ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет информатики, математики и компьютерных наук

Программа подготовки бакалавров по направлению
01.03.02 Прикладная математика и информатика

Шустров Дмитрий Михайлович

КУРСОВАЯ РАБОТА

Пространственные вложения графов знаний

Научный руководитель
Ст. преподаватель

А.А. Пономаренко

Нижний Новгород, 2022

Оглавление

Введение 3

1. Граф знаний 4

2. Функция энергии 5

2.1 Обозначения 5

2.2 Ключевые идеи 5

2.3 Параметризация нейройнной сетью 6

2.3.1 Линейная форма энергии 8

2.3.2 Билинейная форма энергии 9

2.3.3 Общие детали 10

3. Обучение нейронной сети 11

3.1 Критерий обучения 11

3.2 Алгоритм обучения 12

3.2.1 Детали реализации 12

4. Эмпирическое сравнение энергий 14

4.1 Набор данных 14

4.2 Пространственные вложения сущностей 17

4.2.1 Методы проверки результатов 17

4.2.2 Графическая интерпретация 20

5. Заключение 22

Список литературы **23**

Введение

Мультиреляционные данные, которые относятся к графам, узлы которых представляют сущности, а ребра соответствуют отношениям, которые связывают эти сущности, играют ключевую роль во многих областях, таких как системы управления, Семантическая сеть или вычислительная биология. Отношения моделируются в графах знаний, где отношение либо моделирует связь между двумя сущностями, либо между сущностью и значением атрибута. Зачастую, для нахождения более комплексных знаний в готовой сети, необходимо закодировать граф знаний понятным для компьютеря языком - с помощью пространственных вложений.

Целью данной курсовой работы является изучение метода отображения графа знаний в векторные вложения с помощью функции энергии[1, 2].

В соответствии с целью были поставлены задачи:

1. Изучить математические модели, позволяющие осуществить получение векторных представлений, соответсвующих семантическим особенностям графа знаний (линейная и билинейная функция энергии).

2. Изучить процесс обучения параметризированной нейронной сетью функции энергии.

3. Реализовать получение пространственных вложений графа графа знаний с помощью функции энергрии на языке Python и фреймворке PyTorch.

4. Проверить качество полученных вложений (полученных семантических связей), создав диаграмму, на которой будут изображены географические данные из графа знаний WordNet.

1. Граф знаний

Граф знаний - специальная база знаний, которая использует графовую структуру для хранения данных. Чаще всего узлами такого графа являются объекты реального мира (например события, ситуации, решения или абстрактные концепты), а переходы - типизированные отношения между объектами (например, отношение для места рождения отличается от отношения даты рождения). Чтобы обеспечить возможность использования графов зна-

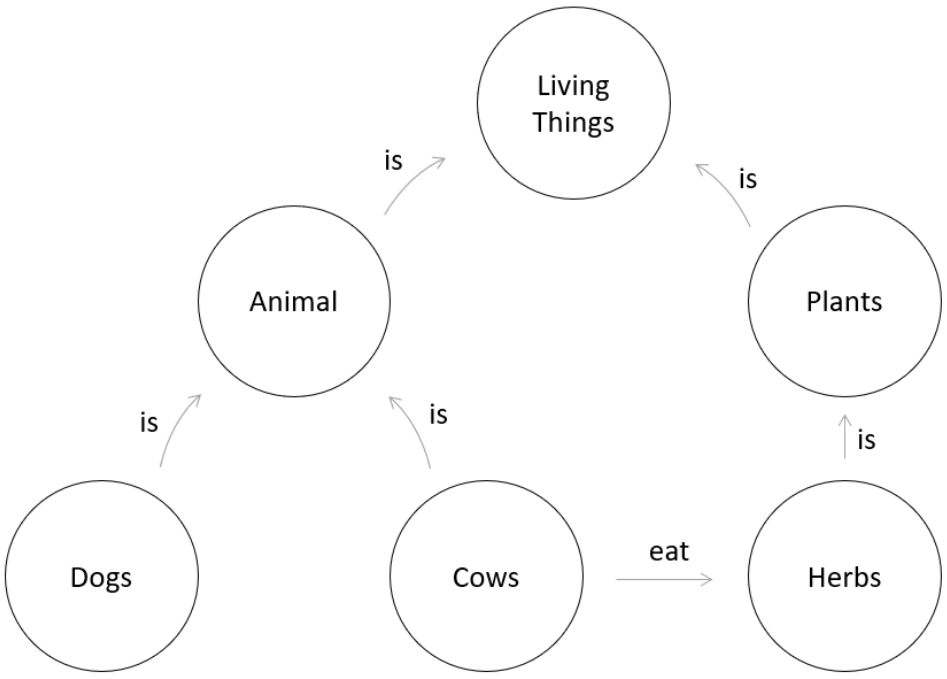

Рис. 1.1. Пример простейшего графа знаний.

ний в различных задачах машинного обучения, было разработано несколько методов получения скрытых представлений объектов и отношений. Эти вложения в графы знаний позволяют подключать их к методам машинного обучения, для которых требуются векторы объектов, такие как вложения слов. Это может дополнить другие оценки концептуального сходства. Одним из таких методов является преобразование через функцию энергии (гл. 2).

2. Функция энергии

В этой главе представлен разбор математической модели TransE[1, стр. 237], разработанная для пространственного вложения мультиреляционных данных в полностью распределенные представления с помощью функции энергии.

2.1. Обозначения

В этой работе рассматриваются графы знаний в качестве мульреляционных баз данных. Данные представлены в виде двух множеств - сущностей и типов связи. Пусть $\mathcal{C}$ - словарь, который включает в себя элементы всех базовых множеств, и пусть $\mathcal{R} \subset \mathcal{C}$ - подмножество типов связи. Пусть (lhs, rel, rhs) - тройка, обозначающая семантическую связь (субъект lhs относится свойством rel к объекту rhs).

2.2. Ключевые идеи

Основные идеи, описывающие функцию энергии семантического соответствия, следующие:

- Именованные символьные сущности и их типы связи ставятся в соотвествие с d-мерным веректорным пространством, называемым «пространство вложений» или «пространство эмбеддингов» (i-я сущность приравнивается к вектору $E_i \in \mathbb{R}^d$). Следует отметить, что существуют куда более общие методы осуществления таких отображений (например, если сущность можно прдставить в виде структурированного объекта или вектора x, то отображение может быть осуществлено параметризированной функцией e, которая переводит x в E_x равное $e_\theta(x)$, где θ получено с помощью нейронной сети. По такому принципу работает модель Word2Vec).

- Именованные символьные сущности группируются в тройки вида (lhs, rel, rhs). В соответствие такой тройке ставится значение параметризированной функции $\mathcal{E}$, которая специальным образом группирует и комбинирует сущности в тройке. В названии функции используется термин «семантическое соответствие» так как она полагается на критерий соответствия, вычисленный между обеими сторонами триплета (тройки).

- Функция энергии $\mathcal{E}$ оптимизирована таким образом, чтобы иметь малые значения для триплетов из тренировочного набора (для остальных, очевидно, это свойство выполняться не должно). Основываясь на такой функции, мы сможем определять недостающие элементы в тройках. Например, для тройки $(lhs, ?, rel)$, в которой нет типа связи rel, среди всех возможных типов связи мы выберем тот, с которым наша тройка имеет наименьшую энергию из всех возможных. Процесс такой оптимизации функции описывается в гл. 3.

2.3. Параметризация нейройнной сетью

Функция энергии $\mathcal{E}$ (далее **SME**) закодирована с помощью нейронной сети. Что дает возможность интуитивного определения недостающих элементов триплетов, а также переноса всех триплетов в одно пространство для сравнений (на рис. 2.1 изображен этот процесс). Следовательно, пары (lhs, rel) и (rel, rhs) сперва комбинируются по отдельности, затем производится слияние двух семантических представлений, полученных на предыдущем шаге. Вычисление функции энергии семантического соответствия $\mathcal{E}$ состоит из трех основных частей (рис. 2.1):

1. Вместо символьных сущностей из исходного набора используются индексы этих сущностей (например, для набора $\{cow, animal, is\}$ тройка будет выглядеть как $(0, 2, 1)$). Каждая сущность (lhs, rel, rhs) переводится эмбеддинги $E_{lhs}, E_{rel}, E_{rhs} \in \mathbb{R}^d$ соответственно.

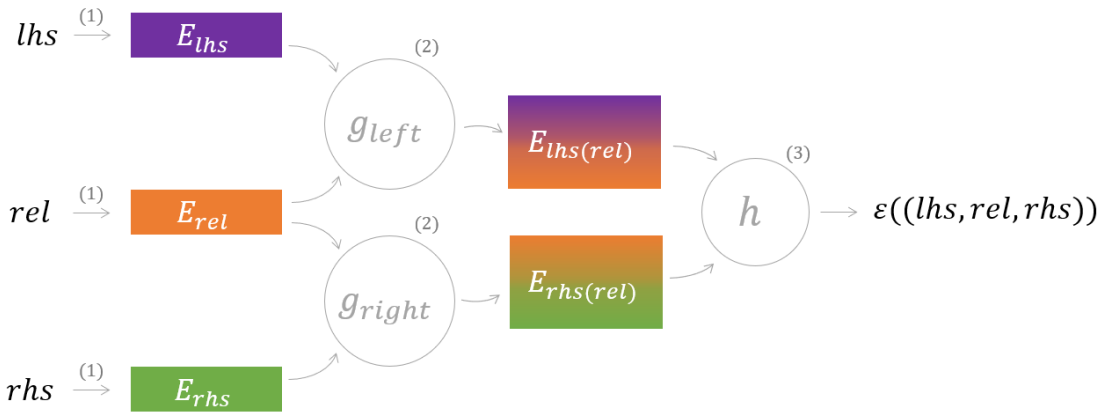

Рис. 2.1. Функция энергии семантического соответствия.

2. Для формирования комбинированного эмбеддинга основанного на типе связи, используется функция g (для пар (E_{lhs}, E_{rel}) и (E_{rhs}, E_{rel}) мы получим новые эмбеддинги $E_{lhs(rel)}$ и $E_{rhs(rel)}$, как $E_{lhs(rel)} = g_{left}(E_{lhs}, E_{rel})$ и $E_{rhs(rel)} = g_{right}(E_{rhs}, E_{rel})$).

3. Результирующая энергия равна $h(E_{lhs(rel)}, E_{rhs(rel)})$, где h - оператор скалярного умножения матриц. В качестве h можно взять и другую более сложную функцию с заранее обученными параметрами.

С таким определением, мы можем обрабатывать различные наборы входных данных, подбирая наиболее выгодные функции g и h. В качестве g мы рассмотрим две раличных реализации (разд. 2.3.1 и разд. 2.3.2), которые по разному реагируют на структурные особенности триплетов. Для h, как было сказано ранее, будет использоваться скалярное матричное умножение, потому что оно простое и не тратит большого объема ресурсов. Таким образом, функция энергии семантического соответствия $\mathcal{E}$ равна $h(g_{left}(E_{lhs}, E_{rel}), g_{right}(E_{rhs}, E_{rel}))$.

2.3.1. Линейная форма энергии

Линейная форма (далее $\mathbf{SME}_{lin}$). В этом случае функции g являются обычными линейными слоями:

$$E_{lhs(rel)} = g_{left}(E_{lhs}, E_{rel}) = W_{l1}E_{lhs}^{\intercal} + W_{l2}E_{rel}^{\intercal} + b_l^{\intercal},$$
$$E_{rhs(rel)} = g_{right}(E_{rhs}, E_{rel}) = W_{r1}E_{rhs}^{\intercal} + W_{r2}E_{rel}^{\intercal} + b_r^{\intercal}.$$

где W_{l_1}, W_{l_2}, W_{r_1}, $W_{r_2} \in \mathbb{R}^{p\times d}$ - веса и b_l, $b_r \in \mathbb{R}^p$ - нейроны смещения и $E^{\intercal}$ обозначает операцию транспонирования матрицы E. Это приводит к следующей форме энергии (под x подразумевается тройка (lhs, rel, rhs)):

$$\mathcal{E}(x) = -(W_{l_1}E_{lhs}^{\intercal} + W_{l_2}E_{rel}^{\intercal} + b_l^{\intercal})^{\intercal}(W_{r_1}E_{rhs}^{\intercal} + W_{r_2}E_{rel}^{\intercal} + b_r^{\intercal}). \tag{2.1}$$

Преобразуем (2.1) к более простой форме:

$$\begin{aligned}
\mathcal{E}(x) &= -((W_{l_1}E_{lhs}^{\intercal})^{\intercal} + (W_{l_2}E_{rel}^{\intercal})^{\intercal} + (b_l^{\intercal})^{\intercal})(W_{r_1}E_{rhs}^{\intercal} + W_{r_2}E_{rel}^{\intercal} + b_r^{\intercal}) = \\
&= -(E_{lhs}W_{l_1}^{\intercal} + E_{rel}W_{l_2}^{\intercal} + b_l)(W_{r_1}E_{rhs}^{\intercal} + W_{r_2}E_{rel}^{\intercal} + b_r^{\intercal}) = \\
&= - E_{lhs}W_{l_1}^{\intercal}W_{r_1}E_{rhs}^{\intercal} - E_{lhs}W_{l_1}^{\intercal}W_{r_2}E_{rel}^{\intercal} - E_{lhs}b_l^{\intercal} \\
&\quad - E_{rel}W_{l_2}^{\intercal}W_{r1}E_{rhs}^{\intercal} - E_{rel}W_{l_2}^{\intercal}W_{r_2}E_{rel}^{\intercal} - E_{rel}b_r^{\intercal} \\
&\quad - b_lW_{r_1}E_{rhs}^{\intercal} - b_lW_{r_2}E_{rel}^{\intercal} - b_lb_r^{\intercal}.
\end{aligned}$$

Взяв $\tilde{W}_1 = W_{l_1}^{\intercal}W_{r_1}, \tilde{W}_2 = W_{l_1}^{\intercal}W_{r_2}, \tilde{W}_3 = W_{l_2}^{\intercal}W_{r_1}, \tilde{W}_4 = W_{l_2}^{\intercal}W_{r_2} \in \mathbb{R}^{d\times d}$, получим итоговое представление $\mathbf{SME}_{lin}$ (слагаемые с нейронами смещения исключены для упрощения записи и понимания смысла полученной формы):

$$\mathcal{E}(x) = -E_{lhs}\tilde{W}_1E_{rhs}^{\intercal} - E_{lhs}\tilde{W}_2E_{rel}^{\intercal} - E_{rel}\tilde{W}_3E_{rhs}^{\intercal} - E_{rel}\tilde{W}_4E_{rel}^{\intercal}. \tag{2.2}$$

Из (2.2) следует, что энергия состоит из трех частей, кодирующих пары $(lhs, rhs), (lhs, rel)$ и (rel, rhs), и дополнительного квадратичного члена (rel, rel). Основываясь на этом, можно сделать вывод о том, что $\mathbf{SME}_{lin}$ представляет тройку как комбинацию попарных пересечений всех элементов.

2.3.2. Билинейная форма энергии

Билинейная форма (далее $\mathbf{SME}_{bil}$). В этом случае функции g будут использовать тензоры третьего ранга в качестве весов:

$$E_{lhs(rel)} = g_{left}(E_{lhs}, E_{rel}) = (W_l \bar{\times}_3 E_{rel}^\intercal) E_{lhs}^\intercal + b_l^\intercal,$$
$$E_{rhs(rel)} = g_{right}(E_{rhs}, E_{rel}) = (W_r \bar{\times}_3 E_{rel}^\intercal) E_{rhs}^\intercal + b_r^\intercal.$$

где W_l, $W_r \in \mathbb{R}^{p \times d \times d}$ - веса и b_l, $b_r \in \mathbb{R}^p$ - нейроны смещения. $\bar{\times}_3$ обозначает тензор-векторное произведение с суммированием вдоль 3-го индекса, которое, для $U, V \in \mathbb{R}^p$ и $W \in \mathbb{R}^{p \times d \times d}$, определено как (строчные буквы обозначают элементы вектора/тензора):

$$\forall i \in 1, \ldots, p, \quad ((W \bar{\times}_3 V^\intercal) U^\intercal)_i = \sum_{j=1}^{d} (\sum_{k=1}^{d} w_{ijk} v_k) u_j = \sum_{j=1}^{d} \sum_{k=1}^{d} w_{ijk} v_k u_j.$$

Таким образом, получаем следующую форму энергии:

$$\mathcal{E}(x) = -((W_l \bar{\times}_3 E_{rel}^\intercal) E_{lhs}^\intercal + b_l^\intercal)^\intercal ((W_r \bar{\times}_3 E_{rel}^\intercal) E_{rhs}^\intercal + b_r^\intercal). \tag{2.3}$$

Преобразуем (2.3) к более простой форме:

$$\begin{aligned}
\mathcal{E}(x) &= -(((W_l \bar{\times}_3 E_{rel}^\intercal) E_{lhs}^\intercal)^\intercal + (b_l^\intercal)^\intercal)((W_r \bar{\times}_3 E_{rel}^\intercal) E_{rhs}^\intercal + b_r^\intercal) = \\
&= -(E_{lhs}^\intercal (W_l \bar{\times}_3 E_{rel}^\intercal)^\intercal + b_l)((W_r \bar{\times}_3 E_{rel}^\intercal) E_{rhs}^\intercal + b_r^\intercal) = \\
&= - E_{lhs}^\intercal (W_l \bar{\times}_3 E_{rel}^\intercal)^\intercal (W_r \bar{\times}_3 E_{rel}^\intercal) E_{rhs}^\intercal \\
&\quad - E_{lhs}^\intercal (W_l \bar{\times}_3 E_{rel}^\intercal)^\intercal b_r^\intercal - b_l (W_r \bar{\times}_3 E_{rel}^\intercal) E_{rhs}^\intercal - b_l b_r^\intercal.
\end{aligned}$$

Сделаем замену $\tilde{W}_{rel} = (W_l \bar{\times}_3 E_{rel}^\intercal)^\intercal (W_r \bar{\times}_3 E_{rel}^\intercal)$, получим итоговое представление $\mathbf{SME}_{bil}$ (слагаемые с нейронами смещения исключены для упрощения записи и понимания смысла полученной формы):

$$\mathcal{E}(x) = -E_{lhs}^\intercal \tilde{W}_{rel} E_{rhs}^\intercal. \tag{2.4}$$

По виду выражения (2.4) очевидно, что $\mathbf{SME}_{bil}$ составлена из единственного слагаемого, которое зависит от всех элементов тройки. *Билинейная* форма представляет из себя оновременное пересечение всех сущностей, в то время как *линейная* комбинирует объекты последовательно.

2.3.3. Общие детали

Определены две функциии энергии, подходящие для пространственного вложения мультиреляционных данных в распределенные представления. Выбор между *биграммой* ($\mathbf{SME}_{lin}$) и *триграммой* ($\mathbf{SME}_{bil}$) для функции g не совсем очевиден. Форма триграммы может обрабатывать сложные тернарные пересечения, но требует обработки большего числа параметров. Ассимтотическая сложность вычислений равна:

$$\mathcal{O}(n_e d + n_r d + p d^2) - \text{для триграммы},$$
$$\mathcal{O}(n_e d + n_r d + p d) - \text{для биграммы}.$$

где n_e - колличесво сущностей, не являющихся типом связи, n_r - количество типов связи, d и p - размерности внутреннего и внешнего слоев пространственного вложения соответсвенно. Обратим внимание, что обе структуры могут с легкостью расширяться до достаточно большого количества связей ($n_r \gg 1$) без обработки громоздких дополнительных параметров. Более детально сравнение двух энергий рассмотрено в гл. 4.

3. Обучение нейронной сети

В этой главе описан процесс построения оптимизированной функциии энергии **SME** с использованием методов обучения нейронных сетей.

3.1. Критерий обучения

Тренировочный набор $\mathcal{D}$ содержит m триплетов $x = (x_{lhs}, x_{rel}, x_{rhs})$, где $x_{lhs}, x_{rhs} \in \mathcal{C}$ и $x_{rel} \in \mathcal{R}$. Напомним, что энергия триплета обозначена как $\mathcal{E}(x) = \mathcal{E}(x_{lhs}, x_{rel}, x_{rhs})$. Также, энергия должна иметь малые значения для триплетов из тренировочного набора (подробнее в разд. 2.2). Для достижения этой цели рассмотрим некоторое количество триплетов, которые не входят в тренировочный набор; будем называть такие тройки $\tilde{x}$ *негативными*, а тройки из набора - *нормальными*.

$$\begin{aligned}
&\forall x = (x_{lhs}, x_{rel}, x_{rhs}) \in \mathcal{D} : \\
&\mathcal{E}(x) < \mathcal{E}(i, x_{rel}, x_{rhs}) \quad \forall i \in \mathcal{C} : (i, x_{rel}, x_{rhs}) \notin \mathcal{D}, \\
&\mathcal{E}(x) < \mathcal{E}(x_{lhs}, i, x_{rhs}) \quad \forall i \in \mathcal{R} : (x_{lhs}, i, x_{rhs}) \notin \mathcal{D}, \\
&\mathcal{E}(x) < \mathcal{E}(x_{lhs}, x_{rel}, i) \quad \forall i \in \mathcal{C} : (x_{lhs}, x_{rel}, i) \notin \mathcal{D}.
\end{aligned} \tag{3.1}$$

Далее, нужно минимизировать следующий стохастический критерий (англ. hinge loss, используется для обучения векторных классификаторов):

$$\sum_{x \in \mathcal{D}} \sum_{\tilde{x} \sim Q(\tilde{x}|x)} \max(\mathcal{E}(x) - \mathcal{E}(\tilde{x}) + 1, 0) \tag{3.2}$$

где $Q(\tilde{x}|x)$ - процесс, генерирущий множесво негативных триплетов из нормального, беря в расмотрения все сущности тренировочного набора $\mathcal{D}$. Стоит также обратить внимание, что нам не нужно проверять наличие негативного триплета в тренировочном наборе, так как для процесса Q выполняется условие симметрии $Q((\tilde{a}, b, c)|(a, b, c)) = Q((a, b, c)|(\tilde{a}, b, c))$ и для остльных элементов тройки условие остается истинным.

3.2. Алгоритм обучения

Для обучения приближенных к оптимальным параметров функции энергрии $\mathcal{E}$ нам нужно перебрать все триплеты из тренировочного набора $\mathcal{D}$ и использовать стохастический градиентный спуск. Более формально, нужно сделать следующее:

1. Случайным образом разбить тренировочный набор $\mathcal{D}$ на непересекающиеся группы, котоыре принято называть «батчи» (далее $\mathcal{B}_i$ - тренировочный батч с индексом i).
2. Выбрать триплет $x_i = (lhs_i, rel_i, rhs_i)$ случайным образом из $\mathcal{B}_j$.
3. Выбрать одно из ограничивающих условий (3.1).
4. Сгенерировать негативный триплет $\tilde{x}_i$, выбрав одну сущность из $\mathcal{R}$ для замены rel_i или из $\mathcal{C}/\mathcal{R}$ для замены lhs_i или rhs_i.
5. Если $\mathcal{E}(x_i) > \mathcal{E}(\tilde{x}_i) - 1$, то выполнить шаг стохастического градиентного спуска для минимизации критерия (3.2).
6. Нормализовать полученные эмбеддинги, $||E_i|| = 1, \forall i$.

Каждое обновление параметров модели выполняется с помощью обратного распространения ошибки. Вычислительная сложность для $\mathbf{SME}_{bil}$ равна $\mathcal{O}(pd^2)$ - тензор-векторное произведение весов, и для $\mathbf{SME}_{lin}$ равна $\mathcal{O}(pd)$ - матричное произведение весов. Соответственно, если в тренировочном наборе $\mathcal{D}$ имеется m триплетов, то для обучения одной эпохи требуется порядка $\mathcal{O}(mpd^2)$ для билинейной формы и $\mathcal{O}(mpd)$ для линейной формы энергии. Обратим внимание на то, что вычилстительная сложность не зависит от количества сущностей и типов связи.

3.2.1. Детали реализации

Весь код реализован на языке Python на основе фреймворка PyTorch. В качестве оптимизатора выбран Adam[3]. Гиперпараметры сети выбирались

автоматически. Размеры эмбеддингов выбирались из $7, 10, 13$. Количество эпох составило $4000 - 4500$ в среднем. Ссылка на исходный код.

4. Эмпирическое сравнение энергий

Эта глава описывает пути экспериментального сравнения **SME** современных методов для изучения представлений мультиреляционных данных.

4.1. Набор данных

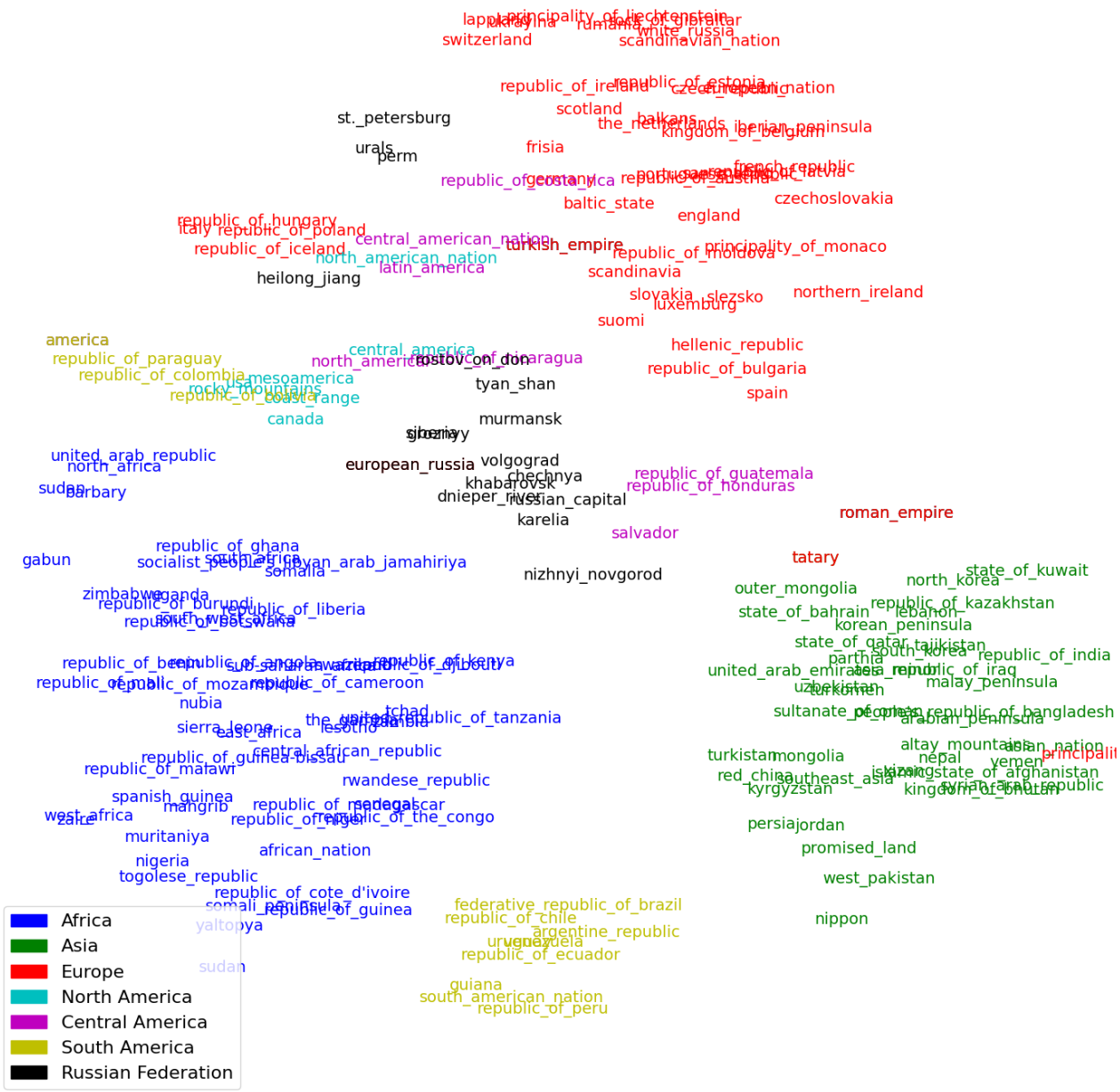

Рис. 4.1. Представление вложений SME_{lin} после 600 эпох обучения.

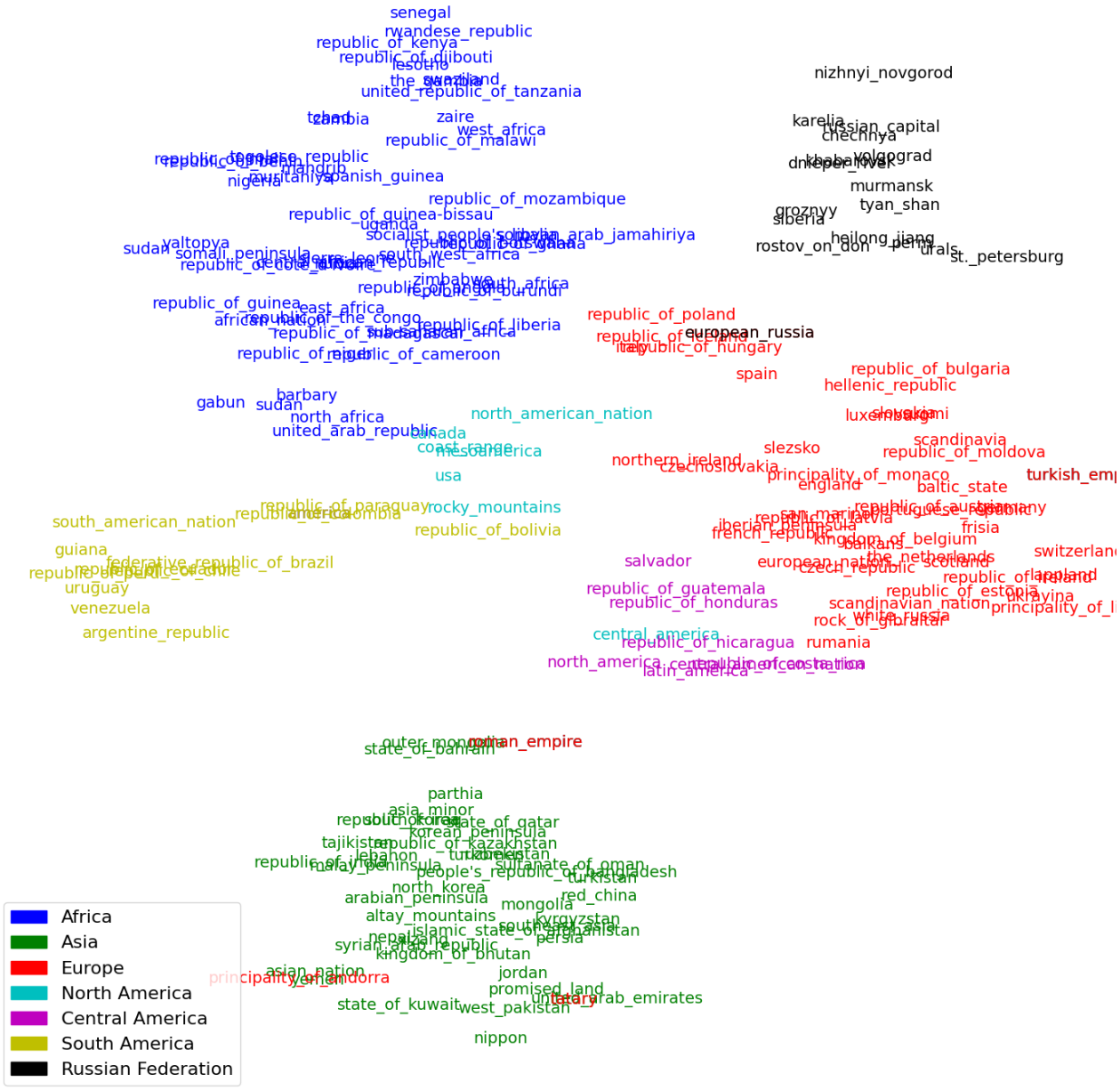

Рис. 4.2. Представление вложений SME_{lin} после 1000 эпох обучения.

Для сравнения рассмотренных функций энергии **SME** будет использоваться тезаурус WordNet, созданный для создания интуитивно понятного словаря и тезауруса; и поддержки автоматического анализа текста, но его можно также рассматривать как граф знаний. Для преобразования тезаурцса в граф знаний необходимо в качестве вершин использовать слова (смыслы), а в качестве ребер использовать семантические связи смыслов слов. В рамках работы будут рассмотрены все сущности, которые были связаны с типами отношений, приведенными в таблице 4.1, также удалены сущности, которые встречаются

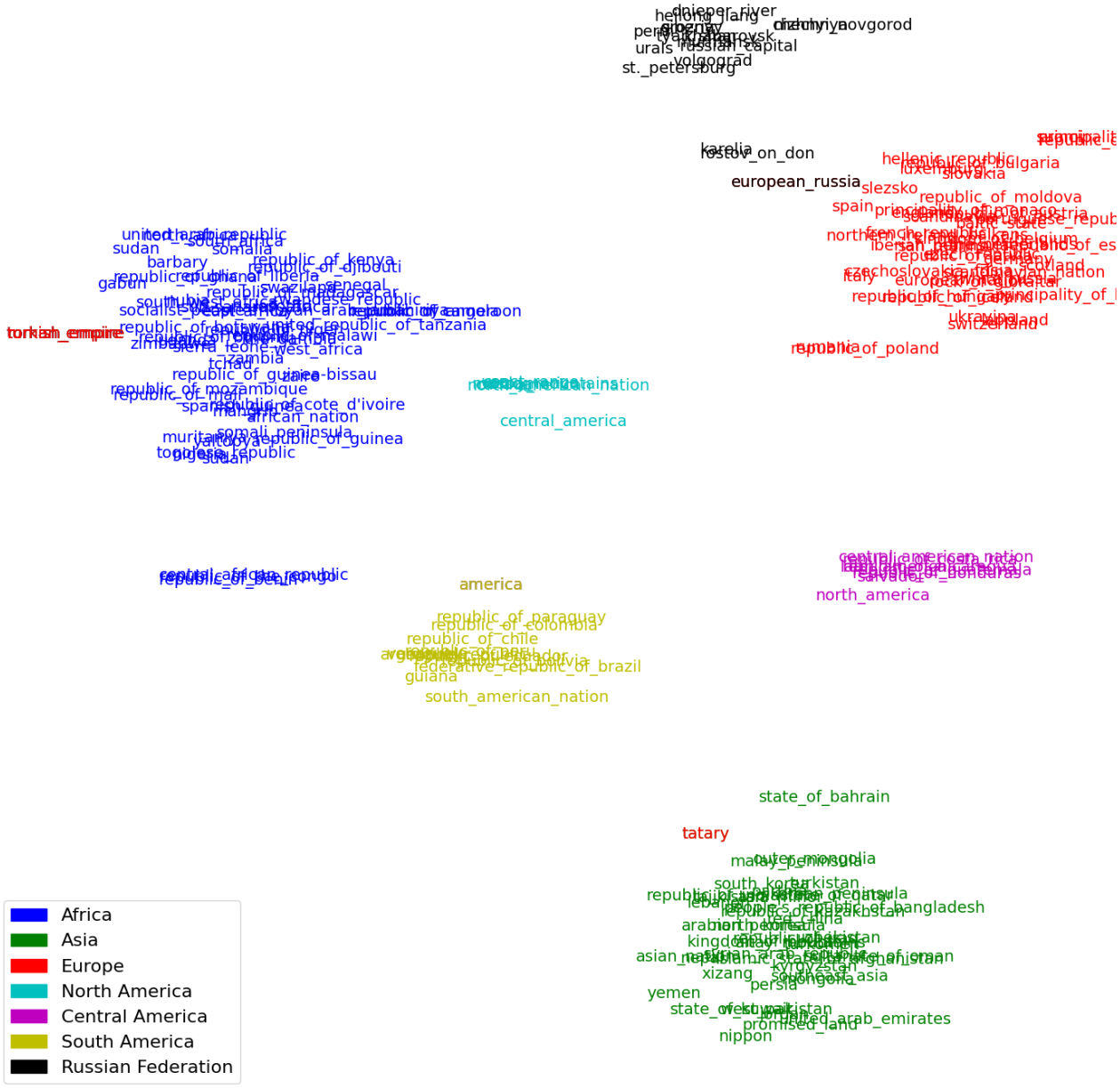

Рис. 4.3. Представление вложений SME_{lin} после 4500 эпох обучения.

в триплетах более 15 раз. Таким образом, мы получили граф знаний с 40943 сущностями и 18 типами связи. Примерами триплетов из этой базы знаний являются (_score_NN_1, _hypernym, _evaluation_NN_1) и (_score_NN_2, _has_part, _musical_notatio_NN_1). Посколько в базе знаний слова могут иметь разный смысл в зависимости от контеста (например омонимы), то все сущности преобразованы с помощью POS-тэгирования (Part-of-speech тэгирования, добавления суффикса, обозначающего часть речи, например NN, как noun, для существительных, VB, как verb, для глаголов, JJ, как adjactive,

для прилагательных и RB, как adverb, для наречий) и цифры, лля обозначения номера смысла (NN_1 - первый смысл существительного). Эта версия Wordnet отличалась от Wordnet, которая использовалась в исследовании[4] тем, что включает в себя намного меньше сущностей, но использует больше типов связи.

_hypernym, _hyponym, _instance_hyponym, _instance_hypernym,
_related_form, _has_part, _part_of, _member_has_part,
_also_see, _attribute, _synset_domain_region,
_synset_domain_topic, _verb_group, _member_of_domain_region,
_member_of_domain_usage, _member_of_domain_topic,
_member_part_of, _synset_domain_usage

Таблица 4.1. Используемые типы связи WordNet.

4.2. Пространственные вложения сущностей

Для возможности наглядно выявлять семантические взаимосвязи сущностей можно использовать несколько подходов и метрик.

4.2.1. Методы проверки результатов

Самый очевидный - использовать метрику AUC (area under curve) или ее модификацию PR AUC (area under the precision-recall curve). В этом случае необходимо построить две ломаные - первая будет отображать желаемые энергии, а вторая - полученные; Искомая величина соответствует площади между двумя кривыми: чем меньше эта площадь - тем лучше получились вложения. Но для успешного оценивания с использованием данного метода требуется заранее знать желаемый результат. Соответсвенно необходимо заранее знать какие негативные триплеты будут использоваться в процессе обучения.

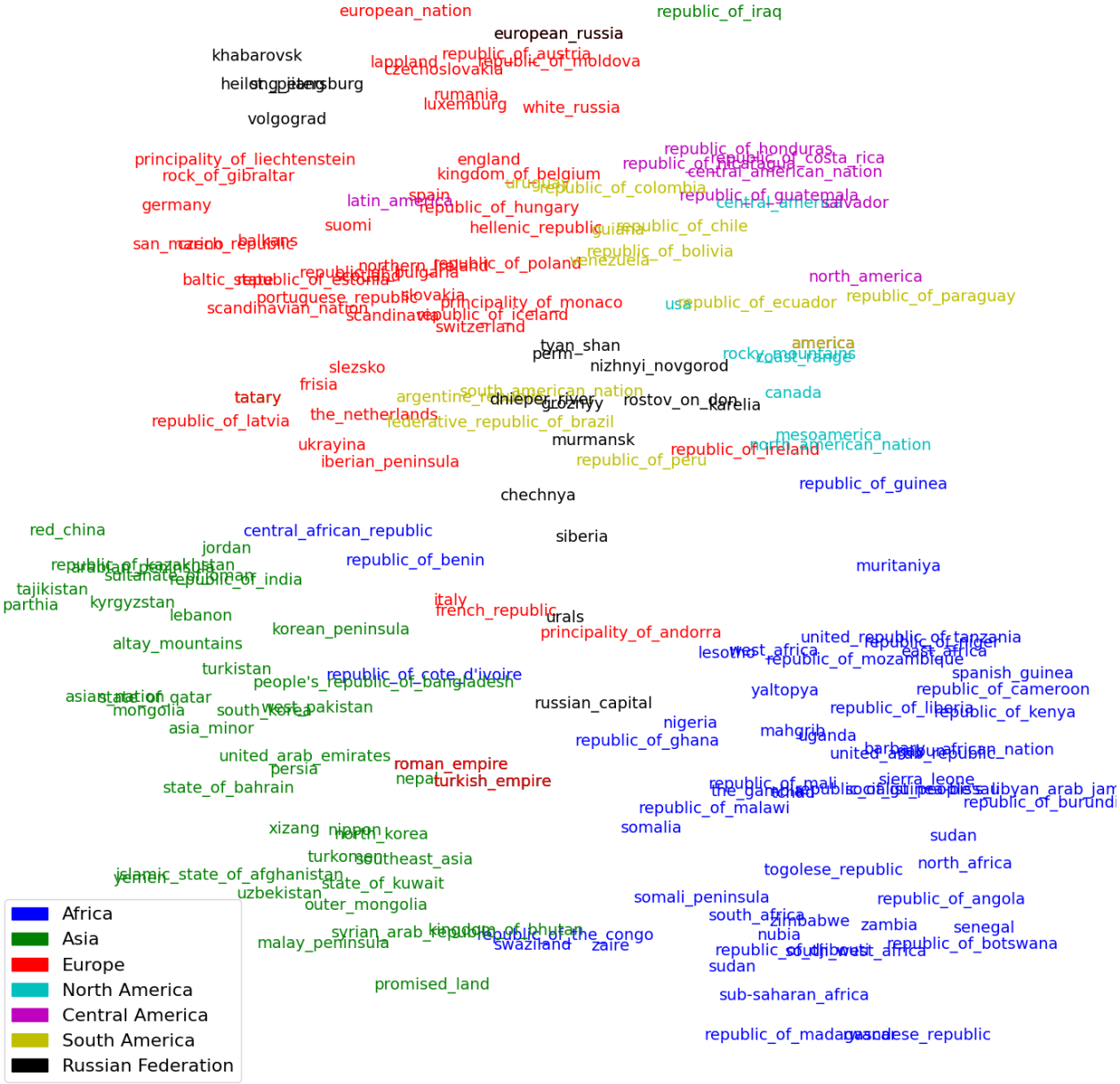

Рис. 4.4. Представление вложений SME_{bil} после 600 эпох обучения.

К сожалению, рассмотренный процесс обучения предполагает случайную генерацию негативных триплетов для каждого нормального триплета, что не позволяет нам проверить качество полученных пространственных вложений. В случае с графом знаний WordNet будет использоваться метод из статьи[2], описанный далее и предполагающий анализ только на нормальных триплетах. Мы измеряем средние прогнозируемые энергии с помощью следующей процедуры: для каждого тестового триплета левая сущность удаляется и заменяется каждой из сущностей словаря по очереди, энергии этих нега-

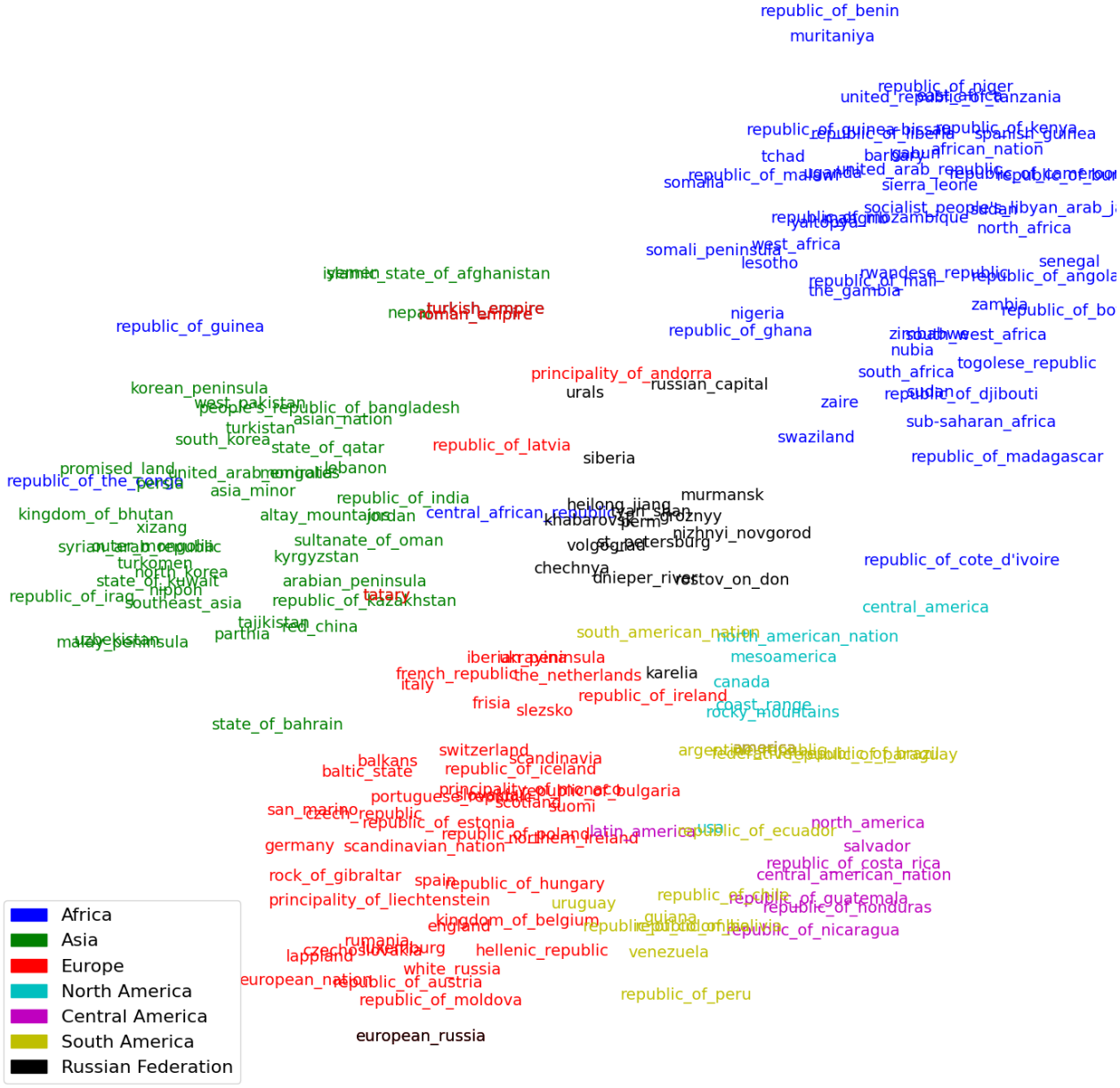

Рис. 4.5. Представление вложений SME_{bil} после 1000 эпох обучения.

тивных триплетов вычисляются моделью, а энергия правильного триплета сохраняется. Вся эта процедура также повторяется при удалении правого аргумента. Вычисляется средняя негативная энергия и сравнивается с нормальной энергией. Следовательно, чем меньше оказывается разница, тем лучше результат.

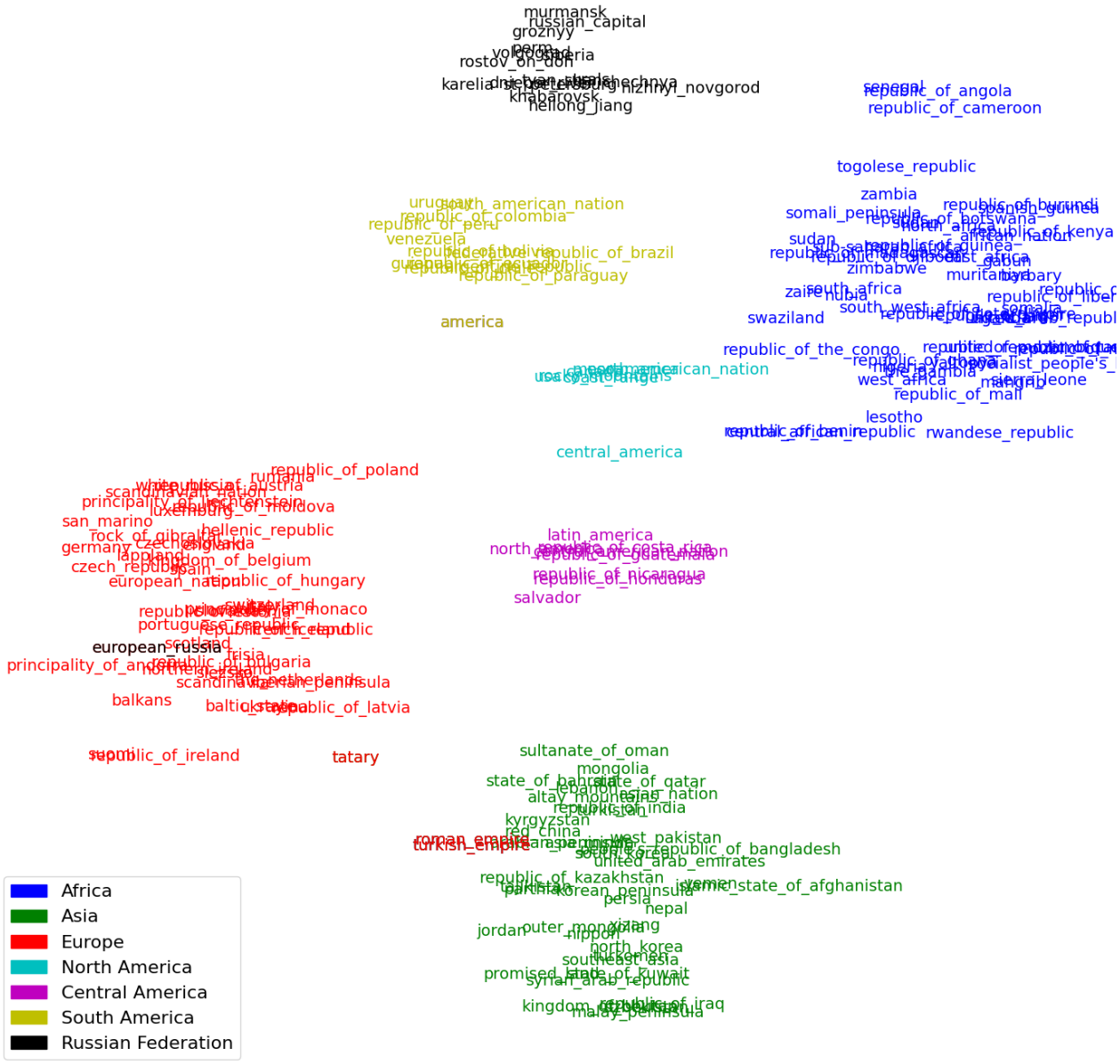

Рис. 4.6. Представление вложений SME_{bil} после 4500 эпох обучения.

4.2.2. Графическая интерпретация

Матрица пространственных вложений E факторизует информацию из всех отношений, в которых появляется сущность. В этом разделе представлены семантики рассматриваемых представлений. Для этого из графа знаний WordNet были выбраны 120 сущностей, которые относятся к странам всего мира. Было выбрано именно такое подмножество, так как оно имеет очевидную и понятную семантическую структуру. После этого, мы отображаем

полученные в ходе обучения $\mathbf{SME}_{lin}$ и $\mathbf{SME}_{bil}$ эмбеддинги в двумерное пространство с помощью алгоритма t-SNE. Далее на рис. 4.1, 4.2, 4.3, 4.4, 4.5 и 4.6 представлены результаты сравнений: различные цвета использованы для различных континентов; суффиксы POS-тэгирования и номера смысла слова удалены для ясности.

Для построения представлений были использованы триплеты (lhs, rel, rhs), где $rel \in \{\_part\_of, \_has\_part\}$ и $lhs, rhs \in \{$Africa, Asia, Europe, North America, Central America, South America, Russian Federation, Orient, Eurasia, West$\}$. Для более плотного расположения, помимо стран также были включены в рассмотрение сущности orient (the hemisphere that includes Eurasia and Africa and Australia), eurasia (the land mass formed by the continents of Europe and Asia) и west (the countries of (originally) Europe and (now including) North America and South America), без дополнительных сущностей эмбеддинги располагались достаточно далеко друг от друга.

5. Заключение

В ходе проделанной работы была исследована проблема построения пространственных вложений графов знаний. Основное внимание было уделено исследованию существующей математической модели и реализации предложенного подхода.

Были реализованы две модели, основанные на функции энергии: $\mathbf{SME}_{lin}$ и $\mathbf{SME}_{bil}$. Полученные представления пространственных вложений, построенные с помощью линейной функции энергии $\mathbf{SME}_{lin}$ (рис. 4.1, 4.2, 4.3) хорошо отражают географическую семантику, собираясь в плотные группы по континентам. Но построенные таким образом эмбеддинги сгущаются основываясь только на континенте, не беря в учет другие единичные признаки. Также некотоыре сущности попали не в свои группы из за небольшого количества в тренировочных данных и случайного процесса выбора негативных триплетов. Такое поведение получается из за большого влияния попарных комбинаций в (2.2). В случае $\mathbf{SME}_{bil}$ (рис. 4.4, 4.5, 4.6), получившиеся эмбеддинги учитывают все возможные знания, располагая вложения не только по принадлежности континентам, но и учитывая добавленные структурыне особенности (Например, Европа и Азия находятся рядом). Также во время экспериментов было замечено, что при использовании билинейной функции энергии все сущности находятся в своих смысловых группах, что очевидно, обеспечивается сильными связями в (2.4).

Рассморенные методы не являются самыми качественными и быстрыми. Выбор метода зависит от решаемой задачи. Например, в задаче нахождения сущности, которая логически подходит связному набору слов переменного размера, лучше использовать метод SMORE[5].

Список литературы

1. *Bordes A.*, *Glorot X.*, *Weston J.*, *Bengio Y.* A semantic matching energy function for learning with multi-relational data // Machine Learning. — 2014. — Т. 94, № 2. — С. 233—259.
2. *Glorot X.*, *Bordes A.*, *Weston J.*, *Bengio Y.* A semantic matching energy function for learning with multi-relational data // arXiv preprint arXiv:1301.3485. — 2013.
3. *Kingma D. P.*, *Ba J.* Adam: A method for stochastic optimization // arXiv preprint arXiv:1412.6980. — 2014.
4. *Bordes A.*, *Weston J.*, *Collobert R.*, *Bengio Y.* Learning structured embeddings of knowledge bases // Twenty-fifth AAAI conference on artificial intelligence. — 2011.
5. *Ren H.*, *Dai H.*, *Dai B.*, *Chen X.*, *Zhou D.*, *Leskovec J.*, *Schuurmans D.* SMORE: Knowledge Graph Completion and Multi-hop Reasoning in Massive Knowledge Graphs // arXiv preprint arXiv:2110.14890. — 2021.