Модель семантического поиска на базе тезауруса

© Д.А. Малахов[1] © В.А. Серебряков[1,2]

[1]Московский государственный университет им. М.В. Ломоносова,
[2]Федеральный исследовательский центр «Информатика и управление» РАН,
Москва, Россия

mda.develop@gmail.ru serebr@ultimeta.ru

Аннотация. Представлена модель семантического поиска, которая базируется на применении тезауруса. Описаны ключевые моменты использования модели. Приведены основные возможности тезаурусов, способы их применения в других поисковых системах, а также особенности нашего подхода.

Ключевые слова: семантический поиск, L-теги, применение тезауруса, WordNet.

The Semantic Search Model Based on the Thesaurus

© Dmitriy Malakhov[1] © Vladimir Serebryakov[1,2]

[1]Lomonosov Moscow State University,
[2]Federal Research Center Computer Science and Control of the Russian Academy of Sciences,
Moscow, Russia

mda.develop@gmail.ru serebr@ultimeta.ru

Abstract. This article presents a model of semantic search, based on the thesaurus. The key points of using the model are described. The main features of the thesaurus, the methods of their application in other search systems, and also features of our approach are presented.

Keywords: semantic search, L-tags, thesaurus application, WordNet.

1 Введение

Семантическим поиском, как правило, называется процесс поиска документов по их содержанию. Нетрудно увидеть, что понятие семантического поиска недостаточно формально определено [7]. В частности, понятие содержания или смысла является многозначным.

Существуют различные подходы к реализации семантического поиска. Как правило, выделяют следующие классы моделей семантического поиска:

- Поиск, основанный на структурированных SPARQL запросах к базе знаний в формате OWL/RDF.
- Поиск, основанный на семантическом аннотировании документа с последующей индексацией аннотаций.
- Полнотекстовый поиск, использующий словари синонимов для индексации документов и расширения запросов.
- Всевозможные гибридные решения.

Далее предложена модель семантического поиска, являющаяся гибридным вариантом, так как содержит элементы семантического аннотирования и полнотекстового поиска. В предыдущей работе [2] была представлена модель семантического поиска, основанная на использовании L-тегов.

Определение 1.1. Алфавитом будем называть любое конечное непустое множество. Элементы этого множества называются символами данного алфавита.

Определение 1.2. Термином $t \in T$ алфавита A будем называть любой упорядоченный конечный непустой набор символов алфавита A.

Определение 1.3. L-тегом на множестве терминов T будем называть любой непустой упорядоченный набор терминов из T.

Основная идея использования L-тегов заключается в разбиении алгоритма расчёта релевантности на два этапа. Первый этап заключается в выделении L-тегов в документе и оценке их значимости для этого документа. Эта оценка характеризуется функцией семантики, отображающей пару («документ», «L-тег») в действительное число от 0 до 1. Для качественного поиска важно, чтобы выделенные L-теги полностью покрывали содержание документа. Выделение этого этапа позволяет производить сложные вычисления для расчёта релевантности, причём не во время выполнения запроса, а на этапе индексации.

Второй этап заключается в поиске L-тегов, схожих с запросом пользователя на естественном языке. Поисковый запрос является L-тегом, поэтому релевантность L-тега запросу может быть оценена с помощью функции схожести двух L-тегов,

Труды XIX Международной конференции «Аналитика и управление данными в областях с интенсивным использованием данных» (DAMDID/ RCDL'2017), Москва, Россия, 10–13 октября 2017 года

отображающей пару («запрос», «L-тег») в действительное число от 0 до 1. Функция схожести должна рассчитываться во время выполнения запроса, поэтому должна выполняться достаточно быстро. Комбинация функции схожести и функции семантики характеризует релевантность запроса документу.

Каждый L-тег описывает некоторую информационную потребность. Различные реализации функций семантики и схожести отличаются друг от друга различным пониманием информационной потребности, различным способом оценки схожести информационных потребностей и удовлетворения информационной потребности. Ниже с помощью понятия контекста будут описаны способы реализации функции семантики и функции схожести.

Если в качестве L-тегов рассматривать предложения или абзацы в тексте, то представленная модель позволяет существенно уменьшить поисковый индекс за счет игнорирования L-тегов с достаточно малым значением функции семантики.

Использование модели позволяет применять единые механизмы поиска в случае индексирования не только текстов, но и семантических аннотаций, привязанных к этим текстам, так как семантическую аннотацию можно представить как L-тег или набор L-тегов. Модель семантического поиска в том виде, в котором она была описана ранее, является достаточно общей и не регламентирует, как именно должны быть определены функции семантики и схожести. В настоящей работе предложены уточнения модели семантического поиска для случаев, когда имеется достаточно хороший тезаурус.

2 Применение тезаурусов

2.1 Термины и понятия

К тезаурусам могут быть отнесены достаточно разные словари и лингвистические ресурсы [6]:

- Идеографический словарь, основное назначение которого – помощь в подборе близких по смыслу слов при написании текста.
- Информационно-поисковый тезаурус описывает отношения между терминами предметной области.
- Тезаурус типа WordNet описывает отношения между лексическими значениями естественного языка.
- Ассоциативный тезаурус, описывающий ассоциации людей или совместную встречаемость слов в тексте, рассчитанную автоматически.

Как правило, тезаурусы оперируют двумя сущностями: термином и понятием. Под термином понимается слово или словосочетание, имеющее некоторое смысловое значение. Особенностью естественного языка является то, что одно и то же смысловое значение может быть передано различными терминами. В тезаурусах смысловое значение принято называть понятием, а набор терминов, которые передают это смысловое значение, – синсетом.

Для естественного языка также характерно, что один и тот же термин в разных контекстах характеризует разные понятия. Хороший тезаурус в рамках сферы своего применения должен определять всевозможные понятия термина, а также предоставлять информацию о том, как определить понятие, которое термин характеризует в некотором контексте.

Под тезаурусом мы будем понимать словарь, оперирующий понятиями, которые характеризуются синонимическими рядами (синсетами) и имеют между собой семантические связи, как вертикальные, так и горизонтальные. Далее мы более подробно рассмотрим информационно-поисковый тезаурус и WordNet.

2.2 Информационно-поисковый тезаурус

Информационно-поисковые тезаурусы создавались для описания различных предметных областей и использовались для ручной разметки документов и запросов. Основная идея использования такого рода тезауруса заключалась в определении применяемой терминологии для использования в запросах и индексации документов. Впоследствии эксперименты показали, что эффективность полнотекстового индексирования сравнима с эффективностью поиска, использующего ручное индексирование по [5], [6]. С учетом трудоемкости ручного индексирования оно все чаще заменялось полнотекстовым поиском.

Казалось бы, что информационно-поисковые тезаурусы могут быть полезными в семантическом поиске, но есть две основные проблемы:

- Ориентированность на узкую предметную область не позволяет описать всевозможные значения того или иного термина в целом. В свою очередь документы зачастую могут охватывать различные предметные области, а у пользователей могут быть различные потребности. Поэтому для наиболее полного описания документа может понадобиться несколько тезаурусов, часть понятий которых может пересекаться. В этом случае мы сталкиваемся с проблемой интеграции тезаурусов.
- Более важной проблемой является то, что такие тезаурусы создавались для людей, а не для машин. Поэтому они могут не содержать полного списка синонимов в синсетах, так как подразумевается, что человек догадается, в каком случае нужно привязывать понятие.

Эксперименты по автоматическому индексированию документов и запросов на базе информационно-поисковых тезаурусов не привели к их практическому использованию для автоматической обработки текстов [6].

Таким образом, информационно-поисковые тезаурусы не могут быть использованы явным образом для задачи семантического поиска.

2.3 WordNet

Основные гипотезы, на базе которых разработан WordNet [1]:

- Гипотеза отделимости означает, что лексический уровень языка может быть отделен от морфологического и синтаксического.
- Гипотеза «образца» означает, что существует формальное описание для большинства слов языка.
- Гипотеза о покрытии означает, что словарь должен быть достаточного размера для покрытия всех понятий, чтобы быть эффективным в задачах автоматической обработки текстов.

Разработчики WordNet считают, что два термина могут находиться в одном синсете понятия, если замена одного термина на другой в контексте этого понятия не изменяет смысла предложения. В таком случае термины считаются синонимами. Большинство синсетов имеет толкования. Если термин имеет несколько значений, то он входит в несколько синсетов.

Самые распространенные связи в WordNet:

- Родовидовое отношение используется для существительных. Синсет X называется гипонимом синсета Y, если считается справедливым утверждение: «X – это вид Y». Родовидовое отношение выстраивает иерархию с наследованием всех свойств вышестоящего нижестоящим.
- Отношение «часть–целое» используется для существительных. Синсет X является частью синсета Y, если считается справедливым утверждение: «X – это часть Y».
- Отношение антонимии используется для существительных, прилагательных и наречий, причем связываются не понятия, а термины. Считается, что термины X и Y – антонимы, если одно исключает второе, например, победа – поражение, мужчина – женщина.
- Отношение между однокоренными словами, используется для существительных и различных глагольных форм.

Для описания глаголов были выделены специальные отношения:

- Отношение следования устанавливается между синсетами V1 и V2, если из предложения «Кто-то V1» следует, что «Этот кто-то V2».
- Отношение тропонимии представляет особый вид следования и устанавливает родовидовые отношения между глаголами: «Делать V1 означает делать V2 особым способом».
- Отношение причины связывает два глагольных синсета V1 и V2 следующим образом: «Если кто-то V1, то кто-то другой V2».

Основная критика тезаурусов типа WordNet касается следующих проблем:

- Много значений одного и того же слова. Эту проблему пытались устранить кластеризацией [4], [6].
- Понятия не связаны по контексту. Так называемая «Теннисная проблема». Это усложняет выделение понятия в тексте с разрешением неоднозначности. Эту проблему пытались устранить введением доменов для большинства понятий, где домен характеризует предметную область понятия [3], [6].
- Проблема родовидовых отношений заключается в том, что под этой связью могут скрываться принципиально разные отношения: типы и роли. Эти отношения различаются в способах наследования свойств, поэтому их стоит различать.

Несмотря на критику, WordNet наилучшим образом подходит как тезаурус для задачи семантического поиска, так как наиболее полно описывает понятия и их синсеты. Далее под тезаурусом будем понимать лингвистический ресурс типа WordNet.

3 Контекст

Под контекстом понятия, как правило, подразумевают факторы, влияющие на то, что некоторый термин обозначает некоторое понятие.

Контекст может быть полезен:

- для разрешения неоднозначности термина при выделении понятий в тексте;
- для определения семантической схожести запроса и текста.

Рассмотрим некоторое множество терминов T и множество понятий N.

Определение 3.1. Под абзацем будем понимать группу предложений, идущих в тексте друг за другом, комбинация которых отражает некоторую единую мысль, что приводит к близости контекстов понятий из этого абзаца. Каждое предложение является упорядоченным набором терминов из T, обозначающих некоторые понятия из N.

Исходя из предположения, что контекст понятия характеризуется терминами, которые находятся в одном абзаце с термином понятия, ниже дано формальное определение контекста.

Определение 3.2. Пусть даны множество терминов T и конечное множество абзацев P, где абзац $p \in P$. Вектором абзаца p будем называть вектор действительных чисел V_p размерности $|T|$, компоненты которого соответствуют терминам из T и равны 0 или 1, если термин включен в p или нет, соответственно. Вектор V_p будем считать элементом нормированного векторного пространства, где норма $\|V_p\| = \sqrt{V_p \cdot V_p}$.

Определение 3.3. Пусть даны множество понятий N, множество абзацев P, и для каждого понятия $n \in N$ множество абзацев P_n, в которых было выделено понятие n. Контекстом понятия $n \in N$ будем называть вектор $C_n = (\sum_{p \in P_n} V_p) / |P_n|$.

Таким образом, под контекстом понятия будем понимать среднее арифметическое векторов абзацев, в которых понятие присутствует. Заметим, для того, чтобы посчитать контекст понятия, нужно сначала выделить его в абзацах, причем этих абзацев должно

быть достаточно много, иначе полученный результат будет неустойчивым.

3.1 Выделение контекста понятия

Чтобы определить контекст понятия, нужно выделить это понятие во всех его абзацах, значит, разрешить неоднозначность терминов.

Существуют различные подходы [3], [6] к разрешению неоднозначности при выделении понятия в тексте. Ниже будет предложен альтернативный подход, основной особенностью которого является использование кластеризации.

Сформулируем задачу следующим образом. Даны:

- множество терминов T, множество понятий N;
- множество понятий N_t, обозначенных термином t, для каждого термина $t \in T$;
- для каждого термина $t \in T$ множество абзацев P_t, в которые включен термин t;
- для каждого понятия $n \in N$ множество терминов T_n, описывающих понятие n.

Для каждого понятия $n \in N$ необходимо определить множество абзацев P_n, в которых выделено понятие n.

Для каждого $t \in T$ нужно разбить множество P_t с помощью алгоритма кластеризации, например, k-means++, на $|N_t|$ кластеров. Каждый полученный кластер абзацев P_{t_i} будет характеризовать некоторое понятие N_t. Контекст этого понятия относительно множества абзацев P_{t_i} обозначим как C_{t_i}.

При достаточном объеме данных контексты понятия, построенные по разным группам абзацев, не должны сильно отличаться. Скалярное произведение двух контекстов будет максимальным, если это контексты одного понятия. Поэтому для каждого понятия $n \in N$ нужно собрать все P_{t_i}, такие, что $t \in T_n$. Для каждого термина $t \in T_n$ необходимо оставить только один кластер P_t, так, чтобы оставшиеся кластеры для разных терминов были максимально близки друг к другу. Близость группы кластеров можно оценить с помощью функции $\sum_{t_i \in T_n,\, t_j \in T_n} C_{t_i} \cdot C_{t_j}$.

Для каждого понятия $n \in N$ оставшиеся кластеры P_t объединяются в P_n, по нему считается контекст понятия C_n.

3.2 Выделение понятий в абзаце

При выделении понятий в абзацах мы сталкиваемся с проблемой многозначности и «Теннисной проблемой» (раздел 2.3). Эти проблемы могут быть решены использованием информации о контексте понятия.

Определение 3.4. Пусть даны множество понятий N и множество абзацев P. Степенью близости понятия $n \in N$ и абзаца $p \in P$ будем называть функцию близости

$$affinity(V_p, C_n) = (V_p \cdot C_n)/(\|V_p\| \, \|C_n\|).$$

Рассмотрим абзац $p \in P$ и термин $t \in p$. Пусть N_t – множество понятий, которые могут обозначать термин t. Будем исходить из того, что мы должны выбрать понятие, контекст которого максимально похож на абзац p, тогда в качестве понятия, обозначаемого термином t, следует выбирать n_t, такое, что:

$$affinity(V_p, C_{n_t}) = \max_{n \in N_t} affinity(V_p, C_n).$$

Из-за многозначности может получиться так, что вектор абзаца похож на контексты сразу нескольких понятий. В этом случае предложенный алгоритм может быть улучшен. Мы можем привязать к термину не одно понятие, а несколько, с условием, что контекст каждого привязанного понятия близок к вектору абзаца как минимум на $M\%$ от близости контекста понятия n_t к вектору абзаца p, где M – некоторый порог.

3.3 Выделение понятий в поисковом запросе

Особенностью выделения понятия в поисковом запросе является то, что поисковый запрос в отличие от абзаца имеет намного меньше терминов. Часто поисковый запрос представляет собой последовательность из нескольких терминов, вот почему приведенный выше способ выделения понятий невозможно применить для поисковых запросов.

Определение 3.5. Пусть даны множество терминов T и множество поисковых запросов Q, где запрос $q \in Q$ является набором терминов из T. Вектором запроса q будем называть вектор действительных чисел V_q размерности $|T|$, компоненты которого соответствуют терминам из T и равны 0 или 1, если термин включен в q или нет, соответственно.

Пусть из запроса q каким-то образом было выделено множество понятий N_q. Тогда мы можем дать определение контексту запроса.

Определение 3.6. Пусть даны множество терминов T и множество поисковых запросов Q, где запрос $q \in Q$ является набором терминов из множества T. Контекстом запроса q будем называть вектор

$$C_q = (\sum_{n \in N_q} C_n)/|N_q|.$$

Если пользователь регулярно использует поисковую систему, работая со своими избранными предметными областями, то у нас есть информация о его интересах, и мы могли бы ее использовать.

Исходя из предположения, что контекст пользователя может быть определен через историю его запросов, можно дать следующее определение.

Определение 3.7. Пусть пользователь u последовательно задал K запросов. Контекстом пользователя u будем называть вектор $C_u = \sum_{k=1}^{K} \frac{C_{q_k}}{2^{K-k}}$.

Изначально контекст пользователя представляет собой вектор нулей. После выполнения очередного запроса q контекст уточняется.

На практике контекст пользователя будет разрастаться, то есть будет появляться все больше

ненулевых компонент. Для обнуления наиболее слабых компонент вектора контекста существуют следующие варианты:

- ограничение минимального значения ненулевой компоненты;
- ограничение максимального количества ненулевых компонент.

Определение 3.8. Семантическим ядром запроса q у пользователя u будем называть вектор $S_q = C_u + V_q$.

Выше мы предположили, что множество понятий N_q для запроса q уже выделено, но не описали процесс выделения понятий из запроса. Далее мы исходим из предположения о том, что понятия, выделяемые из запроса, зависят как от запроса, так и от контекста пользователя. Для выделения понятий N_q из запроса q можно воспользоваться алгоритмом выделения понятия абзаца из раздела 3.2. В этом случае вместо вектора абзаца V_p нужно использовать семантическое ядро запроса S_q.

4 Уточнение модели поиска

Использование тезауруса позволяет привязывать понятия как к текстам документов, так и к поисковым запросам. Для этого необходимо посчитать контексты понятий, используя большой массив данных. Далее мы уточним предложенную ранее модель поиска, определив функции схожести поискового запроса, L-тега и семантики L-тега в контексте документа. Будет продемонстрировано, как выделенные понятия и их связи могут быть использованы для семантического поиска.

4.1 Расчёт функции семантики

В качестве L-тегов рассмотрим абзацы документов. Пусть даны конечное множество документов D, где каждому документу d соответствует набор его абзацев P_d, и множество понятий N, для которых предварительно рассчитаны контексты. Задача заключается в вычислении оценки функции семантики $sem(d,p)$ для документа $d \in D$ и абзаца $p \in P_d$, где из абзаца p выделено множество понятий N_p, у каждого понятия $n \in N_p$ есть контекст C_n.

Определение 4.1. Контекстом абзаца p будем называть вектор $C_p = (\sum_{n \in N_p} C_n) / |N_p|$.

Определение 4.2. Контекстом документа d будем называть вектор $C_d = (\sum_{p \in P_d} C_p) / |P_d|$.

Исходя из предположения, что в абзаце выделены все значимые понятия, можно считать, что контекст абзаца C_p характеризует его смысловое значение. Смысловое значение документа определяется смысловым значением его абзацев, что характеризуется контекстом документа C_d. На основании этого может быть определена функция семантики

$$sem(d,p) = \frac{C_d \cdot C_p}{\|C_d\| \, \|C_p\|}.$$

4.2 Расчет функции схожести

Пусть дано множество понятий N. Между этими понятиями существуют родовидовые связи. Функцию близости двух понятий n_1 и n_2 в иерархии будем обозначать $\rho(n_1, n_2)$. В дальнейшем будем считать, что эта функция задана на основе иерархии понятий в тезаурусе, используемом для поиска. Рассмотрим поисковый запрос $q \in Q$ и абзац $p \in P$. Считаем, что в запросе выделены понятия N_q, а в абзаце выделены понятия N_p.

Функция схожести L-тегов должна определять, насколько пересекается смысл, передаваемый L-тегами. Исходя из предположения, что в абзаце и запросе выделены все значимые понятия, а понятия L-тега полностью передают его смысл, можно для запроса q и абзаца p определить функцию схожести

$$sim(q,p) = \sum_{n_1 \in N_q} \frac{\max_{n_2 \in N_p} \rho(n_1,n_2)}{2|N_q|} + \sum_{n_1 \in N_p} \frac{\max_{n_2 \in N_q} \rho(n_1,n_2)}{2|N_p|}$$

4.3 Расчет релевантности

Пусть даны множество запросов Q и множество абзацев P. Рассмотрим запрос $q \in Q$ и абзац $p \in P$. Для расчета релевантности необходимо учитывать:

- $sem(d,p)$ – функция семантики.
- $sim(p,q)$ – функция схожести.

Сначала с помощью функции семантики отбираются похожие на запрос q абзацы P_q. Далее набор P_q сортируется на основе значений функции семантики и функции схожести. Релевантность должна быть больше, если значение функции семантики или схожести больше.

Функция семантики и функция схожести могут быть неравномерно распределены. В этом случае абзацы, которые больше похожи на свои документы, могут получить необоснованное преимущество перед другими абзацами. Чтобы неравномерность функции семантики не приводила к сильному изменению сортировки, можно воспользоваться следующим подходом:

- сортируем P_q по значениям функции схожести, для каждого $p \in P_q$ получаем порядковый номер в отсортированном наборе $simOrder(q,p)$;
- сортируем P_q по значениям функции семантики, для каждого $p \in P_q$ получаем порядковый номер в отсортированном наборе $semOrder(d,p)$;
- релевантность может быть оценена как сумма или произведение $simOrder(q,p)$ и $semOrder(d,p)$.

5 Применение

Рассмотрим поисковый запрос “Java”. О чем пользователь думал, когда задавал этот запрос? Он мог думать о следующем:

- Java – это язык программирования.
- Java – это остров.

- Java – это кофе.

Очевидно, что без использования истории запроса невозможно догадаться о значении термина “Java”, поэтому история запросов является важным компонентом.

Допустим, в истории часто встречается программирование, поэтому к запросу можно привязать понятие «Java – это язык программирования». Пусть в некотором абзаце встречается термин “Java”, если в этом абзаце также встречаются компьютерные термины, то к абзацу на этапе индексирования будет привязано понятие «Java – это язык программирования». В этом случае мы найдем по запросу все абзацы, связанные с языком программирования Java. Полнотекстовый поиск нашел бы все упоминания термина “Java”, но многие абзацы могли бы быть нерелевантными, кроме того, абзацы, в которых нет термина “Java”, но относящиеся к языку программирования Java, не были бы найдены.

Допустим, что по запросу “Java” найдено много абзацев, и все они одинаково похожи на запрос. Как можно ранжировать такую поисковую выдачу? Для этого может быть использована функция семантики. Абзацы, которые лучше передают смысл документа, имеют большую релевантность.

Пусть к некоторым документам вручную привязан L-тег “Java” и определено значение функции семантики. В этом случае L-тег “Java” может участвовать в поиске вместе с другими L-тегами. Привязка поисковых запросов к документам вручную позволяет улучшить качество поиска в наиболее важных темах, кроме того, такой подход используется в рекламных системах.

Представленная модель позволяет вынести сложные вычисления оценки функции семантики на этап индексации, что снижает нагрузку на сервер в момент поиска. Кроме того, появляется возможность контролировать объем поискового индекса и, как следствие, нагрузку на сервер в момент выполнения поискового запроса. Это возможно за счет ограничения количества тегов по значению функции семантики.

6 Заключение

В работе представлена модель семантического поиска и продемонстрирована полезность тезаурусов типа WordNet. Дан небольшой обзор по типам тезаурусов и предложено решение некоторых проблем.

Были формализованы определения контекстов: понятия, абзаца, документа, запроса и пользователя. Были описаны алгоритмы для выделения контекстов с использованием большого корпуса текстов, наиболее полного тезауруса. Была уточнена модель семантического поиска, введенная ранее. Предложены способы оценки функций семантики и схожести с помощью различных контекстов, связей понятий из тезауруса. Была введена, но недостаточно формализована, функция близости понятий. Предполагается ее формализация в дальнейших работах. Кроме того, планируется:

- Описать особенности индексирования математических текстов.
- Рассказать о программной архитектуре, основанной на представленной модели.
- Оценить качество и быстродействие системы поиска по сравнению с другими решениями.

Благодарности

Работа выполнена при финансовой поддержке РФФИ (проект 17-07-00214).

Литература

[1] Fellbaum, C.: WordNet. Blackwell Publishing Ltd, (1998)

[2] Malakhov, D., Sidorenko, Y., Ataeva, O., Serebryakov, V.: Semantic Search in a Personal Digital Library. In: Kalinichenko L., Kuznetsov S., Manolopoulos Y. (eds). Data Analytics and Management in Data Intensive Domains. DAMDID/RCDL 2016. Communications in Computer and Information Science, 706. Springer, Cham (2017)

[3] Magnini, B., Strapparava, C.: Experiments in Word Domain Disambiguation For Parallel Texts. Proc. of the ACL-2000 Workshop on Word Senses and Multi-linguality. Association for Computational Linguistics, pp. 27-33 (2000)

[4] Miller, G.A., Fellbaum, C., Tengi, R.: WordNet. Cambridge, Princeton University (2006)

[5] Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval (1986)

[6] Лукашевич, Н.В.: Тезаурусы в задачах информационного поиска, М.: Изд-во МГУ (2011)

[7] Серебряков, В.А. Что такое семантическая цифровая библиотека In: RCDL 2014. сс. 21-25 (2014)