ИССЛЕДОВАНИЕ МЕТОДОВ АВТОМАТИЧЕСКОГО ПОПОЛНЕНИЯ ТЕЗАУРУСА НА ОСНОВЕ WORD2VEC

Сорокина С. А. (sophie.sorokina@yandex.ru)
МГТУ им. Н.Э. Баумана, Москва, Россия

Ключевые слова: извлечение отношений, тезаурус, РуТез, векторные представления слов, word2vec

ANALYSIS OF AUTOMATIC THESAURUS CONSTRUCTION METHODS BASED ON WORD2VEC

Sorokina S. A. (sophie.sorokina@yandex.ru)
BMSTU, Moscow, Russia

Abstract

In this paper we propose different approaches for hyponym-hyperonym relation extraction from free text and embedding them to RuThes thesaurus' hierarchy. We investigate the applicability of word embeddings for example word2vec technology to such a task. An attempt is made to improve the quality of solution using extracted from the thesaurus relationships between concepts. Quality assessment is made by comparing with experts' answers using mean reciprocal rank.

Key words: relationship extraction, thesaurus, RuThes, word embedding, word2vec

Введение

Тезаурус – это иерархическая сеть понятий, связанных между собой некоторыми отношениями. В последнее время интерес к ним возрос, поскольку тезаурусы являются эффективным инструментом для улучшения качества таких семантически сложных задач, как, например, разрешение лексической многозначности [1] или поиск ответа на вопрос [2]. Ручное пополнение тезауруса – очень длительный, трудоёмкий и дорогостоящий процесс, поэтому его автоматизация является актуальной на сегодняшний день задачей.

Цель данной работы – создание системы для автоматического извлечения из текста родовидовых отношений между словами и последующего встраивания их в структуру тезауруса. Подобная задача рассматривалась в рамках соревнования SemEval-2015, при этом только 2 команды из 6, принявших участие, использовали обучение без учителя [3]. При извлечении отношений одна опиралась на лексико-синтаксические шаблоны, другая – строила бинарное дерево понятий, используя структуру Википедии и WordNet [4]. Результаты первой оказались лучше, но использование лексико-синтаксических шаблонов имеет свои минусы: сложность создания шаблонов и переноса на другой язык, а также зависимость от корпуса [5].

Вставка новых слов в тезаурус подразумевает в качестве основной процедуры нахождение сходства между словами. В роли средства для измерения сходства между значениями слов используется технология word2vec. Данная технология позволяет измерять семантическую близость благодаря представлению слов в виде контекстных векторов [6]. Принцип работы word2vec основан на дистрибутивной гипотезе и заключается в том, что словам, встречающимся в схожих контекстах, присваиваются схожие векторы, для измерения сходства между векторами используется косинусная мера [7].

1. Задачи и данные

Тезаурус РуТез представляет собой иерархическую систему абстрактных понятий, связанных между собой четырьмя различными типами отношений [8]. В данной работе рассматривается только один тип: родовидовое отношение.

С каждым понятием сопоставлен набор слов (так называемых текстовых входов), которые могут являться конкретным представлением данного понятия в тексте (например, с

понятием «банк [финансовое учреждение]» сопоставлены такие текстовые входы, как «банк», «расчетно-депозитарная организация», «банковский» и др.). В свою очередь, вследствие полисемии некоторые текстовые входы могут соответствовать различным понятиям (например, текстовый вход «банк» соответствует трем понятиям: «банк [финансовое учреждение]», «банк [собрание предметов, объектов сведений]» и «банк в карточной игре»). Таким образом, в тезаурусе хранятся два типа объектов (понятия и текстовые входы), связанные друг с другом отношением ассоциации вида «многие ко многим». Иерархическими же отношениями связаны между собой только сами понятия.

В задаче используются две версии тезауруса RuTes. Версия RuThes-lite 2.0 содержит новые слова и выражения по сравнению с версией RuThes-lite 1.0, что позволяет использовать RuThes-lite 2.0 как эталон при оценке качества реализованного решения.

Для обучения модели word2vec необходимо достаточно много данных. В качестве текстовой коллекции для этой цели используется новостной корпус, содержащий 2 млн. документов

2. Подготовка данных

Для того, чтобы обучить модель, была проведена предварительная обработка исходных текстов, состоящая из следующих этапов:

- сегментация слов на предложения (с учётом сокращений из списка, однобуквенных сокращений, названий сайтов и чисел);
- удаление стоп-слов;
- лемматизация.

После этого на всей коллекции была обучена модель word2vec с помощью библиотеки genism для Python. В качестве минимального порога частоты, по достижении которого слово включалось в модель, выставлялось значение 10. При этом в коллекции было выявлено 341 млн. слов, 1 140 тыс. из которых различны.

Слова – кандидаты для расширения RuThes-lite 1.0 (30 шт.) – были выбраны из подмножества RuThes-lite 2.0, не пересекающегося с RuThes-lite 1.0, по принципу максимальной частотности в корпусе.

3. Предложенный подход

Общая идея заключается в том, чтобы для каждого слова – кандидата на вставку в тезаурус – найти множество потенциально близких к нему понятий. Затем для каждого понятия вычислить схожесть с кандидатом и отсортировать по убыванию этого значения. В роли различных мер сходства между словом и понятием выступают модификации значения схожести, выданного моделью word2vec.

3.1 Создание множества потенциально близких понятий

Алгоритм генерации для кандидата на вставку в тезаурус множества потенциально близких понятий включает в себя следующие шаги:

Шаг 1. Для каждого выбранного слова с помощью обученной модели word2vec выбираются 10 максимально близких текстовых входов из RuThes-lite 1.0.

Это дает нам слова, наиболее похожие по смыслу на исходное с точки зрения word2vec. Например, наибольшие значения сходства со словом «втб» получили: «сбербанк», «госбанк», «госучастие» и «банка». Последнее слово демонстрирует проблему морфологической многозначности: на самом деле в текстах употреблялось слово «банк», но оно было неверно лемматизировано на этапе морфологического анализа.

Также следует отметить, что не для всех слов среди ближайших по word2vec можно найти некоторое обобщение, которое может служить гиперонимом. Например, наиболее близкими к слову «мчс» оказались «гидрометеослужба», «гидрометслужба», «угибдд», «уфсб» и «мвд».

Шаг 2. Для каждого текстового входа, полученного на предыдущем шаге, из RuThes-lite 1.0 выбираются все понятия, связанные с данным текстовым входом (то есть те понятия,

которые может представлять данный текстовый вход).

Таким образом, мы получаем некое начальное множество близких к исходному слову понятий. В качестве примера понятия, связанные с ближайшими к «втб» текстовыми входами, представлены в таблице 1.

Текстовый вход	Связанные с ним понятия
Сбербанк	1. Сберегательный банк
Госбанк	1. Государственный банк
Госучастие	1. Государственное участие
Банка	1. Банка (сосуд) 2. Медицинская банка

Таблица 1. Список понятий, связанных с 4-мя ближайшими к «втб» текстовыми входами.

Неполнота тезауруса влечет за собой следующую проблему: в тезаурусе может в принципе не быть указано понятий для того значения слова, в котором оно используется в тексте. Поясним на примере. Ближайшими текстовыми входами для «цска» оказались другие спортивные клубы: «зенит», «рубин», «терек» и пр. – но ни один из этих текстовых входов не связан в RuThes-lite 1.0 с понятием «спортивный клуб» (с которым «цска» связан в RuThes-lite 2.0), поэтому переход к понятиям для этих текстовых входов априори даст совершенно не схожие по смыслу понятия, такие как: «вершина, кульминация», «рубин (камень)», «терек (река)».

Также на данном этапе проявляется проблема лексической многозначности текстовых входов: для каждого – множество понятий расширяется всеми его значениями, как нужными, так и совсем посторонними, что засоряет множество понятий. Так, для близкого к «мчс» текстового входа «управление» в множество понятий подтягиваются «синтаксическое управление», «управление (крупное административное учреждение)», «управление устройством», «управлять, руководить».

Шаг 3. Для каждого элемента начального множества понятий с помощью родовидового отношения ищется гипероним.

Шаг 4. Процедура повторяется для каждого нового множества, полученного на предыдущем шаге.

Последние два шага рассчитаны на то, чтобы получить некоторые понятия-обобщения. Для тех слов, для которых с помощью word2vec не было найдено обобщающих текстовых входов, эта процедура помогает найти гипероним, с которым можно связать кандидата на вставку в тезаурус. Например, понятие «министерство», с которым связано «мчс» в RuThes-lite 2.0, было найдено только благодаря шагу 4:

- на шаге 1 среди максимально близких текстовых входов оказался «мвд»;
- на шаге 2 в начальное множество понятий было включено связанное с «мвд» понятие «министерство внутренних дел»;
- на шаге 3 был найден его гипероним «силовое министерство»;
- на шаге 4 одним из гиперонимов «силового министерства» оказалось «министерство».

Шаг 5. Объединение всех множеств понятий, найденных на предыдущих шагах, формирует множество потенциально близких понятий. Для каждого понятия из этого результирующего множества находим соответствующие ему текстовые входы из RuThes-lite 1.0 (как слова, так и словосочетания).

Далее нам потребуется определять схожесть понятий с кандидатом на вставку. Так как реализацией понятия в тексте являются соответствующие ему в тезаурусе текстовые входы, мы будем мерить близость между кандидатом и текстовыми входами, а потом различными способами аппроксимировать сходство с понятием полученными значениями, поэтому нам требуется данный шаг. В результате для «мчс», например, получилось 45 потенциально близких понятий и 145 текстовых входов суммарно.

Ключевым моментом является то, что учитываются не только однословные текстовые входы, но и состоящие из нескольких слов. В противном случае качество решения сильно бы ухудшилось – многие кандидаты на вставку в RuThes-lite 1.0 связаны в эталонном RuThes-lite 2.0 именно со словосочетаниями. В word2vec словосочетание можно представить списком слов, и для подсчета схожести такого списка с кандидатом векторные представления отдельных слов складываются и усредняются, а затем ищется скалярное произведение между этими средними.

3.2 Вычисление схожести слова и понятия

Понятие тезауруса – это некая абстрактная сущность, конкретной реализацией которой в тексте являются соответствующие текстовые входы. Поэтому сходство между понятием и кандидатом на вставку в тезаурус оценивается через сходство между текстовыми входами и кандидатом. При этом учитываются текстовые входы не только самого понятия, но и его «ближайших» понятий-соседей («ближайшим» считается понятие, непосредственно связанное с данным родовидовым отношением, т. е. гипоним или гипероним данного понятия).

Пусть $s(t,\ w)$ – это некоторое значение схожести текстового входа t с кандидатом на вставку в тезаурус w. В качестве $s(t,\ w)$ будем использовать три различные меры.

Первая – это значение схожести по word2vec.

В основе второй лежит гипотеза о том, что в следующем предложении мы часто ссылаемся на предыдущее, а для этого могут использоваться синонимы и родовидовые отношения, извлечение которых и является целью данной работы. Чтобы учесть это, используется метрика – нормализованная точечная информация (англ. NPMI, Normalized Pointwise Mutual Information):

$$NPMI(x,y) = -\left(ln\frac{p(x,y)}{p(x)p(y)}\right)/\ln p(x,y),$$

где в данном случае $p(a)$ – это число вхождений a в текстах, делённое на N – общее число слов в корпусе, а $p(x,y)$ – это число появлений x и y в соседних предложениях, делённое на N [9]. При этом x и y могут быть как словами, так и словосочетаниями.

В качестве третьей предлагается сумма первых двух мер.

Таким образом, предлагаются три различных модификации нижеприведённого алгоритма, использующих в качестве меры сходства слов $s(t,\ w)$:

- $W2V(t,\ w)$ – значение схожести t с w по word2vec;
- $NPMI(t,w)$;
- $W2V(t,\ w) + NPMI(t,w)$.

Алгоритм вычисления сходства понятия c и слова w с учётом k дополнительных текстовых входов:

Шаг 1. Для понятия c по текстовым входам t из множества T_c всех соответствующих ему текстовых входов вычисляется максимальное значение схожести с кандидатом:

$$s_c = \max_{t\ \in T_c} s(t,\ w).$$

Шаг 2. Создается множество понятий P, включающее в себя само понятие c и множество всех его гипонимов и гиперонимов.

Шаг 3. Создаётся множество S, включающее в себя все (кроме s_c) значения схожести кандидатов с текстовыми входами, соответствующими понятиям из P:

$$S = \left(\bigcup_{p\ \in P}\bigcup_{t\ \in T_p} s(t,w)\right) \setminus s_c,$$

где T_p – это множество соответствующих понятию p текстовых входов. При этом отрицательные значения s отбрасываются, то есть $s(t,w) > 0$.

Шаг 4. Элементы $s_i \in S$ сортируются в порядке убывания и им присваиваются соответствующие индексы: s_1 – это элемент с максимальным значением схожести и т. д.

Шаг 5. Итоговое значение сходства понятия с кандидатом вычисляется по формуле:

$$s_{result} = s_c + \sum_{i=1}^{k} s_i.$$

Таким образом, максимальное значение схожести текстового входа с кандидатом (s_c) для самого понятия будет учтено всегда. Дополнительные же значения $s(t, w)$ могут принадлежать как самому понятию, так и его соседям, и их число определяется константой k.

Конечным этапом алгоритма является сортировка множества потенциально близких понятий по убыванию значения схожести понятия и кандидата на вставку в тезаурус. Соответственно, для каждой из предложенных мер сходства данная процедура осуществляется отдельно.

4. Результаты

На выходе предложенного алгоритма – список понятий, который отсортирован по убыванию значения сходства понятия с кандидатом на вставку в тезаурус. Поэтому для оценки качества метода, используется учитывающая ранжирование метрика MRR (англ. Mean Reciprocal Rank) [10]:

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i},$$

где Q – это множество правильных ответов, то есть понятий, с которым кандидат связан в RuThes-lite 2.0, а $rank_i$ – это порядковый номер i-го ответа в списке понятий, полученном на выходе алгоритма.

Зависимость MRR от числа k дополнительно учитываемых текстовых входов для различных мер схожести: $NPMI(t,w)$, $W2V(t,w)$ и $W2V(t,w) + NPMI(t,w)$ – представлена в таблице 2. Максимальное значение MRR было достигнуто для меры $W2V(t,w) + NPMI(t,w)$ и $k = 2$.

	k				
MRR	0	1	2	3	4
MRR для $NPMI(t,w)$	0,286	0,374	0,37	0,334	0,322
MRR для $W2V(t,w)$	0,34	0,4	0,464	0,427	0,4
MRR для $W2V(t,w) + NPMI(t,w)$	0,307	0,45	**0,475**	0,398	0,374

Таблица 2. Зависимость MRR от k для мер схожести: NPMI, W2V, W2V+NPMI.

Заключение

В данной статье был описан подход, позволяющий автоматически извлекать из текстовой коллекции родовидовые отношения, что по сути является решением задачи автоматического пополнения тезауруса, если рассматривать отношения только данного типа.

Смысл подхода заключается в том, что для каждого кандидата на вставку в тезаурус находим:

– 10 максимально схожих по word2vec слов (текстовых входов);

– связанные в RuThes-lite 1.0 с этими текстовыми входами понятия, а также «родителей» этих понятий, находящихся на уровень и на два уровня выше в иерархии тезауруса.

Затем для каждого из полученных понятий:

– находим соответствующие ему текстовые входы тезауруса;

– с помощью трёх различных мер схожести слов (сходство по word2vec, значение $NPMI$ и их сумма) оцениваем сходство понятия и кандидата как максимум по текстовым входам меры схожести с кандидатом на вставку в тезаурус;

– к полученной величине добавляем k максимальных значений меры схожести текстовых

входов, соответствующих как данному, так и соседним понятиям (гипонимам/гиперонимам). После ранжируем понятия по оценке схожести понятия с кандидатом на вставку в тезаурус, и чтобы оценить качество метода, используем метрику MRR.

Было показано, что мера схожести $W2V(t,w)$ работает лучше, чем $NPMI(t,w)$ для любых значений параметра k. Мера $W2V(t,w) + NPMI(t,w)$ способна как улучшить, так и ухудшить результаты $W2V(t,w)$. Наилучшее качество достигается при использовании меры $W2V(t,w) + NPMI(t,w)$ и числе дополнительно учитываемых текстовых входов равном двум.

В дальнейшем планируется использовать хранящийся в тезаурусе большой объем данных с целью увеличения качества решения. В частности, выделить из этих данных признаки и обучить модель, с помощью методов машинного обучения.

Список литературы

1. Лукашевич Н.В., (2011), Тезаурусы в задачах информационного поиска, МГУ, Москва.
2. Harabagiu S., Maiorano S., Pasca M., (2003) Open-Domain Textual Question Answering Techniques. Natural Language Engineering 9 (3): 1-38.
3. Bordea G., Buitelaar P., Faralli S., Navigli R., (2015), SemEval-2015 Task 17: Taxonomy Extraction Evaluation (TExEval), in Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 902–910, Denver, Colorado.
4. Ceesay B., Hou W., (2015), NTNU: An Unsupervised Knowledge Approach for Taxonomy Extraction, in Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 938–943, Denver, Colorado.
5. Panchenko A., Adeykin S., Romanov A., Romanov P., Extraction of semantic relations between concepts with knn algorithms on Wikipedia, CDUD 2012--Concept Discovery in Unstructured Data, pages 78, 2012.
6. Mikolov T., Chen K., Corrado G., Dean J., (2013) Efficient estimation of word representations in vector space. CoRR, abs/1301.3781.
7. Sidorov G., Gelbukh A., Gómez-Adorno H., Pinto D., (2014), Soft Similarity and Soft Cosine Measure: Similarity of Features in Vector Space Model. Computación y Sistemas, Vol. 18, No. 3, pp. 491—504.
8. Добров Б.В., Лукашевич Н.В., (2009), Тезаурус РуТез как ресурс для решения задач информационного поиска, Труды Всероссийской Конференции Знания-Онтологии-Теории (ЗОНТ-09), Новосибирск, С. 250–259.
9. Bouma G., (2009), Normalized (Pointwise) Mutual Information in Collocation Extraction, in Proceedings of the Biennial GSCL Conference.
10. Voorhees E., (1999), Proceedings of the 8th Text Retrieval Conference, TREC-8 Question Answering Track Report, pages 77–82.