Покрытие этого многосвязного многоугольника, полученное посредством данного выше алгоритма, описывается следующей последовательностью строк:

(7, 22), (14, 22), (14, 19), (7, 19);
(7, 16), (7, 22), (12, 22), (12, 16);
(20, 15), (20, 10), (18, 10), (18, 15);
(15, 3), (11, 3), (11, 5), (15, 5);
(3, 3), (3, 7), (5, 7), (5, 3);
(3, 7), (5, 13), (6.2, 12.6), (4.2, 6.6);
(14, 22), (20, 15), (19.34, 14.44), (13.34, 21.44);
(15, 14.5), (15, 3), (14, 3), (14, 14.5);
(20, 10), (14, 10), (14, 13), (20, 13);
(12, 5), (12, 10), (9.67, 10), (9.67, 5);
(5, 13), (9, 13), (9, 12), (5, 12);
(8, 10), (8, 22), (9, 22), (9, 10);
(12, 6), (6, 6), (6, 7.5), (12, 7.5);
(11, 12), (14, 12), (14, 15), (11, 15);
(7, 19), (15, 19), (15, 20.83), (7, 20.83);
(18, 16), (18, 10), (19.14, 10), (19.14, 16);
(8, 14), (12, 14), (12, 22), (8, 22);
(6, 12), (11, 7), (12, 8), (7, 13);
(12.62, 21.38), (18, 16), (18.62, 16.62), (13.2, 22);
(6.2, 6.6), (5, 3), (3, 3.67), (4,2, 7.27);
(11, 10), (3, 6), (3.6, 4.8), (11.6, 8.8);
(14.8, 13.4), (10.5, 22), (7.7, 20.6), (12, 12);
(4.25, 3.25), (6, 12), (4.75, 12.25), (3, 3.5);
(19.2, 12.4), (12.2, 15.9), (11, 13.5), (18, 10);
(11, 3), (7.62, 8.08), (9.23, 9.15), (12.62, 4.08).

В этой последовательности каждая строка задает координаты вершин прямоугольника, входящего в покрытие. Координаты вершины, представленные парой чисел, заключены в круглые скобки. Первое из этих чисел задает значение по оси *OX*, второе – значение по оси *OY*.

Заключение. Описанный в настоящей работе эвристический метод покрытия произвольных многосвязных многоугольников прямоугольниками запрограммирован на языке C++. Проведены испытания этой программы на примерах практической сложности. Как правило, покрытия, полученные посредством этой программы, являются корректными, т. е. объединение полученных прямоугольников не содержит иных контуров-разрезов, чем исходный многоугольник. Проверка корректности выполняется по методу, приведенному в работе [7].

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Фейнберг, В.З. Геометрические задачи машинной графики больших интегральных схем. – М.: Радио и связь, 1987. – 178 с.
2. Hegedus A., Algorithms for covering polygons by rectangles, Computer Aided Design, vol. 14, no 5, 1982.
3. Asano Ta., Asano Te., and Imai H. Partitioning a polygonal region into trapezoids. J. ACM, 33:290-312, 1986.
4. Ferrari L., Sankar P.V., and Sklansky J. Minimal rectangular partitions of digitized blobs. Computer Vision, Graphics, and Image Processing, 28:58-71, 1984.
5. Nahar S. and Sahni S. Fast algorithm for polygon decomposition. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 7:473-483, 1988.
6. Ohtsuki T. Minimum dissection of rectilinear regions. In Proceedings of the 1982 IEEE International Symposium on Circuits and Systems, Rome, pages 1210-1213, 1982.
7. Бутов, А.А. Анализ корректности покрытий многосвязных многоугольников / А.А. Бутов, Е.А. Шестаков / Вестник БрГТУ. – № 5: Физика, математика, информатика. – 2008. – С. 57–60.

Материал поступил в редакцию 20.09.08

SHESTAKOV E.A. Decomposition multicoherent polygon in set of rectangulars

The decomposition multicoherent polygon in set of rectangulars is considered. The purpose of work is the search for multicoherent polygon of a covering consisting of the minimal number of rectangulars. Object of research are the multicoherent final areas of a plane, by means of which the elements of topology of photo masks are described.

The heuristic method of decomposition any multicoherent polygon in set of rectangulars is developed.

УДК 004.81

Крапивин Ю.Б.

К ЗАДАЧЕ АВТОМАТИЧЕСКОГО РАСПОЗНАВАНИЯ ВОСПРОИЗВЕДЕННЫХ ФРАГМЕНТОВ ТЕКСТОВЫХ ДОКУМЕНТОВ

Введение. Чем быстрее изменяются научные представления об окружающем нас мире, внедряются в человеческую жизнь информационные технологии, тем все большую значимость приобретает возможность оперативного получения актуальной информации и использования электронной формы хранения подавляющего большинства текстовых документов практически во всех областях человеческой деятельности. Информационные системы, оперирующие большими объемами текстовых документов произвольной предметной области и успешно решающие различные прикладные задачи, становятся все более востребованными как предприятиями и организациями, так и отдельными пользователями. Постоянно увеличивающийся объем информации, доступной в полнотекстовых базах данных и в сети Интернет, кроме очевидных преимуществ, создает множество проблем. Одной из таких проблем является избыточность информации, что выражается в существовании документов, дублирующих, полностью или частично, информацию различной тематической направленности. Последнее затрудняет получение необходимых данных, создаёт предпосылки для нарушения авторских прав, влечёт за собой не только временные, но и экономические потери.

В этой связи разработка методов и алгоритмов автоматического распознавания воспроизведенных фрагментов текстового документа, т.е. тех фрагментов данного (входного) документа, которые заимствованы из других документов, представленных, в конечном счете, в некоторой заданной полнотекстовой базе данных, является актуальной задачей. В настоящее время существуют различные системы, решающие данную задачу. Наибольшее распространение получили среди них системы WCopyfind, CopyCatch, PlagiatInform, АнтиПлагиат, оперирующие алгоритмами распознавания явного, но не всегда точного заимствования фрагментов текста: их соответствие по лексическому составу и позициям лексических единиц либо только по лексическому составу, с учётом простейших морфологических преобразований и отношений синонимии. К тому же, каждая из этих систем поддерживает работу только с одним языком. Существующие системы в большинстве своем не обеспечивают приемлемых результатов работы по таким показателям, как полнота и точность

***Крапивин Юрий Борисович**, аспирант кафедры информационных интеллектуальных технологий Брестского государственного технического университета.*
Беларусь, БрГТУ, 224017, г. Брест, ул. Московская, 267.

анализа текстов, скорость их обработки, объемы используемой памяти ЭВМ. Это объясняется недостаточной эффективностью соответствующих алгоритмов, реализуемых в рамках следующих наиболее распространенных подходов: строкового соответствия, атрибутно-подсчетного и информационно-поискового, опирающегося на технику ранжирования, разработанную для информационного поиска [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 21].

Методы выявления строковых соответствий обнаруживают общие фрагменты текстовых документов, рассматривая последние как символьные строки. Такая интерпретация не учитывает их внутреннюю структуру и в большей степени применима для выявления фактов заимствований в исходных кодах программ [1, 2, 3, 4].

Суть атрибутно-подсчетного подхода сводится к численному выражению определённых признаков анализируемого текста и сравнению полученных значений с соответствующими значениями документов из базы данных. При этом документы, обладающие близкими численными характеристиками атрибутов, считаются похожими, т.е. содержащими одинаковые фрагменты. Примером может служить метод построения отпечатков документов [5, 6]. Здесь для каждого документа коллекции строится его компактное описание или отпечаток, fingerprint, представляющий собой некоторый набор идентифицирующих документ контрольных сумм или хеш-значений [7, 5]. При этом на эффективность метода в значительной степени влияют: функция, используемая для генерации хеш-значений из подстроки документа; степень детализации отпечатка, т.е. длина подстроки текста, извлекаемой из документа; разрешение отпечатка, т.е. число хеш-значений, используемых для построения отпечатка; алгоритм выбора подстрок из документа [8, 9]. Дальнейшее сравнение этих отпечатков позволяет определить вероятность подобия документов. Основной недостаток атрибутно-подсчетного подхода состоит в том, что не связанные между собой параметры плохо описывают документ в целом.

Методы информационно-поискового подхода, как правило, включают два этапа: на первом этапе коллекция документов индексируется, на втором этапе к ней отправляется запрос. При этом запрос используется при проведении вычислений оценки подобия для каждого документа коллекции в соответствии с определённой функцией - мерой подобия [10]. Документы сортируются по убыванию оценки, а документы с наивысшей оценкой возвращаются пользователю в качестве решения задачи. Эта методика не дает чёткого ответа на вопрос о том, являются ли документы релевантными требованиям пользователя, а значит, содержат ли заимствованные фрагменты, но упорядочивает документы с учётом вычисленной вероятности соответствия. Информация же, необходимая для такой оценки, зависит от используемой меры подобия, изменение которой определяет эффективность ранжирования. Меры подобия, как правило, используют статистическую информацию о частоте вхождения i-го терма в документ или коллекцию, их размере и др. Наиболее распространёнными являются методы подобия, основанные на скалярном произведении, нормализованном скалярном произведении, косинусной мере, мере идентичности [5, 10, 11, 8].

В последние годы активно стали разрабатываться методы, позволяющие учитывать лингвистическую информацию анализируемого текста, и они отличаются довольно высокой точностью. Это подтверждают результаты работ, в которых в большей или меньшей степени проводился именно такой анализ данных. Так, в работе [12] он применялся для определения ключевых слов текста, а также именных и глагольных групп. Используя далее семантические категории глаголов [13], а также правила их сочетаемости и употребления в тексте [14, 15], решалась задача обнаружения перефразированных документов. Разработчики опытной системы CHECK [16] проводили лексический анализ текста с целью приведения его слов к каноническому виду. Определяли также ключевые слова, анализировали структуру построения документа. В работах [17, 18] проводилась лемматизация слов, удаление стоп-слов, учет синонимов, а далее – лингвистический анализ типа LSA/PLSA [19] и LSA [20] соответственно.

Наше исследование базируется на анализе разнообразных методов автоматической обработки языковых данных (статистических, лингвистических, гибридных) с целью создания новых эффективных алгоритмов решения рассматриваемой задачи, включая и разработку модели соответствующей системы в целом. Как показал проведенный анализ, проектируемая система должна при этом предоставлять пользователю возможность:

- обработки текстовых документов в многоязычной информационной среде, что позволит обнаруживать в анализируемом, т.е. входном документе фрагменты из текстовых документов, представленных как на языке этого документа, так и на других языках из рассматриваемого их множества;
- анализа текстовых документов не только из локальной полнотекстовой базы данных, но и других Интернет-доступных текстовых документов, потенциально релевантных входному документу;
- получения качественного по точности и полноте, причём, с учётом явного и неявного заимствования фрагментов документов, и скорости решения задачи.

Ориентация системы на решение задачи в многоязычной информационной среде потребует, очевидно, реализации в её составе функциональности, во-первых, распознавания языка текстового документа и, во-вторых, машинного перевода (МП) текстов во множестве L заданных языков, $L = \{L_i\}$, $i = \overline{1,n}$. Причём, в последнем случае речь может идти о разработке/использовании либо множества систем МП с языка L_i на язык L_j, $i, j = \overline{1,n}$, $i \neq j$ (случай, когда все языки из их множества L являются «функционально равными»), либо множества систем МП с L_i на L_j, $1 \le j \le n$ – фиксированное, $i = \overline{1,n}$, $i \neq j$ (случай, когда один из языков из множества L, а именно L_j, является «функционально базовым»).

Наличие в системе возможности анализа/обработки текстовых документов не только из локальной полнотекстовой базы данных, но и других Интернет-доступных текстовых документов предполагает разработку инструментальных средств поиска во множестве таких документов тех, которые в какой-то мере, с точки зрения решаемой задачи, релевантны входному документу. Таким образом, речь в данном случае идёт о реализации ставшей уже классической функциональности информационного поиска.

Что касается функциональности собственно распознавания воспроизведенных фрагментов текстовых документов, то, в силу отмеченного ранее, она должна ориентироваться не только на явное, но и неявное заимствование, по крайней мере с точностью до парадигм лексических единиц и отношений синонимии для них, а также с точностью до перифраз (уровень синтаксического анализа языка). Безусловно, в идеале поставленная задача должна решаться и с точностью до основных типов знаний, а именно объектов (классов объектов), фактов и причинно-следственных отношений, отображающих закономерности внешнего мира/предметной области [22] (уровень семантического анализа языка).

Указанные три базовые функциональности системы, очевидно, требуют в совокупности разработки/использования развитого лингвистического процессора (ЛП), ориентированного на автоматический лексико-грамматический, синтаксический и семантический уровень анализа и синтеза языка.

В целом исследование поставленной задачи позволяет выделить в ней следующие основные подзадачи:

- определение языка текстового документа;
- автоматическое индексирование текстовых документов (входного, интернет-доступных и заданных в полнотекстовой БД) с целью организации информационного поиска документов, релевантных, с точки зрения решаемой задачи, входному;
- поиск релевантных, независимо от языка, документов и их ранжирование по степени релевантности;
- распознавание эквивалентности фрагментов текстовых документов, независимо от их языка и с учётом явного и неявного заимствования.

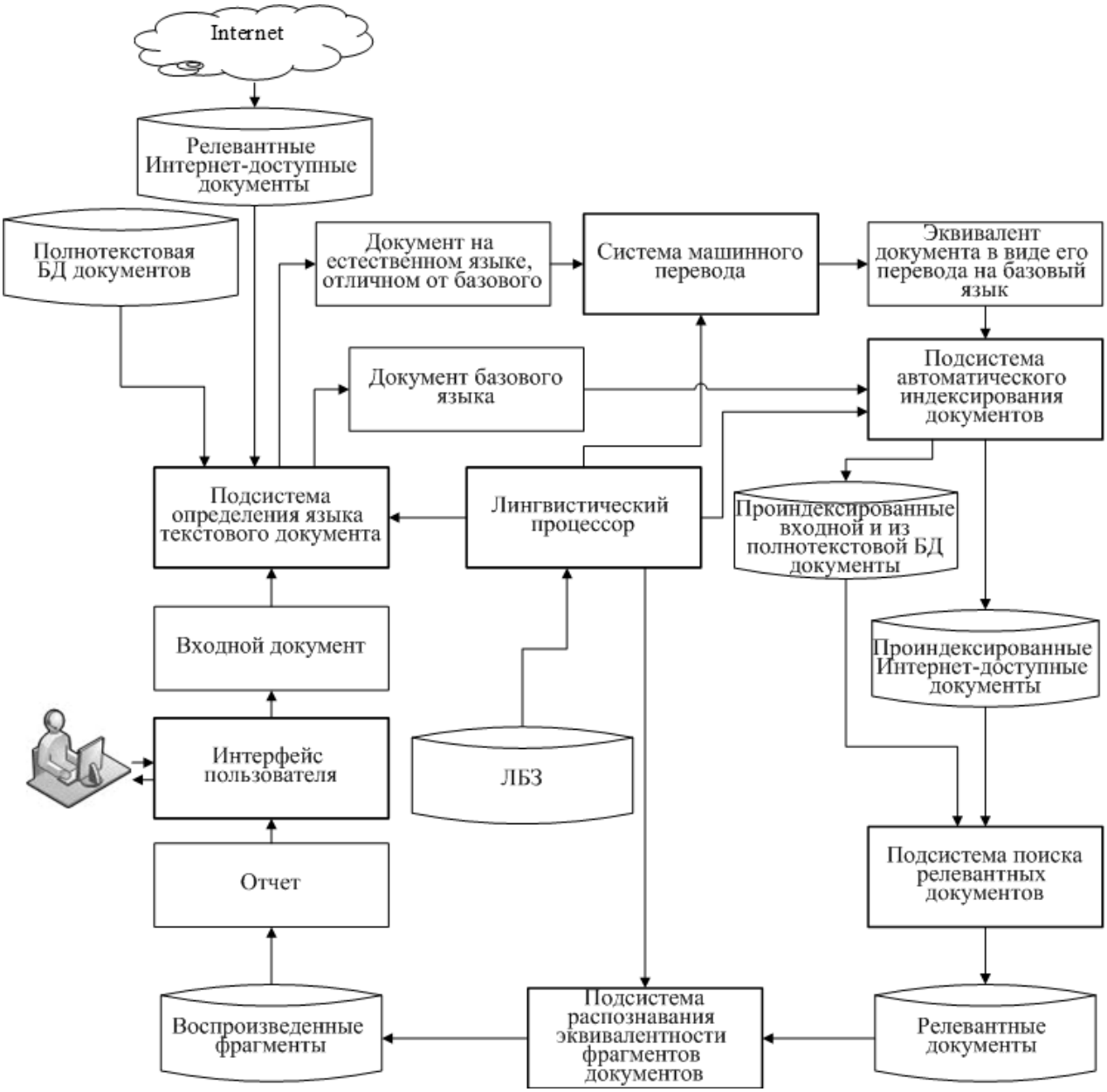

Рис. 1. Структурно-функциональная схема системы

Решение перечисленных подзадач требует разработки в общем случае многоязычного лингвистического процессора, обеспечивающего лингвистический анализ текстовых документов на лексико-грамматическом, синтаксическом и семантическом уровнях глубины языков из их множества L в той степени, в какой это необходимо для решения указанных подзадач, а также отмеченную ранее функциональность МП текстовых документов. Проектирование и реализация такого сложного базового модуля системы, очевидно, предполагает в свою очередь разработку соответствующей лингвистической базы знаний (ЛБЗ), включающей различные, в том числе и эталонные, словари языков и корпуса их текстов, грамматики языков, классификаторы их свойств на различных уровнях глубины языков, так называемые распознающие лингвистические модели анализа текстов в виде лингвистических правил (паттернов) и т.д.

Таким образом, именно лингвистический подход, как базовый, берётся в нашем случае в основу построения системы автоматического распознавания воспроизведенных фрагментов текстовых документов, структурно-функциональная схема которой, в силу вышеизложенного, может быть представлена в виде (рис. 1).

В соответствии с представленной структурно-функциональной схемой каждый документ, будь то документ из Полнотекстовой базы данных, содержащей множество эталонных документов, базы данных релевантных Интернет-доступных документов, полученных в результате Интернет-поиска, или входной документ, заданный пользователем, первоначально поступает в Подсистему определения языка текстового документа. В случае, если язык входного документа совпадает с языком, выбранным для работы системы в качестве базового, то документ сразу обрабатывается в Подсистеме автоматического индексирования документов, иначе – предварительно переводится на базовый язык Системой машинного перевода, и затем также направляется в Подсистему автоматического индексирования документов. Для каждого документа, поступающего в указанную подсистему, строится его поисковый образ (ПОД), который наряду с оригинальным документом сохраняется в поисковый индекс – Проиндексированные входной и из полнотекстовой БД документы или Проиндексированные Интернет-доступные документы, если документ был получен в результате Интернет-поиска по ключевым словам, выделенным из анализируемых документов. Далее подключается функциональность Подсистемы поиска релевантных документов, которая реализуется путём

сравнения их ПОД-ов. На следующем шаге входной документ и все полученные для него релевантные документы поступают в Подсистему распознавания эквивалентности фрагментов документов, которое осуществляется с учётом явного и указанного ранее типа неявного заимствований. Эквивалентные, с точки зрения критериев системы, фрагменты – Воспроизведенные фрагменты – с указанием их источников оформляются в виде Отчёта и поступают пользователю. Его взаимодействие с системой осуществляется посредством интерфейса, который поддерживает ввод документов и просмотр результатов поиска заимствований.

Функциональность Подсистемы определения языка текстового документа, Подсистемы автоматического индексирования документов, Подсистемы распознавания эквивалентности фрагментов документов и Системы машинного перевода обеспечивается ЛП и его ЛБЗ, причём в той мере, в какой это необходимо для качественного решения задачи, т.е., как отмечалось ранее, как минимум с учётом синтаксического уровня языков.

Важно, что представленная принципиальная схема позволяет системе, построенной в соответствии с ней, обладать преемственностью (путём наращивания мощности используемой лингвистической базы знаний) т.е., в данном случае, способностью порождения новых её версий как с точки зрения поддержки работы с другими языками, так и увеличения глубины распознавания неявного заимствования за счёт использования уровня семантического анализа языка.

Заключение. Разработка методов и алгоритмов автоматического распознавания воспроизведенных фрагментов текстовых документов и практическое их применение в прикладных системах автоматической обработки текста, в том числе в системах информационного поиска, позволит улучшить качество индекса вследствие устранения избыточности информации, исключить повторяющиеся документы из списка результатов запроса поисковой машины, обеспечить построение сюжетов, близких по содержанию новостным сообщениям. Кроме того, решение поставленной задачи предоставляет возможность создания новых или улучшения существующих инструментально-программных комплексов анализа текстовых документов на предмет выявления в них случаев заимствования без ссылок на авторов.

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. M. Wise. Running Rabin- Karp Matching and Greedy String-Tiling. Basser Department of Computer Science Technical Report, Sydney University, 1994.
2. A. Ahtiainen, S. Surakka, M. Rahikainen. Plaggie: GNU-Licensed Source Code Plagiarism Detection Engine for Java Exercises. Proc. Of the 6-th Baltic Sea Conference on Computing Education Research, 2006.
3. L. Prechelt, G. Malpohl, M. Philippsen. Finding Plagiarism among a Set of Programs with JPlag. Journal of Universal Computer Science, vol 8(11), 2002. – P. 1016–1038.
4. M. Wise. YAP: Improved Detection of Similarities in Computer Programs and Other Texts. Proc. of SIGCSE'96 Technical Symposium, 1996. – P. 130–134.
5. U. Manber. Finding similar files in a large file system. In 1994 Winter USENIX Technical Conference. – Sun Francisco, CA, January, 1994. – P. 1–10.
6. N. Shivakumar and H. Garcia-Molina. SCAM: a copy detection mechanism for digital documents. In Proc. International Conference on Theory and Practice of Digital Libraries, Austin, Texas, June 1995.
7. A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig. Syntactic clustering of the web. Sixth International World Wide Web Conference, April 1997.
8. Bernstein Y. A Scalable System for Identifying Co derivative Documents / Y. Bernstein, J. Zobel // String Processing and Information Retrieval. – 2004. – P. 55–67.
9. Henzinger M. Finding Near-Duplicate Web Pages: a Large-Scale Evaluation of Algorithms / M. Henzinger // SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval. – 2006. – P. 248–291.
10. I. H. Witten, A. Moffat, and T. C. Bell. Managing Gigabytes: Compressing and indexing documents and images. Morgan Kaufmann, second edition, 1999.
11. J. Zobel and A. Moffat. Exploring the similarity space. SIGIR Forum, 32(1):18-34, 1998.
12. O. Uzuner, B. Katz, and T. Nahnsen. Using Syntactic Information to Identify Plagiarism. CSAIL, Cambridge, 2002.
13. B. Levin. 1993. English Verb Classes and Alternations. A Preliminary Investigation. University of Chicago Press.
14. O. Uzuner, R. Davis, and B. Katz. 2004. Using empirical methods for evaluating expression and content similarity. In Proceedings of the 37th Hawaiian International Conference on System Sciences (HICSS-37). IEEE Computer Society.
15. O. Uzuner and B. Katz. Capturing expression using linguistic information. In Proceedings of the 20th National Conference on Artificial Intelligence (AAAI-05), 2005.
16. Antonio Si, Hong Va Leong, Rynson W. H. Lau. CHECK: A Document Plagiarism Detection System. In Proceedings of ACM Symposium for Applied Computing, pp. 70–77, Feb. 1997.
17. Tuomo Kakkonen, Niko Myller, Jari Timonen, Erkki Sutinen. Automatic Essay Grading with Probablistic Latent Semantic Analysis. In Proceedings of the 2nd Workshop on Building Educational Applications Using NLP, pp. 29–36, June 2005.
18. Zdenek Ceska. The Future of Copy Detection Techniques. DCSE, University of West Bohemia, 2007.
19. T. Hofmann, 2001. Unsupervised Learning by Probabilistic Latent Semantic Analysis. Machine Learning, 42:177-196.
20. T. K. Landauer, P. W. Foltz, and D. Lahm, 1998. Introduction to latent semantic analysis. Discourse Processes, 25:259-284.
21. Wang Y., Waibel A. Modelling with Structures in Statistical Machine Translation // Proceedings of COLING-ACL. – 1998. – P. 1357–1363.
22. Совпель, И.В. Система автоматического извлечения знаний из текста и её приложения // Искусственный интеллект. – 2004. – № 3. – С. 668–677.

Материал поступил в редакцию 14.12.09

KRAPIVIN Yu.B. To a task of automatic recognition of reproduced fragments of the textual documents

The analysis of the most widespread approaches to solve the problem of automatic recognition of reproduced fragments of the text document has been done. The base functionality has been defined and the schematic diagram of the system ensuring the solution of the stated problem has been presented and described.