

Федюкин Д.А.

Кудрявцев С.Д.

Храмцов Д.А.

Гапанюк Ю.Е., кандидат технических наук, доцент

(Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет))

УДК 004.62

МОДУЛЬ ПЕРЕВОДА ЗНАНИЙ ИЗ ПЛОСКИХ ГРАФОВ В МЕТАГРАФОВОЕ ПРОСТРАНСТВО

В статье рассмотрены способы перевода знаний, представленных в формате RDF-триплетов, в метаграфовое пространство с выделением метавершин и атрибутов.

Ключевые слова: метаграф, машинное обучение, марковские алгоритмы, кластеризация, метавершины, атрибуты.

Fedyukin D.A.

Kudryavcev S.D.

Khramtsov D.A.

Gapaniuk Y.E.

MODULE FOR TRANSLATING KNOWLEDGE FROM FLAT GRAPHS INTO METAGRAPH SPACE

The article considers the ways of translating the knowledge presented in the RDF-triplet format into the metagraph space with the allocation of metaverices and attributes.

Keywords: metagraph, machine learning, Markov algorithms, clusterization, metaverices, attributes.

Введение

В настоящее время модели на основе сложных сетей все чаще используются для решения широкого класса прикладных задач. Одной из таких моделей, ориентированных на описание сложных вложенных контекстов, является метаграфовая модель [1]. Ее основным преимуществом является возможность выделения эмерджентных фрагментов графа и объединение их в метавершины.

Для демонстрации преимуществ использования сложных сетей в решении прикладных задач необходимы подготовленные наборы данных (датасеты). В настоящее время большинство графовых датасетов используют в качестве формата хранения RDF.

Формат RDF позволяет описывать плоские графы. В данной статье рассматривается способ построения иерархической метаграфовой модели на основе плоской модели RDF.

Краткое описание метаграфовой модели

Под метаграфом будем понимать следующую структуру: $MG = \langle V, MV, E \rangle$, где MG – метаграф; V – множество вершин метаграфа; MV – множество метавершин метаграфа; E – множество ребер метаграфа.

Вершина метаграфа характеризуется множеством атрибутов: $v_i = \{atr_k\}$, $v_i \in V$, где v_i – вершина метаграфа; atr_k – атрибут.

Ребро метаграфа характеризуется множеством атрибутов, исходной и конечной вершиной и признаком направленности: $e_i = \langle v_s, v_e, \theta, \{atr_k\} \rangle, e_i \in E, \theta = true \mid false$, где e_i – ребро метаграфа; v_s – исходная вершина (метавершина) ребра; v_e – конечная вершина (метавершина) ребра; eo – признак направленности ребра ($eo = true$ – направленное ребро, $eo = false$ – ненаправленное ребро); atr_k – атрибут.

Фрагмент метаграфа: $MG_i = \{ev_j\}, ev_j \in (V \cup E \cup MV)$, где MG_i – фрагмент метаграфа; ev_j – элемент, принадлежащий объединению множеств вершин (метавершин) и ребер метаграфа.

Таким образом, фрагмент метаграфа в общем виде может содержать произвольные вершины (метавершины) и ребра. Метавершина метаграфа является основным элементом предлагаемой модели: $mv_i = \langle \{atr_k\}, MG_j \rangle, mv_i \in MV$, где mv_i – вершина метаграфа; atr_k – атрибут, MG_j – фрагмент метаграфа.

Метавершина в дополнение к свойствам вершины включает вложенный фрагмент метаграфа. Наличие у метавершин собственных атрибутов и связей с другими вершинами является важной особенностью метаграфов. Это соответствует принципу эмерджентности, то есть приданию понятию нового качества, несводимости понятия к сумме его составных частей. Фактически, как только вводится новое понятие в виде метавершины, оно «получает право» на собственные свойства, связи и т.д., так как в соответствии с принципом эмерджентности новое понятие обладает новым качеством и не может быть сведено к подграфу базовых понятий. Метаграф можно охарактеризовать как «сеть с эмерджентностью», то есть фрагмент сети, состоящий из вершин и связей, который может выступать как отдельное целое.

Рассмотрим способ преобразования плоского графа из формата RDF в метаграфовую модель.

Структура входных и выходных данных

В качестве формата хранения данных будем использовать формат JSON. Для представления входных данных и последующей их обработки, введем JSON-структуру, показанную на рис. 1а. Данная модель имеет четыре поля, два из которых могут оставаться пустыми. В поле `bigraph` происходит заполнение RDF триплетов.

Выходные данные также представляются в виде JSON-структуры, которая отражает все возможные элементы метаграфа (показана на рис. 1б).

```
{
  "oriented":false
  "attributes": [attr1, attr2]
  "bigraph": [
    {"subject":1, "predicate": "relation1", "object":2},
    {"subject":2, "predicate": "relation2",
      "object":"hjkdfg"}
  ]
  "ban_list": ["ban_vert_1", "ban_vert_2"]
}
```

```
{
  "oriented":false
  "vertices": [
    {"name": 'a', "parent": 'b'},
    {"name": 'c', "parent": 'b'},
    {"name": 'b', "parent": None}
  ],
  attributes: [
    {"name": 'atr_name', "value": '123', "for_vert": 'a'}
  ],
  edges: [
    {"name": 'ac', "source": 'a', "direction": 'c'}
  ]
}
```

а. Формат входных данных

б. Формат выходных данных

Рис. 1. Представление метаграфа в виде JSON-структуры.

Формирование метаграфа

Для формирования метаграфа применялись два основных концептуальных подхода: это кластеризация «в длину» и «в ширину».

Для кластеризации в ширину применяется алгоритм выявления похожих вершин, исходя из связей, идущих к ним от общего соседа. Вершины объединяются в метавершины тогда, когда их связи с общим предком одинаковы. При этом название метавершины формируется из названия ребра.

Для кластеризации в длину использовался алгоритм марковской кластеризации [2]. Суть алгоритма заключается в том, что на вход подается неориентированный граф, параметр мощности ϵ и параметр инфляции γ . Далее вычисляется матрица вероятностей. На основе матрицы вероятностей начинается случайное блуждание, которое стимулируется чередованием двух операций. Первая операция «вздутие» состоит в том, что для каждой вершины значения перехода изменяются таким образом, что сильные соседние значения усиливаются, а большие соседние значения понижаются, за это отвечает параметр « γ ». Вторая операция «расширение» помогает сделать более удаленные узлы или соседей доступными, за это отвечает параметр « ϵ ». Алгоритм состоит из трех основных шагов:

Шаг 1: Расширяем матрицу, используя параметр « ϵ ».

Шаг 2: Раздуваем полученную матрицу, используя параметр « γ ».

Шаги 1 и 2 повторяются до тех пор, пока не будет достигнута конвергенция, т.е. не будет наблюдаться существенных изменений значений.

Для того, чтобы воспользоваться плюсами обоих подходов к кластеризации, сначала выполняется марковская кластеризация в длину, а потом кластеризация в ширину.

Для экспериментов были взяты датасеты, представляющие собой wiki-страницы портала WikiHow, представленные в виде RDF триплетов.

Анализ результатов

Первичный анализ показал, что алгоритм может выделять метавершины. Иллюстрация работы алгоритма приведена на рис. 2. На нем видно, что вершины, связанные предикатом «has_parent», объединены в единую метавершину, а также алгоритм Марковской кластеризации выделяет в отдельную метавершину все вершины, распознавая связь, идущую сквозь несколько вершин.

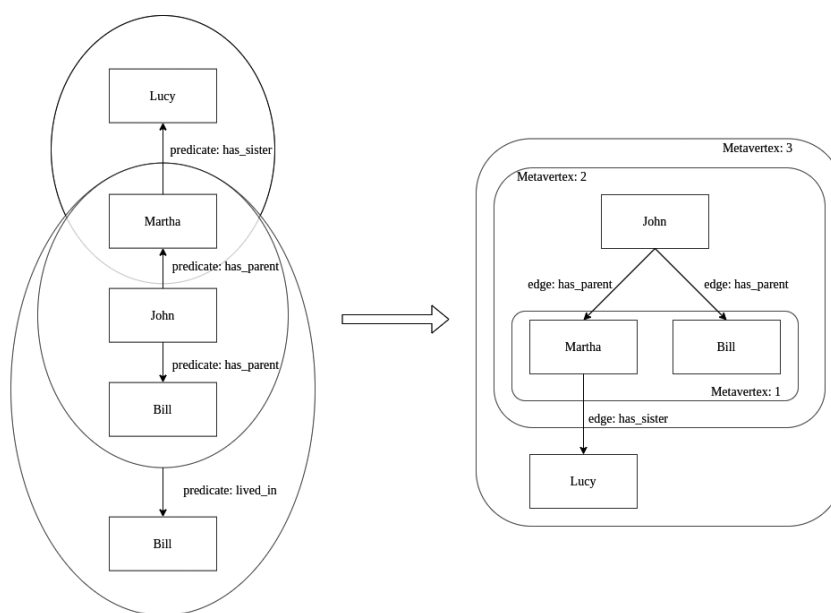


Рис. 2. Иллюстрация работы алгоритма

В дальнейшем предполагается работать над добавлением методов для выделения знаний в метаграфах, например, можно прогнозировать связь между вершинами разных уровней иерархии или выделять кластеры вершин более высокого уровня. Для описания данных предполагается использовать предикатное описание в соответствии с [3].

Заключение

В результате работы был создан модуль для перевода знаний в метаграфовое пространство. Специально для этого были разработаны структуры данных на основе формата JSON. Были применены алгоритмы интеллектуального выделения вершин в плоских графах.

ЛИТЕРАТУРА

1. Черненький В.М., Гапанюк Ю.Е., Ревунков Г.И., Терехов В.И., Каганов Ю.Т. Метаграфовый подход для описания гибридных интеллектуальных информационных систем. Прикладная информатика. 2017. № 3 (69). Том 12. С. 57–79.
2. S. van Dongen. A cluster algorithm for graphs. Technical Report INS-R0010, National Research Institute for Mathematics and Computer Science in the Netherlands, Amsterdam, May 2000.
3. Гапанюк Ю.Е., Ревунков Г.И., Федоренко Ю.С. Предикатное описание метаграфовой модели данных. Информационно-измерительные и управляющие системы. 2016. Выпуск № 12. С. 122–131.