



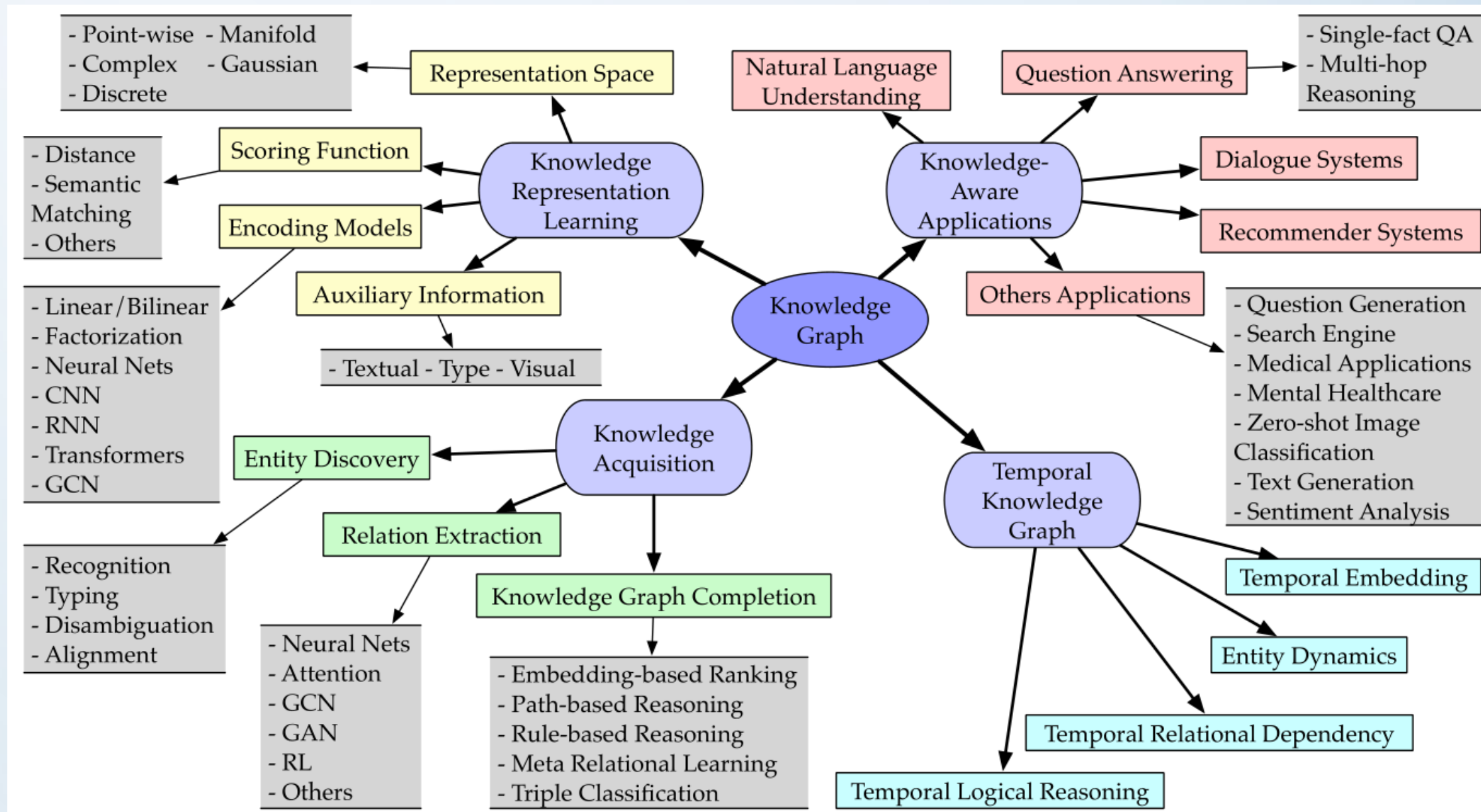
# Обработка графов знаний с использованием методов машинного обучения



# Графы знаний

- Графы знаний как средство улучшения искусственного интеллекта
- Граф знаний – это база знаний, использующая графовую (мульти, гипер-графовую) модель. Основная задача – обеспечение полноты знаний.
  - Тезаурус – специализированная база знаний, предназначенная для хранения (служебной) лингвистической информации.
  - Онтология – прежде всего схема данных, ориентированная на логический вывод. Основная задача – обеспечение непротиворечивости знаний.
- Наиболее известные проекты:
  - DBpedia
  - Yago
  - ConceptNet
    - Статья
  - Atomic 2020 (ориентирован на ситуации и причинно-следственные связи)
- Задачи, решаемые на графах знаний:
  - Предсказание вершин и связей.
  - Логический вывод на графе знаний (эта задача также решается на онтологиях)
  - Основой решения задач машинного обучения является embedding (векторное представление) графов знаний.
    - Статья «A Survey on Knowledge Graphs: Representation, Acquisition and Applications»  
<https://arxiv.org/abs/2002.00388>
    - Статья «Knowledge Graph Embedding: A Survey of Approaches and Applications»  
[https://www.researchgate.net/publication/319947524\\_Knowledge\\_Graph\\_Embedding\\_A\\_Survey\\_of\\_Approaches\\_and\\_Applications](https://www.researchgate.net/publication/319947524_Knowledge_Graph_Embedding_A_Survey_of_Approaches_and_Applications)

# Направления исследований





# Векторное представление графа знаний – 1

- Векторное представление (эмбеddинг) 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. Эмбеddинг позволяет получить сразу несколько преимуществ при работе с графом знаний:
  1. Первое преимущество заключается в оптимизации использования памяти, ведь вектор – это сжатое представление информации из графа. Для действительно больших графов использование матрицы смежности напрямую становится сложно выполнимым, ведь для миллиона вершин размерность матрицы смежности составляет миллион в квадрате ячеек. Эмбеddинг в данном случае является более применимым решением, так как представляет узлы графа с помощью векторов гораздо меньшей размерности.
  2. Второе преимущество состоит в том, что мы получаем возможность использовать уже накопленный математический аппарат для методов машинного обучения, который позволяет работать с векторами.
  3. 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.

# Векторное представление графа знаний – 2

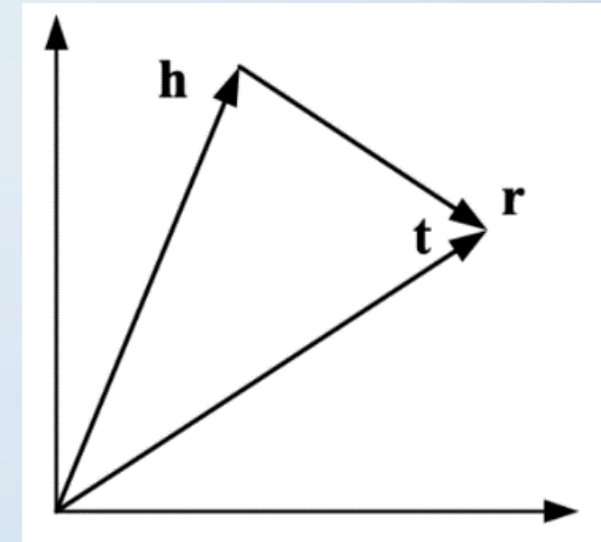
- Процесс эмбединга графа знаний можно условно разделить на три этапа:
  1. Выбор того, каким образом будут представлены в векторном пространстве сущности и отношения. На данном этапе происходит задание способа представления сущностей и отношений в непрерывном векторном пространстве. Сущности обычно 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 RDF-триплета  $\langle h, r, t \rangle$  задается функция правдоподобия  $f(h, t)$ . 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 эмбединга подразумевает решение оптимизационной задачи, заключающейся в максимизации суммарной функции правдоподобия для всех триплетов, содержащихся в графе знаний.

# Векторное представление графа знаний – 3

- На сегодняшний день существуют десятки различных техник эмбединга. Они различаются способами представления сущностей и отношений в векторном пространстве и способами задания функций правдоподобия. Основные модели эмбединга можно разделить на две группы: модели параллельного переноса и модели семантического соответствия:
  1. Модели параллельного переноса (англ. «translational distance models», TDM) 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.
  2. Модели семантического соответствия (англ. «semantic matching models», SMM) используют функции правдоподобия, основанные на семантических характеристиках вершин и отношений исходного графа.

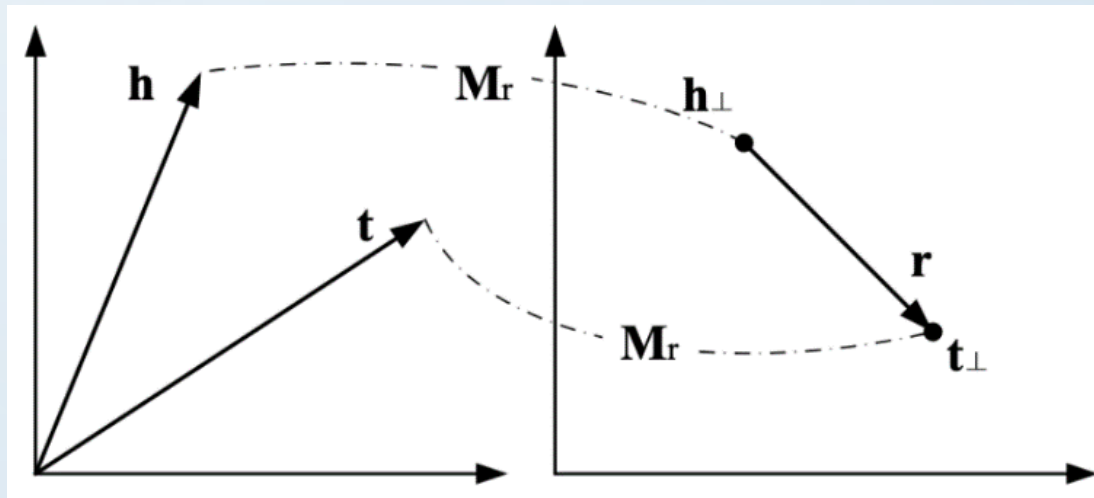
# Модель TransE (TDM)

- 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. Это позволяет на интуитивном уровне уловить семантический смысл операции сложения над векторами: если к субъекту прибавить отношение, то результат будет примерно равен объекту.
- Основным недостатком модели TransE является описание отношений один-ко-многим и много-ко-многим. Например, если у триплетов совпадают значения субъекта или объекта, то в этом случае будут сгенерированы почти совпадающие эмбединги.



# Модель TransR (TDM)

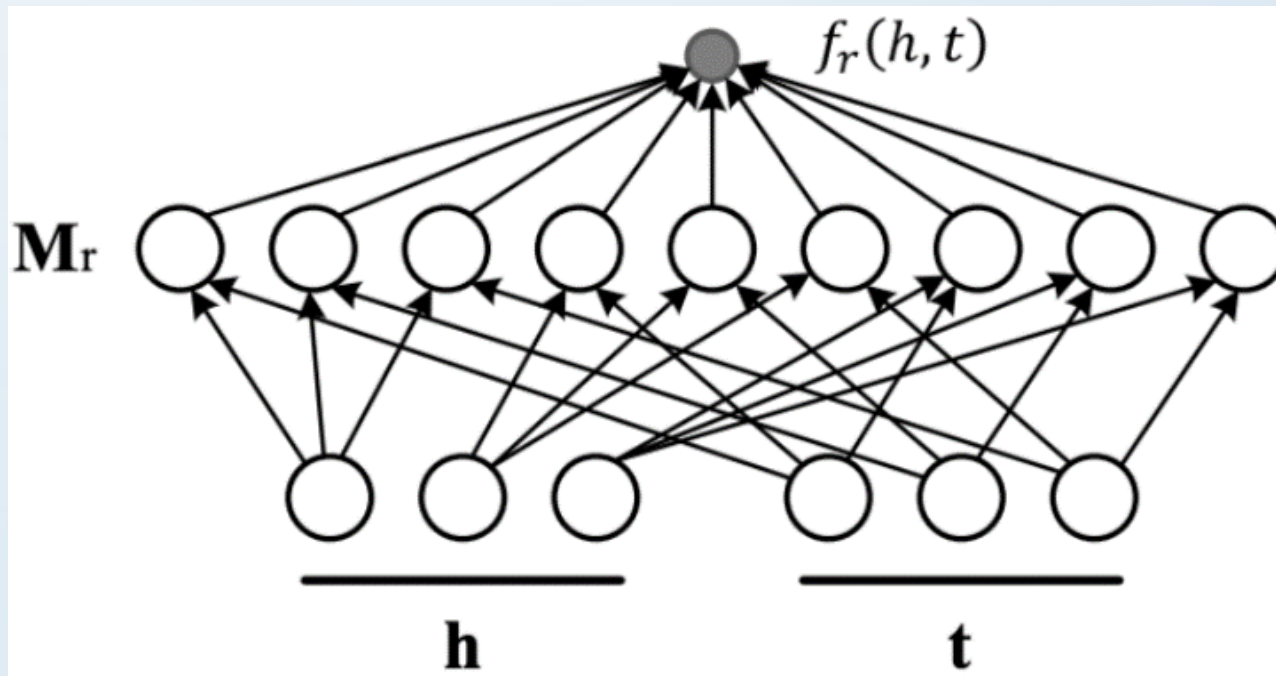
- Модель очень похожа на TransE, но позволяет преодолеть недостатки описания отношений один-ко-многим и много-ко-многим. Для этого модель использует два пространства, одно для встраивания сущностей, а второе для встраивания отношений. При этом, у сущности будут различные проекции на пространство отношений в зависимости от того, в каких отношениях используется данная сущность.
- 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.





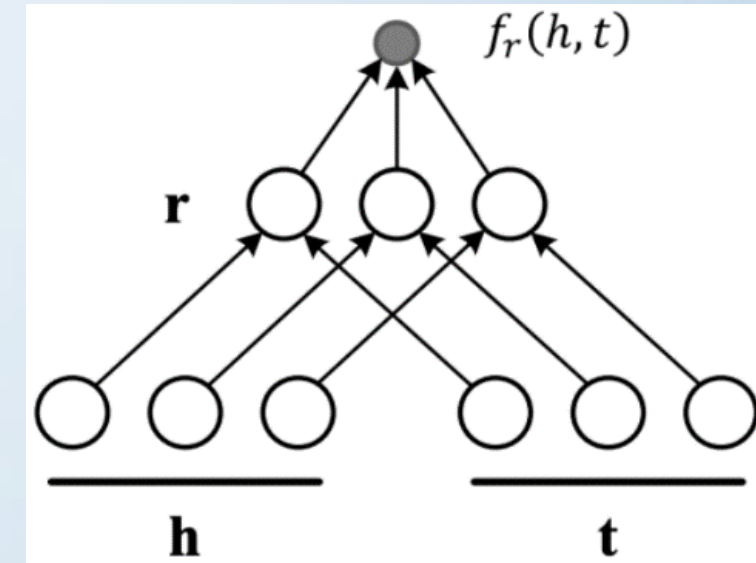
# Модель RESCAL (SMM)

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



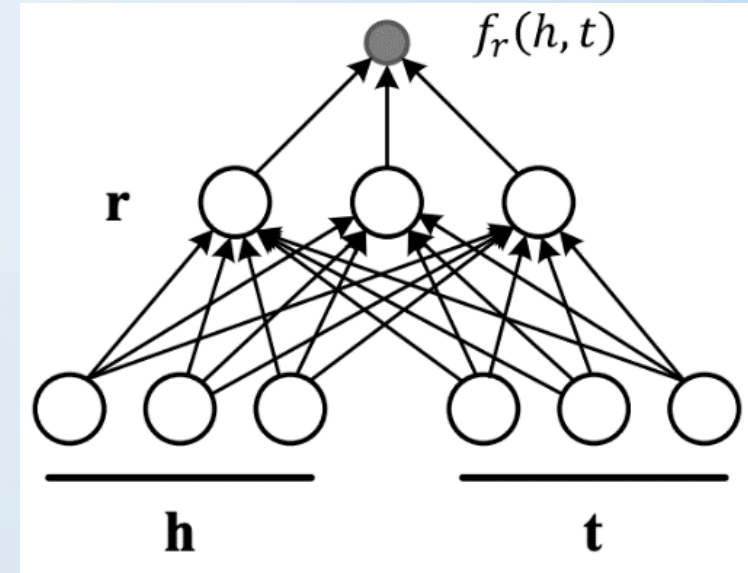
# Модель DistMult (SMM)

- Упрощает модель RESCAL, накладывая ограничение на матрицу отношений. В DistMult матрица отношений обязательно должна быть диагональной для каждого отношения  $r$ .
- Функция правдоподобия 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.



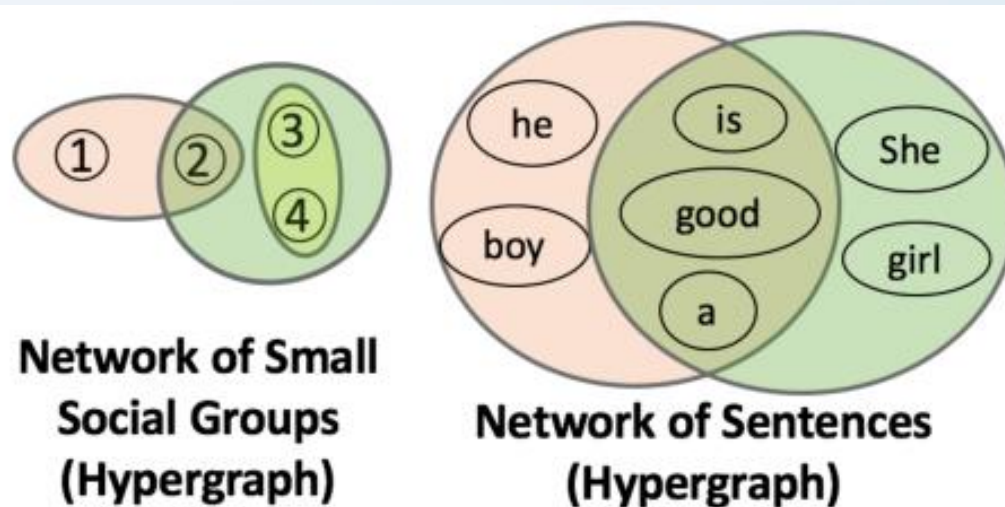
# Модель HoIE (SMM)

- Модель голографического эмбединга. Она совмещает в себе мощность RESCAL и простоту DistMult. В HoIE к представлениям сущностей сначала применяется оператор взаимной корреляции. После этого, для определения значения функции правдоподобия, полученный вектор совмещается с вектором представления СВЯЗИ.
- Взаимная корреляция сжимает попарные взаимодействия компонентов сущностей. Поэтому сложность у модели линейная, что эффективнее, чем у RESCAL. Но при этом, модель HoIE может работать с асимметричными отношениями, так как взаимная корреляция не коммутативна. Эта особенность недоступна в модели DistMult.



# Векторное представление гиперграфа – 1

- [Статья Hyperedge2vec: Distributed Representations for Hyperedges](#)

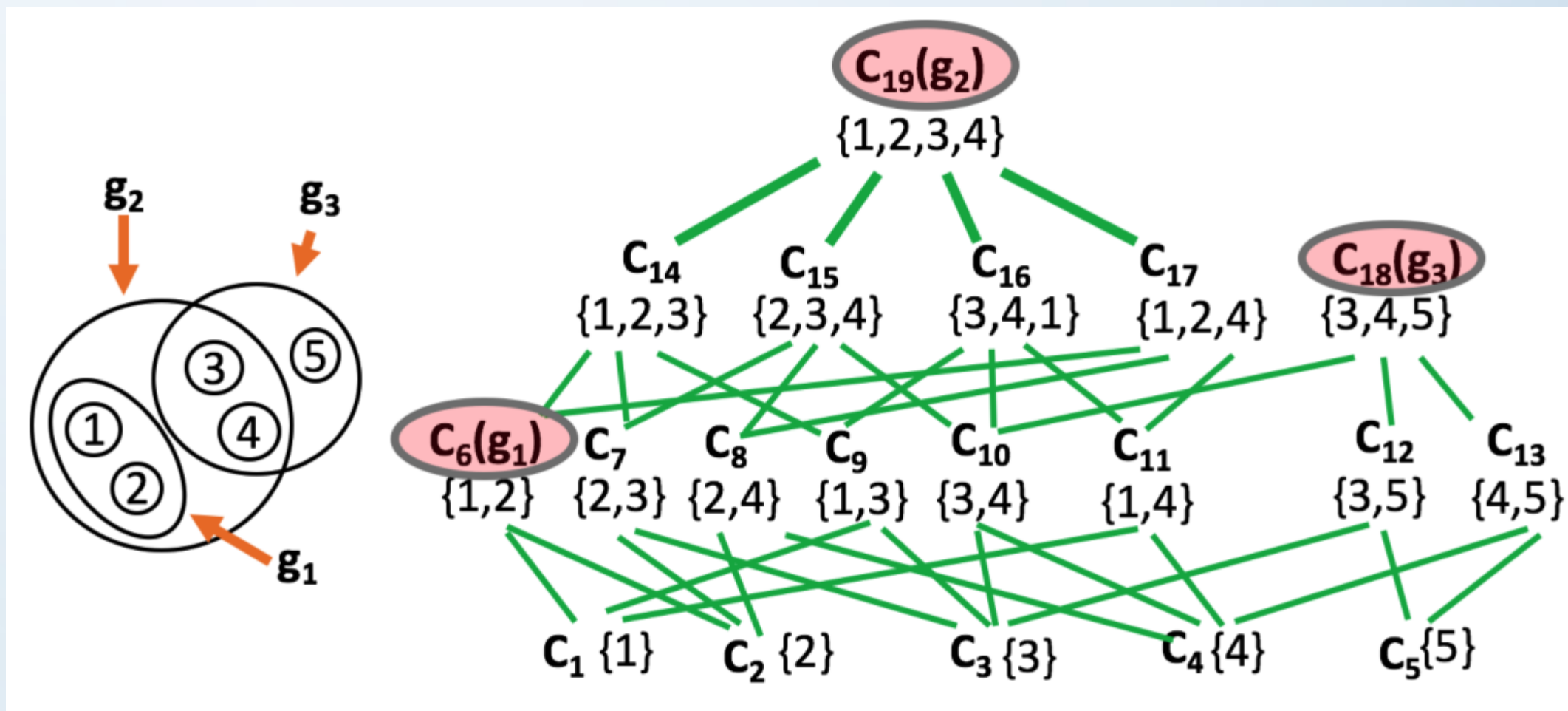


**Figure 1: Example illustrating “set-like” hypergraph structure from two domains. Left is a collaboration network between four individuals and on right is a network resulting from two sentences: “He is a good boy” & “She is a good girl”; and eight word nodes.**



# Векторное представление гиперграфа – 2

- Структура гиперребер может быть представлена в виде диаграммы Хассе.



# Векторное представление гиперграфа – 3

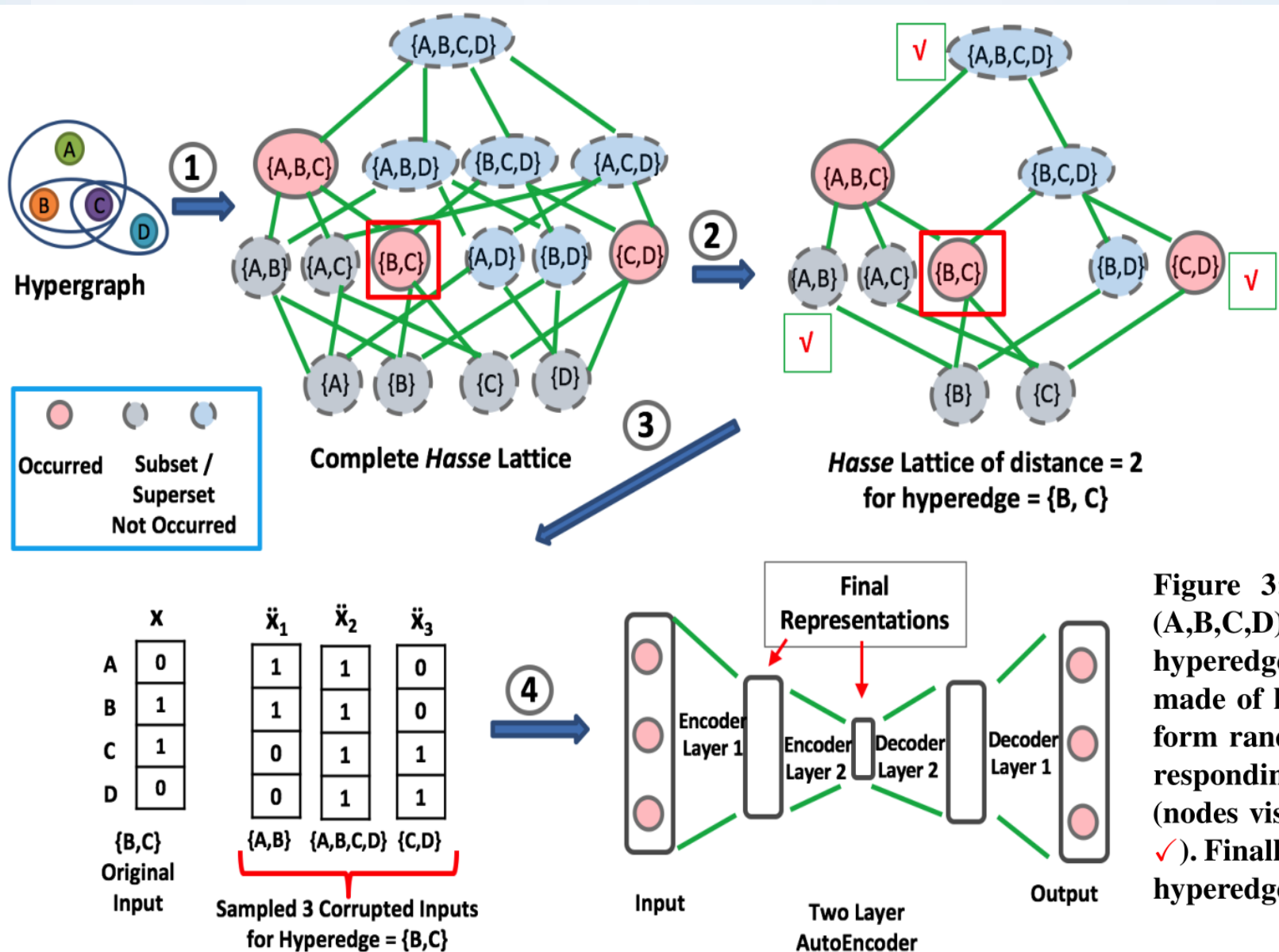


Figure 3: For the given hypergraph between four nodes (A,B,C,D) we consider the complete *hasse* lattice. For a given hyperedge {B,C} (square box) we then construct the sub-lattice made of hyperedges with distance  $h = 2$  from {B,C}. We perform random walk (with  $\tau = 0.2$ ) starting from the node corresponding to hyperedge {B,C} and sample  $p = 3$  hyperedges (nodes visited by the random walk; shown with a check-mark ✓). Finally, we train the autoencoder to reconstruct the original hyperedge from these  $p$  noisy hyperedges.

# Графовые нейронные сети (GNN)

- [Обзорная статья профессора А.Г. Дьяконова](#)
- [Общие принципы работы GNN](#)
- [Статья с примерами кода «What Are Graph Neural Networks? How GNNs Work, Explained with Examples»](#)
- [Серия статей «Graph Representation Learning»](#)
- Graph Convolutional Network (GCN)
  - [Краткая статья «Graph Convolutional Networks – Deep Learning on Graphs»](#)
  - [Детальная статья «Graph convolutional networks: a comprehensive review»](#)
- Graph Attention Network (GAT)
  - [Статья Understanding Graph Attention Networks \(GAT\)](#)
- Графовые трансформеры
  - [Статья с пояснением](#)
  - [Оригинальная статья](#)

# Гиперграфовые нейронные сети – 1

- [Статья «Hypergraph Neural Networks»](#)

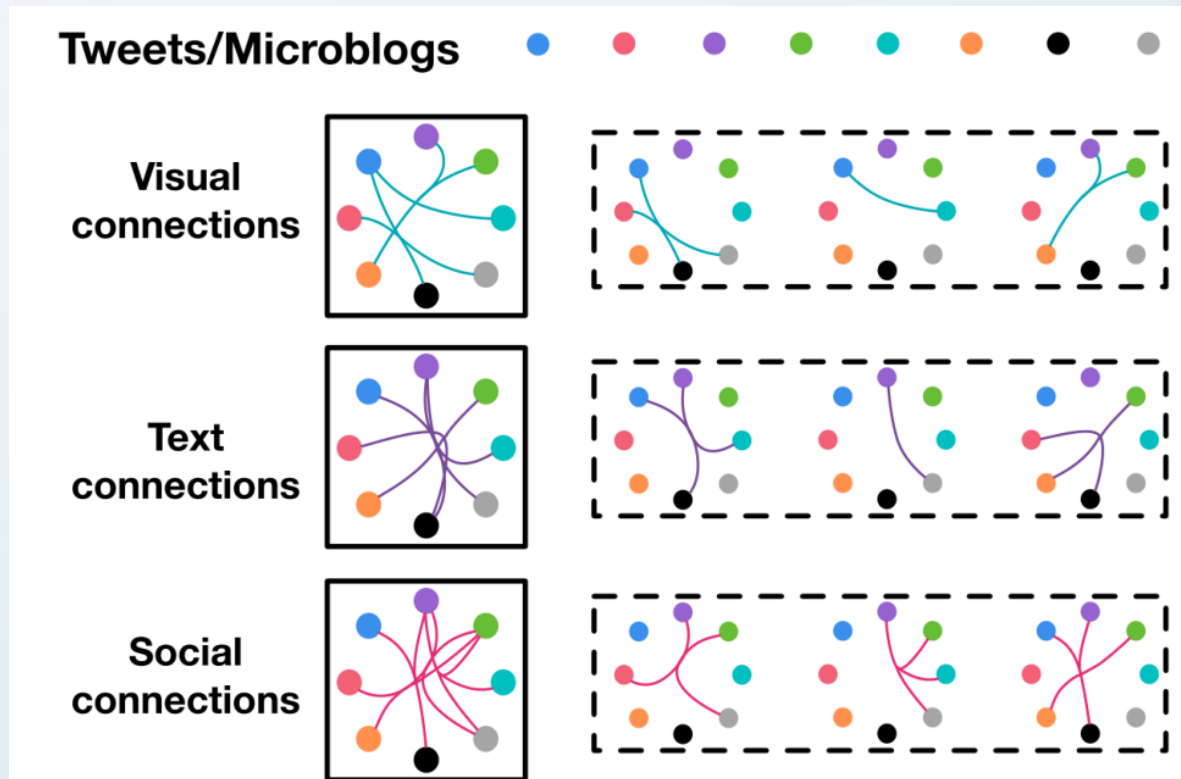
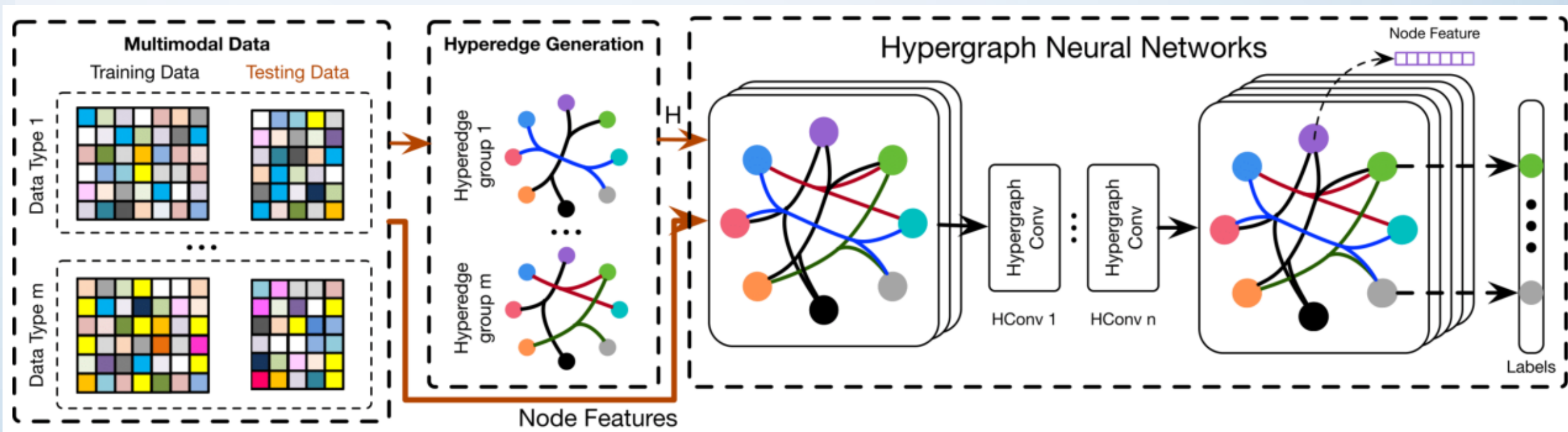


Figure 1: Examples of complex connections on social media data. Each color point represents a tweet or microblog, and there could be visual connections, text connections and social connections among them.



# Гиперграфовые нейронные сети – 2



1. Мультимодальный набор данных делится на обучающие и тестовые выборки, каждая выборка содержит набор вершин с их признаками.
2. Далее на основе выборок (с учетом их корреляции) генерируются группы гиперребер.
3. Группы гиперребер объединяются, чтобы сформировать общую матрицу смежности гиперграфа.
4. Матрица смежности и признаки вершин подаются на вход HGNN-сети, на выходе сети формируются метки выходных вершин.

# Гиперграфовые нейронные сети – 3

HGNN-сеть использует сверточные слои:

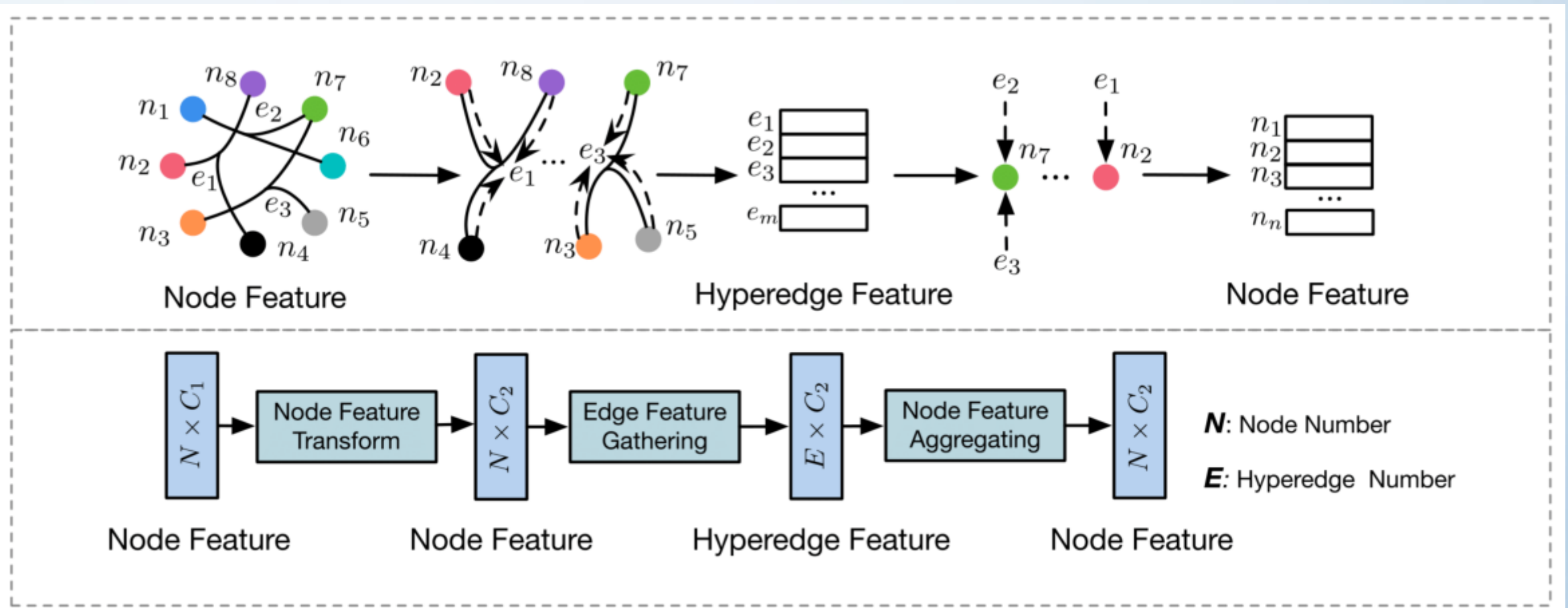
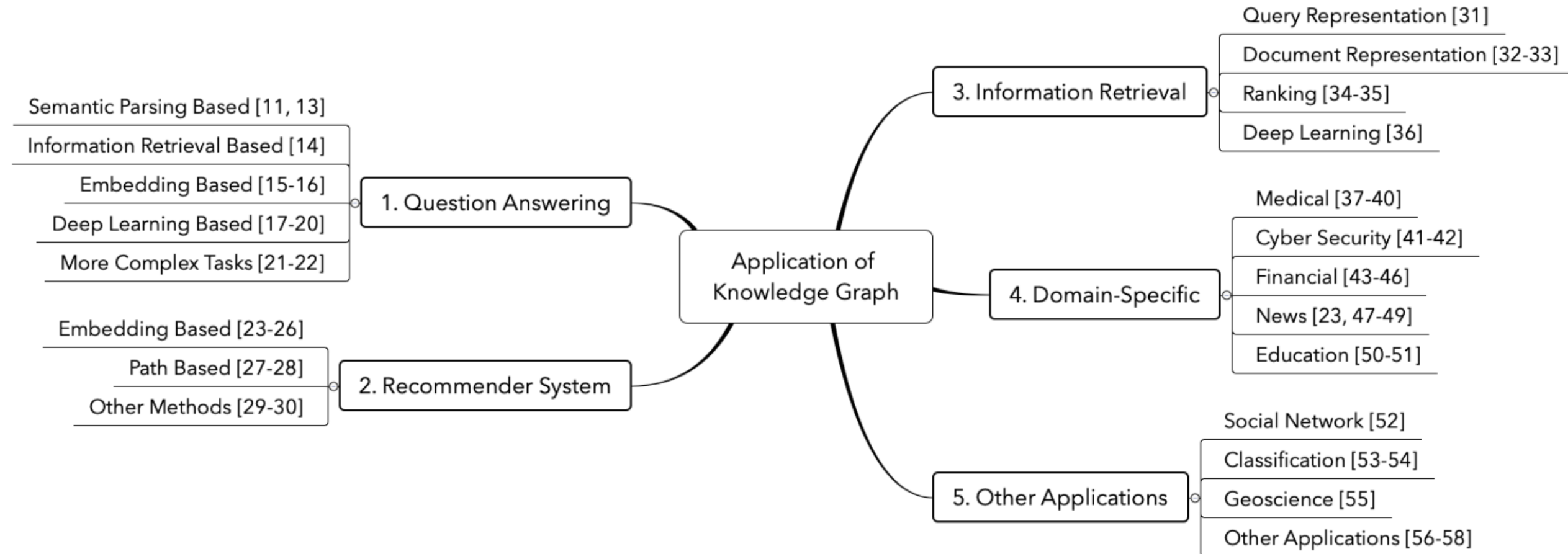


Figure 4: The illustration of the hyperedge convolution layer.

# Области применения графов знаний

Источник: <https://iopscience.iop.org/article/10.1088/1742-6596/1487/1/012016>

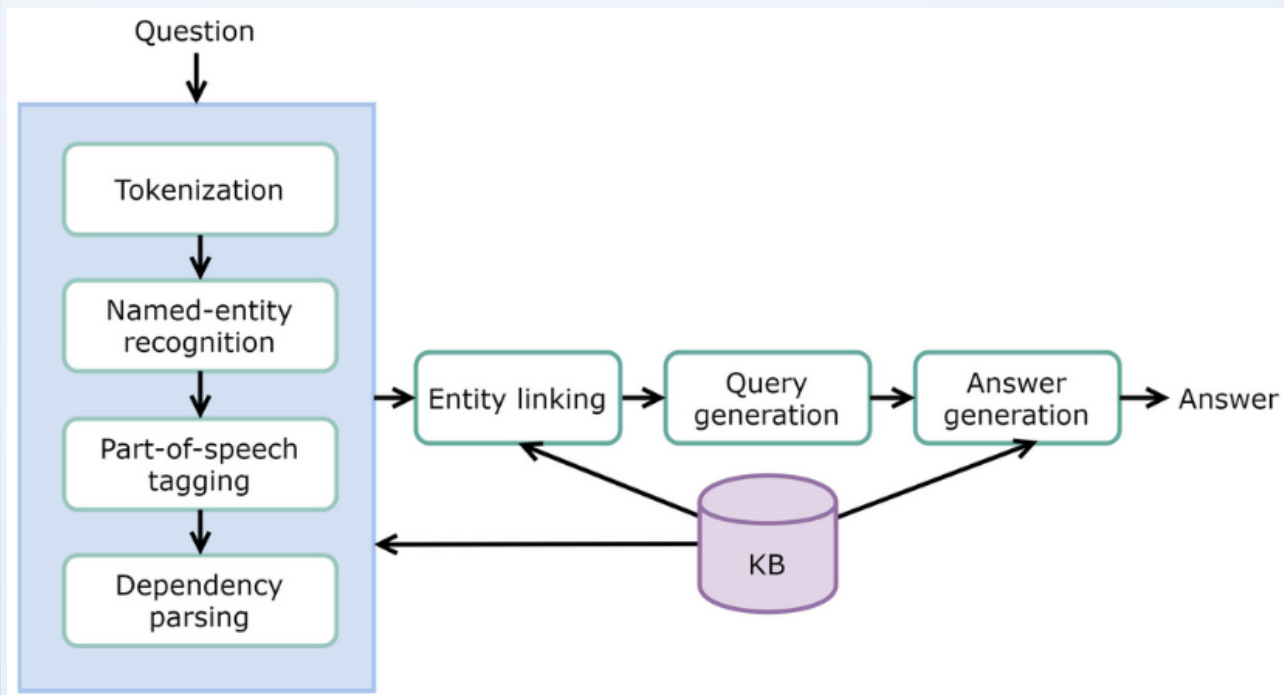


# Вопросно-ответные системы

- Источник:  
<https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/sfw2.12028>
- Для создания вопросно-ответных систем на основе графов знаний существует четыре основных подхода:
  1. Семантический разбор (semantic parsing)
  2. Сравнение подграфов (subgraph matching)
  3. Использование шаблонов (template-based question answering)
  4. Использование баз знаний (question answering over knowledge bases)

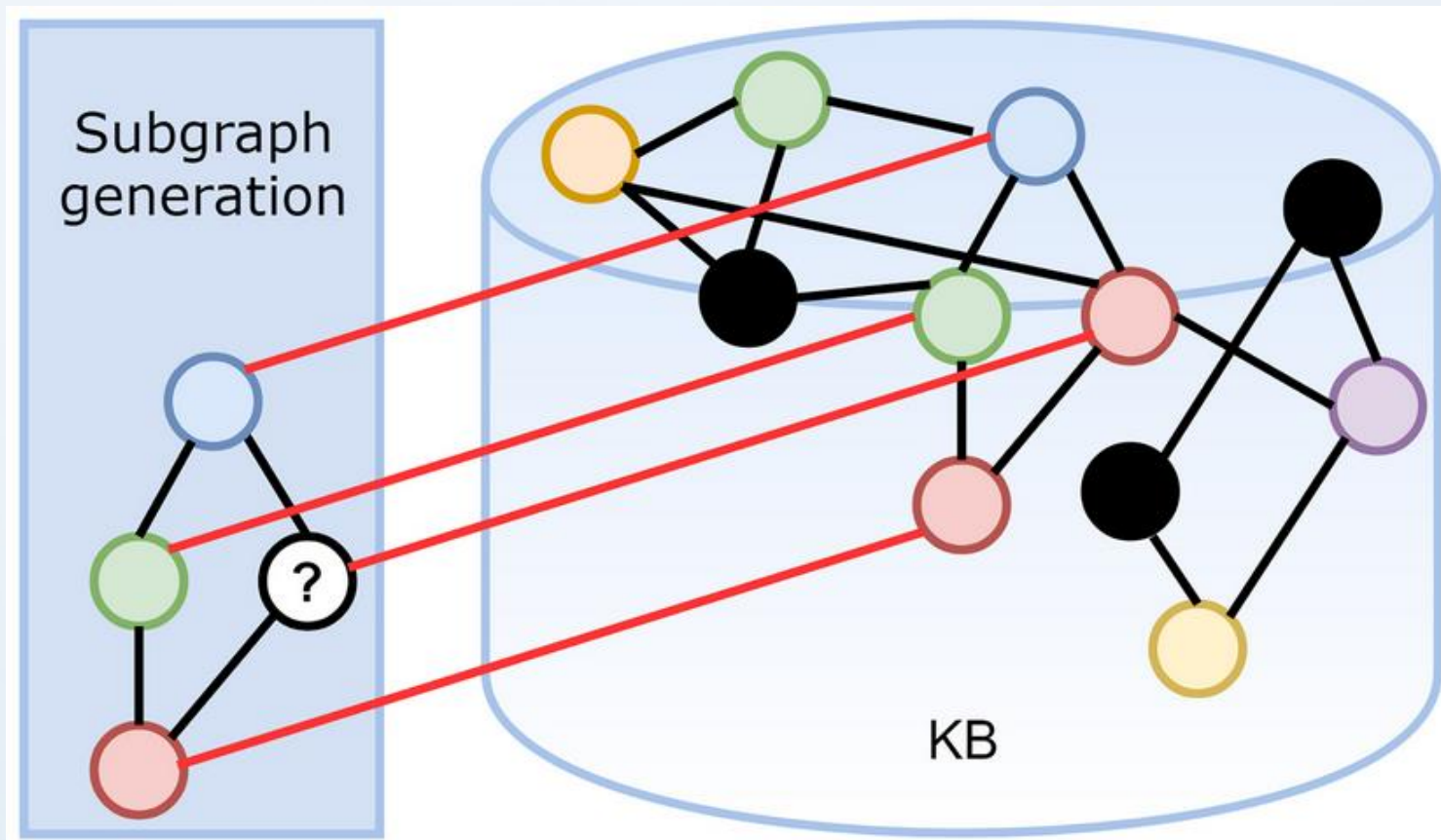


# Семантический разбор



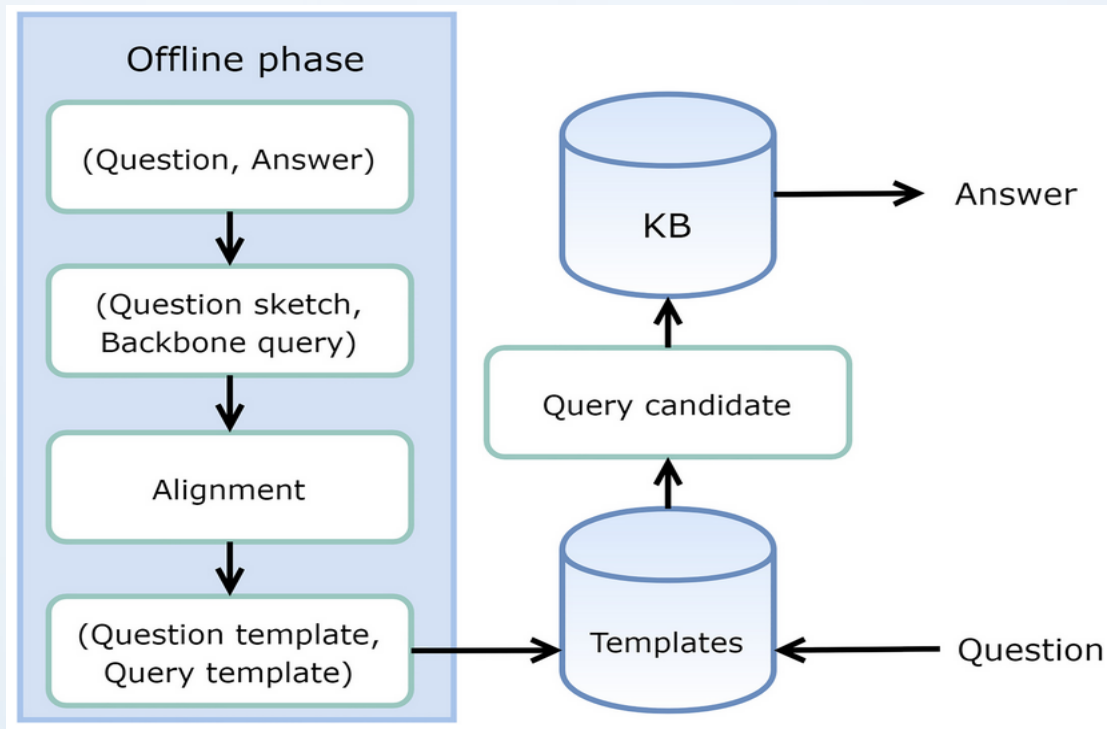
- Идея состоит в том, чтобы применить несколько преобразований данных из вопроса на ЕЯ к логической форме или формальному запросу.
- Для этого используются методы обработки естественного языка (NLP), такие как токенизация, распознавание именованных сущностей (NER), частеречная разметка (POS), анализ зависимостей и связывание сущностей (EL).
- Финальные шаги включают генерацию запроса и генерацию ответа.

# Сравнение подграфов



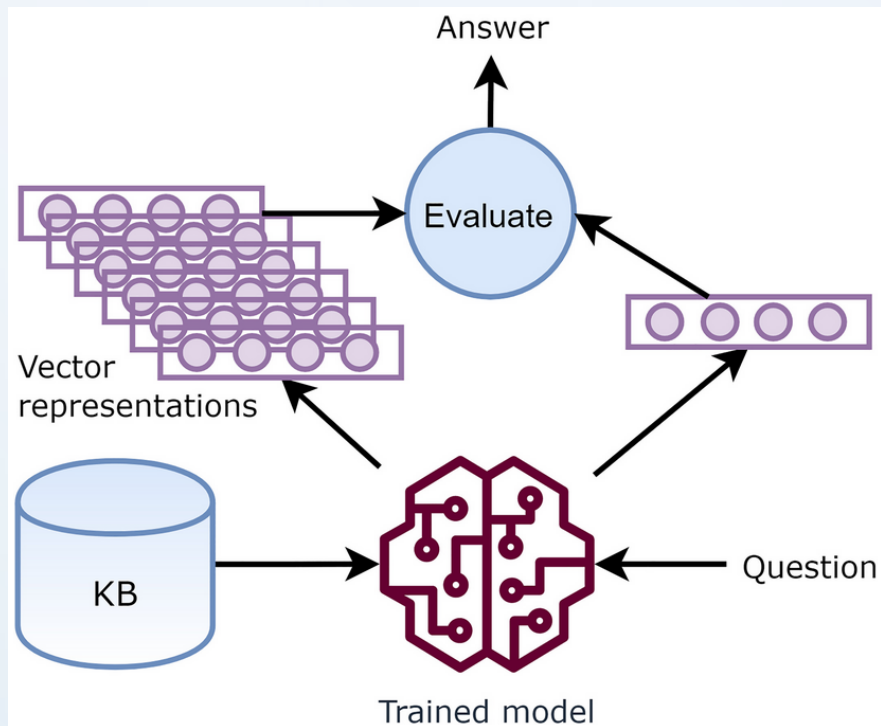
- В качестве альтернативы использованию семантического разбора может быть применено сравнение подграфов, так как выполнение формального запроса эквивалентно поиску подграфа.

# Использование шаблонов



- Шаблон представляет собой заготовку запроса с определенной степенью сложности, которая соответствует базе знаний, и содержит слоты, заполняемые информацией из сущностей и отношений. Качество системы значительно зависит от усилий, затраченных на создание шаблонов. Ранее создание шаблонов в основном осуществлялось с помощью ручных аннотаций; в последнее время появились автоматические подходы.
- Использование данного подхода предполагает наличие оффлайн и онлайн этапов.
- На оффлайн этапе создаются шаблоны. На основе пар вопрос-ответ формируются шаблоны вопроса и запроса для заполнения слотов.
- Онлайн этап включает сопоставление вопроса с шаблоном вопроса. Производится поиск шаблона запроса и заполнение слотов.

# Использование баз знаний



- End-to-end решения выполняют преобразование sequence-to-sequence или применяют методы для извлечения данных непосредственно из базы знаний.
- Выбор окончательного ответа основан на представлениях вопросов на ЕЯ, полученных с применением методов машинного обучения.
- После извлечения ответов-кандидатов из базы знаний их применимость оценивается по заранее заданной шкале с помощью специализированной функции оценки.



# Комбинирование с языковыми моделями

- Источник: <https://arxiv.org/abs/2012.04808>

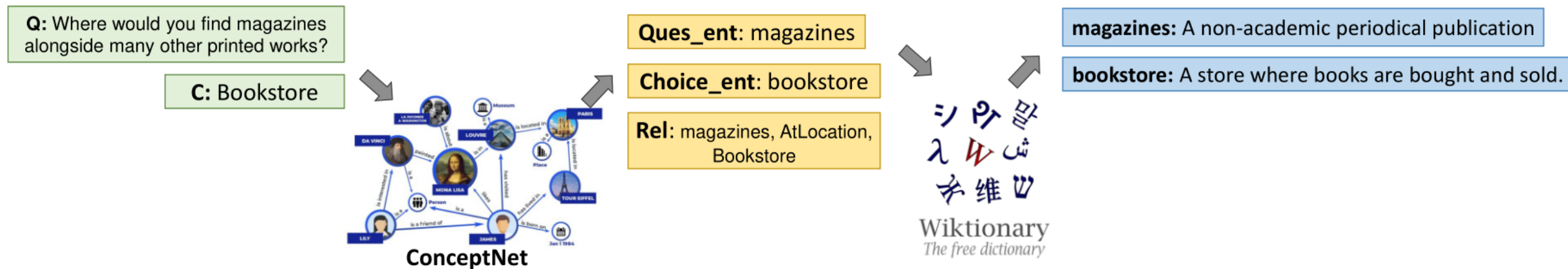
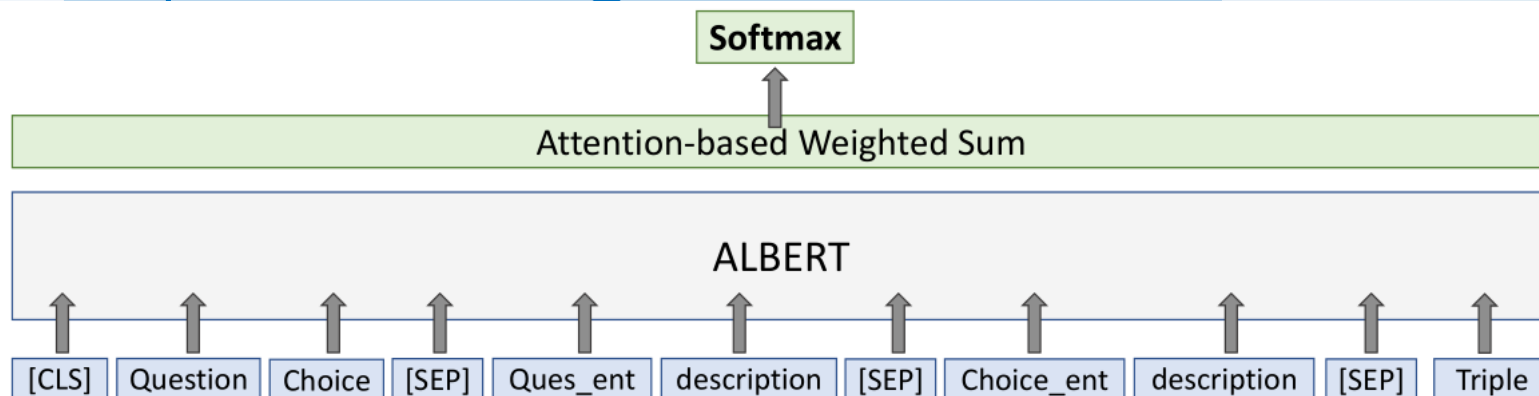


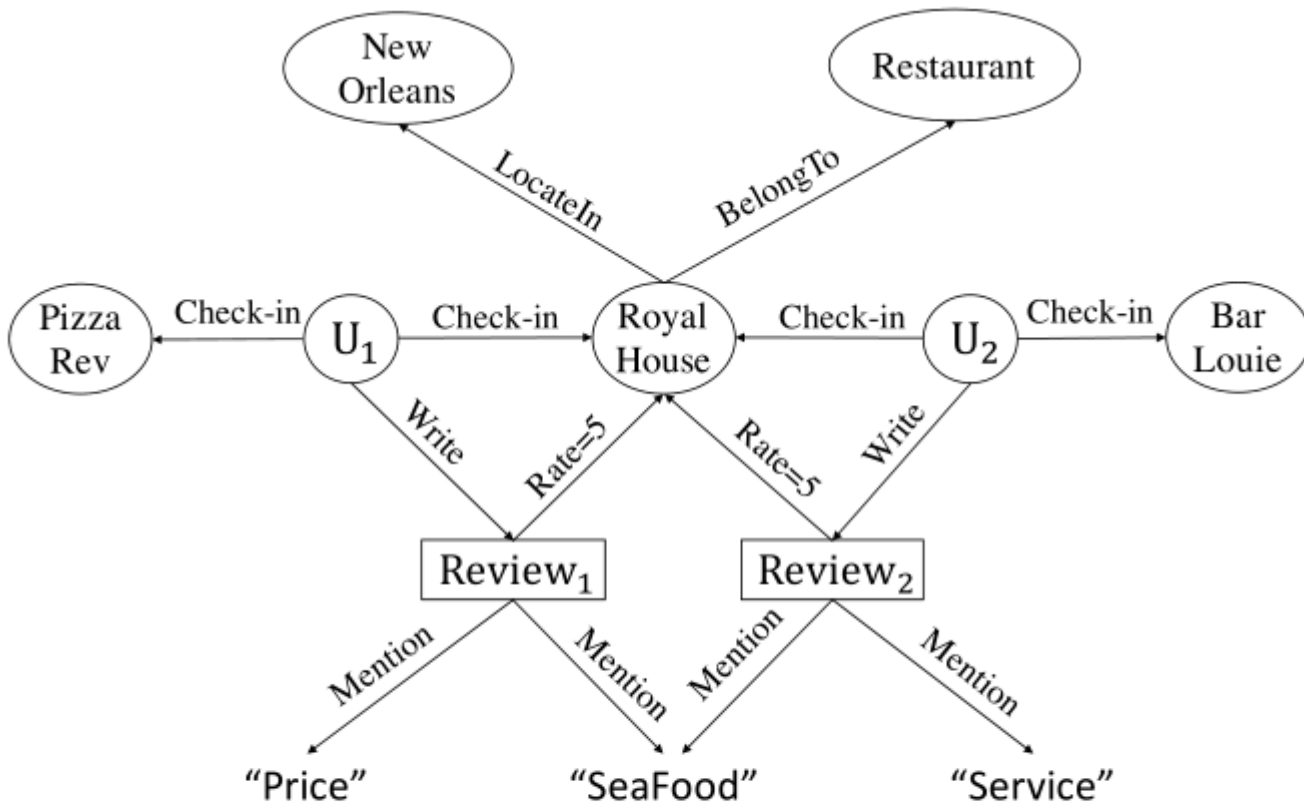
Figure 1: In our model, the input to ALBERT includes the question, choice, entity names, description text and triple. An attention-based weighted sum and a softmax layer process the output from ALBERT to produce the prediction.

# Рекомендательные системы

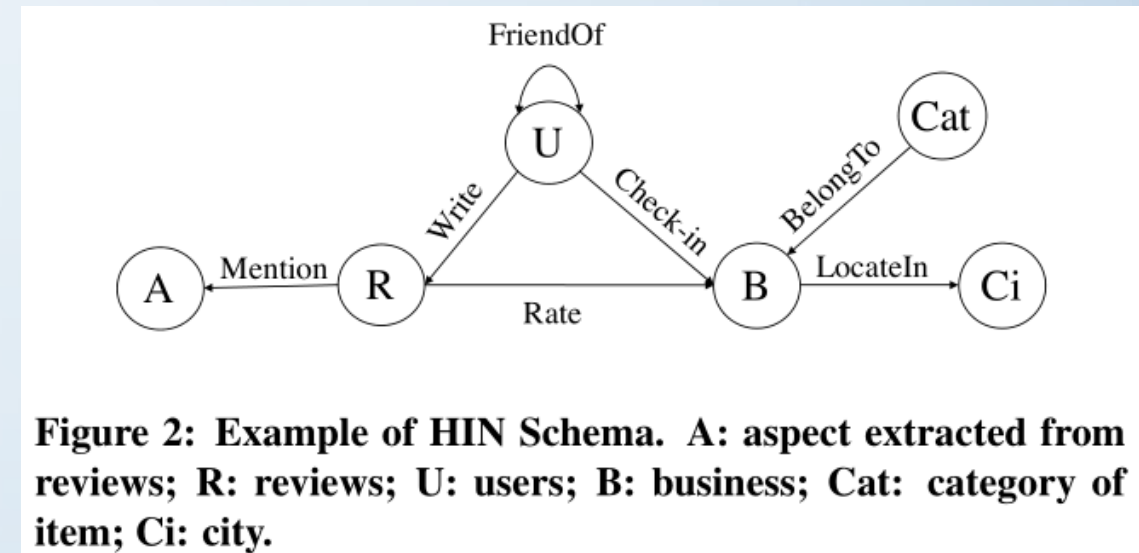
- Источник: <https://www.mdpi.com/2078-2489/12/6/232/pdf>
- Граф знаний может выступать в роли онтологии, на основе которой выдаются фиксированные рекомендации пользователю на основе текущих выбранных элементов (если покупается книга, то рекомендуется подставка для книг).
- Граф знаний в форме векторного представления может быть использован как в подходе на основе содержания, так и при коллаборативной фильтрации.
- Подход на основе гетерогенных информационных сетей предполагает, что база знаний представляется в виде мета-графов.
- С использованием подхода на основе LSTM-сети можно осуществлять эффективный поиск в мета-графах, что может служить основой для выбора рекомендаций.

# Гетерогенная информационная сеть

- Heterogeneous Information Networks (HIN).

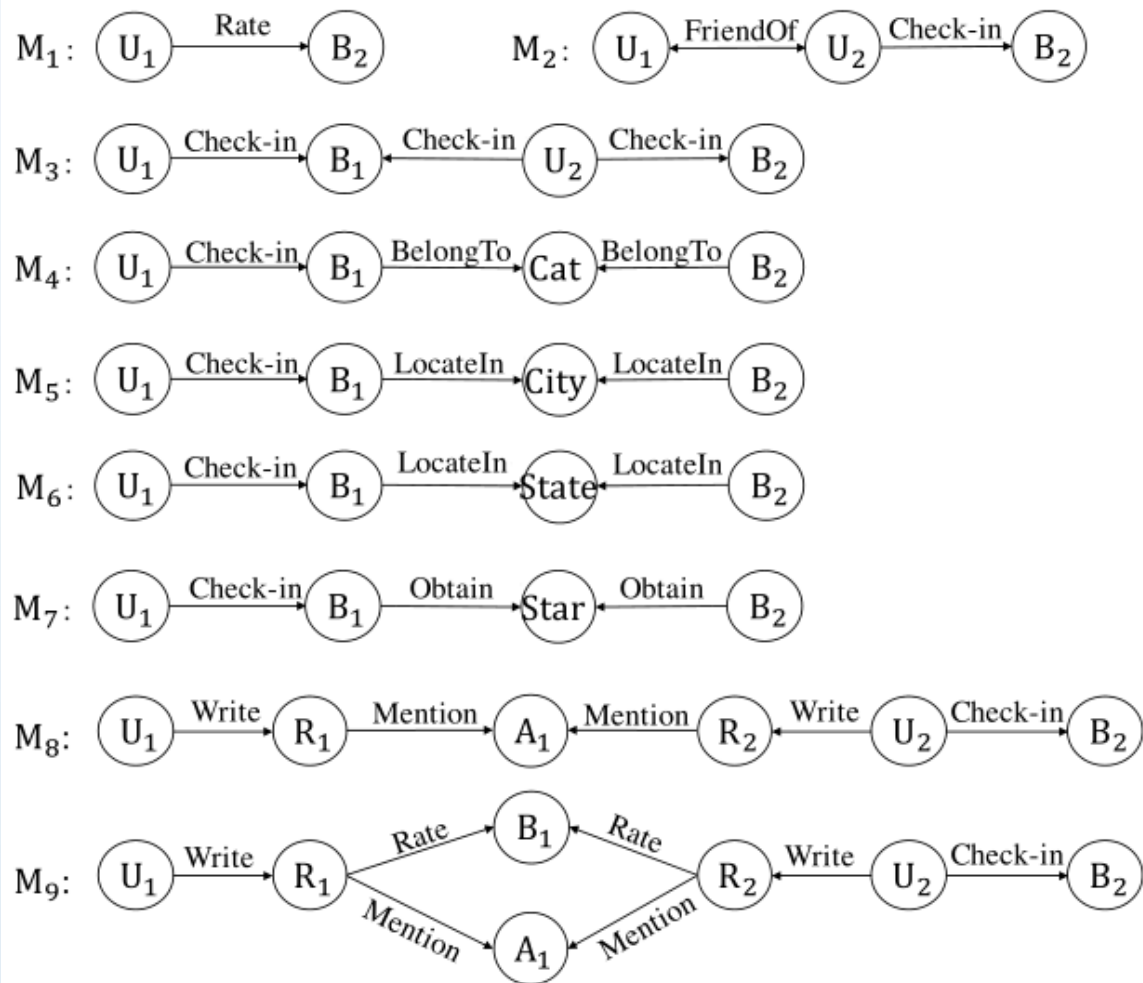


**Figure 1: Example of HIN, which is built based on the web page for Royal House on Yelp.**

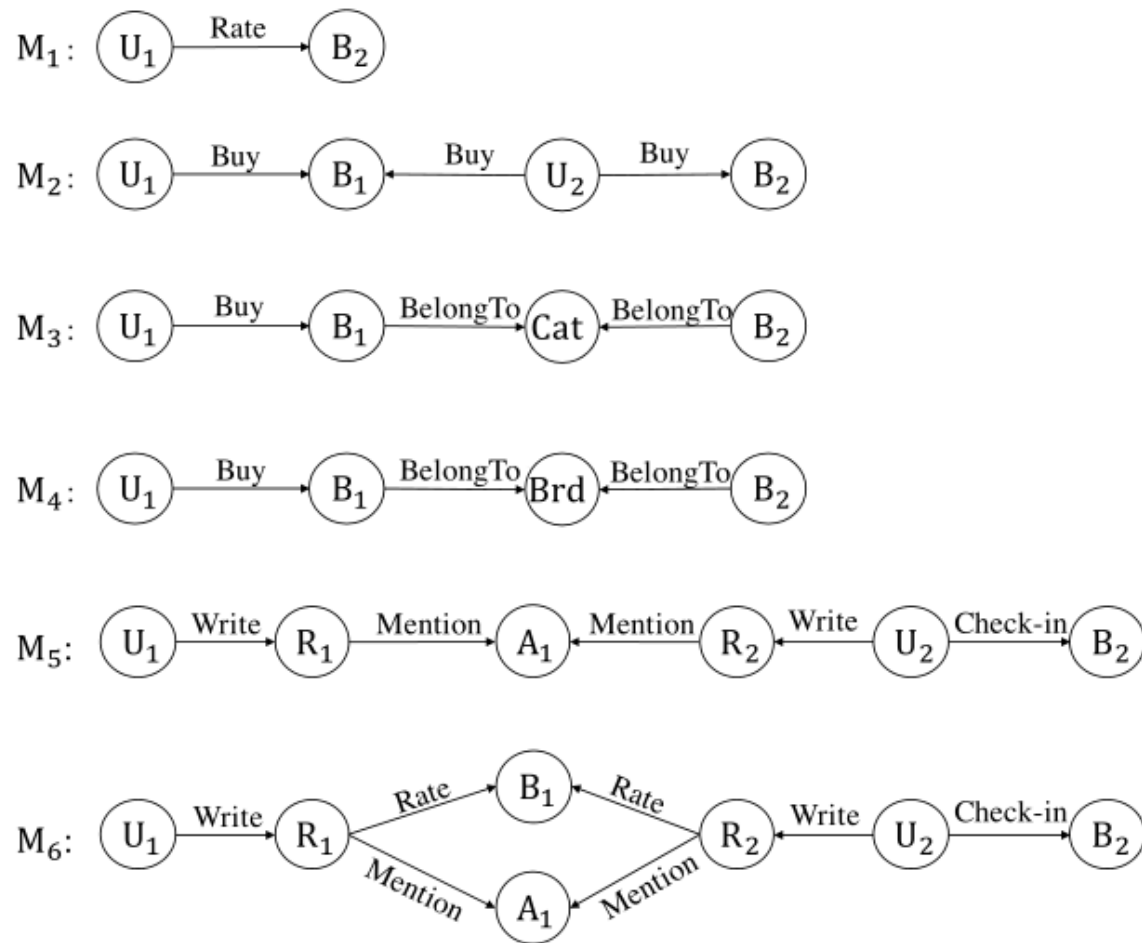


**Figure 2: Example of HIN Schema. A: aspect extracted from reviews; R: reviews; U: users; B: business; Cat: category of item; Ci: city.**

# Пример мета-графов



(a) Yelp-200K (Star: the average stars a business obtained).

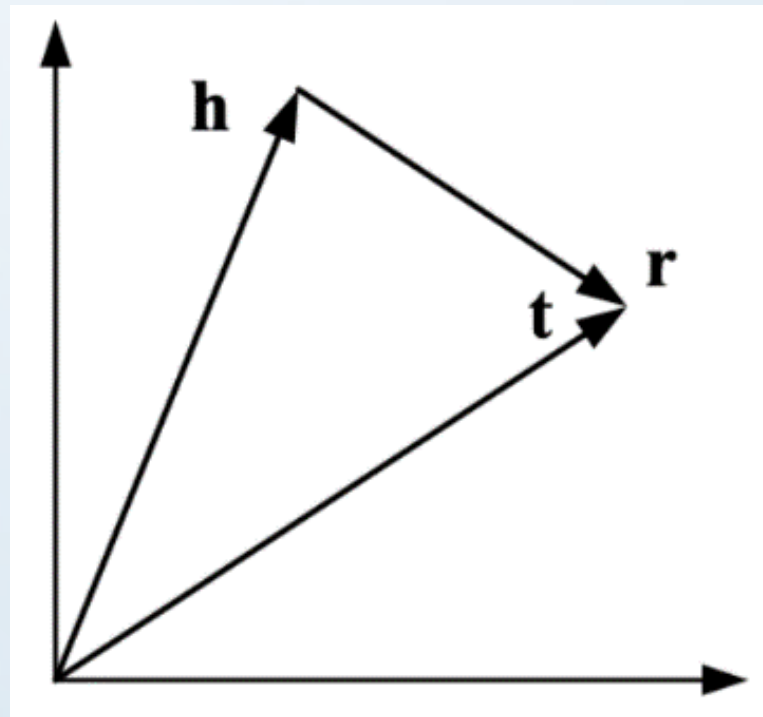


(b) Amazon-200K (Brd: brand of the item).

Figure 3: Meta-graphs used for Amazon and Yelp datasets.

# Knowledge Graph Completion – 1

- Источник: <https://www.semanticscholar.org/paper/Knowledge-Graph-Completion%3A-A-Review-Chen-Wang/b770eaea1124b5f5805713428daf023c792b15cd>
- Подход 1. Основывается на векторных представлениях.





# Knowledge Graph Completion – 2

- ИСТОЧНИК:  
[https://openreview.net/pdf?id=\\_wHAe6eoT8](https://openreview.net/pdf?id=_wHAe6eoT8)
- Подход 2. Основывается на предобученных языковых моделях.

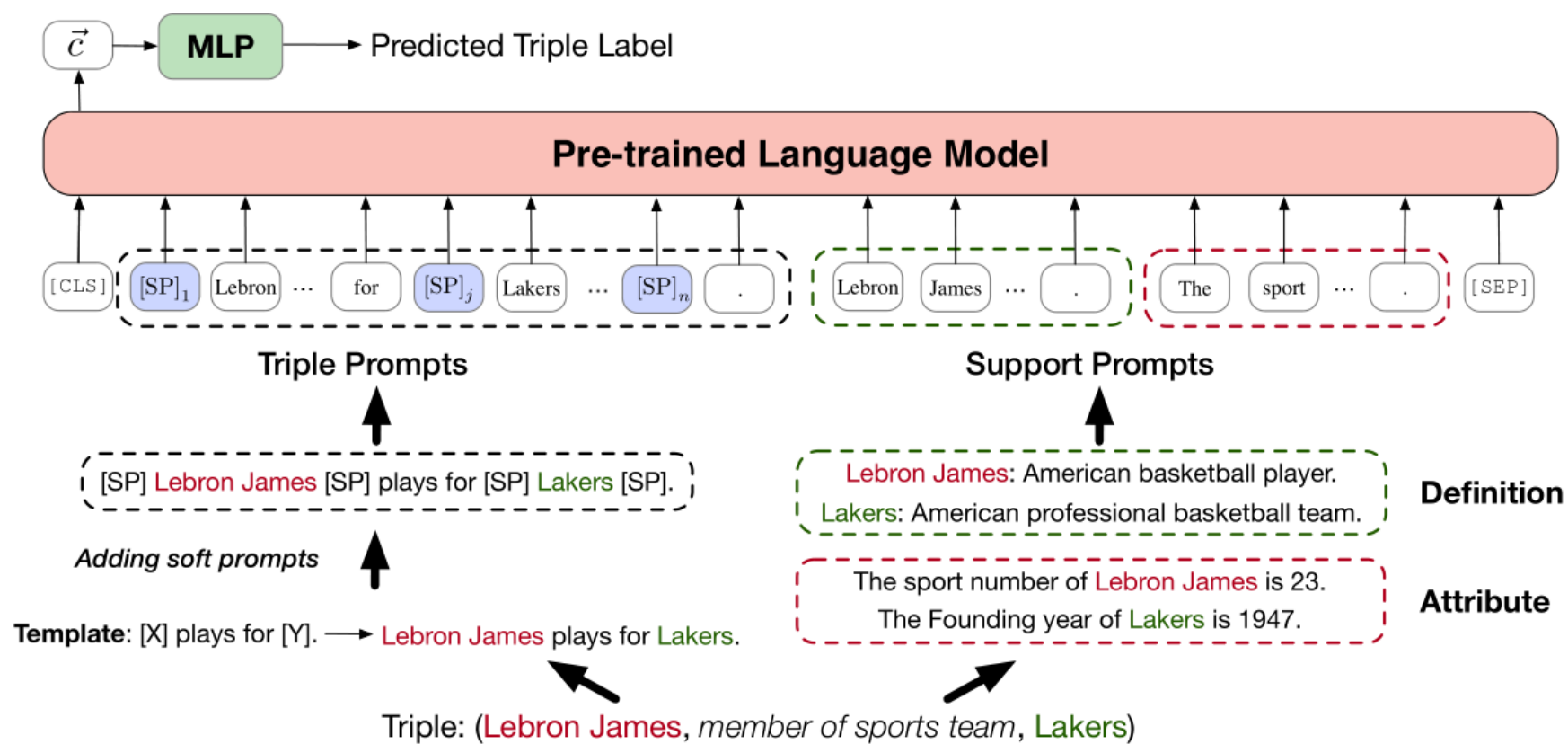


Figure 2: Illustration of our PKGC model for triple classification. Triples are transformed into triple prompts (left part) and support prompts (right part) to do the classification using PLMs.



Figure 3: Illustration of inserting soft prompts into triple prompts. The "Template" in the sentence represents the words in the hard template. We can insert several soft prompts in six positions (underlined) at most, and the sum of the numbers of these soft prompts is  $n$ .

Type	Template
Definition	"[ <i>Entity</i> ]: [ <i>Definition Text</i> ]."
Attribute	"The [ <i>Attribute</i> ] of [ <i>Entity</i> ] is [ <i>Value</i> ]."

Table 1: Templates for support information, where [*Entity*], [*Attribute*], [*Value*] denote the label of entity, attribute and value respectively. [*Definition Text*] is the text corresponding to the entity definition.

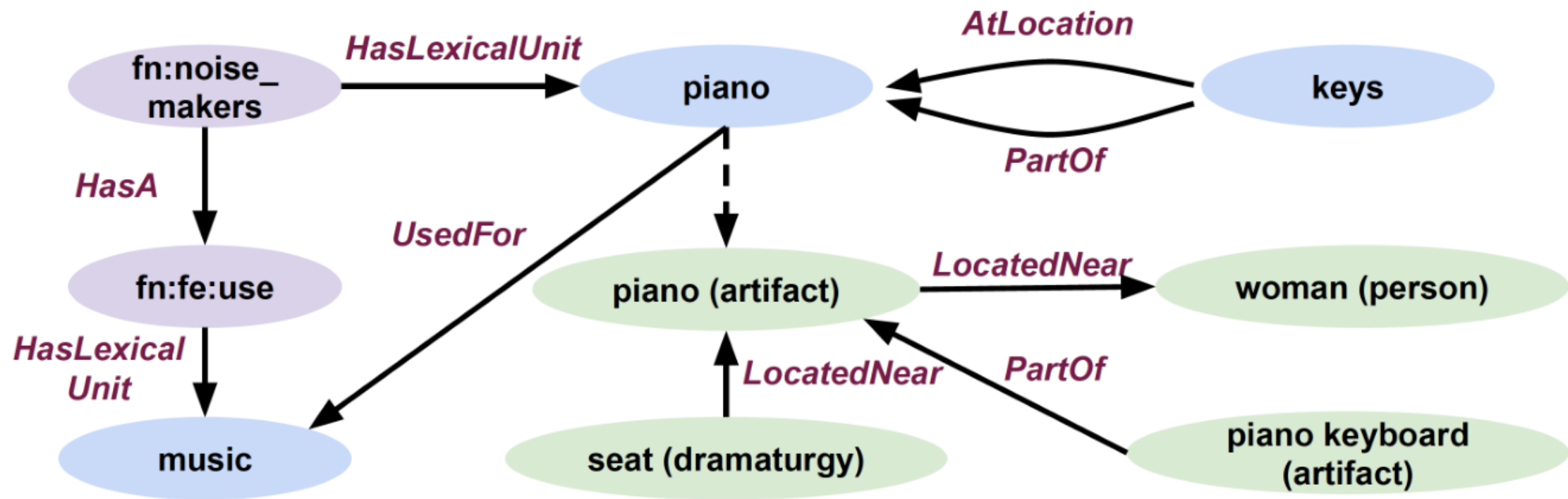
# Commonsense reasoning – 1



Figure 8: A snapshot of commonsense knowledge relationships in ATOMIC<sub>20</sub><sup>20</sup>. Gray nodes represent events. Red, purple and green nodes represent social-interaction commonsense, event-centered commonsense, and physical-entity commonsense, respectively. Rest of the colors represent intersection of the categories.

# Commonsense reasoning – 2

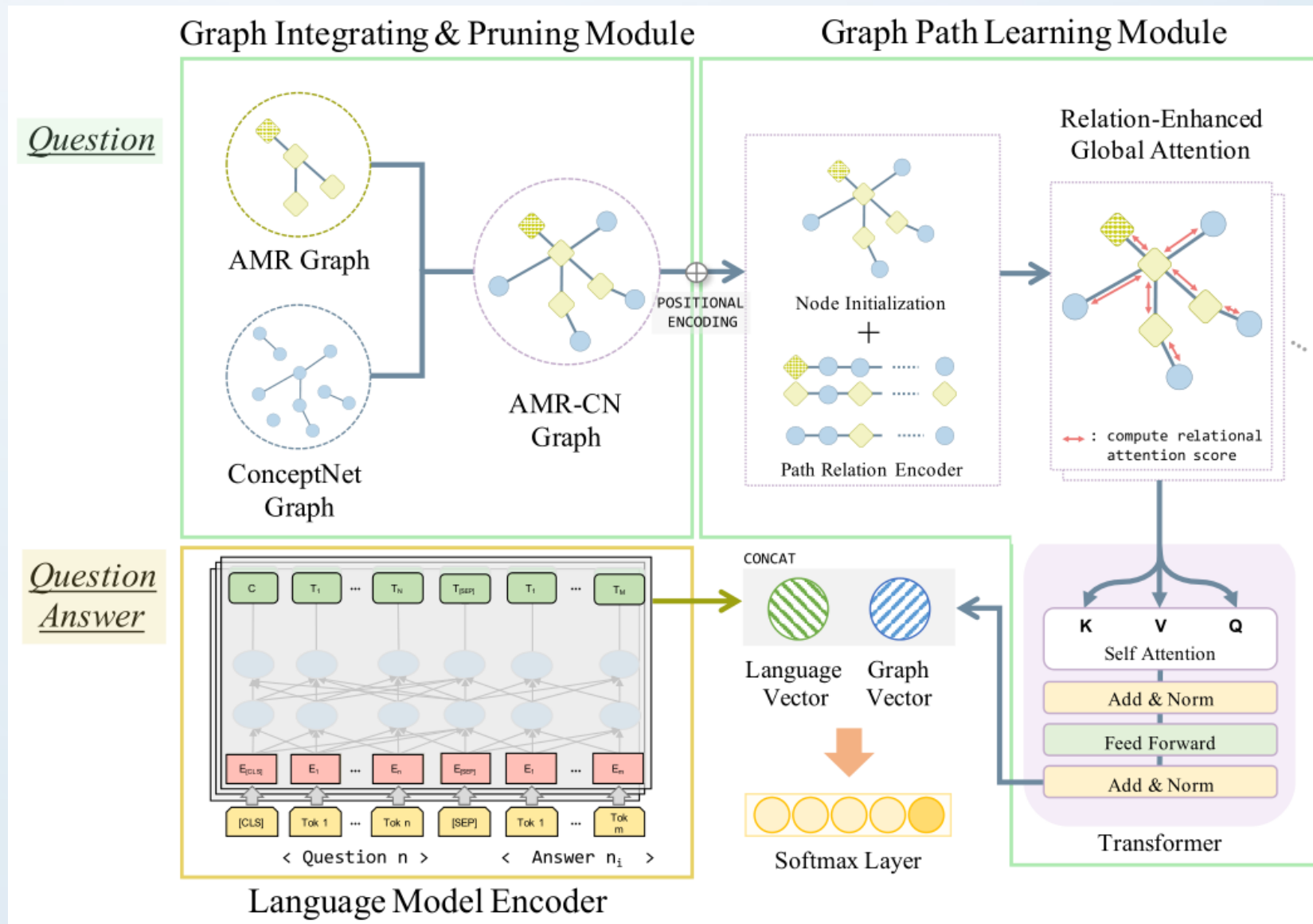
- Источник: <https://arxiv.org/abs/2012.11490>



**Fig. 1.** Snippet of CSKG for the example task of section 1. CSKG combines: 1) lexical nodes (`piano`, `keys`, `music`; in blue), 2) synsets like `piano (artifact)`, `seat (dramaturgy)` (in green), and 3) frames (`fn:noise_makers`) and frame elements (`fn:fe:use`) (in purple). The link between `piano` and `piano (artifact)` is missing, but trivial to infer.

# Commonsense reasoning – 3

- Источник: <https://arxiv.org/abs/2011.00766>





# Bayesian Reasoning Framework

- Источник: <https://ojs.aaai.org/index.php/AAAI/article/view/16565>

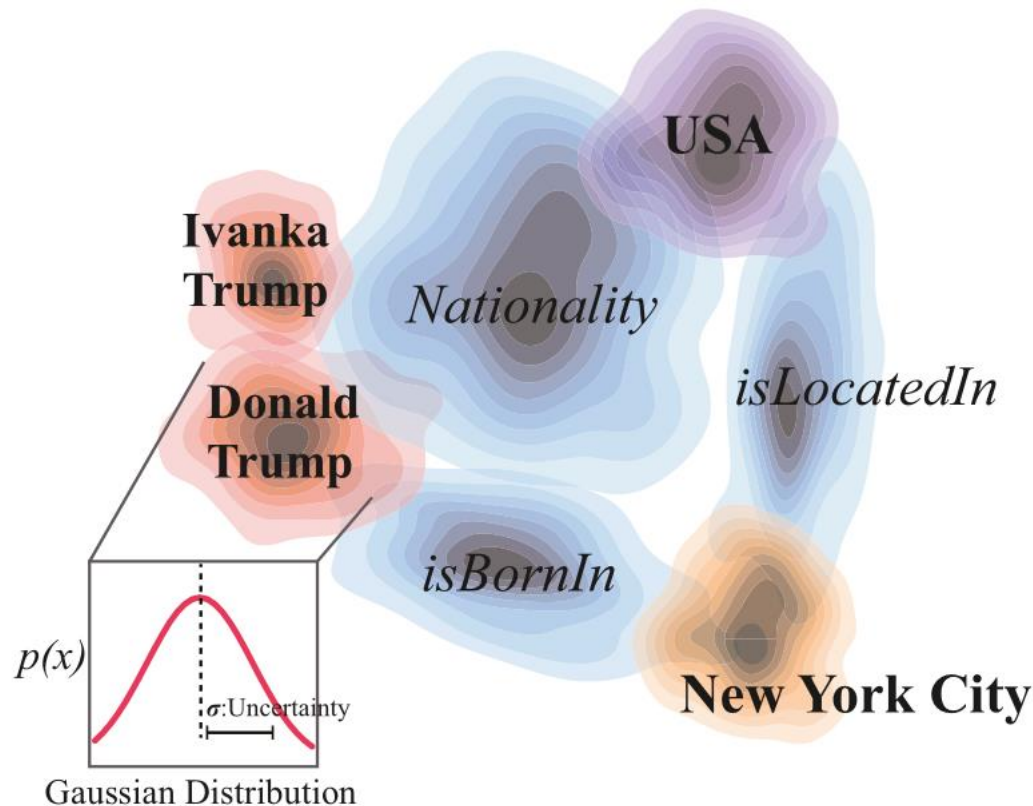


Figure 1: An illustrated toy example which generated from randomized two-dimensional Gaussian distribution for representing entities and relations in a partial KG. The mean of an entity indicates its position, and the variance indicates uncertainty.