УДК: 316.6

1.5. СПЕЦИАЛЬНЫЕ ВЫРАЖЕНИЯ ДЛЯ ПОИСКА В СТРУКТУРИРОВАННОМ ТЕКСТЕ С ИСПОЛЬЗОВАНИЕМ ГРАММАТИЧЕСКИХ СВОЙСТВ

Ночевной Д.С. [1]Бессмертный, И.А. [1], Клименков С.В. [1]
[1]Национальный исследовательский университет ИТМО

В данной статье рассматривается применение специально разработанных регулярных выражений для извлечения словоформ, а также семантических отношений, полученных из структурированных и слабоструктурированных источников, рассматриваются основные элементы семантической сети (концепты, лексемы, словоформы, отношения и атрибуты), а также основные типы связей между элементами. Новизну исследования составляет применение регулярных выражений не к символам, а к лексемам. Приводится классификация методов для автоматизированного извлечения семантических связей из текста. Представлено сравнение производительности разработанного алгоритма и утилиты «grep» с точки зрения количества квантификаторов, входящих в шаблоны для поиска.

Введение

Одним из ключевых элементов систем автоматической обработки текста являются онтологии, или тезаурусы. Обычно в этом качестве используются онтологии, или тезаурусы, построенные на базе тех или иных словарей. Общим недостатком таких онтологий является отсутствие специализированных терминов, специфичных для данной предметной области [Письмак, 2016]. Поэтому всегда актуальной является проблема дополнения существующей онтологии узлами из внешних источников.

С повышением производительности средств вычислительной техники, увеличением объема хранимой информации появилась возможность хранить и обрабатывать гигантский объем накопленной человечеством информации в электронном виде [Клименков, 2020]. Часто эти данные представлены как неструктурированные или слабоструктурированные данные в виде текстов. Важным средством представления и хранения знаний стали семантические сети. Для хранения структурированной информации широко используются разнообразные базы данных. Подходы к проектированию баз данных и реляционная алгебра хорошо изучены многочисленными исследователями. Однако сама задача извлечения информации из текстов и осуществление ее долгосрочного и эффективного хранения в виде базы данных является все более широко востребованной [Карташов, 2018].

Учеными в данной области уже разработаны несколько способов извлечения семантических связей из текстов на естественном языке. Например, Винстон, Чаффин и Герман [Winston, 1987] предложили исчерпывающий список языковых конструкций, позволяющих выделить такие связи. Аусеннак-Гиллес и другие [Aussenac-Gilles, 2009] рассмотрели способы извлечения подобных отношений из структурированных документов в формате XML. Давальцу и другие [Davulcu, 2003] ученые также занимались поиском таких отношений на веб-страницах. Миколов [Mikolov, 2013] предложил использование нейронной сети для нахождения синонимов.

Однако все перечисленные методы не позволяют достичь приемлемой точности и полноты, поэтому предметом данного исследования стало извлечение семантических связей из структурированных текстов с учетом грамматических признаков слов, а в качестве объекта исследования можно выделить семантические сети и их поддержание в актуальном состоянии. Целью работы стало расширение существующих онтологий, а также восстановление отсутствующих связей, а именно – меронимии, описывающей связь «часть-целое» [Письмак, 2016].

Из имеющихся исходных данных и цели были выявлены задачи, требующие решения:

- поиск различных текстовых блоков с помощью регулярных выражений с учетом грамматической информации;
- анализ блоков;
- добавление узлов и связей в исходную онтологию.

Онтологии и семантические сети

Онтология – значимые знания в какой-либо области, представленные в виде удобной структуры данных. Семантическая сеть – это ориентированный граф, узлами которого являются понятия (или смысловые значения), а ребрами – отношения между ними [Письмак, 2016]. Для любой онтологии критически важной является актуальность ее содержимого, а для поддержания ее в актуальном состоянии необходимо вооружиться методами регулярного дополнения и обновления онтологии.

В общем смысле онтология или семантическая сеть состоит из множества смысловых понятий, или концептов, связанных между собой семантическими отношениями, или связями. Концепты и связи могут иметь свойства или атрибуты, характеризующие их. Концепты, связи и атрибуты образуют структурный компонент семантической сети. Кроме этого, концепты, связи и атрибуты могут иметь экземпляры, образующие информационный компонент семантической сети. К информационному компоненту также относятся связанные с концептами леммы (канонические формы лексем), предназначенные для

образования множества словоформ, и глоссы (текстовые словарные определения соответствующего концепту понятия) [Клименков, 2020].

Рассмотрим подробнее следующие лингвистические единицы: лексемы, словоформы и «сенсы» (от английского «sense» – смысл, значение) [Письмак, 2016]. Лексема представляет собой какое-либо слово во всех возможных формах, а именно – в единственном и во множественном числе, а также в различных склонениях, родах или падежах в зависимости от части речи.

Словоформа – это конкретная форма слова [Письмак, 2016]. Для существительных словоформой будет слово в определенном числе и падеже. В качестве примеров словоформ можно привести следующие: «дерево», «деревья», «дереву», «деревьям», «деревом» и так далее. Очевидно, что лексема – это множество или агрегация всех соответствующих ей словоформ. Приведенные в примере выше словоформы можно объединить одной лексемой «дерево».

В то же время с каждой словоформой можно соотнести один или более «сенсов» – значение этой словоформы в данном контексте. Ярким примером наличия множества «сенсов» является словоформа «ключ», которая имеет, по крайней мере, 3 смысла в зависимости от использования в тексте: водный источник, ключ от замка или выключатель в электрической цепи. Каждый из этих смыслов является «сенсом», или смыслом словоформы.

Между лексемами в онтологии устанавливаются следующие основные семантические отношения [Cederberg, 2003]:

- синонимия (слова одной части речи, равные или близкие по значению, например, «радость/веселье», «луна/месяц»);
- антонимия (слова, имеющие противоположное значение, например, «белый/черный», «ночь/день»);
- гипонимия/гиперонимия (отношение частное-общее, например, «вода/жидкость», «дуб/дерево»; гипоним наследует все свойства гиперонима, это отношение является центральным отношением для описания существительных);
- меронимия (отношение часть-целое, в качестве примеров можно привести «двигатель/автомобиль», «комната/квартира»).

Меронимические связи, в свою очередь, подразделяются на следующие подтипы: связь «коллекция-элемент», связь «агрегат-элемент», связь «композит-элемент» [Клименков, 2020].

В рамках исследования была составлена классификация существующих способов извлечения семантических отношений из текста.

Рассмотрим основные группы методов, основанные на:

- шаблонах (поиск в тексте информации, соответствующей одному из шаблонов, описывающих связь, например ЦЕЛОЕ «содержит» ЧАСТЬ) [Winston, 1987] [Pismak, 2020];
- анализе объектной модели документа и форматирования текста (часто вложенные и родительские элементы обладают семантической связью) [Davulcu, 2003] [Aussenac-Gilles, 2009] [Thompson, 1968];
- машинном обучении (ярким примером этой категории является Word2vec [Mikolov, 2013], в котором использована нейронная сеть для составления списка наиболее вероятных синонимов для данного слова).

К сожалению, все перечисленные подходы являются узконаправленными и не могут полностью решить задачу добавления специализированных терминов в онтологию.

Исходя из рассмотренных материалов, была выдвинута гипотеза, что грамматические свойства структурированного текста могут быть использованы для улучшения поиска с использованием шаблонов.

Регулярные выражения для извлечения словоформ

Поскольку уже существующая онтология состоит из узлов и связей для русского языка, то и тексты для ее дополнения необходимо было найти на русском языке. В качестве входных данных для работы данного алгоритма был использован национальный корпус текстов русского языка (а именно его синтаксически размеченный подкорпус «СинТагРус»), который распространяется бесплатно по запросу для научных исследований.

В этом корпусе имеется более 600 текстов на различные темы, причем каждый текст состоит из сотен размеченных предложений. Корпус состоит из множества XML-файлов. Структура каждого файла подразумевает наличие информации об авторе, редакторе, дате создания и изменения текста, а также о его названии. Вся эта информация находится внутри двойного тега <inf>…</inf>. Далее, внутри тегов <body>…</body> перечисляются предложения (в тегах <S>…</S>) и отдельные слова с синтаксической разметкой (в тегах <W>…</W>). Для обработки документов в таком формате был разработан читатель на языке Python. Для каждой словоформы указаны часть речи и множество других признаков. Например, для существительных указывается число, род, одушевленность и падеж. Именно существительные и были выбраны для нашего исследования, поскольку связь «меронимия» может иметь место только между двумя существительными.

Как отмечают авторы статьи о «Методе семантического уточнения для корпоративного поиска» [Pismak, 2020], поисковые системы являются обязательной частью любой цифровой среды в совре-

менных корпоративных системах. В общем случае такая задача выполняется в текстовом корпусе из документов с помощью методов грамматического полнотекстового поиска. Этот способ работы с набором документов позволяет удовлетворить поисковые запросы пользователей, предоставляя достаточно подходящие результаты. Тем не менее, не все результаты такого поиска будут релевантными запросу. Для того чтобы увеличить релевантность поиска, многие авторы предлагают использовать полный лингвистический анализ, основанный на использовании правил. Такой подход дает неплохие результаты, но в основном сводится к предопределенному набору поисковых запросов.

Эта проблема по-прежнему остается актуальной из-за количества данных, подлежащих обработке. Это особенно важно для корпоративных систем, где задача поиска подразумевает огромное количество документов, а также однородность предметной области.

Одним из возможных подходов к решению проблемы является использование дополнительного источника информации, позволяющего расширить знания об обрабатываемом запросе и применять их для поиска в документах базы данных.

Следует подчеркнуть, что в нашем случае имеется лингвистическая онтология, содержащая множество узлов и связей. В том числе у нас имеется возможность получить все возможные словоформы для выбранной смысловой единицы. Поэтому было предложено использовать данные знания для улучшения поиска по набору документов.

Если рассматривать работы авторов, посвященные работе с регулярными выражениями, то можно сделать вывод, что за последние 10–20 лет не произошло каких-либо принципиальных изменений. Один из общепринятых подходов предполагает построение однонаправленного графа для регулярного выражения. Впоследствии этот однонаправленный граф может быть преобразован в недетерминированный конечный автомат, который можно использовать для поиска заданного фрагмента в тексте. Описания данного алгоритма в общем виде приводятся в статьях Кена Топсона [Thompson, 1968] и Расса Кокса [Cox, 2007].

Одним из наиболее интересных аспектов работы регулярных выражений является построение однонаправленного графа для заданного выражения или создание недетерминированного конечного автомата на его основе.

Рассмотрим основные строительные блоки для этой задачи (рисунок 1). Отметим, что каждая вершина графа означает сопоставление с некоторой словоформой в тексте за исключением некоторых служебных вершин графа, которые служат для обработки квантификаторов, а также для выделения начала и конца графа. Соответственно, ребрами графа являются однонаправленные переходы между вершинами. Мы начинаем обработку графа из некоторой начальной служебной вершины. Далее, после обработки каждой вершины мы обязаны перейти в следующее состояние с помощью одного из возможных переходов (ребер графа). Это будет продолжаться до тех пор, пока мы не дойдем до специальной конечной вершины, не имеющей ни одного выходящего ребра. Это и будет означать, что регулярное выражение успешно сопоставлено с некоторым набором словоформ.

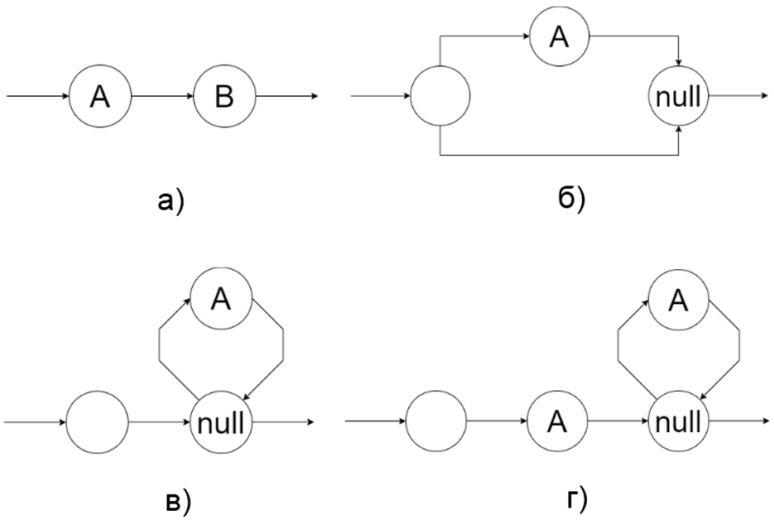

Рис. 1. Преобразование выражений во фрагменты недетерминированного конечного автомата: а) «AB» (неявная конкатенация), б) «A?» (квантификатор «0 или 1»), в) «A*» (квантификатор «0 или ∞»), г) «A+» (квантификатор «1 или ∞»)

Основные преобразования регулярного выражения в граф подразумевают следующие блоки, показанные на рисунке 1: неявная конкатенация, квантификатор «0 или 1», квантификатор «0 или ∞» и квантификатор «1 или ∞».

Таким образом, мы получаем все основные строительные блоки, включая блоки ветвлений и повторений для создания недетерминированного конечного автомата. На основе перечисленных выше блоков можно создать недетерминированный конечный автомат для выражения любой сложности.

Рассмотрим простые примеры для поиска одной словоформы заданного типа, а также примеры поиска меронимов и холонимов в текстах с помощью разработанной системы регулярных выражений. В наших выражениях мы можем использовать буквальный блок, смысловой блок, свободный блок, блок с описанием характеристик объекта (таблица 1).

Таблица 1. Примеры выражений и их значений

Выражение	Значение выражения
«<экономика>»	Буквальный блок. Соответствует всем возможным словоформам заданной лексемы.
«{замок_2}»	Смысловой блок. Соответствует любой словоформе лексемы, имеющей заданный смысл («замок», «крепость», «дворец» и так далее).
«.»	Свободный блок. Соответствует любой словоформе независимо от части речи и других характеристик.
«[s;вин;ед]»	Характеристики. Соответствует существительным в винительном падеже и в единственном числе.
«[a].*[s]»	Соответствует словосочетанию, состоящему из прилагательного и существительного, между которыми может быть неограниченное количество других словоформ.
«[s].*<включать>	
<в><себя>.*[s]'	Соответствует двум существительным (обозначаемым «[s]»), между которыми находятся лексемы «включать в себя». Пример недетерминированного конечного автомата для данного выражения представлен на рис. 2.
«[s].*<содержать>.*[s]»	Соответствует двум существительным, между которыми находится лексема «содержать».

А на рисунке 2 представлено преобразование сложного выражения, состоящего из нескольких блоков разного типа, в недетерминированный конечный автомат.

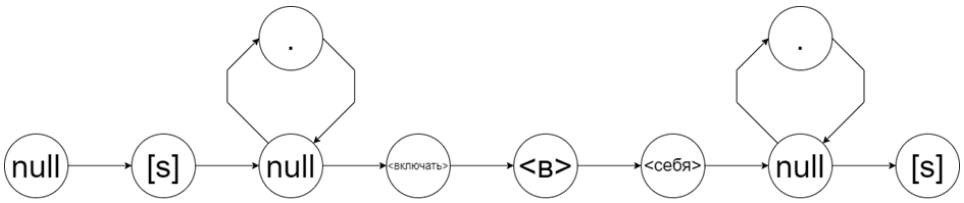

Рис. 2. Преобразование выражения «[s].*<включать><в><себя>.*[s]» в недетерминированный конечный автомат

Обработка шаблонов с использованием недетерминированных конечных автоматов позволяет использовать различные квантификаторы, так как при правильной реализации алгоритма увеличение количества элементов шаблона не будет приводить к значительному росту времени обработки.

Для того чтобы убедиться в корректности реализованного алгоритма, было предложено провести сравнение разработанного инструмента по времени выполнения поиска и команды «grep» (печать глобального регулярного выражения), доступной во многих операционных системах семейства Linux. Для сравнения были взяты выражения «a?^{n}a^{n}» и «a*^{n}a^{n}». Здесь «a» – это буквальный блок, или литерал в случае команды «grep», а «n» – количество повторений блока, или литерала. Например, при n = 1 первое выражение будет представлено в виде «a?a», при n = 3 выражение будет представлено в виде «a?a?a?aaa» и так далее. При увеличении количества повторений «n» от 1 до 50 можно проследить степень увеличения времени выполнения поиска по шаблону.

На рисунках 3 и 4 представлены графики, на которых видно, что время выполнения поиска с использованием разработанного алгоритма отличается от поиска с использованием команды «grep» незначительно. Таким образом, для разработанного алгоритма не проявляется экспоненциальный рост времени выполнения: его зависимость от длины шаблона линейна. Это наиболее заметно для выражения «$a^{*n}a^n$».

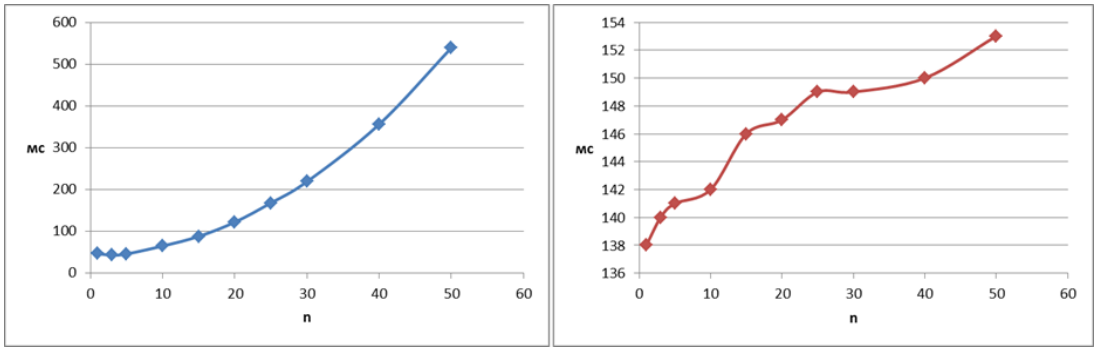

Рис. 3. График зависимости времени выполнения (в миллисекундах) поиска от количества элементов шаблона (n) для выражения «$a?^na^n$» (синий график слева – «grep», красный график справа – разработанный алгоритм)

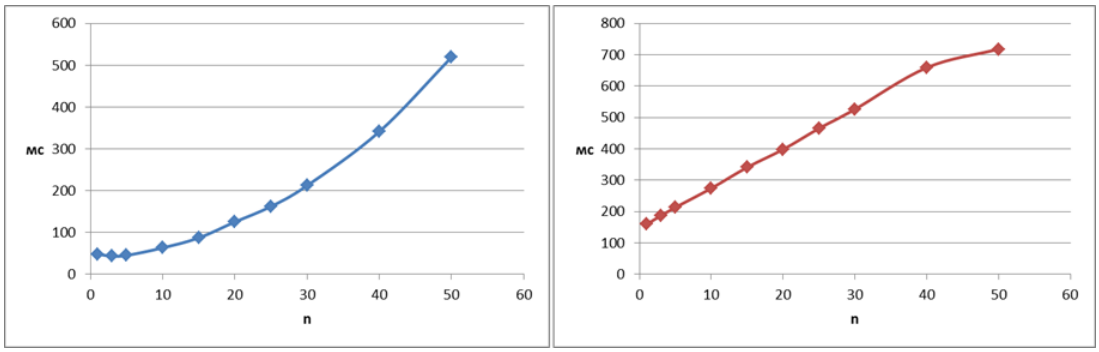

Рис. 4. График зависимости времени выполнения (в миллисекундах) поиска от количества элементов шаблона (n) для выражения «$a^{*n}a^n$» (синий график слева – «grep», красный график справа – разработанный алгоритм)

Заключение

Полученный результат подтвердил гипотезу о том, что грамматические признаки могут быть успешно использованы для извлечения требуемых словоформ и словосочетаний из структурированных текстовых документов.

В рамках проведенного исследования было разработано приложение для работы с шаблонами для поиска в тексте, но минимальной единицей обработки текста с помощью шаблонов стала словоформа, а не отдельный символ, как это сделано в случае с регулярными выражениями. Это позволило упростить работу с текстом для пользователей, знакомых со стандартными регулярными выражениями из различных языков программирования.

С помощью разработанных выражений удалось успешно найти заданные словосочетания, а также отдельные словоформы. Рассмотренные примеры показывают, насколько широким может быть применение механизма подобных выражений. Их можно использовать не только для обыкновенного поиска слов, но и для более сложных задач, таких как извлечение пар смысловых единиц, обладающих некоторой семантической связью. В частности, используя стандартные языковые конструкции, можно извлекать синонимы, антонимы, меронимы, холонимы, гипонимы и гиперонимы.

В дальнейшей научной работе планируется продолжение исследований по этой теме, а именно – внедрение постобработки полученных данных для извлечения семантических отношений.

Литература

1. Карташов О.О., Бутакова М.А., Чернов А.В., Костюков А.В., Жарков Ю.И. Средства представления знаний и извлечения данных для интеллектуального анализа ситуаций // Инженерный вестник Дона. 2018. №4. URL: ivdon.ru/ru/magazine/archive/n4y2018/5421
2. Клименков С.В., Николаев В.В., Харитонова А.Е., Гаврилов А.В., Письмак А.Е., Покид А.В. Применение семантической сети для хранения слабоструктурированных данных // Инженерный вестник Дона. 2020. №2. URL: ivdon.ru/ru/magazine/archive/N2y2020/6339

3. Письмак А.Е., Харитонова А.Е., Цопа Е.А., Клименков С.В. Метод автоматического формирования семантической сети из слабоструктурированных источников // Программные продукты и системы. 2016. №3. С. 74-78.
4. Aussenac-Gilles, N. and M. Kamel, 2009. Ontology Learning by Analyzing XML Document Structure and Content. KEOD, 9: 159-165.
5. Cederberg, S. and D. Widdows, 2003. Using LSA and Noun Coordination Information to Improve the Precision and Recall of Automatic Hyponymy Extraction. Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4, HLT-NAACL, pp: 111-118.
6. Cox R. Regular expression matching can be simple and fast (but is slow in java, perl, php, python, ruby,...) //URL: http://swtch. com/rsc/regexp/regexp1. html. – 2007.
7. Davulcu, H., S. Vadrevu and S. Nagarajan, 2003. Ontominer: Bootstrapping and populating ontologies from domain-specific web sites. IEEE Intelligent Systems, 18(5): 24-33.
8. Mikolov, T., K. Chen, G. Corrado and J. Dean, 2013. Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv:1301.3781, 3.
9. Pismak A. et al. Method of Semantic Refinement for Enterprise Search //KEOD. – 2020. – С. 307-312.
10. Thompson K. Programming techniques: Regular expression search algorithm //Communications of the ACM. – 1968. – Т. 11. – №. 6. – С. 419-422.
11. Winston, M.E., R. Chaffin and D.A. Herrmann, 1987. A taxonomy of part-whole relations. Cognitive science, 11(4): 417–444.

References in Cyrillics

1. Kartashov O.O., Butakova M.A., CHernov A.V., Kostyukov A.V., ZHarkov YU.I. Sredstva predstavleniya znanij i izvlecheniya dannyh dlya intellektual'nogo analiza situacij // Inzhenernyj vestnik Dona. 2018. №4. URL: ivdon.ru/ru/magazine/archive/n4y2018/5421
2. Klimenkov S.V., Nikolaev V.V., Haritonova A.E., Gavrilov A.V., Pis'mak A.E., Pokid A.V. Primenenie semanticheskoj seti dlya hraneniya slabostrukturirovannyh dannyh // Inzhener-nyj vestnik Dona. 2020. №2. URL: ivdon.ru/ru/magazine/archive/N2y2020/6339
3. Pismak A.E., Haritonova A.E., Copa E.A., Klimenkov S.V. Metod avtomaticheskogo formiro-vaniya semanticheskoj seti iz slabostrukturirovannyh istochnikov // Programmnye produkty i sistemy. 2016. №3. S. 74-78.
4. Aussenac-Gilles, N. and M. Kamel, 2009. Ontology Learning by Analyzing XML Document Structure and Content. KEOD, 9: 159-165.
5. Cederberg, S. and D. Widdows, 2003. Using LSA and Noun Coordination Information to Improve the Precision and Recall of Automatic Hyponymy Extraction. Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4, HLT-NAACL, pp: 111-118.
6. Cox R. Regular expression matching can be simple and fast (but is slow in java, perl, php, python, ruby,...) //URL: http://swtch. com/rsc/regexp/regexp1. html. – 2007.
7. Davulcu, H., S. Vadrevu and S. Nagarajan, 2003. Ontominer: Bootstrapping and populating ontologies from domain-specific web sites. IEEE Intelligent Systems, 18(5): 24-33.
8. Mikolov, T., K. Chen, G. Corrado and J. Dean, 2013. Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv:1301.3781, 3.
9. Pismak A. et al. Method of Semantic Refinement for Enterprise Search //KEOD. – 2020. – С. 307-312.
10. Thompson K. Programming techniques: Regular expression search algorithm //Communications of the ACM. – 1968. – Т. 11. – №. 6. – С. 419-422.
11. Winston, M.E., R. Chaffin and D.A. Herrmann, 1987. A taxonomy of part-whole relations. Cognitive science, 11(4): 417-444.

Ключевые слова
Семантическая сеть, онтология, семантическая связь, мероним, холоним, регулярные выражения

Ночевной Дмитрий Сергеевич,
аспирант факультета программной инженерии и компьютерной техники НИУ ИТМО,
182506@niuitmo.ru

Бессмертный Игорь Александрович,
профессор, доктор технических наук,
профессор факультета программной инженерии и компьютерной техники НИУ ИТМО,
bessmertny@itmo.ru

Клименков Сергей Викторович,
старший преподаватель факультета программной инженерии и компьютерной техники
НИУ ИТМО,
serge.klimenkov@cs.ifmo.ru

Keywords
Semantic network, ontology, semantic relation, meronym, holonym, regular expressions

DOI: DE-2023-03-05
JELclassification – C81 Методология сбора, оценки и организации микроэкономических данных, анализ данных; C82 Методология сбора, оценки и организации макроэкономических данных, анализ данных; C87 Эконометрическое программное обеспечение; D83 Поиск, обучение, информация и знания, взаимодействие, мнение, неосведомленность

Abstract
This article is devoted to the specially designed regular expressions for extracting word forms, as well as semantic relations obtained from structured and semi-structured sources, it contains description of the main elements of the semantic network (concepts, lexemes, word forms, relations and attributes), as well as the main types of relations between elements. The novelty of the research is the applicability of regular expressions not to symbols, but to lexemes. A classification of methods for automated extraction of semantic relations from text is given. Comparison of the performance of the developed algorithm and the utility “grep” is presented in terms of the number of quantifiers included in the search patterns.