Автоматизированное пополнение шаблонов для системы извлечения информации из текста

© Д.С. Котельников[1], Н.В. Лукашевич[2]

[1]Факультет вычислительной математики и кибернетики МГУ имени М.В. Ломоносова

[2]Научно-исследовательский вычислительный центр МГУ имени М.В. Ломоносова

info@dmitriu.com, louk@mail.cir.ru

Аннотация

В работе описывается способ автоматизированного пополнения шаблонов для системы извлечения информации из текстов. В качестве источника разнообразных описаний события используются новостные кластеры. Пополнение шаблонов производится за счет нахождения в новостном кластере близких по содержанию предложений, при условии обнаружения хотя бы одном них извлекаемого события. Проведены эксперименты, в которых показана возможность автоматического обнаружения дополнительной лексической информации для отражения в шаблонах системы извлечения информации из текстов.

1 Введение

Задача извлечения информации состоит в выделении из текста на естественном языке структурированной информации. Типичными подзадачами этой задачи являются извлечение совокупности упоминаемых в тексте сущностей, отношений между ними и ситуаций, в которых они принимали участие.

Большинство существующих систем выделения фактов из текстов на русском языке основаны на так называемом инженерном подходе [2 – 4], когда шаблоны для извлечения информации описываются вручную. Характерной особенностью такого рода систем является то, что наблюдается существенная неполнота извлекаемой информации, т. е. реально упомянутая в тексте информация системой не всегда обнаруживается. Проблема связана с тем, что человеку трудно описать все возможные способы упоминания той или иной сущности или факта в предложениях текста.

Особенно сложной задачей для систем извлечения информации является задача извлечения из текста информации о некотором упомянутом факте, в котором задействовано несколько участников, что связано с тем, что возникает большая вариативность выражения одной и той же информации на естественном языке. Такую вариативность трудно полностью отразить в совокупности шаблонов системы извлечения информации.

Так, факт о получении кредита может успешно извлекаться из предложения «*Краткосрочный государственный кредит в размере $4 миллиарда получит компания Chrysler*» и не извлечься из предложения «*Chrysler получил от Минфина США кредит в 4 млрд. долларов*».

Таким образом, актуальной является задача автоматического или автоматизированного пополнения шаблонов для систем извлечения информации, основанных на инженерном подходе.

Если рассмотреть новостной кластер, объединяющий несколько тематически близких сообщений, то в нем часто оказывается достаточное количество близких по смыслу предложений, включающих как предложения, в которых некоторый факт распознан вполне успешно, так и предложения, в которых этот же факт не распознан совсем или распознан частично. Именно эту вторую группу предложений можно использовать для наращивания шаблонов для распознавания данного факта.

В работе исследуется вопрос о пополнении множества шаблонов для извлечения информации из потока новостей за счет двойной кластеризации. Сначала новостные сообщения близкой тематики объединяются в новостные кластеры, затем предложения, в которых обнаружены шаблоны для извлечения информации, служат центрами для кластеров схожих предложений, в которых такие шаблоны не обнаружены.

В качестве исходных данных используются результаты работы демонстрационной версии программы извлечения информации – RCO Fact Extractor, работа которой основана на инженерном методе составления шаблонов для извлечения информации из текстов [2, 3].

2 Обзор работ по исследуемой тематике

Предложено большое количество методов автоматического получения шаблонов, выделяющих описания ситуаций из текстов на английском языке.

Большинство из них [12, 13] использует предварительно размеченную человеком коллекцию текстов. Создание такой коллекции является трудоем-

Труды 12[й] Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» – RCDL'2010, Казань, Россия, 2010

кой задачей и для каждого извлекаемого факта приходится создавать свои коллекции.

Система AutoSlog-Ts [18, 19] использует вместо размеченной коллекции тексты, помеченные как содержащие и не содержащие выделяемый факт, а также правила составления шаблонов. В системе требуется участие человека в проверке автоматически сформированных шаблонов на извлечение определенного события.

В системах KnowItAll [11] и TextRunner [7] применяются независимые от выделяемого отношения правила, поэтому требуется присутствие в описании события фрагмента текста, полностью совпадающего с написанным человеком шаблонов. Таким образом, системе требуется искать предложения, подходящие под шаблон, что накладывает серьезные ограничения на извлекаемые факты.

В системе DIPRE [8] шаблоны для извлечения отношений между сущностями порождаются повторением двух итераций, что помогает избавиться от необходимости участия человека в процессе получения новых шаблонов. Обучение начинается с небольшого количества установленных фактов на больших коллекциях данных. Для извлечения шаблонов система использует контексты в виде трех подстрок (левый, правый и средний) около упоминаний сущностей, для которых нужно установить отношение.

Развитием данного подхода является система Snowball [6], в которой шаблоны представлены тремя векторами, отражающими левый, средний и правый контексты между выделяемыми сущностями. Так же проведена работа по вычислению качества выделяемых шаблонов и фактов.

В работе [17] шаблоны представлены вектором между сущностями, а возможные значения самих слотов ограничены семантическим тегом лемм, входящих в сущности. Для обобщения шаблонов и проверки качества фактов используются близкие по мере взаимной информации PMI [14] слов.

Недостатком методов является необходимость сбора большой коллекции документов для каждого извлекаемого отношения.

Система [15] помогает улучшить полноту извлечения некоторого факта, за счёт композиции нескольких шаблонов и точность при помощи выявления типов извлекаемых сущностей.

Нахождение близких по содержанию предложений в новостном кластере для извлечения парафраз исследуется в работе [10]. Используется расстояние Левенштейна для слов в предложении и эвристика, что новостные источники раскрывают основное содержание новости в начале текста.

3 Кластеризация предложений

Пусть в некоторой системе извлечения информации из текстов, основанной на инженерном подходе, описана совокупность шаблонов для распознавания заданного факта F.

Если такая совокупность шаблонов неполна, то это должно проявиться следующим образом. В новостном кластере, посвященном данному факту, в некоторых предложениях данный факт будет обнаруживаться системой извлечения информации, а в других (синонимичных или сходных) предложениях данный факт обнаруживаться не будет.

Таким образом, для распознавания неполноты описанных шаблонов и формирования новых шаблонов для извлечения факта F необходимо к совокупности $\{S+\}$ предложений, в которых факт F обнаружен, добавить совокупность $\{S\mathrm{sim}\}$ похожих предложений, в которых факт F не обнаружен, но которые могут потенциально содержать этот факт. Предложения, входящие в множество $\{S\mathrm{sim}\}$, ищутся среди всех предложений совокупности $\{S-\}$, в которых факт F не установлен.

Нахождение похожих предложений в данной задаче имеет ряд особенностей, которые могут быть использованы для более качественного нахождения похожих предложений. Например, можно находить не только попарное сходство предложений из множества $\{S-\}$ к множеству $\{S+\}$, но и общее сходство с множеством $\{S+\}$.

Кроме того, для нахождения множества предложений $\{S\mathrm{sim}\}$ может использоваться внутренняя структура извлекаемого факта. Известно, что при извлечении фактов обычно заполняется так называемый фрейм события, который состоит из заголовка и слотов, соответствующих участникам события. Так, информация о выданных кредитах может отражаться во фрейме со слотами: Кредитор (Creditor), Заемщик (Debtor), Сумма (Value), Цель (Purpose).

Если система извлечения фактов в одном из предложений выделила основных участников события $F_{\mathrm{o}}^{\mathrm{k}}$ для заполнения слота F_{o} в фрейме факта F, то это также может быть использовано для обнаружения предложений, потенциально содержащих данный факт.

Рассмотрим подробнее совокупность признаков, извлекаемых из предложений.

3.1 Количество найденных слотов

Обозначим через L множество лемм произвольного предложения P из $\{S-\}$, а через M – множество лемм произвольного предложения из $\{S+\}$. Тогда количество найденных слотов можно вычислить по формуле

$$\mathrm{slots}(P) = \sum_{o=1}^{n} \underset{\exists i,k}{\mathrm{match}}(L_i, F_o^{\,k}),$$

где match возвращает 1, если заполнение для слота является подстрокой леммы предложения или одинаковы числовые представления обоих строк, при равных значениях семантического атрибута и 0 в противном случае.

3.2 Мера косинуса угла

Вычисляются максимальное и среднее арифметическое значений меры косинуса угла между L и векторами M из $\{S+\}$:

$$\cos(L, M) = \frac{|L \cap M|}{|L| \cdot |M|}.$$

3.3 Мера косинуса угла для TFIDF

Вычисляются максимальное и среднее арифметическое значений меры косинуса угла между L и векторами M из $\{S+\}$, но вместо лемм предложений, используется значение TFIDF. Были проверены несколько формул [1, 9].

3.4 Вектор частотности лемм

Вычисляется частотность лемм, содержащихся в предложениях $\{S+\}$ по всем кластерам, и формируется вектор лемм с их частотностью R. Очередное предложение из множества $\{S-\}$ сопоставляется с этим вектором. В качестве признаков используются мера косинуса угла для лемм и TFIDF между вектором R и L, наличие самой частотной леммы, суммарная частотность лемм предложения:

$$\mathrm{sumfreq}(L) = \sum_{i=1}^{n} \mathrm{freqin}(L_i, R),$$

$$\mathrm{freqin}(L_i, R) = \begin{cases} \mathrm{freq}(L_i), & L_i \in R, \\ 0, & L_i \notin R. \end{cases}$$

3.5 Признаки для лемм между ролями

Из предложений множества $\{S+\}$ при формировании векторов берутся леммы, расположенные между крайними извлеченными слотами. Для вновь получившихся векторов подсчитываются все признаки, описанные в п. 3.3.

3.6 Эксперимент по отбору признаков для нахождения сходства между предложениями

Для экспериментов по выявлению лучших признаков сходства предложений и формирования множества предложений $\{S\mathrm{sim}\}$, потенциально содержащих искомый факт, была собрана коллекция из 60 новостных кластеров, в которых обсуждалось событие получения кредита. Все предложения коллекции были просмотрены экспертом и помечены как содержащие или не содержащие описание ситуации получения кредита.

Коллекция была обработана программой RCO Fact Extractor и получены предложения, в которых системе удалось установить фрейм кредита. Для нахождения близких по содержанию предложений были выделены 24 признака, позволяющие сравнивать предложения, в которых системе извлечения информации удалось обнаружить извлекаемое событие $\{S+\}$, и предложения, в которых обнаружить событие не удалось $\{S-\}$.

В табл. 1 представлены меры точности, полноты и F-меры, достигаемые отдельными признаками для нахождения похожих предложений, действительно содержащих пропущенный факт. Как видно, наилучшим признаком оказался так называемый признак косинуса угла между вектором частотности лемм по всему множеству предложений $\{S+\}$ и вектором лемм предложения L из $\{S-\}$ (см. п. 3.4).

Таблица 1. Результаты нахождения предложений, содержащих искомый факт из множества $\{S\mathrm{sim}\}$, для одиночных признаков

Признак	Точность	Полнота	F-мера
Количество слотов	0.5493	0.6801	0.6077
Максимальный косинус угла	0.4588	0.5340	0.4935
Средний косинус угла	0.4391	0.6272	0.5165
Максимальный косинус угла для TFIDF	0.4240	0.6007	0.4971
Средний косинус угла для TFIDF	0.4054	0.7858	0.5349
Косинус угла с вектором частотности	0.5594	0.9181	**0.6952**
Суммарная частотность лемм	0.7461	0.5440	0.6292
Максимальный косинус угла между ролями	0.6139	0.2103	0.3133
Средний косинус угла между ролями	0.6139	0.2002	0.3019
Максимальный косинус угла для TFIDF между ролями	0.6203	0.2078	0.3113
Средний косинус угла для TFIDF между ролями	0.6280	0.1914	0.2934
Суммарная частотность лемм между ролями	0.5906	0.2216	0.3223
Наличие частотных лемм	0.2311	0.8614	0.3645
Наличие самой частотной леммы	0.0626	1.0000	0.1179

Для комбинирования признаков были опробованы различные алгоритмы машинного обучения. В табл. 2 приведены результаты алгоритмов машинного обучения из программной системы RapidMiner [16], которые показывают, что на основе выделенных признаков удалось добиться значительного ка-

чества обнаружения предложений, содержащих пропущенный факт.

Таблица 2. Результаты нахождения предложений для методов машинного обучения

Метод	Точность	Полнота	*F*-мера
Neural Net	0.7485	0.7720	0.7600
Decision Trees	0.6212	0.9005	0.7352
k-Nearest Neighbor	0.7028	0.7286	0.7154
Naive Bayes	0.6806	0.7427	0.7102

Таблица 3. Лучшие признаки для методов машинного обучения

Метод	Точность	Полнота	*F*-мера
Neural Net	0.7485	0.7720	0.7600
Decision Trees	0.6212	0.9005	0.7352
k-Nearest Neighbor	0.7028	0.7286	0.7154
Naive Bayes	0.6806	0.7427	0.7102
Метод	Признаки		
Neural Net	1. Количество слотов 2. Максимальный косинус угла 3. Средний косинус угла 4. Максимальный косинус угла для TFIDF слов 5. Косинус угла с вектором частотности 6. Суммарная частотность лемм 7. Суммарная частотность лемм между ролями		
Decision Trees	1. Количество слотов 2. Средний косинус угла 3. Средний косинус угла для TFIDF слов 4. Косинус угла с вектором частотности слов 5. Суммарная частотность лемм		
k-Nearest Neighbor	1. Количество слотов 2. Средний косинус угла для TFIDF 3. Косинус угла с вектором частотности		
Naive Bayes	1. Суммарная частотность лемм 2. Максимальный косинус угла для TFIDF между ролями 3. Средний косинус угла для TFIDF между ролями		

В табл. 3 приведены результаты процедуры отбора признаков для каждого метода. Как видно, основные признаки, влияющие на принятие решение при классификации, связаны либо с количеством слотов фрейма, найденных в предложении, либо с суммированной характеристикой предложений из множества $\{S+\}$.

По результатам проведенного исследования была выбрана модель классификации, основанная на деревьях решений.

4 Описание работы системы автоматического построения шаблонов

Построенная в предыдущем разделе модель нахождения предложений, в которых потенциально может скрываться не обнаруженный ранее факт, используется в системе автоматического построения шаблонов, которая применяется к большим объемам новостной информации.

Архитектурно система состоит из трех компонент: сборщик новостных кластеров, кластеризатор предложений и экстрактор элементарных шаблонов. На вход системы подается тип извлекаемого факта, который должна уметь выделять используемая система извлечения информации из текста и ключевое слово, необходимое для поиска новостей. Теперь подробно опишем каждый из компонентов системы.

4.1 Сборщик новостных кластеров

В новостном архиве Google [5] производится поиск кластеров по ключевому слову, извлекаются ссылки на документы с полным описанием новости. С сайтов новостных изданий скачиваются HTML страницы, и извлекаются тексты, которые в них содержатся. Получившиеся тексты отправляются на обработку системе извлечения информации из текста, которая выделяет предложения, содержащие описания интересующего события. Таким образом, получается большое количество новостных кластеров, основной темой которых является искомый факт.

4.2 Кластеризатор предложений

Обработанные тексты отправляются на вход кластеризатора предложений, который вычисляет признаки и делает первый отбор предложений, потенциально содержаших пропущенный факт, используя обученный классификатор.

Для каждой леммы из предложений, в которых базовой системе извлечения информации, удалось установить факт наличия извлекаемого события и отобранных классификатором, вычисляется частотность. Леммы, частотность которых больше некоторого порога, считаются значимыми.

Примеры значимых слов для фрейма кредита:

0.989899	КРЕДИТ
0.392817	ПРЕДОСТАВЛЯТЬ
0.37037	ПОЛУЧАТЬ
0.255892	БАНК
0.251403	ВЫДАВАТЬ
0.249158	РОССИЯ
0.246914	БРАТЬ
0.230079	ГОДА
0.181818	РЫНОК
0.176207	НЕДВИЖИМОСТЬ
0.166105	БИЗНЕС

0.152637 КРЕДИТНЫЙ
0.14927 РОССИЙСКИЙ

Далее производится поиск значимых слов в предложениях из множества $\{S-\}$ и добавляются признаки, связанные со значимыми словами. После производится второй отбор предложений классификатором, который обучался с учетом дополнительных признаков.

В результате работы кластеризатора из обработанного множества кластеров извлекаются предложения из множества $\{S+\}$, то есть те предложения, в которых базовая система извлечения информации обнаружила искомый факт. Также извлекаются предложения из множества $\{S\text{sim}\}$, в которых факт не обнаружен, но потенциально может содержаться.

4.3 Экстрактор элементарных шаблонов

Отобранные предложения из множеств $\{S+\}$ и $\{S\text{sim}\}$ поступают на вход экстрактора шаблонов.

Шаблоны (далее элементарные шаблоны или просто шаблоны) в данном случае – это не те, возможно, сложные описания языковых конструкций, на основе которых работает базовая система извлечения информации, а некоторые служебные разбиения предложений.

Элементарные шаблоны строятся для отобранных предложений следующим образом:

- слова, соответствующие слотам целевого фрейма в кластере текущего предложения, заменяются на название этого слота (например, [Debtor], [Creditor]);
- для построения шаблона выделяется непустая подстрока лемм из исходного предложения между двумя разными слотами с добавлением значимых слов (см. п. 4.2.), до первого или после второго слота на расстоянии не более 4 слов.

Например, из предложения «Chrysler получил от Минфина США кредит в 4 млрд. долларов» будет извлечены два шаблона:

[Debtor] {ПОЛУЧАТЬ} ОТ [Creditor] {КРЕДИТ}

{ПОЛУЧАТЬ} ОТ [Creditor] {КРЕДИТ} В [Value]»,

так как леммы *кредит* и *получат*ь являются значимыми слова для описания данного факта.

Далее шаблоны обобщаются удалением прилагательных и наречий. Шаблоны, в которых не найдено ни одного значимого слова или в которых оба слота имеют одно и то же название, считаются ошибочными.

Такие элементарные шаблоны извлекаются из предложений множеств $\{S+\}$ и $\{S\text{sim}\}$. Для каждого шаблона подсчитывается коэффициент $k+$, сколько раз он был извлечен на предложениях из множества $\{S+\}$, и коэффициент $k-$, сколько раз такой шаблон был извлечен из предложений множества $\{S\text{sim}\}$.

5 Эксперименты

Эксперименты проводились на коллекции из 10000 новостных кластеров, собранных из архива Google.

В качестве базовой системы извлечения информации использовалась система RCO Fact Extractor [2, 3], и рассматривалось событие выдачи кредита.

Было выделено 1569 элементарных шаблонов. Все шаблоны, имеющие коэффициент $k+$, равный нулю, были упорядочены по мере снижения величины $k-$. Первыми в таком списке оказались следующие элементарные шаблоны:

[Слот:Debtor] ПРИВЛЕКАТЬ КРЕДИТ НА [Слот:Value]

[Слот:Debtor] ПРИВЛЕКАТЬ СИНДИЦИРОВАТЬ КРЕДИТ НА [Слот:Value]

[Слот:Creditor] ВВОДИТЬ МОРАТОРИЙ НА ВЫДАЧА КРЕДИТ [Слот:Debtor]

[Слот:Debtor] ПОЛУЧАТЬ [Слот:Value]

[Слот:Debtor] ПРИВЛЕКАТЬ КРЕДИТ ОБЪЕМ [Слот:Value]

[Слот:Debtor] ПРИВЛЕКАТЬ КРЕДИТ [Слот:Creditor]

[Слот:Debtor] ОДОБРИТЬ КРЕДИТ РЖД НА [Слот:Value]

[Слот:Debtor] ПРЕДОСТАВЛЯТЬ ПЕРВЫЙ ТРАНШ КРЕДИТ НА [Слот:Value]

[Слот:Debtor] ПРИВЛЕКАТЬ КРЕДИТ В [Слот:Value]

[Слот:Debtor] ПОЛУЧАТЬ КРЕДИТ НА СУММА ДО [Слот:Value]

[Слот:Debtor] БУДЕТ ПРЕДОСТАВЛЯТЬ КРЕДИТ НА [Слот:Value]

[Слот:Creditor] ПРЕКРАЩАТЬ ВЫДАЧА КРЕДИТ [Слот:Debtor]

Сравнение с внутренними описаниями данной ситуации в базовой системе извлечения информации показало, что верно выделены следующие проблемы текущего описания:

- не учтено, что о выделении кредита можно сказать, используя леммы «привлекать», «сообщать», «договариваться», «подписывать», «одобрить»;
- не учитывается, что у кредитов бывают транши; упоминание траншей изменяет структуру предложения и затрудняет извлечение факта.

Извлечение элементарных шаблонов показало, что во многих из них фигурируют одни и те же слова. Поэтому был сделан дополнительный лексический анализ на самые «неудачные» слова, то есть те, которые в первую очередь необходимо добавить в существующие шаблоны исходной системы извлечения информации.

Для определения лемм – кандидатов на пополнение была выполнена следующая процедура, позволяющая учесть употребления леммы как в отдельных предложениях из множества $\{S\text{sim}\}$, так и в различных кластерах.

Для каждой леммы были вычислены следующие величины:

- freqSsim – количество вхождений слова в шаблоны, извлеченные из предложений $\{S\text{sim}\}$;
- freqSsimClusters – количество кластеров, в которые входят предложения из множества $\{S\text{sim}\}$, где встретилось данное слово;
- freqS – количество вхождений леммы во все извлеченные элементарные шаблоны;
- freqClusters – количество вхождений леммы во всех извлеченных кластерах.

Множество $\{W\}$ лемм-кандидатов строится из лемм, для которых выполняются следующие условия:

FreqSsim ≥ 10 (то есть установлен порог по количеству предложений),

FreqSsimClusters ≥ 3 (установлен порог по количеству кластеров),

FreqSsim / FreqS ≥ 0.8,

FreqSsimClusters / FreqClusters ≥ 0.8.

Таблица 4. Список «проблемных» слов с частотными шаблонами

Слово	Пример частотного шаблона
ПРИВЛЕЧЕНИЕ	[Debtor] ОБЪЯВЛЯТЬ О ПРИВЛЕЧЕНИЕ {КРЕДИТ} В РАЗМЕР [Value]
СООБЩАТЬ	[Debtor] СООБЩАТЬ ЧТО {ПОЛУЧАТЬ} {КРЕДИТ} [Creditor]
ПРИВЛЕКАТЬ	[Debtor] ПРИВЛЕКАТЬ {КРЕДИТ} В [Value]
ДОГОВАРИВАТЬСЯ	[Debtor] ДОГОВАРИВАТЬСЯ О {КРЕДИТ} В [Value]
ТРАНШ	[Creditor] {ВЫДАВАТЬ} [Debtor] ПЕРВЫЙ ТРАНШ {КРЕДИТ}
ВЫДЕЛЯТЬ	[Creditor] ВЫДЕЛЯТЬ {КРЕДИТ} В [Value]
ВЫДАЧА	[Creditor] ВВОДИТЬ МОРАТОРИЙ НА ВЫДАЧА {КРЕДИТ} [Debtor]
СОГЛАШЕНИЕ	[Creditor] СОГЛАШЕНИЕ О {КРЕДИТ} НА [Value]
ПОДПИСЫВАТЬ	[Debtor] ПОДПИСЫВАТЬ С [Creditor] СОГЛАШЕНИЕ ПО ПРИВЛЕЧЕНИЕ {КРЕДИТ}
ПРЕДОСТАВЛЕНИЕ	[Creditor] И ПРЕДОСТАВЛЕНИЕ {КРЕДИТ} [Debtor]

В результате процедуры было получено 10 слов, все из которых оказались необходимыми для пополнения шаблонов для базовой системы извлечения информации (см. табл. 4).

6 Заключение

В данной работе представлен способ автоматизированного обнаружения неполноты шаблонов для системы извлечения информации из текста. Метод основан на нахождении в новостном кластере несколько близких по содержанию предложений, в одном из которых удалось обнаружить извлекаемое событие.

Исследован ряд признаков для обнаружения предложений, потенциально содержащих пропущенный факт. Для наилучшего нахождения таких предложений произведено комбинирование признаков с использованием методов машинного обучения.

Качество работы предложенного метода проверялось на извлечении фактов получения кредита. Эксперименты показали применимость данной системы для обогащения шаблонов системы извлечения информации.

В дальнейшем планируется улучшить качество работы системы за счет большего обобщения шаблонов.

Благодарности

Авторы благодарят компанию ЭР СИ О и лично В.В. Плешко за возможность использования версии RCO Fact Extractor в качестве базовой системы для проведения экспериментов.

Литература

[1] Агеев М.С., Добров Б.В., Лукашевич Н.В., Сидоров А.В. Экспериментальные алгоритмы поиска/классификации и сравнение с «basic line»// Труды второго российского семинара «РОМИП 2004». – 2004. – С. 62-89.

[2] Киселев С.Л., Ермаков А.Е., Плешко В.В. Поиск фактов в тексте естественного языка на основе сетевых описаний// Труды межд. конф. «Диалог 2004»: Компьютерная лингвистика и интеллектуальные технологии, 2004. – С. 282-285.

[3] Ермаков А.Е., Плешко В.В. Семантическая интерпретация в системах компьютерного анализа текста//Информационные технологии. – 2009. – № 6. – С. 2-7.

[4] Хорошевский В.Ф. OntosMiner: семейство систем извлечения информации из мультиязычных коллекций документов// Девятая Национальная конференция по искусственному интеллекту с международным участием КИИ-2004. Труды конференции в 3-х томах. – М.: Физматлит, 2004. – Т. 2. – С. 573-581.

[5] Новостной архив Google, 2010. – http://news.google.ru/archivesearch.

[6] Agichtein E., Gravano L. Snowball: extracting relations from large plain-text collections// Proc. of the Fifth ACM Int. Conf. on Digital Libraries. – 2000. – P. 85-94.

[7] Banko M., Cafarella M., Soderland S., Broadhead M., Etzioni O. Open information extraction from

the Web// Communications of the ACM. – 2008. – P. 68-74.
[8] Brin S. Extracting patterns and relations from the World Wide Web//Proc. of the 1998 Int. Workshop on the Web and Databases. – 1998. – P. 172-183.
[9] Callan J., Croft W., Harding S. The INQUERY retrieval system// Proc. of {DEXA}-92, 3rd Int. Conf. on Database and Expert Systems Applications. – 1992. – P. 78-93.
[10] Dolan B., Quirk C., Brockett C. Unsupervised construction of large paraphrase corpora: exploiting massively parallel news sources//Proc. of the 20th Int. Conf. on Computational Linguistics. – 2004. – P. 1-7.
[11] Etzioni O., Cafarella M., Downey D., Kok S., Popescu A., Shaked T., Soderland S., Weld D., Yates A. Web-scale information extraction in knowitall// Proc. of the 13th Int. Conf. on World Wide Web. – 2004. – P. 100-110.
[12] Harabagiu S., Surdeanu M., Morarescu P. Automatic discovery of linguistic patterns for information extraction// AAAI Press. Proc. of the Fourteenth Int. Florida Artificial Intelligence Research Society Conf. – 2001. – P. 449-453.
[13] Huffman S. Learning information extraction patterns from examples. – Lecture Notes in Computer Science. – Springer-Verlag, 1996. – P. 246-260.
[14] Lin D. Automatic retrieval and clustering of similar words//Proc. of the 17th Int. Conf. on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics (COLING-ACL-98), 1998. – P. 768-774.
[15] Michelakis E., Krishnamurthy R., Haas P., Vaithyanathan S. Uncertainty management in rulebased information extraction systems. – 2009. – P. 101-114.
[16] Mierswa I., Wurst M., Klinkenberg R., Scholz M., Euler T. YALE: rapid prototyping for complex data mining tasks//Proc. of the 12th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD-06), 2006. – P. 935-940.
[17] Pasca M., Lin D., Bigham J., Lifchits A., Jain A. Names and similarities on the Web: fact extraction in the fast lane// Proc. of the 21st Int. Conf. on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, 2006. – P. 809-816.
[18] Riloff E. Automatically generating extraction patterns from untagged text//Proc. of the Thirteenth National Conference of Artificial Intelligent, 1996. – P. 1044-1049.
[19] Riloff E., Phillips W. An introduction to the sundance and autoslog systems// School of Computing. University of Utah, 2004.

Automatically generating patterns for information extraction system

D.S. Kotelnikov, N.V. Loukachevitch

This paper describes an approach for automatic generation of patterns for information extraction system. The technique is based on sentence clusterization around sentences with found facts. News clusters are used as a source of various descriptions of events.