УДК 004.021; 004.42

КАК ЭМБЕДДИНГИ ИМЕН СУЩНОСТЕЙ ВЛИЯЮТ НА КАЧЕСТВО ВЫРАВНИВАНИЯ СУЩНОСТЕЙ

Д. И. Гусев[1] [0000-0001-9636-2783], З. В. Апанович[2] [0000-0002-5767-284X]

[1]*Новосибирский государственный университет,* ул. Пирогова, 1 Новосибирск, 630090
[2]*Институт систем информатики им. А.П. Ершова Сибирского отделения Российской академии наук, пр. Академика Лаврентьева, 6, Новосибирск, 630090*

[1]d.gusev1@g.nsu.ru, [2]apanovich_09@mail.ru

Аннотация

Алгоритмы установления соответствия между сущностями осуществляют поиск эквивалентных сущностей в разноязычных графах знаний. Данная проблема возникает, как правило, при интеграции разноязычных графов знаний. В настоящее время решение этой проблемы становится весьма актуальным для практического решения проблем импортозамещения, например, чтобы найти информацию о лекарствах, выпускаемых в разных странах под разными названиями, или же решить проблему поиска эквивалентных запчастей.

В настоящее время известно несколько библиотек с открытым кодом, которые объединяют известные алгоритмы выравнивания сущностей, а также тестовые наборы данных для различных языков. В данной работе описан русско-английский набор данных для экспериментов с нескольким популярными алгоритмами выравнивания сущностей. Особое внимание уделено методам генерации векторных представлений для имен сущностей. В частности, рассмотрены комбинации различных методов генерации векторных представлений (эмбеддингов) имен сущностей с известными алгоритмами выравнивания сущностей. Таблицы с результатами экспериментов дополнены визуализациями.

Ключевые слова*: разноязычные графы знаний, идентификация сущностей, cross-lingual entity alignment, knowledge graphs, relational embeddings, name embeddings.*

ВВЕДЕНИЕ

В последние годы графы знаний (ГЗ) используются все в большем количестве предметных областей, и все большее количество приложений использует этот тип представления для хранения данных без потери их семантики. Чем мощнее граф знаний, тем выше качество приложений, на них базирующихся. Поэтому весьма актуальной является задача интеграции различных графов знаний, а в основе такой интеграции находится решение задачи слияния информации из разных графов знаний об одном и том же объекте реального мира. Данная задача известна под такими названиями, как *сопоставление сущностей*, *выравнивание сущностей*, *идентификация сущностей* и др. В последние несколько лет возрос интерес к интеграции разноязычных графов знаний, поэтому весьма актуальной является задача связывания информации об одних и тех же объектах реального мира, описанных в разноязычных графах знаний. Разные языковые версии графов знаний обладают, с одной стороны, свойством взаимодополнительности, а с другой стороны, каждая языковая версия содержит более точную и полную информацию об объектах, характерных для конкретного языка. Например, русскоязычная версия DBpedia содержит более полную и корректную информацию об объектах, расположенных на территории России.

Графы знаний хранят факты в виде реляционных и литеральных триплет. Реляционные триплеты изображают отношение между двумя объектами реального мира и имеют формат *tr_r* = (*субъектная сущность, отношение, объектная сущность*). Литеральные триплеты хранят информацию об атрибутах объектов реального мира и имеют формат *tr_l* = (*субъектная сущность, атрибут, литеральное значение*). При визуализации литеральные значения принято изображать прямоугольниками, а сущности или объекты реального мира – овалами.

Например, на рис. 1 показаны фрагменты из англоязычной и русскоязычной версий DBpedia. Примером реляционной триплеты является триплета (*Сталкер_(фильм), режиссер, Андрей_Тарковский*), а примером литеральной триплеты является триплета (*Сталкер_(фильм), длительность, 163 минуты*). Красными линиями изображены отношения *owl:sameAs*, имеющиеся между сущностями в русскоязычном и англоязычном графах знаний. Можно видеть, что англоязычной

сущности *Stalker (1979 film)* в англоязычном графе знаний соответствует русскоязычная сущность *Сталкер_(фильм),* англоязычному отношению *dbo:director* – русскоязычное отношение *режиссер*, а англоязычной сущности *Andrei Tarkovsky* – русскоязычная сущность *Андрей Тарковский.* Понятно, что англоязычный и русскоязычный списки актеров, сыгравших роли в этом фильме, должны бы совпадать. Однако в реальных графах знаний наблюдаются некоторые различия в описаниях смежных сущностей. Так, длительность фильма в англоязычной версии указана в секундах, а в русскоязычной версии – в минутах, в русскоязычной версии указаны братья Стругацкие в качестве авторов сценария, а в англоязычной версии этой информации нет. Понятно, что наиболее полное описание сущности можно получить объединением всех триплет, описывающих одну и ту же сущность. Но для решения этой задачи должно быть правильно установлено соответствие между сущностями. Эта задача и носит название *выравнивание сущностей*.

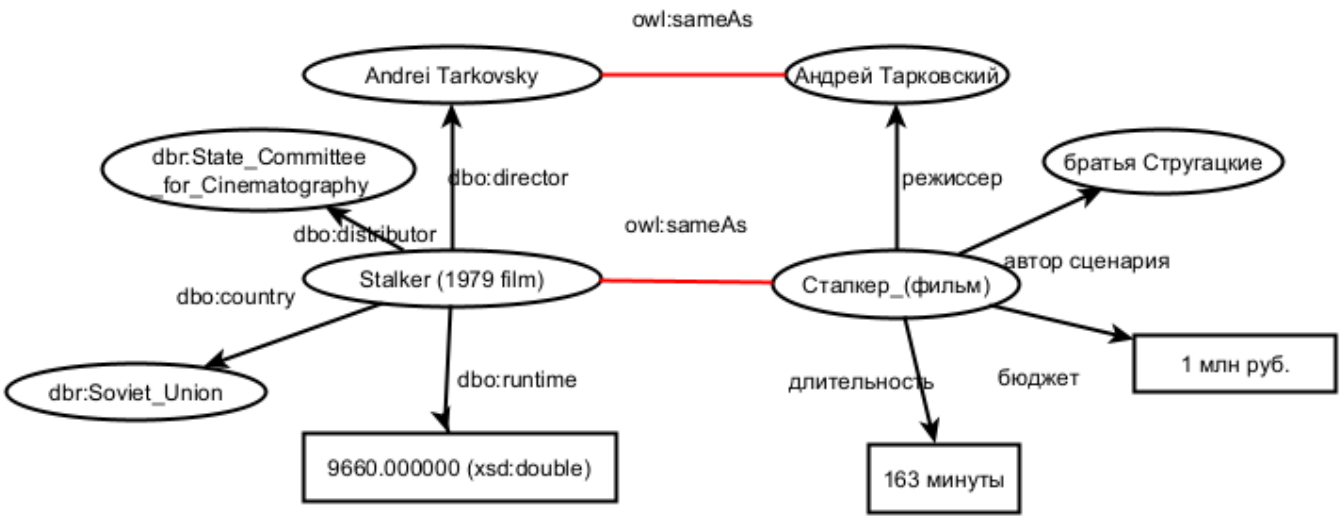

Рис. 1. Соответствие между англоязычными и русскоязычными сущностями.

В последние несколько лет получили распространение методы установления соответствия между сущностями различных графов знаний, использующие так называемые «эмбеддинги» (embeddings), векторные представления заданной размерности для сущностей и отношений графов знаний. Достоинствами подхода на основе эмбеддингов являются высокая масштабируемость и небольшие усилия при подготовке обучающих выборок.

Следует сказать, что создание новых методов основано на интуиции разработчиков, эвристиках и экспериментах проб и ошибок. Поэтому весьма важным является создание общей основы для понимания разнообразных методов. В настоящее время такую общую основу составляют результаты тестирования различных алгоритмов на едином наборе данных. В работе [1] представлена библиотека OpenEA, содержащая несколько десятков алгоритмов ЕА на основе различных стратегий построения векторных представлений, а также результаты экспериментов с этими векторными представлениями на тестовой выборке, содержащей англо-немецкие, англо-французские и англо-китайские данные.

Понятно, что русскоязычному пользователю интересны, прежде всего, эксперименты, использующие русскоязычные данные. Во-первых, такие данные проще интерпретировать, во-вторых, известно, что различные языковые версии графов знаний обладают свойством «смещенности», то есть одни и те же алгоритмы могут давать разные результаты на разных версиях графов знаний из-за различной структуры этих графов.

В работе [2] описан русско-английский набор данных для экспериментов с алгоритмами кросс-языкового выравнивания сущностей. К удивлению авторов, алгоритмы, выдававшие наилучшие результаты на стандартных разноязычных наборах данных, выдавали весьма посредственные результаты на русско-английском наборе данных. Этот вопрос потребовал дополнительного изучения, и в данной работе представлены эксперименты с алгоритмами выравнивания сущностей разного типа на англо-русской обучающей выборке. Рассмотрены различные способы построения векторных представлений имен сущностей, а также возможные комбинации этих методов с методами построения векторных представлений сущностей на основе реляционных триплет.

1. ГРУППЫ АЛГОРИТМОВ СОПОСТАВЛЕНИЯ СУЩНОСТЕЙ НА ОСНОВЕ ВЕКТОРНЫХ ПРЕДСТАВЛЕНИЙ (EMBEDDINGS)

Большинство методов выравнивания сущностей на основе векторных представлений сводится к двум шагам:

- Генерация векторных представлений для сущностей и отношений;

- Отображение этих векторных представлений в единое векторное пространство при помощи предварительно выровненных сущностей (seed alignments) или в различные векторные пространства.

В первом случае вопрос, являются ли две сущности из разных графов эквивалентными (соответствующими одному и тому же объекту реального мира), решается при помощи сравнения их векторов, например, вычислением евклидова расстояния или косинусной близости. При отображении сущностей двух графов знаний в разные векторные пространства нужно также находить матрицу соответствия между векторами этих двух пространств.

Современные решения выравнивания сущностей опираются, в основном, на структурную информацию в графах знаний, то есть реляционные триплеты. Основу этих методов составляет предположение о том, что эквивалентные сущности должны иметь сходные графовые окрестности. При появлении методов выравнивания сущностей преобладал так называемый триплетно-трансляционный подход, который рассматривал вектор, представляющий отношение между двумя сущностями, как вектор сдвига вектора одной сущности относительно вектора второй сущности. Одним из лучших представителей триплетно-трансляционного подхода является MultiKE (Multi-view Knowledge Graph Embedding) [3]. MultiKE строит три типа векторных представлений для каждой сущности, используя так называемые «виды» (views):

- вид, зависящий от названия сущности,
- «реляционный вид», конструируемый по реляционным триплетам каждой субъектной сущности,
- «атрибутный вид», создаваемый по литеральным триплетам субъектной сущности.

Каждый из «видов» строится по собственному алгоритму. Например, для каждого слова из названия сущности находится вектор, полученный с помощью word2vec [4], а если такого не существует, то вектор слова получается с помощью суммирования векторов символов, полученных с помощью алгоритма character embedding. Векторы слов суммируются, и получается вектор названия, который непосредственно участвует в обучении модели.

Для построения реляционного вида используется модель TransE [5], где отношение между двумя сущностями интерпретируется как вектор сдвига между сущностью-субъектом и сущностью-объектом реляционной триплеты. Наконец, атрибутные виды строятся на основе литеральных триплет, в которых данная сущность является субъектом. Для построения атрибутных представлений используются сверточные нейронные сети. Окончательное векторное представление сущности может быть получено при помощи разных способов комбинирования упомянутых трех видов.

В последние годы чрезвычайно популярными стали подходы построения векторных представлений сущностей на основе графовых сверточных сетей. Эти методы выдают очень неплохие результаты, но их основными недостатками являются чрезвычайная сложность, значительное время вычислений и плохая интерпретируемость. Представителем этого подхода является RDGCN (Relation-aware Dual-Graph Convolutional Network) [6]. Подход RDGCN использует для построения векторных представлений не только структуру исходных графов знаний (primal entity graph), но и вспомогательные графы, двойственные по отношению к исходным графам (dual relation graph), вершинами которых являются ребра исходных графов. Для осуществления взаимодействия между исходными графами знаний и двойственными реляционными графами используется механизм графовых сетей внимания (Graph Attention Networks, GAT) [7]. Результирующие векторные представления исходных графов затем подаются в графовые сверточные сети (Graph Convolutional networks, GCN) [8] для извлечения информации о структуре окружений вершин.

Совсем недавно появился чрезвычайно простой подход к выравниванию сущностей под названием SEU (Simple but Effective Unsupervised EA method) [9], не использующий нейронные сети. Основная идея SEU состоит в сведении задачи выравнивания сущностей к давно известной задаче назначения, для которой существует хорошо известный венгерский алгоритм решения. Основным предположением этого подхода является то, что матрицы смежностей двух графов знаний являются изоморфными. В этом случае матрица смежности исходного графа может быть преобразована в матрицу смежности второго графа посредством переупорядочения строк или столбцов.

Тем не менее, большинство недавних исследований указывает на то, что современные методы выравнивания сущностей не способны выдавать удовлетворительные результаты только на основании реляционных триплет, если набор данных имеет распределение степеней сущностей, близкое к реальным КГ. В частности, известно, что примерно половина сущностей в реальных КГ связана с менее чем тремя другими сущностями [9].

Это наблюдение делает важным использование дополнительной информации, такой как имена сущностей и комбинирование информации об именах сущностей со структурной информацией. Названия сущностей необходимо привести к общему языку, а затем сравнить. Возможны два базовых подхода для сравнения имен сущностей: подход на основе строкового сходства и подход на основе семантического сходства. Методы семантического сходства можно разбить на две группы: генерация векторных представлений на основе отдельных слов (модели word2vec, GloVe [10]). В силу ограниченности используемых словарей, часто возникает ситуация, что нужное слово отсутствует в используемом словаре, и в этом случае векторное представление слова строится на основе литер, входящих в его состав (модели fastText [11], name-BERT [12]).

2. РУССКО-АНГЛИЙСКИЙ НАБОР ДАННЫХ И МЕТРИКИ ДЛЯ ОЦЕНКИ КАЧЕСТВА АЛГОРИТМОВ ВЫРАВНИВАНИЯ СУЩНОСТЕЙ

Современные графы знаний имеют значительные размеры, поэтому вместо полномасштабных экспериментов по установлению соответствия между сущностями из разных графов знаний осуществляются эксперименты на выборках ограниченного размера. В настоящее время принято экспериментировать с выборками, содержащими 15 000 и 100 000 соответствий между сущностями из двух графов знаний. Наибольшее распространение получил набор данных DBP15K [1], который содержит по 15 000 пар сущностей, связанных отношениями *owl:sameAs* из разных языковых версий DBpedia, для таких пар языков, как англо-китайский, англо-французский и англо-немецкий. В [1] также описан итеративный алгоритм построения разноязычной выборки на основе степеней сущностей IDS (Iterative Degree-based Sampling), в которой распределение степеней сущностей близко к распределениям степеней в реальных графах знаний. Основная идея алгоритма IDS выглядит следующим образом [1].

В первую очередь удаляются сущности, которые не имеют связей *owl:sameAs* между двумя графами знаний. Пусть $P(x)$ — это доля сущностей, имеющих степень x в текущем графе знаний, а $Q(x)$ – доля сущностей, имеющих степень x в исходном графе знаний. Для оценки различия между распределениями степеней сущностей в двух наборах данных используется дивергенция Иенсена–Шэннона [13]. Доля сущностей, имеющих степень x в текущем наборе данных $P(x)$, не может быть равной доле сущностей, имеющих степень x в исходном наборе данных $Q(x)$. Поэтому вычисляется количество сущностей, которые надо удалить на одном шаге алгоритма, по формуле $dsize(x, \mu)=\mu(1+P(x)–Q(x))$, где μ – это базовый размер шага. Чтобы сбалансировать эффективность и безопасность удаления, устанавливается μ=100 при генерации тестовой выборки из 15 тысяч пар разноязычных триплет и μ=500 при генерации тестовой выборки из 100 тысяч пар разноязычных триплет. Затем при помощи алгоритма PageRank вычисляются сущности, которые реально будут удалены. Сущности, удаленные из одного графа знаний, удаляются и в другом графе знаний. Приемлемым значением дивергенции Иенсена–Шэннона считается значение 5%.

Обычно строятся две версии набора данных. Версия 1 (V1) получается путем прямого использования алгоритма IDS. Для версии 2 (V2) сначала случайным образом удаляются объекты с низкими степенями (d≤5) в графе знаний-источнике, чтобы удвоить среднюю степень, и затем выполняется IDS для соответствия новому графу знаний. В результате набор данных версии V2 вдвое плотнее, чем версии V1, и более похож на реально существующие наборы данных. В литературе по выравниванию графов знаний наиболее популярными являются немецко-английский, французско-английский и китайско-английский тестовые наборы.

Принимая во внимание то, что каждая языковая версия графа знаний имеет свою собственную структуру, отличную от других графов знаний, а также то, что данные, полученные для русскоязычного графа знаний проще интерпретировать, нами был сгенерирован русско-английский набор тестовых данных на основе русскоязычной и англоязычной версий DBpedia [2]. Использовался набор данных англоязычной и русскоязычной DBpedia за 2016 год (DBpedia 2016-10, https://wiki.dbpedia.org/downloads-2016-10).

Набор DBP-15K EN-RU (V1, V2) сгенерирован на основе алгоритма IDS и доступен для свободного скачивания (https://www.dropbox.com/sh/4oh3nkzwdr1w4dv/AACZ4v8jCdR7Y4mDtS654Bega?dl=0).

Для анализа качества работы различных алгоритмов выравнивания сущностей на основе эмбеддингов принято использовать метрики hits@*k* и среднеобратный ранг (Mean reciprocal rank, MRR). Метрика hits@*k*=*n*% означает, что для *n* процентов объектов из одного графа знаний эквивалентный объект из второго графа знаний находится среди ближайший *k* соседей в векторном пространстве. Очевидно, самой показательной считается метрика hits@1, так как эта метрика соответствует алгоритму, который самостоятельно строит правильные отношения *owl:sameAs* между сущностями. Среднеобратный ранг определяется как среднее значение обратных рангов по всем запросам. Обратный ранг в данном случае означает обратное число номера (ранга) первого правильного ответа в списке откликов.

3. ВЛИЯНИЕ ПЕРЕВОДА ЛИТЕРАЛОВ НА КАЧЕСТВО АЛГОРИТМОВ ВЫРАВНИВАНИЯ СУЩНОСТЕЙ

Первая группа экспериментов с различными методами выравнивания сущностей на англо-русском наборе данных описана в [2]. Для этих экспериментов использовались алгоритмы из библиотеки OpenEA, которые применялись к русско-английской выборке. К удивлению авторов оказалось, что наилучшие результаты выдавали такие методы, как BootEA[14] и RSNA[15], в то время как такие методы, как MultiKE и RDGCN, выдававшие наилучшие результаты на англо-французских и англо-немецких данных, давали весьма посредственные результаты. Например, метод MultiKE выдавал оценку hits@1, равную всего 37.344, в то время как этот же алгоритм на англо-французской выборке давал hits@1, равный 74.133. Аналогично метод RDGCN, который выдавал hits@1, равный 77,019 на англо-французской выборке выдавал оценку hits@1, равную всего 43,256 на русско-английских данных.

При более внимательно изучении было замечено, что наибольшее ухудшение качества результатов наблюдалось на методах выравнивания сущностей, которые при построении векторных представлений сущностей использовали не только информацию о реляционной структуре графов знаний, но и информацию

о литералах, в частности, об именах сущностей. Первое предположение о причинах неудачи было связано с тем, что английский и русский языки используют разные алфавиты, поэтому векторные представления литералов попадают в разные векторные пространства.

Для решения указанной проблемы нами был разработан инструмент автоматического перевода на основе Google Translate API. На вход программы подавались язык, с которого будет осуществлен перевод, имена сущностей и литералы. Результат перевода литералов передавался в метод формирования векторного представления.

Для сравнения методов генерации векторных представлений имен сущностей без перевода и с применением предварительного перевода были построены визуализации этих векторных представлений. Эти визуализации показаны на рисунках 2–6. В качестве инструмента снижения размерности использовался метод t-SNE [16] .

На представленных изображениях английские имена сущностей имеют синий цвет, русские – красный. Это позволяет оценить эффективность метода генерации векторных представлений. Высокая степень наложения цветов говорит о том, что семантически связанные данные, представленные на разных языках, имеют сходные векторные представления. Наличие пятен одного цвета говорит о том, что в указанной области расположены сущности из одного графа знаний, а эквивалентные сущности из другого графа знаний находятся на значительном расстоянии. На Рис. 2 показаны векторные представления (эмбеддинги) для имен сущностей из набора данных EN_RU_15K_V1, сгенерированные при помощи word2vec с оригинальными настройками MultiKE. Можно видеть, что названия сущностей из англоязычного и русскоязычного наборов данных почти не пересекаются. Пересечение наблюдается только там, где сущности из двух наборов данных имеют одинаковые англоязычные названия (например, названия музыкальных альбомов или песен).

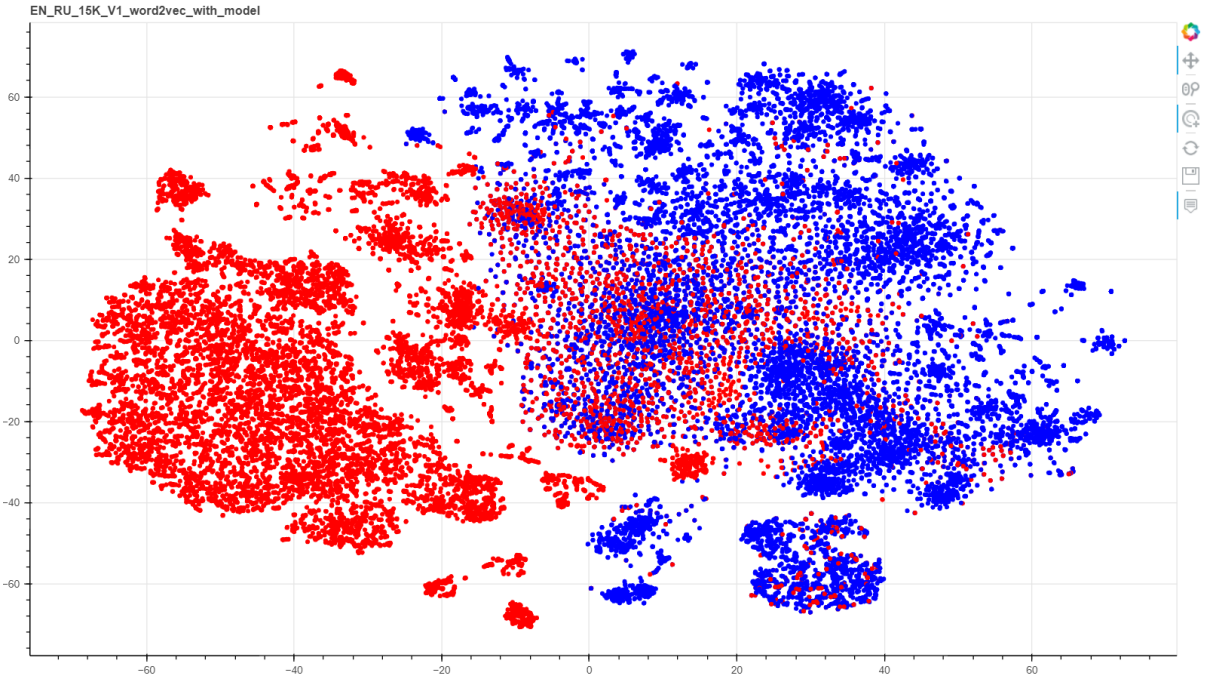

Рис. 2. Эмбеддинги имен сущностей, из набора данных EN_RU_15K_V1, сгенерированные word2vec с оригинальными настройками MultiKE (без перевода).

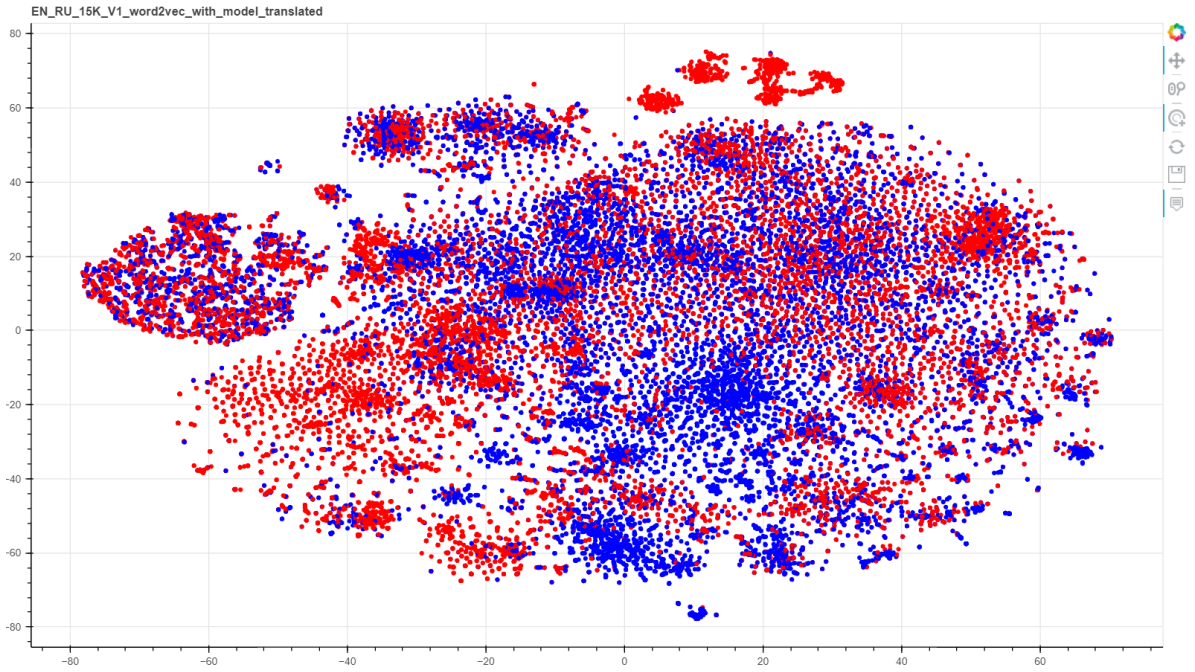

Рис. 3. Эмбеддинги имен сущностей, из набора данных EN_RU_15K_V1, сгенерированные word2vec с оригинальными настройками MultiKE (с переводом).

На рис. 3 показаны векторные представления для имен сущностей из этого же набора данных, сгенерированные word2vec относительно переведенных названий сущностей с настройками MultiKE. Можно видеть, что появилось гораздо больше пересечений синих и красных пятен, что говорит о лучшем качестве сопоставления названий сущностей. Однако результат метода генерации векторных представлений MultiKE имеет выраженные пятна одного цвета, что говорит о невысокой точности этого метода. Аналогичные результаты можно наблюдать на примере метода RDGCN.

На рис. 4 показаны векторные представления для имен сущностей из набора данных EN_RU_15K_V1, сгенерированные при помощи word2vec с оригинальными настройками RDGCN (без перевода названий сущностей).

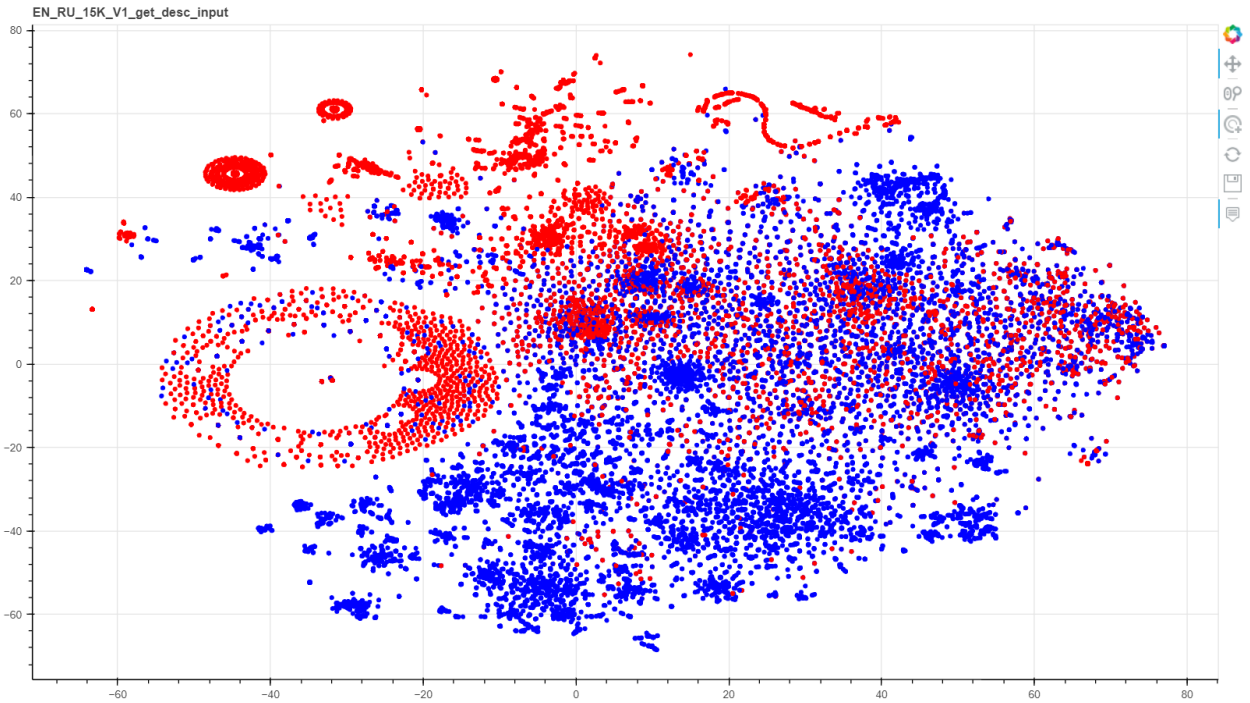

Рис. 4. Эмбеддинги имен сущностей, из набора данных EN_RU_15K_V1, построенные генератором векторных представлений имен сущностей RDGCN с оригинальными настройками (без перевода).

На Рис. 5 показаны векторные представления имен сущностей из этого же набора данных, сгенерированные word2vec относительно переведенных названий сущностей с настройками RGDCN. На этом рисунке в левом нижнем углу имеется кластер эллипсоидной формы. Он возник из-за зануления векторов слов, для

которых алгоритм из RGDCN не нашел значений в предобученной модели. В остальном же данное векторное представление имеет большую степень наложения по сравнению с RGDCN.

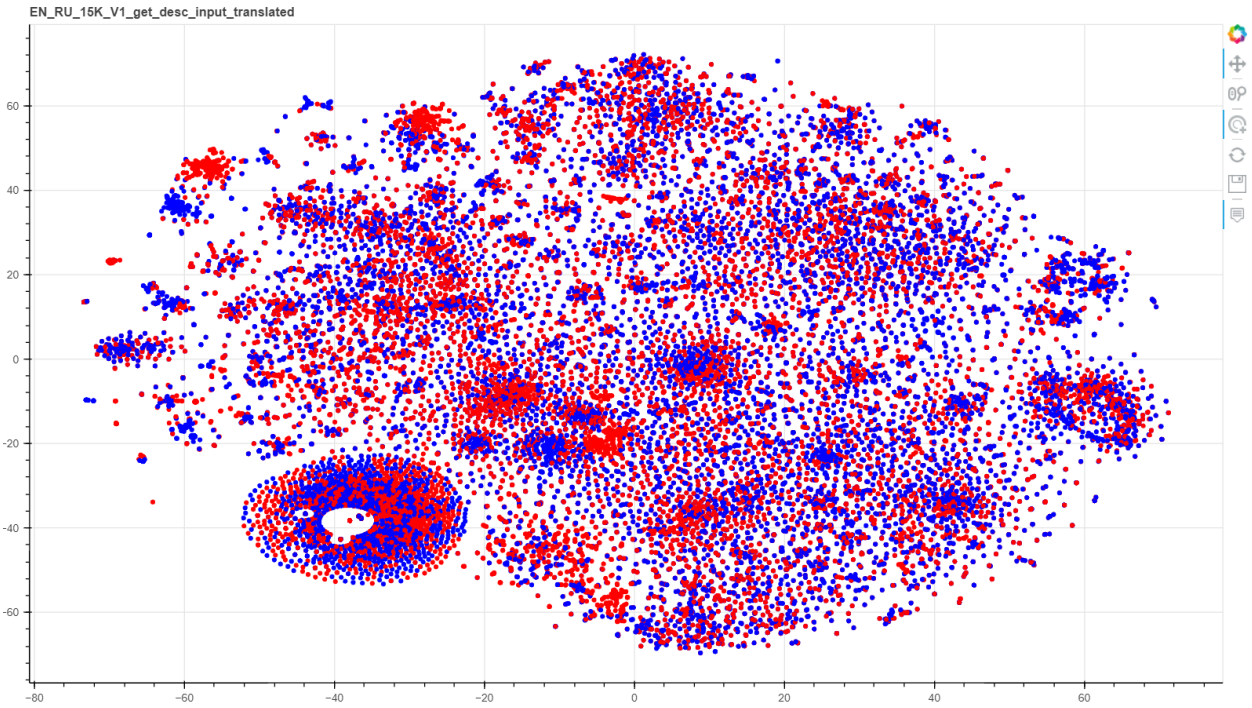

Рис. 5. Эмбеддинги имен сущностей, из набора данных EN_RU_15K_V1, построенные генератором векторных представлений имен сущностей RDGCN с оригинальными настройками и с переводом.

Наконец, на рис. 6 показаны векторные представления имен сущностей из набора данных EN_RU_15K_V1, построенные генератором векторных представлений имен сущностей SEU с оригинальными настройками и переводом. Легко видеть, что это представление имен сущностей дает наилучшее соответствие между англоязычными и русскоязычными именами сущностей. Красные пятна возникают только в случае существенного различия в названиях сущностей. Например, часть сущностей типа *Film* имеет абсолютно другое русскоязычное название. Для сравнения:

- Англоязычное название «The_Death_and_Life_of_Bobby_Z» и русскоязычное «Подстава_(фильм,_2007)»;
- Англоязычное название «The Break-Up» и русскоязычное «Развод по-американски(фильм,_2006)»;

- Англоязычное название «The_Beyond_(film)» и русскоязычное «Седьмые_врата_ада».

Понятно, что переводчик не может должным образом учитывать такие ситуации.

Аналогичная ситуация возникает с футбольными клубами. Здесь также к ошибкам встраивания могут приводить сокращения “FC@ и тег «футбольный_клуб». Например, в пространстве эмбеддингов достточно далеко расположены:

- Англоязычное название «Olympique_Club_de_Khouribga» и русскоязычное «Хурибга_(футбольный_клуб)»;
- Англоязычное название «FC_Tosno» и русскоязычное «Тосно_(футбольный_клуб)» ;
- Англоязычное название «FC_Balzers» и русскоязычное «Бальцерс_(футбольный_клуб)».

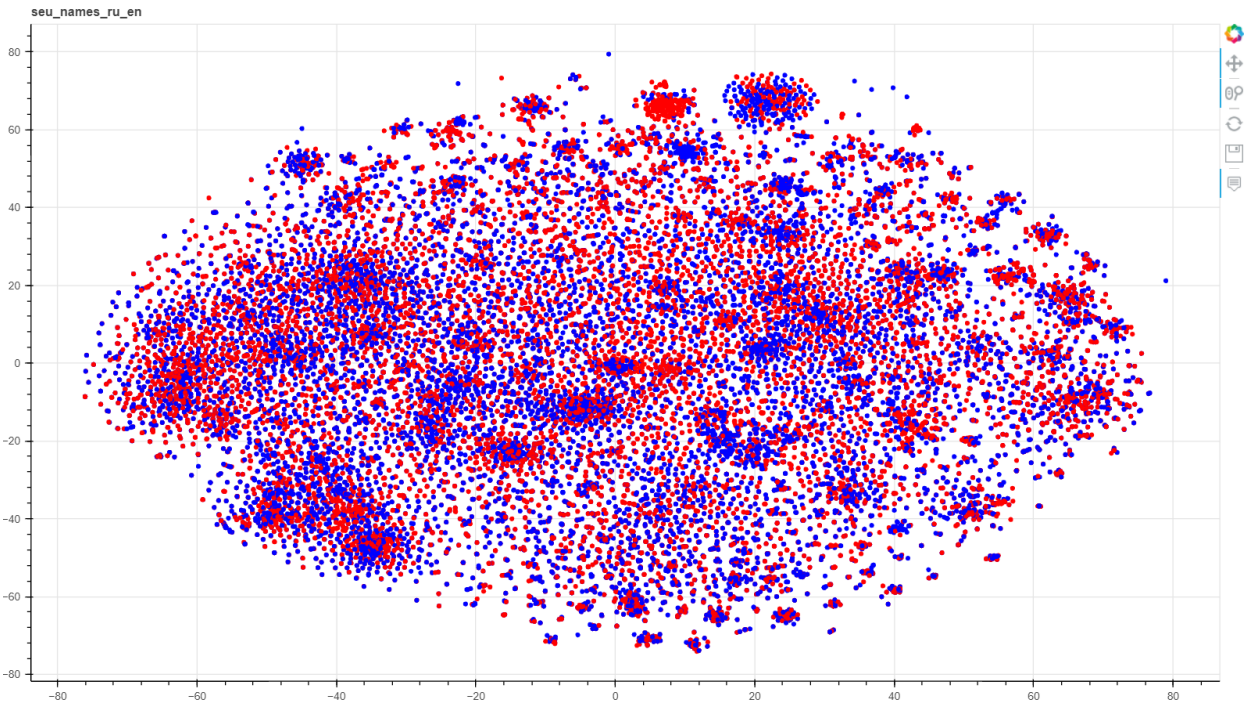

Рис. 6. Эмбеддинги имен сущностей из набора данных EN_RU_15K_V1, построенные генератором векторных представлений имен сущностей SEU с оригинальными настройками и с переводом.

Более подробно качество результатов выравнивания сущностей в зависимости от наличия или отсутствия перевода названий сущностей показано в Таблице 1. В столбце Перевод (Пер.) знаком плюс или минус обозначен факт наличия или отсутствия перевода имен сущностей.

Таблица 1. Влияние перевода имен сущностей на качество алгоритмов EA

Метод	Набор данных	Пер.	hits@1	hits@10	hits@50	mrr	Улучш.
multiKE	EN_FR_15K_V1	-	74,133	83,59	88,867	0,774435	
multiKE	EN_FR_15K_V1	+	74,619	84,324	89,638	0,779689	0,486
multiKE	EN_FR_15K_V2	-	85,495	92,057	95,276	0,878035	
MultiKE	EN_FR_15K_V2	+	85,952	92,238	95,505	0,882119	0,457
MultiKE	EN_RU_15K_V1	-	31,544	45,711	59,933	0,364153	
MultiKE	EN_RU_15K_V1	+	35,667	51,111	63,856	0,409273	4,123
MultiKE	EN_RU_15K_V2	-	45,3	62,289	74,244	0,510486	
MultiKE	EN_RU_15K_V2	+	46,478	62,289	73,956	0,519488	1,178
multiKE	EN_RU_100K_V1	-	16,262	24,038	33,145	0,190415	
Multi	EN_RU_100K_V1	+	19,568	30,158	41,72	0,232864	3,306
Rdgcn	EN_FR_15K_V1	-	77,019	89,181	92,438	0,813097	
Rdgcn	EN_FR_15K_V1	+	76,905	89,324	92,448	0,813125	-0,114
Rdgcn	EN_FR_15K_V2	-	86,19	94,848	97,124	0,895109	
Rdgcn	EN_FR_15K_V2	+	87,095	95,114	97,257	0,902504	0,905
Rdgcn	EN_RU_15K_V1	-	39,633	59,667	71,2	0,460294	
Rdgcn	EN_RU_15K_V1	+	74,378	88,211	92,322	0,791784	34,745
Rdgcn	EN_RU_15K_V2	-	53,656	71,689	80,322	0,599311	
Rdgcn	EN_RU_15K_V2	+	84,378	92,3	96,667	0,88172	30,722

Данные эксперименты показали, что перевод названий сущностей имеет существенное влияние на качество алгоритмов выравнивания сущностей, но имеются и другие параметры, влияющие на качество этих алгоритмов. Возник вопрос, влияют ли на качество методов выравнивания сущностей сами методы генерации эмбеддингов имен сущностей?

Поэтому в дальнейшем были поробно рассмотрены различные способы построения векторных представлений имен сущностей, а также комбинации различных стратегий построения векторных представлений на основе реляционных триплет с различными вариантами построения векторных представлений для имен сущностей.

4. КАЧЕСТВО МЕТОДОВ ВЫРАВНИВАНИЯ СУЩНОСТЕЙ В ЗАВИСИМОСТИ ОТ РАЗНЫХ СПОСОБОВ ПОСТРОЕНИЯ ВЕКТОРНЫХ ПРЕДСТАВЛЕНИЙ ИМЕН СУЩНОСТЕЙ

Было рассмотрено несколько моделей генерации векторных представлений имен сущностей, которые можно использовать для целей выравнивания сущностей. Было выделено несколько типов генераторов векторных представлений имен сущностей.

Генератор 1. Генерация векторных представлений уровня слов (модель word2vec) и уровня литер (модель fastText). Этот генератор применяется в методе выравнивания сущностей multiKE.

Генератор 2. Перевод имен сущностей на английский язык, генерация векторных представлений на уровне слов (модель glove.840B.300d). Этот генератор применяется в методе выравнивания сущностей RDGCN.

Генератор 3. За основу берется предположение, что не только информация о структуре окрестностей, но и текстовая информация эквивалентных сущностей обладают свойством изоморфизма. Построение векторного представления имен сущностей состоит из следующих этапов: перевод входных данных на английский язык, чтение предобученной модели предобработка входных данных, токенизация по словам, формирование биграмм, формирование векторных представлений, снижение размерности. В качестве предобученной модели использовалась glove.6B.300d. Генератор применяется в методе выравнивания SEU.

Также в качестве альтернативных методов генерации векторных представлений имен сущностей были выбраны современные модели обработки естественных языков XLNet [17] и LaBSE [18]. Спецификой этих моделей является возможность строить векторные представления для наборов слов, таких как предложения.

Генератор 4. (XLNet). Целью модели XLNet является изучение распределений для всех перестановок слов в заданной последовательности. Векторные представления формируются в рамках только одного языка, поэтому для решения нашей задачи потребовалось предварительно применить машинный перевод.

Генератор 5. (LaBSE). Данная модель генерирует независимые от языка векторные представления предложений на основе модели BERT. Представление создается путём объединения возможностей маскированного и кросс-языкового моделирований [9].

Эти пять генераторов использовались для генерации векторных представлений имен сущностей, а затем полученные представления имен сущностей встраивались в три различных алгоритма выравнивания сущностей, а именно, MultiKE, RDGCN и SEU. Для оценки качества полученных резхультатов вычислялись метрики hits@k и MRR, а также строились визуализации результатов. В таблице 2 показаны оценки качества работы трех алгоритмов выравнивания сущностей в зависимости от используемого генератора векторных представлений имен сущностей. Таблица 2 демонстрирует, что генератор векторных представлений имен сущностей на основе модели LaBSE показал себя достаточно хорошо. Он оказался эффективнее генераторов 1 и 2.

Таблица 2. Оценки качества работы алгоритмов выравнивания сущностей в зависимости от используемого генератора векторных представлений имен сущностей

Метод	Ген.	Hits@1	Hits@5	Hits@10	Hits@50	MRR
MultiKE	1	52.0	62.1	66.6	76.9	0,570
MultiKE	2	69.9	78.1	81.3	87.8	0,737
MultiKE	3	81.2	87.5	89.1	93.2	0,841
RDGCN	2	74.4	84.7	88.2	92.3	0,792
RDGCN	1	68.0	79.6	82.8	88.4	0,733
RDGCN	3	84.8	92.1	93.5	95.6	0,881
RDGCN	4	43.4	50.0	53.0	60.5	0,467
RDGCN	5	75.4	83.7	85.9	89.7	0,792
SEU	3	97.2	99.1	99.5	99.8	0,981
SEU	1	88.1	93.5	94.8	97.5	0,905
SEU	2	87.4	93.1	95.4	98.6	0,905
SEU	4	32.5	41.3	45.5	54.9	0,369
SEU	5	094.9	97.6	98.4	99.3	0,962

Тем не менее, генератор 3 оказался наиболее эффективным. Методы выравнивания сущностей MultiKE и RDGCN на его основе превысили исходные значения точности. Модель XLNet оказалась непригодной для целей выравнивания сущностей, так как полученные с ее помощью оценки точности методов выравнивания сущностей почти в два раза хуже остальных методов. Результаты подходов на ее основе близки к значениям, полученным без перевода. Эти выводы подтверждаются и визуализациями результатов методов выравнивания сущностей, использующих разные языковые модели. Для сравнения на рис. 7 показана визуализация результата работы метода выравнивания RDGCN с использованием генератора 5 (языковая модель XLNet), который соответствует наихудшим оценкам качества выравнивания. На рис. 8 показана визуализация результата работы метода выравнивания RDGCN с использованием генератора 3 (языковая модель, используемая методом SEU), который соответствует наилучшим оценкам качества.

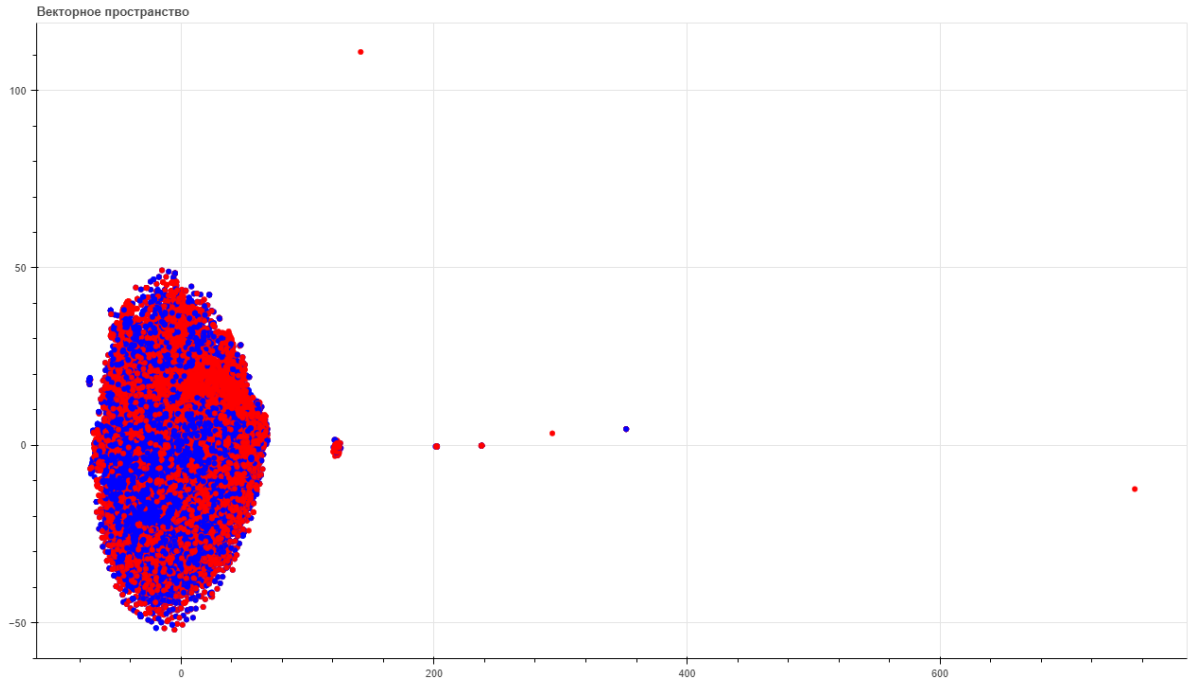

Рис. 7. Визуализация результата работы метода выравнивания RDGCN с использованием генератора 5 (языковая модель XLNet), который соответствует наихудшим оценкам качества.

Результаты применения моделей XLNet и LaBSE к MultiKE не указаны в связи с нехваткой вычислительных ресурсов для построения векторных представлений

литералов. Выводы об их эффективности сделаны на основе значений, полученных на базе других подходов.

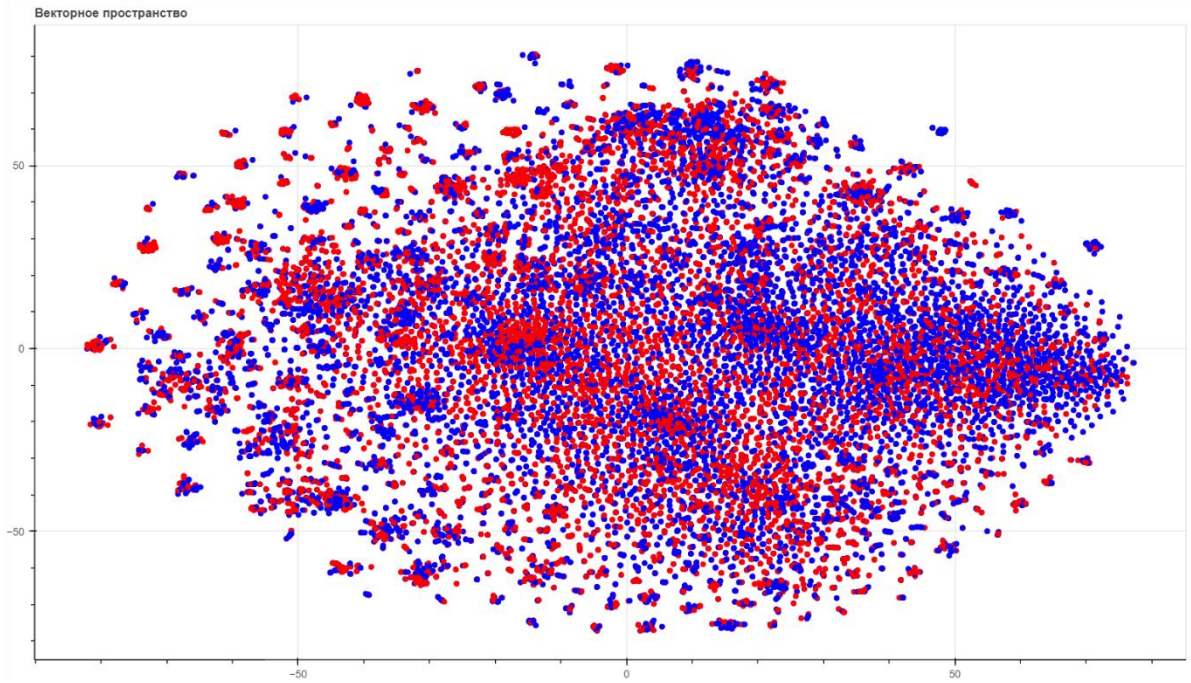

Рис. 8. Визуализация результата работы метода выравнивания RDGCN с использованием генератора 3 (языковая модель, используемая методом выравнивания SEU), который соответствует наилучшим оценкам качества выравнивания.

5. КАЧЕСТВО АЛГОРИТМОВ ВЫРАВНИВАНИЯ СУЩНОСТЕЙ В ЗАВИСИМОСТИ ОТ ТИПОВ СУЩНОСТЕЙ И КОЛИЧЕСТВА ОТНОШЕНИЙ МЕЖДУ СУЩНОСТЯМИ

Для более детального анализа качества алгоритмов выравнивания сущностей была произведена оценка метрик качества по отдельным типам сущностей, количеству отношений и атрибутов.

Для определения типов сущностей использовались файлы «instance_types». Следует отметить, что в разных языковых версиях данных DBpedia часто наблюдаются несоответствия между типами сущностей, связанных отношением *owl:sameAs*. Например, русскоязычная сущность «Эминем» отнесена к типу «MusicalArtist», а ее английский эквивалент «Eminem» относится к вышестоящему по иерархии типу «Person». Было установлено, что только шестьдесят семь процентов эквивалентных сущностей отнесены к одному и тому же типу.

Для решения указанной проблемы была написана программа установления общих типов для сущностей, связанных отношением *owl:sameAs*. Для этого при помощи SPARQL-запроса к англоязычной DBpedia была построена иерархия типов DBpedia. Для каждой пары эквивалентных сущностей осуществлялось сравнение приписанных им типов. В случае, когда ни одна сущность из пары не являлась подтипом другой, но при этом у них имелся общий надтип, им обеим приписывался последний. Например, сущность «Воеводина» относится к типу «AdministrativeRegion», а ее английский эквивалент «Vojvodina» относится к типу «Country». Данным сущностям присвоится общий тип «PopulatedPlace». В результате указанной процедуры в наборе данных EN-RU-15K (V1) было выделено семьдесят три типа сущностей.

На рисунках 8 и 9 приведены значения метрики Hits@1, показывающие качество выравнивания сущностей разного типа, полученные при помощи методов MultiKE и RDGCN. В обоих случаях использовался Генератор 3 векторных представлений имен сущностей. Информация представлена для типов, насчитывающих больше 100 сущностей.

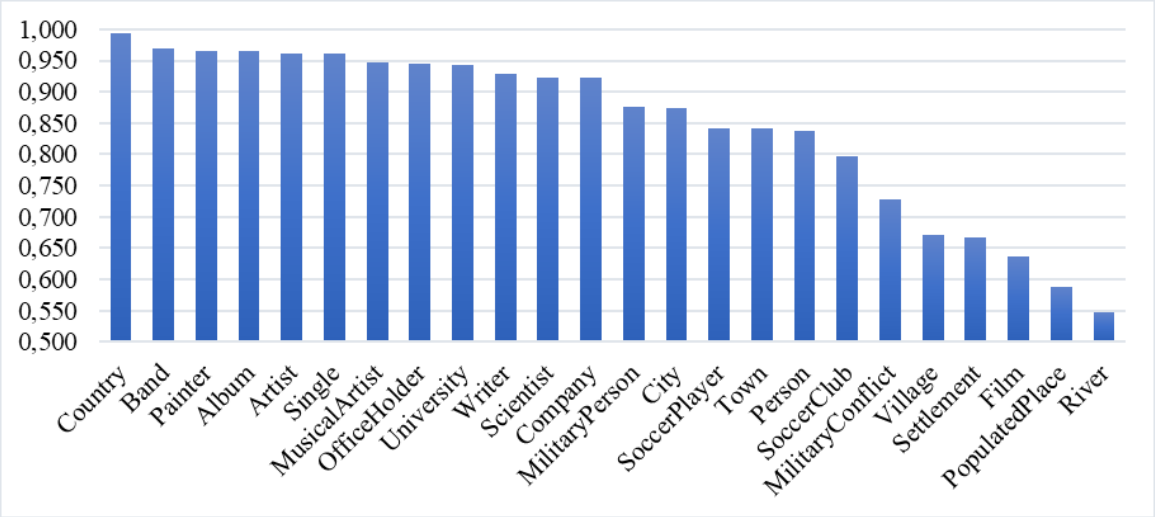

Рис. 9. Значения Hits@1 для разных типов сущностей, полученные методом MultiKE с Генератором 3.

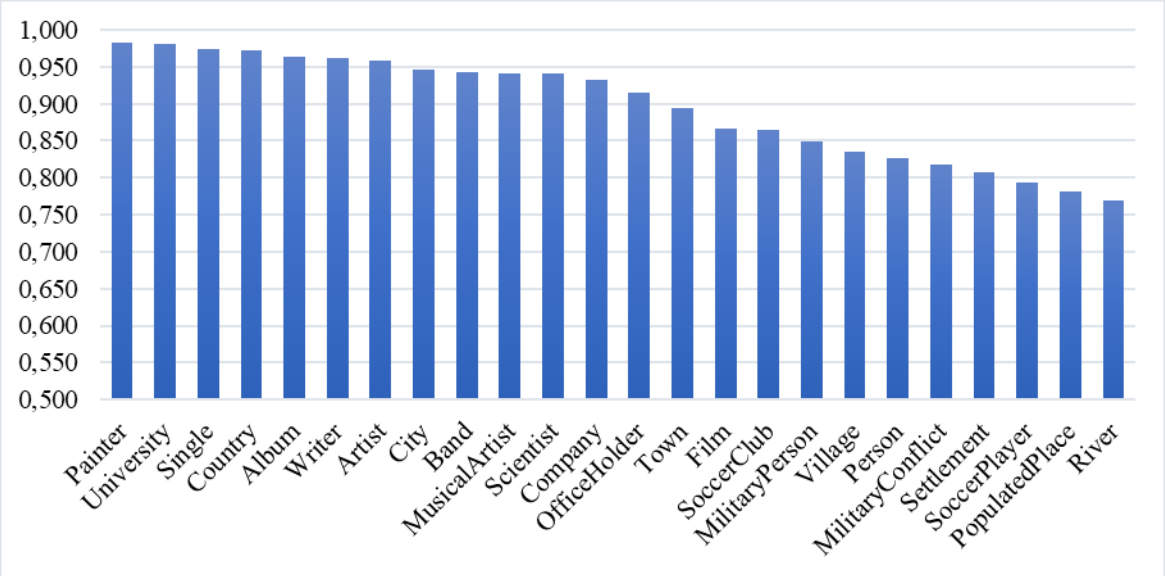

Рис. 10. Значения Hits@1 для разных типов сущностей, полученные методом RDGCN с Генератором 3.

Можно видеть, что типы, дающие наилучшие оценки точности выравнивания, похожи для обоих методов выравнивания, как и типы, дающие наихудшие оценки. При этом более равномерное распределение точности в зависимости от типов сущностей демонстрирует метод RDGCN.

Тем не менее, наилучшую точность методов выравнивания по отдельным типам показал метод SEU. Из семидесяти трех типов сорок один тип выдал оценку 100% для hits@1. Значения hits@1, hits@10 и среднеобратного ранга (MRR) для остальных типов сущностей показаны в Таблице 3.

Таблица 3. Значения hits@1, hits@10 и среднеобратного ранга (MRR) для типов сущностей, полученные методом SEU с Генератором 3.

type	hits1	hits10	Mrr
AdultActor	0	0	4,761905
Award	0	100	33,33333
Insect	44,44444	77,77778	57,48534
Mammal	60	100	76,66667
Bird	66,66667	100	80,55556
GovernmentAgency	75	100	81,25
Reptile	78,57143	100	86,90476

Noble	83,33333	100	87,5
River	86,31579	96,84211	90,60496
MusicalWork	90,90909	95,45455	92,3951
Town	92,0354	99,11504	94,69713
MilitaryConflict	94,0678	99,57627	96,13459
PopulatedPlace	94,45471	99,07579	96,18643
AdministrativeRegion	94,73684	98,68421	95,89808
Film	94,8728	98,98239	96,35096
City	94,97908	99,16318	96,99585
Writer	95,3125	99,21875	97,05116
Village	95,6044	100	97,61905
Settlement	95,78488	99,49128	97,01837
SoccerClub	96,90141	99,43662	97,75137
Royalty	97,4359	100	98,2906
Scientist	97,64706	100	98,82353
SoccerPlayer	97,66472	99,83319	98,46351
Person	97,73196	99,38144	98,37685
MilitaryPerson	98,3871	98,92473	98,72632
Band	98,64603	99,41973	98,90352
MusicalArtist	98,73817	99,68454	99,16708
Company	99,03846	99,51923	99,10003
Album	99,21569	99,66387	99,39928
OfficeHolder	99,28401	99,76134	99,43016
Artist	99,65724	100	99,80291
Single	99,7815	100	99,8689

Как и ранее, более подробный анализ показал, что наихудшие значения выравнивания возникали на сущностях с низким соответствием по названиям.

6. ЗАКЛЮЧЕНИЕ

Мы исследовали влияние методов построения векторных представлений (эмбеддингов) для имён сущностей и литералов на качество результатов различных методов выравнивания сущностей. Был исследован вклад применения перевода и современных моделей обработки естественных языков, таких как LabSE и XLnet. Для интуитивного понимания результатов было построено значительное количество визуализаций. Также эксперименты показали, что количество отношений и атрибутов сущностей в разных наборах данных не влияет на качество выравнивания. Скорее, имеет значение количество совпадающих отношений и атрибутов. В настоящее время разрабатывается новый инструмент визуализации, который позволит анализировать отношения отдельных сущностей и их влияние на качество алгоритмов выравнивания.

СПИСОК ЛИТЕРАТУРЫ

1. *Sun Z., Zhang Q., Hu W., Wang C., Chen M., Akrami F. et al.* A benchmarking study of embedding-based entity alignment for knowledge graphs // Proc. VLDB Endowment. 2020. Vol. 13. P. 2326–2340.

2. *Gnezdilova V.A., Apanovich Z.V.,* Russian-English dataset and comparative analysis of algorithms for cross-language embedding-based entity alignment // Journal of Physics: Conference Series. 2021. Vol. 2099.

3. *Zhang Q., Sun Z., Hu W., Chen M., Guo L. et al.* Multi-view knowledge graph embedding for entity alignment // Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence. 2019. P. 5429–5435.

4. *Mikolov T., Chen K., Corrado G., Dean J.* Efficient estimation of word representations in vector space, January 2013, URL: https://arxiv.org/abs/1301.3781.

5. *Bordes A., Usunier N., Garcia-Durán A, Weston J., Yakhnenko O.* Translating embeddings for modeling multi-relational data // Proceedings of the 26th International Conference on Neural Information Processing Systems. 2013. Vol. 2. P. 2787–2795.

6. *Wu Y., Liu X., Feng Y., Wang Z., Yan R., Zhao D.* Relation-aware entity alignment for heterogeneous knowledge graphs // Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence. 2019. P. 5278–5284.

7. *Veličković P., Cucurull G., Casanova A., Romero A., Liò P., Bengio Y.* Graph attention networks// ICLR. 2018. 12 p.

8. *Wang Z., Lv Q., Lan X., Zhang Y.* Cross-lingual knowledge graph alignment via graph convolutional networks // Proc. of the Conference on Empirical Methods in Natural Language Processing. 201., P. 349–357.

9. *Mao X., Wang W., Wu Y., Lan M.* From alignment to assignment: frustratingly simple unsupervised entity alignment // Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. 2021. P. 2843–2853.

10. *Xu K., Wang L., Yu M., Feng Y., Song Y., et al.* Cross-lingual knowledge graph alignment via graph matching neural network // Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019. P. 3156–3161.

11. *Pennington J, Socher R., Manning C.D.* GloVe: Global Vectors for Word Representation // Conference on Empirical Methods in Natural Language. 2014. P. 1532-1543.

12. *Bojanowski P., Grave E., Joulin A., Mikolov T.* Enriching word vectors with subword information // Transactions of the Association for Computational Linguistics. 2017. P. 135–146.

13. *Devlin J., Chang M.-W., Lee K., Toutanova K.* BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding // Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2019. Vol. 1. P. 4171–4186.

14. *Fuglede B., Topsoe F.* Jensen–Shannon divergence and Hilbert space embedding // Proceedings of the International Symposium on Information Theory, 2004. IEEE.

15. *Sun Z., Hu W., Zhang Q., Qu Y.* Bootstrapping entity alignment with knowledge graph embedding // Proc. 27th International Joint Conference on Artificial Intelligence (IJCAI-18), P. 4396–4402.

16. *Guo L., Sun Z., Hu W.* Learning to Exploit Long-term relational dependencies in knowledge graphs // Proceedings of the 36th International Conference on Machine Learning. 2019. Vol. 57. P. 2505–2514.

17. *Maaten L. van der, Hinton G.* Visualizing data using t-SNE // Journal of Machine LearningResearch. 2008. Vol. 86. P. 2579–2605.

18. *Yang Z., Dai Z., Yang Y., Carbonell J., Salakhutdinov R. et al.* XLNet: generalized autoregressive pretraining for language understanding // Proceedings of the 33rd International Conference on Neural Information Processing Systems. 2019. P. 5753–5763.

19. *Feng F., Yang Y., Cer D., Arivazhagan N0, Wang W.* Language-agnostic BERT sentence embedding. 2020. URL: https://arxiv.org/abs/2007.01852.

HOW ENTITY NAME EMBEDINGS AFFECT THE QUALITY OF ENTITY ALIGNMENT

D. I. Gusev[1] [0000-0001-9636-2783], Z. V. Apanovich[2] [0000-0002-5767-284X]

[1]Novosibirsk State University, Novosibirsk;
[2]A.P. Ershov Institute of Informatics Systems, Siberian Branch of the Russian Academy of Sciences, Novosibirsk State University, Novosibirsk

[1]d.gusev1@g.nsu.ru, [2]apanovich_09@mail.ru

Abstract

Cross-lingual entity alignment algorithms are designed to look for identical real-world objects in multilingual knowledge graphs. This problem occurs, for example, when searching for drugs manufactured in different countries under different names, or when searching for imported equipment. At the moment, there are several open-source libraries that collect implementations of entity alignment algorithms as well as test data sets for various languages. This paper describes experiments with several popular entity alignment algorithms applied to a Russian-English dataset. In addition to translating entity names from Russian to English, experiments on combining the various generators of entity name embeddings with the various generators of relational information embeddings have been conducted. In order to obtain more detailed information about the results of the EA approaches, an assessment by entity types, the number of relationships and attributes have been made. These experiments allowed us to significantly improve the accuracy of several EA algorithms on the English-Russian dataset.

Keywords multi-lingual knowledge graphs, identity resolution, cross-lingual entity alignment, relational embeddings, name embeddings correctness

REFERENCES

1. *Sun Z., Zhang Q., Hu W., Wang C., Chen M., Akrami F. et al.* A benchmarking study of embedding-based entity alignment for knowledge graphs // Proc. VLDB Endowment. 2020. Vol. 13. P. 2326–2340.

2. *Gnezdilova V.A., Apanovich Z.V.,* Russian-English dataset and comparative analysis of algorithms for cross-language embedding-based entity alignment // Journal of Physics: Conference Series. 2021. Vol. 2099.

3. *Zhang Q., Sun Z., Hu W., Chen M., Guo L. et al.* Multi-view knowledge graph embedding for entity alignment // Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence. 2019. P. 5429–5435.

4. *Mikolov T., Chen K., Corrado G., Dean J.* Efficient estimation of word representations in vector space, January 2013, URL: https://arxiv.org/abs/1301.3781.

5. *Bordes A., Usunier N., Garcia-Durán A, Weston J., Yakhnenko O.* Translating embeddings for modeling multi-relational data // Proceedings of the 26th International Conference on Neural Information Processing Systems. 2013. Vol. 2. P. 2787–2795.

6. *Wu Y., Liu X., Feng Y., Wang Z., Yan R., Zhao D.* Relation-aware entity alignment for heterogeneous knowledge graphs // Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence. 2019. P. 5278–5284.

7. *Veličković P., Cucurull G., Casanova A., Romero A., Liò P., Bengio Y.* Graph attention networks// ICLR. 2018. 12 p.

8. *Wang Z., Lv Q., Lan X., Zhang Y.* Cross-lingual knowledge graph alignment via graph convolutional networks // Proc. of the Conference on Empirical Methods in Natural Language Processing. 201., P. 349–357.

9. *Mao X., Wang W., Wu Y., Lan M.* From alignment to assignment: frustratingly simple unsupervised entity alignment // Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. 2021. P. 2843–2853.

10. *Xu K., Wang L., Yu M., Feng Y., Song Y., et al.* Cross-lingual knowledge graph alignment via graph matching neural network // Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019. P. 3156–3161.

11. *Pennington J, Socher R., Manning C.D.* GloVe: Global Vectors for Word Representation // Conference on Empirical Methods in Natural Language. 2014. P. 1532-1543.

12. *Bojanowski P., Grave E., Joulin A., Mikolov T.* Enriching word vectors with subword information // Transactions of the Association for Computational Linguistics. 2017. P. 135–146.

13. *Devlin J., Chang M.-W., Lee K., Toutanova K.* BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding // Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2019. Vol. 1. P. 4171–4186.

14. *Fuglede B., Topsoe F.* Jensen–Shannon divergence and Hilbert space embedding // Proceedings of the International Symposium on Information Theory, 2004. IEEE.

15. *Sun Z., Hu W., Zhang Q., Qu Y.* Bootstrapping entity alignment with knowledge graph embedding // Proc. 27th International Joint Conference on Artificial Intelligence (IJCAI-18), P. 4396–4402.

16. *Guo L., Sun Z., Hu W.* Learning to Exploit Long-term relational dependencies in knowledge graphs // Proceedings of the 36th International Conference on Machine Learning. 2019. Vol. 57. P. 2505–2514.

17. *Maaten L. van der, Hinton G.* Visualizing data using t-SNE // Journal of Machine LearningResearch. 2008. Vol. 86. P. 2579–2605.

18. *Yang Z., Dai Z., Yang Y., Carbonell J., Salakhutdinov R. et al.* XLNet: generalized autoregressive pretraining for language understanding // Proceedings of the 33rd International Conference on Neural Information Processing Systems. 2019. P. 5753–5763.

19. *Feng F., Yang Y., Cer D., Arivazhagan N0, Wang W.* Language-agnostic BERT sentence embedding. 2020. URL: https://arxiv.org/abs/2007.01852.

СВЕДЕНИЯ ОБ АВТОРАХ

ГУСЕВ Даниил Иванович – магистрант Новосибирского государственного университета. Сфера научных интересов – визуализация информации, Semantic Web.

Daniil Ivanovic GUSEV – Masters student of Novosibirsk State University. Research interests include information visualization, Semantic Web.

email: d.gusev1@g.nsu.ru

ORCID: 0000-0001-9636-2783

АПАНОВИЧ Зинаида Владимировна – старший научный сотрудник Института Систем Информатики СО РАН, доцент Новосибирского государственного университета. Сфера научных интересов – визуализация информации, визуализация графов, Semantic Web.

Zinaida Vladimirovna APANOVICH – senior researcher of the Institute of Informatics Systems of SB RAS, Associate Professor of Novosibirsk State University. Research interests include information visualization, graph visualization, Semantic Web.

email: apanovich@iis.nsk.su

ORCID: 0000-0002-5767-284X

Материал поступил в редакцию 12 января 2023 года