Методология создания шаблонов для русского языка в knowledge-based системах IE

В.Д. Соловьев[1], В.В. Иванов[2], Р.М. Гареев[1], С.В. Серебряков[3], Н.С. Васильева[3]

[1]Казанский (Приволжский) федеральный университет,
[2]НИУ Московский институт стали и сплавов, [3]Hewelett-Packard Laboratories,
maki.solovyev@mail.ru, nomemm@gmail.com, aldvinaldvin@gmail.com,
sergey.serebryakov@hp.com, nvassilieva@hp.com

Аннотация

Статья посвящена методологии создания шаблонов для основанных на знаниях систем извлечения знаний из текстов на русском языке. Для обеспечения полноты набора шаблонов, разрабатываемых экспертами, предлагается использовать две фундаментальные теории: контекстно-свободные грамматики Хомского в варианте восходящего анализа и теорию И. Мельчука "Смысл - Текст".

Методология демонстрируется на конкретных примерах – создания систем шаблонов для извлечения имен людей и ситуаций покупки одной компанией другой. Предложен алгоритм перефразирования, основанный на лексических функциях из модели "Смысл – Текст", позволяющий из одного шаблона события покупки автоматически получить 44 варианта.

1. Введение

В системах извлечения информации (IE, Information Extraction) используются два основных подхода: основанный на знаниях (knowledge-based) и основанный на данных. В первом из них используются разработанные экспертами правила. Правила представляют собой продукции, левая часть которых является шаблоном, позволяющим выделять в текстах определенную семантико-синтаксическую конструкцию, а правая часть представляет собой действие – разметку найденного фрагмента текста. Из размеченного таким образом текста потом извлекаются и представляются в нужной форме знания.

Достоинством такого подхода является высокие показатели по точности и полноте, а также интерпретируемость работы системы. К недостаткам относится высокая трудоемкость составления правил. Поэтому представляют интерес методы, направленные на облегчения экспертам составления правил.

Труды XV Всероссийской объединенной конференции «Интернет и современное общество» (IMS-2012), Санкт-Петербург, Россия, 2012.

Knowledge-based подход применялся во многих работах [1-4], но при этом сами наборы правил не публиковались. Более того, в публикациях не описывается, как получаются правила. В данной работе описывается методология построения правил, точнее, наиболее сложной их части – шаблонов. Эта методология была применена при создании системы извлечения из русскоязычных текстов информации о бизнес событиях.

2. Шаблоны

Выделение событий является наиболее высокоуровневой задачей IE. Для нахождения в тексте описаний событий требуется сначала выделить именованные сущности: людей (в шаблонах обозначаемых как Person), компании и некоторые другие. Эти сущности являются аргументами событий. Например, в событии назначения нового руководителя компании аргументами являются название компании, имя назначенного человека, его должность и дата назначения.

Кроме правил в системах IE используется и другой вид знаний – словари. Мы используем словари наименований компаний, а также имен и фамилий людей. Используются и небольшие вспомогательные словари (должностей и т.д.).

Для более точной идентификации людей введено понятие идентификатор персон. Это такие слова как: *бизнесмен*, предприниматель, *миллиардер*, *олигарх* и т.д. Эти слова помещаются в словарь PersonIndicator. Обозначим словари имен и фамилий Name и Surname соответственно. Простой пример правила:

(1) Name Surname -> Person

Правило (1) означает, что если в тексте найдены рядом стоящие слова из словарей имен и фамилий, то они образуют наименование персоны. Для записи правил и разметки текста используется TextMarker [5]. На языке TextMarker это же правило выглядит так:(2) Name Surname {-> MARK(Person, 1, 2)}

Здесь MARK – собственно оператор маркирования, цифры 1 и 2 означают, какие именно по порядку элементы шаблона войдут в маркировку персоны. В дальнейшем, не отвлекаясь натехнические аспекты TextMarker-а, будем записывать правила в форме, как в примере (1).

3. Источники шаблонов

При построении шаблонов разработчик должен учитывать все особенности грамматики естественного языка, т.е. фактически требуется создать формальную модель адекватного задаче фрагмента грамматики языка. С этой целью разработчик может использовать следующие источники информации: а) собственную интуицию, б) примеры из текстов, в) модель языка на основе грамматик Хомского, г) другие формальные модели языка (например, для русского языка модель "Смысл ←→ Текст" И.А.Мельчука [6]).

Полагаться только на собственную интуицию рискованно, можно что-то упустить. Это может себе позволить только очень опытный разработчик. Начинающим мы рекомендуем использовать иные источники информации. Естественной кажется ориентация на реальные тексты. Просматривая их можно писать шаблоны для каждого встретившегося подходящего примера. Однако этот подход принципиально не может гарантировать полноту множества шаблонов. Разработчик не сможет просмотреть миллионы новостных сообщений. В небольших же по размеру выборках вполне могут не встретиться какие-то грамматически правильные конструкции.

Грамматики Хомского, как и другие формальные модели, предназначены для порождения всех грамматически правильных предложений, поэтому их можно использовать для обеспечения полноты множества шаблонов.

В качестве примера приведем множество описывающих персон правил, записанных в форме контекстно-свободной грамматики Хомского (восходящий анализ [7]). В этом примере использованы словари наименований компаний (Company), должностей (Position), а также словарь SurnamePrefix, включающий слова Мак, ван, де, Тер и некоторые другие, и словарь Initial, содержащий все буквы с точками (инициалы). Символ W обозначает любое слово, CW – любое слово с заглавной буквы. Используются также знаки "и" и ",".

- Surname → Person
- SurnamePrefix SurnamePrefix → SurnamePrefix
- SurnamePrefix Surname → Person
- Initial Initial → Initial
- Initial Surname → Person
- Surname Initial → Person
- Name Name → Name
- Surname Surname → Surname
- Name Person → Person
- Person Name → Person
- PersonIndicator Person → Person
- PersonIndicator CW → Person
- CW CW → CW
- Person "," Person → Person
- Person "и" Person → Person
- Person "," W "," → Person
- W W → W
- Position "," Position → Position
- Position Company → Person
- Position W Company → Person
- Person Person → Person

Пример восходящего анализа. Текст "Председатель совета директоров "Газпрома" Виктор Зубков" после словарного поиска преобразуется в "Position Company Name Surname", затем, после применения вышеприведенных правил, последовательно в "Person Person" и "Person". Порядок правил в вышеприведенном перечне соответствует порядку их применения к тексту.

Обратим внимание на то, что правила W W → W и CW CW → CW позволяет сворачивать длинные конструкции. Преобразование правил в формат TextMarker не сложно и может быть осуществлено даже автоматически. При этом следует иметь в виду, что вышеприведенный набор правил является теоретической моделью, при ее практическом воплощении средствами TextMarker следует учитывать ряд особенностей этой системы.

Строго восходящий анализ является не единственной возможностью. Нами реализован более сложный комбинированный алгоритм с измененным порядком различных фаз анализа. Общая идея этого алгоритма следующая. Сначала все слова с заглавной буквы маркируются как PossibleName (первые слова в предложениях обрабатываются отдельно), затем применяются правила, уточняющие тип этих слов (человек или компания) с помощью PersonIndicator и CompanyIndicator, затем применяются шаблоны событий и лишь после этого, если неопределенность в идентификации имен сохраняется, происходит обращение к словарям имен, фамилий, названий компаний.

4. Перефразирование

Для русского языка на уровне предложений эффективным является и использование модели "Смысл ←→ Текст" И.А.Мельчука. Эта модель включает описание сохраняющих грамматическую правильность трансформации (перефразирование). В ней введен набор лексических функций: **Syn, Conv, Der, Gener.** Приведем краткие определения этих функций.

Syn (L) [синонимы] возвращает синонимы L: **Syn (бегемот) = {гиппопотам}**

Conv (L) [конверсивы] возвращает конверсивы L: **Conv (купить) = {продать}**

Der (L) [дериваты] возвращает все глубинные дериваты L. Эта лексическая функция разбивается на 4 случая:

- **S_0 (L)** возвращает существительное, конгруэнтное L: **S_0 (анализировать) = {анализ}**
- **A_0 (L)** возвращает прилагательное, конгруэнтное L: **A_0 (город) = {городской}**

- **V_0 (L)** возвращает глагол, конгруэнтный L: **V_0 (анализ) = {анализировать}**
- **Adv_0 (L)** возвращает наречие, конгруэнтное L: **Adv_0 (следовать) = {после}**

Gener (L) [обобщение] возвращает ближайший более общий концепт для L: **Gener (яблоко) = фрукт.**

Более полные определения и пояснения можно найти в [6]**.**

В ряде работ [3] использовались те ли иные трансформации, но это делалось не систематическим образом.

Приведем пример применения этого подхода для получения новых шаблонов события покупки компании. При этом используется словарь AcqusitionIndicator слов, обозначающих ситуацию покупки.

Входные параметры:

1) ключевое слово C0, обозначающее ситуацию/событие, т.е. индикатор события («купить») из словаря AcqusitionIndicator
2) актанты (аргументы, слоты) ситуации/события (C0: S0, S1).

Выходные параметры:

«заготовки» для формирования шаблонов вида: S_0, C_0, S_1 которые при выборе конкретных классов для S_0, C_0, S_1 преобразуются в “Company Acqusition-Indicator Company”.

Описание метода:

ШАГ 1. Создать «заготовки» шаблонов путем комбинирования C_0, S_0, S_1. При комбинировании допускается применение операций перестановки мест и удаления актантов и ключевого слова.

ШАГ 2. Сгенерировать множество T_{syn}={Syn(C_0)} (например, *покупать, приобрел и др. варианты*) Если T_{syn} – пусто, перейти к ШАГ 3, иначе для каждого элемента t_{syn} этого множества положить $C_0 \leftarrow t_{syn}$ и выполнить ШАГ 1. После обхода всех элементов множества перейти к ШАГ 3.

ШАГ 3. Сгенерировать множество T_{conv}={Conv(C_0)} (например, купить ↔ продать). Если $T_{con\ v}$– пусто, перейти к ШАГ 4, иначе для каждого элемента t_{conv} этого множества положить $C_0 \leftarrow t_{conv}$ и выполнить ШАГ 1 и ШАГ 2. После обхода всех элементов множества перейти к ШАГ 4.

ШАГ 4. Сгенерировать множество T_{der}={Der(C_0)}. Для каждого элемента t_{der} этого множества и каждого S_i сформировать $f(S_i)$ – (возможно, пустой) актант, соответствующий по смыслу S_i в ситуации, обозначаемой t_{der} положить $C_0 \leftarrow t_{der}$, $S_i \leftarrow f(S_i)$ и выполнить ШАГ 1, ШАГ 2 и ШАГ 3.

ШАГ 5 (опциональный). Сгенерировать множества T_{gen}={Gener(C_0)}, а также множества {Gener(Syn(C_0))}, {Gener(Conv(C_0))}, {Gener(Der(C_0))}. Для каждого элемента t полученных множеств сгенерировать множество актантов (см. $f(S_i)$, ШАГ 4), положить $C_0 \leftarrow t$, $S_i \leftarrow f(S_i)$ и выполнить ШАГ 1, ШАГ 2, ШАГ 3 и ШАГ 4.

Данный метод позволяет, например, из конструкции “компания А купила компанию В” получить конструкцию: “владелец продал компанию В компании А”. При применении метода автоматическим образом к конструкции “дата персона А уволила с должности В компании С персону D” получается 44 варианта, которые было бы весьма затруднительно перечислить вручную.

Контекстно-свободные грамматики Хомского и метод перефразирования могут применяться как при первоначальном создании набора шаблонов, так и для его расширения в, так называемом, “Active learning” подходе в машинном обучении шаблонам [8]. Метод перефразирования ранее применялся в области QA (Question Answering) [9, 10].

5. Заключение

Наш опыт разработки системы извлечения знаний для русского языка показывает, что наиболее эффективным является применение комбинированного подхода, включающего рассмотрение реальных примеров ситуаций в текстах и применение теоретических методов, таких как грамматики Хомского и модель “Смысл ←→ Текст” И.А.Мельчука.

В статье приведены конкретные примеры реализации этих методов. Использование теоретических методов имеет своей целью обеспечение полноты набора шаблонов. Кроме того, это позволяет представить набор шаблонов наиболее элегантным образом. Однако эти методы и не лишены некоторых недостатков. Например, генерируется слишком большое число правил. Вероятно, требуется определенная ручная постобработка получаемых наборов шаблонов. Дальнейшие исследования будут направлены на экспериментальную оценку системы, разработанной на основе описанной методологии.

Литература

[1] Hogenboom F., Frasincar F., Kaymak U., and Franciska de Jong. An Overview of Event Extraction from Text // Workshop on Detection, Representation, and Exploitation of Events in the Semantic Web (DeRiVE 2011) at Tenth International Semantic Web Conference (ISWC 2011), Vol. 779. P. 48-57. CEUR-WS.org, 2011.

[2] Borsje J., Hogenboom F., Frasincar F. Semi-Automatic Financial Events Discovery Based on Lexico-Semantic Patterns // Int. J. Web Engineering and Technology. Vol. 6. No. 2.P. 115-140. 2010.

[3] Xu, F., Uszkoreit, H., Li, H. Automatic Event and Relation Detection with Seedsof Varying Complexity // AAAI Workshop on Event Extraction and Synthesis. 2006.

[4] Проект OntosMiner. URL: http://www.ontos.com/?page_id=630, 2012.

[5] Kluegl P., Atzmueller M., and Puppe F. TextMarker: A Tool for Rule-Based Information Extraction // Proc. Unstructured Information Management Architecture UIMA, 2nd UIMA@GSCL Workshop. 2009 Conference of the GSCL Gesellschaft für Sprachtechnologie und Computerlinguistik. 2009.

[6] Мельчук И.А. Опыт теории лингвистических моделей «Смысл-Текст». М. 1974.
[7] Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции. М.: Мир, 1978.
[8] Thompson C.A., Califf M.E., Mooney R.J. Active Learning for Natural Language Parsing and Information Extraction // Proceedings of the XVI International Machine Learning Conference. 1999. P. 4406-4414.
[9] Bouma G., Fahmi I., Mur J., van Noord G., van der Plas L., Tiedemann J. Linguistic Knowledge and question answering // Traitement Automatique des Langues (TAL). 46 (3). 2005.P. 15-39.
[10] Hermjacob U., Echihabi A., Marcu D. Natural language based reformulation resource and wide exploitation for question answering // XI Text Retrieval Conference, Vol. 5500-251 of NIST Special Publication. 2002.

Methodology for Building Extraction Templates for Russian Language in Knowledge-Based IE Systems

V. Solovyev, V. Ivanov, R. Gareev, S. Serebryakov, N. Vassilieva

Rules developed by experts are widely used in knowledge-based information extraction (IE) systems. The rules are productions consisting of two parts. The left-hand side of a rule is a template that matches a certain syntactico-semantic structure and the right-hand side is an action that is executed when left-hand side template is matched against a particular text fragment. A typical action is to annotate a matched text fragment with a new syntactico-semantic structure. Once the text is annotated in such a way, knowledge is extracted and stored in a particular format. Rule-based approach is described in a number of works. To our knowledge, there are no papers providing the set of rules developed for a particular domain or describing a methodology used to develop the rules themselves. In this paper we describe our methodology for building extraction rules. In particular, we describe the process of building a more complex left-hand side part (further in the paper we will refer to left-hand side as template).