УДК 004.853 DOI: 10.14529/cmse170205

КОМПЛЕКС ПРОГРАММ АВТОМАТИЧЕСКОГО ПОСТРОЕНИЯ СЕМАНТИЧЕСКОЙ СЕТИ СЛОВ

© 2017 г. Д.А. Усталов[1,2], А.В. Созыкин[1,2]

[1] *Институт математики и механики им. Н.Н.Красовского Уральского отделения Российской академии наук (620990 Екатеринбург, ул. Софьи Ковалевской, д. 16),*

[2] *Уральский федеральный университет имени первого Президента России Б.Н. Ельцина (620002 Екатеринбург, ул. Мира, д. 19)*

E-mail: dau@imm.uran.ru

Поступила в редакцию: 01.05.2017

Семантическая сеть слов — это ориентированный граф, вершины которого — лексические значения слов, а ребра — отношения между ними. В статье представлен комплекс программ SWN, предназначенный для построения семантической сети слов в автоматическом режиме путем структурирования неразмеченных словарей синонимов и словарей родо-видовых отношений с использованием векторных представлений слов, полученных на основе обработки корпуса неструктурированных текстов на естественном языке. Комплекс программ включает в себя реализацию методов обнаружения групп синонимов и построения отношений между отдельными значениями слов, основанных на обучении без учителя, а также модуля расширения отношений, основанного на обучении с учителем. Приведена модель предметной области с использованием формализма VOWL. Архитектура комплекса программ представлена в формализме UML и включает модуль обнаружения понятий, модуль построения семантических отношений между значениями слов, модуль расширения семантических отношений, модуль преобразования результатов работы в форматы Семантической паутины, и модуль построения оценочного набора данных при помощи краудсорсинга. Представленный комплекс программ является программным обеспечением с открытым исходным кодом и доступнен для интеграции в различные системы интеллектуального анализа данных.

Ключевые слова: семантическая сеть, лексическая семантика, программная инженерия, свободное программное обеспечение, Семантическая паутина, VOWL, UML.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Усталов Д.А., Созыкин А.В. Комплекс программ автоматического построения семантической сети слов // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2017. Т. 6, № 2. С. 69–83. DOI: 10.14529/cmse170205.

Введение

Семантическая сеть — это ориентированный граф, вершины которого — понятия, а ребра — отношения между ними [1]. Такой способ представления знаний широко применяется в области искусственного интеллекта и обработки естественного языка, что подтверждается использованием таких семантических ресурсов, как WordNet для английского языка и РуТез для русского языка [2]. К сожалению, язык изменяется быстрее, чем обновляются подобные ресурсы: возникают новые слова или новые значения существующих слов. Это приводит к тому, что все больше внимания уделяется созданию методов автоматического построения семантических сетей на основе обработки и выравнивания доступных структурированных и неструктурированных языковых ресурсов. Задача автоматического построения онтологии (англ. *ontology learning*) предполагает как интеграцию готовых словарей, так и извлечение информации из неразмеченных

корпусов текстов и использование машинного перевода для обеспечения полноты данных [3]. Известным примером высококачественной семантической сети, построенной автоматическим образом и доступной более чем для двухсот различных естественных языков, является BabelNet [4].

В последние годы особенно заметна тенденция по разработке методов обучения без учителя для автоматического формирования понятий и связей между ними, см. обзоры в [1, 3–5]. Это вызвано двумя причинами: 1) популярностью и полнотой неструктурированных ресурсов, таких как Википедия и Викисловарь, построенных при помощи краудсорсинга, 2) существенным снижением стоимости высокопроизводительных вычислительных ресурсов, что позволяет разрабатывать методы машинного обучения с использованием больших объемов данных. В данной работе представлен комплекс программ, формирующий семантическую сеть путем связывания отдельных лексических значений слов в виде специальной структуры данных — семантической сети слов [6–8]. В отличие от традиционных подходов к связыванию отдельных понятий [2], данный подход не требует высококачественного словаря понятий для их связывания друг с другом [4].

Статья организована следующим образом. В разделе 1 содержится описание предметной области и использованного подхода к построению семантической сети слов — разновидности семантической сети. В разделе 2 представлена архитектура комплекса программ SWN (сокр. англ. *semantic word network* — семантическая сеть слов) и функциональные особенности входящих в него программ. В заключении обсуждаются полученные результаты и рассматриваются направления дальнейших исследований.

1. Автоматическое построение семантической сети слов

Под семантической сетью слов мы будем понимать такую семантическую сеть, вершинами которой являются не понятия, т.е. множества синонимов [2], также известные как *синсеты*, а отдельные лексические значения слов, составляющие эти понятия.

Определение 1. Семантическая сеть слов — это семантическая сеть, вершины которой — лексические значения слов, а ребра — отношения между ними.

При обозначении слова и идентификатора его отдельного значения используется нотация, подобная принятой в BabelNet [4], но без указания части речи в нижнем регистре: запись $лук^1$ и $лук^2$ обозначает два различных значения одной и той же лексемы «лук». Например, слово «лук» имеет не меньше двух значений: $лук^1$ — метательное оружие, $лук^2$ — растение, и т. д. В целях избежания неоднозначности, в тексте статьи сноски не используются. Основное внимание в данной работе посвящено построению семантической сети на основе лексических значений слов, причем рассматривается единственный класс семантических онтошений — родо-видовые отношения, т. е. отношение между менее общим словом (гипонимом) и более общим словом (гиперонимом) [2]. Например, упорядоченная пара слов (*котенок*, *млекопитающее*) является корректной родо-видовой парой слов.

Общая схема метода построения семантической слов представлена на рис. 1. Данные методы ориентированы на использование таких ресурсов, как словарь синонимов и неразмеченная коллекция документов (корпус текстов). Источниками данных для такого метода являются материалы Викисловаря как словаря отношений между словами и неструктурированные тексты электронной библиотеки lib.rus.ec как корпуса текстов, содержащего 13 млрд словоупотреблений. На первом этапе производится выделение

значений слов и объединение их в группы близких слов при помощи кластеризации графа синонимов [6], после чего осуществляется связывание — формирование семантических отношений между лексическими значениями слов [7]. Кроме того, на этапе связывания производится расширение материалов существующих словарей [8], но эта операция не является обязательной.

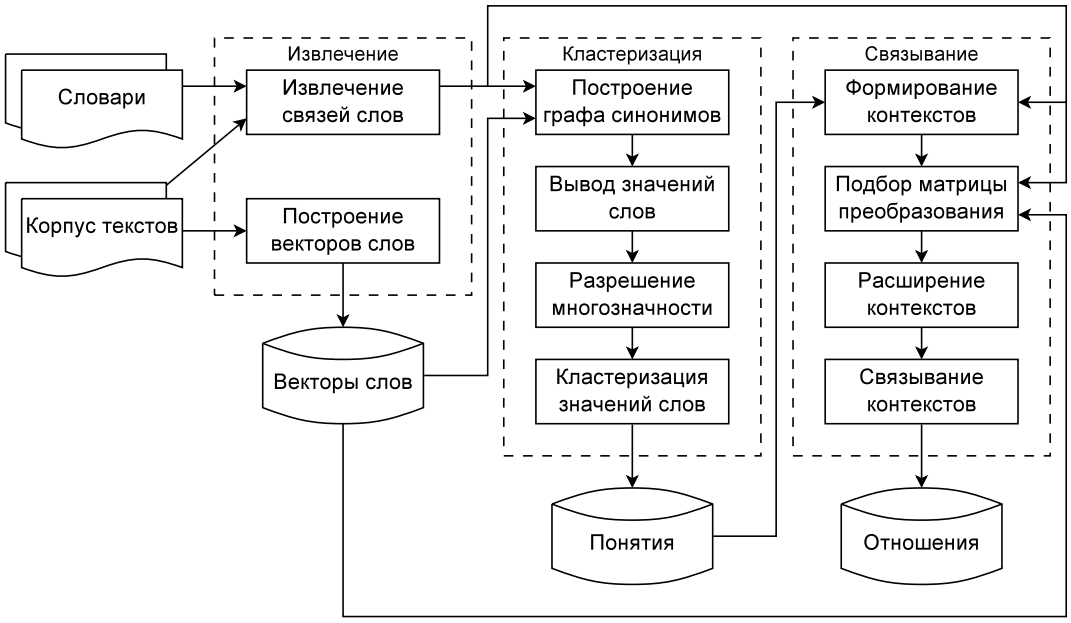

Рис. 1. Общая схема метода построения семантической сети слов

Для обеспечения возможности широкого применения семантической сети слов, используется запись в формализме RDF (англ. *Resource Description Framework*). RDF использует представление данных в виде троек «субъект–предикат–объект» [9]. На рис. 2 изображена диаграмма классов полученной семантической сети слов с точки зрения формализма VOWL [10], использующая модели SKOS [11] и Lemon [12] для записи лексико-семантической информации.

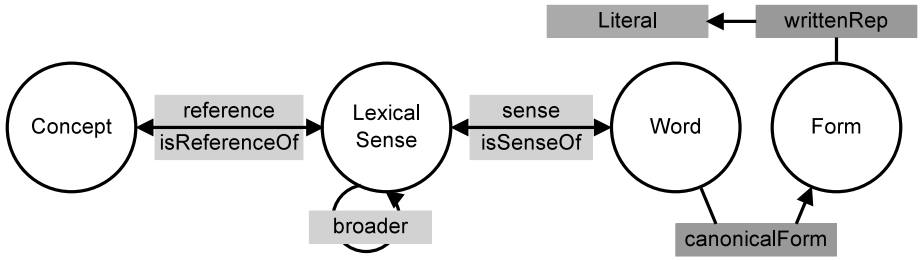

Рис. 2. VOWL-диаграмма информационной модели предметной области

2. Комплекс программ построения семантической сети слов

UML-диаграмма пакетов комплекса программ SWN представлена на рис. 3. Модуль обнаружения понятий Watset (от англ. *what* — «что?» и *set* — множество) реализует метод обнаружения понятий [6]. Модуль связывания Watlink (от англ. *what* — «что?» и *link* — связь) реализует метод связывания значений слов [7]. Модуль подбора матрицы линейного преобразования Hyperstar (от англ. *hyper* — «гипер» и *star* — «любой») реализует метод подбора матрицы линейного преобразования, используемый для расширения словарей [8].

Модуль экспорта данных (SWNRDF) реализует преобразование семантической сети слов в стандартный формат RDF [13] при помощи программы *Converter*.

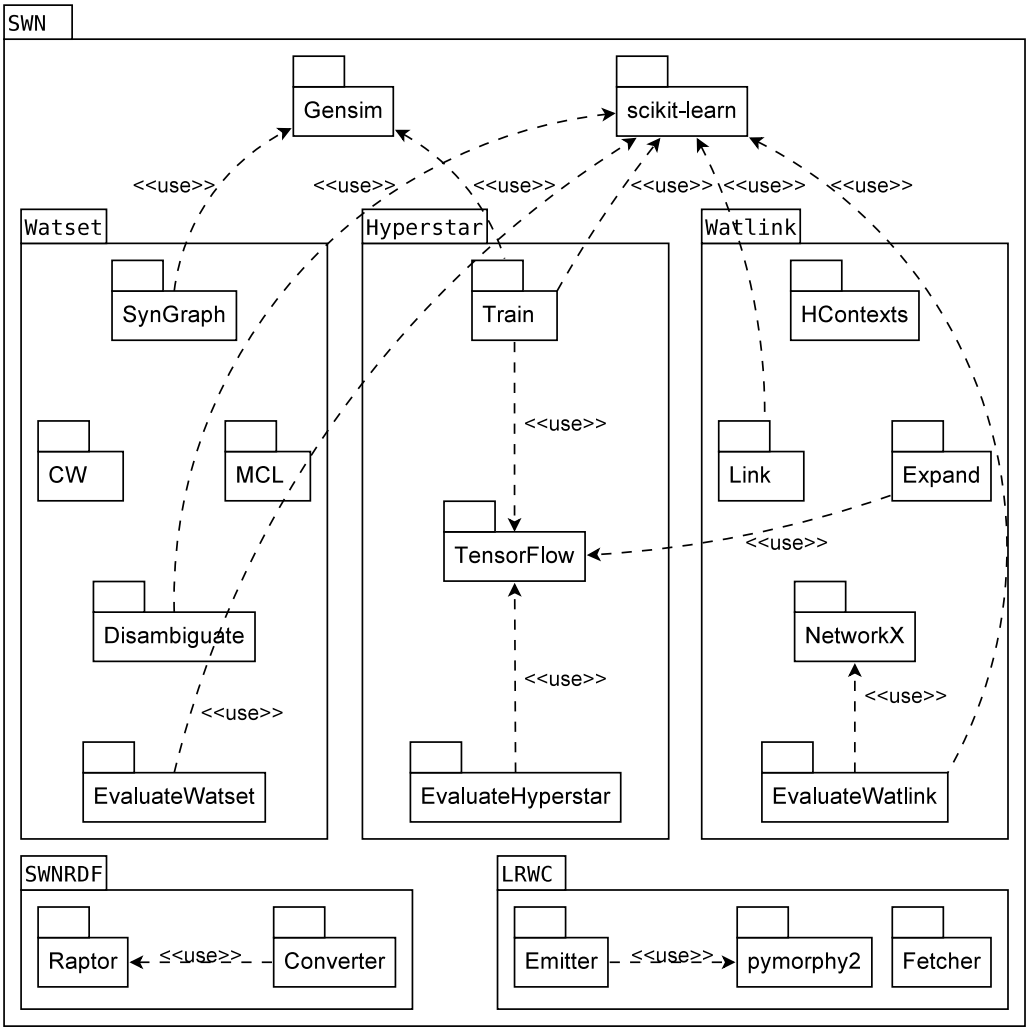

Рис. 3. UML-диаграмма пакетов комплекса программ

При реализации комплекса программ SWN использованы языки программирования Python, AWK, Java и Bash. Применяются внешние библиотеки, в том числе библиотека алгоритмов машинного обучения, подготовки и обработки данных scikit-learn [14], реализация алгоритма кластеризации Chinese Whispers [15] (CW), реализация марковского алгоритма кластеризации [16] (MCL), библиотека тематического моделирования и работы с векторами слов Gensim [17], библиотека методов оптимизации TensorFlow [18], библиотека работы с графами NetworkX [19], а также средства обработки RDF-троек Raptor [20] и морфологический анализатор pymorphy2 [21].

2.1. Модуль Watset

Модуль Watset реализует одноименный метод обнаружения понятий на основе графа синонимов [6]. На рис. 4 представлена UML-диаграмма активности обнаружения понятий, состоящая из трех шагов [6]: подготовка данных (программа *SynGraph*), обнаружение понятий (программы *CW*, *MCL* и *Disambiguate*), тестирование (программа *EvaluateWatset*).

Сначала производится загрузка исходных словарей и извлечение из них множества пар синонимов. При необходимости, вычисляется значение семантической близости

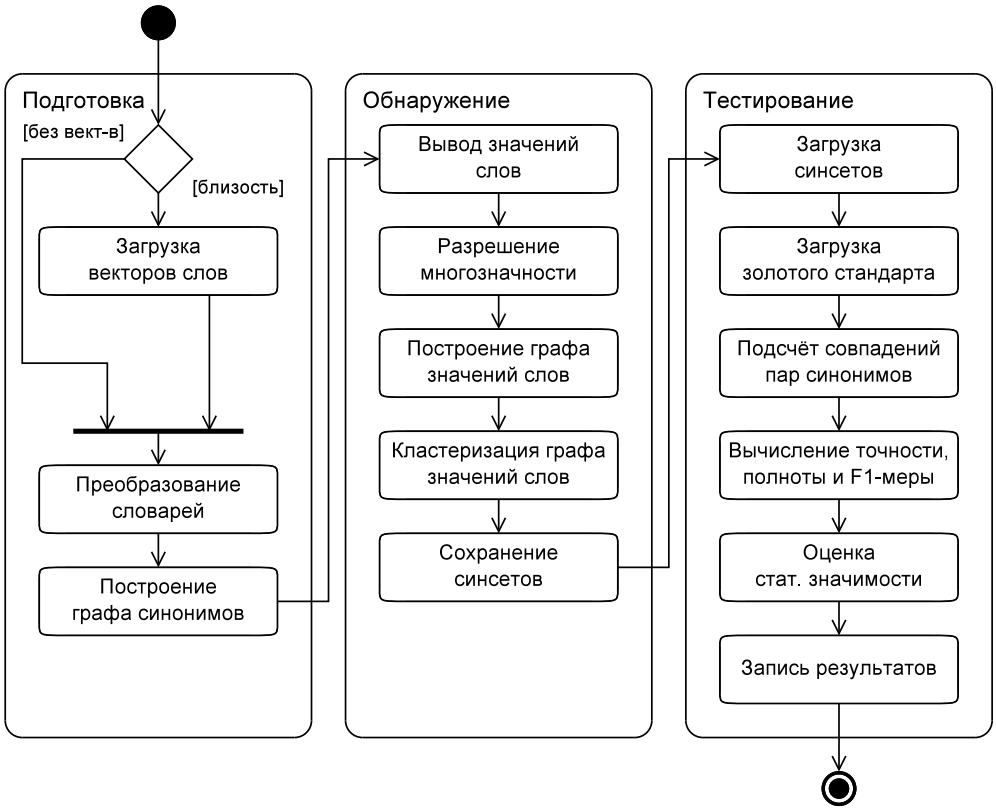

Рис. 4. UML-диаграмма активности обнаружения понятий

между парами синонимов на основе косинусной меры близости между векторами слов. Эти сведения используются при построении графа синонимов. При отсутствии сведений о семантической близости слов предусмотрено два альтернативных варианта: использование единичных весов для каждого ребра графа синонимов или подсчет количества появлений пары синонимов в исходных словарях. На этапе вывода значений слов допускается использование двух различных алгоритмов кластеризации эго-сетей: Chinese Whispers [15] или MCL [16]. На этапе разрешения многозначности производится разрешение многозначности в контекстах, причем в целях повышения производительности используется традиционный прием параллелизма по данным: каждое слово обрабатывается независимо в отдельном процессе. Определение номера значения слова в контексте производится путем максимизации косинусной меры близости [6]:

$$\hat{u} = \underset{u' \in \operatorname{senses}(u)}{\arg\max} \cos(\operatorname{ctx}(s), \operatorname{ctx}(u')), \tag{1}$$

где s — значение некоторого слова с известным номером значения, u — слово с неизвестным номером значения, $\operatorname{senses}(u)$ — множество значений слова u, $\operatorname{ctx}(\cdot)$ — контекст, т. е. множество синонимов слова в указанном значении. В результате разрешения многозначности формируется граф значений слов, кластеризация которого для получения синсетов производится методом Chinese Whispers или MCL. Синсеты получают уникальные номера и записываются в текстовый файл. Это необходимо как для использования данных в других задачах, так и для оценки качества. При оценке качества загружаются полученные синсеты и синсеты золотого стандарта. Затем каждый синсет из n слов преобразуется во множество из $\frac{n(n-1)}{2}$ пар синонимов и производится подсчет совпадений пар синонимов в полученном ресурсе и золотом стандарте. Вычисляются значения

информационно-поисковых критериев точности, полноты и F_1-меры [22] и оценивается статистическая значимость значения каждого критерия [23]. После выполнения всех указанных процедур осуществляется запись результатов оценки в текстовый файл.

2.2. Модуль Hyperstar

Модуль Hyperstar осуществляет подбор матрицы линейного преобразования векторных представлений гипонимов в векторные представления гиперонимов на основе модифицированного подхода, первоначально предложенного в [24]. На рис. 5 представлена UML-диаграмма активности подбора матрицы линейного преобразования, состоящая из трех условных шагов: подготовка данных и обучение модели (программа *Train*), тестирование (программа *EvaluateHyperstar*). Исходными данными для подбора матрицы являются векторы слов и примеры родо-видовых отношений между словами, полученные из словарей. В процессе используются только те пары слов, для которых имеются векторы. Это вызвано тем, что векторы слов строятся на основании большого корпуса текстов с различными подходами к предварительной обработке, например, фильтрации низкочастотных слов. Полученные пары векторов слов разбиваются на три различные выборки в соотношении: 60 % данных составляют обучающую выборку для подбора параметров, 20 % данных составляют валидационной выборки для подбора метапараметров, и оставшиеся 20 % составляют тестовую выборку для оценки качества модели.

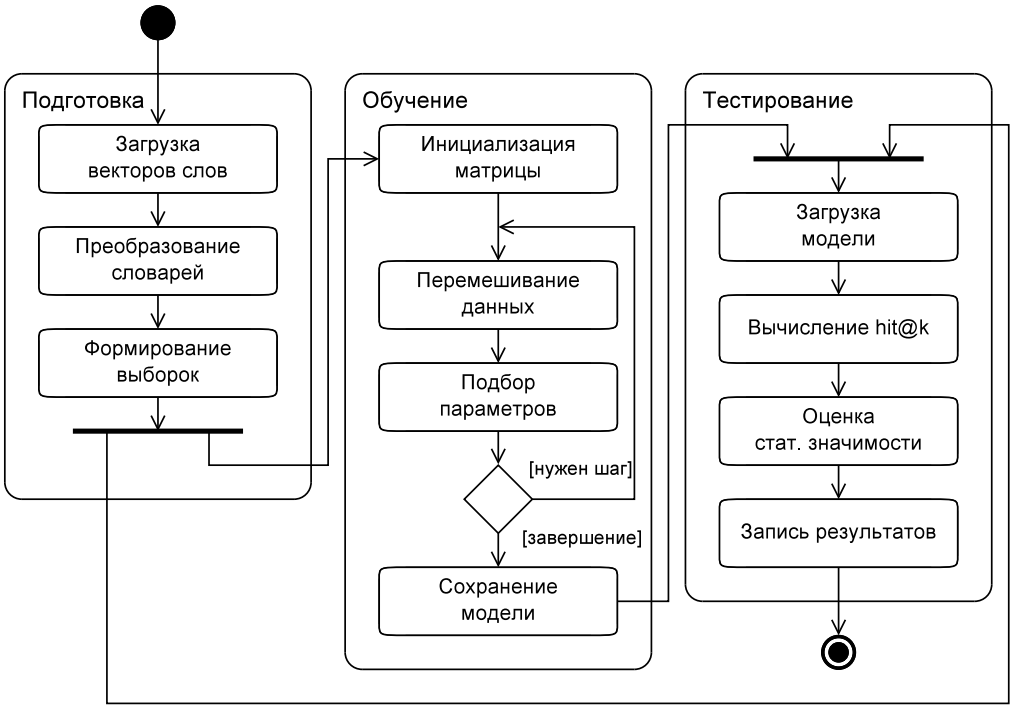

Рис. 5. UML-диаграмма активности подбора матрицы линейного преобразования

В начале процесса обучения все элементы матрицы генерируются как независимые между собой случайные величины, имеющие стандартное нормальное распределение с параметрами $\mu = 0$ и $\sigma = 0{,}1$; допущения о свойствах матрицы не используются [24]. На каждом шаге обучения производится перемешивание данных и выполняется подбор значений элементов матрицы с целью минимизации следующей функции потерь с применением регуляризации на основе выборки отрицательных примеров [8] (при записи

используется нотация вектора-строки, т. е. вектор $\vec{v}$ является матрицей размера $1 \times |\vec{v}|$):

$$\Phi^* = \arg\min_{\Phi} \frac{1}{|\mathcal{P}|} \sum_{(\vec{x},\vec{y})\in\mathcal{P}} \|\vec{x}\Phi - \vec{y}\|^2 + \lambda R,$$
$$R_h = \frac{1}{|\mathcal{P}|} \sum_{(\vec{x},\_)\in\mathcal{P}} (\vec{x}\Phi\Phi \cdot \vec{x})^2, \qquad R_s = \frac{1}{|\mathcal{N}|} \sum_{(\vec{x},\vec{z})\in\mathcal{N}} (\vec{x}\Phi\Phi \cdot \vec{z})^2, \tag{2}$$

где Φ — матрица линейного преобразования, $\vec{x}$ — вектор гипонима, $\vec{y}$ — вектор гиперонима, $\vec{z}$ — вектор синонима $\vec{x}$, $\mathcal{P}$ — обучающая выборка с положительными примерами, $\mathcal{N}$ — обучающая выборка с отрицательными примерами, λ — параметр важности члена регуляризации $R \in \{R_h, R_s\}$, R_h — регуляризатор, накладывающий ограничения на близость преобразования $\vec{x}\Phi\Phi$ к исходному гипониму $\vec{x}$, R_s — регуляризатор, накладывающий ограничения на близость преобразования $\vec{x}\Phi\Phi$ к синониму $\vec{z}$ исходного гипонима $\vec{x}$. Процесс обучения завершается по достижении указанного при запуске количества шагов; двоичное представление полученной матрицы записывается в файл. Оценка качества предполагает загрузку полученных матриц и вычисление значения критерия hit@k по валидационной выборке для подбора параметров или по тестовой выборке для оценки качества работы метода. Оценивается статистическая значимость значения данного критерия [8]. После выполнения всех указанных процедур осуществляется запись результатов оценки в текстовый файл.

2.3. Модуль Watlink

Модуль Watlink реализует одноименный метод построения однозначных семантических отношений [7] и осуществляет построение семантической сети слов на основе родо-видовых пар слов и ранее полученных синсетов. На рис. 6 представлена UML-диаграмма активности построения отношений, состоящая из трех условных шагов: подготовка данных (программа *HContexts*), связывание (программы *Link* и, опционально, *Expand*), тестирование (программа *EvaluateWatlink*).

Исходными данными для построения отношений являются синсеты и сведения о требуемых семантических отношениях. Извлечение семантических отношений производится из словарей, содержащих перечисленные родо-видовые пары в текстовом виде: $\mathrm{hctx}(S) = \{h : w \in \mathrm{words}(S), (w,h) \in \mathcal{R}\}$, где $\mathrm{hctx}(S)$ — иерархический контекст синсета S, $\mathrm{words}(S)$ — слова, входящие в синсет S, $\mathcal{R}$ — множество родо-видовых отношений. На основе этих сведений формируются иерархические контексты каждого синсета. При необходимости загружаются векторы слов, матрица линейного преобразования, и осуществляется расширение иерархических контекстов с использованием ранее полученной матрицы линейного преобразования [7, 8]:

$$\mathrm{hctx}'(S) = \cup_{\substack{w\in\mathrm{words}(S),\\(w,h)\in\mathcal{R}}} \{w\} \times \mathrm{NN}_n(\vec{h}) \cup \mathrm{hctx}(S), \tag{3}$$

где $\mathrm{NN}_n(\vec{h})$ — операция поиска n ближайших соседей векторного представления слова h.

Разрешение многозначности в иерархических контекстах реализовано с использованием трех различных подходов к взвешиванию каждого гиперонима в иерархическом контексте: tf, idf и tf–idf [22], причем под «термином» понимается гипероним, а под «документом» понимается синсет. В целях повышения производительности, используется традиционный прием параллелизма по данным: каждый синсет обрабатывается независимо в отдельном

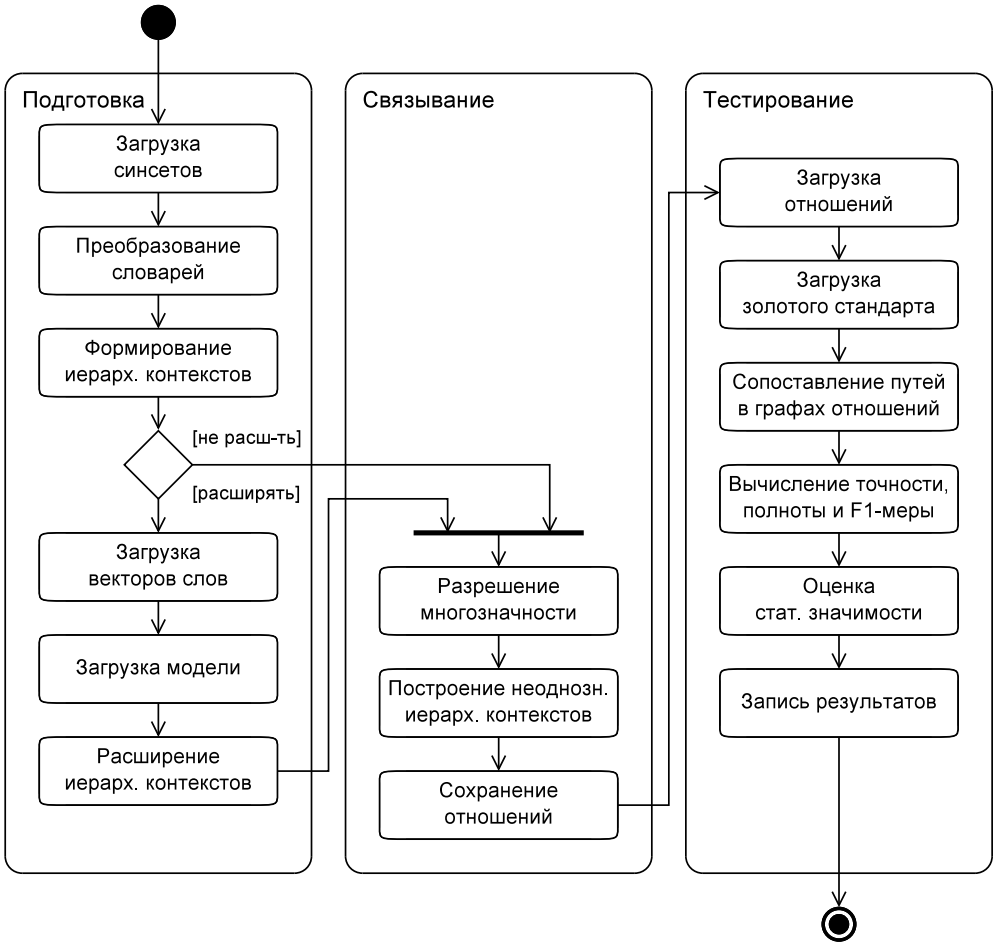

Рис. 6. UML-диаграмма активности построения отношений

процессе. Для каждого слова h в иерархическом контексте $\mathrm{hctx}(S)$ синсета S определяется номер значения путем максимизации косинусной меры близости [7]:

$$\hat{h} = \underset{\substack{S' \in \mathcal{S}, S \neq S', h' \in S', \\ \mathrm{words}(\{h'\}) = \{h\}}}{\arg\max} \cos(\mathrm{hctx}(S), S'). \tag{4}$$

При построении иерархических контекстов со снятой неоднозначностью используется только n гиперонимов, получившие максимальный вес по итогам выполнения этапа разрешения многозначности. Семантическая сеть слов сохраняется в текстовый файл. При оценке качества загружаются полученные отношения и отношения между словами золотого стандарта в виде ориентированных графов. Затем, для каждого отношения определяется существование пути от гипонима к гипериниму в графе золотого стандарта. Вычисляются значения информационно-поисковых критериев точности, полноты и F_1-меры [22] и оценивается статистическая значимость значения каждого критерия [23]. После выполнения всех указанных процедур осуществляется запись результатов оценки в текстовый файл.

2.4. Модуль LRWC

На рис. 7 представлена UML-диаграмма активности построения оценочного набора данных при помощи краудсорсинга на основе выполнения микрозадач, именуемая LRWC (сокр. англ. *Lexical Relations from the Wisdom of the Crowd* — «мудрость толпы о семантических отношениях»). Активность состоит из двух условных шагов [7]: подготовка

заданий и их размещение на платформе краудсорсинга (программа *Emitter*) и получение суждений участников процесса краудсорсинга о корректности семантических отношений (программа *Fetcher*).

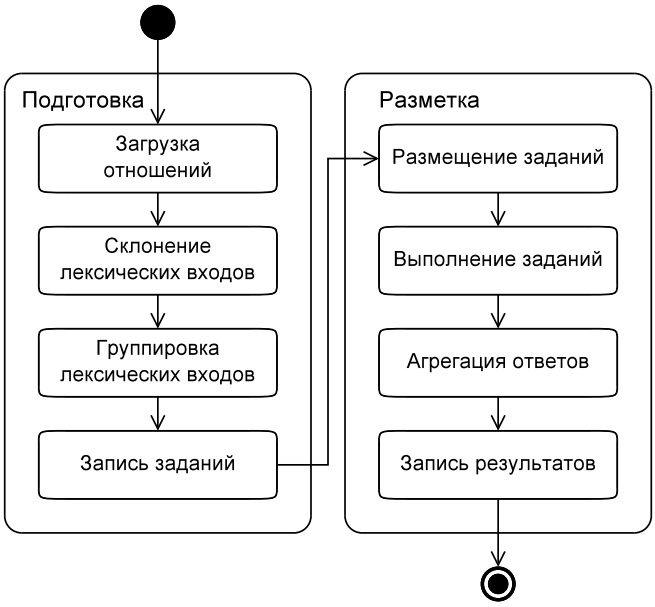

Рис. 7. UML-диаграмма активности построения оценочного набора данных при помощи краудсорсинга на основе выполнения микрозадач

Оценочный набор данных составляется для семантических отношений между значениями слов, т. е. для множества упорядоченных пар вида (*котенок*[1], *млекопитающее*[1]). От участника процесса краудсорсинга требуется предоставить положительный или отрицательный ответ на вопрос вида «Правда ли *котенок* — это разновидность *млекопитающего*?». Для этого вышестоящее слово приводится в форму родительного падежа при помоши морфологического анализатора. Каждая пара слов оценивается независимо несколькими разными участниками, после чего производится агрегация ответов всех участников на все пары слов и сохранение оценочного набора данных в файл.

Заключение

В работе представлен комплекс программ SWN, осуществляющий построение семантической сети слов и ее запись в форматах Семантической паутины на основе информационной модели (рис. 2). Описана архитектура комплекса программ, включающего в себя программы обнаружения понятий, построения и расширения семантических отношений, построения оценочного набора данных. Программы написаны с использованием параллелизма по данным, что позволяет использовать вычислительные узлы с большим количеством доступных ядер центрального процессора для ускорения вычислений. Кроме того, при реализации методов использованы высокоэффективные внешние библиотеки, такие как scikit-learn [14] и TensorFlow [18]. В настоящее время все программы функционируют в режиме командной строки. Связывание программ, написанных на разных языках программирования, осуществляется путем перенаправления потоков стандартного ввода и вывода в сценариях командного процессора Bash и утилиты make. Входные данные представлены в виде текстовых файлов, поля в которых разделены

знаком табуляции. Все промежуточные и итоговые результаты, кроме матрицы линейного преобразования, также представлены в текстовом виде. Кроме того, итоговый результат записывается в формате N-Triples [20].

Эксперименты по построению семантической сети слов для русского языка при помощи данного комплекса программ подтверждают его эффективность по сравнению с аналогичными решениями [6–8]. Важной особенностью методов, лежащих в основе представленного комплекса программ, является независимость от высококачественного исходного семантического ресурса для построения и связывания понятий. Исходный код разработанных программ доступен на GitHub [25–27] вместе с инструкциями по использованию. Среди возможных направлений развития комплекса программ SWN отмечается его интеграция в системы интеллектуального анализа данных, разработка графического интерфейса пользователя, а также использование аппаратных ускорителей вычислений для увеличения производительности программ кластеризации графа и связывания значений слов. В настоящее время комплекс программ предназначен для работы только под операционной системой Linux; поддержка других операционных систем, в т. ч. Windows, не предусмотрена.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 16-37-00354 мол_а и при финансовой поддержке РГНФ в рамках научного проекта № 16-04-12019 «Интеграция тезаурусов RussNet и YARN». Авторы благодарят компанию Microsoft Research за предоставленные вычислительные ресурсы в облачной среде Microsoft Azure в рамках программы Azure for Research.

Литература

1. Gonçalo Oliveira H., Gomes P. ECO and Onto.PT: a Flexible Approach for Creating a Portuguese Wordnet Automatically. Language Resources and Evaluation. 2014. Vol. 48, No. 2. P. 373–393. DOI: 10.1007/s10579-013-9249-9.

2. Лукашевич Н.В. Тезаурусы в задачах информационного поиска. М.: Изд-во МГУ, 2011. 512 с.

3. Wong W. et al. Ontology Learning from Text: A Look Back and into the Future. ACM Computing Surveys. 2012. Vol. 44, No. 4. P. 20:1–20:36. DOI: 10.1145/2333112.2333115.

4. Navigli R., Ponzetto S.P. BabelNet: The Automatic Construction, Evaluation and Application of a Wide-Coverage Multilingual Semantic Network. Artificial Intelligence. Vol. 193. P. 217–250. DOI: 10.1016/j.artint.2012.07.001.

5. Camancho Collados J., Pilehvar M.T., Navigli R. Nasari: Integrating Explicit Knowledge and Corpus Statistics for a Multilingual Representation of Concepts and Entities. Artificial Intelligence. Vol. 240. P. 36–64. DOI: 10.1016/j.artint.2016.07.005.

6. Усталов Д.А. Обнаружение понятий в графе синонимов // Вычислительные технологии. 2017. Т. 22, Специальный выпуск 1. С. 99–112. URL: `http://depot.nlpub.ru/ustalov.jct2017.pdf` (дата обращения: 25.04.2017).

7. Усталов Д.А. Построение семантической сети слов путем расширения иерархических контекстов // Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной Международной конференции «Диалог» (Москва, 31 мая — 3

июня 2017 г.). М.: Изд-во РГГУ, 2017. В печати. URL: http://depot.nlpub.ru/ustalov.dialog2017.pdf (дата обращения: 06.05.2017).

8. Ustalov D.A., Arefyev N.V., Biemann C., Panchenko A.I. // Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. Association for Computational Linguistics, 2017, P. 543–550. URL: https://aclweb.org/anthology/E/E17/E17-2087.pdf (дата обращения: 10.04.2017).
9. Berners-Lee T., Hendler J., Lassila O. The Semantic Web. Scientific American. 2001. Vol. 284, No. 5. P. 28–37. URL: https://www.scientificamerican.com/article/the-semantic-web/ (дата обращения: 10.03.2017).
10. Lohmann S. et al. Visualizing Ontologies with VOWL. Semantic Web. 2016. Vol. 7, No. 4. P. 399–419. DOI: 10.3233/SW-150200.
11. van Assem M. et al. A Method to Convert Thesauri to SKOS // 3rd European Semantic Web Conference, ESWC 2006 Budva, Montenegro, June 11–14, 2006 Proceedings. Springer Berlin Heidelberg, 2006. P. 95–109. DOI: 10.1007/11762256_10.
12. McCrae J., Spohr D., Cimiano P. Linking Lexical Resources and Ontologies on the Semantic Web with Lemon // The Semantic Web: Research and Applications: 8th Extended Semantic Web Conference, ESWC 2011, Heraklion, Crete, Greece, May 29–June 2, 2011, Proceedings, Part I. Springer Berlin Heidelberg, 2011. P. 245–259. DOI: 10.1007/978-3-642-21034-1_17.
13. Усталов Д.А. Тезаурусы русского языка в виде открытых связанных данных // Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной Международной конференции «Диалог» (Москва, 27 — 30 мая 2015 г.). М.: Изд-во РГГУ, 2015. С. 616–625. URL: http://www.dialog-21.ru/digests/dialog2015/materials/pdf/UstalovDA.pdf (дата обращения: 21.02.2017).
14. Pedregosa F. et al. Scikit-Learn: Machine Learning in Python // Journal of Machine Learning Research. 2011. Vol. 12. P. 2825–2830. URL: http://www.jmlr.org/papers/v12/pedregosa11a.html (дата обращения: 07.03.2017).
15. Biemann C. Chinese Whispers: An Efficient Graph Clustering Algorithm and Its Application to Natural Language Processing Problems // Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing. Association for Computational Linguistics, 2006. P. 73–80. URL: http://dl.acm.org/citation.cfm?id=1654774 (дата обращения: 15.03.2017).
16. van Dongen S. Graph Clustering by Flow Simulation. Ph.D. Thesis. University of Utrecht, 2000. URL: https://dspace.library.uu.nl/handle/1874/848 (дата обращения: 27.03.2017).
17. Řehuřek R., Sojka P. Software Framework for Topic Modelling with Large Corpora // New Challenges for NLP Frameworks Programme: A workshop at LREC 2010. European Language Resources Association, 2010. P. 51–55. URL: https://radimrehurek.com/gensim/lrec2010_final.pdf (дата обращения: 03.04.2017).
18. Abadi M. et al. TensorFlow: A System for Large-Scale Machine Learning // 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16). USENIX Association, 2016. P. 265–283. URL: https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi (дата обращения: 10.04.2017).

19. Hagberg A.A., Schult D.A., Swart P.J. Exploring Network Structure, Dynamics, and Function using NetworkX // Proceedings of the 7th Python in Science Conference. 2008. P. 11–15. URL: http://conference.scipy.org/proceedings/scipy2008/paper_2/ (дата обращения: 05.12.2016).

20. Beckett D. The Design and Implementation of the Redland RDF Application Framework. Computer Networks. 2002. Vol. 39, No. 5. P. 577–588. DOI: 10.1016/S1389-1286(02)00221-9.

21. Korobov M. Morphological Analyzer and Generator for Russian and Ukrainian Languages. Analysis of Images, Social Networks and Texts: 4th International Conference, AIST 2015, Yekaterinburg, Russia, April 9–11, 2015, Revised Selected Papers. Springer International Publishing, 2015. P. 320–332. DOI: 10.1007/978-3-319-26123-2_31.

22. Manning C.D., Raghavan P., Schütze H. Introduction to Information Retrieval. Cambridge University Press, 2008. 506 p.

23. Riedl M., Biemann C. Unsupervised Compound Splitting With Distributional Semantics Rivals Supervised Methods // Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, 2016. P. 617–622. URL: https://aclweb.org/anthology/N/N16/N16-1075.pdf (дата обращения: 16.02.2017).

24. Fu R. et al. Learning Semantic Hierarchies via Word Embeddings. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, 2014. P. 1199–1209. URL: https://aclweb.org/anthology/P/P14/P14-1113.pdf (дата обращения: 26.04.2016).

25. dustalov/watset: Concept Discovery from Synonymy Graphs. URL: https://github.com/dustalov/watset (дата обращения: 10.04.2017).

26. dustalov/watlink: Concept Linking. URL: https://github.com/dustalov/watlink (дата обращения: 10.04.2017).

27. dustalov/projlearn: Learning Word Subsumption Projections. URL: https://github.com/dustalov/projlearn (дата обращения: 10.04.2017).

Усталов Дмитрий Алексеевич, м.н.с., Институт математики и механики им. Н.Н. Красовского УрО РАН; ассистент, кафедра высокопроизводительных компьютерных технологий, Уральский федеральный университет (Екатеринбург, Российская Федерация)

Созыкин Андрей Владимирович, к.т.н., Институт математики и механики им. Н.Н. Красовского УрО РАН; зав. каф., кафедра высокопроизводительных компьютерных технологий, Уральский федеральный университет (Екатеринбург, Российская Федерация)

DOI: 10.14529/cmse170205

A SOFTWARE SYSTEM FOR AUTOMATIC CONSTRUCTION OF A SEMANTIC WORD NETWORK

© 2017 D.A. Ustalov[1,2], A.V. Sozykin[1,2]

[1]*Krasovskii Institute of Mathematics and Mechanics (ul. Sofii Kovalevskoy 16, Yekaterinburg, 620990 Russia),*

[2]*Ural Federal University (ul. Mira 19, Yekaterinburg, 620002 Russia)*

E-mail: dau@imm.uran.ru

Received: 01.05.2017

A semantic word network is a network that represents the semantic relations between individual words or their lexical senses. In this paper, we present a software system for automatic construction of a semantic word network. The system, called SWN, is designed for the construction for such a semantic word network and includes the implementation of unsupervised concept discovery and semantic relation establishing methods as well as the implementation of a supervised relation expansion method. The methods use widely available language resources, such as semantic relation dictionaries and background text corpora. The domain model has been presented using the VOWL notation. The system architecture is represented using the UML notation and is composed of the concept discovery module, semantic relation construction module, the Semantic Web export module, and the evaluation dataset construction module based on microtask-based crowdsourcing. The present software system is open source and is available for integration into third-party data mining systems.

Keywords: semantic network, lexical semantics, software engineering, free software, Semantic Web, VOWL, UML.

FOR CITATION

Ustalov D.A., Sozykin A.V. A Software System for Automatic Construction of a Semantic Word Network. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering.* 2017. vol. 6, no. 2. pp. 69–83. (in Russian) DOI: 10.14529/cmse170205.

References

1. Gonçalo Oliveira H., Gomes P. ECO and Onto.PT: a Flexible Approach for Creating a Portuguese Wordnet Automatically. Language Resources and Evaluation. 2014. vol. 48, no. 2. pp. 373–393. DOI: 10.1007/s10579-013-9249-9.

2. Loukachevitch N.V. *Tezaurusy v zadachakh informatsionnogo poiska* [Thesauri in Information Retrieval Tasks]. Moscow, MSU Publishing, 2011. 512 p.

3. Wong W. et al. Ontology Learning from Text: A Look Back and into the Future. ACM Computing Surveys. 2012. vol. 44, no. 4. pp. 20:1–20:36. DOI: 10.1145/2333112.2333115.

4. Navigli R., Ponzetto S.P. BabelNet: The Automatic Construction, Evaluation and Application of a Wide-Coverage Multilingual Semantic Network. Artificial Intelligence. vol. 193. pp. 217–250. DOI: 10.1016/j.artint.2012.07.001.

5. Camancho Collados J., Pilehvar M.T., Navigli R. Nasari: Integrating Explicit Knowledge and Corpus Statistics for a Multilingual Representation of Concepts and Entities. Artificial Intelligence. vol. 240. pp. 36–64. DOI: 10.1016/j.artint.2016.07.005.

6. Ustalov D.A. Concept Discovery from Synonymy Graphs. *Vychislitel'nye tekhnologii* [Computational Technologies]. 2017. vol. 22, Special Issue 1. pp. 99–112. Available at: `http://depot.nlpub.ru/ustalov.jct2017.pdf` (accessed: 25.04.2017).

7. Ustalov D.A. Expanding Hierarchical Contexts for Constructing a Semantic Word Network. *Komp'yuternaya lingvistika i intellektual'nye tekhnologii: Po materialam ezhegodnoi Mezhdunarodnoi konferentsii «Dialog» (Moskva, 31 maya — 3 iyunya 2017 g.)* [Computational Linguistics and Intellectual Technologies: papers from the Annual conference "Dialogue" (Moscow, May 31–June 3, 2017)]. Moscow, RSUH, 2017. In press. Available at: `http://depot.nlpub.ru/ustalov.dialog2017.pdf` (accessed: 06.05.2017).

8. Ustalov D.A., Arefyev N.V., Biemann C., Panchenko A.I. Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. Association for Computational Linguistics, 2017, pp. 543–550. Available at: `https://aclweb.org/anthology/E/E17/E17-2087.pdf` (accessed: 10.04.2017).

9. Berners-Lee T., Hendler J., Lassila O. The Semantic Web. Scientific American. 2001. vol. 284, no. 5. pp. 28–37. Available at: `https://www.scientificamerican.com/article/the-semantic-web/` (accessed: 10.03.2017).

10. Lohmann S. et al. Visualizing Ontologies with VOWL. Semantic Web. 2016. vol. 7, no. 4. pp. 399–419. DOI: 10.3233/SW-150200.

11. van Assem M. et al. A Method to Convert Thesauri to SKOS. 3rd European Semantic Web Conference, ESWC 2006 Budva, Montenegro, June 11–14, 2006 Proceedings. Springer Berlin Heidelberg, 2006. pp. 95–109. DOI: 10.1007/11762256_10.

12. McCrae J., Spohr D., Cimiano P. Linking Lexical Resources and Ontologies on the Semantic Web with Lemon. The Semantic Web: Research and Applications: 8th Extended Semantic Web Conference, ESWC 2011, Heraklion, Crete, Greece, May 29–June 2, 2011, Proceedings, Part I. Springer Berlin Heidelberg, 2011. pp. 245–259. DOI: 10.1007/978-3-642-21034-1_17.

13. Ustalov D.A. Russian Thesauri as Linked Open Data. *Komp'yuternaya lingvistika i intellektual'nye tekhnologii: Po materialam ezhegodnoi Mezhdunarodnoi konferentsii «Dialog» (Moskva, 27 — 30 maya 2015 g.)* [Computational Linguistics and Intellectual Technologies: papers from the Annual conference "Dialogue" (Moscow, May 27–30, 2015)]. Moscow, RSUH, 2015, pp. 616–625. Available at: `http://www.dialog-21.ru/digests/dialog2015/materials/pdf/UstalovDA.pdf` (accessed: 21.02.2017).

14. Pedregosa F. et al. Scikit-Learn: Machine Learning in Python. Journal of Machine Learning Research. 2011. vol. 12. pp. 2825–2830. Available at: `http://www.jmlr.org/papers/v12/pedregosa11a.html` (accessed: 07.03.2017).

15. Biemann C. Chinese Whispers: An Efficient Graph Clustering Algorithm and Its Application to Natural Language Processing Problems. Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing. Association for Computational Linguistics, 2006. pp. 73–80. Available at: `http://dl.acm.org/citation.cfm?id=1654774` (accessed: 15.03.2017).

16. van Dongen S. Graph Clustering by Flow Simulation. Ph.D. Thesis. University of Utrecht, 2000. Available at: `https://dspace.library.uu.nl/handle/1874/848` (accessed: 27.03.2017).

17. Řehuřek R., Sojka P. Software Framework for Topic Modelling with Large Corpora. New Challenges for NLP Frameworks Programme: A workshop at LREC 2010. European Language Resources Association, 2010. pp. 51–55. Available at: `https://radimrehurek.com/gensim/lrec2010_final.pdf` (accessed: 03.04.2017).

18. Abadi M. et al. TensorFlow: A System for Large-Scale Machine Learning. 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16). USENIX Association, 2016. pp. 265–283. Available at: https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi (accessed: 10.04.2017).

19. Hagberg A.A., Schult D.A., Swart P.J. Exploring Network Structure, Dynamics, and Function using NetworkX. Proceedings of the 7th Python in Science Conference. 2008. pp. 11–15. Available at: http://conference.scipy.org/proceedings/scipy2008/paper_2/ (accessed: 05.12.2016).

20. Beckett D. The Design and Implementation of the Redland RDF Application Framework. Computer Networks. 2002. vol. 39, no. 5. pp. 577–588. DOI: 10.1016/S1389-1286(02)00221-9.

21. Korobov M. Morphological Analyzer and Generator for Russian and Ukrainian Languages. Analysis of Images, Social Networks and Texts: 4th International Conference, AIST 2015, Yekaterinburg, Russia, April 9–11, 2015, Revised Selected Papers. Springer International Publishing, 2015. pp. 320–332. DOI: 10.1007/978-3-319-26123-2_31.

22. Manning C.D., Raghavan P., Schütze H. Introduction to Information Retrieval. Cambridge University Press, 2008. 506 p.

23. Riedl M., Biemann C. Unsupervised Compound Splitting With Distributional Semantics Rivals Supervised Methods. Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, 2016. pp. 617–622. Available at: https://aclweb.org/anthology/N/N16/N16-1075.pdf (accessed: 16.02.2017).

24. Fu R. et al. Learning Semantic Hierarchies via Word Embeddings. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, 2014. pp. 1199–1209. Available at: https://aclweb.org/anthology/P/P14/P14-1113.pdf (accessed: 26.04.2016).

25. Ustalov D.A. dustalov/watset: Concept Discovery from Synonymy Graphs. Available at: https://github.com/dustalov/watset (accessed: 10.04.2017).

26. Ustalov D.A. dustalov/watlink: Concept Linking. Available at: https://github.com/dustalov/watlink (accessed: 10.04.2017).

27. Ustalov D.A. dustalov/projlearn: Learning Word Subsumption Projections. Available at: https://github.com/dustalov/projlearn (accessed: 10.04.2017).