УДК 519.68; 681.513.7; 612.8.001.57; 007.51/.52

А. А. Перфильев, **Ф. А. Мурзин**, **Т. В. Шманина**

Институт систем информатики им. А. П. Ершова СО РАН
пр. Акад. Лаврентьева, 6, Новосибирск, 630090, Россия
E-mail: a_perfilev@mail.ru; murzin@iis.nsk.su; rain1605@yandex.ru

МЕТОДЫ СИНТАКСИЧЕСКОГО АНАЛИЗА И СОПОСТАВЛЕНИЯ КОНСТРУКЦИЙ ЕСТЕСТВЕННОГО ЯЗЫКА, ОРИЕНТИРОВАННЫЕ НА ПРИМЕНЕНИЕ В ИНФОРМАЦИОННО-ПОИСКОВЫХ СИСТЕМАХ

Работа посвящена проблеме релевантности информации искомой в сети Интернет. Предложенный метод основывается на использовании алгоритмов сравнения предложений, которые принимают во внимание схемы синтаксического анализа, создаваемые программным приложением Link Grammar Parser. Основная идея состоит в том, что синтаксические диаграммы дают примитивную структуру текста, и это позволяет выбрать фразы в тексте, похожие на те, которые имеются в поисковом запросе. На базе этих идей была разработана информационно-поисковая система (ИПС) iNetSearch. Исследования показали, что часто достаточно оставаться на уровне синтаксиса, чтобы получить хорошие результаты. Приведены результаты тестирования метода в рамках системы iNetSearch.

Ключевые слова: информационно-поисковая система, Link Grammar Parser, синтаксический анализ, семантическое дерево, релевантность.

Введение

В условиях стремительного роста объемов информационных ресурсов возникает необходимость повышения качества информационного поиска [1]. Это, в свою очередь, заставляет разработчиков поисковых систем совершенствовать алгоритмы поиска и ранжирования документов так, чтобы они были способны учитывать семантику поступающих запросов.

Многие исследователи склоняются к необходимости проведения глубокого семантического анализа текстов для создания их семантических образов, на основе которых можно проводить тонкое ранжирование документов [1]. Этот подход, несомненно, является наиболее разумным, однако требует тщательной и долгой работы над созданием подходящих инструментов для автоматической обработки текстов. В частности, может потребоваться детальное описание различных областей знаний. Поэтому имеет смысл также поиск частичных решений, одно из которых представлено в данной работе.

Основная задача состоит в том, чтобы построить алгоритмы, которые, проникая в структуру текста, смогут вывести адекватную оценку его релевантности. Важно, чтобы данная оценка выводилась на основе контекста поискового запроса и не ограничивалась только ключевыми словами, их близостью или частотой.

Описываемый в данной работе метод позволяет сопоставлять конструкции естественного языка и в ряде случаев отождествлять даже перефразированные варианты предложений, основываясь на анализе их синтаксических структур. Таким образом, мы можем сопоставить поисковый запрос и текст с целью определения релевантности текста поисковому запросу. Метод основывается на обработке и использовании диаграмм связей, создаваемых программным приложением Link Grammar Parser.

[1] Text REtrieval Conference (TREC). URL: http://trec.nist.gov/. Last updated: 03-Nov-2011.

Метапоисковая система iNetSearch

В рамках реализуемого проекта был создан поисковый робот iNetSearch, который автоматизирует работу по поиску информации в сети (рис. 1). Интерфейс максимально упрощен. Работа пользователя сводится к тому, чтобы ввести запрос программе и дождаться, когда она закончит поиск и сбор информации из сети Интернет. При завершении она предложит просмотреть результаты поиска.

Особенности ИПС: 1) система находится на стороне пользователя, требует подключения к сети Интернет; 2) использует результаты запросов к существующим поисковым системам (например, для тестирования использовался поисковый сервис nigma.ru, так как эта система переправляет запрос другим поисковым системам, тем самым увеличивая возможный круг поиска); 3) реализованная система корректирует результаты поиска и уточняет их.

Система просматривает текстовое содержимое интернет-страничек, полученных из стандартного поискового сервиса (например, из nigma.ru), как базу для анализа. Если источник не содержит текста, соответствующего определенным критериям, он отбрасывается.

Процесс закачки интернет-страниц предполагает ряд действий.

1. Пополнение запроса пользователя при помощи словарей синонимов, гиперонимов (возможно и гипонимов). Отправка запроса поисковой системе, разбор полученного гипертекста, пополнение списка ссылок.
2. Загрузка содержимого интернет-страниц из списка.
3. Просмотр гипертекста, поиск и сбор ссылок.
4. Сбор информации, удовлетворяющей запросу пользователя.

ИПС может работать в нескольких режимах: целенаправленная загрузка списка введенных интернет-адресов, поиск информации, соответствующей запросу; отправка запроса поисковой системе, получение списка интернет-адресов, просмотр этого списка, поиск информации, соответствующей запросу; рекурсивный просмотр каталога, просмотр файлов, поиск информации, соответствующей запросу; целенаправленная загрузка с сайта файлов указанных типов.

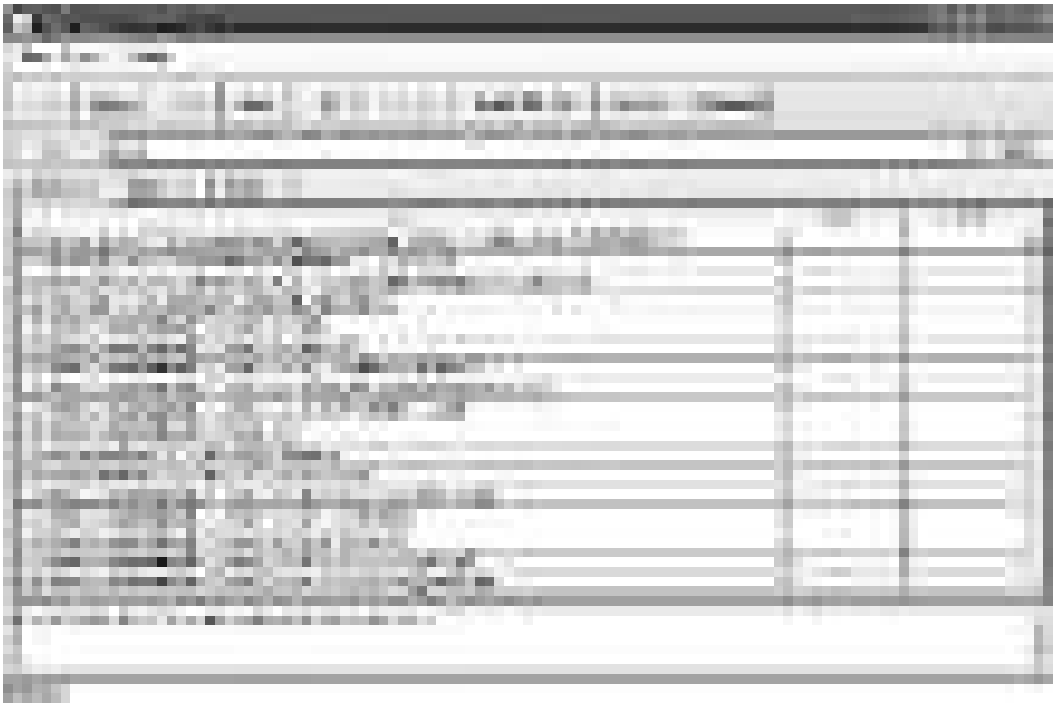

Рис. 1. Главное рабочее окно поисковой системы iNetSearch

Поисковое ядро системы iNetSearch

От пользователя в систему поступает текстовый запрос, где из него выделяются ключевые слова и термины (рис. 2). Пополнение запроса с помощью синонимов и гипонимов (слов с более узким значением) дает более широкий круг поиска слов. При недостаточном количестве найденных документов поисковый запрос повторяется с привлечением гиперонимов (слов с более общим смыслом, например, собака – зверь). Очевидно, что это значительно расширяет область поиска.

Базой поиска в системе iNetSearch служит текстовое содержимое интеренет-страниц, которые приходят от встроенного менеджера загрузок. Далее текстовые образцы поступают в систему первичных фильтров, где проходит первичная оценка релевантности текста. Наличие соответствующих ключевых слов говорит о возможной релевантности рассматриваемых образцов. Первичные фильтры ограничивают работу высокопроизводительных алгоритмов синтаксического анализатора, что значительно способствует ускорению работы, а именно: если количество совпадающих ключевых слов (с учетом синонимов и т. д.) в поисковом запросе и в анализируемой фразе текста мало, то данная фраза не передается синтаксическому анализатору и происходит переход к следующей фразе.

Рис. 2. Схема поискового ядра системы iNetSearch

Получаемые на входе предложения транслируются в синтаксические диаграммы. Транслятор проводит лемматизацию слов, приписывание метаинформации словам. Проводится добавление синтаксических связей между словами, типизация этих связей.

Происходит приписывание зависимостей между придаточными предложениями. Совокупность этих особенностей дает достаточную информацию о предложении. Синтаксический анализатор генерирует диаграммы синтаксического разбора, которые используются в системе. Они отображают синтаксическую взаимосвязь между словами.

Link Grammar Parser

Несколько слов о синтаксическом анализаторе, который был использован в системе. Link Grammar Parser (или Link) – это синтаксический анализатор английского языка, разработанный в 1990-е гг. в университете Карнеги-Меллона, США. Он базируется на использовании грамматических связей из некоторой неклассической теории синтаксиса английского языка [2].

Получив предложение, Link приписывает ему синтаксическую структуру, которая состоит из множества помеченных связей, соединяющих пары слов. Пометка каждой связи соответствует некоторому случаю правильного употребления данной пары слов в предложении. Например, пометка *S* соответствует связи между субъектом и предикатом, *O* – между объектом и предикатом. Кроме того, пометка может иметь составной нижний индекс, который необходим для проверки грамматического согласования и контроля сочетаемости слов. Дополнительно система приписывает словам предложения значения их базовых классов. Например, существительные получают подпись «.n», глаголы – «.v» и т. д.

Link реализован на языке Си для Unix и Windows, имеет открытый код, распространяется по лицензии, совместимой с GNU GPL. Синтаксический анализатор имеет словарь, вклю-

чающий около 60 000 словарных форм. Он охватывает огромную долю синтаксических конструкций, включая многочисленные редкие выражения и идиомы. Link довольно устойчив; он может пропустить часть предложения, которую не может понять, и определить некоторую структуру оставшейся части предложения. Он способен обработать неизвестный лексикон и делать разумные предположения из контекста и написания о синтаксической категории неизвестных слов. У него есть данные насчет различных названий, числовых выражений и разнообразных знаков препинания. Внутри синтаксический анализатор использует методы динамического программирования для сопоставления связей между словами [3; 4].

Ниже приведен пример разбора предложения «the fox ate the rabbit»:

```
 +-----Os----+
 +-Ds-+--Ss-+ +--D*u-+
 | | | | |
the fox.n ate.v the rabbit.p
```

Получаемые диаграммы, по сути, являются аналогом так называемых деревьев подчинения предложений. В деревьях подчинения от более главного слова в предложении можно задать вопрос к более второстепенному.

Базовый алгоритм отождествления

Предположим, что у нас есть дерево разбора. Дальше происходит обобщение таких деревьев. На этом этапе происходит нормализация словоформ. Могут быть произведены преобразования предложений. Например, обратный порядок слов заменяется на прямой. Сложные формы глаголов «обрезаются». Глаголы переводятся в одну нормализованную форму в Present Simple. В результате получается основа дерева, в котором остались только ключевые слова, несущие семантическую информацию. Такие деревья проходят процесс сравнения с диаграммой запроса пользователя (рис. 3).

Фильтрация диаграмм. Перед сличением слова проходят простой фильтр на словоформу. Очевидно, что нельзя считать глагол и существительное одинаковым словом. Само сличение или наложение слов происходит просто: проверяются гипотезы на соответствие двух слов по набору правил, если все правила проверены и соответствия не выявлены, то слова считаются далекими по смыслу. Набор правил представляет собой условия, при которых мы можем считать слова близкими. Туда входят такие правила, как прямое равенство слов, совпадение с точностью до лексической основы, синонимическая близость слов, наличие отношения «гипоним-гипероним», слова с опечатками, перестановками и прочие возможные близости между словами [5].

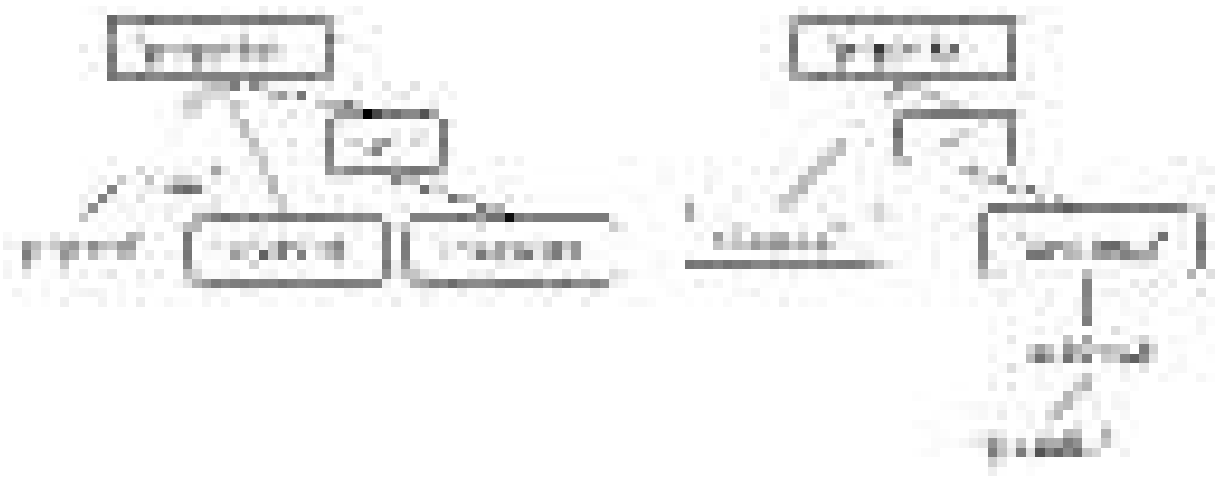

Рис. 3. Пример наложения двух деревьев

Алгоритм сличения выражений напоминает работу конечного автомата, работающего на узлах дерева. По словосочетанию строится конечный автомат. Если в тексте находятся словосочетания, удовлетворяющие условиям, что зависит от набора используемых приемов, то автомат переводит свои головки в очередные состояния, туда, где условия удовлетворены. Если хоть одна фраза привела автомат в конечное состояние, значит, это выражение из набора, и оно релевантно. Количество головок автомата зависит от набора проверяемых схем. Использование автоматов существенно ускоряет обработку выражений. В частности, в сочетании с быстрым синтаксическим анализатором это делает поиск текста очень быстрым, учитывая, что большая часть предложений фильтруется на начальном этапе.

Таким образом, приведенная степень оценок позволяет ввести определенную меру близости на предложениях. Она учитывает связь между словами, поиск по словосочетаниям, а также связность слов.

Предложения, прошедшие последний фильтр, считаются релевантными и выдаются пользователю. Для завершения своей работы система iNetSearch формирует аннотацию из найденного релевантного текста.

Алгоритмы для отождествления перефразированных предложений

Элементы математической модели. Для того чтобы сопоставлять конструкции естественного языка и отождествлять перефразированные варианты предложений, основываясь на анализе их синтаксических структур, было необходимо произвести развитие некоторой теории и соответствующих методов.

Пусть L – множество слов естественного языка, представленного в словарях и в текстах. На L задается функция $x' = \mathrm{Norm}(x),\ \ x, x' \in L,$ сопоставляющая слову его нормальную форму. Например, произвольному существительному сопоставляется его форма в именительном падеже и единственном числе. Кроме того, на L определяется множество одноместных, двуместных и трехместных предикатов и отображений. При этом истинность каждого заданного в модели предиката устанавливается при помощи соответствующих словарей и алгоритмов, а каждое отображение, определенное в модели, соответствует одному из предикатов и задается в соответствии с его семантикой.

Базовые метаслова. Пусть POS – множество частей речи; $NF \subset L$ – множество всех слов языка, находящихся в нормальной форме. На декартовом произведении $POS \times NF$ задается двуместный предикат:

$\mathrm{PartOfSpeech}(\hat{P}, x')$ истинен тогда и только тогда, когда $x' \in NF$ и $\hat{P}$ – часть речи слова x'.

Пусть теперь $\overline{x} = \{x_1, ..., x_n\}$ – произвольное предложение над L, т. е. $\forall i\ x_i \in L$. Зададим отображение $\varphi : \overline{x} \to POS \times NF$ такое, что $\varphi(x) = \left\langle \hat{P}, x' \right\rangle$, где $x \in \overline{x}$, $x' = \mathrm{Norm}(x)$, и истинен предикат $\mathrm{PartOfSpeech}(\hat{P}, x')$, сопоставляющее каждому слову предложения его часть речи и нормальную форму. Это отображение однозначно, если предположить, что всегда можно однозначно определить часть речи данного слова *x*, например, из контекста.

Пару $\left\langle \hat{P}, x' \right\rangle$ (далее обозначается $\hat{P}[x']$) назовем базовым метасловом, соответствующим слову *x*.

Кроме того, рассматриваются на L некоторые дополнительные предикаты, устанавливающие факт принадлежности данного слова $x \in L$ одной из групп вспомогательных глаголов:

1) $vaux_{11}(x)$, где $x \in \{will, 'll, may, might, should, must, can, could, would, 'd, shall\}$;
2) $vaux_{12}(x)$, где $x \in \{won't, shouldn't, mustn't, can't, couldn't, wouldn't\}$;
3) $vaux_2(x)$, где $x \in \{isn't, aren't, wasn't, weren't\}$;

4) $vaux_3(x)$, где $x \in \{hasn't, haven't, hadn't\}$;

5) $vaux_4(x)$, где $x \in \{don't, doesn't, didn't\}$.

Предикаты, ассоциированные с системой Link Grammar Parser. Связям системы Link Grammar Parser поставим в соответствие двуместные предикаты. Если Q – название некоторой связи из множества связей системы Link Grammar Parser, а x_1 и x_2 – два слова из предложения $\bar{x}$, то $Q(x_1, x_2)$ истинен на $\bar{x}$ тогда и только тогда, когда в диаграмме связей, построенной для данного предложения, между словами x_1 и x_2 будет существовать связь с пометкой Q. Например, на предложении «The cat chased a snake», разбор которого выглядит следующим образом:

```
+----Os---+
+-Ds-+---Ss--+ +-Ds-+
|    |   |   | |
the cat.n chased.v a snake.n,
```

будут истинны предикаты $Ds(the, cat)$, $Ss(cat, chased)$, $Ds(a, snake)$, $Os(chased, snake)$. Других предикатов рассматриваемого вида, истинных на данном предложении, нет.

Производные метаслова и формулы их построения. Пусть предложению $\bar{x} = \{x_1, ..., x_n\}$ соответствует набор базовых метаслов $\varphi(\bar{x}) = \{\varphi(x_1), ..., \varphi(x_n)\} = \left\{\hat{P}_1[x_1'], ..., \hat{P}_n[x_n']\right\}$; $P(\bar{x})$ – множество подмножеств слов из $\bar{x}$; $P(\varphi(\bar{x}))$ – множество подмножеств метаслов из $\varphi(\bar{x})$, а $MN = \{PredAct, PredActNo, PredPas, PredPasNo, InfAct, InfActNo, InfPas, InfPasNo, ...\}$ – множество идентификаторов типов составных слов и членов предложения.

Под составными словами и составными членами предложения (далее составными единицами предложения) подразумеваются такие единицы предложения, которые выражены несколькими словами, но при этом являются неделимыми, т. е. их нельзя разбить на части без изменения или потери смысла этих частей. Например, составными единицами предложения считаются инфинитивы и герундии, именные и глагольные сказуемые.

На декартовом произведении $MN \times P(\bar{x})$ зададим двуместный предикат, сопоставляющий составной единице предложения идентификатор ее типа:

$SentenceMember(Name, U)$ – истинен тогда и только тогда, когда $U \in P(\bar{x})$ – составная единица предложения, $Name \in MN$ – идентификатор типа составной единицы предложения U.

При этом множество идентификаторов MN определяется таким образом, что

$$\forall U \left(\begin{array}{l} SentenceMember(Name, U) \rightarrow \\ \left(\neg \exists Name' \in MN \left((Name \neq Name') \,\&\, SentenceMember(Name', U) \right) \right) \end{array} \right).$$

Зададим отображение $\psi : P(\varphi(\bar{x})) \rightarrow MN \times \left(NF^+\right)$ такое, что $\psi(\varphi(U)) = \left\langle Name, \tilde{U} \right\rangle$ для любого $U = \{u_1, ..., u_k\} \in P(\bar{x})$ такого, что истинен предикат $SentenceMember(Name, U)$ и $\tilde{U} = u_{i1}'...u_{ir}'$ – конкатенация нормальных форм слов, входящих в составную единицу предложения U (если в состав U входили вспомогательные глаголы или глаголы-связки, то они отбрасываются), $\{u_{i1}', ..., u_{ir}'\} \subset \{u_1', ..., u_k'\}$.

Пару $\left\langle Name, \tilde{U} \right\rangle$ (или $Name[\tilde{U}]$) будем называть производным метасловом, соответствующим U.

Приведем принцип построения проекции отображения ψ на множество $\{PredAct\} \times NF$, где идентификатор $PredAct$ соответствует сказуемому, выраженному глаголом в активном залоге и положительной форме.

Пусть предложению $\overline{x}$ сопоставлена диаграмма связей, порожденная Link. Тогда истинность предиката $SentenceMember(PredAct, U)$ на некотором $U \subset \overline{x}$ равносильна истинности на U следующей формулы:

$$(\exists x \in U)(\exists x' \in NF)[x' = Norm(x) \,\&\, [PartOfSpeech(Verb, x') \,\&\, (\exists y \in L)\big(S(y,x)\big) \vee$$
$$(\exists y \in U)[PartOfSpeech(Verb, x') \,\&\, Vaux_{11}(y) \,\&\, I(y,x) \,\&\, \neg N(y, "not") \vee$$
$$PartOfSpeech(PartAct, x) \,\&\, Norm(y) = "be" \,\&\, Pg(y,x) \,\&\, \neg N(y, "not") \vee$$
$$PartOfSpeech(PartAct, x) \,\&\, Vaux_{11}(y) \,\&\, I(y, "be") \,\&\, Pg("be", x) \,\&\, \neg N(y, "not") \vee$$
$$PartOfSpeech(PartPas, x) \,\&\, Norm(y) = "have" \,\&\, PP(y,x) \,\&\, \neg N(y, "not") \vee$$
$$PartOfSpeech(PartPas, x) \,\&\, Vaux_{11}(y) \,\&\, I(y, "have") \,\&\, PP("have", x) \,\&\, \neg N(y, "not") \vee$$
$$PartOfSpeech(PartAct, x) \,\&\, Norm(y) = "have" \,\&\, PP(y, "be") \,\&\, Pg("be", x) \,\&\, \neg N(y, "not") \vee$$
$$PartOfSpeech(PartAct, x) \,\&\, Vaux_{11}(y) \,\&\, I(y, "have") \,\&\, PP("have", "be") \,\&\, Pg("be", x) \,\&$$
$$\&\, \neg N(y, "not")]]]$$

Здесь задействовано несколько связей системы Link Grammar Parser:

I – соединяет глагол с инфинитивом;

PP – соединяет форму «have» с причастием прошедшего времени;

Pg – соединяет форму глагола «be» с причастием настоящего времени.

Формулы, аналогичные рассмотренной, можно поставить в соответствие каждому предикату $SentenceMember(Name, \cdot)$, где $Name \in MN$.

Предикаты семантико-синтаксических отношений и метасвязи. Далее рассматриваются отношения синтаксического подчинения между словами или составными единицами предложения (последние без ограничения общности можно также считать словами). В каждой паре слов, связанных отношением синтаксического подчинения, одно слово является главным, а второе – зависимым. Наличие отношения синтаксического подчинения между ними определяется возможностью задать вопрос от главного слова к зависимому.

Отношения синтаксического подчинения рассматриваются только между значимыми словами предложения, т. е. между словами, не относящимися к служебным частям речи или вспомогательным глаголам.

На множестве слов определяются трехместные предикаты отношений синтаксического подчинения:

$SyntacticRelation(RelationName, w_1, w_2)$ – истинен тогда и только тогда, когда w_1 и w_2 – пара значимых слов предложения, связанных отношением синтаксического подчинения типа $RelationName \in SR$, причем w_1 – главное слово в этой паре, а w_2 – зависимое.

Вычисление коэффициента степени совпадения двух предложений

При оценке степени совпадения предложения-претендента с предложением-запросом целесообразно учитывать как число совпадающих дуг в семантическом графе претендента, так и число несовпадающих.

Представим некоторую формулу вычисления степени совпадения предложений, подходящую для ранжирования претендентов и отвечающую вышеизложенным принципам:

$$y = \frac{\sum_{i=1}^{N} p_i - \left(\frac{\sum_{i=1}^{M} q_i}{\sum_{i=1}^{\tilde{M}} t_i} \right)}{\sum_{i=1}^{K} r_i},$$

где

y – коэффициент совпадения претендента предложения с запросом; K, N – число дуг в семантическом графе запроса и подграфе претендента, состоящем из совпадающих дуг; $\tilde{M}$ – общее число дуг в семантическом графе претендента; M – число несовпадающих дуг в семантическом графе претендента; $M = \tilde{M} - \sum_{i=1}^{N} N_i$, r_i, t_i – веса дуг семантических графов запроса и претендента соответственно; p_i – вес совпадающей дуги в семантическом графе претендента; q_i – вес несовпадающей дуги в семантическом графе претендента.

Простейший способ задания весов следующий. Множество связей Link-а разбивается на три непересекающихся подмножества, которые можно назвать важные связи, связи средней важности, маловажные связи. Далее им присваиваем веса, равные 3, 2, 1 соответственно. Более сложные метасвязи формируются иерархически. При этом внизу иерархии находятся элементарные связи системы Link. Индукцией по уровням можно определить вес метасвязи, как сумму весов метасвязей более низкого уровня, в нее входящих.

Таким образом, чем больше в семантическом графе претендента имеется совпадающих дуг и чем больше их веса, тем большую оценку он должен получить.

Результаты тестирования системы iNetSearch

Для демонстрации эффективности работы системы были произведены испытательные загрузки с помощью данной системы. Были сформированы десять простых запросов из области неорганической химии. По каждому запросу были загружены списки адресов с их описанием, которые поисковики обычно выдают пользователю. По этим коротким сниппетам (*snippet*) производилась оценка ресурса. Система оставляла релевантные ссылки, отбрасывая, по ее мнению, нерелевантные. В итоге на проведенных тестах в среднем из 100 ссылок, полученных из поискового сервиса nigma.ru, система выделяла 5–15 качественных релевантных ссылок, около 5 ссылок система ошибочно принимала за релевантные и остальные отбрасывала как нерелевантные, что соответствовало действительности. Это показывает, что данная система смогла произвести фильтрацию на хорошем уровне. Результаты тестирования показаны в таблице.

Результаты тестовых испытаний базовых алгоритмов системы iNetSearch

Запрос	Всего ссылок получено от поисковой системы	Количество релевантных ссылок, одобренных системой	Количество релевантных ссылок, пропущенных системой	Количество нерелевантных ссылок, одобренных системой
the burning rate of rocket fuels	99	15	8	1
using of liquid nitrogen	85	29	2	0
physical and chemical properties of zirconium	96	8	2	9
raw material for produce of medicine	121	26	7	9
use of zirconium in medicine	97	9	1	1
harmful influence of strontium on a man	102	6	0	0
molecular structure of products of disintegration of alcohol	85	20	1	12
ways of getting of glycerin	89	3	2	0
physical properties of oxides	95	17	4	8
classifying of separation techniques	107	10	0	1

Далее было проведено сравнение двух методов сопоставления конструкций естественного языка – базового (используемого в первоначальной версии системы iNetSearch) и нового (с учетом перефразирования предложений).

Запросы, перефразировки которых необходимо было найти, составлялись по разным тематикам. Источниками запросов служили:

1) коллекция научных статей более чем по 20 темам;
2) коллекция текстов общеобразовательного плана.

Для оценки качества поиска были выбраны следующие характеристики:

1) точность поиска: $Precision = \frac{|Relevant \cap Retrieved|}{|Retrieved|}$;
2) полнота поиска: $Recall = \frac{|Relevant \cap Retrieved|}{|Relevant|}$;
3) выпадение: $Fall - out = \frac{|NotRelevant \cap Retrieved|}{|NotRelevant|}$.

Здесь *Relevant* – множество документов коллекции, релевантных запросу; *NotRelevant* – множество документов, нерелевантных запросу; *Retrieved* – множество документов, одобренных системой iNetSearch.

В качестве коллекции документов рассматривалось все множество документов, полученных системой iNetSearch от поисковых систем. Ниже приведены результаты тестирования, а именно усредненные значения точности, полноты и выпадения, полученные для каждого запроса:

	Точность, %	Полнота, %	Выпадение, %
Базовый метод iNetSearch	0,520	0,875	0,576
Сопоставление семантических деревьев	0,551	0,893	0,504

Таким образом, в среднем поисковая система стала одобрять меньше нерелевантных документов и больше релевантных.

Отметим, что в таблице даны усредненные результаты. Естественно, что они зависят от выбора коллекции документов. Они лучше для тех тематик, где сложилась устоявшаяся терминология и часто используются единообразные синтаксические конструкции. Например, для таких тематик, как «обработка сигналов» или «обработка изображений». Менее хорошие результаты получаются, например, для такой тематики, как «мультиагентное моделирование», так как соответствующие тексты имеют более аморфный характер.

Заключение

Основной целью данной работы была разработка методов, позволяющих сопоставлять конструкции естественного языка и отождествлять, в том числе, перефразированные варианты предложений на основе анализа их синтаксической структуры.

В процессе решения поставленных задач были предложены способы представления семантико-синтаксических отношений между смысловыми единицами предложения, методы построения этого представления на основе диаграмм Link Grammar Parser, а также способ вычисления степени совпадения естественно-языковых конструкций. В итоге мы видим высокую эффективность предложенного подхода. С другой стороны, метод, учитывающий перефразирования, позволил улучшить работу системы iNetSearch, но, как показало тестирование, незначительно по сравнению с базовым алгоритмом.

Одной из причин является то, что возможности Link Grammar Parser на данном этапе работы почти полностью исчерпаны. И несмотря на то, что Link Grammar Parser обладает рядом преимуществ (высокая скорость работы, частичный охват семантики, обилие примеров его успешного применения в системах фильтрации текстов из сети Интернет), он вынуж-

дает оставаться на уровне синтаксиса с частичным охватом семантики. Поэтому, чтобы получить существенное продвижение, необходимо перейти на более высокий уровень [6; 7], к инженерии знаний.

Список литературы

1. *Salton G.* Automatic Information Organization and Retrieval. McGraw-Hill, 1968. 514 p.

2. *Temperley D., Sleator D., Lafferty J.* Link Grammar Documentation. 1998. URL: http://www.link.cs.cmu.edu/link/dict/index.html

3. *Батура Т. В., Мурзин Ф. А.* Машинно-ориентированные логические методы отображения семантики текста на естественном языке: Моногр. / Институт систем информатики им. А. П. Ершова СО РАН. Новосибирск, 2008. 248 с.

4. *Grinberg D., Lafferty J., Sleator D.* A Robust Parsing Algorithm for Link Grammars. Pittsburgh, 1995. (Tech. Rep. / Carnegie Mellon Univ. Computer Science; CMU-CS-95–125).

5. *Schrampfer A.* Understanding and Using English Grammar. 3rd ed. N. Y.: Pearson Education, 2002. 567 p.

6. *Nirenburg S., Raskin V.* Ontological Semantics. Cambridge, MA: MIT Press, 2004. 420 p.

7. *Thompson C.* Acquiring Word-Meaning Mappings for Natural Language Interfaces // J. of Articial Intelligence Res. 2003. Vol. 18. P. 1–44.

Материал поступил в редколлегию 17.08.2011

A. A. Perfiliev, F. A. Murzin, T. V. Shmanina

METHODS OF SYNTACTIC ANALYSIS AND COMPARISON OF CONSTRUCTIONS OF A NATURAL LANGUAGE, FOCUSED ON APPLICATION IN INFORMATION RETRIEVAL SYSTEMS

This work is dedicated to an actual problem of efficient information search in the Internet. The work is based on the algorithms of sentences comparison taking into account the schemes of syntactic analysis generated by Link Grammar Parser software. The main idea is that syntactic diagrams give us a primitive structure of a text, which allows us to select phrases in a text, which have a syntactic structure similar to that given in a request. According to these ideas, the Information Retrieval System (IRS) iNetSearch was developed. Our study showed that it is often sufficient to remain on the syntactic level and obtain rather good search results. The final part of the article represents the results of testing for the methods implemented within iNetSearch.

Keywords: Information Retrieval System, Link Grammar Parser, syntactic analysis, semantic tree, relevance.