УДК 004.912 + 004.8
DOI 10.25205/1818-7900-2021-19-2-65-75

Автоматическое связывание терминов из научных текстов с сущностями базы знаний

А. А. Мезенцева [1], **Е. П. Бручес** [1, 2], **Т. В. Батура** [1, 2]

[1] *Новосибирский государственный университет*
Новосибирск, Россия

[2] *Институт систем информатики им. А. П. Ершова СО РАН*
Новосибирск, Россия

Аннотация

В настоящее время в связи с ростом научных публикаций все большую актуальность приобретают задачи, связанные с обработкой текстов научных статей. Такие тексты имеют особую структуру, лексическое и семантическое наполнение, что нужно учитывать при автоматическом анализе. Использование информации из баз знаний способно улучшить качество систем обработки текстов. Данная работа посвящена задаче связывания сущностей в текстах научных статей на русском языке, где в качестве сущностей выступают научные термины. Нами был размечен корпус научных текстов, где каждый термин связывался с сущностью из базы знаний. Также мы реализовали алгоритм связывания сущностей и протестировали его на полученном корпусе. Алгоритм состоит из двух этапов: генерация сущностей-кандидатов для входного термина и ранжирование полученного множества кандидатов. На этапе генерации список кандидатов формируется на основе построчного совпадения термина и сущности. Для ранжирования и выбора наиболее релевантной сущности для входного термина используется информация о количестве отношений сущности в базе знаний с другими сущностями, а также о количестве ссылок у сущности на другие базы знаний. Проведен анализ результатов и предложены возможные пути улучшения алгоритма, в частности использование информации о контексте термина и структуры графа знаний. Размеченный корпус выложен в открытый доступ и может быть полезен для других исследователей.

Ключевые слова

связывание сущностей, база знаний, научные термины, разметка данных

Благодарности

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 19-07-01134

Для цитирования

Мезенцева А. А., Бручес Е. П., Батура Т. В. Автоматическое связывание терминов из научных текстов с сущностями базы знаний // Вестник НГУ. Серия: Информационные технологии. 2021. Т. 19, № 2. С. 65–75. DOI 10.25205/1818-7900-2021-19-2-65-75

Automatic Linking of Terms from Scientific Texts with Knowledge Base Entities

A. A. Mezentseva [1], **E. P. Bruches** [1, 2], **T. V. Batura** [1, 2]

[1] *Novosibirsk State University*
Novosibirsk, Russian Federation

[2] *A. P. Ershov Institute of Informatics Systems SB RAS*
Novosibirsk, Russian Federation

Abstract

Due to the growth of the number of scientific publications, the tasks related to scientific article processing become more actual. Such texts have a special structure, lexical and semantic content that should be taken into account while

processing. Using information from knowledge bases can significantly improve the quality of text processing systems. This paper is dedicated to the entity linking task for scientific articles in Russian, where we consider scientific terms as entities. During our work, we annotated a corpus with scientific texts, where each term was linked with an entity from a knowledge base. Also, we implemented an algorithm for entity linking and evaluated it on the corpus. The algorithm consists of two stages: candidate generation for an input term and ranking this set of candidates to choose the best match. We used string matching of an input term and an entity in a knowledge base to generate a set of candidates. To rank the candidates and choose the most relevant entity for a term, information about the number of links to other entities within the knowledge base and to other sites is used. We analyzed the obtained results and proposed possible ways to improve the quality of the algorithm, for example, using information about the context and a knowledge base structure. The annotated corpus is publicly available and can be useful for other researchers.

Keywords

entity linking, knowledge base, scientific terms, data annotation

Acknowledgements

The study was funded by RFBR according to the research project no. 19-07-01134

For citation

Mezentseva A. A., Bruches E. P., Batura T. V. Automatic Linking of Terms from Scientific Texts with Knowledge Base Entities. *Vestnik NSU. Series: Information Technologies*, 2021, vol. 19, no. 2, p. 65–75. (in Russ.) DOI 10.25205/1818-7900-2021-19-2-65-75

Введение

Использование информации из баз знаний для решения различных прикладных задач в последнее время становится очень актуальным. Информация из базы знаний повышает качество автоматической системы, помогая разрешать лексическую неоднозначность слов и понятий, точнее определить их значение в текстах. Особую сложность представляет работа с информацией из узких предметных областей, когда подходящей терминологией владеют только специалисты. Вот почему для качественного автоматического извлечения информации важно, чтобы в системе присутствовал компонент связывания элементов текста с базой знаний.

Данная работа посвящена задаче связывания сущностей, где в качестве сущностей рассматриваются термины. Задача связывания сущностей (EL) состоит в определении упоминания сущности в неструктурированном тексте и установлении связи с сущностью в структурированной базе знаний [1].

Например, в зависимости от контекста слово «Владимир» может обозначать город в России или указывать на конкретного человека. При этом, в отличие от задачи разрешения омонимии типа сущности, в этой задаче также требуется найти конкретную сущность в базе знаний. Под базой знаний понимается база данных, содержащая структурированную информацию. Базу знаний также можно представить в виде графа знаний, в котором узлами являются сущности, а ребрами – отношения между ними. Такое представление знаний позволяет не только хранить имеющуюся информацию, но и на основе нее выводить новые факты. В качестве баз знаний традиционно выступают Википедия [1] и Викиданные [2]. Также из известных баз знаний стоит упомянуть DBPedia – база знаний, содержащая структурированную информацию, извлеченную из Википедии [2], Google Knowledge Graph [3] и Freebase (не поддерживается в настоящее время) [4]. Такие базы знаний используются в информационном поиске [5], при построении диалоговых систем [6], извлечении семантических отношений [7] и во многих других задачах.

В данной работе предложен алгоритм автоматического связывания сущностей в текстах научных статей на русском языке. В эксперименте осуществляется привязка к Викиданным, в качестве сущностей рассматриваются научные термины. Эксперимент проводился на размеченном нами корпусе, который является открытым и доступен по ссылке https://github.com/iis-research-team/ruserrc-dataset.

[1] https://wikipedia.org

[2] https://www.wikidata.org

Обзор существующих работ

Обзор методов

Традиционно задача связывания сущностей делится на 4 этапа.

Этап 1: распознавание именованных сущностей. Чаще всего этот этап выделяется в отдельную задачу, и уже выделенные сущности подаются на вход следующему этапу. Обзор методов распознавания сущностей приведен в статье [8].

Этап 2: генерация кандидатов. На этом шаге создается краткий список возможных сущностей (кандидатов) для выделенного термина. Обычно такой список создается на основании строкового совпадения (полного или частичного) упоминания в тексте с сущностями, а также применяют различные эвристики и методы для расширения этого списка (например, поиск по синонимам). Так, например, авторы статьи [9] для генерации множества кандидатов используют страницы разрешения неоднозначности и редиректов Википедии, которые в том или ином виде содержат омонимичные и синонимичные слова и фразы. Если для сущности не находится таких страниц, то используют *n*-граммы для нахождения кандидатов. Так как количество кандидатов может оказаться большим, то применяют ранжирование кандидатов: по расстоянию Джаро – Винклера [10] между сущностью и упоминанием и косинусному расстоянию между вектором контекста и вектором сущности. В финальное множество кандидатов попадают *k* первых кандидатов. В статье [11] описывается подход, который заключается в сопоставлении словоформ с заранее построенным индексом, а также применяются методы нормализации строки и меры схожести триграмм для генерации кандидатов, если ничего не было найдено по полному совпадению. Для уменьшения списка потенциальных кандидатов авторы используют априорную вероятность (на основании того, что некоторые сущности встречаются в текстах чаще, чем другие) и схожесть контекстов сущности и упоминания. Другие исследователи (например, [12]) прибегают к вычислению априорной вероятности совместной встречаемости сущности и упоминания в различных источниках: в Википедии (в заголовках страниц, в заголовках редиректов и в гиперссылках), в словаре, полученном на основе WebCorpus [3], и в словаре YAGO [4]. Максимальное значение получают те пары, которые встречаются в нескольких источниках.

Этап 3: ранжирование кандидатов. На этом шаге происходит оценка того, насколько хорошо объект-кандидат соответствует контексту. Здесь можно выделить три основных подхода. Первый подход основан на вычислении схожести контекстов, которые представляются в виде векторных представлений на основании как вручную сформированных признаков [13], так и полученных из языковых моделей [14]. При другом подходе задача ранжирования трансформируется в задачу бинарной классификации, в которой целью является определить, относится ли данное упоминание к сущности. В качестве классификатора могут использоваться наивный байесовский классификатор [15], SVM классификатор [16], глубокие нейронные сети [17]. В последнее время широкое распространение получили подходы, использующие векторные представления, полученные из графов знаний. Такая информация помогает понять, какое положение сущность занимает в графе, какими отношениями она связана с другими сущностями и др. Например, в статье [18] авторы строят векторные представления ребер графа, полученного из Dbpedia, с помощью алгоритма DeepWalk [19]. В работе [20] авторы используют алгоритм TransE [21] для векторизации сущностей в графе.

Этап 4: определение несвязанных упоминаний, для которых база знаний (KB) не содержит соответствующей сущности. Зачастую в системах этот этап отсутствует.

Больше подходов описано в работе [1], которая представляет собой обзор современных нейросетевых моделей для задачи связывания сущностей (EL). Также в данной статье отмечаются особенности данных для EL:

[3] http://www.webcorp.org.uk
[4] https://yago-knowledge.org

1) в обучающих наборах может отсутствовать даже один пример для конкретной сущности или упоминания. Для решения этой проблемы модели EL должны иметь возможности для обобщения;

2) базы знаний неполны, поэтому некоторые упоминания в тексте не могут быть правильно сопоставлены ни с одной записью в KB.

Разработано множество готовых решений, которые поддерживают английский язык и классический набор сущностей (например, OpenTapioca [22]). Библиотека DeepPavlov [5], в свою очередь, имеет предобученные модели для русского языка. Алгоритм состоит из следующих компонентов:

1) выделенная NER-моделью подстрока подается на вход в TfidfVectorizer, и получившийся разреженный вектор преобразуется в плотный;

2) Faiss-библиотека используется для нахождения *k* ближайших соседей для tf-idf векторов в матрице, где строки соответствуют tf-idf векторам слов в заголовках сущностей;

3) сущности ранжируются по числу отношений в Викиданных (количество исходящих ребер узлов в графе знаний);

4) BERT (English) или BERT (Russian) используется для ранжирования сущностей по описанию и по контексту, в котором упоминается сущность.

Обзор корпусов

Существуют корпуса для оценивания систем, решающих задачу связывания сущностей. Между собой они отличаются по нескольким важным аспектам:

1) база знаний, на которой они основаны (Википедия, WikiNews, Yago, DBpedia);

2) поддерживаемые языки (в большинстве случаев – английский, DBpedia Abstracts [23] – 7 языков);

3) источник текстов: твиты [24], новостные статьи [25]);

4) типы размеченных сущностей (большинство – классические PER, LOG, ORG; WikiMed [26] – сущности, связанные с медициной).

Для английского языка удалось найти два корпуса, в которых размечены научные термины: STEM-ECR [27] и SemEval 2015 Task 13 [28]. Насколько нам известно, в настоящее время для русского языка не существует подобного корпуса научных текстов с разметкой связанных сущностей.

Разметка корпуса

Описание разметки

Для русского языка существует корпус научных статей RuSERRC, в котором размечены термины и отношения между ними [29]. Мы дополнили этот корпус разметкой – связали выделенные термины с сущностями в Викиданных. Это свободная, совместно наполняемая, многоязычная, вторичная база данных, в которой собрана структурированная информация для обеспечения поддержки Википедии, Викисклада, а также других вики-проектов. Данная база знаний состоит из:

1) элементов, каждый из которых имеет уникальный идентификатор с префиксом Q и числовой частью, как, например, Дуглас Адамс (Q42).

2) утверждений, которые идентифицируются кодом, имеющим префикс P и числовую часть, например, учебное заведение (P69).

3) ссылки на сайты (Sitelinks) связывают каждый элемент с соответствующими ему статьями во всех клиентских вики, таких как Википедия, Викиучебник и Викицитатник.

Данный корпус содержит разметку не только терминов, но и вложенных в них сущностей, например: [*самосогласованное* [*электрическое поле*]]. При разметке связывания сущностей

[5] https://deeppavlov.ai

мы двигались от самой «большой» сущности к более «мелким» вложенным, т. е. если для самого первого уровня сущность была найдена в базе знаний, то вложенные сущности не размечаются.

Для поиска терминов в графе знаний мы допускали следующие видоизменения сущностей.

1. Все извлеченные сущности ищутся в базе знаний в нормализованной форме с учетом согласования и без учета регистра, например: *Линейных уравнений → линейное уравнение*.

2. Если из текста была извлечена сущность, подходящая по шаблону «общее понятие + название» (например, *язык программирования Python*, *операционная система Windows*), при этом в базе знаний находится только сущность с названием (например, *Python* (Q28865)), то такие две сущности связываются.

3. Если в тексте сущность написана с опечаткой, то в графе знаний мы ищем сущность без опечатки, например: *3Dреконструкцию → 3d реконструкция*.

4. Допускается поиск синонима сущности в базе знаний (проверяется запросом в поисковую систему или Википедию), например: *статистическая зависимость → корреляция*, *генетическая последовательность → нуклеотидная последовательность*, также допускается поиск перевода сущности, например, на английском языке.

5. Допускаются трансформации вида *архитектура системы → системная архитектура*.

6. Расшифрованные аббревиатуры, например: *wps → Wi-Fi Protected Setup*.

7. Если две и более сущности представлены как набор однородных членов с одним общим элементом, то каждый однородный член с общим элементом рассматривается как сущность, например: *спутниковая и мобильная связь → спутниковая связь*, *мобильная связь*.

8. Разного рода кореференции также связываются с одной сущностью, например: если в начале текста упоминается *метод k-means*, а затем в тексте *предложенный* [*метод*], то эти две сущности следует связать одним идентификатором.

9. Также мы считаем синонимами термины *подход* и *метод*.

Мы связывали термины только с сущностями в Википедии, эти сущности имеют идентификатор с префиксом «Q», в отличие от отношений, которые имеют идентификатор с префиксом «P». Также мы не связывали термины с сущностями, которые имеют тип «Научная статья».

Каждая сущность была размечена двумя асессорами. Мера согласованности была рассчитана как отношение количества сущностей без конфликта в разметке к общему количеству сущностей в корпусе и составила 82,33 %.

Наблюдения и выводы

Всего в корпусе выделено 3 386 терминов, 1 337 из которых удалось связать с сущностями в Викиданных. Средняя длина связанной сущности – 1,55 токена, минимальная длина – 1 токен, максимальная – 8 токенов.

Так как Викиданные является свободной и открытой базой знаний, то в ней встречаются ошибки, повторения сущностей и другие случаи, усложняющие работу с ней. Так, например, сущности *столбчатая диаграмма* и *гистограмма* являются в базе знаний двумя разными сущностями. Более того, есть дублирующиеся случаи, например *математический анализ* представлен сущностями Q7754 и Q149972, по описанию которых можно сделать вывод, что подразумевается одна и та же сущность. В таких случаях в разметке мы отдавали предпочтение той сущности, которая содержит больше информации, Этот показатель можно рассчитать по количеству отношений с другими сущностями, а также по наличию ссылок на эту же сущность в других базах знаний.

В Викиданных может встретиться искомая словоформа или фраза, но которая имеет смысл, отличный от того, что подразумевается в конкретном контексте. Такие сущности мы не связывали.

Вложенная сущность может иметь совершенно другое значение вне контекста: [*комплексный анализ*] [*поэтических текстов*] – в данном контексте *комплексный анализ* – это анализ текста на всех уровнях языка (об этом далее и идет речь в статье), но вне контекста термин *комплексный анализ* означает *раздел математического анализа, в котором рассматриваются и изучаются функции комплексного аргумента*.

Описание алгоритма

Нами был реализован алгоритм связывания сущностей и проверена его работа на корпусе с размеченными научными терминами. Поскольку нам не удалось найти подобных экспериментов для научных текстов на русском языке, то описанный алгоритм, скорее, является базовым и может служить отправной точкой для дальнейших исследований в данной области.

В качестве входных данных алгоритму подаются последовательность или единичный токен, соответствующий термину. Далее выполняются два основных шага: создание массива кандидатов для связывания, нахождение наиболее подходящей сущности в полученном множестве кандидатов.

Перед этапом генерации кандидатов входная строка проходит предварительную обработку – лемматизацию и приведение в нижний регистр. Здесь важно лемматизировать не отдельные слова, а сохранить согласование, например, из *обработке текстов* нужно получить *обработка текстов*. Для этого мы использовали библиотеку для анализа текстов на русском языке Natasha [6], которая позволяет приводить к нормальной форме не только отдельные словоформы, но и словосочетания, она также неплохо работает с русским языком и его лингвистическими особенностями. Стоит отметить, что данная библиотека приводит словоформу или фразу к начальной форме, сохраняя число, т. е. если грамматическая форма термина была во множественном числе, то оно сохранится, например: *мобильных приложений → мобильные приложения*. Также ошибка может возникнуть в результате омонимии, например: *у* [*приложения*] будет приведено к начальной форме *приложения*, так как на вход подается только сущность, без контекста.

На этапе генерации кандидатов входная строка сравнивается с названием сущности и ее синонимами. Если есть совпадение, то сущность добавляется в список кандидатов.

На этапе ранжирования кандидатов мы используем информацию о количестве ссылок у сущности на другие базы знаний и количестве отношений данной сущности с другими сущностями. Гипотеза состоит в том, что чем больше сущность наполнена информацией, тем более релевантной она является. Таким образом, выбор сущности для входного термина определяется по следующей формуле:

$$linked\_entity = max\ (f\ (ent_1),\ \ldots,\ f\ (ent_n)),$$

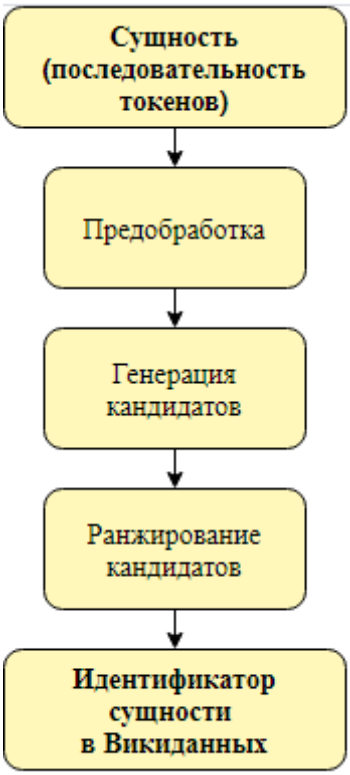
где

n – количество сущностей в полученном множестве кандидатов,

$f\ (ent_i) = numL_{ent} + numR_{ent}$, где $numL_{ent}$ – количество ссылок на другие базы знаний для данной сущности; $numR_{ent}$ – количество отношений данной сущности с другими сущностями в базе знаний.

Результатом работы алгоритма является идентификатор элемента Викиданных для входного упоминания. Общая схема работы алгоритма выглядит следующим образом:

[6] https://github.com/natasha/natasha



Следует отметить, что этот алгоритм не подразумевает использования информации из контекста, а также положения сущности в графе знаний (например, какие отношения она имеет). Добавление такой информации в алгоритм может существенно повысить качество. Кроме этого, качество алгоритма можно повысить за счет генерации синонимов и альтернативных написаний сущности для поиска кандидатов, что также пока не реализовано.

Тестирование

Алгоритм тестировался на размеченном нами корпусе (см. раздел Разметка корпуса). Для оценки качества нашего алгоритма мы использовали следующие метрики.

Точность – определяется как отношение количества верно связанных терминов ко всем терминам. Так как нам удалось связать не все термины в корпусе, информативнее будет разделить эту метрику на две: ***accuracy*** – принимает во внимание все сущности, и ***linked_accuracy*** – считается только на том наборе терминов, для которых нашлась сущность в графе знаний в корпусе. Таким образом, ***accuracy*** вычисляется по формуле

$$accuracy = n\_correct\_entities / all\_entities,$$

где

n_correct_entities – количество верно связанных терминов;

all_entities – количество всех терминов в корпусе.

Обозначим *n_all_linked_entities* количество всех терминов в корпусе, которые имеют связь с сущностью в Викиданных. Тогда ***linked_accuracy*** вычисляется по формуле

$$linked\_accuracy = n\_correct\_linked\_entities / n\_all\_linked\_entities,$$

где *n_correct_linked_entities* – количество верно связанных терминов среди всех связанных терминов.

Среднее количество кандидатов. Эта метрика показывает, насколько хорошо работает этап генерации кандидатов: если значение относительно мало, то можно улучшить алгоритм – например, также рассматривать синонимы, переводы, альтернативные написания сущностей и др. Если значение, наоборот, велико, то это может вызвать сложности при ранжировании кандидатов. Эту метрику мы также разбили на две: ***averaged_candidates*** – среднее количество кандидатов для всех сущностей, ***linked_averaged_candidates*** – среднее количество кандидатов для набора терминов, которые удалось связать.

$$averaged\_candidates = \frac{\sum_1^n |Candidates_i|}{n\_all\_entities},$$

где

$Candidates_i$ – множество полученных кандидатов для сущности;

n_all_entities – количество всех терминов в корпусе.

Обозначим *n_all_linked_entities* количество всех терминов в корпусе, которые имеют связь с сущностью в Викиданных, и $Linked\_candidates_i$ – множество сгенерированных кандидатов для всех терминов, связанных с Викиданными. Тогда формула для метрики ***linked_averaged_candidates*** имеет вид

$$linked\_averaged\_candidates = \frac{\sum_1^n |Linked\_candidates_i|}{n\_all\_linked\_entities}.$$

Наличие подходящего кандидата в списке, найденном алгоритмом (*top_candidates*). Высокое значение данной метрики показывает, что алгоритм генерации кандидатов работает с хорошей полнотой. При этом если данная метрика существенно выше точности, то это означает, что алгоритм ранжирования нуждается в доработках, так как нужная сущность имеется в списке кандидатов, но не получает наивысшего приоритета при ранжировании. Данная метрика считалась только для множества терминов в корпусе, которые имеют связь с сущностью из графа знаний, и вычислялась по формуле

$$top\_candidates = num\_correct\_sets \,/\, n\_all\_linked\_entities,$$

где *num_correct_sets* – количество множеств кандидатов для терминов, которые входят во множество *n_all_linked_entities*, содержащих верную сущность.

Полученные метрики для baseline-алгоритма:

Название метрики	Значение метрики
accuracy	0.66
linked_accuracy	0.38
averaged_candidates	1.06
linked_averaged_candidates	1.79
top_candidates	0.48

Довольно низкое значение метрики *linked_accuracy* показывает, что большая доля связанных терминов имеет форму, отличную от сущностей в базе знаний, Это означает, что этап генерации кандидатов требует доработок – нужно генерировать синонимы и другие возможные виды написания терминов. Значение метрики *top_candidates* выше значения метрики *linked_accuracy*. Это говорит о том, что алгоритм ранжирования не всегда работает корректно – здесь нужно учитывать не только наполненность информацией сущности, но и принимать во внимание контекст, в котором находится термин, чтобы сделать наиболее точный выбор. Все эти задачи мы планируем реализовать в ходе нашей дальнейшей работы.

Заключение

В ходе работы мы разметили корпус, состоящий из научных статей на русском языке, указав в нем для каждого выделенного термина его идентификатор из Викиданных. Также нам удалось реализовать алгоритм и получить метрики для данного корпуса. В дальнейшем мы планируем улучшать имеющийся алгоритм за счет использования иных подходов на каждом из этапов. При генерации важно учитывать контекст, чтобы корректно соотносить многозначные термины, а ранжирование осуществлять, используя векторные представления не только из моделей машинного обучения, но и с помощью графов.

Кроме того, мы столкнулись с тем, что базы знаний требуют обновления и дополнения, так как периодически появляются новые термины, а старые могут быть недостаточно или ошибочно заполнены. Наша система могла бы помочь в выделении из текстов необходимых сущностей и дополнении ими базы знаний в полуавтоматическом режиме.

Список литературы / References

1. **Sevgili O.**, **Shelmanov A.**, **Arkhipov M.**, **Panchenko A.**, **Biemann C.** Neural Entity Linking: A Survey of Models Based on Deep Learning. 2020. arXiv:2006.00575.
2. **Lehmann J.**, **Isele R.**, **Jakob M.**, **Jentzsch A.**, **Kontokostas D.**, **Pablo N. Mendesf**, **Hellmann S.**, **Morsey M.**, **Patrick van Kleef**, **Auer S.**, **Bizer C.** DBpedia – A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web Journal*, 2015, vol. 6, no. 2, p. 167–195. DOI 10.3233/SW-140134
3. **Dong X.**, **Gabrilovich E.**, **Heitz G.**, **Horn W.**, **Lao N.**, **Murphy K.**, **Strohmann T.**, **Sun S.**, **Zhang W.** Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In: Proceedings of SIGKDD, 2014, p. 601–610.
4. **Bollacker K.**, **Evans C.**, **Paritosh P.**, **Sturge T.**, **Taylor J.** Freebase: A collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data. Vancouver, British Columbia, Canada, 2008, p. 1247–1249. DOI 10.1145/1376616.1376746
5. **Otegi A.**, **Arregi X.**, **Ansa O.**, **Agirre E.** Using knowledge-based relatedness for information retrieval. *Knowledge and Information Systems*, 2015, vol. 44, p. 689–718. DOI 10.1007/s10115-014-0785-4
6. **Shalaby W.**, **Arantes A.**, **GonzalezDiaz T.**, **Gupta C.** Building chatbots from large scale domain-specific knowledge bases: challenges and opportunities. In: Proceedings of the International Conference on Prognostics and Health Management, 2019, p. 1–8.
7. **Tripodi I.**, **Boguslav M.**, **Hailu N.**, **Hunter L.** Knowledge-base-enriched relation extraction. In: Proceedings of the Sixth BioCreative Challenge Evaluation Workshop. Bethesda, MD USA, 2018, vol. 1, p. 163–166.
8. **Li J.**, **Sun A.**, **Han R.**, **Li C.** A Survey on Deep Learning for Named Entity Recognition. In: IEEE Transactions on Knowledge and Data Engineering, 2020, p. 1–20. DOI 10.1109/TKDE.2020.2981314.
9. **Fang Z.**, **Cao Y.**, **Li Q.**, **Zhang D.**, **Zhang Z.**, **Liu Y.** Joint entity linking with deep reinforcement learning. In: The World Wide Web Conference, WWW'19. New York, NY, USA, ACM, 2019, p. 438–447.
10. **Winkler W. E.** String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. In: Proceedings of the Section on Survey Research Methods. American Statistical Association, 2020, p. 354–359.
11. **Zwicklbauer S.**, **Seifert Ch.**, **Granitzer M.** Robust and collective entity disambiguation through semantic embeddings. In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2016, p. 425–434. DOI 10.1145/2911451.2911535

12. **Cao Y.**, **Hou L.**, **Li J.**, **Liu Z.** Neural collective entity linking. In: Proceedings of the 27th International Conference on Computational Linguistics. Santa Fe, New Mexico, USA, 2018, p. 675–686.
13. **Bunescu R. C.**, **Pasca M.** Using encyclopedic knowledge for named entity disambiguation. In: Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics, 2006, p. 9–16.
14. **Yin X.**, **Huang Y.**, **Zhou B.**, **Li A.**, **Lan L.**, **Jia Y.** Deep Entity Linking via Eliminating Semantic Ambiguity With BERT. *IEEE Access*, 2019, vol. 7, p. 169434–169445. DOI 10.1109/ACCESS.2019.2955498
15. **Varma V.**, **Pingali P.**, **Katragadda R.**, **Krishna S.**, **Ganesh S.**, **Sarvabhotla K.**, **Garapati H.**, **Gopisetty H.**, **Reddy V.B.**, **Reddy K.**, **Bysani P.** IIIT Hyderabad at TAC 2009. In: Proceedings of Text Analysis Conference, 2009, p. 102–114.
16. **Zhang W.**, **Su J.**, **Tan C. L.**, **Wang W. T.** Entity linking leveraging: Automatically generated annotation. In: Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), 2010, p. 1290–1298.
17. **Huang H.**, **Heck L.**, **Ji H.** Leveraging deep neural networks and knowledge graphs for entity disambiguation. 2015. arXiv:1504.07678.
18. **Parravicini A.**, **Patra R.**, **Bartolini D.**, **Santambrogio M.** Fast and Accurate Entity Linking via Graph Embedding. In: Proceedings of the 2nd Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA), 2019, p. 1–9. DOI 10.1145/3327964.3328499
19. **Perozzi B.**, **Al-Rfou R.**, **Skiena S.** DeepWalk: Online Learning of Social Representations. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, p. 701–710. DOI 10.1145/2623330.2623732
20. **Nedelchev R.**, **Chaudhuri D.**, **Lehmann J.**, **Fischer A.** End-to-End Entity Linking and Disambiguation leveraging Word and Knowledge Graph Embeddings. 2020. arXiv:2002.11143.
21. **Bordes A.**, **Usunier N.**, **Garcia-Duran A.**, **Weston J.**, **Yakhnenko O.** Translating Embeddings for Modeling Multi-relational Data. In: Proceedings of the 26th International Conference on Neural Information Processing Systems, 2013, vol. 2, p. 2787–2795.
22. **Delpeuch A.** OpenTapioca: Lightweight Entity Linking for Wikidata. 2019. arXiv:1904.09131.
23. **Brümmer M.**, **Dojchinovski M.**, **Hellmann S.** DBpedia Abstracts: A Large-Scale, Open, Multilingual NLP Training Corpus. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), 2016, p. 3339–3343.
24. **Noullet K.**, **Mix R.**, **Farber M.** KORE 50DYWC: An Evaluation Data Set for Entity Linking Based on DBpedia, YAGO, Wikidata, and Crunchbase. In: Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), 2020, p. 2389–2395.
25. **Minard A.**, **Speranza M.**, **Urizar R.**, **Altuna B.**, **Marieke van Erp**, **Schoen A.**, **Chantal van Son.** MEANTIME, the NewsReader Multilingual Event and Time Corpus. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), 2016, p. 4417–4422.
26. **Vashishth S.**, **Joshi R.**, **Dutt R.**, **Newman-Griffis D.**, **Rose C.** MedType: Improving Medical Entity Linking with Semantic Type Prediction. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2020, p. 229–240.
27. **D'Souza J.**, **Hoppe A.**, **Brack A.**, **Yaser Jaradeh M.**, **Auer S.**, **Ewerth R.** The STEM-ECR Dataset: Grounding Scientific Entity References in STEM Scholarly Content to Authoritative Encyclopedic and Lexicographic Sources. In: Proceedings of the 12th Language Resources and Evaluation Conference, 2020, p. 2192–2203.
28. **Moro A.**, **Navigli R.** SemEval-2015 Task 13: Multilingual All-Words Sense Disambiguation and Entity Linking. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), 2015, p. 288–297. DOI 10.18653/v1/S15-2049

29. **Bruches E.**, **Pauls A.**, **Batura T.**, **Isachenko V.** Entity Recognition and Relation Extraction from Scientific and Technical Texts in Russian. In: Proceedings of the Science and Artificial Intelligence Conference, 2020, p. 41–45. DOI 10.1109/S.A.I.ence50533.2020.9303196

Материал поступил в редколлегию
Received
01.03.2021

Сведения об авторах

Мезенцева Анастасия Алексеевна, студентка, Новосибирский государственный университет (Новосибирск, Россия)

anastasiamez@mail.ru

Бручес Елена Павловна, аспирантка, Институт систем информатики им. А. П. Ершова СО РАН (Новосибирск, Россия); ассистент, Новосибирский государственный университет (Новосибирск, Россия)

bruches@bk.ru

Батура Татьяна Викторовна, кандидат физико-математических наук, старший научный сотрудник, Институт систем информатики им. А. П. Ершова СО РАН (Новосибирск, Россия); доцент, Новосибирский государственный университет (Новосибирск, Россия)

tatiana.v.batura@gmail.com
ORCID 0000-0003-4333-7888

Information about the Authors

Anastasia A. Mezentseva, Student, Novosibirsk State University (Novosibirsk, Russian Federation)

anastasiamez@mail.ru

Elena P. Bruches, PhD Student, A. P. Ershov Institute of Informatics Systems SB RAS (Novosibirsk, Russian Federation); Assistant, Novosibirsk State University (Novosibirsk, Russian Federation)

bruches@bk.ru

Tatiana V. Batura, PhD in Physics and Mathematics, Senior Researcher, A. P. Ershov Institute of Informatics Systems SB RAS (Novosibirsk, Russian Federation); Associate Professor, Novosibirsk State University (Novosibirsk, Russian Federation)

tatiana.v.batura@gmail.com
ORCID 0000-0003-4333-7888