УДК 004.512:004.89

И. В. Захлебин, студент магистратуры, **В. А. Фомичев,** д-р техн. наук, проф., e-mail: vfomichov@hse.ru,
Национальный исследовательский университет "Высшая школа экономики", Москва

Разработка метода семантического поиска специалистов в корпоративной базе данных по естественно-языковым запросам

Обосновывается актуальность проблемы создания компьютерных систем семантического поиска специалистов по естественно-языковым описаниям их компетенций. Описывается разработанный и программно реализованный на языке Python метод проектирования прикладных компьютерных систем этого класса. Важной отличительной чертой метода является использование класса СК-языков (стандартных концептуальных языков), предлагаемого теорией К-представлений, для формализации семантики естественно-языковых запросов и представления семантики слов и коротких словосочетаний в лингвистической базе данных.

Ключевые слова: *семантический поиск специалистов, естественно-языковой запрос, семантическое представление, теория К-представлений, СК-язык, концептуальный базис, лингвистическая база данных*

Введение

Сегодня компании вынуждены осуществлять непрерывный поиск специалистов, например, для набора персонала на открывшиеся вакансии, образования команд, устранения нештатных ситуаций и организации внутренней мобильности [1]. Ускорение поиска компетентных сотрудников дает им конкурентное преимущество: компания IBM, внедрив в начале 2000-х годов собственную систему поиска специалистов, смогла к 2006 г. сэкономить более $500 млн [2].

Как правило, компании имеют базы данных (БД), содержащие профили тысяч соискателей и собственных сотрудников. Эффективная обработка такого объема информации выходит за пределы человеческих возможностей. В связи с этим возникает потребность в разработке интеллектуальных систем, позволяющих автоматизировать часть задач по отбору кандидатов.

Заметим, что определение компетенций специалистов по неструктурированной текстовой информации (например, по резюме или рабочим документам) является отдельной сложной задачей. Для ее решения предложен ряд эффективных подходов, например, в работах [3—5]. Поэтому мы будем рассматривать проблему поиска специалистов в реалиях больших компаний, используя структурированную информацию о них из имеющейся корпоративной БД и предполагая, что она могла быть получена не только путем целенаправленного сбора, но и автоматически.

Отечественные системы поиска специалистов прежде всего представлены Интернет-ресурсами, агрегирующими резюме и предоставляющими поиск по ним. Пятью наиболее популярными в России ресурсами этой категории являются "SuperJob", "HeadHunter", "JOB.RU", "Работа.RU" и "НГС.Работа"[1]. Данные системы способны эффективно работать только с критериями отбора, заданными в структурированном виде, как правило, при заполнении специальных форм. Возможности же использования в них запросов на естественном языке (ЕЯ-запросов) ограничены простым поиском по ключевым словам. В данной области, насколько известно авторам, более совершенных методов обработки запросов для русского языка на практике не реализовано.

Но стоит заметить, что ЕЯ-запросы являются наиболее удобным для пользователя системы способом выражения своих мыслей. Исследование [6] показывает, что пользователи (на примере выборки из 48 человек) быстрее и с более высоким качеством справлялись с задачей поиска, используя системы с естественно-языковым интерфейсом (ЕЯ-интерфейсом). Сравнение проводилось с системами, где запросы составлялись графически и с помощью контекстных меню. ЕЯ-интерфейсы испытывают меньшие трудности при масштабировании и адаптации к новым форматам данных по сравнению с аналогами.

На первый взгляд, для поиска сотрудников по текстовым запросам приемлемо использовать традиционные методы поиска по коллекциям текстов вплоть до применения готовых технологий, например Google Custom Search Engine[2]. Но такие сис-

[1] Измерено по статистике посещаемости ресурсов на сервисе LiveInternet по состоянию на 1 ноября 2014 г.

[2] http://www.google.com/cse/

темы в силу своего общего назначения не полностью учитывают при поиске имеющуюся структуру информации. Поэтому они склонны выводить результаты, в которых найдены совпадения в неправильных полях, или же результаты, не соответствующие всем критериям.

С начала 2000-х годов можно наблюдать значительный прогресс в разработке лингвистических процессоров (ЛП) прикладных компьютерных систем. В частности, в разных странах и разных исследовательских центрах разрабатывались:

- ЕЯ-интерфейсы рекомендательных систем [7, 8];
- ЕЯ-интерфейсы вопросно-ответных систем для взаимодействия с онтологиями, разработанными в рамках или под влиянием проекта Семантический Веб [9, 10];
- преобразователи описаний фрагментов знаний на ЕЯ в наборы выражений языка проектирования онтологий OWL [11];
- ЛП для семантической обработки текстов по биологии и медицине как одно из центральных направлений биоинформатики [12].

Таким образом, в контексте современного состояния исследований по разработке лингвистических информационных технологий представляется актуальной проблема создания компьютерных систем семантического поиска специалистов по естественно-языковым описаниям их компетенций. Целью данной статьи, продолжающей линию работы [13], является разработка метода создания таких систем на основе применения теории K-представлений (концептуальных представлений) для формализации семантики запросов и отображения семантики слов и коротких словосочетаний в лингвистической базе данных. Важным рассматриваемым требованием при поиске специалистов является использование системой структуры имеющейся корпоративной БД.

Для достижения поставленной цели в статье решаются следующие задачи:

- выбирается способ формального представления структурированных значений ЕЯ-текстов;
- разрабатывается метод семантического поиска специалистов по ЕЯ-запросам;
- программно реализуется система семантического поиска, работающая с корпоративной БД.

Выбор подхода к формальному представлению значений текстов

С начала 1980-х годов исследователи из разных стран искали более удобные и широко применимые методы формального описания семантической структуры ЕЯ-текстов по сравнению с языками логики предикатов первого порядка (ЛППП) и более математически развитые подходы (более тонко структурированные) по сравнению с теорией семантических сетей. В этот период были, в частности, разработаны теория представления дискурсов (Discourse Representation Theory), теория концептуальных графов (Theory of Conceptual Graphs), эпизодическая логика (Episodic Logic), теория расширенных семантических сетей, компьютерная семантика русского языка и теория неоднородных семантических сетей.

Среди всех указанных подходов своим замыслом и результатами выделяется теория K-представлений (концептуальных представлений). Эта теория, первоначально называвшаяся теорией K-исчислений и K-языков, представлена в большой серии публикаций на русском и английском языках, в том числе в работах [14—21]. В отличие от всех перечисленных выше подходов, цель разработки теории K-представлений (ТКП) заключалась в поиске системы операций на концептуальных структурах, позволяющих шаг за шагом строить формальное представление структурированного значения, или семантическое представление (СП), произвольно сложных предложений и связных текстов (или дискурсов) на русском, английском, немецком, французском и других языках; совокупность таких языков в лингвистике называется естественным языком (ЕЯ).

Базовая модель ТКП описывает систему, состоящую из 10 частичных операций на концептуальных структурах. Эта модель задает новый класс формальных языков — класс СК-языков (стандартных концептуальных языков) [14—17, 20, 21]. Общим преимуществом СК-языков по сравнению с перечисленными подходами к формализации семантики ЕЯ-текстов является уникальный набор новых выразительных механизмов, открывающих возможности моделирования семантической структуры словосочетаний и текстов следующих видов (при сохранении выразительных средств ЛППП):

- инфинитивных и герундиальных конструкций (выражающих действия, цели, обязательства и т. д.);
- фраз со сложными придаточными предложениями, в том числе с прямой и косвенной речью, с придаточными цели;
- произвольно сложных составных обозначений множеств и понятий;
- связных текстов (дискурсов) со ссылками на смысл предыдущих или последующих фраз и более крупных частей текста;
- обозначений упорядоченных наборов объектов [14—17, 20, 21].

ТКП включает широко применимую математическую модель лингвистической базы данных (ЛБД) и несколько сильно структурированных алгоритмов семантико-синтаксического анализа ЕЯ-текстов, использующих эту модель ЛБД [17—19, 21]. Модель и алгоритмы применены при разработке рекомендательной системы с ЕЯ-интерфейсом [22], информационно-поисковой системы, реализующей семантический поиск по ЕЯ-запросам пользователей [23] и при проектировании системы управления файлами с ЕЯ-интерфейсом [24]. Теория K-представ-

лений включает оригинальную концепцию преобразования Веба в Мультилингвистический Семантический Веб [21, 25].

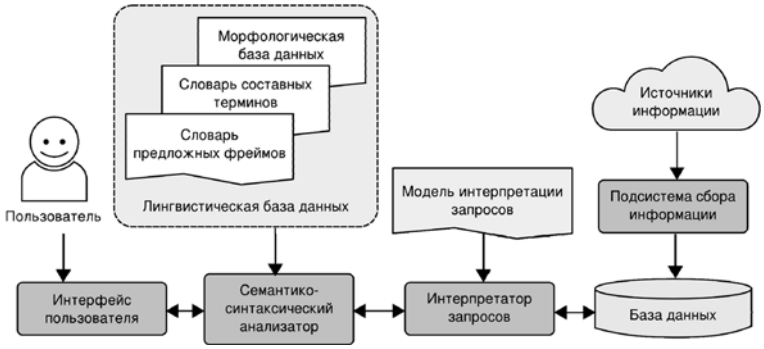
На основании перечисленных выше аргументов в качестве базового подхода к формальному описанию семантической структуры ЕЯ-текстов в проводимом исследовании была выбрана теория К-представлений.

Для построения СП ЕЯ-текстов из первичных информационных единиц и нескольких служебных символов теория К-представлений вводит набор некоторых правил Р[0]—Р[10]. Каждое из правил задает частичную операцию на множестве концептуальных структур. По гипотезе В. А. Фомичева [14—17, 20, 21] данный набор правил позволяет строить СП сколь угодно сложных ЕЯ-текстов. Перечислим здесь лишь те из них, которые непосредственно используются в работе:

- правило Р[0] задает по концептуальному базису множество простейших формул (простейшие и сложные формулы основного класса называются К-цепочками);
- по правилу Р[1] образуются К-цепочки вида *int.qtr concept*, где *int.qtr* — интенсиональный квантор (семантическая единица, соответствующая словам "некоторый", "произвольный", "все" и т. д.), *concept* — простое или составное обозначение понятия;
- правила Р[2] и Р[3] позволяют соответственно строить К-цепочки видов $f(a_1, ..., a_n)$ и $(b_1 \equiv b_2)$, где $n \geqslant 1$, f — n-арный функциональный символ, $a_1, ..., a_n, b_1, b_2$ — К-цепочки;
- с помощью Р[4] строятся К-цепочки вида $r(a_1, ..., a_n)$, где $n \geqslant 1$, r — n-арный реляционный символ;
- правило Р[7] использует логическую связку s (конъюнкция $\wedge$ или дизъюнкция $\vee$) для построения К-цепочек вида $(a_1 \text{ s } a_2 \text{ s } ... \text{ s } a_n)$, где $n > 1$;
- по правилу Р[8] строятся К-цепочки вида $c * (r_1, b_1)...(r_n, b_n)$, где c — базовое понятие, описывающее объект, а $r_1, ..., r_n$ — бинарные отношения и $b_1, ..., b_n$ — соответствующие значения вторых атрибутов;
- правило Р[10] нужно для формирования обозначений упорядоченных наборов объектов — К-цепочек вида $<a_1, ..., a_n>$, где для $i = 1, ..., n$ элемент a_i — более простая К-цепочка.

Структура разработанной поисковой системы

Для реализации предлагаемого подхода к семантическому поиску специалистов разработана система-прототип ExpSearch (Experts Search). Ее структура изображена на рисунке. Она похожа на структуру системы, предложенной в работе [13], но отличается составом лингвистической базы данных (ЛБД) и выделением отдельного компонента для интерпретации запросов вместо подсистемы поиска по ключевым словам. Предполагается, что перед началом использования системы имеется некоторая



Структура системы ExpSearch

локально доступная БД с загруженной в нее информацией о специалистах. Также предполагается, что эта БД обновляется с некоторой периодичностью, в связи с чем в структуру системы включена подсистема сбора информации. В частности, в работе используется БД, собранная на основе данных из Интернета с помощью указанной подсистемы.

Человек взаимодействует с системой через интерфейс пользователя, его основа — поле для ввода запроса. Семантико-синтаксический анализатор служит для построения СП запросов пользователя. В работе он опирается на ЛБД, служащую для связи слов, входящих в запросы, с соответствующими им концептуальными единицами и для нахождения семантико-синтаксических связей между элементами текста. На выходе анализатор выдает СП запроса в установленном формате.

Интерпретатор запросов по построенному СП пользовательского запроса конструирует запрос к БД, записанный на стандартном языке, например SQL или JSON. Модель интерпретации запросов показывает, как транслировать конструкции используемого СК-языка в формальные конструкции целевого языка для составления синтаксически корректных запросов.

- ***Используемые данные***

В качестве источника фактической информации о наборе специалистов в данной работе было решено использовать информацию о сотрудниках Национального исследовательского университета "Высшая школа экономики" (НИУ ВШЭ), размещенную на официальном сайте вуза[3]. Данный выбор обусловлен нахождением всех материалов в открытом доступе, формальностью описания компетенций и большим объемом выборки специалистов.

Профили сотрудников ВШЭ содержат следующие разделы (звездочкой отмечены обязательные):

1) фамилия, имя и отчество*;

2) подразделение и занимаемая должность*;

3) высшее образование (уровень, вуз, факультет, специальность и год окончания);

4) опыт работы (периоды трудоустройства и названия организаций);

[3]http://www.hse.ru/org/persons/

5) проводимые курсы в текущем академическом году (номер курса, факультет, специальность, название);

6) профессиональные интересы;

7) публикации (авторы, название, сведения об издании).

Приведенный набор данных содержит информацию как о достижениях сотрудников, так и о сферах их интересов. Он в значительной степени отражает компетенции специалистов и может быть использован для поиска по ним.

Используемая системой информация хранится в БД и имеет определенный формат. Так, сведения об образовании представлены в виде пятерок <*уровень образования, институт, факультет, специальность, год выпуска*>.

- ***Структура концептуального базиса***

В теории К-представлений на первом этапе определения класса СК-языков вводится понятие концептуального базиса (к.б.). Задание произвольного к.б. равносильно заданию длинного набора формальных объектов: множеств символов, выделенных элементов множеств, некоторой функции и двух бинарных отношений на одном из множеств. Такой набор задает, в частности, множество X рассматриваемых первичных информационных единиц и сведения, связанные с элементами множества X и необходимые для построения СП ЕЯ-текстов. В множестве X выделяется конечное подмножество St, элементы которого называются сортами и интерпретируются как наиболее общие понятия, рассматриваемые в выбранной области (физический объект, организация и т. д) [15—17].

В проведенном исследовании для описания сущностей из рассматриваемой области введена специально адаптированная для нее система сортов. Например, введены сорта, относящиеся к работе и образованию специалиста: *длительность*, *организация*, *должность* и *год-выпуска*. Заметим, что введенные сорта можно упорядочить с помощью отношения общности *Gen* (частичного порядка), задающего иерархию на множестве сортов *St*. Например, сорт *время* является более общим, чем сорта *длительность* и *год-выпуска*, что дает пары (*время*, *длительность*) и (*время*, *год-выпуска*), входящие в отношение *Gen*.

- ***Формат семантических представлений запросов***

Заметим, что традиционные системы поиска специалистов полагаются на заполнение полей форм для задания соответствующих критериев отбора. Данный подход является обоснованным, особенно с учетом того, что сами специалисты привыкли структурировать информацию о себе по не связанным друг с другом категориям, содержащим записи в единообразном формате. Соответственно, базы данных имеют схожую структуру, и в нашей системе логично использовать формат запросов, который бы позволял в текстовом виде одновременно задавать множество критериев отбора.

Для представления смысла запросов наиболее подходят К-цепочки вида $ref\ c * (r_1, b_1)...(r_n, b_n)$, построенные с применением правил Р[8] и Р[1]. Здесь *ref*-квантор референтности (в примерах — единица *нек*, от слова "некоторый"), c — базовое понятие, описывающее объект (в нашем случае — *чел*, "человек"), $r_1, ..., r_n$ и $b_1, ..., b_n$ — бинарные отношения и значения их вторых атрибутов [15—17].

Заметим, что критерии отбора могут быть составными. Например, для описания характеристик образования требуется определить одно из нескольких названий вуза, год выпуска и специальность. Для этого будем использовать составные значения $(b_i, i = 1, ..., n)$, построенные по правилу Р[10]. Это К-цепочки вида $<a_1, ..., a_n>$, где $a_1, ..., a_n$ — отдельные более простые К-цепочки. Для каждого типа отношений последовательность и число элементов определяются схемой информации, содержащейся в БД. Если в отношении определено меньше критериев отбора, чем предусмотрено, на недостающие места ставятся нулевые элементы *nil*. Если критерий предполагает выбор из нескольких вариантов, для их комбинации используем правило Р[7], определяющее операции конъюнкции и дизъюнкции ("$\wedge$" и "$\vee$").

Предложена также следующая модификация языка СП. Для представления выражений вида "*⩾ 2 года*" расширен язык СП входных запросов путем включения в него конструкций вида "символ бинарного отношения + число + единица измерения".

Основываясь на приведенном формате запросов, введем набор отношений для задания критериев отбора специалистов, а также рассмотрим простейшие примеры запросов и их СП:

- область профессиональных интересов — (*Компетенция*, *область-знания*).

Пример 1: "человек, интересующийся теорией коллективного выбора".

СП 1: *нек чел * (Компетенция, теория-коллективного-выбора)*.

Пример 2: "специалист по встраиваемым системам".

СП2: *нек чел * (Компетенция, встраиваемые-системы);*

- подразделение и должность — (*Работа*, <*организация, должность*>)

Пример: "профессор кафедры менеджмента".

СП: *нек чел * (Работа, <кафедра-менеджмента, профессор>);*

- опыт работы — (*Опыт-работы*, <*время, организация, должность*>)

Пример: "специалист, работавший в Microsoft не менее 2 лет".

СП: *нек чел * (Опыт-работы, <⩾ 2/год, Microsoft, nil>);*

- образование — (*Оконч.уч.зав.*, <*уровень-образования, институт, факультет, специальность, год-выпуска*>).

Пример: "выпускник МГУ по специальности "мировая экономика".

СП: *нек чел * (Оконч.уч.зав., <nil, МГУ, nil, мировая экономика, nil>)*;

- ведение учебных курсов — (*Преподавание*, *<дисциплина, организация, номер-курса>*).

Пример: "преподаватель дискретной математики".

СП: *нек чел * (Преподавание, <дискретная математика, nil, nil>).*

Такой подход к конструированию запросов позволяет составлять сложные и выразительные структуры. Например, рассмотрим возможный запрос к системе: "выпускник МГУ или МИЭМ, или МИФИ, имеющий не менее чем двухлетний опыт работы программистом". Тогда возможное СП этого запроса: *нек чел * (Оконч.уч.зав., <nil, (МГУ ∨ МИЭМ ∨ ∨ МИФИ), nil, nil, nil>) (Опыт-работы, <⩾2/год, nil, программист>).*

Алгоритм обработки запросов

Для того чтобы стало возможным представлять структурированные значения текстов в выбранной ранее форме (в виде К-представлений), требуется разработать алгоритм, позволяющий по фрагменту текста на ЕЯ автоматически генерировать соответствующее ему СП, а затем интерпретировать его и осуществить поиск с выделенными критериями отбора.

В книгах [17, 21] предложены алгоритмы такого рода SemSyn и SemSynt1, реализующие метод преобразования ЕЯ-текстов в СП, изложенный в работах [18, 19]. Но эти алгоритмы не устанавливают предпочтительный формат СП, без чего чрезвычайно сложно строить запросы к БД. В алгоритмах SemSyn и SemSynt1 также не предусмотрена и обработка текстов с ошибками.

В данной работе предлагается алгоритм, являющийся модификацией алгоритма SemSynt1 и ориентированный на применение в разработанной системе поиска специалистов. В частности, предложен метод выявления составных единиц в текстах. Алгоритм позволяет строить СП установленного вида без формирования матричного семантико-синтаксического представления текста. Работа алгоритма включает следующие этапы.

1. Предварительная обработка текста. Поступивший на вход алгоритма текст разбивается на токены — последовательности символов, соответствующие элементам текста: словам и разделителям. Пример: "человек, работавший в Microsoft не менее 2 лет" → ("человек", ",", "работавший", "в", " Microsoft", "не", "менее", "2", "лет").

Затем для токенов, представляющих собой слова, выполняется морфологический анализ. По информации, содержащейся в окончаниях слов, с помощью грамматического словаря определяются части речи, базовые формы слов и их морфологические характеристики.

В данной работе для обозначения морфологических признаков применяется нотация, используемая в Открытом корпусе русского языка — OpenCorpora[4]. Следует заметить, что обозначения, используемые для описания различных признаков, не совпадают друг с другом. Это позволяет представлять морфологические характеристики слов в виде неупорядоченных списков, что отличается от морфологического представления, описанного в работах [17, 21].

Таким образом, морфологический анализ представляет собой отображение *слово* → (*lec, Morph*), где *lec* — начальная форма слова (записанная прописными буквами), а *Morph* — список обозначений его морфологических характеристик.

2. Проекция компонентов словаря составных терминов. Для сопоставления слов ЕЯ, переведенных в начальную форму, с единицами семантического уровня удобно использовать лексико-семантический словарь (ЛСС), предложенный в работах [17, 21]. Он задает отображение (*lec, pt*) → (*sem, st_1, ..., st_k*). Начальной форме, или лексеме, *lec* и части речи *pt* слова ставится в соответствие некоторая семантическая единица *sem*, ассоциированная с набором сортов st_1, ..., st_k. т. е. имеющая тип $st_1 * ... * st_k$.

При таком подходе словам с ошибками единицы семантического уровня не будут сопоставлены вообще. Для замены ЛСС предлагается словарь составных терминов *Terms*, представляющий собой его обобщение. Он является массивом записей вида (*Lecs*, *sem*, *st*), компоненты которых интерпретируются следующим образом: *Lecs* — набор лексем слов, соответствующих одной единице информационного уровня (пример: единице *научно-учебная лаборатория-макроэкономического-анализа* соответствует набор {"научно-учебный", "лаборатория", "макроэкономический", "анализ"}); *sem* — К-цепочка (возможно, составная) со значением рассматриваемого словосочетания (пример: словосочетанию "старший преподаватель" сопоставляется цепочка *нек чел ∗* (*Работа*, *<nil*, *старший преподаватель>*), так как это слово определяет должность обсуждаемого человека. Наконец, *st* — это обозначение сорта единицы *sem* (пример: единица *НИУ-ВШЭ* имеет сорт *организация*). Несколько возможных записей из ЛСС приведены в таблице. В качестве ин-

Пример записей словаря составных терминов

Lecs	*sem*	*st*
специалист	*нек чел*	*интел.система*
профессор	*нек чел * (Работа, <nil, профессор>)*	*интел.система*
выпускник	*нек чел * (Оконч.уч.зав., <nil, nil, nil, nil, nil>)*	*интел.система*
бухгалтерия	*бухгалтерия*	*организация*
высший, школа, менеджмент	*высшая-школа-менеджмента*	*организация*
наука, о, данные	*науки-о-данных*	*специальность*
дискретный, математика	*дискретная-математика*	*дисциплина*

[4]http://opencorpora.org/dict.php?act=gram

дикатора отсутствия значения используется цепочка *nil*.

Алгоритм применения словаря составных терминов можно представить следующим образом. В качестве предварительного шага строится отображение (хэш-таблица) *Map*, сопоставляющее лексемам номера записей словаря составных терминов *Terms*, в которых употребляются эти лексемы. Формально, *Map*: $lec \rightarrow \{i | lec$ — компонент набора *Lecs*, являющегося *i*-й записью словаря *Terms*$\}$.

При разборе запроса каждому слову с помощью *Map* сопоставляется набор номеров релевантных ему записей. Если лексема *lec* не входит в *Map*, выбираются все элементы из области определения *Map*, расстояние Левенштейна до которых меньше некоторой величины λ, и операция проводится для них.

При проходе слева направо алгоритм комбинирует номера записей, к которым могут относиться слова запроса, и последовательно выделяет наибольшие по объему подмножества слов, имеющих одну общую запись словаря. Каждому слову из подмножества сопоставляется семантическая единица *sem* и сорт *st* из соответствующей записи. Работа алгоритма прекращается, когда всем словам поставлены в соответствие семантические единицы.

Предложенный алгоритм применения словаря составных терминов позволяет правильно обрабатывать широкий диапазон случаев искаженного написания по сравнению с записями словаря, например, опечатки (в том числе "слипания" и неправильные разбиения слов), употребление буквы "ё", а также пропуск слов или использование вместо них сокращений.

Отметим еще одну особенность алгоритма. Если после данного шага осталось много слов, не сопоставленных с семантическими единицами, с высокой степенью уверенности можно сказать, что данный запрос не будет правильно разобран. Это может происходить не только из-за неполноты словаря составных терминов или несовершенства метода разбора, но прежде всего из-за ввода запросов, не имеющих отношения к поиску специалистов.

3. Проекция словаря предложных фреймов. После выделения семантических единиц на основе слов запроса установим связи между ними для последующего построения СП. Для этого мы используем информацию о семантико-синтаксических связях между словами запроса.

В работах [17, 21] предложены словарь глагольно-предложных фреймов и словарь предложных семантико-синтаксических фреймов. Они задают связи в сочетаниях вида "глагольная форма—предлог—существительное" и "существит. 1—существит. 2" соответственно. Для того чтобы унифицировать процесс их применения, в данной работе предлагается обобщенный словарь предложных фреймов. Его записи имеют вид (st_1, $Morph_1$, *prep*, st_2, $Morph_2$), где *prep* — необходимый предлог между словами (может быть пустым); st_1, st_2 — обозначения сортов, которые можно связать с первым и вторым словом в лингвистически правильном словосочетании "$word_1$ + *prep* + $word_2$" соответственно; $Morph_1$, $Morph_2$ — обозначения характеристик, которые должны иметь первое и второе слово соответственно (части речи, формы глагола — время/лицо/число, формы существительного — число/падеж и т. д.).

Если пара слов запроса удовлетворяет всем условиям какой-то записи словаря, считается, что между ними существует связь, и второе слово зависит от первого. Поэтому для установления связей между словами выполним следующую процедуру. Будем последовательно брать упорядоченные пары слов из запроса, их число равно $n(n-1)$, где n — длина запроса в словах за вычетом знаков пунктуации и слов-связок (союзы, предлоги и т. п.). Для каждой пары проверим входящие в нее слова на соответствие условиям каждой из записей словаря. Если найдено совпадение, считается, что между этими словами имеется смысловая связь. Пара запоминается как связанная, и дальнейшая сверка по словарю для нее прекращается. Таким образом, информация о связях слов представляется асимметричным бинарным отношением $Rels \subset W \times W$ на множестве слов W, которое передается на следующий шаг алгоритма.

4. Построение семантического представления. На данном шаге используем связи между отдельными словами, выделенными на предыдущих этапах, чтобы определить связи между семантическими единицами и составить итоговое семантическое представление запроса. Заметим, что для этого слова запроса, представляющие одну и ту же семантическую единицу *sem*, не должны быть обязательно связаны друг с другом отношением *Rels*, так как они уже объединены своим значением. То есть нас интересуют только связи между словами, представляющими разные единицы.

Рассмотрим все подобные связи. Согласно направлению каждой связи будем называть две рассматриваемых единицы главной и зависимой. Если в главной единице есть пустое поле (на месте аргумента стоит ключевое слово *nil*) и при этом сорт зависимой единицы совпадает с требованием к сорту этого поля (или соответствующие сорта входят в отношение *Gen*), зависимая единица подставляется в пустое поле главной. В итоге это дает на выходе финальное СП.

Заметим, что система до этого шага не использовала информацию о содержащихся в тексте пунктуационных символах. Если на данном этапе две и более семантические единицы сопоставляются с одним и тем же полем другой единицы, то по стоящим запятым и использованию союзов ("и" или "или") определяется, какую логическую связку использовать в формуле ($\wedge$ или $\vee$). Это делает систему более устойчивой к некорректному вводу, так как пользователь мог пропустить символ (например, запятую между существительным и стоящим после

него зависимым причастием) или добавить лишний символ.

5. Составление запроса к базе данных. На заключительном этапе по построенному СП ЕЯ-запроса пользователя создается запрос к БД. Выбранная нами структура семантических представлений позволяет очень удобно преобразовывать их в запросы к хранилищу структурированной информации, например, на языке SQL. Если в СП используются функции *Работа*, *Оконч.уч.зав.*, *Опыт-работы* или *Преподавание*, их аргументы рассматриваются как критерии, по которым отбираются специалисты. Так, при использовании одной функции *Работа* будут выбраны только специалисты с совпадающими аргументами типа *место-работы* и *должность*, если на месте одного из них в запросе не стоит ключевое слово *nil*. Это достигается тем, что в шаблон запроса подставляются условия в зависимости от содержания функций.

В качестве примера покажем, как данные конструкции могут быть спроецированы в язык SQL. Рассмотрим функцию *Работа*. Она может задавать два критерия отбора — подразделение и должность. Тогда для условного запроса с общей частью "select person.id from position, department where person.id= department.person_id and ..." вводятся два условных выражения: "position.department=..." и "position.position=...". Они заполняются соответствующими значениями функции, подставляются в секцию "where" запроса и связываются условием "and", если условий больше одного.

При использовании аргументов вида "⩾ 2 года" и подобных для соответствующих полей БД о специалистах осуществляется проверка на соответствие указанному логическому условию. Тогда формат условий немного изменяется, например, для опыта работы: "experience.duration >= 2".

Аналогичным образом обрабатывается СП, если запрос нужно сформировать в формате JSON (для No-SQL хранилища). Отличие заключается в формате представления. Так, основа запроса — пустой словарь "{}", куда добавляются элементы с критериями. Например, для отношения *Работа*: ""position": {"department": ..., "position": ...}".

Описанный этап является последним в алгоритме обработки запроса пользователя. После него сформированный структурированный запрос выполняется СУБД, и информация из профилей сотрудников, удовлетворяющих запросу, возвращается пользователю через графический интерфейс.

Программная реализация системы

Для реализации системы поиска был выбран язык программирования Python. Это высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис языка минималистичен, но его стандартная библиотека включает большой объем полезных функций. Язык выделяется наличием мощных встроенных типов данных, таких как строки, массивы и словари, что делает его удобным средством для разработки лингвистических процессоров.

Так как у авторов нет доступа к БД сотрудников ВШЭ, было решено составить достаточно точную ее локальную копию по информации, которую предоставляет веб-сайт НИУ ВШЭ. Для разбора разметки HTML использовался модуль lxml для языка Python. Он представляет собой интерфейс к библиотеке libxml, предназначенной для построения моделей документов (document-object model, DOM) по HTML-разметке и манипуляции ими.

В качестве СУБД используется нереляционное хранилище MongoDB. Все данные в нем представлены в нотации JSON, а именно, в виде пар "ключ—значение" (key—value pairs) и упорядоченных множеств (collections). Так как все транзакции в ней подчиняются критериям ACID (Atomicity, Consistency, Isolation, Durability), работу подсистемы сбора информации можно в любой момент прервать и возобновить, при этом целостность данных не пострадает.

Для морфологического анализа слов использована библиотека pymorphy2 — морфологический анализатор для русского языка. Для слов, включенных в словарь составных терминов, она показала точность распознания более чем 95 %.

Для отображения пользовательского интерфейса использована библиотека PySide — программная среда для создания графических интерфейсов, основанная на библиотеке Qt. При работе система демонстрирует пользователю СП его запроса и найденные профили сотрудников, а на другой вкладке — более подробную информацию о выбранном сотруднике. Разработан также аналогичный веб-интерфейс с использованием сетевого фреймворка Flask.

Система протестирована на работоспособность на компьютере с ОС Windows 8.1 и Ubuntu 14.04, процессором Intel Core i7 (4 ядра, тактовая частота 1.9 ГГц) и ОЗУ объемом 10 гибибайтов. Система также способна работать на системах Mac OS/Unix с необходимым окружением.

В качестве тестовых данных была использована информация о 8039 сотрудниках НИУ ВШЭ, снятая с их профилей на официальном сайте (http://www.hse.ru) 1 ноября 2014 г. Она была предварительно извлечена из Интернета и загружена в локальную базу данных (MongoDB).

Пример результатов тестирования. Согласно определенному формату запросов составим множество примеров запросов к системе: "сотрудник кафедры маркетинга фирмы"; "профессор, ведущий курс лекций по базам данных"; "стажер-исследователь, закончивший до 2010 года специалитет Высшей школы экономики по специальности "Менеджмент". Прототип системы построил правильные СП для данных запросов и успешно провел по ним поиск. По данным запросам система вывела 29, 12 и 1 результат соответственно.

Заключение

В ходе проведенного исследования разработан метод семантического поиска специалистов по ЕЯ-запросам, работающий с корпоративным хранилищем данных. На основе использования методологии теории К-представлений описан формат запросов, позволяющий определять набор критериев отбора специалистов и эффективно осуществлять по ним поиск. Предложена модифицированная модель лингвистической БД (ЛБД) по сравнению с моделями, введенными в работах [17, 21]. Эти два результата стали отправной точкой разработки и программной реализации на языке Python алгоритма построения СП-запросов пользователя, а из них — запросов к структурированному хранилищу информации.

Научная новизна исследования заключается в следующем:

- предложен новый практически полезный алгоритм разбора естественно-языковых выражений, рассматривающий группы слов для связывания их с информационными единицами и успешно обрабатывающий случаи некорректного пользовательского ввода;
- разработана расширенная модель ЛБД, упрощающая структуру словарей семантико-синтаксических фреймов;
- в язык семантических представлений, являющийся СК-языком в рассматриваемом концептуальном базисе, введены дополнительные выразительные механизмы.

Кроме того, следует заметить, что разработанный метод может быть применен для анализа текстов как на других естественных языках (например, на английском или немецком), так и для других профессиональных задач. Например, его перспективной областью применения может стать создание естественно-языковых интерфейсов для извлечения информации из онтологий посредством создания запросов на языке SPARQL.

В связи с этим наметим следующие направления для продолжения исследования:

- сбор данных о ЕЯ-запросах пользователей и изучение особенностей их обработки системой;
- поддержка более сложных на семантическом уровне форматов запросов;
- интернационализация и повышение универсальности алгоритма анализа ЕЯ для обеспечения возможности его применения в других задачах.

Проведенное исследование показало удобство использования на практике аппарата СК-языков, предложенного теорией К-представлений, и математической модели лингвистической базы данных.

Список литературы

1. **Lin C.-Y., Cao N., Liu S. X., Papadimitriou S., Sun J., Yan X.** SmallBlue Social Network Analysis for Expertise Search and Collective Intelligence // IEEE 25th International Conference on Data Engineering. ICDE '09.2009. P 1483—1486.
2. **Farrell R. G., Pan F.** Computing Similarities between Natural Language Descriptions of Knowledge and Skills // RC24060 Computer Science, IBM Research Division. 2006. V. 25.
3. **Bull S., Greer J., McCalla G., Kettel L., Bowes J.** User modelling in I-Help: What, why, when and how // User Modeling 2001. Springer, 2001. P. 117—126.
4. **Staab S.** Human language technologies for knowledge management // Intelligent Systems, IEEE. 2001. V. 16, N. 6. P. 84—94.
5. **Maybury M. T.** Expert Finding Systems. MITRE Technical Report. Bedford, Massachusetts: The MITRE Corporation, 2006. 52 p.
6. **Kaufmann E., Bernstein A.** How useful are natural language interfaces to the semantic web for casual end-users? // The Semantic Web. Springer. 2007. P. 281—294.
7. **Chai J., Horvath V., Nicolov N., Stys M., Kambhatla N., Zadrozny W., Melville P.** Natural Language Assistant — A Dialog System for Online Product Recommendation // AI Magazine. 2002. V. 23. N. 2. P. 63—76.
8. **Berger H., Dittenbach M., Merkl D.** An Adaptive Multilingual Interface for Tourism Information // International Journal of Electronic Business. 2004. V. 2, N. 5. P. 531—541.
9. **Wang C., Xiong M., Zhou Q., Yu Y.** PANTO: a portable NL-interface to ontologies // 4th European Semantic Web Conference Proceedings. Springer. 2007. P. 473—487.
10. **Cimiano P., Haase P., Heizmann J., Mantel M., Studer P.** Towards Portable Natural Language Interfaces to Knowledge Bases — the Case of the ORAKEL System // Data and Knowledge Engineering (DKE). 2008. V. 65. N. 2. P. 325—354.
11. **Schwitter R.** Creating and querying formal ontologies via controlled natural language // Applied Artificial Intelligence. 2010. V. 24, N. 1. P. 149—174.
12. **Prince V., Roche M.** Information Retrieval in Biomedicine: Natural Language Processing for Knowledge Integration. Hershey, Pennsylvania: IGI Global, 2009.
13. **Захлебин И. В.** Использование семантического анализа текстов для поиска специалистов // Supplementary Proceedings of AIST 2014. — CEUR Workshop Proceedings. 2014. V. 1197. P. 187—191 (in Russian).
14. **Fomichov V. A.** A Mathematical Model for Describing Structured Items of Conceptual Level // Informatica. An International Journal of Computing and Informatics (Slovenia). 1996. V. 20, N. 1. P. 5—32.
15. **Фомичев В. А.** Математические основы представления смысла текстов для разработки лингвистических информационных технологий. Часть I // Информационные технологии. 2002. № 10. С. 16—25.
16. **Фомичев В. А.** Математические основы представления смысла текстов для разработки лингвистических информационных технологий. Часть II // Информационные технологии. 2002. № 11. С. 34—45.
17. **Фомичев В. А.** Формализация проектирования лингвистических процессоров. М.: МАКС Пресс, 2005. 368 с.
18. **Фомичев В. А.** Понятие текстообразующей системы как компонент нового формального аппарата для проектирования лингвистических процессоров // Информационные технологии. 2005. № 8. С. 22—27.
19. **Фомичев В. А.** Новый метод преобразования естественно-языковых текстов в семантические представления // Информационные технологии. 2005. № 10. С. 25—35.
20. **Фомичев В. А.** Математические основы представления содержания посланий компьютерных интеллектуальных агентов. М.: ГУ-ВШЭ, ТЕИС, 2007. 176 с.
21. **Fomichov V. A.** Semantics-Oriented Natural Language Processing: Mathematical Models and Algorithms. New York, Dordrecht, Heidelberg, London; Springer, 2010. 354 p.
22. **Правиков А. А., Фомичев В. А.** Разработка рекомендательной системы с естественно-языковым интерфейсом на основе математических моделей семантических объектов // Бизнес-информатика. Междисциплинарный научно-практический журнал ГУ-ВШЭ. 2010. № 4(14). С. 3—11.
23. **Fomichov V. A., Kirillov A. V.** A Formal Model for Constructing Semantic Expansions of the Search Requests about the Achievements and Failures // Artificial Intelligence: Methodology, Systems, and Applications / Ed. by A. Ramsay, G. Agre. Lecture Notes in Computer Science. V. 7557. Berlin, Heidelberg: Springer, 2012. P. 296—304.
24. **Razorenov A. A., Fomichov V. A.** The Design of a Natural Language Interface for File System Operations on the Basis of a Structured Meanings Model // Procedia Computer Science, Elsevier. 2014. V. 31. P. 1005—1011; open access, URL: http://authors.elsevier.com/sd/artic1e/S1877050914005304.
25. **Fomichov V. A.** Theory of K-representations as a Comprehensive Formal Framework for Developing a Multilingual Semantic Web // Informatica. An Intern. Journal of Computing and Informatics (Slovenia). 2010. V. 34, N. 3. P. 387—396.

I. V. Zakhlebin, Master student, e-mail: izakhlebin@hse.ru,
V. A. Fomichov, Professor, e-mail: vfomichov@hse.ru,
National Research University Higher School of Economics (HSE), Moscow, Russia

Development of a Method for Semantic Search of Specialists in Corporate Databases Using Natural Language Queries

In the context of the studies on natural language processing, this paper substantiates the topicality of semantic search for specialists based on natural language queries. It also states the requirements to applied computer systems aimed at solving this problem, such as the necessity to extensively use semantics of natural language and structure of an available corporate database in the process of search. The principal ideas for development of such computer systems are set forth. A significant distinguished feature of the proposed method is the usage of SK-languages (standard knowledge languages), introduced in the V. A. Fomichov's theory of K-representations (knowledge representations), for formalizing semantics of natural language queries and for reflecting semantics of words and short word combinations in linguistic databases. The described method underpinned the design of a semantic search system ExpSearch, it was implemented in the programming language Python.

Keywords: *semantic search of specialists, natural language query, natural language query semantic representation, theory of K-representations, SK-language, conceptual basis, linguistic database, corporate database, semantic-syntactic analyzer, Python*

References

1. **Lin C.-Y., Cao N.** et al. SmallBlue: Social Network Analysis for Expertise Search and Collective Intelligence. *IEEE 25th International Conference on Data Engineering. ICDE'09. IEEE.* 2009. P. 1483—1486.

2. **Farrell R. G., Pan F.** Computing Similarities between Natural Language Descri ptions of Knowledge and Skills. *RC24060 Computer Science, IBM Research Division*. 2006. V. 25.

3. **Bull S., Greer J., McCalla G., Kettel L., Bowes J.** User modelling in I-Help: What, why, when and how. *User Modeling* 2001. Springer, 2001. P. 117—126.

4. **Staab S.** Human language technologies for knowledge management. *Intelligent Systems, IEEE.* 2001. V. 16, N. 6. P. 84—94.

5. **Maybury M. T.** *Expert Finding Systems. MITRE Technical Report.* Bedford, Massachusetts: The MITRE Corporation, 2006. 52 p.

6. **Kaufmann E., Bernstein A.** How useful are natural language interfaces to the semantic web for casual end-users? *The Semantic Web*. Springer, 2007. P. 281—294.

7. **Chai J., Horvath V., Nicolov N., Stys M., Kambhatla N., Zadrozny W., Melville P.** Natural Language Assistant — A Dialog System for Online Product Recommendation. *AI Magazine*. 2002. V. 23. N. 2. P. 63—76.

8. **Berger H., Dittenbach M., Merkl D.** An Adaptive Multilingual Interface for Tourism Information. *International Journal of Electronic Business.* 2004. V. 2, N. 5. P. 531—541.

9. **Wang C., Xiong M., Zhou Q., Yu Y.** PANTO: a portable NL-interface to ontologies. *4th European Semantic Web Conference Proceedings*. Springer. 2007. P. 473—487.

10. **Cimiano P., Haase P., Heizmann J., Mantel M., Studer P.** Towards Portable Natural Language Interfaces to Knowledge Bases — the Case of the ORAKEL System. *Data and Knowledge Engineering (DKE)*. 2008. V. 65. N. 2. P. 325—354.

11. **Schwitter R.** Creating and querying formal ontologies via controlled natural language. *Applied Artificial Intelligence.* 2010. V. 24, N. 1. P. 149—174.

12. **Prince V., Roche M.** *Information Retrieval in Biomedicine: Natural Language Processing for Knowledge Integration*. Hershey, Pennsylvania: IGI Global, 2009.

13. **Zakhlebin I. V.** Ispolzovanie semanticheskogo analiza tekstov dlya poiska spetsialistov. *Supplementary Proceedings of AIST 2014.* — CEUR Workshop Proceedings. 2014. V. 1197. P. 187—191 (in Russian).

14. **Fomichov V. A.** A Mathematical Model for Describing Structured Items of Conceptual Level. *Informatica. An International Journal of Computing and Informatics* (Slovenia). 1996. V. 20. N. 1. P. 5—32.

15. **Fomichov V. A.** Matematicheskie osnovy predstavleniya smysla tekstov dlya razrabotki lingvisticheskikh informatsionnykh tekhnologii. Part I. *Informatsionnye Tekhnologii.* 2002. N. 10. P. 16—25 (in Russian).

16. **Fomichov V. A.** Matematicheskie oshovy predstavleniya smysla tekstov dlya razrabotki lingvisticheskikh informatsionnykh tekhnologii. Part II. *Informatsionnye Tekhnologii.* 2002. N. 11. P. 34—45 (in Russian).

17. **Fomichov V. A.** *Formalizatsiya proektirovaniya lingvisticheskikh protsessorov*. M.: MAKS Press, 2005. 368 p.

18. **Fomichov V. A.** Ponyatie tekstoobrazuyshchei sistemy kak component formalnogo apparata dlya proektirovaniya lingvisticheskikh protsessorov. *Informatsionnye Tekhnologii* 2005. N. 8. P. 22—27 (in Russian).

19. **Fomichov V. A.** Novyi metod preobrazovaniya yestestvennoyazykovykh tekstov v semanticheskie predstavleniya. *Informatsionnye Tekhnologii.* 2005. N. 10. P. 25—35 (in Russian).

20. **Fomichov V. A.** *Matematicheskie osnovy predstavleniya soderzhaniya poslanii kompyuternykh intellektualnykh agentov*. Moscow: State University —Higher School of Economics, Publishing House "TEIS". 2007. 176 p.

21. **Fomichov V. A.** *Semantics-Oriented Natural Language Processing: Mathematical Models and Algorithms.* New York, Dordrecht, Heidelberg, London; Springer, 2010. 354 p.

22. **Pravikov A. A., Fomichov V. A.** Razrabotka rekomendatelnoi sistemy s yestestvenno-yazykovym interfeisom na osnove matematicheskikh modelei semanticheskikh ob'ektov. *Biznes-informatika. Mezhdistsiplinarnyi nauchno-prakticheseskii zhurnal.* 2010. N. 4(14). P. 3—11.

23. **Fomichov V. A., Kirillov A. V.** A *Formal Model for Constructing Semantic Expansions of the Search Requests about the Achievements and Failures.* Artificial Intelligence: Methodology, Systems, and Applications. Ed. by A. Ramsay, G. Agre. Lecture Notes in Computer Science. V. 7557. Berlin, Heidelberg: Springer, 2012. P. 296—304.

24. **Razorenov A. A., Fomichov V. A.** The Design of a Natural Language Interface for File System Operations on the Basis of a Structured Meanings Model. *Procedia Computer Science*, Elsevier. 2014. V. 31. P. 1005—1011; open access, URL: http://authors.elsevier.com/sd/artic1e/S1877050914005304.

25. **Fomichov V. A.** Theory of K-representations as a Comprehensive Formal Framework for Developing a Multilingual Semantic Web. *Informatica. An Intern. Journal of Computing and Informatics* (Slovenia). 2010. V. 34, N. 3. P. 387—396.