DOI 10.37882/2223–2966.2021.05.15

АНАЛИЗ И КЛАССИФИКАЦИЯ АЛГОРИТМОВ ИЗВЛЕЧЕНИЯ ОТНОШЕНИЙ ИЗ ТЕКСТОВЫХ ДАННЫХ

Кобышев Кирилл Сергеевич
Аспирант, Санкт-Петербургский политехнический университет Петра Великого
kobyshev2.ks@edu.spbstu.ru
Молодяков Сергей Александрович
Д.т.н., профессор, Санкт-Петербургский политехнический университет Петра Великого
molodyakov_sa@spbstu.ru

Аннотация. На данный момент поиск информации в неструктурированных данных, представленных в виде текста, является нетривиальной задачей. Текстовые данные возможно преобразовать в структурированные данные за счет алгоритмов автоматического извлечения отношений. Структурированное представление текста позволяет воспользоваться преимуществами структурированных данных: объяснимость найденных фактов, простота поиска фактов, возможность применения механизмов ускорения доступа к данным. В данной статье рассматриваются и сравниваются между собой подходы к извлечению знаний из текстовых данных: полуавтоматический подход, подход на основе обучения с частичным привлечением учителя, с полным привлечением учителя, с использованием внешних баз знаний и обучения без учителя.

Ключевые слова: извлечение отношений, структурированные данные, компьютерная лингвистика, извлечение знаний, обработка текстовых данных.

ANALYSIS AND CLASSIFICATION OF ALGORITHMS FOR RELATION EXTRACTION FROM TEXT DATA

K. Kobyshev
S. Molodyakov

Summary. Now, searching for information in unstructured data presented in the form of text is a non-trivial task. Text data may be casted to structured data with automatic relation extraction algorithms. Structured text representation allows to use structured data advantages: explainability of found facts, simplicity of fact search, possibility of data access acceleration mechanisms using. In this paper the following relation extraction approaches were considered and compared: semi-automatic approach, weakly supervised learning, supervised learning, distantly supervised learning, unsupervised learning.

Keywords: relation extraction, structured data, computational linguistics, knowledge extraction, text data processing.

Введение

В настоящее время наблюдается непрерывный рост объема цифровых данных, размещаемых как для общего пользования, так и для локального доступа. Цифровые данные можно разделить на 2 типа: структурированные данные и неструктурированные.

Структурированные данные представляют собой способ представления информации (знаний) в виде заданных упорядоченных структур. Примером хранения структурированных данных может послужить файл с информацией, представленной в каком-либо формате (JSON, XML, CSV), таблица в реляционной БД. Структурированные данные представляют информацию согласно строгой спецификации и позволяют упростить разработку алгоритмов поиска интересуемой информации. Кроме того, представление данных в виде структур позволяет применить механизмы ускорения доступа к данным такие, как кэширование [1], индексирование, партиционирование, шардирование. Структурированные данные позволяют извлечь не только элементарную информацию из атомарных фактов (простых структур данных), но и позволяют извлечь менее очевидные факты с помощью сложных поисковых запросов, которые включают преобразования и объединения различных множеств структурированных данных (например, JOIN запросы в SQL-подобных языках, метод резолюций в экспертных системах). При этом извлеченный факт будет подкреплен доказательством, состоящим из атомарных фактов и которое объяснимо для человека (в противовес методам машинного обучения, которые не могут объяснить полученные факты из-за чрезмерной абстракции признаков, формируемых математическими моделями).

Менее тривиальной является автоматизация поиска информации в неструктурированных данных, которые так же могут содержать полезную информацию. Неструктурированные данные представляют собой необработанные цифровые данные и могут быть представлены в виде изображений, аудиозаписей, видеозаписей, текста. На сегодня разрабатывается и используется большое множество алгоритмов извлечения знаний из неструктурированных данных. Применение конкретного алгоритма определяется типом хранимых данных и эффективностью извлечения данных. Часто удается использовать несколько алгоритмов одновременно [2].

Широкое практическое применение нашли алгоритмы для преобразования текста в структурированные данные. Представление текста в виде набора структурированных данных позволяет воспользоваться преимуществами структурированных данных: объяснимость найденных фактов (знаний), относительная простота поиска фактов, возможность применения механизмов ускорения доступа к данным. Кроме того, на данный момент довольно распространены открытые источники текстовых данных: новостные веб-сайты, электронные книги, медицинские справочники и так далее.

Текст представляет собой способ представления информации на каком-либо естественном языке. Любой естественный язык в большей или меньшей степени подчинен синтаксическим, грамматическим правилам, что делает возможным построение его математической модели, которую впоследствии возможно применить для извлечения знаний из текста на данном языке. Онтология является наиболее распространенной структурой данных, используемой для представления текстовых данных в структурном виде. Онтология представляет собой графовую структуру данных, которая состоит из понятий, отношений между понятиями, экземпляров понятий, и над понятиями которой возможно задать множество аксиом (логических утверждений, которые невозможно представить в виде отношений) [3]. В самом простом случае отношение представляет собой триплет — структуру данных, которая включает в себя 2 понятия и взаимосвязь между ними [4]. Отношение может иметь более сложную структуру, иметь в своем составе более двух понятий [5]. Можно привести следующие примеры типов отношений: род-вид, целое-часть, причина-следствие, термин-синоним термина, объект- действие — субъект действия [6]. Возможно привести следующие примеры систем для извлечения знаний из текста: Tomita Parser [6] (автоматическое извлечение фактов из отзывов об автомобилях, почтовый помощник, группировка новостных сюжетов), Prospector [7] (используется геологами для поиска мест для бурения), PUFF (медицинская система для диагностики респираторных заболеваний) [7], Design Advisor (используется разработчиками микропроцессорных систем) [7], LITHIAN (дает рекомендации археологам для изучения каменных орудий труда) [7].

При решении задачи извлечения знаний из текстовых данных необходимо знакомство с базовыми алгоритмами, их эффективностью и областями применения. В данной статье рассматриваются и сравниваются между собой подходы к извлечению знаний из текстовых данных. Исходя из преимуществ и недостатков методов извлечения знаний из текстов в данной статье сформулированы области применения методов извлечения знаний из неструктурированных данных. Основными подходами извлечения знаний из текста являются: подход на основе правил, на основе обучения с частичным привлечением учителя, на основе обучения с учителем, на основе обучения с учителем с использованием внешних баз знаний, на основе обучения без учителя.

Полуавтоматический подход извлечения отношений

Первые алгоритмы извлечения знаний начали разрабатываться в конце 1980-х годов и по большей части были основаны на ручном составлении правил специалистами по лингвистике [8]. Например, одной из таких первых систем извлечения знаний была AutoSlog [9], Ontosminer [9]. Самый простой способ задать правила — это использовать регулярные выражения с непосредственным определением общего вида строк [10]. Например, можно извлекать отношения типа столица-страна: “<T1>is a capital of<T2>”. После того, как участок текста по указанному правилу найден, из него извлекаются 2 понятия и отношение. Правила для извлечения отношений часто на практике бывают сложнее, чем простые регулярные выражения. Правила извлечения отношений могут использовать другие алгоритмы обработки естественного языка. Таким образом, они будут более точно определять отношения в тексте. Например, правила могут содержать описание синтаксического дерева, в котором располагается отношение, описание частей речи у слов, которые могут состоять в отношении, типы именованных сущностей, падежи в словах (для русского языка).

Извлечение знаний на основе обучения с частичным привлечением учителя

Большая часть алгоритмов данной группы основана на парадигме «открытого извлечения информации», идея которой заключается в том, чтобы задать небольшое множество правил, которое автоматически расширяется в процессе чтения текстового корпуса. При этом при образовании новых правил в большей части алго-

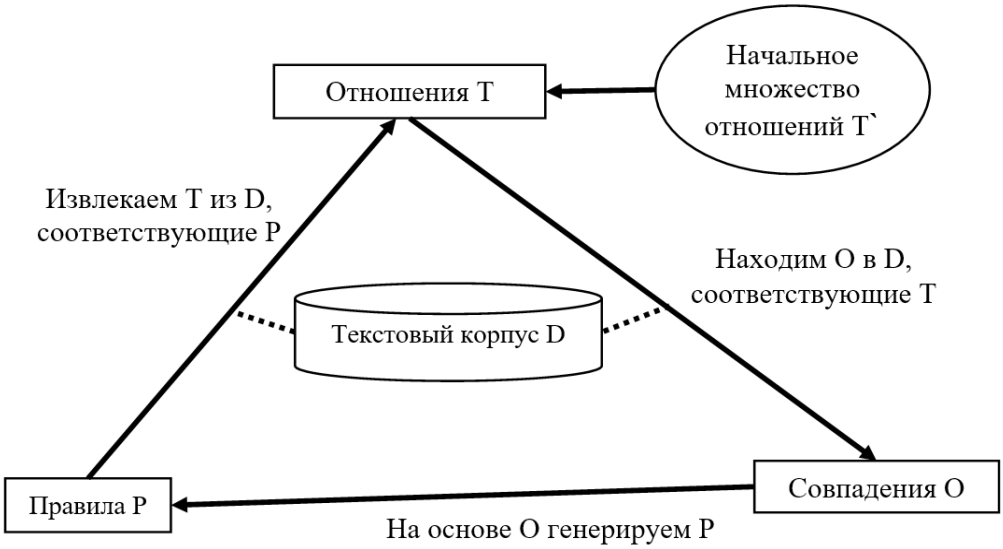

Рис. 1. Алгоритм DIRPE

ритмов используется бэггинг (bootstraping), чтобы отбросить сгенерированные правила с низкой точностью. Наиболее известные алгоритмы с частичным привлечением учителя, которые принимают на вход примеры отношений, являются DIRPE и Snowball. Наиболее известные алгоритмы, которые принимают на вход небольшое множество правил и согласно парадигме открытого извлечения информации расширяют его, являются KnowItAll, TextRunner, ReVerb.

Идея алгоритма DIRPE (Dual iterative pattern expansion) [11] заключается в том, чтобы на основе небольшого набора из примеров для одного типа отношений S сгенерировать правила P и с каждой итерацией их дополнять (см. рис. 1).

Сначала вручную составляются примеры отношений T` = (t1, t2), где t1 — первое понятие, t2 — второе понятие. Из примеров отношений T=T` определяются совпадения в тексте O. Из совпадений мы можем выделить слово, которое следовало до первого понятия (l); слова между двумя понятиями (m); порядок расположения понятий (order); слово после второго понятия (r); адрес веб-страницы, где найдено отношение (urlprefix). Таким образом, формируются новые правила, состоящие из набора (order, url, l, m, r). Согласно данным правилам, находим новые отношения из текста T, и по ним снова находим совпадения O. Шаги данного алгоритма повторяются, пока не будет достигнуто состояние, когда новые совпадения не будут найдены.

Snowball является улучшением алгоритма DIRPE. Важное отличие от DIRPE заключается в том, что алгоритм Snowball включает в себя этап отбрасывания ненадежных сгенерированных правил [12]. В отличие от DIRPE найденные совпадения помимо prefix, middle, suffix дополняются числом, который зависит от частоты встречаемости слова рядом со словами-понятиями в окне-контексте размером в w слов. Найденные совпадения группируются по данным числам в кластеры, после чего определяются центроиды ($order_c$, url_c, l_c, m_c, r_c), остальные совпадения удаляются. При кластеризации за расстояние между совпадениями считается скалярное произведение их векторов по выражению 1:

$$Match(O_1, O_2) = l_1 * l_2 + m_1 * m_2 + r_1 * r_2 \quad (1)$$

На каждой итерации пересчитываются степень надежности отношений (выражение 3) и степень надежности правил (выражение 4, где C_i — контекст l, m, s который был найден по правилу P_i) на основе надежности правила на текущей итерации (выражение 2, где $P_{positive}$ — это правильно определенные отношения в тексте; $P_{negative}$ — отношения, конфликтующие с предыдущими найденными отношениями).

$$Conf(P) = \frac{P_{positive}}{P_{positive} + P_{negative}} \quad (3)$$

$$Conf(T) = 1 - \prod_{i=0}^{|P|}(1 - (Conf(P_i) * Match(C_i, P_i))) \quad (4)$$

$$Conf(P) = Conf_{new}(P) * W_{updt} + Conf_{old}(P) * (1 - W_{updt}) \quad (5)$$

Если степень надежности правила или отношения меньше заданного порога, то данное правило или отношение удаляется из сохраненного списка.

Недостаток алгоритмов DIRPE и Snowball заключается в том, что они неэффективны в неоднороднородных текстах, предназначены для узконаправленных текстов на одну тематику, в которых могут использоваться одинаковые слова. В алгоритме KnowItAll предлагается решить эту проблему за счет того, что формируемые системой правила основаны не на символьных регулярных выражениях, а на разметке частей речи у слов. Данный алгоритм автоматически преобразует правила в поисковые запросы, с помощью которых загружаются страницы, потенциально содержащие примеры отношений. KnowItAll не требует в качестве входных данных примеры отношений и вместо них принимает на вход множество правил, которое расширяется процессе обучения. Надежность правила определяется как отношение встречаемости слов в соответствии с данным правилом к встречаемости данного слова без правила. Точность найденных отношений определяется с помощью байесовского классификатора, и зависит от надежности правил, с помощью которых происходит поиск данного отношения.

Недостаток алгоритма KnowItAll заключается в долгой обработке большого количества поисковых запросов для получения новых данных для обучения. В алгоритме TextRunner предлагается решить данную проблему за счет распределения процесса поиска каждого типа отношений по отдельным вычислительным узлам [13]. Также в данном алгоритме в отличие от алгоритма KnowItAll предлагается обучить байесовский классификатор не на основе надежности составляющих отношение правил, а на встречаемости тех или иных признаков составляющих отношение слов (части речи, количество слов отношении, число стоп-слов, часть речи слова слева от понятия e_i, часть речи слова справа от понятия e_j).

Алгоритм ReVerb позволяет устранить недостаток предшествующих алгоритмов открытого извлечения информации, который заключается в том, что могут сформироваться неинформативные (с вероятностью 4–7%) и бессмысленные отношения (с вероятностью 13–30%) [14]. Данный недостаток достигается за счет синтаксического ограничения: слова из кандидата на правило для определения типа отношения должны соответствовать регулярному выражению "V | V P | V W*P", где V является регулярным выражением "verb particle? adv?", W соответствует регулярному выражению "noun | adj | adv | pron | det", P соответствует "prep | particle | inf. marker". Также, согласно данному алгоритму, устраняются избыточные, содержащие слишком много слов отношения, по которым практически невозможно найти совпадения в тексте, за счет лексического ограничения: слова из текста кандидата на новое правило для типа отношения должны встречаться достаточно часто в большом текстовом корпусе. Таким образом, в основе алгоритма ReVerb лежит процесс поиска отношений по заданному синтаксическому ограничению так, чтобы охватить как можно больше слов между двумя именами существительными, но при этом не превысить лексическое ограничение (чем больше слов, тем меньше встречаемость сгенерированного правила, которое содержит данные слова). Однако модель, предлагаемая в ReVerb для извлечения отношений не способна вычислить степень надежности извлеченного отношения. Для этого используется отдельная модель логистической регрессии, которую необходимо обучить на примерах, сделанных вручную (авторы обучили эту модель на примерах из 1000 предложений).

Извлечение знаний на основе обучения с учителем

Алгоритмы данной группы основаны на обучении на большой структурированной выборке данных, состоящих из правильных и неправильных примеров отношений. При этом идея большинства алгоритмов состоит в обучении классификатора, который определяет принадлежность к одному определенному типу отношений. На выходе классификатора формируется число от –1 (не является отношением) до +1 (является отношением). Существует два основных типа алгоритмов данной группы: основанные на признаках и основанные на ядрах [15].

Идея алгоритмов, основанных на признаках, заключается в том, чтобы извлечь из предложения такие признаки как: понятия, типы именованных сущностей у двух понятий, последовательность слов между понятиями, число слов между двумя понятиями, путь в синтаксическом дереве между двумя понятиями. Данные признаки подаются на модель, которая изменяет свои параметры, веса так, чтобы ее выход соответствовал типу отношений из обучающей выборки. В качестве модели, которая обучается на признаках можно использовать наивный байесовский классификатор [15], то есть решение о типе отношения будет приниматься на основе вероятности из выражения 6, где P(C) — абсолютная вероятность типа отношений C, $P(F_i|C)$ — вероятность наличия признака i в отношениях типа C:

$$P(C|F_1, \ldots, F_n) = P(C) * \prod_{i=1}^{n} P(F_i|C) \qquad (6)$$

В качестве модели также на практике используют глубокие нейронные сети. Например, существует решение, где используется сверточная нейронная сеть [16]. При-

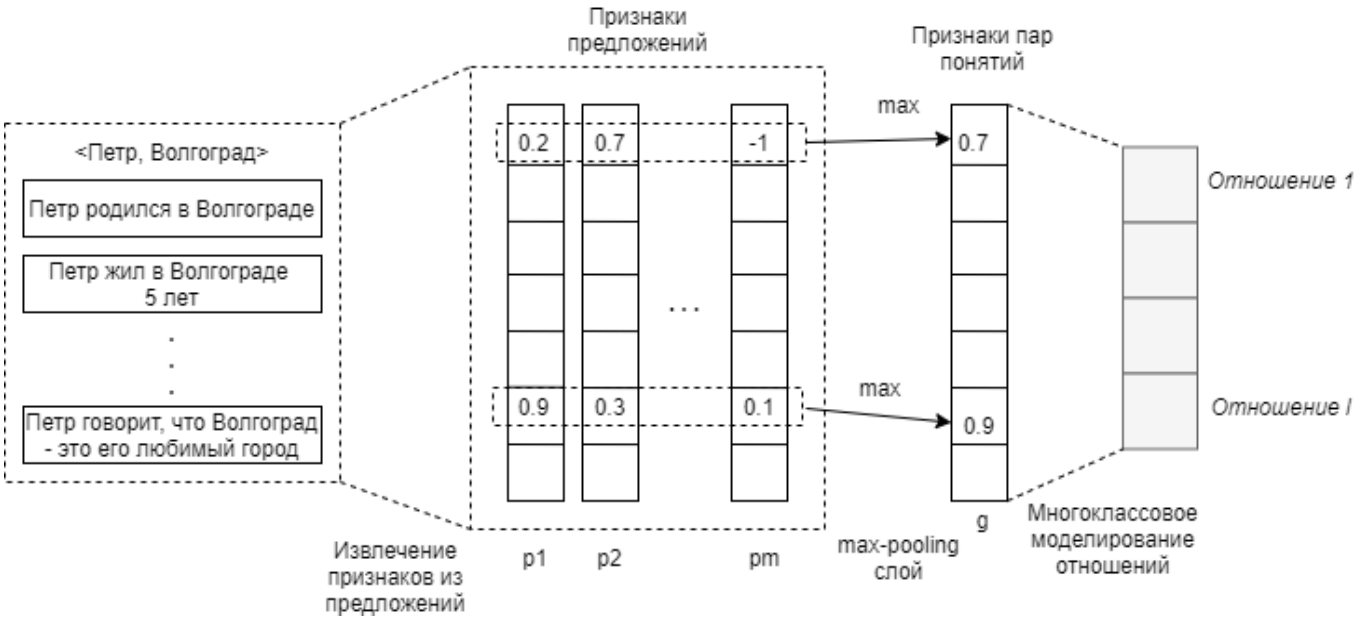

Рис. 2. Модель многоклассовой классификации отношений

знаки представляются в форме матрицы, где номерам строк соответствуют слова из предложения, а номерам столбцов соответствуют признаки слов. В простейшем случае сверточная нейронная сеть состоит из слоев: свертка, max-pooling, полносвязный слой, слой активации softmax. На выходном слое формируется вектор вероятностей типов отношений. Также существует решение, где была использована реккурентная нейронная сеть с LTSM-блоками [17].

Алгоритмы на основе ядер в отличие от алгоритмов на основе признаков могут принимать на вход синтаксические признаки (например, синтаксическое дерево предложения), которые в случае с методом на основе обучения на признаках будут иметь слишком большую размерность. Методы на основе ядер используют вместо признаков для обучения только функцию для сравнения (функция близости) каких-либо пар объектов из обучающей выборки (косинусное расстояние, евклидово расстояние и любая другая функция, которую задаст исследователь) [18]. В качестве моделей, которые обучаются при помощи функции близости, используются: перцептрон Розенблатта, метод опорных векторов, деревья принятия решений, ядро мешка слов (Bag of words kernel), а также алгоритмы, совмещающие разные ядра [19].

Извлечение знаний с использованием внешних баз знаний

Существует подход по автоматическому построению правил извлечения отношений между понятиями при помощи уже подготовленных примеров отношений из удаленных баз знаний, таких как DBPedia, Wikidata, Freebase [20]. Данный способ совмещает подходы с частичным привлечением учителя и подход на основе обучения с учителем. Допустим, в удаленной базе данных имеется отношение r_k (e_i, e_j), где r_k тип отношений под номером k, e_i — понятие под номером i, e_j — понятие под номером j. Тогда можно построить предположение, что все упоминания s_{ij} понятий e_i и e_j в одном предложении есть тоже данное отношение r_k. Множество слов между двумя понятиями из упоминаний формируют правила для отношения r_k, либо используя данное множество слов можно обучить бинарный классификатор для отношения r_k.

Однако существует вероятность, что найденные примеры s_{ij} являются неверными, что они не соответствуют отношению r_k, либо они соответствуют не только отношению r_k, но и другим нескольким отношениям. Данную проблему возможно решить, использовав модели многоклассовой классификации [21] [22], согласно которым модель обучается не на отдельных примерах, а на “мешках” из примеров отношений для пары понятий e_i и e_j, которые составляют какое-либо отношение, а на выходе модели формируется вектор вероятностей относительно всех типов отношений.

Начиная с середины 2010-х годов активно развивается идея использования моделей сверточных сетей для многоклассовой классификации с использованием примеров отношений из удаленных баз знаний. Например,

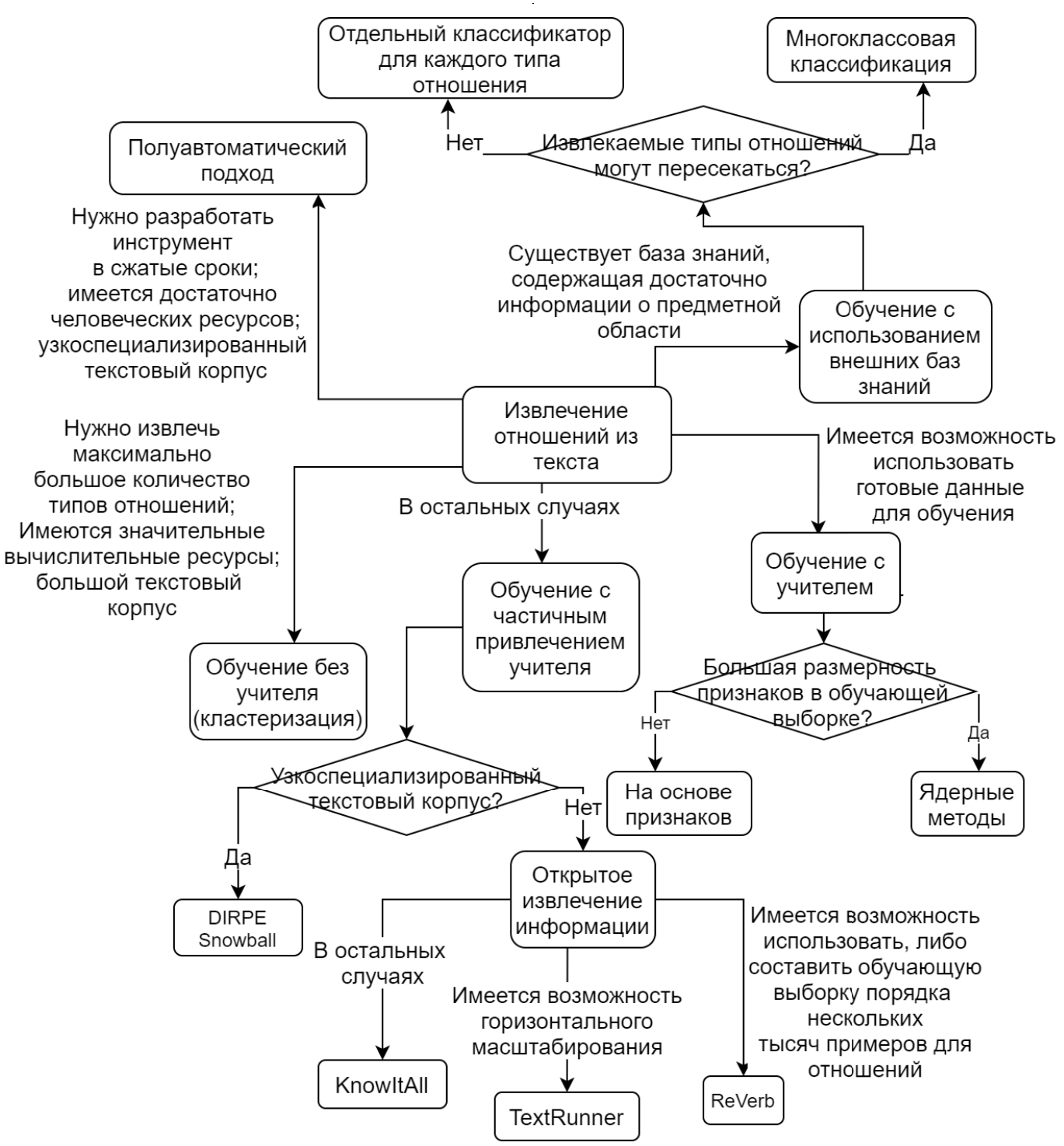

Рис. 3. Схема выбора методов извлечения отношений из текста

существует модель [23], которая схематично показана на Рис. 2.

Согласно данной схеме на вход модели подается набор предложений с упоминаниями понятий e_i и e_j (на рисунке в качестве примера такой пары понятий приводятся "Петр" и "Волгоград"). Сначала из данных предложений извлекаются признаки (части речи слов, синтаксическое дерево и т.д.). Затем формируется вектор из максимумов строк матрицы признаков. Затем слой активации преобразует вектор признаков в вектор вероятностей отношений (в качестве функции активации используется сигмоида).

Извлечение знаний на основе обучения без учителя

Алгоритмы, основанные на парадигме открытого извлечения информации, практически являются алгоритмами обучения без учителя. Однако данные методы требуют небольшое множество правил на входе. Полнота (recall) и точность (precision) результатов выполнения данных алгоритмов зависит от того, насколько удачно были подобраны правила, которые подаются на вход данных алгоритмов. Существуют методы извлечения отношений из текста, которые не требуют входных данных (не требуется даже небольшое множество) и для которых нужен объемный текстовый корпус. Наиболее распространенный метод обучения без учителя — деление на кластеры пространства признаков пар слов-понятий, образующих отношения между понятиями. Например, возможно применить NER-тегирование для определения именованных сущностей (понятий), чтобы затем построить матрицу совместной встречаемости между парами понятий по словам из связующего контекста (слова между двумя понятиями), после чего сгруппировать пары слов в кластеры по их векторам из полученной матрицы [24]. Для данного алгоритма извлечения отношений возможно уменьшить влияние бессмысленных слов на кластеризацию [25], если важность каждого слова $I(w_k)$ из контекста между двумя именованными сущностями, как информационную энтропию после исключения признака w_k (исключаем данное слово из матрицы совместной встречаемости). Авторы статьи [26] предлагают использовать в качестве признаков для кластеризации грамматику зависимостей.

Сравнение подходов извлечения знаний из текста

Методы извлечения отношений из текста возможно систематизировать в виде древовидной схемы из Рис. 3, где блоки представляют собой методы или группы методов, а соединительные линии представляют собой условия для решаемой задачи извлечения отношений (некоторые соединительные линии заменены условным блоком для сокращения размеров схемы).

При условии, если имеется достаточно человеческих ресурсов для ручного составления правил извлечения отношений из текста (регулярных выражений), также если текстовый корпус достаточно узко специализирован, возможно в относительно сжатые сроки реализовать инструмент для извлечения отношений на основе полуавтоматического подхода. Возможен случай, когда имеется достаточный объем подготовленных данных для обучения, например, если некоторое время данные об отношениях, извлеченных из текста, формировались сотрудниками предприятия вручную или отношения были взяты из открытого источника с наборами данных для обучения). В этом случае нужно сделать выбор между ядерными методами обучения и методами на основе признаков. Выбор должен быть сделан в пользу ядерных методов, если пространство признаков в обучающей выборке достаточно велико (порядка десяти и более). Если существует открытая база знаний, которая содержит достаточный объем информации о заданной предметной области, то можно воспользоваться ней, чтобы построить собственную базу знаний с интересуемыми типами отношений. Если необходимо учесть возможность пересечения типов отношений, то нужно воспользоваться методами многоклассовой классификации отношений, иначе — обучить для каждого типа отношений собственный классификатор. Обучение без учителя рекомендуется, если требуется извлечь максимально большое число типов отношений. При этом требуется достаточно большой текстовый корпус, а также достаточно большие вычислительные ресурсы для обучения на данном текстовом корпусе. Во всех остальных случаях остается только использовать методы обучения с частичным привлечением учителя, когда от пользователя требуется задать небольшое начальное множество. С узкоспециализированным текстовым корпусом могут достаточно эффективно (с высокой точностью и полнотой результатов) справиться алгоритмы DIRPE, Snowball. Для неоднородных текстовых корпусов требуется использовать методы открытого извлечения информации. Рекомендуется сделать выбор в пользу алгоритма TextRunner, если имеется возможность горизонтального масштабирования. Если имеется возможность задать обучающую выборку вручную для проверки надежности отношений, то можно использовать метод ReVerb, а также гибридный алгоритм TextRunner-R, комбинирующий ReVerb и TextRunner. В остальных случаях, если горизонтальное масштабирование и составление выборки для проверки надежности отношений не представляется возможным, остается сделать выбор: алгоритм TextRunner или KnowItAll.

Заключение

В статье на основе известных знаний в области обработки естественного языка [1–26] предлагается классификация методов извлечения отношений

из текстовых данных согласно пяти базовым подходам: полуавтоматическое извлечение информации, обучение с частичным привлечением учителя, с привлечением учителя, обучение с привлечением учителя и использованием внешних баз знаний, обучение без учителя. Классификацию и область применения методов извлечения информации из текстовых данных возможно представить в виде схемы, где узлам соответствуют группы методов извлечения информации и сами методы извлечения информации, а ребрам соответствуют условия поставленной задачи извлечения информации. На данный момент наибольший интерес представляют собой методы обучения с использованием внешних баз знаний ввиду открытости больших объемов текстовых данных во многих предметных областях и использования моделей многоклассовой классификации, учитывающих пересечения множеств отношений.

ЛИТЕРАТУРА

1. Ермаков Н.В., Молодяков С.А. Модель кэширования для системы быстрого доступа к файлам // Информационные технологии в управлении: материалы конференции.— Санкт-Петербург.— 7–8 октября 2020 г.— с.144–147.
2. Монастырев В.В., Молодяков С.А. Применение методов машинного обучения для анализа интересов пользователей // Современная наука: Актуальные проблемы теории и практики, Серия: Естественные и технические науки.— 2021.— № 1.— с.97–101.
3. Константинова Н.С., Митрофанова О.А. Онтологии как системы хранения знаний // Всероссийский конкурсный отбор обзорно-аналитических статей по приоритетному направлению «Информационно-телекоммуникационные системы».— 2008.
4. Min Song, Won Chul Kim, Dahee Lee, Go Eun Heo, Keun Young Kang. PKDE4J: Entity and relation extraction for public knowledge discovery // Journal of Biomedical Informatics.— 2015.— Vol. 57.— P. 320–332,
5. Шелманов А.О., Исаков В.А., Станкевич М.А., Смирнов И.В. Открытое извлечение информации из текстов // Искусственный интеллект и принятие решений.— 2018.— № 2.— с.47–61.
6. Найханова Л.В. Основные типы семантических отношений между терминами предметной области // Известия ВУЗов. Поволжский регион. Технические науки.— 2008.— № 1.— с.62–71.
7. Vizureanu P. Introductory Chapter: Enhanced Expert System — A Long-Life Solution // Enhanced Expert Systems.— April 2019.— № 1.
8. Stephen Roller, Douwe Kiela, Maximilian Nickel. Hearst Patterns Revisited: Automatic Hypernym Detection from Large Text Corpora // Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia.— 2018.— Vol.2.— P. 358–363.
9. Ellen Riloff, William Phillips. An Introduction to the Sundance and AutoSlog Systems.— School of Computing University of Utah.— Salt Lake City, USA.— 2004.
10. Большакова Е.И., Воронцов К.В., Ефремова Н.Э., Клышинский Э.С., Лукашевич Н.В., Сапин А.С. Автоматическая обработка текстов на естественном языке и анализ данных. Изд-во НИУ ВШЭ.— 2017.— 269 с.
11. Brin S. Extracting Patterns and Relations from the World Wide Web. // The World Wide Web and Databases. WebDB1998. Lecture Notes in Computer Science,— Vol 1590.— Springer, Berlin, Heidelberg.
12. Eugene A., Luis G. Snowball: Extracting Relations from Large Plain-Text Collections // Proceedings of the Fifth ACM Conference on Digital Libraries (DL).— 2000.
13. Banko M, Cafarella M.J., Soderland S., Broadhead M., Etzioni O. Open Information Extraction from the Web // Proceedings of Communications of the ACM conference.— December 2008.— Vol.51(12).— P. 68–74.
14. Fader A., Soderland S., Oren Etzioni. Identifying Relations for Open Information Extraction // Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing.— Edinburgh, Scotland, UK,— 2011.— P. 1535–1545.
15. Bach N., Badaskar S.A Review of Relation Extraction // Semantic Scholal.— 2007. URL: http://www.cs.cmu.edu/~nbach/papers/A-survey-on-Relation-Extraction.pdf (accessed on: 12.02.2021).
16. Xue L., Qing S., Pengzhou Z. Relation Extraction Based on Deep Learning // 2018 IEEE/ACIS17th International Conference on Computer and Information Science (ICIS), 6–8 June 2018.
17. Li Z., Yang J., Gou X., Qi X. Recurrent neural networks with segment attention and entity description for relation extraction from clinical texts // Artificial Intelligence in Medicine.— Vol. 97.— June 2019.— P. 9–18.
18. Jung H., Choi S., Lee S., Song S. Survey on Kernel-Based Relation Extraction // IntechOpen, 2012. URL: https://www.intechopen.com/books/theory-and-applications-for-advanced-text-mining/survey-on-kernel-based-relation-extraction#fn1 (accessed on: 01.02.2021).
19. Bui Q.C. Relation extraction methods for biomedical literature: PhD thesis, Informatics Institute, Faculty of Science, Universiteit van Amsterdam.— 2012.
20. Mintz M., Bills S., Snow R., Jurafsky D. Distant supervision for relation extraction without labeled data // Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP.— Suntec, Singapore.— August 2009.— P. 1003–1011.
21. Riedel S., Yao L., McCallum A. Modeling Relations and Their Mentions without Labeled Text // Joint European Conference on Machine Learning and Knowledge Discovery in Databases.— Berlin, Heidelberg.— 2010.— P. 148–163.
22. Surdeanu M., Tibshirani J., Nallapati R., Manning C.D. Multi-instance Multi-label Learning for Relation Extraction // Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning.— Jeju Island, Korea.— July 2012,— P. 455–465.

23. Jiang X., Wang Q., Li P., Wang B. Relation Extraction with Multi-instance Multi-label Convolutional Neural Networks // Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers.— Osaka, Japan.— December 2016,— P. 1471–1480.
24. Hasegawa T., Sekine S., Grishman R. Discovering relations among named entities from large corpora // Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics.— P. 415–422.
25. Chen J., Ji D., Tan C.L., Niu Z. Unsupervised feature selection for relation extraction // Proceedings of IJCNLP.— 2005,— P. 262–267.
26. Lin D., Pantel P. Discovery of Inference Rules from Text // Natural Language Engineering.— Vol.7(04).—2001,— P. 323–328.

Санкт-Петербургский политехнический университет Петра Великого