Синтаксические и семантические модели и алгоритмы в задаче вопросно-ответного поиска

© А.А. Соловьёв

МГТУ им.Н.Э.Баумана
a-soloviev@mail.ru

Аннотация

Вопросно-ответный поиск – особый вид информационного поиска, результатом которого является не документ, а краткий и лаконичный ответ на вопрос, сформулированный на естественном языке. Рассматривается задача проверки ответов. После анализа литературы был сделан вывод: некоторые алгоритмы обработки семантических структур применимы к синтаксическим структурам, и наоборот. Планируется провести недостающие эксперименты на основе таблиц релевантности РОМИП, полученных после участия в прошлом году.

1. Введение

Вопросно-ответный поиск – это особый вид задачи информационного поиска, активно использующий методы компьютерной лингвистики. В отличие от классического поиска по ключевым словам, результатом поиска является не документ, а краткий и лаконичный фрагмент текста – ответ на вопрос, сформулированный пользователем на естественном языке[10]. Ответ ищется в коллекции документов. В качестве коллекции часто используется Интернет, обычно опосредованно через некоторую классическую поисковую систему. Предметно-специализированные системы могут использовать свою закрытую коллекцию тематических документов.

Вопросно-ответная система способна обрабатывать некоторые предопределённые классы вопросов. Наиболее успешно решается задача ответа на вопросы об определениях (*англ.: definitional*) и фактографические (*англ.: factoid*). Сегодня системы ограничиваются поиском текста ответа и не занимаются логическим выводом неявной информации.

Типичной архитектурой вопросно-ответной системы является архитектура метапоисковой системы, т.е. система надстраивается поверх классической системы поиска по ключевым словам (Рис. 1). Выделяют 4 подзадачи: анализ вопроса (*A1*), поиск фрагментов текста (*A2*), выделение ответов-кандидатов (*A3*) и проверка ответов (*A4*) [11]. На Рис. 2 изображена функциональная схема системы, построенной в рамках диссертационного исследования.

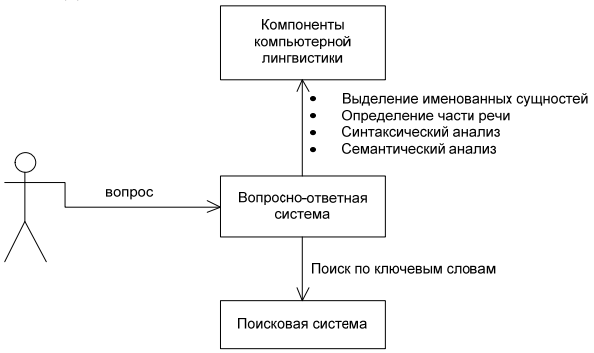

Рис. 1 Архитектура метапоисковой системы[20]

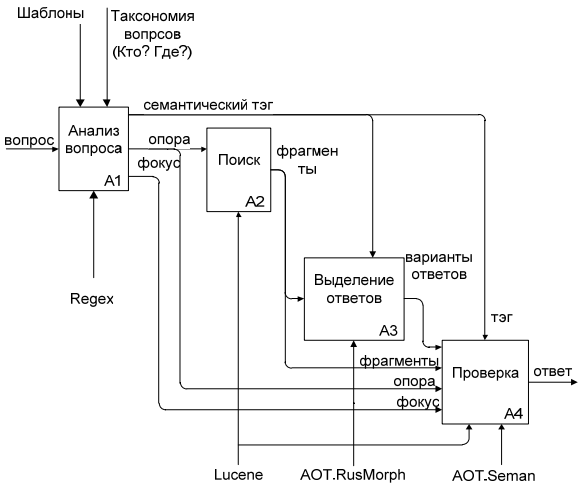

Рис. 2 Подзадачи вопросно-ответного поиска[20]

Труды 13[й] Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» - RCDL'2011, Воронеж, Россия, 2011.

Для решения лингвистических задач используются разные методы: символьные, вероятностные, основанные на правилах или больших словарях. В работе [3] подробно обсуждается вероятностный подход для решения задач *A1*, *A2*, *A3*, а в [21] рассмотрена реализация модуля анализа вопроса, разработанного в данном диссертационном исследовании.

В последующих разделах обсуждается только последняя подзадача типового конвейера – проверка ответа (*A4*).

Оценку вопросно-ответных систем выполняют на традиционных конференциях по оценке методов информационного поиска: TREC, CLEF, ROMIP – в каждой из этих конференций есть дорожка вопросно-ответного поиска. Организаторы предоставляют коллекцию документов и набор заданий – вопросов. Участники выполняют прогоны своих систем в разных конфигурациях и отправляют результаты всех заданий. Далее асессоры проверяют правильность ответов, и рассчитываются общие метрики качества для всех участников. Таким образом, экспериментально сравнивают методы, реализованные разными участниками, а также разные конфигурации системы одного участника. В [20] опубликован отчёт об участия в РОМИП 2010 системы Умба, разработанной автором. В таблице ниже перечислены несколько вопросов из заданий РОМИП.

Таблица 1 Примеры заданий РОМИП (орфография оригинальная)

№	Вопрос, жирным шрифтом выделен фокус
nqa2009_6368	**как отключить** перехват клавиатуры?
nqa2009_7185	**сколько стоит** починить гнездо у телефона сони эрикссон?
nqa2009_6425	в каких религиях **как рассматривается** карма?
nqa2009_3123	отечественная война **кто с кем** ?
nqa2009_8557	**являются ли** чердаки пожароопасными помещениями?
nqa2009_7801	**какое колличество циклов** чтения/записи предусмотренно компанией fujifilm для картриджей стандарта lto 4?
nqa2009_856	**где** собирают меганы?
nqa2009_2256	**кто** использовал стволовые клетки?

Далее статья организована следующим образом. Во втором разделе обсуждается подзадача проверки ответа, отличия в методе оценки её результатов от оценки системы в целом. В третьем разделе обсуждаются существующие синтаксические и семантические модели представления текста. В четвёртом разделе рассмотрены алгоритмы проверки ответов, работающие на моделях из третьего раздела. В пятом разделе обсуждается применимость существующих алгоритмов к синтаксической модели представления текста.

2. Задача проверки ответа

При оценке вопросно-ответной системы возникает серьёзная проблема: невозможность использовать полученные ранее таблицы релевантности в новых экспериментах. Результатом каждого задания является не просто краткий ответ, но и фрагмент текста из конкретного документа, явно подтверждающий этот ответ. Таких фрагментов может быть в коллекции много, и система может сделать вывод на основании какого-то одного из них или даже в условии избыточности [6] – нескольких разных фрагментов в разных документах, содержащих один и тот же ответ. При этом в таблице релевантности, в отличие от классического поиска, окажется не только идентификатор документа, но и фрагмент текста (сниппет) и краткий ответ из этого фрагмента. Эта запись подтверждена асессором. В следующий же раз, когда исследователь захочет измерить качество модифицированной системы (вне ежегодной кампании), он не будет иметь доступа к тем же асессорам, но будет иметь таблицу релевантности прошлого года. Однако модифицированная система может найти новый фрагмент нового документа с тем же или даже новым вариантом ответа, который не встречался в предыдущих результатах. И это не означает, что система ошиблась. Подобная ситуация (новый неоценённый ранее документ) возможна и в классическом поиске, однако в случае вопросно-ответного поиска она гораздо более вероятна.

Подзадача проверки вопроса лишена этой проблемы. Модуль проверки должен для каждого кортежа <*вопрос, документ, фрагмент, ответ*> принять решение: да или нет. В такой формулировке таблица релевантности с позитивными и негативными примерами может быть успешно использована. Примером такого подхода к оценке является семинар CLEF Answer Validation Exercise [12], организаторы которого в свою очередь заимствовал многое из более общей задачи компьютерной лингвистики Recognizing Textual Entailment [15]. Такой подход к оценке использовался в работах [9], [2] и [1].

Для оценки методов проверки вопросов мы будем использовать таблицу релевантности, построенную на основе результатов вопросно-ответной дорожки РОМИП 2010.

3. Модели представления текста в задаче валидации ответов

В основе любого метода обработки информации лежит модель представления этой информации. Рассмотрим существующие модели представления текста в задаче проверки ответов.

3.1 Набор слов

Успешная в традиционном поиске модель набора слов (*англ: bag of words*) часто применяется в базовом (*англ: baseline*) прогоне системы. Пусть Q - множество слов в вопросе, а T – множество слов во фрагменте подтверждающего текста. Тогда отношение $E = |Q \cap T| / |Q|$ может является мерой «подтверждения» ответа на вопрос Q текстом T.

3.2 Символьные шаблоны

Для проверки некоторых типов ответов можно использовать заранее подготовленные регулярные выражения. Например, для вопроса «*В каком городе родился X*», хорошим шаблоном подтверждающего текста может быть «*X родился в городе A*», где A – ответ.

3.3 Дерево синтаксического разбора граматики составляющих

Следующий естественной структурой представления текста является его дерево синтаксического разбора. Для русского языка синтаксический разбор предложения можно выполнить, например, с помощью библиотеки AOT (см. Рис. 3).

Рис. 3 Синтаксический разбор предложения, выполненный библиотекой AOT [17]

3.4 Дерево грамматических зависимостей

Другой вид представления синтаксической структуры предложения предлагает грамматика зависимостей (*англ.: dependency grammars* [4]). В работе [9] используется дерево грамматических зависимостей (*англ.: dependency tree*). На Рис. 4 представлены синтаксические деревья для вопроса и двух утверждений.

Отметим, существуют простой алгоритм для построения дерева грамматических зависимостей по дереву синтаксического разбора предложения: для английского языка [4], для русского языка в [19] (досемантический анализ). Такое преобразование не считается переходом от синтаксического к семантическому уровню представления. В [9] дерево тоже строится по выходу Collins' Parser.

Для русского языка грамматические зависимости могут быть построены с помощью системы RCO [11]. Например, результат анализа предложения "Отдел новостроек желает арендовать у нашего комбината малую строительную и погрузочную технику" описывается такой структурой:

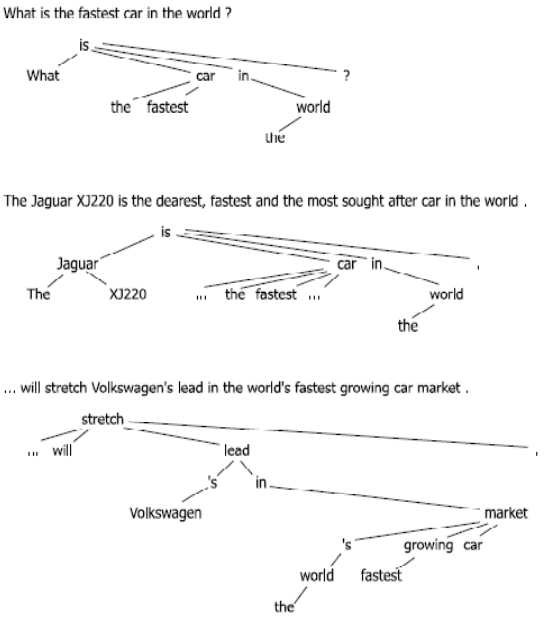

Рис. 4 Деревья грамматических зависимостей

Отдел (существительное):
- *генитивное отношение с "новостроек";*
- *Новостроек (существительное);*

Желает (глагол):
- *предикативное отношение с "отдел" в роли "Субъект действия";*
- *предикативное отношение второго порядка с "арендовать";*

Арендовать (глагол):
- *предикативное отношение с "комбината" в роли "Источник";*
- *предикативное отношение с "технику" в роли "Объект действия";*
- *Нашего (местоименное прилагательное);*

Комбината (существительное):
- *Атрибутивное отношение с "нашего";*
- *Малую (прилагательное);*
- *Строительную (прилагательное);*
- *Погрузочную (прилагательное);*

Технику (существительное):
- *Атрибутивное отношение с однородным членом "погрузочную";*
- *Атрибутивное отношение с однородным членом "строительную";*
- *Атрибутивное отношение с "малую";*

3.5 Разбор на основе грамматики связей

Третьей популярной формой представления синтаксический отношений в предложении является грамматика связей (*англ.: link grammar*) [13]. Грамматика связей состоит из слов, которые имеют ограничения по связям. Последовательность слов является предложением языка если:

1. Связи между собой не пересекаются (связи рисуются графически над словами).
2. Отсутствуют изолированные слова или несвязанные группы слов.
3. Выполнены все ограничения на связи для каждого слова.

В работе [18] предложена грамматика связей для русского языка. Сообщается, что скорость работы синтаксического анализатора крайне мала – одно предложение в секунду при потреблении памяти 200 Мб. Однако при неограниченном объёме ОЗУ автор допускает возможность разбора 100 предложений в секунду. На Рис. 5 изображён пример разбора предложения русского языка.

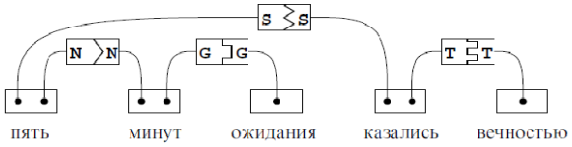

Рис. 5 Грамматика связей для русского языка [18]

Отметим, что результатом разбора является неориентированный граф с возможными циклами, а не дерево, как в случае грамматики зависимостей.

3.6 Логические формы

Логические формы (*англ.: Logic Forms)* используются во многих вопросно-ответных системах. Такое представление позволяет применять математических аппарат лямбда-выражений и автоматическое доказательство теорем.

Простейшим примером являются тернарные выражения в системе STAR [5] (*T-Expressions*). Выражения имели вид <субъект отношение объект>, где каждый элемент замещается некоторой лексемой. Сами выражения транзитивны:

Wilson presented Joe with a gift.
<<Wilson present Joe> with gift>
Wilson presented a gift to Joe.
<<Wilson present gift> to Joe>

Используя ряд лексических правил можно выводить эквивалентность этих выражений.

Рассмотрим более сложную форму логических форм из работы [8].

Текст: *Bin Laden reportedly sent representatives to Afghanistan opium farmers to buy large amounts of opium, probably to raise funds for al-Qaeda.*

Логическая форма: *Bin_NN(x14) & Laden_NN(x15) & nn_NNC(x16, x14, x15) & reportedly_RB(e2) & send_VB(e2, x16, x17) & representative_NN(x17) & to_TO(e2, x21) & Afghanistan_NN(x18) & opium_NN(x19) & farmer_NN(x20) & nn_NNC(x21, x19, x20) & buy_VB(e3, x17, x22) & large_JJ(x22) & amount_NN(x22) & of_IN(x22, x23) & opium_NN(x23) & probably_RB(e4) & raise_VB(e4, x22, x24) & funds_NN(x24) & for_IN(x24, x27) & al_NN(x25) & Qaeda_NN(x26) & nn_NNC(x27, x25, x26).*

Используя в качестве правил вывода аксиомы, построенные на базе лексического словаря WordNet, можно доказывать выводимость утверждения вопрос-ответ из текста.

3.7 Семантические узлы и отношения

Существует много моделей семантических отношений, но все они с точки зрения математического аппарата очень похожи друг на друга. В отличие от логических форм, существует ограниченный набор отношений, возможных между словами. Поиск этих отношений неразрывно связан с разрешением смысловой неоднозначности.

Англоязычной литературе эта техника называется *Semantic Role Labeling* [4]. Так, среди семантических отношений (ролей), используемых в [14], есть TARGET, ARG1, ARGM_LOC, ARGM_TMP. Пример из того же источника:

The CMU campus at the US west coast was founded in the year 2002.

TARGET: founded
ARG1: The CMU campus
ARGM_LOC: at the US west coast
ARGM_TMP: in the year 2002

Для русского языка аналогичный разбор выполняется системой АОТ[17]. Легко заметить, что набор отношений легко визуализировать в виде семантического графа. На Рис. 6 изображён граф семантических отношений для предложения из коллекции РОМИП «*Ученые использовали мезенхимные стволовые клетки, извлеченные из образцов костного мозга мужчин-добровольцев.*»

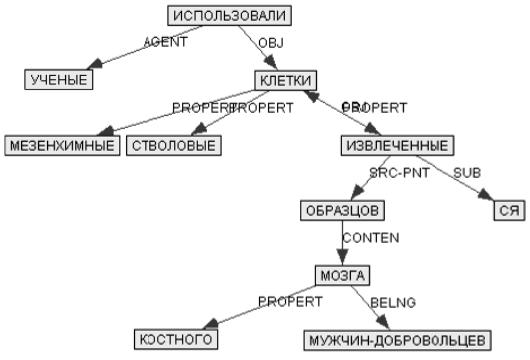

Рис. 6 Граф семантических отношений, построенный системой АОТ [19]

Следует отметить, что для одного языка существует модели семантических отношений, разработанные разными учёными-лингвистами (Н.Н.Леонтьевой в [17] и Г.А.Золотовой в [22]).

4. Алгоритмы сравнения семантических структур вопроса и подтверждающего текста

Рассмотрим существующие алгоритмы, используемые для проверки ответа в вопросно-ответной системе, на основе вычисления схожести структуры вопроса от текста, из которого извлекается ответ.

4.1 Подсчёт пересечения множеств отношений

По аналогии с формулой из раздела 3.1 многие исследователи рассматривают текст как множество несвязанных отношений – будь то представление в логической форме, синтаксический разбор или семантические роли. Рассмотрим граф на Рис. 6. Каждая дуга графа – семантическое отношение между соседними узлами *R(N1,N2)*, или записывая в виде кортежа *<N1,R,N2>*. Пусть Q – множество таких кортежей-отношений в вопросе, а A – множество кортежей-отношений в ответе. Воспользовавшись той же формулой из 3.1 получаем $E = |Q \cap T| / |Q|$.

Из двух разных фрагментов текста *A1* и *A2*, содержащих ответ (м.б. два разных ответа) на один вопрос *Q*, более правдоподобным ответом будет считаться тот, у которого мера *E* больше. Модуль валидации ответа, построенный на этой формуле, может либо выбирать из кандидатов ответ с наибольшим числом *E*, либо установить некоторое пороговое значение $E_{пор}$ для признания ответа верным. В работе [16] этот метод используется в качестве «запасной стратегии» – в случае, когда более сложный алгоритм применить не удаётся по тем или иным причинам.

Заметим, что одну и ту же формулу можно применять для четырёх моделей представления текста из раздела 3: набора слов, грамматических зависимостей в предложении, семантических отношений и даже логических форм. Так, в той же работе [16] используются две запасные стратегии, обе на основе формулы выше: первая запасная стратегия использует наборы семантических триплетов, а вторая – наборы слов.

4.2 Сопоставление предикатов

В работе [14] используется усложнённая модификация формулы из предыдущего раздела – *Predicate Matching*. Рассматриваются не триплеты из двух вершин графа, а все отношения семантического узла (т.н. предикативные отношения) во главе с глаголом. Сравнивается предикат вопроса (со всеми зависимыми аргументами) с предикатом в тексте ответа. На основе словаря WordNet и формулы Жаккарта вводится мера схожести двух термов:

$$Sim_{Term}(t_1, t_2) := J(W_1, W_2) = \frac{|W_1 \cap W_2|}{|W_1 \cup W_2|},$$ где W_1 и W_2 – множества контекстных слов из описания значения терма в словаре WordNet. Схожесть же всех аргументов предиката вопроса p_q и предиката ответа p_a вычисляется следующим образом:

$$Sim_{Args}(p_a, p_q) := \frac{\sum_{t_a \in T_a} \max_{t_q \in T_q} \left(Sim_{ExpTerm}(t_a, t_q) \right)}{|T_q| + \left| \left\{ t_a \in T_a \middle| \max_{t_q \in T_q} \left(Sim_{ExpTerm}(t_a, t_q) \right) = 0 \right\} \right|}$$

А мера схожести всего предиката определяется как произведение схожести аргументов, вычисленной выше, и схожести глаголов:

$$Sim_{Pred} = Sim_{Verb} \text{Ч} Sim_{Arg}$$ 瞅

Данный алгоритм позволяет нестрогое совпадение слов, семантически схожих на основании лексической онтологии WordNet.

4.3 Расстояние редактирования для дерева

В работе [9] рассматривается задача вычисления схожести деревьев грамматических зависимостей между словами двух предложений: вопросительного и повествовательного. В отличие от формул выше, деревья сравниваются в целом, а не в контексте отдельных отношений/предикатов. Авторы [9] применили естественную метрику схожести деревьев: минимальное число операций редактирования, необходимых для трансформации одного графа в другой.

Доступные операции редактирования: удаление вершины, вставка, замена – представлены на Рис. 7.

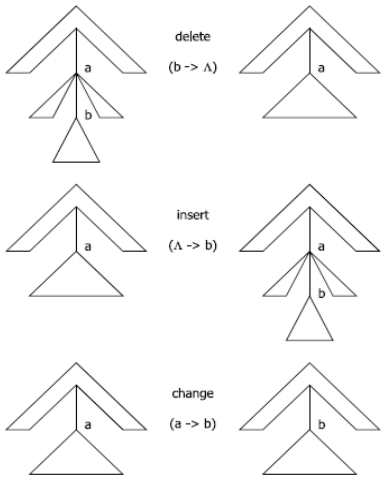

Рис. 7 Элементарные операции редактирования дерева [9]

Разным операциям приписан разный вес:

1. delete:

   if a is a stop word, $\gamma(a \to \Lambda) - 5$,

   else $\gamma(a \to \Lambda) = 200$.

2. insert:

   if a is a stop word, $\gamma(\Lambda \to a) = 200$,

   else $\gamma(\Lambda \to a) = 5$.

3. change:

   if a is *ANS*,

   if b matches the expected answer type, $\gamma(a \to b) = 5$,

   else $\gamma(a \to b) = 200$,

   else

   if word a is identical to word b, $\gamma(a \to b) = 0$,

   else if a and b have the same lemma form, $\gamma(a \to b) = 1$,

   else $\gamma(a \to b) = 200$.

Также алгоритм поиска предписания редактирования модифицирован таким образом, чтобы удаление лишних поддеревьев в подтверждающем тексте не штрафовалось, т.к. текст с ответом почти всегда содержит дополнительные грамматические конструкции, не относящиеся к вопросу.

$$DR(T_q, T_a) = \min_{F \in F(T_a)} д(T_q, T_a \backslash F)$$

$$д(T_q, T_a \backslash F) = \min_{S} \{г(S) | S(T_q) = T_a \backslash F\}$$

Где *F(T)* – множество всех возможных поддеревьев, а S множество всех возможных последовательностей операций редактирования г.

Отметим, что оригинальный алгоритм работает с синтаксическим представлением текста (деревом грамматических зависимостей). Однако его можно перенести и на семантическое представление, применяя меры схожести термов, описанные в 4.2.

4.4 Совмещение деревьев зависимостей

В работе [7] предлагается сравнивать два дерева зависимостей в задаче лексического вывода, используя алгоритмы, применяемые для обработки параллельных двуязычных текстов. Для двух деревьев *D* и *D'* строится матрица соответствия элементов *M* размера *NxN'*, где *N* – число элементов в дереве *D*, а *N'* – число элементов в дереве *D'*. Каждый элемент *S(v,v')* в матрице вычисляется алгоритмом динамического программирования, используя следующие рекурсивные формулы:

$$S(v, v') = max \begin{pmatrix} \textsc{TreeMatch}(v, v') \\ max_i\ S(v_i, v') \\ max_j\ S(v, v'_j) - SP \end{pmatrix}$$

$$\textsc{TreeMatch}(v, v') = PW \cdot \textsc{ParentMatch}(v, v') + (1 - PW) \cdot \textsc{ChildMatch}(v, v')$$

$$\textsc{ChildMatch}(v, v') = \max_{p \in \mathcal{P}(v,v')} \left[\sum_{(i,j) \in p} \frac{|v'_j|}{|v'|} \cdot S(v_i, v'_j) \right]$$

$$\textsc{ParentMatch}(v, v') = \begin{cases} 1 & \text{if } word(v) = word(v') \\ 1 & \text{if } lemma(v) = lemma(v') \\ 1 & \text{if } synonym(v, v') \\ 1 & \text{if } hypernym(v, v') \\ sim(v, v') & \text{if } sim(v, v') > 0.1 \\ 0 & otherwise \end{cases}$$

Такая матрица *M* несомненно полезна для сопоставления параллельных переводов текстов. Однако для задачи проверки лексической выводимости авторы предложили использовать в качестве меры схожести двух текстов один единственный элемент этой матрицы – лучшее найденное соответствие корневой вершины гипотезы (обычно глагол).

4.5 Неточное совпадение поиском вглубину

В диссертационной работе предложен оригинальный метод неточного сравнения семантических графов поиском в глубину[20]. Предложенный метод основан на вычислении схожести семантических графов вопроса и фрагмента, содержащего ответ. Для вопроса и фрагмента строятся семантические графы, с использование библиотеки AOT[17].

Рассмотрим пример семантического графа для вопроса nqa2009_2256 «*кто использовал стволовые клетки?*» и фрагмента из документа 419883 «*Ученые использовали мезенхимные стволовые клетки, извлеченные из образцов костного мозга мужчин-добровольцев*» (Рис. 8). Граф построен библиотекой AOT.Seman.

В основе метода лежит интуиция, что если у простого вопроса «кто?» или «где?» заменить вопросительное слово (фокус) кратким ответом, мы получим семантически верное утверждение. Мы не рассматриваем проблему синтаксической корректности полученного предложения. На Рис. 8 подграф УЧЕНЫЕ-ИСПОЛЬЗОВАЛИ-КЛЕТКИ-СТВОЛОВЫЕ во фрагменте очевидным образом соответствует графу вопроса КТО-ИСПОЛЬЗОВАЛИ-КЛЕТКИ-СТВОЛОВЫЕ, если заменить КТО на УЧЕНЫЕ. Любой строгий алгоритм поиска изоморфизма подграфов обнаружит это равенство подграфов.

Однако, более часты случаи с менее строгим совпадением подграфов. Например, вопрос nqa2009_856: «*где собирают меганы?*» и фрагмент из документа 477114: «*Может это от части потому, что часть Сцеников, как и Меганов, собиралась в Турции*» (Рис. 9).

Здесь присутствуют узлы-связки однородных членов. Стоит заметить, что дерево фрагмента в данном примере также содержит ошибку: алгоритм неправильно обработал оборот «как и».

Алгоритм вычисления меры схожести подграфов выглядит следующим образом:

1. Найти вершину с фокусом в вопросе.
2. Найти вершину с ответом во фрагменте.
3. Выполняя операции, аналогичные поиску в глубину, продвигаемся одновременно по обоим графам от исходных вершин по рёбрам и вершинам с совпадающими метками (метка из графа вопроса должна совпадать с меткой из графа фрагмента).
4. При каждом совпадении ребра/вершины суммируем в общий накопитель баллы совпадения:
   - 4.1. Совпадение рёбер.
     - 4.1.1. Рёбрам разного типа можно присваивать свой вес. Интуитивно, метки AUTHOR, LOK, NAME AGENT должны иметь больший вес, но в окончательных прогонах использовался вес 1 для всех этих типов рёбер.
     - 4.1.2. Некоторые рёбра и вершины разрешается «сокращать»: пропускать при продвижении в глубину в одном графе, не продвигаясь в другом. Например: ACT, F-ACT, S-ACT, MUA.
   - 4.2. Совпадение вершин:
     - 4.2.1. Точное посимвольное совпадение слов – 1 балл
     - 4.2.2. Совпадение лемм – 0.5 балла.
     - 4.2.3. Лемма одной вершины входит в лемму другую как подстрока – 0.5 балла.
5. Накопленная сумма баллов прибавляется к баллу, проставленному предыдущими фильтрами. Заметим, что никакой нормализации баллов здесь не используется, т.к. по построению, шкала схожести с заданным вопросом ограничена размерами графа вопроса и не зависит от фрагмента.

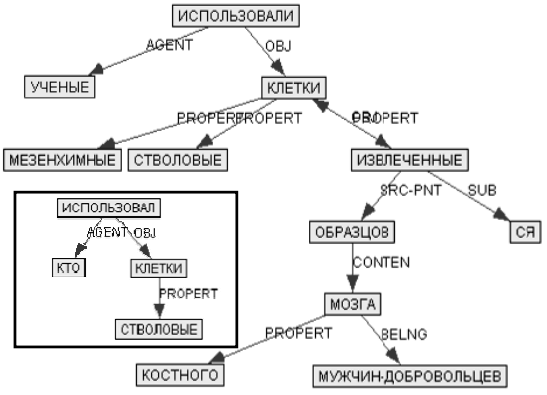

Рис. 8 Семантические графы для вопроса «*кто использовал стволовые клетки?*» и фрагмента с ответом

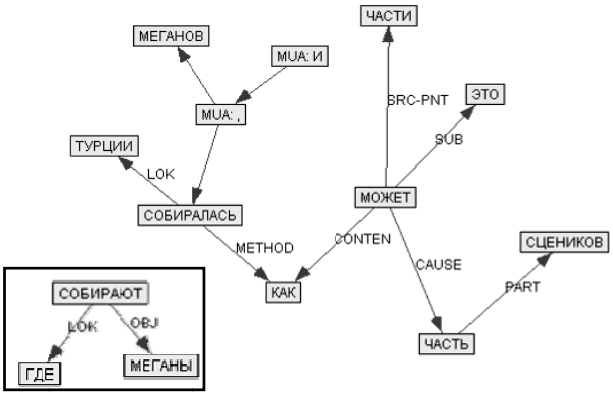

Рис. 9 Семантические графы для вопроса «*где собирают меганы?*» и фрагмента текста с ответом.

В отличие алгоритмов, рассмотренных в разделах 4.2 и 4.4, алгоритм сразу начинает работу от известных ему пар сопоставленных вершин – от слова-ответа в подтверждающем тексте и от вопросительного слова в вопросе. Алгоритм сопоставления предикатов же вынужден рассмотреть все пары предикатов (глаголов), а алгоритм сопоставления вершин вообще перебирает все возможные пары вершин. Не стоит рассматривать данное свойство как способ экономии процессорного времени. Алгоритмы 4.2 и 4.2 были заимствованы из другой прикладной задачи - машинного перевода – когда как предложенный алгоритм с самого начала основывается на специфике простых фактографических вопросов и ответов: пара вершин для старта алгоритмов сравнения уже известна, её не надо искать.

5. Подмена модели для некоторых алгоритмов

Рассмотренные выше алгоритмы в оригинальных работах использовались на одной из моделей: синтаксической (грамматика зависимостей) или семантической. В этой работе предлагается рассмотреть возможность применения алгоритма на другой модели: подменить семантические отношения синтаксическими отношениями, и наоборот. В таблице ниже приведено соответствие алгоритмов и моделей, найденное в литературе. Пустые позиции A, B, C, D, E являются областью интереса в данной работе.

Таблица 2 Соответствие алгоритмов и моделей

	Набор слов	Грамматические зависимости	Семантические отношения	Логические формы
Пересечение множеств	[16]	**A**	**[16]**	
Сопоставление предикатов		**B**	**[14]**	
Сопоставление вершин		**[7]**	**C**	
Расстояние редактирования		**[9]**	**D**	
Совпадение в глубину		**E**	**[20]**	
Автоматическое доказательство теорем				[1]

Вот некоторые общие для всех экспериментов шаги:

1. Подготовить коллекцию из нескольких десятков русскоязычных кортежей <*вопрос, фрагмент, ответ, да/нет*>. Вопросы взять из заданий РОМИП (вопросы что/где), фрагменты либо из результатов прогонов участников и из поисковой выдачи Яндекса. Выбирать как правильные, так и неправильные ответы. Ответы выделить вручную.
2. До проверки с помощью существующего модуля анализа вопросов [21] в вопросе будет выделен т.н. фокус и определён ожидаемый семантический тэг: PERSON или LOCATION.
3. Дерево синтаксических зависимостей строить в два этапа:
   3.1. Синтаксический разбор на основе грамматики составляющих.
   3.2. Построение дерева зависимостей на основе полученного разбора на составляющие (см. Досемантический анализ в [19]).
4. Граф семантических отношений выделять с помощью системы RCO Entity Extractor. К сожалению, на момент написания статьи компоненты семантического анализа АОТ уже не были доступны.
5. Во всех случаях будем игнорировать разметку графов именами зависимостей. Практически во всех работах отмечается, что использование названий зависимостей (синтаксических или семантических) только ухудшает результаты.
6. Результаты работы оценивать с помощью метрики «ошибка» – отношение числа неправильно принятых решений к общему числу решений.

Далее рассмотрим интересующие методы подробнее на примере кортежа <*«Кто использовал стволовые клетки?», «Ученые использовали мезенхимные стволовые клетки, извлеченные из образцов костного мозга мужчин-добровольцев», «Учёные», «да»*>. Модуль анализа вопроса выделит фокус «кто» и семантический тэг PERSON.

A. Пересечение множеств грамматических зависимостей

Каждая грамматическая зависимость будет представлена упорядоченной парой слов – главное и зависимое. Тогда множество зависимостей вопроса:

1. использовал->кто
2. использовал->клетки
3. клетки->стволовые

Множество грамматических зависимостей фрагмента:

1. использовали->учёные
2. использовали->клетки
3. клетки->стволовые
4. клетки->мезенхимные
5. клетки->извлечённые
6. извлечённые->из образцов
7. образцов->мозга
8. мозга->костного
9. мозга->мужчин
10. мозга->добровольцев

Зависимости «использовал->кто» и «использовали->учёные» будут признаны совпадающими т.к. а) сравниваются леммы слов, б) разрешено равенство фокуса ответу.

Используя формулу из раздела 3.1 получаем:

$$E = |Q \cap T| / |Q| = 3/3 = 1$$

Ответ будет признан верным, если E больше некоторого порогового значения *Et* (0<*Et*<1).

B. Сопоставление предикативных грамматических зависимостей

В синтаксического разбора, не выделяющего предикаты в явном виде, предикатом будем считать глагольную фразу, причастный оборот или деепричастный оборот. В случае нашего примера будут сравниваться предикат вопроса «использовал» с предикатом фрагмента текста «использовали». Аргументами предикатов будем считать все транзитивно зависимые от глагола слова. Т.е. в вопросе это будут: *кто, стволовые, клетки*, стволовые. В фрагменте: *учёные, клетки, стволовые, мезенхимные*. Слова же «*из образцов костного мозга мужчин-добровольцев*» зависят от другого предиката: *извлечённые*.

$$Sim_{Pred} = Sim_{Verb} \text{Ч} Sim_{Args} = Sim_{Term}(\text{использовал, использовали}) + Sim_{Term}(\text{клетки, клетки}) \text{Ч} \frac{+Sim_{Term}(\text{стволовые, стволовые})}{4 + |\text{учёные, меземхимные}|} = 1\text{Ч}\frac{3}{4+2} = 0{,}5$$

C. Сопоставление вершин семантических зависимостей

Следуя опубликованным в [7] результатам, будем использовать набор параметров алгоритма для дорожки RTE2.QA: *SP=0,9 PW=0,2 TH=0,6*.

Матрица сопоставления вершин будет выглядеть следующим образом:

	Кто	использовал	стволовые	клетки
Учёные	1	0,1	0	0
использовали	1	1	0	1
мезенхимные	0	0	0	0.0
стволовые	0	0	1	0,1429
клетки	0	0,1	0,1429	1
извлечённые	0	0	0	0
из образцов	0	0	0	0
костного	0	0	0	0
мозга	0	0	0	0
мужчин-добровольцев	0	0	0	0

0,1429 – это схожесть слов «stem» и «cells» на основе Wordnet (мера схожести по Lin). 0,1 – остаток после штрафа *SP* за разрешённый пропуск слова в вопросе. Матрица несимметрична, т.к. штрафы за пропуск слова в вопросе и ответе не совпадают: 0,9 и 0.

Мерой схожести будет максимальный элемент в столбце «использовал» - 1. Это значение больше с *TH*, что подтверждает выводимость гипотезы (вопроса с фокусом, заменённым на ответ) из текста.

D. Расстояние редактирования семантического графа

Дерево ответа превращается в дерево вопроса следующими операциями:

1. Удаление поддерева F_1 «мезенхимные».
2. Удаление поддерева F_2 «извлечённые из образцов костного мозга мужчин-добровольцев».
3. Замена *учёные→кто*.
4. Замена *использовали→использовал*.

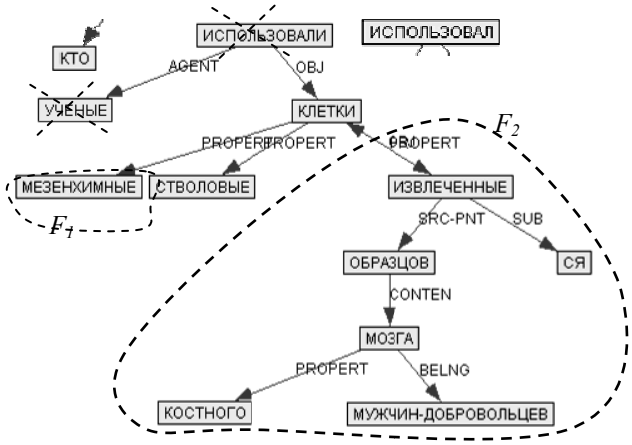

Рис. 10 Операции редактирования дерева ответа

В итоге получаем стоимость редактирования:

Для фильтрования неподходящих вопросов следует нормировать эту метрику (например, на длину вопроса) и экспериментально подобрать некоторое пороговое значение.

E. Сопоставление деревьев грамматических зависимостей в глубину

Семантический граф на Рис. 8 полностью повторяет структуру грамматических зависимостей, за исключением, м.б. словосочетания «мужчин-добровольцев», однако алгоритм игнорирует эти слова. Так что для простоты в данном примере заимствуем иллюстрацию семантического графа. Следуя алгоритму 4.5, начиная с вершин «кто» и «учёные» обходом в глубину будет найдено совпадение следующих рёбер и вершин:

1. вершина кто==учёные. 1 балл.
2. ребро. 1 балл.
3. вершина использовал ~= использовали (совпадение лемм). 0,5 балла.
4. ребро. 1 балл.
5. клетки==клетки. 1 балл.
6. ребро. 1 балл.
7. стволовые==стволовые. 1 балл.

В итоге? накопленная сумма баллов: 6,5.

Отметим, что в отличие от оригинального метода [20] в данном случае мы игнорируем подписи рёбер.

6. Заключение

Рассмотрена подзадача проверки ответов в вопросно-ответном поиске. Обзор литературы с последующей классификацией моделей и алгоритмов выявил пробелы: есть практическая возможность применить алгоритмы, оригинально реализованные для семантических структур, к синтаксическим структурам, и наоборот.

Чтобы исследовать вклад вычислительно сложного семантического анализа в задаче проверки ответа, планируется поставить 11 экспериментов (в первую очередь «набор слов» и синтаксические - A, B, E) на вопросах «кто? где?» из таблиц релевантности РОМИП 2010. Пять из них – воспроизведение экспериментов других авторов, но на русскоязычных заданиях, один – повторение нашего эксперимента РОМИП 2010 [20]. Остальные 5 экспериментов (A, B, C, D, E) проводятся впервые.

Литература

[1] Akhmatova, E. Textual Entailment Resolution via Atomic Propositions // Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment, Southampton, UK (2005) 61–64.

[2] Ferrrandez и др. Deep vs. Shallow Semantic Analysis Applied to Textual Entailment Recognition // Advances in Natural Language Processing 5th International Conference on NLP, FinTAL 2006 Turku, Finland, August 23-25, 2006 Proceedings.
[3] Ittycheriah, Abraham. A Statistical Approach for Open Domain Question Answering // Advances in Open Domain Question Answering. Springer Netherlands, 2006. Part 1. Vol.32.
[4] Jurafsky, D. & Martin, James H. Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition. – 2nd ed.: Upper Saddle River 2009.
[5] Katz, B., Borchardt, G., and Felshin, S. Natural Language Annotations for Question Answering // Proceedings of the 19th International FLAIRS Conference (FLAIRS 2006), May 2006, Melbourne Beach, FL.
[6] Magnini, B., Negri, M., Prevete, R. and Tanev, H. Is It the Right Answer? Exploiting Web Redundancy for Answer Validation // Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-2002), Philadelphia, PA
[7] Marsi, E. and Krahmer, E. and Bosma, W.E. and Theune, M. (2006) Normalized Alignment of Dependency Trees for Detecting Textual Entailment. In: Second PASCAL Recognising Textual Entailment Challenge, 10-12 April 2006, Venice, Italy.
[8] Moldovan, D., Pasca, M., Surdeanu, M. Some Advanced Features of LCC's Poweranswer // Advances in Open Domain Question Answering. Text, Speech and Language Technology, 2006, Volume 32, Part 1, 3-34.
[9] Panyakanok, V., Roth, D. and Yih, W. Natural language interface via dependency tree mapping: An application to question answering // AI and Math.- January 2004.
[10] Prager, John. Open-Domain Question-Answering // Foundation and Trends in Information Retrieval, vol 1, no 2, pp 91-231, 2006.
[11] RCO – Russian Context Optimizer. Технологии анализа и поиска текстовой информации [Электронный ресурс]. – URL: http://rco.ru/
[12] Rodrigo, Б., Pecas, A., and Verdejo, F. 2009. Overview of the answer validation exercise 2008. In Proceedings of the 9th Cross-Language Evaluation Forum Conference on Evaluating Systems For Multilingual and Multimodal information Access (Aarhus, Denmark, September 17 - 19, 2008). Lecture Notes In Computer Science. Springer-Verlag, Berlin, Heidelberg, 296-313.
[13] Sleator D. Temperley D. Parsing English with Link Grammar // Carnegie Mellon University Computer Science technical report CMU-CS-91-196, 1991.
[14] Schlaefer, Nico. A Semantic Approach to Question Answering: Saarbrьcken 2007.
[15] TAC 2011 Recognizing Textual Entailment Track (RTE-7) [Электронный ресурс]. URL: http://www.nist.gov/tac/2011/RTE/
[16] Wang, Neumann. Using Recognizing Textual Entailment as a Core Engine for Answer Validation // Working Notes for the CLEF 2008 Workshop.
[17] Автоматическая Обработка Текста [Электронный ресурс]. URL: http://aot.ru
[18] Протасов С. В. Преимущества грамматики связей для русского языка // Международная конференция Диалог 2005.
[19] Сокирко А. В. Семантические словари в автоматической обработке текста (по материалам системы ДИАЛИНГ) // Диссертация на соискание ученой степени кандидата технических наук: М. 2001.
[20] Соловьев А.А. Кто виноват и где собака зарыта? Метод валидации ответов на основе неточного сравнения семантических графов в вопросно-ответной системе. // Российский семинар по Оценке Методов Информационного Поиска. Труды РОМИП 2010: Казань, 2010.
[21] Соловьёв А.А., Пескова О.В. Построение вопросно-ответной системы для русского языка: модуль анализа вопросов // Новые информационные технологии в автоматизированных системах: материалы 13 научно-практического семинара.- Моск. гос. ин-т электроники и математики.- 2010.- с.41-49. URL: http://nps.itas.miem.edu.ru/2010/sbornik13.pdf
[22] Тихомиров И. А. Вопросно-ответный поиск в интеллектуальной поисковой системе Exactus // Российский семинар по Оценке Методов Информационного Поиска. Труды четвертого российского семинара РОМИП'2006: Спб. 2006.

Syntactic and Semantic Models and Algorithms in Question Answering

© Alexander Solovyev

Question Answering is a specific task of information retrieval, which results not in a document, but in a short neat answer to the question posed in natural language. An Answer Validation task is considered. Literature study concluded with a notice about practical applicability of some algorithms to syntactic structures despite they were originally applied to semantics, and vice versa. Running of additional experiments is planned to base on relevance tables derived after participation in the ROMIP seminar last year.