УДК 004.853

СЕМАНТИЧЕСКАЯ СЕТЬ ЭЛЕКТРОННОГО УЧЕБНИКА ДЛЯ ДИАЛОГА С ВИРТУАЛЬНЫМ ПРЕПОДАВАТЕЛЕМ

Гурин Н. И., Жук Я. А.

Белорусский государственный технологический университет, Минск, Республика Беларусь, root@belstu.by

Статья посвящена описанию генератора базы знаний электронного учебника в виде семантической сети для диалога студентов в дистанционном режиме. В работе описывается метод организации базы знаний и алгоритм автоматизированного наполнения семантической сети путем анализа текста произвольной информационной системы.

The article describes the generator of the electronic textbook knowledge in the form of a semantic network for dialogue students remotely. The paper describes a method for organization of knowledge base and algorithm for automated filling of semantic network by analyzing the text of any information system.

В настоящее время широко распространено дистанционное обучение при помощи справочных, информационных и обучающих систем, размещаемых в компьютерной сети. Содержание данных систем составляется в научном стиле. Для повышения эффективности самостоятельного изучения содержания таких систем предлагается реализовать текстовый или речевой диалог обучаемого с виртуальным преподавателем или консультантом. Анализ обращений за консультацией в call-центры показывает, что в большинстве случаев запрашиваемую информацию можно было найти в информационной системе. Поэтому работа системы автоматического консультирования состоит в предоставлении пользователям кратких ответов на возникающие вопросы на основании сведений, помещенных в базу знаний. Ключевым моментом разработки системы автоматического консультирования является генерация базы знаний конкретной информационной системы для обеспечения диалога с пользователем по вопросам соответствующей предметной области.

Для хранения знаний, извлекаемых из содержания информационной системы произвольной предметной области предлагается использовать наиболее общую модель представления знаний – семантическую сеть. Под семантической сетью понимают ориентированный граф, в вершинах которого находятся информационные единицы, а дуги характеризуют отношения и связи между ними [1]. Опыт организации порталов научных знаний, построенных в виде семантической сети, показывает, что семантическая сеть по одной предметной области состоит из порядка 1000 вершин и 2000 дуг между ними [2]. Это говорит о том, что наиболее компактной формой хранения семантической сети является список дуг.

Ключевым отличием списка дуг семантической сети от списка дуг обычного графа является наличие типов отношений, обозначаемых дугами, т.е. каждая дуга кроме пары идентификаторов связываемых вершин характеризуется типом отношения между этими вершинами. Поскольку наполнение семантической сети является трудоемким процессом, требующим чтения и анализа содержимого информационной системы, по которой строится семантическая сеть. Актуальной задачей является разработка генератора семантической сети, который будет анализировать содержание информационной системы и выполнять наполнение семантической сети автоматически. Таким образом, архитектура системы автоматического консультирования включает в себя базу знаний, модуль ее автоматического наполнения и модуль диалога с пользователем.

Реализация семантической сети для диалоговой системы и анализатора текста на естественном языке имеет свою специфику. Во-первых, появляется возможность в качестве идентификаторов вершин использовать не числовые обозначения, а текстовые названия объектов предметной области, которые могут состоять из нескольких слов (например, «ионы» и «ионы в растворе» – два разных объекта). Во-вторых, каждый тип отношения может быть

сопоставлен набору грамматических шаблонов, используемых в языке для выражения данного типа отношения. Учитывая направленный характер семантических связей грамматические шаблоны необходимо хранить парами для выражения как прямого, так и обратного отношения (например, пара шаблонов «А состоит из Б и В» и «Б и В являются частями А»). Также для создания системы, способной распознавать вопросы пользователей, требуется дополнить шаблоны утвердительных предложений соответствующими шаблонами вопросительных предложений (например, «из чего состоит А?» и «в состав чего входит Б?»). Таким образом, каждый семантический тип отношения между информационными единицами может быть сопоставлен набору из четырех грамматических шаблонов:

– грамматический шаблон вопросительного предложения при прямом прочтении семантического отношения;

– грамматический шаблон утвердительного предложения при прямом прочтении семантического отношения;

– грамматический шаблон вопросительного предложения при обратном прочтении семантического отношения;

– грамматический шаблон утвердительного предложения при обратном прочтении семантического отношения.

Следует отметить, что в отличие от художественной литературы, в научном стиле порядок расположения членов предложения всегда одинаков: подлежащее, сказуемое, вспомогательные члены предложения.

Поскольку тип отношения между информационными единицами выражается при помощи глагола, в качестве основной составляющей грамматических шаблонов были использованы глагольные сказуемые. Кроме них в состав шаблонов входят вспомогательные предлоги и специальные теги, выделяемые квадратными скобками. В ходе рассмотрения содержания электронных учебно-методических комплексов по электрохимии и микробиологии был выработан следующий набор тегов:

– [А] – информационная единица, выступающая в роли подлежащего при прямом прочтении семантической связи и в роли вспомогательного члена – при обратном;

– [Б] – информационная единица, выступающая в роли вспомогательного члена при прямом прочтении семантической связи и в роли подлежащего – при обратном;

– [Г1] – глагольное окончание первого спряжения;

– [Г2] – глагольное окончание второго спряжения.

Рассмотрим запись списка дуг для фрагмента семантической сети, представленного на рисунке 1.

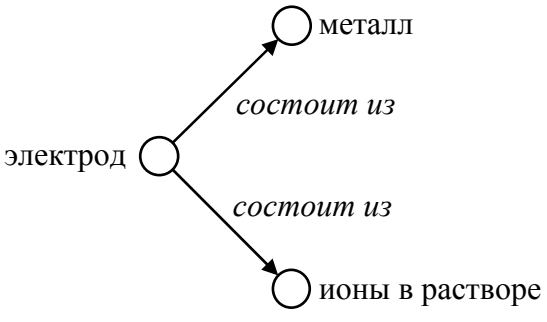

Рисунок 1 – Графическое представление графа

На основании изложенных принципов две связи данного фрагмента могут быть записаны в виде следующего списка дуг (1):

$$e_1 = \begin{pmatrix} \text{"электрод"} \\ \text{"металл"} \\ \text{"[А] состо[Г 2] из [Б]"} \end{pmatrix}$$
$$e_2 = \begin{pmatrix} \text{"электрод"} \\ \text{"ионы в растворе"} \\ \text{"[А] состо[Г 2] из [Б]"} \end{pmatrix} \quad (1)$$

Для хранения семантической сети 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 семантической сети в виде трех полей: двух информационных единиц и типа связи. В силу возможности применения как прямых, так и обратных грамматических шаблонов, информационные единицы нельзя однозначно назвать подлежащим и вспомогательным членом предложения, т.к. в зависимости от грамматического шаблона информационные единицы могут играть в предложении различные роли. Поэтому поля информационных единиц были названы «А» и «Б». Первичный ключ данной таблицы включает все 3 поля, что обеспечивает уникальность связей в семантической сети не исключая возможности связи каждого объекта с несколькими другими объектами различными типами отношений. Вторая таблица предназначена для хранения наборов грамматических шаблонов предложений, соответствующих типам отношений. Тип связи в первой таблице является внешним ключом, ссылающимся на один из шаблонов второй таблицы. Поскольку модуль ответа на вопросы пользователя разрабатывался первым, в качестве ключа используется грамматический шаблон вопросительного предложения при прямом прочтении семантического отношения. Для синтеза предложений модулем диалога с пользователем во второй таблице хранятся падежи для прямого и обратного прочтений семантического отношения, применяемые к информационным единицам в зависимости от грамматического шаблона. Применение реляционной БД является эффективным приемом программирования т.к. предоставляет возможности администрирования семантической сети, индексирования информационных единиц и типов связей для быстрого поиска в больших объемах данных, применения при поиске регулярных выражений для распознавания информационных единиц по названиям и приведения их к нужному падежу [3]. Это позволяет избежать применения специализированных средств разработки и значительных затрат времени на разработку собственного механизма хранения знаний в оперативной памяти как в проектах [4, 5].

Для автоматизации наполнения семантической сети используется разработанный в лингвистике принцип актуального членения предложений на исходную часть сообщения (тему), новую часть сообщения (рему) и связующий член, выражаемый глагольным сказуемым [6]. Следует отметить, что такое членение предложения наилучшим образом подходит для формирования семантической сети в предлагаемой форме списка дуг т.к. разбивает предложение на две информационные единицы (тему и рему) и выражаемый глагольным сказуемым тип отношения между ними. Однако следует отметить, что такой подход применим только к простым предложениям. В то же время в академической прозе широко употребляются различные обороты, некоторые из которых не несут смысла и используются для связи предложений, а другие – для внесения в предложение дополнительных знаний, часто используя для выражения типа связи знаки препинания (например, скобки). Еще одной сложностью в применении актуального членения предложения является необходимость определения глагольного сказуемого в предложении, что в свою очередь требует составления базы глаголов или правил их распознавания. Таким образом, для выделения семантических связей при помощи принципа актуального членения предложения требуется проведение дополнительного анализа текста. Он заключается в ряде операций: удалении из текста слов и оборотов, не несущих смысловой нагрузки; расшифровке содержащих точки сокращений и разбиении текста на предложения по оставшимся точкам; преобразовании сложных предложений в простые, при необходимости дополняя их подлежащим и сказуемым. Следует отме-

тить, что перечень слов и оборотов, не несущих смысловой нагрузки, может быть взят из существующих поисковых систем, а перечень сокращений содержащих точки сравнительно невелик. Наиболее сложным является преобразование сложных предложений в простые. Данная процедура должна учитывать и распознавать различные способы представления дополнительных семантических связей в предложении, такие как причастные, деепричастные, анафорические обороты, перечисления различных членов предложения и обороты в скобках.

На основании выявленных требований к предварительной обработке предложений разработан алгоритм генерации семантической сети. После ввода и отправки пользователем фрагмента текста генератор семантической сети выполняет замену сокращений, содержащих точки, на их полные аналоги для корректного определения границ предложений. Кроме того, на данном этапе выполняется удаление из текста оборотов, не несущих смысловой нагрузки. После выполнения данных операций выполняется разбиение текста на предложения по точкам. Следующим этапом работы генератора семантической сети является разбиение каждого предложения на отдельные семантические блоки, границами которых будут выступать скобки и запятые. Каждый такой блок несет самостоятельную смысловую нагрузку. Блокам, находящимся в скобках, назначается более высокий уровень вложенности. Следующим этапом работы генератора семантической сети является анализ наличия в блоках подлежащего и сказуемого. Для этого 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 используется последнее существительное предыдущего блока или подлежащее предыдущего блока в случае с анафорическим оборотом. В результате дополнения каждый семантический блок будет представлять собой самостоятельное простое предложение. Затем выполняется актуальное членение каждого предложения путем выполнения поиска подходящего шаблона предложения в БД при помощи регулярных выражений. Для этого в шаблонах предложений выполняется 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 генератора семантической сети. Блок-схема рассмотренного алгоритма работы генератора семантической сети изображена на рисунке 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. В результате разбиения по скобкам будет получено 4 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 генератор 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 т.к. поиск подходящего шаблона выполняется по глаголу, а после распознавания информационных единиц в соответствии с шаблоном они для хранения приводятся к именительному падежу.

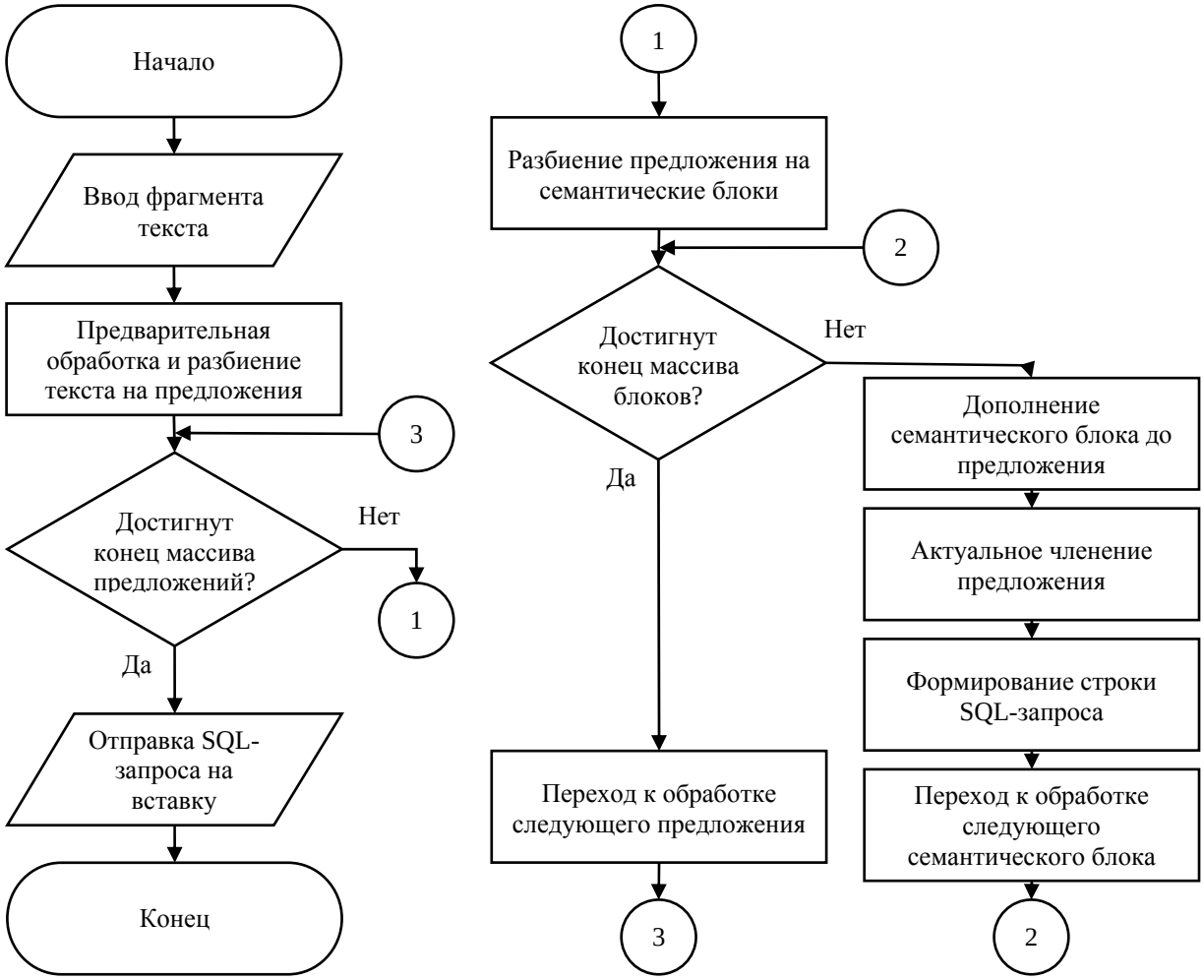

Рисунок 2 – Блок-схема алгоритма генерации семантической сети

Третий семантический блок также не имеет собственного подлежащего и сказуемого, однако первым словом данного блока является союз «и». Генератор семантической сети воспринимает данную ситуацию как перечисление и дополняет текущий блок подлежащем 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 генератора семантической сети 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", "(.*)")*». Следует отметить в конце данного запроса вызов функции *replaceCodesInSQL*, тело которой приведено в листинге на рисунке 3.

```
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)')"
```

Рисунок 3 – Функция подготовки SQL-запроса на выборку

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

В результате работы разработанного генератора семантической сети 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.

После выполнения полученного запроса на вставку можно получить соответствующие вопросы от модуля диалога, который может быть выполнен как на отдельном информационном ресурсе, так и в виде блока, расположенного на полях обучающей системы. Диалоговый модуль выполнен в клиент-серверной архитектуре с применением библиотеки JQuery для клиентской части и языка программирования PHP – для серверной. Вопрос может быть введен голосом при помощи Google Voice Recognition или при помощи клавиатуры. Клиентская часть отправляет текст введенного вопроса на сервер для поиска ответа при помощи механизма AJAX. Серверная часть в качестве входного параметра принимает текст вопроса. Анализ вопроса выполняется путем его сравнения с шаблонами, хранящимися в БД, для определения типа искомых отношений и предмета вопроса. Для поиска подсети ответа предмет вопроса приводится в начальную форму по таблице окончаний, а также рекурсивно составляются списки синонимов типа связи и предмета вопроса. При повторе вопроса или запросе более подробного ответа строгость критерия синонимичности уменьшается. В качестве подсети ответа выбираются дуги отобранных типов, инцидентные объектам из списка синонимов предмета вопроса. Для дуг подсети ответа выполняется выбор соответствующих шаблонов ответа. Затем в шаблоны ответов подставляются подписи вершин, инцидентных дугам, поставленные в соответствующие падежи. При указании в клиентской части требования озвучить ответ выполняется отправка текста на синтезатор речи Google и последующее его воспроизведение в клиентской части при помощи HTML5-тега <audio>. Также в полученном тексте выполняется замена специальных обозначений на основе отдельной таблицы БД для вывода изображений, озвученных flash-анимаций и видео. Ключевые слова и выражения в тексте ответа выделяются гиперссылками, при нажатии на которые формулируется новый вопрос серверу. Журнал вопросов и ответов сохраняется до обновления страницы, что позволяет возвращаться к упомянутым ранее понятиям.

Апробация разработанного генератора семантической сети на фрагментах текста компьютерной обучающей системы 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 заданной предметной области в таблицу реляционной БД. После выполнения полученных SQL-запросов модуль диалога с пользователем сформировал удовлетворительные ответы на заданные по рассматриваемому тексту вопросы.

Список литературы

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

2. Загорулько, Ю. А. 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. Разработка семантических сетей и анализаторов для компьютерных обучающих систем / Н. И. Гурин, Я. А. Жук // Современные информационные компьютер-

ные технологии mcIT-2013: материалы III Международной научно-практической конференции [Электронный ресурс] / УО «Гр. ун-т им. Я. Купалы». – Гродно, 2013. – 1 электр. компакт диск (CD-R). – 792 с. – Рус. – Деп. в ГУ «БелИСА» 19.09.13, № Д201315.

4. Гурин Н. И. Интеллектуальный анализатор запросов к базе знаний мультимедийного электронного учебника / Н. И. Гурин, О. В. Герман // Труды БГТУ: Физико-математические науки и информатика. – БГТУ: 2010. – С. 167-170.

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

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