[2] *Christen P., Churches T., Zhu J. X.* Probabilistic name and address cleaning and standardisation. // The Australasian data mining workshop, 2002. С. 99-108.

[3] *Адуенко А. А., Кузьмин А. А., Стрижов В. В.* Выбор признаков и оптимизация метрики при кластеризации коллекции документов. // Известия ТулГу, 2012. №3. С. 119–131.

[4] *Chapelle O., Schölkopf B., Zien A.* Semi-supervised learning. // Cambridge: The MIT Press, 2006.

[5] *Aliguliyev R. M.* Performance evaluation of density-based clustering methods. // Information sciences, 2009. Vol. 179. No. 20. Pp. 3583–3602.

[6] *Gabrilovich E., Markovitch S.* Overcoming the brittleness bottleneck using wikipedia: enhancing text categorization with encyclopedic knowledge. // Proceedings of the National Conference on Artificial Intelligence, Menlo Park, CA, 2006. Vol. 21, No 2. С. 1301.

[7] *Ахо А., Хопкрофт Дж., Ульман Дж.* Построение и анализ вычислительных алгоритмов. // М.: Мир, 1979. С. 535.

[8] *Zhao Y., Karypis G.* Criterion function for document clustering: experiments and analysis // Machine Learning, 2001. Vol. 55, No 3. P. 311—331.

[9] *Дюкова Е. В., Инякин А. С.* Задача таксономии и тупиковые покрытия целочисленной матрицы. М.: Вычислительный центр РАН, 2001.

A.V. Ivanova, Moscow Institute of Physics and Technology

A. A. Aduenko, Moscow Institute of Physics and Technology

V. V. Strijov, Computing Center of the Russian Academy of Sciences

Algorithm of construction logical rules for text segmentation

Consider the method of recovery of BibTeX-structure bibliographic records from their text representation. Structure is recovered using logical rules defined on an expert-given set of regular expressions. Algorithm based on stub covers is proposed for constructing the logic rules. The algorithm is illustrated with the problem of searching the structure in bibliographic records, represented by text strings.

Keywords: bibliographic record, BibTeX, regular expression, stub cover, clustering.