УДК 004.912 + 004.8

Извлечение информации из научных текстов на русском языке

Батура Т.В. (Институт систем информатики им. А.П. Ершова СО РАН),

Бручес Е.П. (Институт систем информатики им. А.П. Ершова СО РАН, Новосибирский государственный университет),

Мезенцева А.А. (Институт систем информатики им. А.П. Ершова СО РАН, Новосибирский государственный университет)

В статье описаны методы автоматического извлечения терминов и связывания их с Викиданными. Преимуществом предложенных методов является потенциальная возможность их применения к любым областям знаний при наличии только неразмеченных текстов и начальных словарей терминов небольшого размера. Для проведения экспериментов был собран и размечен корпус научных текстов RuSERRC. Корпус и модели находятся в открытом доступе и могут быть полезны для дальнейших исследований другими научными коллективами.

Ключевые слова: *извлечение информации, машинное обучение, компьютерная лингвистика,* обработка текстов, *извлечение терминов, связывание сущностей.*

1. Введение

С распространением Интернета количество информации растет чрезвычайно быстро. По данным журнала “Nature” [15] во всем мире ежегодное количество научных публикаций с 2008 по 2018 г. выросло с 1.8 миллиона до 2.6 миллионов статей только по биомедицинской тематике. Однако эффективная обработка и извлечение наиболее важной информации из текстов является трудоемкой задачей. Тексты разных жанров отличаются по структуре и содержанию. Например, научные отчеты и публикации содержат ценные сведения о передовых достижениях в разных областях знаний, а правительственные документы (распоряжения, отчеты, постановления) описывают проделанную работу и запланированные мероприятия по развитию регионов. Очевидно, что для более эффективного анализа большого потока информации, необходимо создавать новые автоматические методы и инструменты.

Одной из фундаментальных задач извлечения информации из текстов является распознавание именованных сущностей (Named Entity Recognition, NER), которая в

настоящее время не является полностью решенной. Под сущностями понимаются слова или группы слов, отражающие основной смысл текста. Для того, чтобы решить обозначенную задачу, необходимо найти и классифицировать упоминания именованных сущностей в тексте по заранее определенным категориям, таким как имена людей, организации, местоположения, медицинские коды, научные термины, выражения времени, денежные значения и т.д.

Не менее важной является задача связывания сущностей (Entity Linking, EL), которая состоит в том, чтобы алгоритм мог автоматически соотнести упоминание сущности в тексте с сущностью в структурированной базе знаний, такой как Wikidata, DBPedia и др. Информация из базы знаний повышает качество автоматической системы, помогая разрешать лексическую неоднозначность слов и понятий, точнее определять их значение в текстах. Особую сложность представляет работа с информацией из узких предметных областей, когда подходящей терминологией владеют только специалисты. Поэтому для качественного автоматического извлечения информации важно, чтобы в системе присутствовал компонент связывания элементов текста с базой знаний.

Существующие на сегодняшний день автоматические методы, как правило, относительно неплохо решают обозначенные задачи для английского языка, но качество обработки текстов на русском языке оставляет желать лучшего. Для построения современных языковых моделей, которые используют алгоритмы машинного обучения, требуется большое количество обучающих данных. Разметка таких данных выполняется вручную и зависит от конкретной предметной области, поэтому существует проблема доступности специально подготовленных обучающих данных для большинства языков, в том числе для русского эта проблема стоит довольно остро.

В данной статье исследуются методы автоматического извлечения сущностей и связывания их с базой знаний. Для экспериментов собран и размечен корпус научных текстов RuSERRC, который также описан в данной статье. Исходный код и данные опубликованы в открытом доступе[1].

2. Обзор существующих методов

В настоящее время существует некоторое количество готовых решений, работающих с английским языком: OpenTapioca [8], OpenNRE [12], spaCy[2], Stazna[3]; есть открытые

[1] https://github.com/iis-research-team/terminator

[2] https://spacy.io

размеченные корпуса большого размера: TACRED [31], DocRED [27], SciERC [17], NNE [21], DWIE [29] и др. Для русского языка данных и исследований по извлечению информации из текстов значительно меньше: для извлечения сущностей из новостных текстов могут использоваться библиотеки DeepPavlov[4] и Natasha[5]; открытые коллекции данных содержат разметку только сущностей и отношений: FactRuEval [24], NEREL [16], RURED [11].

В данной работе под сущностью понимается слово или словосочетание, являющееся названием некоторого понятия из области науки, техники, искусства и др. В научных текстах (научных статьях, отчетах, диссертациях, монографиях) в роли сущностей выступают термины. Общая идея, которая лежит в основе традиционных подходов, состоит в том, что автоматическое извлечение терминов происходит в два этапа: на первом этапе из текстов извлекаются *n*-граммы слов, которые потенциально могут быть терминами, а на втором этапе выполняется классификация, в результате которой принимается решение, является ли данная фраза термином. Алгоритмы, архитектура которых соответствует этой идее, можно условно разделить на несколько групп.

Первая группа предполагает использование правил для выделения из текстов фраз, которые являются терминами. Например, в работе [23] предлагается использование словарей и информации о синтаксической структуре предложения для извлечения многословных терминов. Однако составление терминологических словарей вручную требует привлечения специалистов и затратно по времени.

Вторую группу представляют методы, в основе которых лежат алгоритмы машинного обучения с вручную извлеченными признаками. Например, в статье [6] авторы используют несколько групп признаков для извлечения терминов: лингвистические (части речи, главное слово фразы, количество имен существительных во фразе и др.), статистические (длина фразы, TF, IDF, TF-IDF и др.) и гибридные признаки (например, частота встречаемости фразы в корпусах обычных и научных текстов). Также было исследовано применение алгоритма PageRank для более точной классификации [32]. В работе [1] предлагается использовать признаки, основанные на информации из Викиданных. Главным недостатком таких методов является необходимость извлечения признаков вручную.

В третью группу входят методы глубокого обучения. В работе [26] исследуется проблема отсутствия достаточного количества данных. Для этого авторы на ограниченном количестве данных обучают две модели (CNN и LSTM), которые на вход принимают векторные

[3] https://stanfordnlp.github.io/stanza
[4] https://github.com/deepmipt/DeepPavlov/blob/master/docs/features/models/ner.rst
[5] https://github.com/natasha/natasha

представления слов фразы, а на выходе определяют, является данная фраза термином или нет. Затем этими моделями размечается новая порция данных, которая добавляется в обучающую выборку, и процесс обучения повторяется еще раз.

Четвертая группа опирается на методы тематического моделирования. В статье [2] описывается попытка применения различных методов тематического моделирования для улучшения нахождения однословных терминов: невероятностные (разные методы кластеризации – K-means, NFM и др.) и вероятностные (в качестве метода такой группы был выбран алгоритм LDA).

К пятой группе можно отнести методы, рассматривающие задачу извлечения терминов как задачу сопоставления последовательностей входных токенов с последовательностями меток из заранее определенного множества (sequence labelling task), т. е. для каждого токена в тексте требуется определить его класс (является он термином или нет). Таким образом, решение задачи осуществляется в один этап. Так, в работе [14] исследуются различные архитектуры и векторные представления слов при решении задачи sequence labelling. Большим преимуществом данного подхода является то, что во внимание принимается контекст (как синтаксический, так и семантический) употребления конкретной фразы, что составляет один из ключевых признаков для нахождения терминов в тексте.

Далее автоматическое связывание сущностей с элементами базы знаний выполняется в два этапа: генерация кандидатов и их ранжирование.

Для генерации множества кандидатов применяются различные подходы: сопоставление словоформ с заранее построенным индексом, методы нормализации строки и меры схожести триграмм [33]; страницы разрешения неоднозначности и редиректов Wikipedia, которые в том или ином виде содержат омонимичные и синонимичные слова и фразы [10]; априорную вероятность совместной встречаемости сущности и упоминания в различных источниках [5].

При ранжировании кандидатов 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 [4], так и полученных из языковых моделей [28]. 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 [25], SVM классификатор [30], глубокие нейронные сети [13]. В последнее время широкое распространение получили подходы, которые использую 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 [19] авторы строят векторные представления ребер графа, полученного из DBpedia, с помощью алгоритма DeepWalk [20]. В работе [18] авторы используют алгоритм TransE [3] для векторизации сущностей в графе.

Как правило, большинство упомянутых алгоритмов требуют для обучения большого количества специально подготовленных данных. Отличительной особенностью предлагаемых нами методов является возможность использования их, когда вручную размеченных данных имеется совсем немного.

3. Подготовка данных

Коллекции научных текстов существуют для английского языка и активно используются научным сообществом для обучения и оценки качества алгоритмов извлечения информации, однако в настоящее время на русском языке такие корпуса в открытом доступе не представлены. Поэтому было решено подготовить подобный корпус самостоятельно.

В собранную коллекцию RuSERRC[6] вошли тексты аннотаций научных статей по теме информационные технологии на основе данных, находящихся в открытом доступе, из журналов “Вестник НГУ. Серия: Информационные технологии”[7], “Программные продукты и системы”[8]. Объем корпуса 1600 неразмеченных документов и 80 текстов, вручную размеченных сущностями.

Разметка сущностей выполнялась в формате BIO (каждой единице текста присваивается значение тега B-TERM, если она является начальной для сущности, I-TERM, если она находится внутри термина или O, если она находится вне любой сущности). В рамках такой разметки предполагается, что именованные объекты не являются рекурсивными и не перекрываются. Всего в 80 размеченных текстах содержатся 11 157 токена и 2 027 терминов. Средняя длина термина – 2.43 слова. В качестве терминов рассматривались существительные и именные группы. Самый длинный термин состоит из 11 токенов.

Каждый документ был размечен двумя аннотаторами независимо, разногласия были разрешены модератором. Для аннотаторов была написана подробная инструкция с примерами. Процент согласия аннотаторов в задаче выделения сущностей составил 51.77%. Значение было вычислено как отношение пересечения выделенных терминов к объединению

[6] https://github.com/iis-research-team/ruserrc-dataset

[7] https://journals.nsu.ru/jit/archive/

[8] http://www.swsys.ru/

выделенных терминов. Полученное значение показывает высокую степень субъективности при нахождении слов и фраз, являющихся терминами, и при определении точных границ сущностей, что свидетельствует о сложности решаемой задачи.

Для поиска терминов в Викиданных были допущены следующие видоизменения сущностей.

– Все извлечённые сущности ищутся в базе знаний в нормализованной форме с учётом согласования и без учёта регистра, например: “*Линейных уравнений*” -> “*линейное уравнение*”.

– 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*”.

– 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.

– Также мы считаем синонимами термины “*подход*” и “*метод*”.

– 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.

– Если в тексте сущность написана с опечаткой, то в графе знаний мы ищем сущность без опечатки, например: “*3Dреконструкцию*” -> “*3d реконструкция*”.

– Допускаются трансформации вида “*архитектура системы*” -> “*системная архитектура*”.

– Расшифрованные аббревиатуры, например “*wps*” -> “*Wi-Fi Protected Setup*”.

– 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.

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 %. Всего в корпусе выделено 3386

терминов, 1337 из которых удалось связать с сущностями в Викиданных. Средняя длина связанной сущности – 1,55 токен, минимальная длина – 1 токен, максимальная – 8 токенов.

4. Описание предлагаемых методов

4.1. Извлечение терминов

Для извлечения терминов были реализованы: словарный метод, статистический метод и методы на основе архитектуры BERT.

В качестве базового алгоритма был реализован метод на основе словаря. Его идея состоит в том, чтобы собрать конечный словарь фраз, которые являются терминами, а затем искать их во входном тексте. Как правило, метод такого типа обладает высокой точностью, но низкой полнотой, т.к. учесть разнообразие всех форм терминов, а также появление новых, невозможно. В рамках работы был собран словарь из 17 252 терминов длиной от 1 до 12 токенов. При реализации словарного подхода были использованы библиотеки NLTK[9] для токенизации, pymorphy2[10] для лемматизации и ahocorapy[11] для построения префиксного дерева и работы с ним.

Для сравнения был рассмотрен статистический метод RAKE (Rapid automatic keyword extraction) [22], который кратко может быть описан следующим образом. Сначала применяется список стоп-слов и разделителей для выделения многословных терминов. После чего используется статистическая информация: для каждого слова из ключевых фраз-кандидатов оценивается частота, с которой оно встречалось, и количество связей между этим словом и остальными. На основании этих двух величин вычисляется вес ключевой фразы, и все фразы сортируются по весам, наиболее вероятные ключевые фразы получают максимальный вес. Этот метод хорошо применим к динамическим корпусам документов и к абсолютно новым областям знаний, при этом не зависит от языка и его особенностей. Было замечено, что данный алгоритм среди результатов часто выдает словосочетания, содержащие глагольные формы. Так как в качестве терминов мы рассматриваем существительные или именные группы, было решено оптимизировать эксперимент и выполнить предобработку текстов, убрав глаголы и их формы перед применением RAKE.

Кроме этого, была проведена серия экспериментов с использованием методов машинного обучения. Сложность проведения экспериментов с использованием различных алгоритмов

[9] https://pypi.org/project/nltk/
[10] https://pypi.org/project/pymorphy2/
[11] https://pypi.org/project/ahocorapy/

машинного обучения заключается в отсутствии размеченных данных. Эта проблема была решена следующим образом. Были взяты 1 118 полных текстов научных статей (включая, аннотацию и основную часть), которые предварительно были очищены от формул, таблиц, схем и пр., и автоматически размечены терминами из словаря, описанного выше. Таким образом, у нас получился размеченный набор данных, общим объёмом 1 992 498 токенов и содержащий 177 050 терминов.

Была поставлена гипотеза, что обобщающая способность модели позволит находить термины в текстах, где, предположительно, концентрация терминов выше, в то время как, модель была обучена на полных текстах статей, в которых концентрация терминов ниже. Также, таким способом, будут находиться термины в текстах, которые отсутствовали в исходном словаре.

Для проверки этой гипотезы были проведены эксперименты с посимвольной нейронной сетью, а также предложен итеративный метод на основе слабо контролируемого обучения к извлечению терминов. Для получения векторных представлений слов была использована предобученная модель BERT bert-base-multilingual-cased [9]. На вход модели подаётся токенизированный текст (входные тексты никак не преодобрабатываются). Выход модели представляет собой последовательность предсказанных классов для соответствующих токенов. Были проведены эксперименты с двумя архитектурами моделей: BERT-LSTM: полученные векторные представления подавали на вход двунаправленной LSTM, за которой идут два полносвязных слоя, и BertForTokenClassification: после векторных представлений идёт один полносвязный слой. Идея предложенного подхода заключается в том, чтобы обучить модель на небольшом количестве размеченных данных, а затем разметить полученной моделью некоторое количество новых текстов, добавить их к обучающему множеству и обучить вторую модель.

Для более точного определения границ терминов, были реализованы несколько эвристик, которые учитывали части речи слов, входящих в состав термина, и ближайших к термину, а также некоторые другие грамматические характеристики.

4.2. Связывание терминов с элементами базы знаний

В качестве входных данных алгоритму подается 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 слова по отдельности, а сохранить согласование, например, из “*обработке текстов*” нужно получить “*обработка текстов*”. Для этого мы использовали библиотеку для анализа текстов на русском языке Natasha[12], которая позволяет приводить к нормальной форме не только отдельные словоформы, но и словосочетания, а 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*”, т.к. на вход подаётся только сущность, без контекста.

На этапе генерации кандидатов входная строка сравнивается с названием сущности и её 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.

Стоит отметить, что этот алгоритм не подразумевает использование 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 счёт генерации синонимов и альтернативных написаний сущности для поиска кандидатов, что также пока не реализовано.

5. Результаты экспериментов

Все подходы для извлечения терминов сравнивались друг с другом по известным метрикам информационного поиска – точность, полнота, F-мера на описанном выше корпусе RuSERRC. При реализации метрик были использованы библиотеки Scikit-learn[13] и Seqeval[14]. Для большей информативности учитывалось также, была ли найдена сущность полностью или только частично – из-за того, что определение границ термина является субъективной

[12]https://github.com/natasha/natasha
[13] https://pypi.org/project/scikit-learn/
[14] https://pypi.org/project/seqeval/

задачей, это разделение видится важным. Полученные значения для всех подходов представлены в Таблице 1.

Таблица 1 – Результаты для задачи извлечения терминов

Метод	**Полное совпадение**			**Частичное совпадение**		
	Точность	**Полнота**	**F1**	**Точность**	**Полнота**	**F1**
Словарный подход	0.25	0.17	0.20	**0.82**	0.34	0.48
RAKE	0.36	0.28	0.32	0.62	0.63	0.63
RAKE оптимизированный	**0.44**	**0.35**	**0.39**	0.65	0.57	0.61
Посимвольная нейронная сеть	0.19	0.13	0.15	**0.82**	0.28	0.42
BERT-LSTM + эвристики + словарный подход	0.39	0.31	0.35	0.78	**0.78**	**0.77**
BertForTokenClassification + эвристики + словарный подход	0.40	0.31	0.35	0.77	0.77	**0.77**

Полученные результаты показали, что статистический подход с модификацией даёт лучшие значения метрик при определении чётких границ терминов, в то время как модели, полученные на основе слабо контролируемого обучения, показывают более высокие результаты, чем остальные методы, и являются достаточными для применения подхода при решении практических задач.

Для оценки алгоритма связывания терминов использовались известные метрики: *accuracy, linked_accuracy, averaged_candidates, linked_averaged_candidates и top_candidates*. Значения полученных метрик представлены в таблице 2.

Таблица 2 – Результаты для задачи связывания сущностей

Метрики	**Baseline-1**	**Baseline-2**
accuracy	**0.71**	0.55
linked_accuracy	0.53	**0.54**
averaged_candidates	1.95	**10.29**
linked_averaged_candidates	2.72	**7.38**
top_candidates	0.68	**0.76**

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*, что 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. Эти задачи планируется реализовать в ходе дальнейшей работы.

Также стоит отметить, что все эксперименты проводились на текстах из области информационных технологий, но реализованные алгоритмы могут быть потенциально применимы и расширены для других областей знаний при наличии только неразмеченных текстов и начального словаря терминов.

6. Заключение

В данной статье описаны методы автоматического извлечения терминов и связывания их с Викиданными. Для извлечения терминов были описаны и реализованы: словарный метод, статистический метод и методы на основе архитектуры BERT. Лучшие результаты показал метод на основе слабо контролируемого обучения, в котором используется архитектура BERT-LSTM и эвристики. Для задачи связывания сущностей на похожем англоязычном корпусе STEM-ECR согласно опубликованным данным [7] было получено значение *accuracy* равно 0.37, что согласуется с нашими результатами.

Для проведения экспериментов был собран и размечен корпус научных текстов RuSERRC. Преимуществом предложенных методов является потенциальная возможность их применения к любым областям знаний при наличии только неразмеченных текстов и начальных словарей терминов небольшого размера. Корпус и модели находятся в открытом доступе и могут быть полезны для дальнейших исследований другими научными коллективами.

Список литературы

1. Bilu Y., Gretz Sh., Cohen E., Slonim N. What if we had no Wikipedia? Domain-independent Term Extraction from a Large News Corpus. arXiv: 2009.08240. 2020.
2. Bolshakova E., Loukachevitch N., Nokel M. Topic Models Can Improve Domain Term Extraction. In:European Conference on Information Retrieval (ECIR 2013). Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, 2013, vol. 7814, p. 684–687.

3. 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.
4. 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.
5. 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.
6. Conrado M., Pardo T., Rezende S. O. A Machine Learning Approach to Automatic Term Extraction using a Rich Feature Set. In: Proceedings of the NAACL HLT 2013 Student Research Workshop. Atlanta, Georgia, 2013, p. 16–23.
7. D'Souza J., Hoppe A., Brack A., Jaradeh M., Auer S., Ewerth R. The STEM-ECR Dataset: Grounding Scientific Entity References in STEM Scholarly Content to Authoritative Encyclopedic and Lexicographic Sources. Proceedings of the 12th Language Resources and Evaluation Conference. European Language Resources Association, Marseille, France, 2020. pp. 2192–2203.
8. Delpeuch A. OpenTapioca: Lightweight Entity Linking for Wikidata. 2019. arXiv:1904. 09131.
9. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of NAACL-HLT 2019. 2019. pp. 4171–4186.
10. 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.
11. Gordeev D., Davletov A., Rey A., Akzhigitova G., Geymbukh G. Relation extraction dataset for the russian language. In: Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialog" [Komp'iuternaia Lingvistika i Intellektual'nye Tehnologii: Trudy Mezhdunarodnoj Konferentsii "Dialog"]. 2020. DOI: 10.28995/2075-7182-2020-19-348-360.
12. Han X., Gao T., Yao Yu., Ye D., Liu Zh., Sun M. OpenNRE: An open and extensible toolkit for neural relation extraction. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations. ACL, 2019. pp. 169-174.
13. Huang H., Heck L., Ji H. Leveraging deep neural networks and knowledge graphs for entity disambiguation. 2015. arXiv:1504.07678.
14. Kucza M., Niehues J., Zenkel T., Waibel A., Stüker S. Term Extraction via Neural Sequence Labeling a Comparative Evaluation of Strategies Using Recurrent Neural Networks. In: Proceedings of Interspeech 2018. 2018. p. 2072-2076.
15. Landhuis E. Scientific literature: Information overload. Nature. 2016. N 535. pp. 457–458.
16. Loukachevitch N., Artemova E., Batura T., Braslavski P., Denisov I., Ivanov V., Manandhar S., Pugachev A., Tutubalina E. NEREL: A Russian dataset with nested named entities, relations and

events. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021). 2021. pp. 876-885.

17. Luan Y., He L., Ostendorf M., Hajishirzi H. Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. 2018. pp. 3219-3232.
18. 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.
19. 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.
20. 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.
21. Ringland N., Dai X., Hachey B., Karimi S., Paris C., Curran J.R. NNE: A dataset for nested named entity recognition in english newswire. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019. pp. 5176-5181.
22. Rose S., Engel D., Cramer N., Cowley W. Automatic Keyword Extraction from Individual Documents // Text mining: applications and theory. 2010. pp.1–20.
23. Stanković R., Krstev C., Obradović I., Lazić B., Trtovac A. Rule-based Automatic Multiword Term Extraction and Lemmatization. In: Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC'16). 2016, p. 507–514.
24. Starostin A., Bocharov V., Alexeeva S., Bodrova A., Chuchunkov A., Dzhumaev S., Efienko I., Granovsky D., Khoroshevsky V., Krylova I., Nikolaeva M., Smurov I., Toldova S. Factrueval 2016: Evaluation of named entity recognition and fact extraction systems for Russian. In: Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialog" [Komp'iuternaia Lingvistika i Intellektual'nye Tehnologii: Trudy Mezhdunarodnoj Konferentsii "Dialog"]. 2016. pp. 702-720.
25. 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.
26. Wang R., Liu W., McDonald C. Featureless Domain-Specific Term Extraction with Minimal Labelled Data. In: Proceedings of Australasian Language Technology Association Workshop, 2016, p. 103−112.
27. Yao Y., Ye D., Li P., Han X., Lin Y., Liu Z., Liu Z., Huang L., Zhou J., Sun M. DocRED: A large-scale document-level relation extraction dataset. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019. pp. 764-777.

28. 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.
29. Zaporojets K., Deleu J., Develder C., Demeester T. DWIE: An entity-centric dataset for multi-task document-level information extraction. Information Processing & Management. 2021. V. 58. N 4. pp. 102563.
30. 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.
31. Zhang Y., Zhong V., Chen D., Angeli G., Manning C.D. Position-aware attention and supervised data improve slot filling. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017). 2017. pp. 35-45.
32. Zhang Z., Gao J., Ciravegna F. SemRe-Rank: Improving Automatic Term Extraction by Incorporating Semantic Relatedness with Personalised PageRank. ACM Transactions on Knowledge Discovery from Data (TKDD), 2018, vol. 12, no. 5, p. 1–41.
33. 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.