УДК 004.6

Обзор подходов к векторному представлению графов знаний и метаграфов

Overview of approaches of knowledge graphs and metagraphs embeddings

1. **Гапанюк Ю.Е. (Gapanyuk Yu.E.), доцент кафедры «Системы обработки информации и управления» МГТУ им. Н.Э. Баумана, gapyu@bmstu.ru**
2. **Ревунков Г.И. (Revunkov G.I.), доцент кафедры «Системы обработки информации и управления» МГТУ им. Н.Э. Баумана (Bauman Moscow State Technical University), revunkov@bmstu.ru**
3. **Злобина С.В. (Zlobina S.V.), студент кафедры «Системы обработки информации и управления» МГТУ им. Н.Э. Баумана (Bauman Moscow State Technical University), svetlanazlobina97@gmail.com**
4. **Кадиев З.Д. (Kadiev Z.D.), студент кафедры «Системы обработки информации и управления» МГТУ им. Н.Э. Баумана (Bauman Moscow State Technical University), zz600571@gmail.com**

1. Введение

В настоящее время модели на основе сложных сетей находят все более широкое применение в различных областях науки от математики и информатики до биологии и социологии. Основополагающими русскоязычными статьями являются работы И.А. Евина [1], О.П. Кузнецова и Л.Ю. Жиляковой [2]. Профессор К.В. Анохин [3] предлагает рассматривать сложные сети как основу для построения комплексных биологических моделей.

В настоящее время термины «сложная сеть» или «комплексная сеть» (которые являются различными переводами англоязычного термина «complex network») и термин «сложный граф» (англ. «complex graph») часто употребляются как синонимы. В работе [4, стр. 14] отмечается, что термин «сложная сеть», как правило, употребляется для обозначения реальной исследуемой системы, в то время как термин «сложный граф» обычно используют для обозначения математической модели такой системы.

Наибольшие разночтения вызывает термин «сложный» применительно к графовым моделям. Как правило, термин «сложный» трактуется в двух вариантах:

I. Плоские графы (сети) очень большой размерности. Такие сети могут включать миллионы и более вершин. Ребра, соединяющие вершины, могут быть ненаправленными или направленными. Иногда используется модель мультиграфа, в этом случае две вершины могут соединяются не одним, а несколькими ребрами. Именно такую модель в литературе чаще всего называют «сложной сетью». Важно, что в рамках данного подхода «сложная сеть» остается плоским графом (мультиграфом).

II. Сложные графы, в которых используется сложное (комплексное) описание вершин, ребер и/или их расположения. Часто в таких моделях используется не плоский, а пространственный вариант расположения вершин и ребер. Именно подобный подход может быть наиболее полезен при описании сложных моделей данных и знаний. На сегодняшний день наиболее широко известны три подобных модели: гиперграф, гиперсеть и метаграф. В данной статье мы будем говорить только о метаграфовой модели.

В настоящее время во многих прикладных задачах необходимо рассматривать сложные графы очень большой размерности (включающие миллионы вершин). Обработка плоского графа или мультиграфа (сложный граф I типа) является чрезвычайно сложной задачей как по памяти, так и по вычислительным ресурсам. В случае обработки сложных графов типа II задача еще более усложняется за счет комплексного описания моделей.

Одним из наиболее современных подходов к решению задачи обработки графов большой размерности является отказ от традиционного представления графа в виде множества вершин и множества ребер непосредственно при обработке графа, что показано на рис. 1.

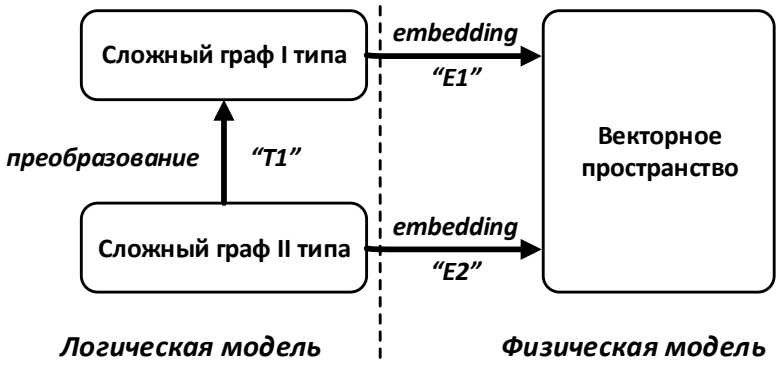

Рис. 1. Модель сложного графа с точки зрения его обработки.

Модель сложного графа с точки зрения его обработки рассматривается на двух уровнях. По аналогии с моделями данных реляционных СУБД, эти уровни можно назвать «логической» и «физической» моделью сложного графа.

Логическая модель сложного графа I типа – это традиционная модель плоского графа или мультиграфа, включающая множества вершин и ребер.

В настоящее время в качестве физической модели представления графов традиционно используются непрерывные векторные пространства. При этом нет никаких ограничений на использование других видов пространств. Операция преобразования графа в векторное пространство называется «векторным представлением» (связь “E1” на рис. 1). В англоязычной литературе для обозначения такого преобразования традиционно используется термин «embedding», то есть «встраивание» графа в векторное пространство.

Логическая модель сложного графа II типа может быть или непосредственно преобразована в физическое представление (связь “E2” на рис. 1) или предварительно преобразована в модель сложного графа I типа (связь “T1” на рис. 1) а затем преобразована в физическое представление (связь “E1” на рис. 1).

Операция «векторного представления» широко используется в машинном обучении, в частности, для обработки текстов. Хорошо известны алгоритмы векторного представления для текстов: Word2Vec, Glove и др. Часто англоязычный термин не переводят и в русском варианте используют термин «эмбеддинг», как синоним термина «векторного представления». Далее в статье мы также будем использовать термин «эмбеддинг».

Необходимо отметить, что в отличие от реляционной СУБД, где преобразование «логической» модели в «физическую» является относительно несложной операцией, эмбеддинг графа является достаточно сложной процедурой, как правило, связанной с решением задач оптимизации.

Необходимо также отметить, что в отличие от реляционной СУБД, где преобразование «логической» модели в «физическую» является однозначным отображением, эмбеддинг графа можно рассматривать как типичную связь «один-ко-многим», на основе одной логической модели с помощью различных методов можно сформировать различные результирующие векторные пространства физической модели. Полученные физические модели, как правило, оптимизированы для выполнения конкретных алгоритмов над графами. В зависимости от задачи, может производиться эмбеддинг только вершин графа, комбинированный эмбеддинг вершин и ребер графа, параллельный эмбеддинг вершин и ребер графа в различные векторные пространства.

В данной статье мы рассмотрим основные подходы, которые на сегодняшний день используются для эмбеддинга графов (связь “E1” на рис. 1). Также мы рассмотрим способы преобразования метаграфовой модели к модели плоского графа (связь “T1” на рис. 1) с тем, чтобы применить к преобразованной модели рассмотренные процедуры эмбеддинга.

Значительная часть наиболее простых техник эмбеддинга основана на том факте, что ребрам графа приписывается определенная числовая метрика, которую можно трактовать как расстояние между вершинами графа или пропускную способность каналов, соединяющих вершины.

В данной статье мы сконцентрируемся на эмбеддинге графов и метаграфов знаний. В соответствии с [5] существуют различные определения графа знаний. Наиболее простым является следующее определение [5]: «Графы знаний представляют собой большие сети (large networks), содержащие сущности, их семантические типы, отношения между сущностями и свойства». Сущности являются аналогом вершин в обычном графе. Отношения между сущностями (аналог ребер в обычном графе) могут быть аннотированы свойствами, но такие свойства совсем не обязательно должны быть числовыми. Таким образом, задача эмбеддинга

графа знаний сложнее, чем задача эмбеддинга обычного графа, потому что в случае графа знаний отсутствует числовая метрика ребер графа.

2. Особенности эмбеддинга графов знаний

Изложение механизмов эмбеддинга графов знаний в данном и следующих разделах в основном опираются на обзор [6], а также на статьи, посвященные конкретным алгоритмам эмбеддинга.

В соответствии с [6], эмбеддинг графа знаний – это математическое преобразование графа в вектор или в набор векторов в заданном векторном пространстве. Его можно проводить отдельно для вершин графа, для вершин и ребер графа, и даже для всего графа целиком. В первых двух случаях результатом будет набор векторов, в последнем – один вектор для целого графа. Главное условие состоит в том, что такое преобразование должно адекватно передавать семантику и топологию исходного графа. Эмбеддинг позволяет получить сразу несколько преимуществ при работе с графом знаний.

Первое преимущество заключается в оптимизации использования памяти, ведь вектор – это сжатое представление информации из графа. Представим себе матрицу смежности размера $V \times V$, где V – количество вершин в графе. Для действительно больших графов использование матрицы смежности напрямую становится сложно выполнимым, ведь для миллиона вершин размерность матрицы смежности составляет миллион в квадрате ячеек. Эмбеддинг в данном случае является более применимым решением, так как представляет узлы графа с помощью векторов гораздо меньшей размерности.

Второе преимущество состоит в том, что мы получаем возможность использовать уже накопленный математический аппарат для методов машинного обучения, который позволяет работать с векторами.

Третье преимущество состоит в производительности результирующих алгоритмов, использующих физическое представление графа знаний. В этом случае выполнение операций

над векторами более производительно, чем выполнение операций над традиционной графовой моделью в виде множеств вершин и ребер. Кроме того, на сегодняшний день видеокарты позволяют еще больше ускорить работу с векторами.

Представим, что у нас есть граф знаний, состоящий из n сущностей и m отношений между ними. Знания в этом графе хранятся в виде набора триплетов $D+=\{\langle h,r,t\rangle\}$. Каждый триплет состоит из субъекта $h\in E$ (обозначение h от англ. «head entity» – левая часть триплета), объекта $t\in E$ (обозначение t от англ. «tail entity» – правая часть триплета) и отношения между ними $r\in R$. Здесь E – множество всех сущностей, R – множество всех отношений в графе.

Процесс эмбеддинга графа знаний можно условно разделить на три этапа:

1. Выбор того, каким образом будут представлены в векторном пространстве сущности и отношения. На данном этапе происходит задание способа представления сущностей и отношений в непрерывном векторном пространстве. Сущности обычно представляются в виде векторов (точек в векторном пространстве), а отношения задаются в виде операторов в этом векторном пространстве. В зависимости от используемых методов, отношения могут быть заданы в виде векторов (например, вектор суммы или разности), матриц, тензоров, Гауссовских распределений и их смесей.
2. Задание функции правдоподобия преобразования (англ. scoring function). Для каждого триплета $\langle h,r,t\rangle$ задается функция правдоподобия $f_r(h,t)$. Чем больше значение этой функции, тем более вероятно то, что факт, описываемый триплетом, является истинным. Таким образом, в соответствии с функцией правдоподобия, факты, содержащиеся в графе знаний, набирают больше баллов, чем факты, которых нет в графе знаний.

3. Обучение модели эмбеддинга подразумевает решение оптимизационной задачи, заключающейся в максимизации суммарной функции правдоподобия для всех триплетов, содержащихся в графе знаний $D+$.

3. Обзор существующих техник эмбеддинга

На сегодняшний день существуют десятки различных техник эмбеддинга. Они различаются способами представления сущностей и отношений в векторном пространстве и способами задания функций правдоподобия. В соответствии с [6], все существующие модели эмбеддинга можно разделить на две группы: модели параллельного переноса и модели семантического соответствия.

Модели параллельного переноса (англ. «translational distance models») используют функции правдоподобия, основанные на измерении расстояния между двумя вершинами графа. Причем оператор перемещения из одной вершины в другую задается на основе отношения между этими узлами в исходном графе.

Модели семантического соответствия (англ. «semantic matching models») используют функции правдоподобия, основанные на семантических характеристиках вершин и отношений исходного графа.

3.1. Модели параллельного переноса

Модель TransE [7]**.** Эта модель представляет и узлы, и связи как векторы в одном и том же векторном пространстве. Причем оператор отношения рассматривается как вектор перемещения между субъектом и объектом (рис. 2). Это позволяет на интуитивном уровне уловить семантический смысл операции сложения над векторами: если к субъекту прибавить отношение, то результат будет примерно равен объекту: $h + r \approx t$.

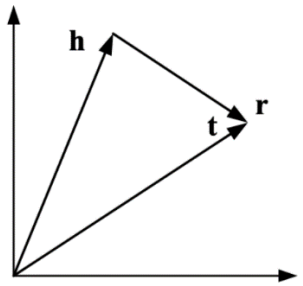

Рис. 2. Модель TransE. Пространство сущностей и отношений.

Функция правдоподобия определяется как отрицательное расстояние между $h+r$ и t, взятое по норме L1 или L2: $f_r(h,t) = -\|h+r-t\|_{1/2}$. Предполагается, что значение функции тем больше, чем больше вероятность того, что описываемый триплетом $\langle h,r,t \rangle$ факт является истинным.

Основным недостатком модели TransE является описание отношений один-ко-многим и много-ко-многим. Например, если у триплетов совпадают значения h или t, то в этом случае будут сгенерированы почти совпадающие эмбеддинги.

Модель TransR [8]**.** Эта модель очень похожа на TransE, но позволяет преодолеть недостатки описания отношений один-ко-многим и много-ко-многим. Для этого модель использует два пространства, одно для встраивания сущностей, а второе для встраивания отношений. При этом, у сущности будет различные проекции на пространство отношений в зависимости от того, в каких отношениях используется данная сущность (рис. 3).

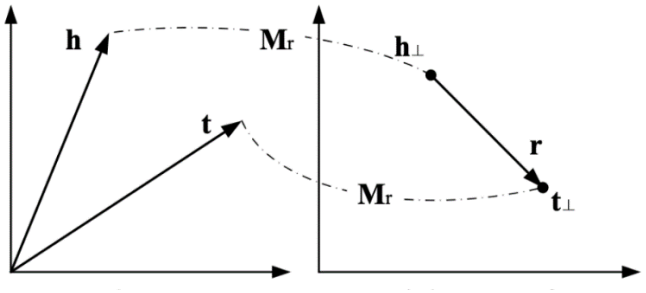

Рис. 3. Модель TransR. Слева пространство сущностей, справа пространство отношений

Модель TransR проецирует представления сущностей h и t на пространство отношений: $h_\perp = M_r h$, $t_\perp = M_r t$. Здесь $M_r \in \mathbb{R}^{k \times d}$ – проекционная матрица из пространства сущностей в пространство отношений.

Функция правдоподобия также использует проекции: $f_r(h,t) = -\left\| h_\perp + r - t_\perp \right\|_2^2$.

Основной недостаток модели TransR состоит в том, что из-за необходимости вычисления проекционной матрицы для каждого отношения, модель теряет простоту и эффективность TransE. Обучение модели TransR займет больше времени и ресурсов чем обучение модели TransE.

3.2. Модели семантического соответствия

Модель RESCAL [9]. Модель ассоциирует каждую сущность с вектором и старается передать скрытые смыслы (факторы) этой сущности. Каждое отношение представлено в виде матрицы, которая моделирует попарные взаимодействия скрытых факторов (рис. 4).

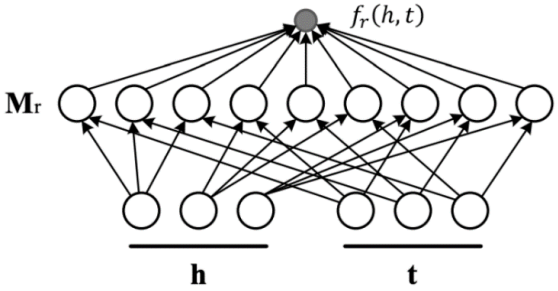

Рис. 4. Модель RESCAL

Функция правдоподобия определена как билинейная функция:

$$f_r(h,t) = h^T M_r t = \sum_{i=0}^{d-1} \sum_{j=0}^{d-1} [M_r]_{ij} \cdot [h]_i \cdot [t]_j ,$$

где $h, t \in \mathbb{R}^d$ – векторное представление сущностей; $M_r \in \mathbb{R}^{d \times d}$ – матрица, соответствующая отношениям. Функция правдоподобия в этом случае учитывает попарные взаимодействия компонентов h и t.

К недостаткам модели можно отнести ее вычислительную сложность, связанную с большой размерностью матрицы отношений M_r .

Модель DistMult [10]. Эта модель упрощает модель RESCAL, накладывая ограничение на матрицу отношений. В DistMult матрица отношений обязательно должна быть диагональной для каждого отношения r (рис. 5).

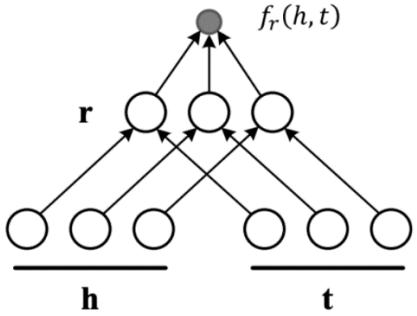

Рис. 5. Модель DistMult.

Функция правдоподобия определена следующим образом:

$$f_r(h,t) = h^T diag(r) t = \sum_{i=0}^{d-1} [r]_i \cdot [h]_i \cdot [t]_i ,$$

Такая функция отражает попарные взаимодействия только между компонентами сущностей, лежащими в одинаковых измерениях. За счет этого сложность модели становится линейной, а не квадратичной. Однако эта модель слишком сильно упрощена и может работать только с симметричными отношениями. Этого явно недостаточно для абстрактного графа знаний, в котором отношения могут быть произвольными.

Модель HolE [11]. Модель голографического эмбеддинга. Она совмещает в себе мощность RESCAL и простоту DistMult. В HolE к представлениям сущностей сначала применяется оператор взаимной корреляции:

$$[h * t]_i = \sum_{k=0}^{d-1} [h]_k \cdot [t]_{(k+i) \bmod d}$$

После этого, для определения значения функции правдоподобия, полученный вектор совмещается с вектором представления связи:

$$f_r(h,t) = r^T(h * t) = \sum_{i=0}^{d-1}[r]_i \sum_{k=0}^{d-1}[h]_k \cdot [t]_{(k+i) \bmod d}$$

Взаимная корреляция сжимает попарные взаимодействия компонентов сущностей (рис. 6). Поэтому сложность у модели линейная, что эффективнее, чем у RESCAL. Но при этом, модель HolE может работать с асимметричными отношениями, так как взаимная корреляция не коммутативна. Эта особенность недоступна в модели DistMult.

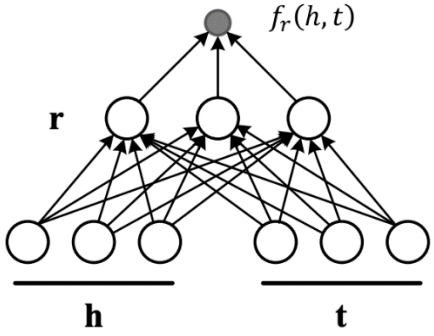

Рис. 6. Модель HolE.

4. Сравнение моделей эмбеддинга

Технику эмбеддинга обычно выбирают по совокупности показателей: сложности модели по требуемой памяти, времени обучения и получаемому качеству эмбеддинга.

В таблице 1 приведены на основе [6] сложности рассмотренных выше моделей по памяти и по времени (где n – количество сущностей, m – количество отношений, d – размерность пространства сущностей, k – размерность пространства отношений).

Таблица 1. Сложности моделей эмбеддинга [6].

Модель	**Сложность по памяти**	**Сложность по времени**
TransE	$O(nd + md)$	$O(d)$
TransR	$O(nd + mdk)$	$O(dk)$
RESCAL	$O(nd + md^2)$	$O(d^2)$
DistMult	$O(nd + md)$	$O(d)$

HolE	*O(nd + md)*	*O(d logd)*

Для оценки качества эмбеддинга используется следующий подход. Возьмем тестовую выборку, и из каждого триплета исключаем либо левую, либо правую часть, получая: $\langle ?, r, t \rangle$ или $\langle h, r, ? \rangle$. Далее необходимо выдать ранжированный список сущностей, которые наиболее подходят на место отсутствующей (список ранжируется по значению функции правдоподобия). Затем истинную сущность (которая стояла в этом триплете изначально) сравнивают с выданным списком и находят место, на котором она в этом списке значится. Очевидно, что чем ближе она к началу списка, тем лучше качество эмбеддинга модели.

В таблице 2 приведены результаты экспериментов по оценке качества для рассмотренных выше моделей с использованием тестового корпуса WN18 и фреймворка OpenKE (все модели обучены на 500 итерациях). В таблице Hit@10 – доля попадания правильной сущности в 10 первых элементов списка, Hit@3 и Hit@1 – в 3 и 1 первых элементов соответственно.

Таблица 2. Результаты экспериментов.

Модель	**Hit@10**	**Hit@3**	**Hit@1**
TransE	0,75	0,62	0,19
TransR	0,85	0,71	0,45
RESCAL	0,72	0,59	0,46
DistMult	0,93	0,91	0,66
HolE	0,94	0,93	0,92

Из таблицы 2 видно, что для тестового корпуса WN18 лучшие результаты показывает модель HolE.

5. Эмбеддинг метаграфов знаний

Метаграфовая модель детально рассмотрена в [12]. Основным элементом метаграфовой модели является метавершина. Метавершина в дополнение к свойствам вершины включает вложенный фрагмент метаграфа, который может также содержать вложенные вершины, метавершины, ребра.

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. Таким образом, метаграф можно охарактеризовать как «граф с эмерджентностью», то есть фрагмент графа, состоящий из вершин и связей, который может выступать как отдельное целое.

В нашей статье [13] показано, что метаграф может быть преобразован в многодольный плоский граф. Такое преобразование соответствует связи “T1” на рис. 1. Полученный плоский граф может быть преобразован в векторное представление с помощью рассмотренных выше моделей эмбеддинга.

Таким образом, логическая модель сложного графа II типа (метаграфовая модель) может быть предварительно преобразована в модель сложного графа I типа (связь “T1” на рис. 1) а затем преобразована в физическое представление (связь “E1” на рис. 1) с помощью рассмотренных выше моделей эмбеддинга для плоских графов.

Методы непосредственного эмбеддинга метаграфов в векторные пространства (связь “E2” на рис. 1) могут рассматриваться как предмет дальнейших исследований.

6. Выводы

Одним из современных подходов к решению задачи обработки графов большой размерности является отказ от традиционного представления графа в виде множества вершин и множества ребер непосредственно при обработке графа.

Модель сложного графа с точки зрения его обработки рассматривается на двух уровнях. По аналогии с моделями данных реляционных СУБД, эти уровни можно назвать «логической» и «физической» моделью сложного графа.

В настоящее время в качестве физической модели представления графов традиционно используются непрерывные векторные пространства. Операция преобразования графа в векторное пространство называется «векторным представлением» или «эмбеддингом».

На сегодняшний день существуют десятки различных техник эмбеддинга, которые можно разделить на две группы: модели параллельного переноса и модели семантического соответствия. Результаты экспериментов показывают, что наиболее удачной для эмбеддинга графов знаний является модель HolE, относящаяся ко второй группе.

Метаграф может быть преобразован в многодольный плоский граф и затем обработан с использованием моделей эмбеддинга для плоских графов.

Литература

1. Евин И.А. Введение с теорию сложных сетей //Компьютерные исследования и моделирование. 2010, Том 2, №2, с. 121-141.
2. Кузнецов О.П., Жилякова Л.Ю. Сложные сети и когнитивные науки // Нейроинформатика-2015. XVII Всероссийская научно-техническая конференция. Сборник научных трудов. Ч. 1. М.: МИФИ. 2015. С. 18.
3. Анохин К.В. Когнитом: гиперсетевая модель мозга // Нейроинформатика-2015. XVII Всероссийская научно-техническая конференция. Сборник научных трудов. Ч. 1. М.: НИЯУ МИФИ. 2015. С. 14-15.

4. Chapela V., Regino Criado, Santiago Moral, Miguel Romance. Intentional risk management through complex networks analysis. – Springer, 2015: SpringerBriefs in optimization.

5. Ehrlinger L., Wöß W. Towards a Definition of Knowledge Graphs. Joint Proceedings of the Posters and Demos Track of 12th International Conference on Semantic Systems – SEMANTiCS2016 and 1st International Workshop on Semantic Change & Evolving Semantics (SuCCESS16), 2016. Leipzig, Germany, Volume: 1695.

6. Wang Q., Mao Z., Wang B., Guo L. Knowledge Graph Embedding: A Survey of Approaches and Applications. IEEE Transactions on Knowledge and Data Engineering, 2017, vol. 29, no. 12, pp. 2724-2743.

7. Bordes A., Usunier N., Garcia-Duran A., Weston J., Yakhnenko O. Translating embeddings for modeling multirelational data. Proceedings of the 26th International Conference on Neural Information Processing Systems, 2013, vol. 2, pp. 2787–2795.

8. Lin H., Liu Y., Wang W., Yue Y., Lin Z. Learning Entity and Relation Embeddings for Knowledge Resolution. Procedia Computer Science, 2017, vol. 108, pp. 345-354.

9. Nickel M., Tresp V., Kriegel H. A Three-way Model for Collective Learning on Multi-relational Data. Proceedings of the 28th International Conference on International Conference on Machine Learning, 2011, pp. 809-816.

10. Yang B., Yih W., He X., Gao J., Deng L. Embedding Entities and Relations for Learning and Inference in Knowledge Bases, 2015. arXiv:1412.6575

11. Nickel M., Rosasco L., Poggio T. Holographic Embeddings of Knowledge Graphs. Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 2016, pp. 1955-1961.

12. Черненький В.М., Терехов В.И., Гапанюк Ю.Е. Структура гибридной интеллектуальной информационной системы на основе метаграфов. Нейрокомпьютеры: разработка, применение, 2016. Выпуск №9. С. 3-14.

13. Дунин И.В., Гапанюк Ю.Е., Ревунков Г.И. Особенности преобразования метаграфа в модель плоского графа. Динамика сложных систем — XXI век, 2018. Выпуск №9. С. 47-51.