УДК 681.3

А.В. АНИСИМОВ, А.А. МАРЧЕНКО, В.К. КИСЕНКО

МЕТОД ВЫЧИСЛЕНИЯ СЕМАНТИЧЕСКОЙ БЛИЗОСТИ-СВЯЗНОСТИ МЕЖДУ СЛОВАМИ ЕСТЕСТВЕННОГО ЯЗЫКА

Ключевые слова: *компьютерная лингвистика, семантический анализ текстов на естественном языке, семантическая близость-связность слов, смысловая неоднозначность слов.*

ВВЕДЕНИЕ

Ключевым элементом в машинном моделировании естественно-языковых процессов является возможность определять семантическую близость — смысловое расстояние между понятиями, которое часто задается на графе понятий-концептов онтологической базы знаний. Вычисление семантического расстояния широко используется во многих задачах вычислительной лингвистики, таких как: автоматическое реферирование и аннотирование текстов, разрешение смысловых неоднозначностей, анализ анафор, индексирование и поиск, машинный перевод.

В естественном языке существует ряд классических проблем, представляющих значительную сложность для большинства задач компьютерной лингвистики, а именно: полисемия, омонимия, анафорические ссылки, местоимения и другие языковые феномены, компьютерная обработка которых невозможна без семантического анализа и смысловой интерпретации текста. Суть проблем полисемии и омонимии в том, что одни и те же слова означают множества различных понятий (например, английское слово *bank* имеет разные семантические значения: финансовое учреждение и берег реки). Контекст, в котором находится данное слово, подсказывает, в каком значении оно употреблено. Для того чтобы учесть влияние контекста и определить реальное значение некоторого слова, компьютерной системе необходимо для каждого значения этого слова найти оценку семантической близости по отношению к значениям слов, расположенных рядом с ним в тексте. Это решается применением функции вычисления семантической близости и связности понятий.

Проблема анафор в компьютерной лингвистике заключается в том, что одна и та же сущность в тексте упоминается с использованием разных слов-названий; частный случай анафоры — местоимения. Для каждого местоимения может существовать целый набор кандидатов на замену (антецедентов) — группы существительных, расположенные выше по тексту, на которые может указывать данное местоимение. Определить, какой из кандидатов — правильный антецедент, можно, подставив каждого из них вместо местоимения (анафоры) и вычислив, насколько контекст кандидата на замену соответствует контексту местоимения (анафоры). Такое соответствие также находится с помощью функции вычисления семантической близости и связности понятий.

Отношение семантической близости указывает не только на отношение синонимии — понятия могут быть близки по смыслу, но не тождественны. Наличие множества других отношений обусловливает уточнение семантической связности: *двигатель* и *автомобиль* связаны отношением часть–целое, *холодное* и *горячее* — антонимы. В то же время между многими словами сложно установить прямое отношение (например, *зима* и *метель*), но, несмотря на это, между ними видна явная семантическая связь.

Отношения семантического близости и семантической связности различаются. Если *лодка* и *катер* — семантически близкие концепты, то *двигатель* и *топливо* — семантически связные понятия, хотя и не подобны по смыслу.

Семантическая близость и семантическая связность — отношения, традиционно определяемые на семантическом графе онтологической базы знаний. Определение наличия того или иного отношения между понятиями реализуется проверкой существования в онтологической сети семантических связей между узлами, которые содержат соответствующие понятия. Часто такая проверка сводится к задаче поиска кратчайшего пути между вершинами–понятиями в графе базы знаний. После того как путь построен, следует этап его анализа и интерпретации, цель которых — определение семантического значения найденного пути, т.е. какой тип семантической связи существует между данными понятиями и какова глубина этой связи.

Существует также другой подход к определению оценки семантической близости-связности понятий, предложенный в [1]. Методы этого направления вычисляют пересечение лексического состава статей-определений для двух входных понятий, и чем больше слов попадают в пересечение, тем более связанными считаются эти понятия.

В данной статье предложен новый метод определения семантической связности понятий. Предполагается, что целесообразнее вычислять и рассматривать не простое пересечение множеств лексем двух статей некоторого тезауруса, дающих определение для двух входных понятий, а учитывать также позицию каждого слова внутри статьи-определения понятия. Для этого необходимо структурировать статью тезауруса разбиением на зоны различной степени приоритета, например, «название», «определение», «ссылки на другие термины», «описательная часть». В зависимости от того, куда попало то или иное значащее слово, ему присваивается определенный приоритетный вес. Таким образом, рассматривается не простое множество лексем текста определения понятия, а множество подмножеств терминов, где каждое подмножество имеет свой вес. Предлагается вычислять и анализировать не пересечение двух лексических множеств текстов определений входных понятий, а пересечение структурированных «многоуровневых» множеств. Это позволяет просмотреть все варианты попарных пересечений подмножеств из первого и второго множества и учесть тонкие нюансы лексической структурной организации текстов: например, сколько общих слов в названиях первого и второго понятия (такое пресечение имеет наивысший вес приоритета), сколько общих слов в определении первого понятия и в названии второго (очевидно, вес должен быть меньше предыдущего), сколько общих слов в определении первого понятия и описательной части статьи второго (вес понижается еще больше) и т.д. Анализируя все возможные варианты многоуровневых пересечений и подбирая оптимальный вес для каждого варианта, можно построить качественно новую эффективную оценку семантической близости-связности слов естественного языка.

СОВРЕМЕННЫЕ МЕТОДЫ ВЫЧИСЛЕНИЯ СЕМАНТИЧЕСКОЙ БЛИЗОСТИ

Рассмотрим ранее созданные методы вычисления семантического расстояния. С начала 80-х годов прошлого столетия разработано несколько эвристических методов.

Очень важным является выбор источника данных — основы для вычисления семантической близости. В исследованиях чаще всего используются лингвистические базы знаний WordNet, ConceptNet; задействованы также Wikipedia, поиск Google. Наиболее значительные результаты достигнуты при использовании WordNet и Wikipedia [2–4].

Один класс методов базируется на вычислении расстояния $\rho(c_1, c_2)$ между двумя концептами (узлами) c_1, c_2 в некоторой таксономии (WordNet, дерево категорий Wikipedia). Так, например, может быть использован кратчайший путь между двумя соответствующими вершинами в данной таксономии. Одна из первых таких метрик предложена в работе [5]:

$$\rho(c_1, c_2) = \frac{1}{N_p},$$

где N_p — количество вершин в кратчайшем пути, связывающем узлы c_1, c_2. Отмечено, что минусом этой метрики является неравномерность глубин некоторых концептов в таксономии. В [6] приведена нормализованная версия данного метода, учитывающая высоту используемой таксономии:

$$\rho(c_1, c_2) = -\log \frac{N_p}{2D},$$

где D — максимальная глубина дерева таксономии.

Еще один метод описан в [7]. В предложенном алгоритме учитывается $\mathrm{LSO}(c_1, c_2)$ — глубина наименьшего общего предка (Lowest Super Ordinate) двух узлов графа таксономии, которые соответствуют концептам c_1, c_2:

$$\rho(c_1, c_2) = -\log \frac{\mathrm{depth}(\mathrm{LSO}(c_1, c_2))}{\mathrm{depth}(c_1) + \mathrm{depth}(c_2)},$$

где $\mathrm{depth}(x)$ — расстояние от корня таксономии до узла x.

В работе [8] впервые использована Wikipedia для вычисления семантического расстояния. Метод WikiRelate! применяет описанные выше метрики на дереве категорий Wikipedia.

Другой класс алгоритмов разработан М. Леском [1]. Он построил алгоритм, основанный на идее определения близких понятий с помощью схожего набора слов. В качестве семантического расстояния между понятиями использовано отношение количества одинаковых слов в определениях понятий к общему количеству слов в двух определениях.

На протяжении последних пяти лет разработано несколько методов, основанных на использовании Wikipedia, которые обладают недостижимой ранее точностью. В [9] предложен метод вычисления семантической близости Wikipedia Link-based Measure (WLM), основанный на использовании ссылок между страницами. Главной его идеей является предположение о том, что понятие (в данном случае представленное статьей Wikipedia) достаточно точно описывается с помощью входящих и исходящих ссылок. Каждая ссылка имеет свой вес, определяемый частотой ее появления среди всех страниц энциклопедии. Таким образом, каждой статье соответствует вектор со ссылками. Вес ссылки вычисляется с применением известной формулы TD-IDF. Расстояние между статьями находится с помощью косинусного расстояния между векторами весов статей.

Один из наиболее эффективных методов — Explicit Semantic Analysis (ESA) — описан в [4]. По сравнению с ранее известным алгоритмом Latent Semantic Analysis (LSA), в котором определяются неявные связи между текстами статей, в данном методе понятие представляется в явном виде с помощью взвешенной сумы терминов, полученных из Wikipedia. Заданное понятие проектируется в пространство векторов-статей Wikipedia. Таким образом, семантическая близость определяется как косинусное расстояние между векторами, спроектированными в пространство статей Wikipedia.

В работе 10] представлен метод WikiWalk, применяющий технику случайных блужданий на графе. Рассматривается два типа графов: построенные с помощью WordNet и Wikipedia. Этот метод использует алгоритм Personalized PageRank: некая частица случайно блуждает по вершинам графа (в случае Wikipedia — по статьям) и переходит на новую страницу с некой вероятностью. Таким образом, каждая вершина графа определяется вектором вероятностей переходов на другие страницы (вектором телепортаций). Такой вектор оказывается уникальной характеристикой страницы Wikipedia (а с ней и описанного понятия). Семантическая близость вычисляется как расстояние между векторами телепортаций соответствующих страниц.

МЕТОД ВЫЧИСЛЕНИЯ СЕМАНТИЧЕСКОЙ БЛИЗОСТИ-СВЯЗНОСТИ

Источником данных, который используется в настоящей работе, служит свободная интернет-энциклопедия Wikipedia. В данный момент английская версия Wikipedia содержит более 3,5 миллионов словарных статей, русская — более 600 тысяч, украинская — более 250 тысяч. Такое большое количество статей обеспечивается «свободностью» проекта. Каждый пользователь может создавать, исправлять и дополнять статьи. Благодаря модерации это не ведет к снижению качества текстов статей, практически каждое изменение проверяется одним или группой пользователей, которые ранее тем или иным образом доказали свою компетентность. Очень важным фактором также является возможность загрузки полной локальной копии Wikipedia. Однако эта энциклопедия имеет определенные недостатки. Некоторые статьи не полностью объективны: например, автор может внести свое личное мнение по поводу того или иного вопроса. Еще один минус — недостаточная строгость формата описания статьи, что очень усложняет разработку программы-анализатора текстов энциклопедии. Интернет-энциклопедия Wikipedia является уникальным и ценным, но не формализированным источником данных.

Структура Wikipedia имеет ряд свойств, которые можно использовать при вычислении семантической близости. Эти свойства могут моделировать некоторые типы лексических отношений между словами.

• **Синонимия.** Определяется с помощью страниц-перенаправлений. Как правило, содержимое таких статей состоит из строки «#REDIRECT <имя страницы >». Например, статья *кот* направляет на страницу *кошка*, а статья *авто* — на *автомобиль*.

• **Омонимия.** Задается специальными страницами со списком возможных значений данного понятия. В качестве примера можно привести страницу *нота*, которая содержит ссылки на различные значения этого слова, например: *музыкальный знак*, *дипломатическое обращение*, *финансовая облигация*, *марка магнитофонов*, *название реки*. В данной работе страницы такого типа используются для разрешения смысловых неоднозначностей, в частности для получения списка возможных значений термина.

• **Перекрестные ссылки.** Представлены ссылками на другие статьи Wikipedia. Например, статья *вода* содержит ссылки на статьи: *химическое вещество*, *жидкость*, *лед*, *снег*, *пар*, *растворитель*, *океан*, *река*, *жизнь*, *погода*, *климат* и т.д. Такие ссылки указывают на взаимосвязь между понятиями.

Предлагаемый метод, назовем его «оценочное взвешенное пересечение» (Estimated Weighted Overlap — EWO), является развитием ранее упомянутого подхода Леска. Его метод исходит из предположения, что близкие понятия описываются (или определяются) с использованием подобного набора слов, т.е. количество общих слов в словарных определениях может показывать, насколько

эти два понятия близки семантически. Предлагаемое функционально-структурное обобщение метода Леска основывается на идее о том, что в тексте отражено смысловое упорядочение между словами. Некоторые слова являются более важными, чем другие, исходя из их позиции в тексте. Например, слово из названия статьи (или из определения термина) обычно имеет большее значение, чем слово из конца текста. Для введения такого различия в предлагаемом алгоритме каждому слову из текста статьи присваивается вес, соответствующий важности слова. Веса слов рассчитываются на основе следующих признаков: название статьи содержит данное слово; слово принадлежит определению понятия; слово принадлежит первому параграфу статьи; слово является ссылкой на другую статью; другие слова.

Итак, предположим, алгоритм получает на вход два слова для оценки. Прежде всего он выбирает соответствующие словарные статьи из Wikipedia. После этого тексты статей разбиваются на слова. Далее алгоритм удаляет слова из «стоп-списка». Стоп-список содержит слова, которые не несут большой семантической нагрузки: предлоги, союзы, местоимения, общеупотребительные слова и т.д. На следующем шаге алгоритм разбивает множества слов на подмножества, соответствующие заданным факторам. Например, для описанных выше признаков множества будут такими: L_1 — слова из названия; L_2 — слова из определения понятия; L_3 — слова из первого параграфа; L_4 — слова, являющиеся перекрестными ссылками; L_5 — остальные слова. Причем если некоторое слово w попадает в L_i, то оно исключается из L_j для любого $j > i$.

В данном методе предлагается анализировать не просто пересечения двух лексических наборов статей Wikipedia для двух входных понятий, а учитывать структуру статей. Если названия и определения понятий содержат общие термины, то список пересечения лексем для названий и определений должен иметь намного больший вес важности, чем список пересечения для всего остального тела статьи. Предлагается разбить значащие слова обеих статей по соответствующим признакам на группы $L_1^1, \ldots, L_n^1, L_1^2, \ldots, L_n^2$ и далее считать пересечения попарно: $L_i^1 \cap L_j^2$. В рассматриваемом случае количество признаков равно пяти (число признаков может быть другим в иной реализации алгоритма). Для каждого возможного пересечения определяется соответствующий вес приоритета: максимальный — для случая пересечения терминов из названия $L_1^1 \cap L_1^2$, минимальный — для общих терминов из описательной части статьи $L_5^1 \cap L_5^2$. Промежуточному варианту, например, когда некоторые термины используются в определении первого понятия, а для второго понятия они фигурирует описательно в конце статьи (пересечение $L_2^1 \cap L_5^2$), присваивается промежуточный вес.

Каждому слову из множества L_i присвоен вес w_i. На основании множеств L_i^1 и L_i^2 строится матрица D, элемент которой $D[i, j]$ равен количеству общих слов в L_i^1 и L_i^2 — $|L_i^1 \cap L_j^2|$, умноженному на вес $w_{ij} = w_i + w_j$. Предположим, что семантическая близость равна нормализованной сумме элементов матрицы D.

Алгоритм выполняет следующие действия.

1. Для двух понятий c_1, c_2 извлечь статьи t_1 и t_2, определяющие эти понятия. Выбрать все слова из статей t_1 и t_2. Обозначить множества слов как T_1 и T_2 соответственно.

2. Удалить из T_1, T_2 слова из стоп-списка.

3. Разбить множества T_1, T_2 на подмножества $L_1^1, \ldots, L_n^1$ и $L_1^2, \ldots, L_n^2$ по заданным признакам, где n — количество признаков.

4. Построить матрицу D:

$$\begin{pmatrix} w_{11}|L_1^1 \cap L_1^2| & \ldots & \ldots & w_{1n}|L_1^1 \cap L_n^2| \\ \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots \\ w_{n1}|L_n^1 \cap L_1^2| & \ldots & \ldots & w_{nn}|L_n^1 \cap L_n^2| \end{pmatrix}.$$

5. Вычислить величину семантической близости как нормализованную сумму:

$$\mathrm{EWO}(c_1, c_2) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} D_{i,j}}{\sum_{i=1}^{n} w_i (|L_i^1| + |L_i^2|)}.$$

Процедура получения весов w_i, основанная на алгоритме имитации отжига (метод глобальной дискретной оптимизации), детально описана далее.

РАЗРЕШЕНИЕ СМЫСЛОВЫХ НЕОДНОЗНАЧНОСТЕЙ

Некоторые понятия могут иметь одинаковое написание и разное значение. Например, слово *ягуар* может означать животное из рода кошачьих и марку британского автомобиля. Таким образом, необходимо правильно выбирать значение (и статью из Wikipedia), в зависимости от второго слова пары. Например, если на вход алгоритма подана пара слов <*ягуар*; *лев*>, то *ягуар* должен считаться большой кошкой, а если пара <*ягуар*; *мерседес*>, то интерпретироваться как марка автомобиля. Для разрешения таких неоднозначностей разработан алгоритм.

Как и в предыдущем случае, алгоритм получает на вход пару слов. Для обоих слов получаем список возможных статей-кандидатов (значений). Затем для каждой пары значений, где первое значение принадлежит одному списку, второе — другому, вычисляется величина семантической близости. После этого выбирается пара с наибольшим значением. Более формально алгоритм запишется следующим образом.

1. Для обоих слов получить список значений:
   - извлечь из индекса список статей с названием вида <слово> (уточнение);
   - (дополнительно) извлечь из страницы с описанием неоднозначностей список возможных значений.
2. Для каждой пары статей подсчитать значение семантической близости-связности.
3. Выбрать пару с наибольшей семантической близостью.

В практических реализациях этот процесс можно оптимизировать: вместо пересечения полного текста статей использовать только первые параграфы. Такая оптимизация значительно снижает трудоемкость процесса, при этом не влияя на точность вычислений.

ОЦЕНКА ВЕСОВ

Для оценки весов w_{ij} используется метод имитации отжига [11] — вероятностная эвристика для решения задач глобальной оптимизации. Данный метод оперирует точками в пространстве решений. В рассматриваемом случае точ-

кой является вектор из пяти весов, которые соответствуют выбранным признакам. На каждой итерации алгоритма хранится одна точка — текущая, которая может быть изменена по определенному вероятностному правилу. Псевдокод [12] этого алгоритма для максимизации функции $F(x)$ имеет следующую структуру.

1. Выбрать случайным образом начальную точку x_0.
2. Положить $x_{\text{best}} = x_0$.
3. Пока $i < k$, выполнять такие шаги:
   - случайно выбрать точку x среди соседей точки x_i;
   - если $F(x_{\text{best}}) < F(x)$, то $x_{\text{best}} = x$;
   - если $F(x_i) < F(x)$, то $x_{i+1} = x$;
   - если $\text{rnd} < e^{(F(x)-F(x_i))/t_i}$, то $x_{i+1} = x$.
4. Вернуть x_{best}.

Здесь rnd — случайное число от 0 до 1, параметр t_i — элементы некоторой убывающей последовательности. Эти значения называются температурой отжига.

В целом, данный метод подобен методу градиентного спуска, но использование вероятностного закона не позволяет алгоритму «застревать» в точках локального максимума. Это свойство помогает получать более эффективные результаты.

В качестве функции для максимизации применен коэффициент ранговой корреляции Спирмена. Пространством решений для поиска является пространство векторов, размерность которых равна количеству признаков, используемых в алгоритме, т.е. каждой координате вектора соответствует вес некоторого признака. Для оценки весов создана небольшая тренировочная база, состоящая из пар слов, принадлежащих основным классам отношений семантической близости-связности: очень близкие понятия, абсолютно независимые понятия, слова с множеством значений и т.д. Несколько раз запущена оптимизирующая процедура и выбраны веса, которые дают максимальную корреляцию с тренировочной базой.

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

Разработана программная реализация предложенного метода. Программа написана на языке программирования Scala [13, 14] — современном, хорошо проработанном языке, удобном для создания программ обработки текстов. Текущая реализация Scala компилирует исходный текст в байт-код для виртуальной машины JVM. Это свойство дает возможность выполнять программу на всех операционных системах, которые поддерживаются JVM (например, Windows, GNU/Linux, MacOS X). В качестве источника данных используется локальная копия Wikipedia, загруженная с веб-сайта проекта. Общий размер архива чрезвычайно велик (более 5,5 Гб), поэтому для реализации эффективного, быстрого поиска статей выполнена предварительная обработка. Отметим, что для создания архива используется блочная архивация. Это позволяет разбить большой архив на множество маленьких (около 1 Мб каждый) и создать поисковый индекс для них. В средине архива находится единственный XML-файл (размером около 25 Гб), который содержит все статьи Wikipedia. Для извлечения статей из этого файла с учетом блочной структуры архива разработана программа-парсер, способная обрабатывать большие объемы данных. В общих чертах предварительную обработку можно описать следующим образом.

1. Для каждой статьи из локальной копии Wikipedia:
   - извлечь название и текст;
   - исключить из текста части, которые не важны для алгоритма, например ссылки на внешние ресурсы, комментарии, описания изображений;
   - сохранить название и обработанный текст статьи в текстовом файле;
   - добавить в базу данных пару <Название статьи; название текстового файла, в котором хранится содержимое>.

2. После обработки всех статей из Wikipedia создать индекс базы данных для поля «название статьи».

Таким образом, статьи сохраняются в обычных текстовых файлах. В качестве базы данных используется MongoDB — современная нереляционная, документно-ориентированная база данных, которая, согласно результатам множества тестирований, считается одной из наиболее производительных. Важным также является возможность поиска в базе данных по регулярным выражениям, что активно используется при разрешении смысловых неоднозначностей. Размер конечной базы данных — 1,5 Гб. В целом, такой подход к хранению данных позволил добиться чрезвычайно высокой производительности в поиске и извлечении статей.

Для оптимизации весовых параметров разработано отдельное приложение (реализация метода имитации отжига). Взаимодействие оптимизатора с программой происходит посредством конфигурационных файлов. Программа-оптимизатор выдает ответ в виде вектора вещественных чисел — весовых параметров алгоритма, при которых достигается наибольшая корреляция с обучающей выборкой.

Программа вычисления семантического расстояния разработана в двух версиях: с консольным и графическим интерфейсом. Графический интерфейс дает возможность в интерактивном режиме вводить пары слов для оценки семантической близости. Такой интерфейс более удобен для пользователя и позволяет, кроме непосредственно оценки, просматривать множество дополнительной информации: тексты статей, списки статей-кандидатов, веса слов и т.д. Консольный интерфейс является более подходящим для вызова из других программ и контролируется с помощью параметров командной строки. Планируется также разработка отдельной подгружаемой библиотеки для лучшей интеграции со сторонними приложениями.

Для тестирования алгоритмов вычисления семантической близости-связности часто используется набор взвешенных пар слов Finkelstein WordSimilarity-353 [15]. Он содержит 353 пары слов, которые оценены экспертами-людьми. Каждая пара оценена действительным числом от 0 до 10. В качестве оценки работы предложенного алгоритма использовался коэффициент ранговой корреляции Спирмена. Далее приведены коэффициенты корреляции вычисленных предложенным алгоритмом значений с оценками из Finkelstein WordSimilarity для трех режимов:

- без разрешения смысловой неоднозначности — 0,63;
- с частичным разрешением смысловой неоднозначности (кандидатами являются статьи с названием вида <слово> (<уточнение>)) — 0,68;
- с полным разрешением смысловой неоднозначности (кандидаты получены из статей-списков неоднозначностей; как правило, это статьи с названием <слово> (disambiguation)) — 0,74.

Данные значения указывают на существенное улучшение результатов при использовании разрешения смысловой неоднозначности. Для сравнения с некоторыми другими методами построена диаграмма (рис. 1), отражающая результаты измерений для различных алгоритмов вычисления семантической близости. На диаграммае приведены оценки, полученные разными методами:

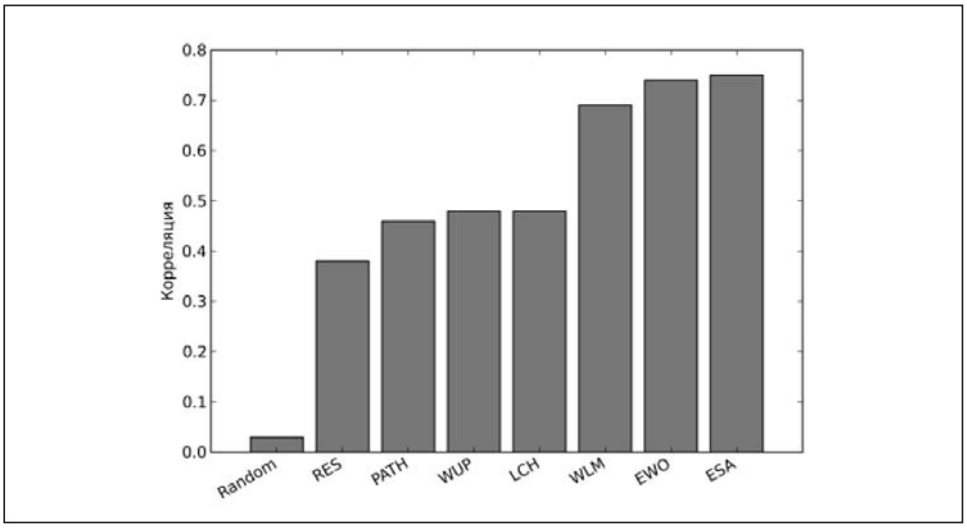

Рис. 1

- метод RND, возвращающий случайное значение для пары слов;
- методы, основанные на поиске пути в графе, а именно метод кратчайшего пути (PATH), метод Ликока–Чодорова (LCH), метод Ву–Палмера (WUP), метод Резника (RES) [8, 16];
- метод WLM [9];
- метод ESA [4, 9];
- метод EWO.

Таблица 1

Пара слов		Оценка семантической близости-связности слов	
слово 1	**слово 2**	**эксперт**	**алгоритм**
car	automobile	8.94	9.99
magician	wizard	9.02	6.93
glass	magician	2.08	1.1
money	currency	9.04	5.67
noon	string	0.54	0.82
FBI	fingerprint	6.94	4.05
tiger	cat	7.35	4.13
tiger	tiger	10	10
book	paper	7.46	4.44
computer	keyboard	7.62	4.38
computer	internet	7.58	4.04
physics	chemistry	7.35	4.28
drink	ear	1.31	1.13

Программная реализация метода EWO показывает его высокую производительность: оценка 20–100 пар слов в секунду. Примеры результатов работы программы вычисления оценки семантической близости-связности слов на тестовой выборке приведены в табл. 1.

ЗАКЛЮЧЕНИЕ

В данной статье описан новый эффективный метод вычисления семантической близости-связности между словами естественного языка. Представленный алгоритм является модификацией известного подхода Леска. Он построен на основе позиционного структурирования текста словарных статей глоссария, после которого каждый значимый термин получает приоритетный вес в зависимости от расположения в той или иной части текста статьи, что позволяет вычислять разноуровневые лексические пересечения с разным весом приоритета. При этом учитываются нюансы лексической структуры статей определений понятий, а не простое словарное пересечение двух текстов. В качестве источника данных для вычислений используется интернет-энциклопедия Wikipedia. Для определения весовых параметров применяется метод имитации отжига.

Описанный метод показал высокой уровень корреляции с тестовыми данными. Таким образом, предложенный алгоритм демонстрирует результаты на уровне лучших современных методов, при этом являясь прозрачным и интуитивным. Разработана программная реализация метода, высокая скорость работы которой позволяет использовать ее при решении разнообразных задач компьютерной лингвистики.

Возможно несколько путей улучшения качества оценки:

- добавление новых факторов в весовую модель;
- интеграция с другими техниками вычисления семантической близости для построения комплексной оценки.

Производительность может быть повышена, например, разработкой параллельной версии программы. Это позволит использовать современные многопроцессорные и многоядерные вычислительные системы.

Данная программа вычисления семантической близости-связности между словами естественного языка разработана в рамках комплекса многоцелевых прикладных систем семантического анализа и смысловой обработки текстовых документов.

СПИСОК ЛИТЕРАТУРЫ

1. Lesk M. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone // SIGDOC'86: Proc. of the 5th Annu. Intern. Conf. on Syst. document. — New York: ACM, 1986. — P. 24–26.
2. Wubben S. Using free link structure to calculate semantic relatedness: (Rep.) / ILK Res. Group Techn. — N 08-01. — Tilburq: Tilburq Univ., 2008.
3. Ponzetto S.P., Strube M. Knowledge deriver from Wikipedia for computing semantic relatedness // Artif. Intell. Res. — 2007. — N 30. — P. 181–212.
4. Gabrilovich E., Markovitch S. Computing semantic relatedness using Wikipedia-based explicit semantic analysis // Proc. of the 20th Intern. Joint Conf. on Artif. Intell., Hyderabad (India), 2007. — San Francisco: Morgan Kauffman Publ., 2007. — P. 1606–1611.
5. Resnik P. Using information content to evaluate semantic similarity in a taxonomy // Proc. of Intern. Joint Conf. on Artif. Intell., Montreal, 1995. — San Francisco: Morgan Kauffman Publ., 1995. — P. 448–453.
6. Leacock C., Chodorow M., and Miller G.A. Using corpus statistics and wordnet relations for sense identification // Comput. Ling. — 1998. — **24**, N 1. — P. 147–165.
7. Wu Z., Palmer M. Verb semantics and lexical selection // 32nd. Annu. Meet. of the Assoc. for Comput. Ling., Las Cruces (USA), 1994. — San Francisco: Morgan Kauffman Publ., 1994.— P. 133–138.
8. Strube M., Ponzetto S.P. WikiRelate! Computing semantic relatedness using Wikipedia // Proc. of the 21st Nat. Conf. on Artif. Intell., Boston, 2006. — Berlin: Springer, 2004. — P. 1419–1424.
9. Milne D., Witten I.H. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links // Proc. of the first AAAI Workshop on Wikipedia and Artif. Intell. (CIKM'2008), Chicago, 2008. — Menlo Park (USA): AAAI Press, 2008.
10. WikiWalk: Random walks on Wikipedia for semantic relatedness / E. Yeh, D. Ramage, C.D. Manning, et al. // ACL-IJCNLP TextGraphs-4 Workshop 2009. — Singapore, 2009.
11. Kirkpatrick S., Gelatt C.D., Vecchi M.P. Optimization by simulated annealing // Science. New Ser. — 1983. — N 220. — P. 671–680.
12. Luke S. Essentials of metaheuristics. — 2009. — http://cs.gmu.edu/!sean/book/metaheuristics/.
13. Odersky M. Scala by example / Progr. meth. lab., EPFL. — Lausanne, 2009. — 145 p.
14. Odersky M., Spoon L., Venners B. Programming in Scala. — Montain View: Artima Press, 2008. — 754 p.
15. Placing search in context: The concept revisited / L. Finkelstein, E. Gabrilovich, Y. Matias, et al. // ACM Trans. Inform. Systems. — 2002. — **20**, N 1. — P. 116–131.
16. Pedersen T., Pathwardhan S., Michelizzi J. Wordnet::Similarity — Measuring the relatedness of concepts // Proc. of the 19th Nat. Conf. on Artif. Intell., San Jose (USA), 2004. — Berlin: Springer, 2004. — P. 1024–1025.

Поступила 10.03.2011