УДК 004.853

Н. И. Гурин, Я. А. Жук
Белорусский государственный технологический университет

ГЕНЕРАТОР СЕМАНТИЧЕСКОЙ СЕТИ ИНФОРМАЦИОННОЙ СИСТЕМЫ В ТАБЛИЦУ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ

Статья посвящена описанию генератора семантической сети информационных систем для автоматического консультирования пользователей в дистанционном режиме. В работе описан способ организации базы знаний такой системы путем записи семантической сети в виде списка дуг в таблицу реляционной базы данных, ориентированной на организацию текстового или речевого диалога человека с компьютером. В базу данных кроме семантической сети входят наборы грамматических шаблонов, выражающих семантические отношения различных типов. Разработан алгоритм автоматизированного наполнения семантической сети посредством анализа текста информационных систем, основанного на членении предложений на тему, рему и элементы перехода. Это отличает предложенный алгоритм от существующих подходов обработки текста в компьютерной лингвистике. Кроме того, предлагаемый алгоритм предусматривает этап предварительной обработки текста путем его разбиения на предложения и последующего преобразования в набор семантических блоков, каждый из которых дополняется до самостоятельного простого предложения при помощи контекста. В результате формируется SQL-запрос на вставку семантических связей в таблицу реляционной базы данных. На основе предложенного алгоритма на языке Python разработан генератор семантической сети базы знаний информационной системы, размещенной в компьютерной сети.

Ключевые слова: семантическая сеть, реляционная база данных, диалог человека с компьютером, автоматическая обработка текста.

N. I. Gurin, Ya. A. Zhuk
Belarusian State Technological University

THE INFORMATION SYSTEM SEMANTIC NETWORK TO A RELATIONAL DATABASE TABLE GENERATOR

This article describes the information system semantic network generator for automatic remote user consultation. The article describes the way of organizing the knowledge base of this system by writing the semantic network if the form of arcs list to a relational database table, organized for text or speech dialogue between a human and a computer. This database in addition to the semantic network contains the sets of grammatical patterns, expressing semantic relations of various types. The algorithm of automated filling of the semantic network is developed. It is based on the information system text analyzing by division of the sentences into subject, object and transition elements. This method is different from existing approaches to text processing in computational linguistics. The proposed algorithm includes the pre-processing of the text by splitting it into the sentences and subsequent transformation into a set of semantic units. Each of these units is transformed into an independent simple sentence using context. The result is a SQL query for semantic relations insertion into a relational database table. Based on the proposed algorithm the information system semantic network generator was designed in Python language and placed in a computer network.

Key words: semantic network, relational database, human-computer dialogue, natural language processing.

Введение. В настоящее время широко распространено дистанционное обучение на основе справочных, информационных и обучающих систем, размещаемых в компьютерной сети. Самостоятельное изучение содержания таких систем поддерживается текстовым или речевым диалогом обучаемого с преподавателем или консультантом, что во многих случаях является наиболее оптимальным способом получения справочной или учебной информации. Как правило, обращение за консультацией относится к той информации, которую можно найти в информационной системе, поэтому требуется реализация специализированного модуля автоматического консультирования, позволяющего пользователю получить ответ на возникший вопрос при помощи самой системы. Одним из ключевых элементов разработки системы автоматического консультирования является генерация базы знаний для обеспечения диалога с пользователем по вопросам соответствующей предметной области.

Основная часть. Для создания базы знаний информационной системы используется наиболее общая модель представления знаний – семантическая сеть (СС). Под СС понимают орграф, в вершинах которого находятся информационные единицы, а дуги характеризуют отношения и связи между ними [1]. Опыт организации порталов научных знаний, построенных в виде СС, показывает, что СС по одной предметной области состоит из порядка 1000 вершин и 2000 дуг между ними [2], поэтому наиболее компактной формой хранения СС является список дуг. Ключевым отличием списка дуг СС от списка дуг обычного графа является наличие типов отношений, обозначаемых дугами. Наполнение СС является трудоемким процессом, требующим чтения и анализа содержимого информационной системы, по которой строится СС, поэтому разработка генератора СС (ГСС) является актуальной задачей.

Реализация СС для диалоговой системы и анализатора текста на естественном языке имеет свою специфику. Во-первых, в качестве идентификаторов вершин используются текстовые названия объектов. Во-вторых, каждый тип отношений сопоставлен набору грамматических шаблонов, выражающих данный тип отношений. Поскольку тип отношения между информационными единицами выражается при помощи глагола, в качестве грамматических шаблонов используются глагольные сказуемые со вспомогательными предлогами и специальными тегами: для глагольного окончания первого спряжения – тег [Г1], для второго – [Г2], для информационной единицы, выступающей в роли подлежащего при прямом прочтении семантической связи, – [А], для выступающей в роли вспомогательного члена предложения – [Б]. В отличие от художественной литературы, в научном стиле порядок расположения членов предложения всегда одинаков: подлежащее, сказуемое, вспомогательные члены предложения. Поскольку семантические связи носят направленный характер и существует необходимость в распознавании вопросов пользователей, грамматические шаблоны необходимо хранить четверками: двух вопросительных и двух утвердительных при прямом и обратном прочтениях семантического отношения (например, четверка шаблонов «из чего состоит [А]?», «в состав чего входит [Б]?», «[А] состоит из [Б]» и «[Б] является частью [А]»).

Рассмотрим запись списка дуг для фрагмента СС, представленного на рис. 1.

На основании изложенных принципов две связи данного фрагмента могут быть записаны в виде следующего списка дуг:

$$e_1 = \begin{pmatrix} \text{«электрод»} \\ \text{«металл»} \\ \text{«[А] состо[Г 2] из [Б]»} \end{pmatrix}$$

$$e_2 = \begin{pmatrix} \text{«электрод»} \\ \text{«ионы в растворе»} \\ \text{«[А] состо[Г 2] из [Б]»} \end{pmatrix}$$

Для хранения СС и грамматических шаблонов, используемых для выражения ее связей, была составлена схема реляционной базы данных (БД) из двух таблиц. Первая таблица предназначена для хранения списка дуг СС в виде трех полей: двух информационных единиц и типа связи. Вторая таблица необходима для хранения наборов грамматических шаблонов предложений, соответствующих типам отношений. Применение реляционной БД является эффективным приемом программирования, так как предоставляет ряд возможностей для развития быстродействия и функционала [3]. Это позволяет избежать применения специализированных средств разработки и значительных затрат времени на разработку собственного механизма хранения знаний в памяти как в проектах [4, 5].

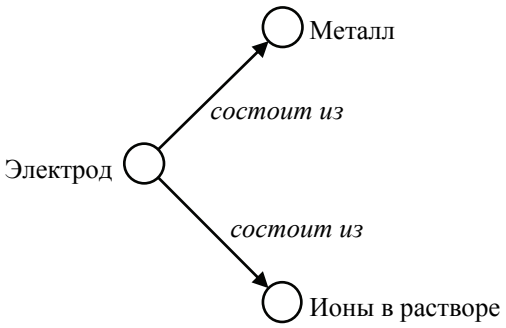

Рис. 1. Графическое представление фрагмента графа

Для автоматизации наполнения СС используется разработанный в лингвистике принцип актуального членения предложений на исходную часть сообщения (тему), новую часть сообщения (рему) и связующий член, выражаемый глагольным сказуемым [6]. Такое членение предложения идеально подходит для формирования СС, поскольку разбивает предложение на две информационные единицы (тему и рему) и выражаемый глагольным сказуемым тип отношения между ними. Однако данный подход применим только к простым предложениям и не может применяться к текстам информационных систем, в которых широко употребляются обороты в скобках и причастные, деепричастные, анафорические обороты, без предварительного анализа текста.

На основании изложенных положений разработан следующий алгоритм генерации СС. После ввода и отправки пользователем фрагмента текста ГСС выполняет замену сокращений, содержащих точки, на их полные аналоги для корректного определения границ предложений. Кроме того, на данном этапе осуществляется удаление из текста оборотов, не несущих смысловой нагрузки. После выполнения данных операций осуществляется разбиение текста на предложения по точкам. Следующим этапом работы ГСС является разбиение каждого предложения на отдельные семантические блоки, несущие самостоятельную смысловую нагрузку, по скобкам и запятым. Блокам, находящимся в скобках, назначается более высокий уровень скобочной вложенности. Следующим этапом работы ГСС является дополнение неполных блоков до самостоятельных простых предложений, при котором блоки обходятся в порядке убывания уровня вложенности. При необходимости в качестве сказуемого может использоваться глагол из предыдущего семантического блока, глагол, зависящий от контекста, и преобразованное в глагол причастие или деепричастие. В качестве подлежащего применяется последнее существительное предыдущего блока или подлежащее предыдущего блока в случае анафорического оборота. В результате дополнения каждый семантический блок будет представлять собой самостоятельное простое предложение. Затем выполняется актуальное членение каждого предложения путем поиска подходящего шаблона предложения в БД при помощи регулярных выражений. Для этого в шаблонах предложений проводится замена тегов на обозначения произвольных строк и допустимых наборов окончаний. В результате тема, рема и глагольное сказуемое записываются в соответствующие поля запроса на вставку в БД, который является результатом работы ГСС. Блок-схема рассмотренного алгоритма работы ГСС изображена на рис. 2.

В качестве демонстрации работы алгоритма рассмотрим обработку предложения *«Электрод состоит из металла M z^- (восстановленная форма системы) и ионов M z^+ в растворе (окисленная форма системы)»*. В данном предложении нет сокращений и оборотов, не несущих смысловой нагрузки, поэтому первым действием генератора будет разбиение исходного предложения на семантические блоки.

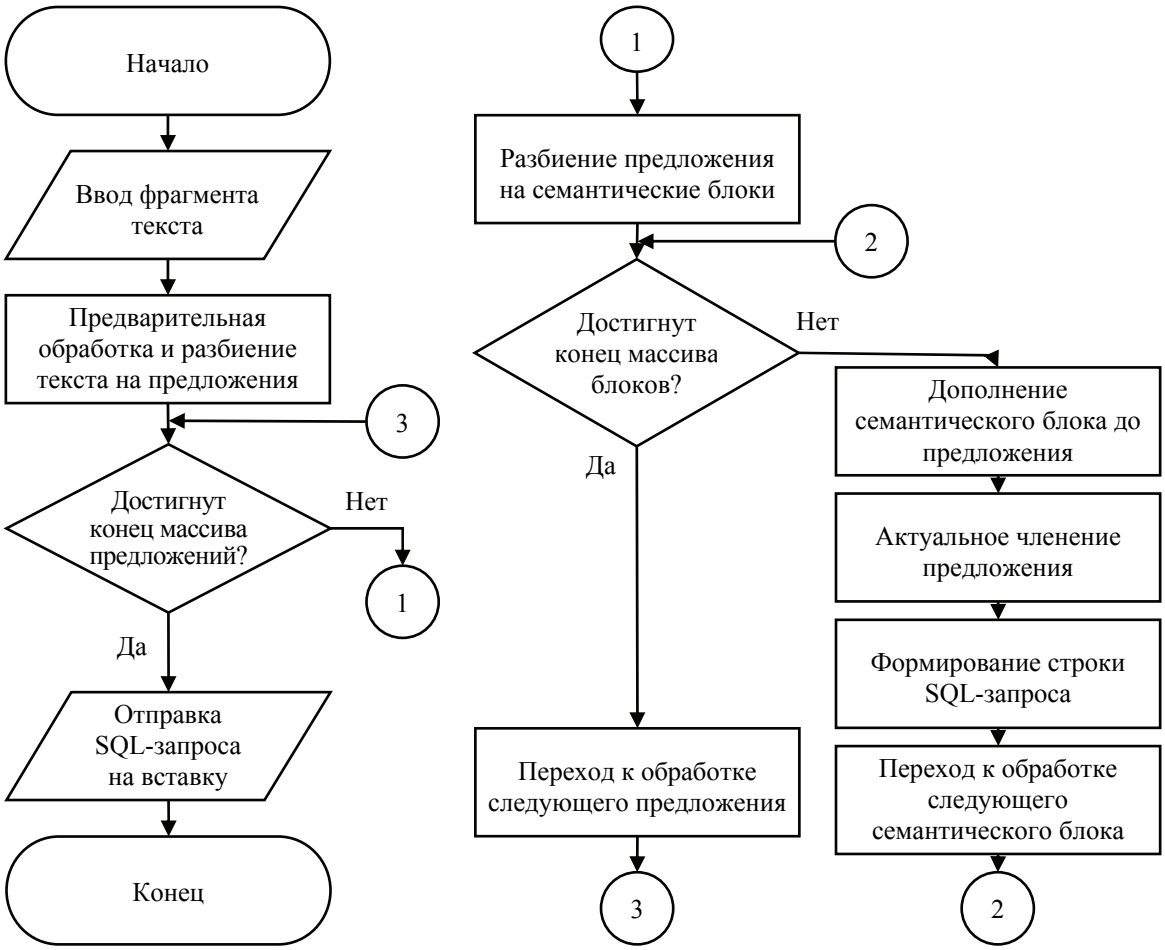

Рис. 2. Блок-схема алгоритма генерации СС

В результате разбиения по скобкам будет получено четыре семантических блока: *«электрод состоит из металла M z^-»*, *«восстановленная форма системы»*, *«и ионов M z^+ в растворе»*, *«окисленная форма системы»*. Следует отметить, что второму и четвертому семантическим блокам назначен первый уровень вложенности из-за того, что они расположены в скобках, а первому и третьему – нулевой. Поэтому первыми анализируются второй и четвертый блоки.

Данные семантические блоки не имеют собственных подлежащих и сказуемых, поэтому ГСС дополняет их при помощи последних существительных предыдущих блоков и сказуемого *«еще называется»*. В результате блоки трансформируются в самостоятельные простые предложения *«металла еще называется восстановленная форма системы»* и *«растворе еще называется окисленная форма системы»*. Следует отметить отсутствие согласованности падежей в полученных предложениях, однако в ней нет необходимости, так как поиск подходящего шаблона выполняется по глаголу, а после распознавания информационных единиц в соответствии с шаблоном они приводятся к именительному падежу.

Третий семантический блок также не имеет собственного подлежащего и сказуемого, однако первым словом данного блока является союз *«и»*. ГСС воспринимает данную ситуацию как перечисление и дополняет текущий блок подлежащим и сказуемым из предыдущего блока того же уровня вложенности. В результате будет получено простое предложение *«Электрод состоит из ионов M z^+ в растворе»*. Первый семантический блок содержит подлежащее и сказуемое, что говорит об отсутствии необходимости его дополнения.

Следующим этапом работы ГСС является поиск в БД для каждого из полученных простых предложений подходящего грамматического шаблона. Для этого составляется SQL-запрос на выборку вида *«SELECT Ответ, Вопрос FROM шаблоны WHERE '<предложение>' REGEXP " + replaceCodesInSQL ("Ответ", "(.*)")»*. В конце данного запроса вызывается функция *replaceCodesInSQL*, тело которой приведено в листинге.

Полученные при помощи данной функции SQL-запросы выполняют поиск в БД грамматических шаблонов посредством регулярных выражений. Для этого в шаблонах места подстановки информационных единиц заменяются на обозначение произвольной последовательности символов *(*)*, а глагольные окончания – на наборы допустимых окончаний.

```
   def replaceCodesInSQL( fieldName,
strToInsert):
      return "replace( replace(
replace( replace( replace( replace(
replace( replace( replace( replace("
+ fieldName + ", '[А]', '" +
strToInsert + "'), '[Б]', '" +
strToInsert + "'), '[Г1]',
'(ет|ут|ют|я)'), '[Г2]',
'(ит|ат|ят|я)'), '[Г3]',
'(жет|гут|жа)'), '[О]','" +
strToInsert + "'), '[Т]', '" +
strToInsert + "'), '[М]', '" +
strToInsert + "'), '[Ч]', '" +
strToInsert + "'), '[К]',
'(классу|категории|группе)')"
```

Листинг. Функция подготовки SQL-запроса на выборку

В результате работы разработанного ГСС исходное предложение было преобразовано в следующий SQL-запрос: *«INSERT INTO `связи` (`А`, `Б`, `Вопрос`, `ID`) VALUES ('электрод', 'металл M z^-', 'из каких [К] состо[Г2] [А]', NULL), ('электрод', 'ионы M z^+ в растворе', 'из каких [К] состо[Г2] [А]', NULL), ('металл', 'восстановленная форма системы', 'как еще называ[Г1]ся [А]', NULL), ('раствор', 'окисленная форма системы', 'как еще называ[Г1]ся [А]', NULL)»*. Это означает, что генератор успешно обработал перечисление с союзом «и», а также два оборота в скобках, использующихся для уточнения.

Заключение. Апробация реализованного на языке Python ГСС на фрагментах текста компьютерной обучающей системы по электрохимии показала, что данное веб-приложение успешно справляется с разбором различных сложных предложений. В результате работы генератора были получены SQL-запросы на вставку семантических связей по предметной области в таблицу реляционной БД. После выполнения полученных SQL-запросов модуль диалога с пользователем сформировал удовлетворительные ответы на заданные по выбранному тексту вопросы.

Литература

1. Аверкин А. Н., Гаазе-Рапопорт М. Г., Поспелов Д. А. Толковый словарь по искусственному интеллекту. М.: Радио и связь, 1992. 256 с.

2. Разработка портала знаний по компьютерной лингвистике / Ю. А. Загорулько [и др.] // Материалы XI национальной конференции по искусственному интеллекту с международным участием (КИИ-08). Дубна, 2008. С. 352–360.

3. Гурин Н. И., Жук Я. А. Разработка семантических сетей и анализаторов для компьютерных обучающих систем // Современные информационные компьютерные технологии mcIT-2013: материалы III Междунар. науч.-практ. конференции [Электронный ресурс]. Гродно, 2013. 1 электр. компакт диск (CD-R). 792 с. Рус. Деп. в ГУ «БелИСА» 19.09.13, № Д201315.

4. Гурин Н. И., Герман О. В. Интеллектуальный анализатор запросов к базе знаний мультимедийного электронного учебника // Труды БГТУ. 2010. № 6: Физ.-мат. науки и информатика. С. 167–170.

5. Голенков В. В., Гулякина Н. А. Семантическая технология компонентного проектирования систем, управляемых знаниями // Открытые семантические технологии проектирования интеллектуальных систем: материалы V Междунар. науч.-техн. конф. / редкол.: В. В. Голенков (отв. ред.) [и др.]. Минск, 2015. С. 57–78.

6. Лингвистический энциклопедический словарь / гл. ред. В. Н. Ярцева. М.: Сов. энциклопедия, 1990. 685 с.

References

1. Averkin A. N., Gaaze-Rapoport M. G., Pospelov D. A. *Tolkovyy slovar' po iskusstvennomu intellektu* [Dictionary of artificial intelligence]. Moscow, Radio i svyaz' Publ., 1992. 256 p.

2. Zagorulko Yu. A., Borovikova O. I., Zagorulko G. B., Kononenko I. S., Sokolova E. G. [The development of the knowledge portal for computational linguistics]. *Materialy XI natsional'noy konferentsii po iskusstvennomu intellektu* [Proceedings of the XI national conference on artificial intelligence with international participation]. Dubna, 2008, pp. 352–360 (In Russian).

3. Gurin N. I., Zhuk Ya. A. [The development of semantic networks and systems for computer-based training systems]. *Sovremennyye informatsionnyye komp'yuternyye tekhnologii mcIT-2013: materialy III mezhdunarodnoy nauchno-prakticheskoy konferentsii* [Modern computer information technologies mcIT-2013: proceedings of the III Inter. sci.-pract. conf.]. Grodno, 2013, 792 p. (In Russian).

4. Gurin N. I., German O. V. Intellectual knowledge base query analyzer of the multimedia electronic textbook. *Trudy BGTU* [Proceedings of BSTU], 2010, no. 6: Physical-mathematical sciences and informatics, pp. 167–170 (in Russian).

5. Golenkov V. V., Gulyakina N. A. [Semantic technology of component design of manipulated by knowledge systems]. *Otkrytyye semanticheskiye tekhnologii proyektirovaniya intellektual'nych system: materialy V mezhdunarodnoy nauchno-tekhnicheskoy konferentsii* [Open semantic technologies for designing intelligent systems: proceedings of the V Inter. sci.-tech. conf.]. Minsk, 2015, pp. 67–78 (In Russian).

6. *Lingvisticheskiy entsiklopedicheskiy slovar'* [Linguistic encyclopedic dictionary]. Moscow, Sovetskaya entsiklopedyia Publ., 1990. 685 p.

Информация об авторах

Гурин Николай Иванович − кандидат физико-математических наук, доцент кафедры информационных систем и технологий. Белорусский государственный технологический университет (220006, г. Минск, ул. Свердлова, 13а, Республика Беларусь). E-mail: ngourine@mail.ru

Жук Ярослав Александрович − магистрант. Белорусский государственный технологический университет (220006, г. Минск, ул. Свердлова, 13а, Республика Беларусь). E-mail: zhuk@belstu.by

Information about the authors

Gurin Nikolay Ivanovich − Ph. D. (Physics and Mathematics), Assistant Professor, the Department of Information Systems and Technologies. Belarusian State Technological University (13a, Sverdlova str., 220006, Minsk, Republic of Belarus). E-mail: ngourine@mail.ru

Zhuk Yaroslav Aleksandrovich − undergraduate. Belarusian State Technological University (13a, Sverdlova str., 220006, Minsk, Republic of Belarus). E-mail: zhuk@belstu.by

Поступила 12.03.2015