ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ НАУКИ
ИНСТИТУТ СИСТЕМ ИНФОРМАТИКИ ИМ. А.П. ЕРШОВА
СИБИРСКОГО ОТДЕЛЕНИЯ РОССИЙСКОЙ АКАДЕМИИ НАУК

На правах рукописи

Бручес Елена Павловна

МЕТОДЫ И АЛГОРИТМЫ РАСПОЗНАВАНИЯ И СВЯЗЫВАНИЯ СУЩНОСТЕЙ ДЛЯ ПОСТРОЕНИЯ СИСТЕМ АВТОМАТИЧЕСКОГО ИЗВЛЕЧЕНИЯ ИНФОРМАЦИИ ИЗ НАУЧНЫХ ТЕКСТОВ

Специальность 05.13.17 – Теоретические основы информатики

Диссертация на соискание учёной степени
кандидата технических наук

Научный руководитель
кандидат физико-математических наук, доцент
Батура Татьяна Викторовна

Новосибирск – 2021

Оглавление

Введение ……..4

Глава 1. Задача извлечения именованных сущностей и отношений между ними, обзор методов и подходов ……..11

1.1. Извлечение именованных сущностей……11

1.1.1. Описание задачи ……..11

1.1.2. Методы и подходы к решению задачи извлечения именованных сущностей……...12

1.1.3. Методы и подходы к решению задачи извлечения научных терминов……..15

1.2. Извлечение и классификация семантических отношений ……...17

1.2.1. Описание задачи извлечения и классификации отношений ……...17

1.2.2. Методы и подходы к решению задачи ……..18

1.3 Задача одновременного извлечения именованных сущностей и отношений……..21

1.3.1 Описание задачи……...21

1.3.2 Методы и подходы к решению задачи ……...22

1.4. Задача связывания сущностей ……24

1.4.1. Описание задачи связывания сущностей……..24

1.4.2. Методы и подходы к решению задачи связывания сущностей……...24

Глава 2. Корпуса для решения поставленных задач ……27

2.1 Существующие размеченные корпуса ……27

2.2. Создание корпуса RuSERRC……...28

2.2.1. Состав корпуса ……28

2.2.2. Описание разметки научных терминов……29

2.2.3. Описание разметки отношений между научными терминами ……...32

2.2.4 Описание разметки связывания сущностей……...37

2.3 Выводы……...39

Глава 3. Извлечение научных терминов ……41

3.1 Формальная постановка задачи ……...41

3.2 Описание методов ……41

3.2.1 Метод на основе словаря……41

3.2.2 Метод RAKE……44

3.2.3 Метод на основе машинного обучения ……45

3.2.4 Метод на основе слабо контролируемого обучения (weak supervision)……..47

3.3 Описание результатов……...50

3.3.1 Метрики ……50

3.3.2 Результаты……51

3.4 Применение модели к текстам другой области знаний……54

3.5 Выводы ……..57

Глава 4. Извлечение и классификация отношений между научными терминами…….58

4.1 Формальная постановка задачи ……...58
4.2 Классификация отношений ……..58
4.3 Извлечение отношений……60
4.3.1 Использование модели классификации отношений ……61
4.3.2 Подход, основанный на лексических шаблонах……61
4.3.3 Подход, основанный на zero-shot learning ……62
4.3.4 Ансамбль решений……...62
4.4 Описание результатов ……...63
4.4.1 Метрики ……63
4.4.2 Результаты……64
4.4.2.1 Результаты для задачи классификации отношений ……64
4.4.2.2 Результаты для задачи извлечения отношений……65
4.5 Выводы……...68

Глава 5. Автоматическое связывание сущностей ……69
5.1 Формальная постановка задачи ……...69
5.2. Описание алгоритма ……69
5.3 Описание результатов ……...71
5.3.1 Метрики ……71
5.3.2 Результаты……73
5.4 Выводы……...73

Заключение……...75

Список сокращений и условных обозначений……...77

Список литературы……79

Приложение 1. Пример разметки корпуса ……91

Приложение 2. Фрагмент матрицы переходов ……..97

Приложение 3. Лексико-синтаксические шаблоны для извлечения отношений ……...99

Приложение 4. Метрики извлечения отношений по сущностям ……102

Приложение 5. Схема работы системы извлечения информации ……...105

Приложение 6. Грамоты ……..106

Приложение 7. Акты о внедрении ……108

Приложение 8. Свидетельство о регистрации программы для ЭВМ ……112

Введение

Актуальность темы исследования. В связи с бурным ростом количества данных, в том числе и текстовых, активно развивается область обработки естественных языков. Решение таких задач позволяет более эффективно анализировать информацию для своих целей, экономя силы и время.

В последнее время особый интерес представляет автоматический анализ научных публикаций. Согласно исследованиям, ежегодное количество публикаций с 2008 г. до 2018 г. выросло с 1.8 миллиона до 2.6 миллионов статей [1]. Очень важно следить за трендами и исследованиями в научных статьях, сравнивать предлагаемые методы для тех или иных задач, находить нужную информацию и многое другое. Очевидно, что проделать всю эту работу вручную невозможно, именно поэтому разработка инструментов для текстов научной тематики сегодня является одной из самых актуальных задач.

Стоит отметить, что такие тексты отличаются от остальных особой морфологией и лексикой, а также определёнными синтаксическими и семантическими структурами. Кроме того, тексты научных статей состоят из блоков, которые располагаются в общепринятом порядке: так, например, сначала идёт название статьи, авторы и их аффилиации, затем аннотация статьи; основной текст состоит, как правило, из введения, обзора работ по данной теме, описания предложенного метода, результатов, заключения и списка литературы. Такое деление на блоки упрощает поиск нужной информации не только для человека, но и при автоматической обработке текстов.

Много работ ведётся в области обработки текстов именно научных статей, и решаются абсолютно разные задачи. Например, работа [2] посвящена нахождению терминов, формул, таблиц в тексте статьи и связывании их друг с другом, помогая читателю лучше ориентироваться в таких объектах, не возвращаясь каждый раз к определениям. Активно решается задача автоматического реферирования текстов научных статей [3, 4].

Существует много работ, посвящённых извлечению различной информации из научных текстов: в работе [5] авторы извлекают библиографические данные из текстов статей; в работе [6] авторы предлагают метод для извлечения условий эксперимента; в статье [7] авторы работают над извлечением и нормализацией физических характеристик (критическая температура, давление и пр.); в статье [8] описывается метод извлечения информации о химических процессах и условиях их протекания; в работе [9] предлагается метод извлечения упоминаний наборов данных, которые используются в экспериментах, что может быть

полезным для автоматического сравнения метрик на этих корпусах; в работе [10] авторы извлекают изображения из текстов научных статей.

Современные подходы для решения таких задач подразумевают использование алгоритмов машинного обучения. Качество таких алгоритмов напрямую зависит от качества данных, которые используются для их обучения. Для подготовки и разметки данных необходимо наличие специалистов и времени. Поэтому сегодня особенно актуальными являются методы, не требующие большого количества размеченных данных. Здесь можно выделить следующие основные направления:

1. Обучение на неразмеченных данных – это различные методы кластеризации;
2. Использование мультиязычных моделей – идея состоит в том, чтобы взять языковую модель, обученную на текстах разных языков, затем дообучить на данных высокоресурсных языков, а использовать на данных малоресурсных языков. Такой подход применяется при решении задачи машинного перевода [11], различных задачах тэгирования последовательностей (англ. sequence labelling): извлечение именованных сущностей, определение семантических ролей, извлечение аспектов [12], семантического анализа [13] и др.;
3. Аугментация данных – увеличение количества данных для обучения за счёт изменения существующих данных. Эта идея пришла из области компьютерного зрения, где в качестве аугментирования данных выступают такие операции над изображениями, как повороты, отражения, кадрирование, внесение шума и др. Примеры работ: [14, 15, 16]. В задачах обработки текстов использование данной методики тоже способно улучшить качество системы. Здесь могут быть использованы такие изменения, как замена синонимов, мена слов местами, добавление опечаток и пр. Примеры работ: [17, 18, 19].

Задача извлечения информации из текстов является не только важной задачей самой по себе, но также и основным этапом для других задач (например, автоматического реферирования), поэтому требуется высокое качество её решения. Можно сказать, что эта задача хорошо решается для английского языка, что связано с наличием большого количества данных, исследователей, вовлечённых в работу, и пр. Но использовать такие системы для русского языка представляется невозможным, т.к. русский язык имеет свои морфологические и синтаксические особенности, которые должны учитываться при разработке подобных алгоритмов.

Более того, русский язык считается малоресурсным – это означает, что количество данных (не только размеченных, но и неразмеченных) существенно ниже, чем для английского языка.

Это тоже вызывает сложности при построении систем для решения любых задач обработки текстов для русского языка.

Эти факты обуславливают актуальность темы исследования. В данной диссертационной работе рассмотрены методы и алгоритмы для решения нескольких задач извлечения информации, которые не требуют большого количества вручную размеченных данных. Полученные результаты показали, что при полном отсутствии вручную размеченных данных возможно разработать систему извлечения информации с достаточным качеством для применения на практике.

Степень разработанности темы исследования. В последнее время наблюдается рост публикаций, посвященных анализу именно научных текстов.

Извлечение научных терминов исследуется в трудах Н.В. Лукашевич, Е.И.Большаковой, Kucza M., Niehues J. и др.

Извлечение отношений в научных текстах является тесно связанной с извлечением терминов и решается такими исследователями, как Hearst M., Huang K., Wang G. и др.

Также в последнее время особое внимание уделяется задаче одновременного извлечения сущностей и отношений между ними, например, в работах Ryuichi T., Tianyang Z., Eberts M., Ulges A. и др.

Объектом исследования являются тексты научных статей на русском языке.

Предметом исследования являются методы автоматического извлечения информации из текстов на естественном языке.

Цель и задачи работы. Целью работы является исследование и разработка методов, применяемых для решения задач извлечения терминов и семантических отношений между ними, а также связывания их с внешней базой знаний, и реализация основных компонентов системы извлечения информации из научных текстов на русском языке.

Требования к предлагаемым алгоритмам:

1. Реализация в условиях недостаточного количества размеченных данных;
2. Независимость от области знаний.

Для достижения поставленной цели были определены следующие задачи:

1. Предложить и реализовать метод извлечения научных терминов, слабо зависящий от области знаний;
2. Адаптировать метод извлечения отношений между терминами, основанный на переносе обучения моделей с английского языка на русский в постановке zero-shot learning;

3. Описать алгоритм и реализовать метод связывания терминов с сущностями в базе знаний;
4. Разработать методику разметки корпуса текстов на русском языке для обучения и оценки качества алгоритмов и методов;
5. Разработать программный комплекс для извлечения терминов и отношений из научных текстов и связывания терминов с внешней базой знаний.

Соответствие диссертации паспорту научной специальности. Диссертация соответствует области исследований специальности 05.13.17 – Теоретические основы информатики по п. 5 «Разработка и исследование моделей и алгоритмов анализа данных, обнаружения закономерностей в данных и их извлечениях разработка и исследование методов и алгоритмов анализа текста, устной речи и изображений»; п. 6 «Разработка методов, языков и моделей человеко-машинного общения; разработка методов и моделей распознавания, понимания и синтеза речи, принципов и методов извлечения данных из текстов на естественном языке»; п. 12 «Разработка математических, логических, семиотических и лингвистических моделей и методов взаимодействия информационных процессов, в том числе на базе специализированных вычислительных систем».

Методы исследования. Методологической основой исследования являются методы компьютерной лингвистики, статистические методы и методы машинного обучения, успешно зарекомендовавшие себя в задачах анализа текстов. Для программной реализации системы использовались методы объектно-ориентированного программирования.

Научная новизна работы заключается в следующем:

1. Предложен новый метод извлечения терминов из научных текстов, основанный на частичном обучении, который может применяться к текстам разных областей знаний.
2. Разработан и реализован метод извлечения семантических отношений, позволяющий решать задачу в условиях ограниченного количества размеченных данных. Метод основан на технике "обучения без примеров" (zero-shot learning) путем переноса обучения моделей с английского языка на русский и потенциально применим для широкого круга малоресурсных языков.
3. Разработана методика подготовки и разметки данных. В ходе исследования подготовлен корпус текстов на русском языке, который содержит трехуровневую разметку и служит основой для обучения и оценки качества современных автоматических методов извлечения информации.

Теоретическая ценность и практическая значимость состоит в том, что в работе даны формальные описания предлагаемых алгоритмов и методов. На базе разработанных методов создан программный комплекс для извлечения информации из научных текстов на русском языке. Разработанные методы, алгоритмы и программное обеспечение могут применяться для построения систем машинного понимания текста, систем автоматической обработки текста, информационно-поисковых систем и других информационных систем, основанных на знаниях. Предложенные методы могут быть легко адаптированы к текстам других областей знаний.

Полученная система использовалась в работе, которая ведётся в рамках проекта РФФИ № 19-07-01134 А «Создание моделей, методов и программных средств анализа текстов на естественном языке для использования в интеллектуальных информационных системах», а также поддержаны стипендией Правительства Российской Федерации для студентов высшего профессионального образования и аспирантов, обучающихся по имеющим государственную аккредитацию образовательным программам, соответствующим приоритетным направлениям модернизации и технологического развития экономики России.

Получено свидетельство о государственной регистрации программы для ЭВМ №20216111340 от 26.01.2021.

Основные положения, выносимые на защиту:

1. Разработана методика подготовки и разметки данных для задач извлечения терминов, отношений и связывания сущностей с элементами Wikidata. С помощью этой методики подготовлен корпус. Показана значимость данного корпуса для исследовательских целей. В частности, он может служить основой для обучения и оценки качества современных автоматических методов извлечения информации.
2. Предложен новый метод извлечения терминов из научных статей. Метод основан на частичном обучении и не зависит от области знаний и жанра текстов.
3. Адаптирован метод извлечения семантических отношений, основанный на технике "обучения без примеров" (zero-shot learning). Показано, что метод переноса обучения моделей с английского языка на русский хорошо работает для задачи классификации отношений.
4. Реализован алгоритм автоматического связывания научных терминов с сущностями в базе знаний Wikidata. Предложен ряд метрик для оценки качества метода, учитывающих различные аспекты. Описанные метрики показали сильные и слабые стороны реализованного алгоритма.

Достоверность результатов. Все полученные результаты подтверждаются экспериментами, проведенными в соответствии с общепринятыми стандартами.

Апробация результатов исследования. Основные результаты работы докладывались на следующих конференциях:

1. XXIII "Data analytics and management in data intensive domains" conference (DAMDID), Россия, Москва, 2021;
2. XXII Всероссийская конференция молодых учёных по математическому моделированию и информационным технологиям, Россия, Новосибирск, 2021;
3. Science and Artificial Intelligence conference (SAIC-2020), Россия, Новосибирск, 2020;
4. Международная научно-техническая конференция "Автоматизация" (RusAutoCon), Россия, Сочи, 2018;
5. 12-ая международная научно-практическая конференция «Виртуальные и интеллектуальные системы – ВИС-2017», Россия, Барнаул, 2017;
6. International Conference on Analysis of Images, Social Networks and Texts 2016 (AIST 2016), Россия, Екатеринбург, 2016.

Кроме того, результаты исследования обсуждались на ряде регулярных семинаров в Институте систем информатики им. А.П. Ершова СО РАН, Федеральном исследовательском центре информационных и вычислительных технологий, Новосибирском государственном университете.

Публикации. Основные результаты диссертации опубликованы в 10 научных статьях, из них: 3 в журналах из перечня ВАК РФ, 3 в изданиях, индексируемых Scopus; докладывались автором на 6 международных научных конференциях (Москва, Екатеринбург, Барнаул, Сочи, Новосибирск).

Получено 1 свидетельство о государственной регистрации программ для ЭВМ.

Основные результаты диссертации содержатся в работах [95-105].

Личный вклад соискателя. Содержание диссертации и основные положения, выносимые на защиту, отражают персональный вклад автора в опубликованные работы. Все представленные в диссертации результаты получены лично автором.

Объём и структура диссертационной работы. Диссертация состоит из введения, пяти глав, заключения и 8 приложений. Полный объем диссертации составляет 112 страниц, включая 7 рисунков и 22 таблицы. Список литературы содержит 105 наименований.

Содержание работы. Во ***введении*** обосновывается актуальность исследований, проводимых в рамках данной диссертационной работы, приводится обзор научной литературы

по изучаемой проблеме, формулируется цель, ставятся задачи работы, излагается научная новизна и практическая значимость представляемой работы.

В ***первой главе*** формулируются задачи извлечения сущностей, отношений между ними, а также связывания сущностей с внешней базой знаний. Приводится обзор существующих работ для каждой из этих задач.

Во ***второй главе*** проводится анализ существующих размеченных наборов данных для задачи извлечения сущностей и отношений между ними. Описывается процедура разметки корпуса для поставленных задач: приводится подробная инструкция разметки, процесс, а также анализ полученного корпуса.

В ***третьей главе*** дано формальное описание задачи извлечения научных терминов. Описаны алгоритмы, которые были реализованы в рамках данной работы: словарный подход, статистический подход, а также подходы, основанные на использовании алгоритмов глубокого обучения. Предложены метрики для оценки качества реализованных подходов, а также проведён анализ полученных результатов.

В ***четвёртой главе*** дано формальное описание задачи извлечения отношений между научными терминами. Решена задача классификации отношений в постановке zero-shot learning. Реализованы алгоритмы для задачи извлечения отношений: с использованием лексико-синтаксических шаблонов, с использованием модели для классификации отношений, а также алгоритмы zero-shot learning с различными подходами к сэмплированию данных.

В ***пятой главе*** дано формальное описание задачи автоматического связывания сущностей с внешней базой знаний, а также реализован алгоритм, основанный на эвристическом и статистическом подходе.

В ***заключении*** сделаны выводы, подведены итоги проведенного исследования, а также изложены рекомендации и перспективы дальнейшей разработки темы.

В ***приложениях*** приведён пример разметки корпуса, создание которого описано в данной работе; фрагмент матрицы переходов для конечного автомата, который используется в рамках словарного подхода для извлечения научных терминов; лексико-синтаксические шаблоны для определения типа отношений одним из методов; метрики извлечения отношений для отдельных классов; схема работы системы извлечения информации, а также грамоты, акты о внедрении и свидетельство о государственной регистрации ПО.

Глава 1. Задача извлечения именованных сущностей и отношений между ними, обзор методов и подходов

1.1. Извлечение именованных сущностей

1.1.1. Описание задачи

Извлечение информации (англ. Information extraction, IE) – это процесс поиска в тексте необходимой информации, включая извлечение сущностей, отношений и, самое сложное, событий (с описанием того, когда и где происходило событие, кто участники и др.). Он требует более глубокого анализа, чем поиск по ключевым словам. Результатом задачи является преобразование неструктурированной информации к структурированному виду [20].

Извлечение информации включает в себя несколько подзадач, одной из которых является извлечение именованных сущностей. Традиционно под именованными сущностями понимались фразы, содержащие имена людей, названия организаций и географических объектов, например: “$[Иван]_{PER}$ *поехал в* $[Московскую\ область]_{LOC}$” [21]. Со временем, под именованными сущностями стали также рассматриваться такие категории, как временные выражения, валюта, процентные выражения и др.

С развитием технологий обработки текстов само понятие "именованной сущности" стало различаться в зависимости от области знаний. Так, например, в области медицины является актуальной задача извлечения названий болезней, лекарств, химических составляющих и т.д. [22, 23].

Задача извлечения именованных сущностей состоит в нахождении фрагментов текстов, которые состоят из сущностей (что является сущностью – зависит от конкретных области и задачи), и затем определении типа сущности. Процесс нахождения фрагментов осложняется неоднозначностью границ сущности – нужно решить, что является сущностью, а что – нет, и где проходят границы сущности [24]. Так, выделилась отдельная задача – выделение вложенных сущностей. Например, сущность “*Московский государственный университет имени Ломоносова*” содержит сущности двух типов: организация и персона.

Также задача осложняется неоднозначностью типов сущностей. Например, сущность “*Владимир*” может относиться к городу России, а может быть именем человека; “*Форд*” может относиться к имени человека или названию организации; “*Yesterday*” может быть названием песни, а может указывать на определённый промежуток времени. Один из подходов к решению

данной проблемы описан в статье [103]: авторы предлагают статистический подход, учитывающий только контекст предполагаемых географических названий, для снятия омонимии такого типа.

Другой тип неоднозначности состоит в определении конкретного объекта, о котором идёт речь в тексте: так, город Москва – это столица России и город в штате Айдахо, США. И определить, о каком именно объекте идёт речь, можно только на основании контекста. Эта задача – автоматического связывания сущностей. Выделенные сущности в тексте связываются с сущностями во внешней базе знаний, в которой сущность представляет собой конкретный объект окружающего мира. Решение этой задачи может обеспечить использование информации о мире при решении других задач. Например, для задачи автоматического перевода использование такое информации может быть ключевым для достижения высокого качества системы.

1.1.2. Методы и подходы к решению задачи извлечения именованных сущностей

Классическое решение этой задачи – это решение задачи тэгированния последовательности: для каждого токена из входной последовательности требуется определить класс (входит ли токен в состав сущности, и если входит, то в состав сущности какого типа). Традиционно, подходы делятся на две группы: с использованием различного вида вручную созданных правил и с использованием алгоритмов машинного обучения. В последнее время в связи с усовершенствованием аппаратного обеспечения, среди алгоритмов машинного обучения получили большое распространение методы глубокого обучения (англ. deep learning, DL).

Методы, основанные на правилах. К этой группе относятся методы, которые требуют огромного количества человеческого труда, заключающегося в ручном задании правил (например, на основе регулярных выражений, деревьев синтаксического разбора и т.д.) и создании словарей для предметной области. Для этих подходов характерна высокая интерпретируемость моделей и высокое качество за счет длительной разработки и невысокой обобщающей способности. Один из таких подходов, основанный на использовании индуктивного логического программирования, описан в работе [25]

Методы машинного обучения. Методы этой группы условно можно разделить на две группы: методы, которые в качестве входных признаков используют вручную сформированные признаки (hand-crafted features), и методы, которые в качестве входных признаков используют

только входной текст (модели полного цикла, end-to-end модели).

Рассмотрим методы, которые используют вручную извлечённые признаки. Такими признаками могут являться частеречные тэги, информация о синтаксических зависимостях, вхождение слова в тот или иной словарь или справочник и др. В частности, для русского языка в работе [26] предлагается использовать информацию и граф сущностей Wikidata для обучения модели, автоматического создания корпуса и сбора словаря именованных сущностей.

Методы второй группы работают только с текстом – он может быть представлен, как в виде токенов (word level), так и в виде отдельных символов (character level).

Работая на уровне слов, токены, как правило, получают векторные представления из уже предобученной языковой модели, такой как word2vec, ELMo (Embeddings from Language Model), BERT (Bidirectional Encoder Representations from Transformers) и другие. Например, в работе [27] было показано использование векторных представлений модели ELMo для решения задачи извлечения именованных сущностей. В статье [28] для решения задачи используются векторные представления BERT.

Если говорить про решение этой задачи на уровне символов, то основная идея состоит в том, чтобы закодировать текст на уровне символов, получить их векторные представления (как правило, здесь используются свёрточные или рекуррентные слои), а затем решать обычную задачу тэгирования последовательностей. В работе [29] описана архитектура применительно к русскому языку. В модели успешно комбинируются слои Bi-LSTM (Bidirectional long short-term memory) и CRF (Conditional Random Fields), что позволяет значительно увеличить качество распознавания именованных сущностей. Похожая модель предложена в статье [30]. Отличие состоит в том, что посимвольные векторные представления слов сначала проходят через свёрточную нейронную сеть – утверждается, что это позволяет лучше извлекать морфологические характеристики слов, и использовать их далее.

В работе [31] описан метод совместного обучения нейронной сети для задачи извлечения цепочек токенов, которые являются ключевыми словами в тексте, и задачи ранжирования ключевых слов. Такой подход позволяет находить ключевые слова большой длины, а также извлекать фразы, которые не являются сущностями, но имеют значение в тексте.

Отдельно хочется выделить относительно новые направления для развития методов выделения именованных сущностей.

К одному из таких направлений можно отнести методы, опирающиеся на технику обучения на малом количестве примеров (англ. few-shot learning) или “обучения без примеров” (англ. zero-shot learning). Так, целью работы [32] является автоматическое извлечение

сущностей с использованием всего лишь нескольких примеров для каждого типа сущности. Например, есть предложение с сущностью *xbox game*: “*I purchased a game called NBA 2k 19*”, в котором *NBA 2k 19* является сущностью. Тогда ожидается, что в предложении “*I cannot play Minecraft with error code 0x111*” будет распознана сущность *Minecraft* типа *xbox game*. Этот пример демонстрирует сценарий “один пример – одна сущность”. В работе описан подход, который покрывает сценарии, встречающиеся в реальной жизни – есть несколько типов сущности, на каждый из них приходится по несколько примеров.

Совершенно иная концепция решения задачи извлечения именованных сущностей представлена в работе [33]. В ней авторы предлагают решать эту задачу как машинное чтение и понимание текста (англ. machine reading comprehension, MCR): например, извлечение сущности типа PERSON можно формализовать как извлечение из текста ответа на вопрос “*Кто был упомянут в тексте?*”.

Другое направление – активное обучение (англ. active learning). Идея этого метода состоит в том, чтобы модель в случае неуверенности в предсказании обращалась к пользователю с целью разметить такие данные. В работе [34] приведён процесс обучения модели, использующий активное обучение, применительно к извлечению именованных сущностей из медицинских текстов.

В Таблицах 1 и 2 приведены метрики, полученные в результате описанных работ, для русского и английского языков соответственно. При анализе значений можно заметить, что современные контекстные векторные представления слова хорошо справляются с задачей извлечения именованных сущностей. Информация из дополнительных источников, таких как Википедия, может существенно увеличить точность модели, а решение задачи в постановке машинного понимания текстов показывает многообещающие результаты.

Таблица 1. Метрики извлечения именованных сущностей для русского языка

Источник	FactRuEval			Gareev's dataset			Persons-1000		
	Точность	Полнота	F1	Точность	Полнота	F1	Точность	Полнота	F1
Sysoev et al., 2016 [26]	**0.88**	0.65	0.75	-	-	-	-	-	-
Le et al., 2018 [29]	0.84	**0.80**	**0.82**	0.89	0.85	0.87	0.99	0.99	0.99

Таблица 2. Метрики извлечения именованных сущностей для английского языка

Источник	CoNLL 2003			OntoNotes 5.0		
	Точность	Полнота	F1	Точность	Полнота	F1
Li et al., 2021 [33]	0.92	0.94	**0.93**	0.93	0.90	0.91
Ma et al., 2016 [30]	-	-	0.91	-	-	-
Peters et al., 2018 [27]	-	-	0.92	-	-	-
Devlin et al., 2019 [35] (BERT base)	-	-	0.92	-	-	-
Devlin et al., 2019 [35] (BERT large)	-	-	0.93	-	-	-

1.1.3. Методы и подходы к решению задачи извлечения научных терминов

Общая идея, которая лежит в основе традиционных подходов к автоматическому извлечению терминов, состоит из двух этапов: на первом этапе из текстов извлекаются n-граммы, которые потенциально могут быть терминами, а на втором этапе выполняется классификация, является ли данная фраза термином или нет.

Алгоритмы, архитектура которых соответствует этой идее, можно также разделить на несколько групп.

Первая группа предполагает использование правил для выделения из текстов фраз, которые являются терминами. Например, в работах [36, 37] описывается использование словарей, синтаксическая и морфологическая информация для извлечения многословных терминов. Нередко для решения этой задачи используются заранее построенные онтологии. В работе [38] описан инструмент ANDDigest, позволяющий выполнять поиск информации по биомедицинским статьям, в основе работы которого лежат заранее построенные онтологии. В статье [39] описывается извлечение информации, в том числе, извлечение терминов, на основе медицинской онтологии.

Другую группу составляют методы, в основе которых лежит использование алгоритмов машинного обучения с вручную извлечёнными признаками. Так, в работе [40] описывается алгоритм извлечения как однословных, так и многословных терминов: на первом этапе из текстов извлекаются n-граммы, которые потенциально могут быть терминами, а затем на основании различных входных признаков алгоритм определяет, является ли n-грамма термином или нет. В статье [41] авторы используют несколько групп признаков для извлечения терминов: лингвистические (части речи, главное слово фразы, количество имён существительных во фразе и другие), статистические (длина фразы, TF (term frequency), IDF (inverse document frequency), TF-IDF (term frequency – inverse document frequency) и другие) и гибридные признаки (например, частота встречаемости фразы в корпусах обычных и научных текстов). Также было исследовано применение алгоритма PageRank для более точной классификации [42]. В работе [43] предлагается использовать признаки, основанные на информации из Википедии.

В третью группу входят методы глубокого обучения. В работе [44] исследуется проблема отсутствия достаточного количества данных. Для этого авторы на ограниченном количестве данных обучают две модели (CNN (Convolutional Neural Network) и LSTM), которые на вход принимают векторные представления слов фразы, а на выходе определяют, является ли данная фраза термином или нет. Затем этими моделями размечается новая порция данных, которая добавляется в обучающую выборку, и процесс обучения повторяется ещё раз.

В работе [45] предлагается архитектура, состоящая из этапов, отличных от тех, что были описаны: на первом шаге классификатор определяет, содержит ли входное предложение термины или нет; если содержит, то на втором этапе происходит непосредственно нахождение терминов в предложении.

Другая общая идея рассматривает задачу извлечения терминов, как задачу тегирования последовательности (англ. sequence labelling), т.е. для каждого токена в тексте требуется определить его класс (является он термином или нет). Таким образом, решение задачи осуществляется в один этап. Как правило, при таком подходе используется разметка сущностей в формате BIO (BILOU и другие). Большим преимуществом данного подхода является то, что во внимание принимается контекст (как семантический, так и синтаксический) употребления конкретной фразы, что является одним из ключевых признаков для нахождения терминов в тексте. Так, в работе [46] исследуются различные архитектуры и векторные представления слов при решении задачи sequence labelling.

Ещё одна идея, которая отличается от описанных выше, состоит в использовании методов тематического моделирования (англ. topic modelling) для извлечения терминов. В статье [47] описывается попытка применения различных методов тематического моделирования для улучшения нахождения однословных терминов: невероятностные (разные методы кластеризации – K-means, NFM (Non-negative matrix factorization) и другие) и вероятностные (в качестве метода такой группы был выбран алгоритм LDA (Latent Dirichlet allocation)).

Значения метрик для задачи извлечения научных терминов для английского языка представлены в Таблице 3.

Таблица 3. Метрики извлечения научных терминов для английского языка

Источник	GENIA			ACL RD-TEC		
	Точность	Полнота	F1	Точность	Полнота	F1
Wang et al., 2016 [44]	0.35	0.59	0.44	0.71	0.68	0.69

1.2. Извлечение и классификация семантических отношений

1.2.1. Описание задачи извлечения и классификации отношений

Следующим этапом после извлечения именованных сущностей в системе извлечения информации является определение семантических отношений между извлечёнными сущностями. Задача состоит из двух этапов. На первом этапе определяется, связаны ли две

именованные сущности семантическим отношением. Если связаны, то на втором этапе определяется тип семантического отношения, которым связаны сущности.

Универсального набора семантических отношений не существует – обычно исследователи определяют типы рассматриваемых отношений, исходя из цели задачи, области знаний и жанра текста, с которым предстоит работать. В качестве примеров отношений можно привести следующие:

- Таксономические (отношение гипонимии/гиперонимии) – один из участников отношения является более общим понятием для другого, например: "*... такие [животные], как [собаки]*";
- Пространственные – один из участников отношения является пространственным понятием, локацией для другого, например: "*[Иван] живёт в [Москве]*.";
- Аффилиации – один из участников отношения является организацией, в которой работает второй участник, например: "*[Иван] работает в [НГУ]*";
- Родственные – двух участников этого отношения связывают родственные связи, которые могут быть как симметричными (например, "б*ыть супругом*"), так и несимметричными (например, "*быть родителем*") и др.

Обычно для этой задачи вводятся следующие ограничения. Как правило, отношение связывает только две сущности, при этом предполагается, что обе сущности упоминаются в тексте, отношения с количеством аргументов больше двух встречаются редко. Часто рассматриваются отношения между сущностями в пределах одного предложения – это облегчает как разметку данных, так и процесс создания алгоритма. Тем не менее, в последнее время всё большую актуальность приобретают работы, в которых отношения извлекаются из целого текста, не ограничиваясь предложениями.

Для семантических отношений действуют свойства алгебраических отношений:

1. Симметричность: $\forall\ x, y \in M: (xRy \Rightarrow yRx)$; пример: отношение "*быть супругом*";
2. Асимметричность: $\forall\ x, y \in M: (xRy \Rightarrow \neg(yRx))$, пример: отношение "*проживать в*";
3. Транзитивность: $\forall\ x, y, z \in M: (xRy \wedge yRz \Rightarrow xRz)$, пример: отношение "*быть предком*".

1.2.2. Методы и подходы к решению задачи

Традиционно задача извлечения отношений рассматривается как задача классификации.

Современные подходы для решения этой задачи предполагают использование различных нейросетевых архитектур. Например, в качестве базовой модели, которая не использует

никакую дополнительную информацию, авторы описали модель, основанную на архитектуре BERT [48]. Модель принимает на вход предложение, содержащее пару сущностей, для которой требуется определить тип отношения, маскируется специальным токеном, в котором содержится информация о типе сущности (например, название организации, географическое название и пр.), а в конце через специальные разделительные токены добавляются непосредственно сущности, которые были скрыты. Выходом модели является тип отношения для данной пары сущности. Но задача является довольно сложной для автоматической обработки, так как требует глубокого понимания не только синтаксической структуры предложения, но и семантической. Поэтому исследователи прибегают к различным улучшениям базового алгоритма.

Использование вручную извлечённых признаков. Естественным улучшением базового алгоритма является добавление различной информации. Например, в работе [49] предлагается использовать так называемые синтаксические индикаторы: изначально для каждого из отношений формируется словарь лексико-синтаксических маркеров, указывающих на то или иное отношение (например, "*moved into*" для отношения Destination, "*of*" для отношения Component-Whole и т.д.). На вход нейронной сети подают конкатенацию изначального предложения с индикаторной фразой ("e_1 *indicator* e_2, где e_1 и e_2 – пара сущностей, для которых нужно извлечь семантическое отношение, а *indicator* – лексико-синтаксический маркер"). В работе [50] используют информацию о синтаксических зависимостях для извлечения отношений. На первом шаге для двух заданных сущностей алгоритм находит кратчайший путь в дереве зависимостей (англ. shortest dependency path, SDP). Далее в классификации участвуют только те слова, которые входят в этот путь, их синтаксические роли в данном предложении, частеречная информация и типы сущностей, для которых определяется тип отношения. В статье [51] авторы используют информацию о синтаксической структуре предложения.

Использование баз знаний. Для того, чтобы учитывать информацию о семантике, очевидным шагом является использование источников дополнительной информации, которая хранится в онтологиях, базах знаний и др. Например, в работе [52] показано, как использовать векторы, полученные на данных из дополнительных лексических ресурсов (онтологий), в модели классификации отношений. Более того, авторы предлагают архитектуру, состоящую только из Attention-слоёв, что обеспечивает бОльшую производительность по сравнению с CNN и LSTM архитектурами. Авторы статьи [53] предлагают способ получения векторов, отражающих отношения между парами сущностей, чтобы потом использовать его при дообучении на конкретном корпусе текстов. Идея состоит в том, чтобы взять множество

текстов, в которых сущности связаны с уникальными идентификаторами из базы знаний. Факт отношения – это блок текста, содержащий две сущности. Таким образом создаётся обучающий набор данных, содержащий факты отношений, в которых сущности заменены специальным символом "*[BLANK]*". Задача обучения состоит в том, чтобы векторные представления отношений были похожими для одних и тех же пар сущностей. Полученные вектора можно использовать для дообучения.

Создание алгоритмов в условиях малого количества данных. Другая группа методов направлена на решение этой задачи в условиях малого количества обучающих данных. Одна из идей, которая находит применение в различных задачах обработки текстов, – это мультиязычные модели. Идея состоит в том, чтобы обучить модель на данных того языка, в котором они представлены в достаточном объёме, а затем применить её к языку, для которых данных меньше. Например, в работе [54] предлагается следующий подход. Задача классификации отношений обучается на данных для английского языка с использованием мультиязычной модели. Для предсказания отношений для текста на малоресурсном языке, вектора слов для входного предложения отображаются на вектора соответствующих английских слов, и модель делает предсказания на этих векторах. Другой подход, который успешно применяется в условиях малого количества размеченных данных, – это обучение модели на шумных данных. В работе [55] авторы используют синтаксический разбор предложения и предобученные векторные представления слов, чтобы извлекать небольшое количество отношений, но с высокой точностью, которые затем используются для автоматической разметки большого количества данных. Полученный набор данных используются для дообучения модели BERT для задачи извлечения отношений.

В последнее время активно публикуются работы, описывающие алгоритмы извлечения отношений из научных текстов различных областей, например: [56-58]. Как правило, для научных текстов используются те же самые методы и подходы, что применяются к текстам общей тематики. Отличие состоит только в наборе отношений, а также (в некоторых работах) в типах сущностей, которые присутствуют в текстах.

В таблице 4 приведено сравнение алгоритмов для извлечения семантических отношений на разных наборах данных (для английского языка). Можно заметить, что значения метрик сильно различается от корпуса к корпусу, и в целом, они значительно меньше, чем в задаче извлечения именованных сущностей. Это указывает на то, что извлечение отношение всё ещё остаётся сложной задачей для автоматических методов.

Таблица 4. Метрики извлечения отношений для английского языка

Источник	SemEval-2010 Task 8			NYT			TACRED		
	Точность	Полнота	F1	Точность	Полнота	F1	Точность	Полнота	F1
Tao et al., 2019 [49]	-	-	**0.90**	-	-	-	-	-	-
Nayak et al., 2019 [51]	-	-	-	0.54	0.59	0.56	-	-	-
Li et al., 2019 [52]	-	-	-	-	-	-	0.67	0.68	0.68
Soares et al., 2019 [53]	-	-	0.83	-	-	-	-	-	**0.70**
Shanchan et al., 2019 [59]	-	-	0.89	-	-	-	-	-	-

1.3 Задача одновременного извлечения именованных сущностей и отношений

1.3.1 Описание задачи

В предыдущих разделах задачи извлечения именованных сущностей и отношений между ними были описаны как две отдельные задачи: на первом этапе извлекаются сущности, а затем – отношения между ними. При этом при решении задачи извлечения отношений всегда подразумевается, что задача распознавания сущностей решена, и используется уже готовый

результат. Но очевидно, что эти две задачи тесно связаны: информация о том, какие семантические отношения присутствуют в данном тексте помогла бы лучше находить сущности; а агрегированная информация о сущностях в тексте повысила бы качество извлечения семантических отношений. Оказалось, что такая информация, действительно, помогает, и в последнее время особенно активно развиваются методы одновременного извлечения именованных сущностей и отношений между ними.

1.3.2 Методы и подходы к решению задачи

В работе [60] авторы предлагают архитектуру, которая последовательно извлекает сущности и отношения между ними, но в одном цикле обучения. Токены из входной строки кодируются векторными представлениями, полученные от предобученной модели BERT. Затем цепочки токенов (spans) классифицируются по типам сущностей. Те цепочки, которые не были распознаны как сущности, отфильтровываются. Затем все оставшиеся сущности (и их контексты – токены, находящиеся между конкретной парой сущности) попарно комбинируются и классифицируются по типам отношений. Данная архитектура показывает высокие результаты, но тем не менее авторы отмечают несколько её слабых мест: иногда модель предсказывает неверные границы сущностей; отношения, которые в явном виде не выражены в тексте (но логически их можно вывести) плохо обнаруживаются моделью; также, для пары сущностей, которая связана отношением, неверно выбирается тип отношения. Такая же архитектура, но с механизмом внимания была предложена в статье [61]. Авторы утверждают, что механизм внимания способен лучше накапливать семантическую информацию в векторных представлениях сущностей и их контекста. Похожая архитектура описана в работе [62]. Здесь авторы предлагают улучшить модель BERT, которая используются для кодирования входной последовательности. Оригинальная модель BERT была обучена на двух заданиях: предсказания случайно замаскированного токена в тексте и предсказание следующего предложения. Авторы обучили модель, добавив в качестве задания также предсказание предыдущего предложения. Также используется метод smooth labelling – метод регуляризации для задач классификации, позволяющий предотвратить слишком уверенное предсказание меток модели во время обучения и плохое обобщение. Его идея состоит в использовании вероятности того, что данная последовательность токенов является сущностей (в отличие от дискретных классов, которые используются традиционно). В статье [63] описана похожая концепция решения данной задачи,

но авторы также используют частеречную информацию и синтаксическую структуру предложения.

Другая архитектура представлена в работе [64]. Здесь авторы так же используют BERT для получения векторных представлений для входных токенов, а затем обучают модель с двумя выходами – одна часть модели обучается извлечению сущностей (задача sequence labelling), выходным слоем которой является CRF; а другая часть модели обучается определять тип отношения (но использует информацию, полученную из первой части модели).

Также есть попытки использовать обучение с подкреплением для решения этой задачи. Авторы статьи [65] применяют иерархическое обучение с подкреплением (англ. hierarchical reinforcement learning, HRL) для объединенной задачи извлечения сущностей и отношений. Процесс начинается с обнаружения отношений, который, в свою очередь, запускает извлечение объектов. Подобная архитектура усиливает взаимное влияние между упоминаниями сущностей и типами отношений.

В таблице 5 представлено сравнение нескольких алгоритмов из тех, что были описаны выше. Анализ результатов показывает, что решение одновременно двух задач: извлечения сущностей и отношений между ними, действительно, является обоснованным, и информация, полученная с каждого из этапов может помочь увеличить качество второй задачи. Но, по сравнению с метриками, полученными отдельно для каждой из задач, пока видится более перспективным решать эти две задачи в два отдельных этапа, разными архитектурами.

Таблица 5. Метрики совместного извлечения сущностей и отношений для английского языка

Источник	SciERC		CoNLL04	
	F1 Сущности	F1 Отношения	F1 Сущности	F1 Отношения
Eberts et al., 2020 [58]	0.70	0.51	0.86	0.73
Ji et al., 2020 [59]	-	-	**0.90**	**0.74**

1.4. Задача связывания сущностей

1.4.1. Описание задачи связывания сущностей

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. Под базой знаний (БЗ; англ. knowledge base, KB) понимается база данных, содержащая правила вывода и информацию о человеческом опыте и знаниях в некоторой предметной области. Задача связывания сущностей (англ. Entity Linking, EL) состоит в определении упоминания сущности в неструктурированном тексте и установлении связи с сущностью в структурированной базе знаний [66].

1.4.2. Методы и подходы к решению задачи связывания сущностей

Традиционно задача связывания сущностей делится на 4 этапа. Далее опишем подробнее каждый из шагов.

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 привёден в главе 1.

Этап 2: генерация кандидатов. На этом шаге создаётся краткий список возможных сущностей (кандидатов) для выделенного термина. Обычно такой список создаётся 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 [67] 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 [68] между сущностью и упоминанием

и косинусному расстоянию между вектором контекста и вектором сущности. В финальное множество кандидатов попадают k первых кандидатов. В статье [69] 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, [70]) прибегают к вычислению априорной вероятности совместной встречаемости сущности и упоминания в различных источниках: Википедия[1] (в заголовках страниц, в заголовках редиректов и в гиперссылках), в словаре, полученном на основе WebCorpus[2], и в словаре YAGO[3]. 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 – как на основании вручную сформированных признаков [71], так и полученных из языковых моделей [72]. При другом подходе задача ранжирования трансформируется в задачу бинарной классификации, в которой целью является определить, относится ли данное упоминание к сущности или нет. В качестве классификатора могут использоваться наивный байесовский классификатор [73], SVM (support vector machine) классификатор [74], глубокие нейронные сети [75]. 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 [76] авторы строят векторные представления рёбер графа, полученного из Dbpedia[4], с помощью алгоритма DeepWalk [77]. В работе [78] авторы используют алгоритм TransE [79] для векторизации сущностей в графе.

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

---

[1] https://ru.wikipedia.org/
[2] https://www.webcorp.org.uk/live/
[3] https://yago-knowledge.org/
[4] https://www.dbpedia.org/

Разработано множество готовых решений, которые поддерживают английский язык и классический набор сущностей (например, OpenTapioca [80]). Библиотека DeepPavlov[5], в свою очередь, имеет предобученные модели для русского языка. Алгоритм состоит из следующих компонентов:

1. Выделенная NER-моделью подстрока векторизуется (в качестве признаков используются значения TF-IDF) 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 Wikidata (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.

К сожалению, системы, разработанные для английского языка, сложно адаптировать для русского языка. Кроме того, система нуждается в настройке под конкретную предметную область – так, алгоритмы, предназначенные для связывания классического набора сущностей (персоны, названия организаций, географические наименования), зачастую оказываются неприменимы для работы с сущностями другого типа, например, терминами, как в данной работе.

[5] https://deeppavlov.ai/

Глава 2. Корпуса для решения поставленных задач

2.1 Существующие размеченные корпуса

Для извлечения информации из научных текстов существуют открытые наборы данных на английском языке с размеченными сущностями и/или отношениями. Ниже приведено описание некоторых таких корпусов.

STEM-ECR [81] – корпус, состоящий из аннотаций научных статей из 10 дисциплин, в которых были размечены сущности. Сущности выделялись 4 видов: процесс (например, "*наводнение*"), метод (например, "*магнитно-резонансная томография*", материал (например, "*почва*") и данные (например, "*вращательная энергия*").

SemEval-2018 [82] – корпус, состоящий из 500 аннотаций научных статей, в которых были размечены сущности и отношения между ними. Сущности выделялись в текстах безотносительно их типа. Разметка отношений содержит шесть типов семантических отношений: Usage (инструмент) – методы, задачи, данные могут быть связаны этим отношением (например, *подход – модель*); Result (результат) – сущность приводит к чему-то (например, *порядок – производительность*); Model (модель) – одна сущность является абстрактной моделью другой сущности (например, *категории – слова*); Part-Whole (часть-целое) – одна сущность является часть другой (например, *концепт – онтология*); Topic (тема) – отношение связывает научную работу с её темой или идеей (например, *статья – метод*); Comparison (сравнение) – одна сущность сравнивается с другой (например, *результат – стандарт*).

SCIERC [83] – данный датасет состоит из 500 научных аннотаций, которые содержат разметку научных сущностей и отношений между ними. Научные сущности были выделены шести типов: задача, метод, метрика, материал, другие научные термины и общие термины. Рассматривались семь типов отношений: сравнение, часть-целое, конъюнкция, оценка, признак, инструмент, гипоним.

SemEval 2017 [84] – этот корпус состоит из 500 публикаций в дисциплинах компьютерные науки, материаловедение и физика. Тексты размечены сущностями трёх видов: процесс, задача и материал, а также двумя типами отношений – гипонимия и синонимия.

Приведённые выше корпуса содержат тексты из разных научных областей, что делает их универсальными – типы сущностей и отношений не привязаны к конкретной дисциплине.

Но также существуют наборы данных, специфичные для какой-либо предметной области. Ниже приведены описания таких корпусов.

ACL RD-TEC 2.0 [85] – корпус, состоящий из аннотаций научных статей из области компьютерной лингвистики. Тексты размечены сущностями семи типов: технология и метод (например, “машинный перевод”), инструмент и библиотека (например, “OpenNLP”), языковой ресурс (например, “лексикон”), продукт языкового ресурса (например, “WordNet)”, модели (например, “языковая модель”), метрики (например, “BLEU”) и остальное (например, “орфографический вариант”).

The BioText Project [86] – датасет состоит из текстов статей из медицинской области, в которых размечены сущности двух типов: заболевания и лечения, а также семантические отношения 7 типов, которые являются специфичными для данной предметной области.

Таким образом, можно сделать вывод, что среди корпусов, содержащих размеченные тексты научных работ, можно выделить универсальные, которые могут быть использованы при решении задачи вне зависимости от конкретной предметной области, и специфичные для определённых дисциплин. Последние корпуса отличаются не только набором текстов из той или иной области, но также и определёнными типами сущностей и отношений, которые в этих текстах выделяются.

Еще раз повторим, что все датасеты, приведённые выше, представлены для английского языка. Для русского языка существуют размеченные тексты общей тематики с классическими типами сущностей: персоны, названия организаций и географические наименования: ***Persons-1000*** [87], ***FactRuEval*** [88], ***RuRED*** [28]. Но размеченных корпусов научных текстов для русского языка не удалось найти, поэтому было принято решение о создании корпуса для извлечения информации из научных текстов на русском языке.

2.2. Создание корпуса RuSERRC

2.2.1. Состав корпуса

Собранный в рамках данного исследования корпус состоит из аннотаций научных статей по теме информационные технологии, находящихся в открытом доступе [97]. Данные были взяты из журналов “Вестник НГУ. Серия: Информационные технологии”[6], “Программные продукты и системы”[7], “Cloud of science”[8], “Информационно-управляющие системы”[9].

[6] https://journals.nsu.ru/jit/
[7] http://www.swsys.ru/
[8] http://cloudofscience.ru/
[9] http://www.i-us.ru/index.php/ius

Объем корпуса составил 1 600 неразмеченных документов и 80 текстов, которые содержат вручную размеченные сущности и отношения между ними, а также ссылки на соответствующие сущности в Викиданных. Каждый документ был размечен двумя аннотаторами независимо, разногласия были разрешены модератором.

Более подробная инструкция разметки для всех трёх задач приведена ниже.

2.2.2. Описание разметки научных терминов

Задача разметки состояла в выделении терминов в текстах научных статей.

В качестве сущностей рассматриваются имена существительные и именные группы, являющиеся терминами в данной предметной области.

В разметке научными терминами считаются следующие фразы:

1. термины, состоящие из одного токена, в том числе, являющиеся аббревиатурой на русском языке (“*ПО*”, “*БД*”, “*бустинг*”, “*алгоритм*”);
2. названия языков программирования (“*Python*”, “*Kotlin*”, “*Java*”, “*C++*”);
3. названия библиотек (“*Pytorch*”, “*Keras*”, “*pymorphy2*”);
4. понятия, написанные через дефис и содержащие латинские символы (“*SPARQL-запрос*”, “*n-грамма*”, “*web-сервис*”, “*f-мера*”);
5. названия методов, архитектур, техник и др. на английском языке, в том числе аббревиатуры (“*zero-shot learning*”, “*long short-term memory*”, “*LSTM*”, “*text-to-speech*”, “*NLP*”).

Терминами не являются:

1. организации, локации, персоны, кроме тех случаев, когда они входят в состав термина (“*теорема Байеса*”, “*расстояние Левенштейна*” будут являться терминами);
2. цифровые комплексы, число + единица измерения и т.д. (“*10 км/час*” – не термин), если указана версия, то она входит в состав соответствующего термина (например, “*python 3.7*”);
3. URL;
4. даты;
5. любые идентификаторы.

Если термин написан с ошибкой или опечаткой, то все равно его следует выделить.

Если сущности перечислены через "*,*" "*;*" или соединены союзом "*и*", то по возможности следует выделять их раздельно, например во фразе “*...был проведён синтаксический*

и семантический анализ текста” нужно выделить два термина: “*синтаксический*” и “*семантический анализ*”.

Не считаются терминами общеупотребимые многозначные слова, такие как “*решение задачи*”, “*данные*”, “*запись*” и т.д.

Особую сложность представляет выделение многословных терминов. Многословным термином считается цепочка токенов максимальной длины, которая при отбрасывании токенов преобразуется в более общий термин. Как правило, это названия программных продуктов, методов, алгоритмов, задач, подходов (“*Quantum GIS*”, “*операционная система Android*”, “*метод k ближайших соседей*”, “*метод SPH*”, “*метод опорных векторов*”). Так например, должен быть выделен составной термин “*модель структурной организации единого информационного пространства*”. Просто слово “*модель*” не несет полной информации об особенностях метода, поэтому не хотим пропустить полное название метода и выделяем такой составной термин.

Другие примеры составных терминов: “*формальная модель процесса*”, “*транзакционная модель*”, “*математическая модель упругопластических сред*”, “*теоретико-модельный подход*”, “*задача геонавигации*”, “*информационно-аналитическая система*”.

Границы таких сущностей обычно задаются контекстом и довольно часто неоднозначны. Чтобы учесть эту неоднозначность, в составном термине может быть выделен вложенный термин, когда это целесообразно. Например, в составе сущности “*численное моделирование динамики пучков*” может быть выделена вложенная сущность “*численное моделирование*”.

Следует учитывать, что если вложенных сущностей несколько, то их границы не должны пересекаться. Если части большой сущности не встречаются в конкретном тексте как самостоятельные смысловые единицы, то они не считаются вложенными сущностями в этом тексте.

Рассмотрим следующий пример. Фраза “*информационная система представления результатов комплексного анализа поэтических текстов*” считается термином. Если рассмотреть один из уровней вложенности, то правильно будет выделить “*анализа поэтических текстов*” и “*система*”. Слово “*система*” встречается в тексте для отсылки к создаваемой системе, которая названа длинной сущностью, поэтому оно тоже может быть выделено как вложенная сущность. На следующем уровне вложенности в длинном термине можно выделить такие сущности: “*информационная система*”, "*поэтические тексты*”, т.к. они встречаются в тексте самостоятельно.

Кавычки и скобки не рекомендуется выделять как часть сущности. Например, если в тексте встретилось название системы, записанное в кавычках, то сущностью считается только само название без кавычек (Например, “*Ixodes*”). Однако, вместе с этим, в зависимости от контекста, может быть выделена составная сущность (например, “*информационно-аналитическая система "Ixodes"*”), в которую неизбежно попадут кавычки. Тогда в данном примере сущность “*Ixodes*” будет являться вложенной.

В научных текстах часто встречаются отглагольные существительные, которые обозначают процессы (*анализ – анализировать*, *исследование – исследовать*, *создание – создать* и т.д.). С точки зрения семантики, процесс приводит к изменениям, влияет на результат. Такие существительные желательно включать в состав сущности, их включение влияет на правильность определения отношений. Примеры: “*обработка изображений*”, “*тестирование системы*”, “*анализ текста*” и др.

Отдельно рассмотрим случай с предлогом “*для*”. Часто этот предлог употребляется в значении “*используется для*” или “*применяется для*”, а значит, между сущностями, которые он соединяет, есть отношение USAGE (см. п.2.2.3). Поэтому предлог “*для*” не должен включаться в сущность. Пример: “*современные [портативные приборы] для [электромагнитного профилирования] и [малоглубинного зондирования]*”.

Поскольку информационные технологии применяются для решения большого круга задач в разных областях, то в текстах в качестве сущностей могут быть выделены разновидности данных (например, “*ЭЭГ-данные*”, “*гамма-каротаж*” и пр.), а также могут считаться терминами понятия из других предметных областей, если они непосредственно связаны с постановкой задачи или ее решением (например, “*спектр гамма-излучения*”, “*метроритмическая характеристика*”, “*генетическая последовательность*”).

Разметка сущностей выполняется 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 сущности).

Процент согласия аннотаторов в задаче выделения сущности составил 51.77%, что показывает высокую степень субъективности при нахождении слов и фраз, являющихся терминами, а также при определении точных границ сущности. Значение было вычислено как отношение пересечения выделенных терминов к объединению выделенных терминов.

2.2.3. Описание разметки отношений между научными терминами

Задача разметки состояла в определении типа семантического отношения между двумя уже выделенными сущностями в пределах одного предложения.

Если в предложении перечислены несколько однородных сущностей, которые семантически связаны с другой сущностью, то в каждой такой паре следует указывать отношение. Одна сущность может участвовать в нескольких отношениях одновременно.

Так как для работы были выбраны научные тексты из области информационных технологий, то классический набор отношений не подходил. Более того, из-за того, что были выбраны не все научные тексты, а тексты конкретной области, набор отношений также стоило пересмотреть, чтобы выбрать релевантные типы. Для выбора набора отношений были проанализированы несколько работ из разных областей знаний.

Один из наборов отношений между сущностями был предложен на соревновании SemEval-2010 [89]. Он состоит из 10 семантических отношений, являющихся общими для текстов всех отраслей и жанров:

1. Причина-Эффект (Cause-Effect) – объект или событие приводит к некоторому результату, например: “*эти виды [рака]$_{Entity2}$ были вызваны радиационным [воздействием]$_{Entity1}$*”;
2. Инструмент-Деятель (Instrument-Agency) – агент использует инструмент, например: “[*телефонный]$_{Entity1}$ [оператор]$_{Entity2}$*”;
3. Продукт-Производитель (Product-Producer) – производитель производит продукт, например: “[*фабрика]$_{Entity2}$ производит [костюмы]$_{Entity1}$*”;
4. Контент-Контейнер (Content-Container) – объект физически хранится на очерченном участке пространства, например: “*была взвешена [бутылка]$_{Entity2}$, полная [мёда]$_{Entity1}$*”;
5. Сущность-Происхождение (Entity-Origin) – сущность происходит из чего-либо (позиции или материала), например: “[*письма]$_{Entity1}$ из зарубежных [стран]$_{Entity2}$*”;
6. Сущность-Назначение (Entity-Destination) – сущность движется по направлению к точке назначения, например: “[*мальчик]$_{Entity1}$ пошёл в [кровать]$_{Entity2}$*”;
7. Часть-Целое (Component-Whole) – объект является частью чего-либо целого, например: “*в моей [квартире]$_{Entity2}$ есть большая [кухня]$_{Entity1}$*”;
8. Элемент-Коллекция (Member-Collection) – элемент образует часть коллекции, например: “*в [лесу]$_{Entity2}$ много [деревьев]$_{Entity1}$*”;
9. Сообщение-Тема (Message-Topic) – сообщение, устное или письменное, на определённую тему, например: “[*лекция]$_{Entity1}$ была о [семантике]$_{Entity2}$*”.

При более детальном анализе этих отношений становится ясно, что не все из них будут частотными в научных текстах. Например, в текстах научных статей, как правило, не упоминаются конкретные люди, деятели (или агенты, в терминах семантики), поэтому отношение Инструмент-Деятель не подходит. Отношения Контент-Контейнер, Сущность-Происхождение, Сущность-Назначение подразумевают физическое присутствие объекта (не всегда, конечно, но как правило), но в текстах из областей информационных технологий и математики речь обычно идёт об абстрактных сущностях, к которым сложно применить подобные отношения. Сущности типа “Сообщение” и “Тема” также редко встречаются в текстах подобного жанра.

В статье [82] описан набор отношений для работы с текстами научных статей, предложенный в рамках соревнования SemEval-2018. Он включает следующие семантические отношения:

1. USAGE – данное отношение связывает методы, задачи и данные (например, “*система машинного перевода – японский язык*”);
2. RESULT – данное отношение имеет место, когда одна сущность влияет на другую или приводит к какому-либо результату (например, “*парсер – эффективность*”);
3. MODEL – данное отношение имеет место, когда сущность является аналитической характеристикой или абстрактной моделью другой сущности (например, “*категории – слова*”);
4. PART-WHOLE – отношение часть-целое (например, “*фраза – текст*”);
5. TOPIC – данное отношение связывает научную работу с её темой (например, “*статья – тема*”);
6. COMPARISON – одна сущность сравнивается с другой сущностью (например, “*результат – стандарт*”).

Классы отношений были выбраны в результате анализа работ [82], [89] на основе следующих критериев:

1. Отношение должно толковаться однозначно (например, в данной работе не рассматривается семантическое отношение Entity-Destination, т.к. оно имеет также косвенное значение);
2. Отношение должно быть способно связывать между собой научные термины (например, актантами семантического отношения Communication-Topic (акт коммуникации на какую-либо тему) выступают не научные термины, поэтому такое отношение не подходит).

Таким образом, были выбраны семь семантических отношений: CAUSE, COMPARE, ISA, PARTOF, SYNONYMS, TOOL, USAGE.

Подробное описание каждого из них приведено в Таблице 6, примеры сущностей, связанных отношениями, представлены на Рисунке 1.

Таблица 6. Описание отношений

Отношение	**Пояснение**	**Примеры**
CAUSE	Причинно-следственное отношение: X **вызывает** Y; X **является причиной** Y; X **приводит к результату** Y X **даёт в результате** Y. Объект или событие дает некоторый результат.	(взаимодействие высокоэнергетичных пучков : деформация) (научные исследования : данные) (обработка исходной информации : переход к нереляционным БД)
COMPARE	Сравнение: X **в сравнении с** Y; X **больше/меньше** Y; X **лучше/хуже** Y. Является противоположным отношению COORDINATE. Следует выделять это отношение, только если в тексте есть явное указание на сравнение.	(модули : ручная обработка данных) (реляционные базы данных : объектно-ориентированные) (ООСУБД : СУБД)
ISA	Таксономическое отношение, отношение наследования, родо-видовое отношение, отношение между объектом и множеством, обозначающим, что объект принадлежит этому множеству: X **– это** Y.	(жанр : высокоуровневые характеристики) (импульсный нейтронный гамма-каротаж : метод) (Python : язык программирования)

PART_OF	Отношение часть-целое, меронимы: X **является частью** Y. Объект X является составляющей большего целого Y. Обратное – холонимы, тогда аргументы меняются местами: Y **содержит** X; Y **состоит из** X.	(спектры гамма-излучения : сигнал) (приложение : Android) (модуль : система)
SYNONYMS	Отношение синонимии. Синонимы – слова одной части речи с полным или частичным совпадением значения. Часто это отношение связывает полное название с аббревиатурой или переводом, который приводится рядом в скобках.	(GPU : графический процессор) (CNN : сверточная нейронная сеть)
TOOL	Отношение описывает конкретную возможность использования, иногда гипотетическую: X **позволяет** - **создавать** Y, - **анализировать** Y, - **изучать** Y, - **исследовать** Y, - **сравнивать** Y, - **решать** Y, - **выполнять** Y, - **вычислить** Y, - **автоматизировать** Y, - **управлять** Y.	(портативные приборы : распределение удельного электрического сопротивления грунта) (методы нелинейной динамики : хаотичность сигналов-переносчиков защищённых систем связи)

USAGE	Отношение использования в общем случае, без дополнительной конкретизации; X **используется для/в** Y; X **применяется для/в** Y В состав Y могут входить следующие слова: - создания, - анализа, - изучения, - исследования, - сравнения, - решения, - выполнения, - вычисления. Аргументы меняются местами, если отношение обратное: Y **выполняется с помощью** X.	(система : анализ генетического разнообразия) (WEB-технологии : программное обеспечение) (метод статистической обработки : анализ текстов) (трехмерное моделирование : модели сред)

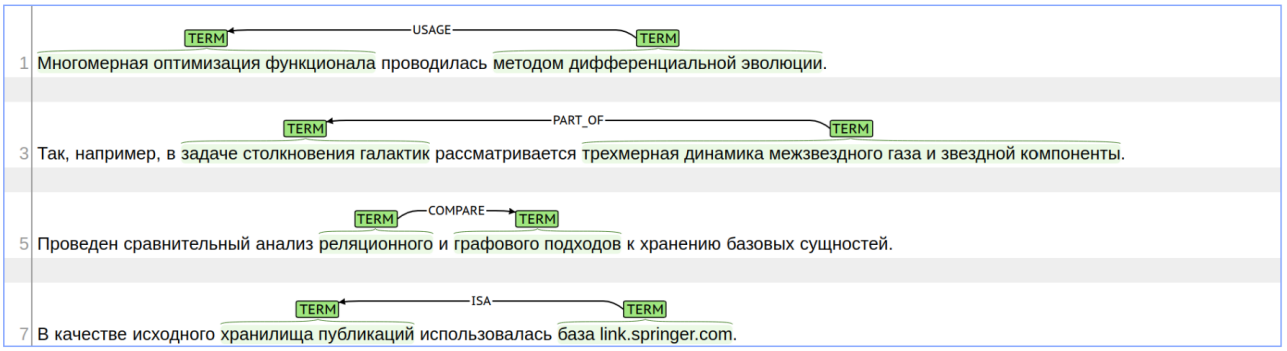


Рисунок 1. Пример разметки семантических отношений

Следует обратить внимание на различия между отношениями USAGE и TOOL. Отношение TOOL выражается такими глаголами, как “*изучать*”, “*решать*”, “*анализировать*”, “*создавать*” и т.д. В отличие от него, отношение USAGE характеризуется глаголами “*использовать*” и “*применять*”, к которым присоединяются сущности.

Глагол, характеризующий отношение (например, “*применяется*”) и существительное, которое вступает в отношение, но при этом характеризует процесс (например, “*применение чего-то*”) имеют схожую семантику. Как правило, глагол соответствует отношению, а отглагольное существительное является сущностью или ее частью. Поэтому для этих двух случаев предусмотрены разные отношения.

Рассмотрим конкретные примеры:

1. Дано предложение “*Современные [портативные приборы] для [электромагнитного профилирования] и [малоглубинного зондирования] позволяют изучать [распределение удельного электрического сопротивления грунта]*”. В этом предложении можно выделить следующие отношения: USAGE (“*портативные приборы*” – “*электромагнитное профилирование*”); USAGE (“*портативные приборы*” – “*малоглубинное зондирование*”); TOOL (“*портативные приборы*” – “*распределение удельного электрического сопротивления грунта*”).

2. Дано предложение “*Для эффективного использования аппаратурного комплекса было создано [программное обеспечение QZond], позволяющее управлять [процессом сбора информации]*”. Здесь можно выделить отношение TOOL (“*программное обеспечение QZond*” – “*процесс сбора информации*”).

3. Дано предложение “*Применение [методов нелинейной динамики] для [исследования хаотичности сигналов-переносчиков защищённых систем связи] на основе динамического хаоса*”. Здесь можно выделить отношение USAGE (“*методы нелинейной динамики*” – “*исследования хаотичности сигналов-переносчиков защищённых систем связи*”).

4. В части предложения “*...использование [мобильных устройств] для [тестирования]*” выделяется отношение USAGE (“*мобильные устройства*” – “*тестирование*”).

5. В части предложения “*[Трехмерное моделирование] и инверсия данных комплекса методов электрокаротажа в [моделях сред]...*” можно выделить отношение USAGE (“*трехмерное моделирование*” – “*модели сред*”).

2.2.4 Описание разметки связывания сущностей

Описанный выше корпус был дополнен разметкой, в которой выделенные термины связаны с сущностями в Викиданных[10]. 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:

[10] https://www.wikidata.org/wiki/

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.

На первом шаге существующая разметка терминов была дополнена вложенными сущностями, например: “[*самосогласованное* [*электрическое поле*]]”. Разметка связывания сущностей осуществлялась 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 допускались следующие видоизменения сущностей.

1. Все извлечённые сущности ищутся в базе знаний в нормализованной форме с учётом согласования и без учёта регистра, например: “*Линейных уравнений*” → “*линейное уравнени*е”.
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 искалась 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 считались синонимами термины “*подход*” и “*метод*”.

Связывание терминов выполнялось только с сущностями в Википедии – эти сущности имеют идентификатор с префиксом “*Q*”, в отличие от отношений, которые имеют идентификатор с префиксом “*P*”. Также не связывались 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 %.

2.3 Выводы

Таким образом, были проанализированы наборы данных, в которых имеется разметка сущностей и отношений между ними. Для английского языка такие данные имеются для научных текстов, в то время как для русского языка присутствуют корпуса только для текстов общей тематики (как следствие, в которых размечены сущности традиционных типов, и общие типы семантических отношений). Предложена инструкция по разметке такой информации в текстах научных статей на русском языке и выполнена сама разметка.

Таким образом, вручную были размечены 80 аннотаций научных статей по теме “Информационные технологии”. Каждый текст размечался независимо двумя аннотаторами, спорные случаи разрешались третьим аннотатором.

Всего в 80 размеченных текстах содержатся 11 157 токенов и 2 047 терминов. Средняя длина термина – 2.43 слова. Самый длинный термин состоит из 11 токенов. Процент согласия аннотаторов в задаче выделения сущности составил 51.77%, что показывает высокую степень субъективности при нахождении слов и фраз, являющихся терминами, а также при определении точных границ сущности. Значение было вычислено как отношение пересечения выделенных терминов к объединению выделенных терминов.

Отношения между сущностями выделялись только в границах одного предложения, ограничения на количество отношений, в которые может вступать одна сущность, не накладывались. Всего в размеченной части корпуса было выделено 604 отношений между сущностями, из них CAUSE – 19, COMPARE – 9, ISA – 95, PARTOF – 90, SYNONYMS – 22,

TOOL – 38, USAGE – 331. Больше половины составляют отношения использования (54.8%), на втором месте таксономические отношения (15.7%).

Всего в корпусе выделено 3386 терминов (с учётом вложенных сущностей), 1337 из которых удалось связать с сущностями в Викиданных. Средняя длина связанной сущности – 1,55 токен, 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 отдавалось 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 не связывались.

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 идёт речь в статье), но вне контекста термин "*комплексный анализ*" означает "*раздел математического анализа, в котором рассматриваются и изучаются функции комплексного аргумента*".

Пример текста аннотации научной статьи с разметкой терминов, отношений между ними и связанных сущностей приведён в Приложении 1. В столбцах содержится следующая информация:

1. *id* – порядковый номер токена в данном тексте;
2. *token* – словоформа;
3. *nested_n* – метки, определяющие термины в тексте на *n* разных уровнях вложенности (здесь $n = 3$);
4. *wiki_id* – идентификатор сущности в базе Викиданные; через запятую указаны номера токенов, которые составляют термин для связанной сущности;
5. *relation* – тип отношения, в котором находится токен; в скобках указан номер зависимого токена (важно для несимметричных отношений, например, PART_OF, ISA, USAGE).

Глава 3. Извлечение научных терминов

3.1 Формальная постановка задачи

Как было сказано выше, задача извлечения именованных сущностей не ограничивается извлечением фраз, содержащих имена людей, названия организаций и географические наименования. В зависимости от области изменяется понимание именованной сущности. Когда речь заходит о научных областях, задача извлечения сущностей преобразуется в задачу извлечения терминов.

В данной работе под термином понимается слово или словосочетание, являющееся названием некоторого понятия какой-нибудь области науки, техники, искусства и др. [90].

Назовем токеном x_i – слово или знак препинания в тексте. Рассмотрим последовательность всех токенов $X = \{x_0, x_1, ..., x_n\}$ и множество меток $Y = \{B\text{–}TERM,\ I\text{–}TERM,\ O\}$, где $B\text{–}TERM$ – метка для токена, который занимает первую позицию в термине, $I\text{–}TERM$ – метка для токена, который занимает вторую и последующие позиции в термине, O – метка для токена, который не входит в состав термина.

Требуется построить классификатор, который произвольной входной последовательности токенов ставит в соответствие последовательность меток, т.е. $\varphi: X \to Y$.

3.2 Описание методов

3.2.1 Метод на основе словаря

В качестве базового алгоритма мы реализовали 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.

Назовём словарём D конечное множество строк, которые являются терминами: $D = \{T_0, T_1, ..., T_n\}$. Каждый термин T_i в свою очередь представляет собой последовательность символов, длиной больше нуля. Определим множество символов $C = \{Cyr, Lat, Punct, Digits, whitespace\}$, где

Cyr – множество всех кириллических символов;

Lat – множество всех латинских символов;

$Punct$ – множество всех знаков препинания;

$Digits$ – множество всех цифр;

$whitespace$ – знак пробела.

Для произвольной последовательности символов $S = \{c_{0,} c_1, ..., c_m\}$ требуется найти подпоследовательности символов, входящие в словарь D.

Для этого построим детерминированный конечный автомат A, для которого определены следующие множества:

V – входной алфавит, равный множеству C;

Q – множество внутренних состояний;

q_o – начальное состояние автомата ($q_o \in Q$);

F – множество конечных состояний;

σ – функция переходов $Q \times V \rightarrow Q$.

Начальное состояние соответствует состоянию, при котором распознавание термина не начато. Конечный автомат начинает работу в начальном состоянии q_0, последовательно получает по одному символу из входной последовательности символов. Считанный символ переводит автомат в новое состояние или оставляет его в начальном состоянии в соответствии с функцией переходов. Если после получения последнего символа автомат оказался в состоянии, которое принадлежит множеству конечных состояний F, то цепочка символов считается распознанной и, соответственно, является термином. Если распознано несколько цепочек, то выбирается цепочка с максимальной длиной.

Рассмотрим словарь, например, состоящий из четырёх терминов: “*матрица*”, “*матрица Якоби*”, “*матроид*”, “*мейоз*”. Префиксное дерево для данного словаря представлено на Рисунке 2. Красным отмечены конечные состояния, буквосочетание *ws* обозначает знак пробела.

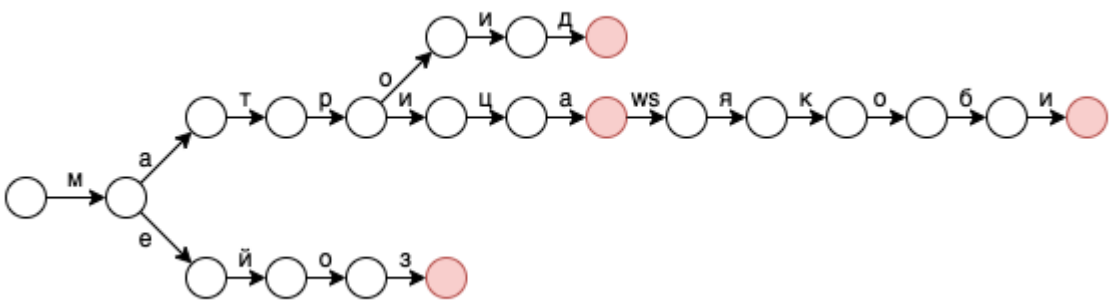


Рисунок 2. Префиксное дерево

В соответствие с данным префиксным деревом можно построить матрицу переходов (Приложение 2). В данной матрице состояние p_0 является начальным, состояния p_5, p_{11}, p_{15}, p_{22} являются конечными, т.е. принадлежат множеству F. Красным цветом для наглядности выделены конечные состояния.

Формирование словаря терминов *D* осуществляется в полуавтоматическом режиме двумя способами:

1. Из текстов научных статей извлекаются 2-, 3- и 4-граммы слов, отсортированные по значению tf-idf, затем из них вручную выбираются те фразы, которые потенциально могут быть терминами.

2. Из заголовков статей из Википедии, входящих в подграф категории “Наука”, вручную выбираются те слова и фразы, которые потенциально могут быть терминами.

Таким образом, был получен словарь из 17252 терминов:

- 1-граммы (6509 терминов): “*этнолингвистика*”, “*равносоставленность*”, “*пластида*” и др.;
- 2-граммы (6785 терминов): “*нокдаун гена*”, “*математическое ожидание*” и др.
- 3-граммы (2322 терминов): “*уравнение переноса излучений*”, “*метод анализа иерархий*” и др.;
- 4-граммы (1098 терминов): “*теорема Бертрана о выборах*”; “*стабилизированный метод бисопряжённых градиентов*” и др.;
- 5-граммы (348 терминов): “*удалённый доступ к памяти*”, “*теория реакционной способности химических соединений*” и др.;
- 6-граммы (120 терминов): “*теорема Римана об условно сходящихся рядах*”, “*задача о назначении минимального количества исполнителей*” и др.;

- 7-граммы (40 терминов): "*теорема о свойстве Дарбу для непрерывной функции*" и др.;
- 8-граммы (22 термина): "*теорема Пуанкаре о разложении интегралов по малому параметру*" и др.;
- 9-граммы (7 терминов): "*теорема Стоуна о группах унитарных операторов в гильбертовом пространстве*" и др.
- 12-граммы (1 термин): "*автоматический выключатель, управляемый дифференциальным током, со встроенной защитой от сверхтока*".

Основная сложность составления словаря терминов заключается в том, что без контекста сложно понять, является ли фраза термином или нет. Более того, в разных контекстах одна и та же фраза может быть как термином, так и нет: например, "*модель"*, "*текст*", "*язык*" и др.

При реализации словарного подхода были использованы библиотеки NLTK[11] для токенизации, pymorphy2[12] для лемматизации и ahocorapy[13] для построения префиксного дерева и работы с ним.

Результаты, полученные с применением данного подхода, представлены в Таблице 9.

3.2.2 Метод RAKE

Для того, чтобы сделать выводы о качестве полученной системы, были проведены эксперименты с инструментом RAKE, основанном на статистическом подходе. Rapid automatic keyword extraction (RAKE) – алгоритм, предназначенный для автоматического извлечения ключевых слов [91]. 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 встречается, 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 алгоритм хорошо применим к динамическим корпусам документов и к абсолютно новым доменам, при этом не зависит от языка и его особенностей.

Использовалась реализация на языке Python[14], которая поддерживает работу с русским языком и автоматическое выделение стоп-слов из текста. Следует отметить, что при

[11] https://pypi.org/project/nltk/
[12] https://pypi.org/project/pymorphy2/
[13] https://pypi.org/project/ahocorapy/
[14] https://pypi.org/project/rake/

использовании заранее подготовленного списка стоп-слов из открытых источников сети интернет алгоритм показывает лучшие результаты.

Было замечено, что зачастую алгоритм добавляет в ключевые фразы словосочетания, содержащие глагольные формы, например, “посвящена разработке нового программного обеспечения” или “решения задач геонавигации используются алгоритмы”. Так как в качестве сущностей рассматривались существительные и именные группы, было решено выполнить предобработку текстов и убрать глаголы и их формы перед применением RAKE. Для этого использовалась обертка на Python для инструмента Mystem[15] от компании Яндекс . При помощи Mystem в тексте выделялись все глагольные формы и принудительно удалялись из него.

Результаты, полученные с применением данного подхода, представлены в Таблице 9.

3.2.3 Метод на основе машинного обучения

Сложность проведения экспериментов с использованием различных алгоритмов машинного обучения заключается в отсутствии размеченных данных. В данной работе 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 разметили тексты терминами из словаря, описанного в предыдущем разделе. Таким образом, у нас получился размеченный набор данных, общим объёмом 1 992 498 токенов и содержащий из 177 050 терминов.

Предполагается, что обобщающая способность модели позволит находить термины в аннотациях текстов, где, предположительно, концентрация терминов выше, в то время как, модель была обучена на полных текстах статей, в которых концентрация терминов ниже. Также, таким способом, система сможет находить термины в текстах, которые отсутствовали в исходном словаре.

Входная последовательность кодировалась посимвольно, one-hot способом. Входная последовательность представляет собой множество токенов $T = \{t_0, t_1, ..., t_n\}$. Назовём $Chars$ взаимооднозначное отображение возможных символов, состоящих из кириллических, латинских символов, некоторых пунктуационных знаков и символа, обозначающего все остальные символы, в последовательные натуральные числа, включая 0. Закодируем каждый токен из входной последовательности в вектор V, длина которого равна количеству возможных символов. Используя данное отображение для каждого элемента мы получаем определённое

[15] https://pypi.org/project/pymystem3/

числовое значение n. Элемент вектора V, который находится на позиции, соответствующей значению n, равен 1, все остальные элементы равны 0.

Таким образом получается матрица M размером $|Chars| \times maxTokenLength \times MaxSequenceLength$, где $maxTokenLength$ – это максимальное количество символов, из которого может состоять токен (в экспериментах значение равно 30), $maxSequenceLength$ – это максимальное количество токенов, из которого может состоять последовательность (в экспериментах значение равно 100), которая подаётся на вход модели. Все значения были выбраны в результате анализа данных их обучающего множества.

Модель состоит из слоя двунаправленной LSTM, затем – CRF для формирования выходной последовательности тэгов (Рисунок 3).

При реализации алгоритма были использованы следующие библиотеки и инструменты: NLTK[16] для токенизации, Tensorflow[17] для построения и обучения модели.

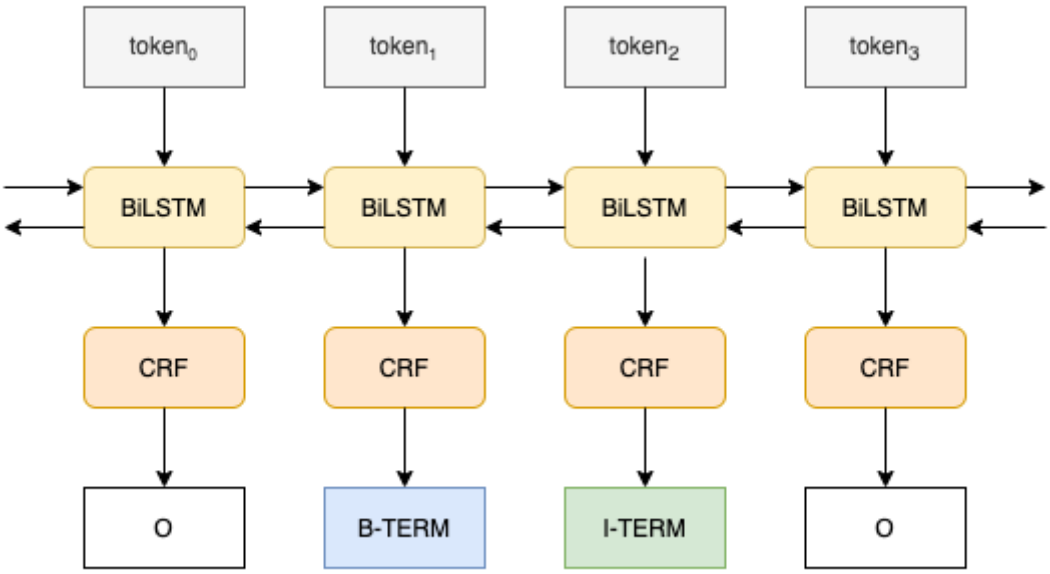


Рисунок 3. Архитектура посимвольной модели

Полученные метрики приведены в Таблице 9. Можно назвать следующие причины низких метрик.

Во-первых, в исходной разметке не было примеров с перечислением (частей) терминов. Например, в размеченном предложении "*Каждый компонент проекта относится к одному*

[16] https://www.nltk.org/
[17] https://pypi.org/project/tensorflow/

из структурных уровней анализа текста: структурный, семантический, прагматический" токены "*структурный*", "*семантический*", "*прагматический*" являются терминами, в то время как в исходном словаре терминов, который использовался для автоматической разметки, содержатся только фразы, которые представляют собой целостные термины, т.е. являются полноценной именной группой.

Во-вторых, модель часто не относит слова "*алгоритм*", "*метод*" и др. к составу термина, если далее идёт уточнение алгоритма, например: фраза "*алгоритм численного моделирования цунами*" в разметке является полностью термином, модель же отнесла к термину только часть фразы "*численного моделирования цунами*".

Также был проведён анализ того, сколько новых терминов, не содержащихся в исходном словаре, который использовали для разметки, смогла извлечь модель. Из всех уникальных, правильно извлечённых терминов, 26.5% составляют термины, которые модель ранее не видела, что подтверждает нашу гипотезу.

3.2.4 Метод на основе слабо контролируемого обучения (weak supervision)

Метод, описанный выше в п.3.2.3 показал, что подход, предполагающий автоматическую разметка текстов с помощью словаря, а затем обучение на полученном наборе данных, является обоснованным. Поэтому алгоритм был изменён для того, чтобы он был способен находить больше разнообразных терминов.

Для этого было использовано слабо контролируемое обучение (англ. weak supervision) – метод, в котором шумные или неточные источники данных используются для разметки большого количества обучающих данных, которые затем будут использованы для обучения модели традиционным способом с учителем (англ. supervised learning) [92].

Идея 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 [95].

Таким образом алгоритм получения модели для извлечения терминов состоит из следующих шагов:

1. Получить размеченный корпус для первой итерации обучения модели;
2. Обучить модель на полученном корпусе из п.1;
3. Разметить новые тексты моделью, полученной в результате выполнения п.2;
4. Обучить модель на корпусе текстов из п.1 + из п.3.

Рассмотрим каждый из шагов более детально.

При ***получении размеченного корпуса для первой итерации обучения модели*** применялся словарный подход, который был подробно описан в п. 3.2.1. Таким образом было размечено 1118 текстов научных статей из журнала “Программные системы и продукты”, которые использовались в качестве обучающего множества для модели в первой итерации (Bert-iter_1).

Получение размеченного корпуса для второй итерации обучения модели. На этом шаге были взяты 808 аннотаций научных статей из журналов “Cloud of science”, “Программные системы и вычислительные методы”, “Информационно-управляющие системы” и размечены комбинированной моделью. Данная модель (Bert-iter_2) представляет собой комбинацию словарного метода, с помощью которого были размечены тексты в предыдущем пункте, и модели, полученной после первой итерации обучения.

Обучающим множеством для модели второй итерации стало объединение текстов первой итерации и новых размеченных текстов.

Основная гипотеза при использовании предсказаний модели для разметки текстов состоит в том, что обобщающая способность модели позволяет не только выделять термины, которые уже содержатся в словаре, но находить новые паттерны, и соответственно, новые слова и фразы, которые являются терминами.

Описание модели. Для получения векторных представлений слов была использована предобученная модель BERT bert-base-multilingual-cased[18]. 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:

1. BERT-LSTM: полученные векторные представления подавали на вход двунаправленной LSTM, за которой шли два полносвязных слоя (архитектура модели представлена на Рисунке 4);
2. BertForTokenClassification: после векторных представлений идёт один полносвязный слой (архитектура модели представлена на Рисунке 5).

[18] https://huggingface.co/bert-base-multilingual-cased

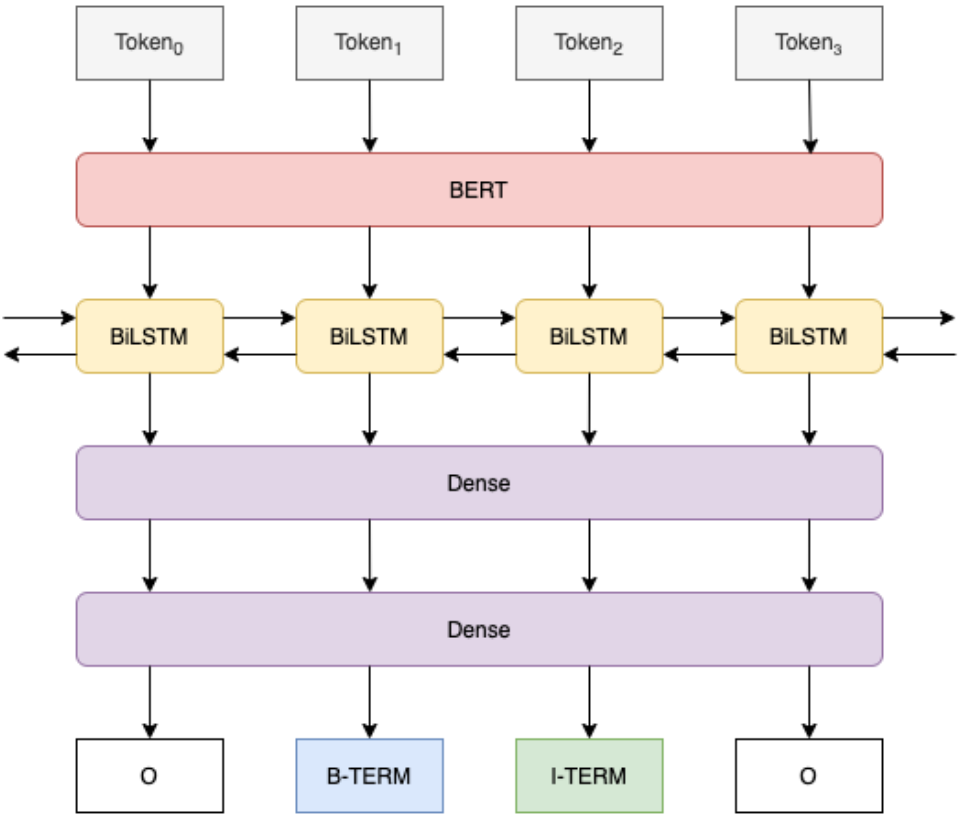


Рисунок 4. Архитектура модели BERT + LSTM

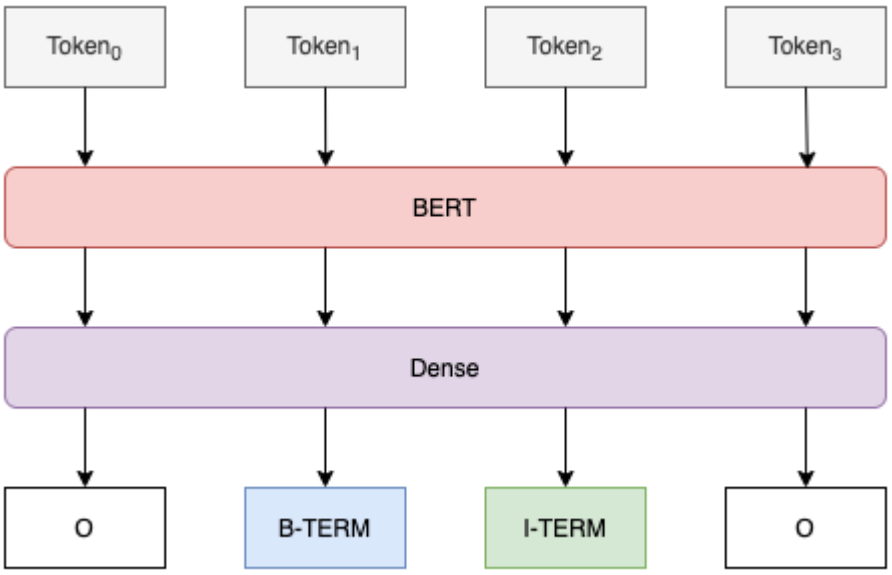


Рисунок 5. Архитектура модели BertForTokenClassification

Описание эвристик. Для улучшения качества извлечения терминов, были написали эвристики для валидации границ терминов. Ниже приводится описание используемых эвристик.

Эвристика 1: если токен (1) распознан как термин и является именем существительным или именем прилагательным, и следующий токен (2) распознан как не термин и имеет форму родительного падежа, то токену (2) присваивается тэг “I-TERM”. Примеры последовательностей: “методы сжатия (1) данных (2)”, “реляционных баз (1) данных (2)”, “системы (1) проверки (2) заданий” и др.

Эвристика 2: если токен (1) распознан как “B-TERM” и является именем прилагательным, и токен (2) распознан как “B-TERM” и является именем существительным, то меняем тэг токена (2) на “I-TERM”. Примеры последовательностей: “в учебном (1) процессе (2)”, “нечёткая (1) модель (2)”, “физические (1) величины (2)” и др.

Эвристика 3: если последний токен (1) в цепочке токенов, образующих термин, имеет часть речи имя прилагательное, а следующий за ним токен имеет часть речи имя существительное (2), то токену (2) присваивается тэг “I-TERM”. Примеры последовательностей: “возможности мобильных (1) устройств (2)” и др.

Эвристика 4: если токен (1) входит в состав термина, а следующий за ним токен состоит только из латинских символов, то включаем его в состав термина. Пример последовательностей: “в пакете (1) MOST (2)”, “по базе данных (1) Cistrome (2)” и др.

Также в явном виде запрещалось помечать тэгами терминов следующие классы токенов:

1. знаки пунктуации: “.”, “,”, “:”, “;”;
2. предлоги и союзы, если они имеют тэг “B-TERM” (допускается появление предлогов и союзов в составе термина, но не допускается то, что данные части речи начинают термин);
3. однозначные глаголы и деепричастия (под однозначными подразумевается, что токены не имеют морфологической омонимии).

3.3 Описание результатов

3.3.1 Метрики

Для оценки качества моделей были использованы стандартные метрики классификации:

1. Точность (Precision) – это доля верно извлечённых терминов относительно всех терминов, которые извлекла модель;
2. Полнота (Recall) – это доля верно найденных моделью терминов относительно всех терминов в тестовой выборке;

3. F-мера (F-measure) представляет собой гармоническое среднее между точностью и полнотой.

Назовём T_{rel} множество терминов, которые есть в корпусе, а T_{pred} – множество терминов, которые были извлечены системой. Тогда метрики рассчитываются по следующим формулам:

$$Precision = \frac{|T_{rel} \cap T_{pred}|}{|T_{rel}|},$$

$$Recall = \frac{|T_{rel} \cap T_{pred}|}{|T_{rel}|},$$

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall}.$$

При этом рассматривалось несколько вариантов того, что имеется в виду под словом “термин”.

В первом случае, под термином понимается вся последовательность токенов, входящая в состав термина. Если хотя бы один токен распознан неверно, то считается, что весь термин распознан неверно. В таблицах ниже такая метрика указана как “Полное совпадение”.

Во втором случае, под термином понимается токен, тэг которого принадлежит множеству {“B-TERM”, “I-TERM”}. Ввиду большой неоднозначности определения границ терминов, которая присутствовала также при разметке корпуса асессорами, данная метрика видится релевантной – она показывает, насколько хорошо модель способна распознать фразы, которые могут быть терминами, без указания точных границ. В таблицах ниже такая метрика указана как “Частичное совпадение”.

При реализации метрик были использованы библиотеки Scikit-learn[19] и Seqeval[20].

3.3.2 Результаты

В таблицах 7 и 8 приведены метрики для каждого из этапов построения метода на основе слабо контролируемого обучения для архитектур Bert-LSTM и BertForTokenClassification соответственно.

[19] https://pypi.org/project/scikit-learn/
[20] https://pypi.org/project/seqeval/

Таблица 7. Метрики для архитектуры Bert-LSTM

Метод	Полное совпадение			Частичное совпадение		
	Точность	Полнота	F1	Точность	Полнота	F1
Bert-iter_1	0.22	0.19	0.20	0.76	0.71	0.68
Bert-iter_1 + heuristics	0.39	0.28	0.33	0.76	0.75	0.74
Bert-iter_2	0.30	0.25	0.28	**0.77**	0.75	0.74
Bert-iter_2 + heuristics	**0.40**	0.29	0.34	0.77	0.77	0.76
Bert-iter_2 + heuristics + dictionary	0.39	**0.31**	**0.35**	**0.78**	**0.78**	**0.77**

Таблица 8. Метрики для архитектуры BertForTokenClassification

Метод	Полное совпадение			Частичное совпадение		
	Точность	Полнота	F1	Точность	Полнота	F1
Bert-iter_1	0.24	0.18	0.21	0.74	0.69	0.66
Bert-iter_1 + heuristics	0.39	0.26	0.31	0.75	0.74	0.73
Bert-iter_2	0.34	0.26	0.29	**0.77**	0.74	0.73
Bert-iter_2 + heuristics	**0.41**	0.29	0.34	0.76	0.76	0.75
Bert-iter_2 + heuristics + dictionary	0.40	**0.31**	**0.35**	**0.77**	**0.77**	**0.77**

В Таблице 9 приведены метрики для всех подходов, описанных в этой главе.

Таблица 9. Полученные результаты

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
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**

Относительно низкие метрики во многом связаны с различием разметки обучающего множества и золотого стандарта. В силу того, что обучающее множество было получено с помощью автоматической разметки терминов из словаря, последовательность токенов, являющаяся термином, не претерпевала никаких изменений. На самом деле, в реальных текстах термин может быть “разорван”, содержать синонимы, сокращения, пунктуационные знаки или вовсе быть неполным. Например, “*Анализ текста выполнен на трёх уровнях: морфологическом, синтаксическом и семантическом*”. Очевидно, что терминами здесь будут фразы “*морфологический уровень*”, “*синтаксический уровень*” и “*семантический уровень*”, но реализованный словарный подход не позволяет находить и конкатенировать строки таким образом. Как следствие, модели также не справляются с такими случаями.

Если проанализировать метрики частичного совпадения токенов, то видно, что модель способна находить места в тексте, в которых может быть термин, без точного определения границ термина. Учитывая, что задача определения границ термина является достаточно сложной даже для человека (что, например, показывает метрика согласованности асессоров при разметке сущностей в корпусе), то полученные метрики кажутся достаточными для

использования данного подхода для решения других задач (например, для задачи связывания именованных сущностей или классификации отношений между сущностями).

Также можно заметить, что оптимизированная версия RAKE гораздо лучше находит границы терминов – на это указывают метрики полного совпадения. Но при этом, статистический алгоритм сильно проигрывает алгоритмам машинного обучения в частичном совпадении. Это означает, что RAKE способен лучше находить целые цепочки терминов, в то время как модель способна точнее и полнее определить, входит ли конкретный токен в цепочку терминов или нет.

3.4 Применение модели к текстам другой области знаний

Для этого эксперимента были использованы размеченные тексты с соревнования RuREBus [93]. В качестве исходной коллекции текстов использовался корпус Минэкономразвития РФ, содержащий программы стратегического развития. Так как разработанная система работает с текстами из области математики и информационных технологий, то область экономики как раз подходит для проверки способности модели к обобщению. Но стоит отметить, что жанры текстов также различаются: модель была обучена на текстах научных статей, в то время как в соревновании используются тексты различных экономических документов.

В разметке этого датасета используется 8 типов именованных сущностей:

1. ***Metric*** – индикатор или объект, на основании которого производится сравнение (например, “*уровень рождаемости*”, “*экономический рост*”);
2. ***Economics*** – экономическая сущность или объект инфраструктуры (например, “*ПАО Газпром*”, “*библиотечные и музейные фонды*”);
3. ***Institution*** – институты, структуры и организации (например, “*Центр занятости молодёжи*”, “*системы дорог*”);
4. ***Binary*** – бинарная характеристика или единичное действие (например, “*модернизация*”, “*функционирует*”);
5. ***Compare*** – сравнительная характеристика (например, “*рост*”, “*негативная динамика*”);
6. ***Qualitative*** – качественная характеристика (например, “*эффективный*”, “*безопасный*”);
7. ***Social*** – социальный объект (например, “*научный и образовательный потенциал*”, “*досуг*”);
8. ***Activity*** – деятельность, события (например, “*реставрационные работы*”, “*ярмарка выходного дня*”).

Из приведённых примеров видно, что под сущностью авторы соревнования понимают не только именные группы, но и глаголы и глагольные группы и просто отдельные прилагательные. Такой принцип выделения именованных сущностей расходится с тем, который используется в данной работе – под именованными сущностями подразумеваются только существительные и именные группы.

В силу того, что понимание термина (и, следовательно, принципы разметки), принятое в данной работе, отличается от того, что используется в соревновании, сравнение метрик кажется некорректным. Тем не менее, можно проанализировать результаты и сделать некоторые выводы.

В Таблице 10 приведены примеры разметки предложений из корпуса RuREBus с помощью предложенного выше подхода – в квадратных скобках заключены выделенные сущности.

Из данной таблицы видно, что с помощью предложенного подхода в тексте выделяются последовательности токенов, которые являются сущностью в данном контексте.

Также были рассчитаны метрики, которая модель способна достичь на данном корпусе. Метрики рассчитывали на двух подмножествах RuREBus:

1. Оценивались только те сущности, которые попадают в категории Metric, Economics, Compare, т.к. именно эти группы по описанию наиболее близки к тем, что рассматриваются в данной работе;
2. Оценивались все сущности, независимо от их категории.

Было рассчитано только полное совпадение сущностей. Полученные результаты приведены в Таблице 11.

Таблица 10. Примеры разметки корпуса RuREBus моделью Bert-LSTM

№	Текст
1	*Повышение **[доступности транспортных услуг]** для **[населения]**.*
2	*Организация **[временного трудоустройства отдельных категорий безработных граждан]**.*
3	***[Право]** **[граждан]** на благоприятную **[среду жизнедеятельности]** закреплено в основном **[законе государства]** – **[Конституции Российской Федерации]**.*
4	*Контроль за исполнением **[постановления]** возложить на **[первого заместителя руководителя администрации]** МР "Усть-Вымский" Карпову А.Д.*
5	***[Оценка эффективности рисков]** – **[риски]** низкие.*
6	***[Деятельность учреждений культуры]** и искусства является одной из важнейших составляющих современной культурной жизни.*
7	***[ПОСТАНОВЛЕНИЕ]** об утверждении муниципальной программы "**[Поддержка сельскохозяйственных товаропроизводителей]** и **[создание условий]** для **[развития сферы заготовки]** и **[переработки дикорастущего сырья Верхнекетского района]** на 2016-2021 годы".*
8	***[Паспорт]** **[муниципальной программы]** городского округа Кашира "**[Спорт городского округа]** Кашира" на 2017-2021 годы.*
9	*Приложение к **[постановлению администрации муниципального имущества]***
10	*За этот период было установлено и заменено 366 дорожных **[знаков]** и 43 сигнальных столбика на **[железнодорожных переездах]**.*

Таблица 11. Метрики извлечения терминов на корпусе RuREBus

Подмножество	Точность	Полнота	F1
Сущности из категорий Metric, Economics, Compare	0.07	0.15	0.10
Все сущности	0.17	0.13	0.15
Результаты, полученные авторами статьи [93]			
BERT	0.83	0.86	0.85

3.5 Выводы

В этой главе описаны эксперименты с различными подходами для извлечения терминов – словарный подход, статистический (RAKE), с использованием нейронных сетей, а также подход на основе слабо контролируемого обучения с использованием архитектур на основе трансформеров.

Все подходы сравнивались друг с другом по основным метрикам информационного поиска – точность, полнота, F-мера. Для большей информативности учитывалось также, была ли найдена сущность полностью или только частично – из-за того, что определение границ термина является субъективной задачей, это разделение видится важным.

Полученные результаты показали, что статистический подход с определёнными улучшениями показывает лучшие результаты при определении чётких границ терминов, в то время как модели, полученные в результате слабо контролируемого обучения, показывают значительно более высокие результаты, чем остальные методы, и являются достаточными для применения подхода для решения практических задач.

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

Глава 4. Извлечение и классификация отношений между научными терминами

4.1 Формальная постановка задачи

Пусть дано предложение $S = \{x_0, x_1, ..., x_m\} (m \leq n)$, где x_i – токены. Для его элементов определена операция сцепления (конкатенации): $e_i = x_k x_{k+1} \dots x_{k+l} (0 \leq i, k \leq m; l \geq 0)$,), где возможны три случая:

- если какой-то из элементов, идущих после x_k, является знаком препинания, он присоединяется к предыдущему без добавления знака «пробел»;
- если элемент является дефисом, он присоединяется без добавления пробелов до и после него;
- в остальных случаях элемент присоединяется к предыдущему с добавлением знака «пробел».

Такое e_i назовём сущностью.

Рассмотрим пару сущностей (e_i, e_j) для $i \neq j$, и множество меток $Rel = \{CAUSE, COMPARE, ISA, PART\_OF, SYNONYMS, TOOL, USAGE, NONE\}$. Требуется построить классификатор, который паре (e_i, e_j) сопоставляет метку из Rel, т.е. $\gamma: (e_i, e_j) \rightarrow Rel$.

Все отношения, кроме SYNONYMS и NONE, являются *асимметричными*, т.е. для остальных отношений выполняется условие: $\forall R \in Rel \setminus \{SYNONYMS, NONE\} (e_i R e_j \Rightarrow \neg e_j R e_i)$.

4.2 Классификация отношений

В русском языке размеченные данные для этой задачи также представлены в небольшом количестве. Это означает, что применение стандартного цикла обучения нейронных сетей затруднено. Для решения этой задачи был применён подход zero-shot learning. Идея этого подхода состоит в том, чтобы взять предобученную мультиязыковую модель и дообучить её на данных на том языке, в котором они хорошо представлены. Затем оценить качество модели на русскоязычном корпусе. Гипотеза состоит в том, что информация из другого языка поможет модели делать предсказания в том числе и на данных на целевом языке.

В качестве предобученной языковой модели для получения векторных представлений была взята модель BERT bert-base-multilingual-cased. Для классификации отношений использовалась архитектура модели R-BERT, которая была предложена в статье [59] и представлена на Рисунке 6.

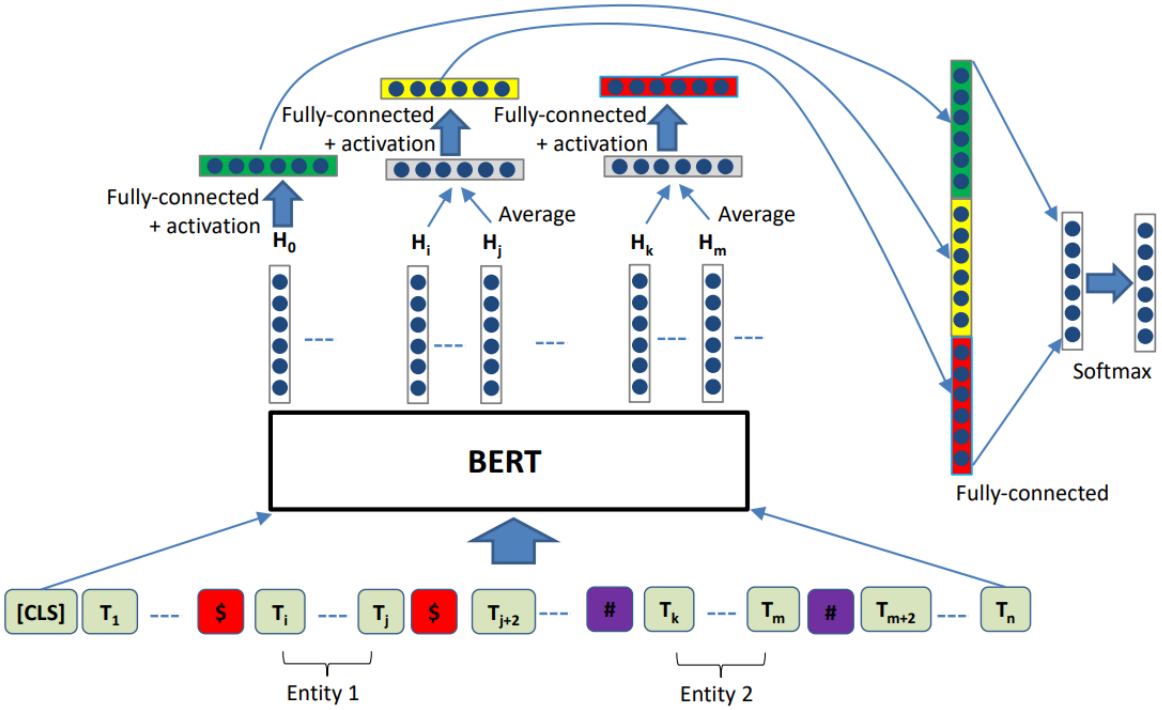


Рисунок 6. Архитектура R-BERT

Идея этой архитектуры состоит в том, чтобы для входного текста получить три векторных представления: от токена CLS (в языковых моделях BERT данный токен содержит векторное представление всей входной последовательности) и для двух сущностей. Векторное представление каждой сущности усредняется, затем все три вектора конкатенируются и подаются на вход полносвязному слою, который выполняет роль классификатора. Выходом модели является тип отношения, которым связаны две входные сущности.

Дообучали модели мы на англоязычном корпусе SciERC [83], который, в том числе, содержит информацию об отношениях между научными терминами. Список отношений, их значения и статистика встречаемости в корпусе приведены в Таблице 12.

Таблица 12. Описание отношений из корпуса SciERC

Отношение	Значение	Количество примеров		
		train	dev	test
USED-FOR	B используется для A; B моделирует A; A обучено на B; B использует A; A основано на B	1690	214	533
FEATURE-OF	B принадлежит A; B является характеристикой A	267	51	59
HYPONYM-OF	B является гипонимом A; B – это тип A	300	44	67
PART-OF	B является частью A	183	27	64
COMPARE	Симметричное отношение, сравнение двух моделей, методов и др.	203	36	47
CONJUNCTION	Симметричное отношение, указывает на то, что две сущности выполняют одну функцию	403	59	123

4.3 Извлечение отношений

Следующий наш шаг состоял в применении этой концепции, но уже к задаче совместного извлечения и классификации отношений. Основное отличие от предыдущего пункта состоит в том, что теперь в данные добавляются примеры пар терминов, которые не связаны никаким семантическим отношением. То есть, требуется классифицировать не только тип отношения между двумя сущностями, но и определить, связаны ли два термина каким-либо отношением или нет.

4.3.1 Использование модели классификации отношений

Первый эксперимент, который был взят за базовый, состоял в использовании той же модели, что применяется для классификации отношений, но отношение NONE определялось следующим образом. Если вероятность предсказанного класса ниже определённого порога, то считаем, что модель предсказала отношение NONE, иначе – оставляем предсказанный класс. Идея здесь заключается в том, что низкая вероятность предсказанного класса указывает на то, что модель не может уверенно выбрать тип отношения, а значит, данная пара сущностей не связана никаким отношением. Были проведены эксперименты с несколькими значениями пороговой величины: 0.4, 0.5, 0.6, 0.7 и 0.8. Полученные результаты представлены в Таблицах 15 и 16.

4.3.2 Подход, основанный на лексических шаблонах

Идея этого подхода состоит в том, чтобы вручную собрать лексические маркеры, которые однозначно указывают на то или иное семантическое отношение. В Приложении 3 приведены лексические шаблоны для каждого из отношений.

Алгоритм извлечения отношений можно описать следующим образом:

1. Для пары сущностей получаем контекст – последовательность токенов, которые находятся между этими сущностями (не включая сами сущности);
2. Если длина контекста больше 10 токенов, то считаем, что данная пара сущностей не связана семантическим отношением;
3. Если длина контекста менее 3 токенов, то проверяем в нём наличие маркеров для коротких контекстов; если есть совпадение, то считаем, что пара сущностей связана данным отношением;
4. Если длина контекста более 3 токенов и менее 10 токенов, то проверяем в нём наличие маркеров для длинных контекстов; если есть совпадение, то считаем, что пара сущностей связана данным отношением.

Сложность этого подхода состоит в том, что очень часто семантические связи выражены имплицитно – это означает, что они могут быть распознаны только при анализе контекста, без опоры на конкретные лексические единицы. Это вызывает трудности как при формировании словаря маркеров, так и в целом при решении данной задачи. Метрики, которые удалось получить с помощью этого подхода, представлены в Таблицах 17 и 18.

4.3.3 Подход, основанный на zero-shot learning

Следующая группа экспериментов состояла в обучении модели (как и для задачи классификации отношений), но теперь в данные добавлялись примеры, в которых две сущности не связаны отношениями, с меткой NONE. Так как некоторые из отношений в корпусе являются несимметричными, то есть информация о субъекте и объекте является важной при определении типа отношений (HYPERONYM-OF является примером такого типа отношения), то примеры с парами сущностей без отношений были получены путём попарного объединения сущностей, причём в обе стороны.

Были проведены эксперименты с двумя архитектурами – той, что была описана в предыдущем пункте, а также с архитектурой BertForSequenceClassification из библиотеки HuggungFace[21].

Очевидно, что количество примеров с сущностями, которые не связаны никаким семантическим отношением, значительно превышает количество связанных сущностей. Такой дисбаланс напрямую влияет на качество модели. Поэтому были проведены эксперименты с различным количеством таких примеров, сущности которых не связаны отношением. Сэмплирование выполнялось двумя способами:

1. Случайным образом выбирали n% пар сущностей, которые не связаны отношением;
2. Для пары сущностей случайным образом выбирали направление отношений (в таком случае, в обучающее множество попадал только один пример с данной парой сущностей).

Важно отметить, что сэмплирование выполнялось только для обучающих данных, на валидационном и тестовом наборах данных учитывались все примеры с несвязанными сущностями. Полученные результаты представлены в Таблице 19. В Приложении 4 приведены метрики для каждого отношения отдельно. Прочерком “-” обозначены отношения, которые не участвовали в тестировании.

4.3.4 Ансамбль решений

Для улучшения качества решения задачи извлечения отношений был проведён эксперимент по объединению двух подходов: с использованием лексико-синтаксических шаблонов и основанного на zero-shot learning. Комбинирование указанных алгоритмов производилось следующим образом:

[21] https://huggingface.co/

1. Если оба подхода предсказали, что между парой сущностей нет отношения, то результатом является тип отношения NONE;
2. Если подход, основанный на zero-shot learning, предсказал конкретный тип отношения (отличный от NONE) для пары сущностей, то результатом является этот тип отношения;
3. Если подход, основанный на лексико-синтаксических шаблонах предсказал конкретный тип отношения (отличный от NONE) для пары сущностей, то результатом является этот тип отношения.

Такая комбинация подходов позволила улучшить качество извлечения отношений в целом. Метрики, которые удалось получить с помощью этого подхода, представлены в Таблицах 20 и 21.

4.4 Описание результатов

4.4.1 Метрики

Для оценки качества моделей была использована F1-метрику (подробное описание приведено в п.3.3.1).

Для подходов, основанных на zero-shot learning, были использованы два корпуса: SciERC для английского языка и RuSERRC для русского языка. Из-за того, что наборы отношений отличаются в этих двух корпусах, при расчёте метрик для русского языка использовалось следующее соответствие типов отношений:

1. Отношению ISA (RuSERRC) соответствует отношение HYPONYM-OF (SciERC);
2. Отношению PART-OF (RuSERRC) соответствует отношение PART-OF (SciERC);
3. Отношению USAGE (RuSERRC) соответствует отношение USED-FOR (SciERC);
4. Отношению TOOL (RuSERRC) соответствует отношение USED-FOR (SciERC);
5. Отношению COMPARE (RuSERRC) соответствует отношение COMPARE (SciERC).

Отношения CAUSE и SYNONYMS из набора данных RuSERRC были исключены из процесса тестирования для моделей zero-shot learning.

Для подхода, основанного на лексических маркерах, тестирование проводилось на всём корпусе RuSERRC, без исключения каких-либо видов отношений.

4.4.2 Результаты

4.4.2.1 Результаты для задачи классификации отношений

Полученные метрики для задачи классификации отношений приведены в Таблице 13. В Таблице 14 представлены значения метрики F1 macro по каждому из отношений.

Таблица 13. Метрики для классификации отношений

Метрика	SciERC	RuSERRC
f1-micro	0.76	**0.68**
f1-macro	0.57	**0.51**

Таблица 14. Метрики классификации по отношениям

Отношение	SciERC	RuSERRC
USED-FOR	0.85	0.82
FEATURE-OF	0.47	-
HYPONYM-OF	0.75	0.53
PART-OF	0.10	0.48
COMPARE	0.47	0.20
CONJUNCTION	0.78	-

Приведённые метрики доказывают поставленную гипотезу – действительно, дообучение на данных другого языка помогает при классификации данных целевого языка.

Низкое значение метрики для отношения COMPARE, по-видимому, связано с разным пониманием этого отношения в двух представленных корпусах. Кроме того, данное отношение является наименее представленным в наборе RuSERRC – оно встречается только 9 раз.

4.4.2.2 Результаты для задачи извлечения отношений

В таблицах 15 и 16 приведены значения метрик для baseline-эксперимента (итоговые и значения метрики F1 macro отдельно по каждому из отношений), который заключался в использовании модели для классификации отношений, а отсутствие отношения определялось по порогу. Все значения получены на корпусе RuSERRC.

Таблица 15. Метрики для baseline-эксперимента

Метрика	Порог				
	0.4	0.5	0.6	0.7	0.8
F1-micro	0.08	0.21	0.37	0.51	**0.65**
F1-macro	0.06	0.11	0.15	0.19	**0.21**

Таблица 16. Метрики для baseline-эксперимента по отношениям

Отношение	Порог				
	0.4	0.5	0.6	0.7	0.8
USED-FOR	0.06	0.06	0.07	0.08	**0.1**
HYPONYM-OF	0.09	0.11	0.14	0.14	**0.15**
PART-OF	0.04	0.04	0.04	0.05	**0.05**
COMPARE	0.0	0.0	0.0	0.0	0.0
NONE	0.12	0.34	0.53	0.67	**0.79**

Таблица 17. Метрики подхода, основанного на шаблонах

Метрика	Точность	Полнота	F1
F1-micro	-	-	0.88
F1-macro	0.24	0.30	0.23

Таблица 18. Метрики подхода, основанного на шаблонах, по отношениям

Отношение	Точность	Полнота	F1
CAUSE	0.07	0.05	0.06
COMPARE	0.00	0.00	0.00
ISA	0.13	0.15	0.14
NONE	0.95	0.92	0.94
PART_OF	0.28	0.06	0.10
SYNONYMS	0.23	0.82	0.35
TOOL	0.08	0.03	0.04
USAGE	0.18	0.36	0.24

Таблица 19. Метрики для экспериментов с трансформер-архитектурами

Эксперимент	SciERC		RuSERRC	
	F1-micro	F1-macro	F1-micro	F1-macro
R-BERT: 10% примеров с None; для пары сущностей проверяются две связи	0.73	0.33	0.76	0.20
R-BERT: 10% примеров с None; для пары сущностей проверяется одна связь	0.57	0.29	0.65	0.18
R-BERT: 50% примеров с None; для пары сущностей проверяются две связи	**0.89**	0.28	**0.94**	0.20

R-BERT: 50% примеров с None; для пары сущностей проверяется одна связь	0.81	0.33	0.85	**0.23**
BertForSequenceClassification: 50% примеров с None; для пары сущностей проверяются две связи	0.67	**0.51**	0.63	0.18

Таблица 20. Метрики комбинированного подхода

Метрика	Точность	Полнота	F1
F1-micro	-	-	0.86
F1-macro	0.30	0.29	0.27

Таблица 21. Метрики комбинированного подхода по отношениям

Отношение	Точность	Полнота	F1
USED-FOR	0.14	0.36	0.20
HYPONYM-OF	0.12	0.13	0.12
PART-OF	0.28	0.06	0.10
COMPARE	0.00	0.00	0.00
NONE	0.96	0.90	0.93

Анализ полученных метрик показывает, что задача извлечения семантических отношений, действительно, является сложной для автоматического решения. Это также подтверждают и результаты других исследователей. К сожалению, эти результаты были получены на других наборах данных, поэтому сравнивать их между собой не является корректным. Например, в статье [94] описан размеченный корпус DocRED для извлечения информации, в том числе семантических отношений, на уровне документа. В наборе данных имеется разметка 96 типов отношений, которые были выбраны на основе их популярности в Викиданных. Авторы также

предоставили результаты нескольких моделей, которые были обучены и протестированы на этом датасете. Максимальное значение метрики F1, которое им удалось получить, составило 0.48.

4.5 Выводы

Для извлечения информации о семантических отношениях была применена техника zero-shot learning, идея которой заключается в следующем. Сначала межъязыковую модель дообучают на данных того языка, которые представлены в достаточном количестве, а затем применяют эту модель к данным малоресурсного языка без дообучения. В качестве данных для обучения был использован размеченный корпус на английском языке.

Анализ результатов показал, что данный метод хорошо работает для задачи классификации отношений – для заданной пары сущности известно, что они связаны отношением, и нужно определить тип этого отношения. Модель, которая не видела примеров на русском языке, тем не менее, показывает хорошее качество классификации.

Задача извлечения отношений подразумевает ещё определение того, связана ли пара сущностей отношением или нет. Здесь были проведены эксперименты не только с различными моделями, но также исследовали влияние сэмплирования на качество алгоритма. Очевидно, что примеров пар сущностей, которые не связаны отношениями, гораздо больше, чем тех, которые связаны – отсюда возникает дисбаланс классов, который влияет на работу моделей. Было опробовано два способа сэмплирования обучающих данных для сглаживания этого дисбаланса. В целом, задача извлечения отношений видится сложной и нуждается в дальнейшем исследовании, что подтверждают полученные метрики, а также анализ результатов работ других исследователей.

Глава 5. Автоматическое связывание сущностей

5.1 Формальная постановка задачи

Данная работа посвящена задаче связывания сущностей, где в качестве сущностей рассматриваются термины.

Назовём $Entities$ множество сущностей и $Properties$ множество свойств. База знаний состоит из множества троек вида $< Subject\, Predicate\, Object >$, где $Subject$ и $Object$ являются элементами множества $Entities$, а $Predicate$ – элементом множества $Properties$.

Назовём токеном x_i – слово или знак препинания в тексте. Рассмотрим последовательность токенов $X = \{x_1, x_2, ..., x_n\}$. Сущностью Ent будет называться подпоследовательность таких токенов, которая представляет собой термин. Тогда мощность множества E, которое содержит в себе сущности Ent, всегда меньше либо равна мощности множества X, включая значение 0.

Задача автоматического связывания сущности состоит в построении такой функции F, которая бы для каждой сущности из множества E ставила бы в соответствие элемент из множества $Entities$ либо ε, где ε – отсутствие сущности в заданной базе знаний:

$$F\colon\ E \to Entities \cup \varepsilon.$$

В данной работе в качестве базы знаний используются Викиданные. 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 элементов, утверждений и ссылок на сайты.

1. Каждый из элементов имеет уникальный идентификатор с префиксом Q и числовой частью, как, например, "*Дуглас Адамс (Q42)*".
2. Утверждения 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.

5.2. Описание алгоритма

В рамках исследования был реализован алгоритм связывания сущностей (Рисунок 7) и проверена его работа на корпусе с размеченными научными терминами [98]. Поскольку

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.

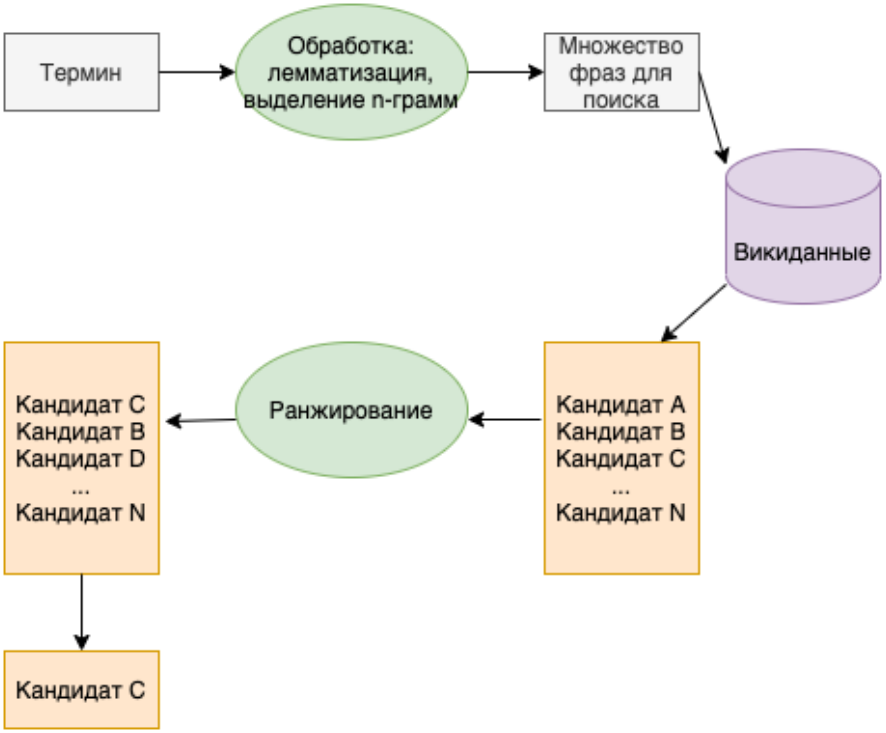


Рисунок 7. Схема алгоритма связывания сущностей

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

Все сущности – входные и в базе знаний – проходят лемматизацию с помощью MyStem. Это нужно для более точного поиска совпадений, т.к. русский язык отличается богатой морфологией и большим количеством словоформ.

На этапе создания массива кандидатов ищется построчное совпадение входной сущности с сущностями в базе знаний. Кроме того, для более полного формирования этого списка искалось не только полное название входной сущности, но также униграммы, биграммы и триграммы, полученные из её названия. Например, для сущности “*Язык программирования Python*” для поиска в Викиданных будут использованы следующие подстроки (с учётом лемматизации):

- *язык*;
- *программирование*;
- *python*;
- *язык программирование*;
- *программирование python*;
- *язык программирование python*.

Этап нахождения релевантной сущности из базы знаний рассматривался как задача ранжирования. Чтобы учитывать не только названий сущностей, но и её контекст, были использованы:

1. Для входного упоминания – название сущности, 5 токенов до неё и 5 токенов после неё (без учёта границ предложений);
2. Для сущности из Викиданных – название сущности, её синонимов и описание.

Каждую сущность (входную и из полученного множества кандидатов) была представлена в виде вектора V, который был получен по формуле:

$$V = \frac{\sum_{i=0}^{n} vector_i}{n}, \text{ где}$$

$vector_i$ – векторное представление для i-ого токена сущности,

n – количество токенов в сущности.

Векторные представления были получены с использованием предобученной модели Fasttext.

Затем полученные для каждого вектора сущности из базы знаний было рассчитано косинусное расстояние между ним и вектором входного упоминания. Кандидаты были отранжированы по этому расстоянию, далее кандидат, вектор которого наиболее близок к вектору входной сущности, считается связанной сущностью.

5.3 Описание результатов

5.3.1 Метрики

Для определения качества алгоритма был использован ряд метрик.

1. ***Accuracy*** – определяется как отношение количества верно связанных терминов ко всем терминам. Так как не все термины в корпусе удалось связать, информативнее будет разделить эту метрику на две: ***Accuracy*** – принимает во внимание все сущности,

и ***LinkedAccuracy*** – считается только на том наборе терминов, для которых нашлась сущность в графе знаний в корпусе. Таким образом, $Accuracy$ вычисляется по формуле:

$$Accuracy = \frac{CorrectEntities}{AllEntitties}, \text{ где}$$

$CorrectEntities$ – количество верно связанных терминов,

$AllEntities$– количество всех терминов в корпусе.

Обозначим $AllLinkedEntities$количество всех терминов в корпусе, которые имеют связь с сущностью в Викиданных. Тогда $LinkedAccuracy$ вычисляется по формуле:

$$LinkedAccuracy = \frac{CorrectLinkedEntities}{AllLinkedEntities}, \text{ где}$$

$CorrectLinkedEntities$ – количество верно связанных алгоритмом терминов среди всех связанных терминов.

2. ***Среднее количество кандидатов***. Эта метрика показывает, насколько хорошо работает этап генерации кандидатов: если значение относительно мало, то это означает, что 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. Эта метрика также была разбита на две: $AveragedCandidates$ – среднее количество кандидатов для всех сущностей и $LinkedAveragedCandidates$ – среднее количество кандидатов для набора терминов, которые удалось связать.

$$AveragedCandidates = \frac{\sum_{1}^{n} |Candidates_i|}{AllEntities}, \text{ где}$$

$Candidates_i$– множество полученных кандидатов для сущности.

Обозначим $LinkedCandidates_i$ множество сгенерированных кандидатов для всех терминов, связанных с Викиданными. Тогда формула для метрики $LinkedAveragedCandidates$ имеет вид:

$$LinkedAveragedCandidates = \frac{\sum_{1}^{n} |Linked\_candidates_i|}{n\_all\_linked\_entities}.$$

3. ***Наличие подходящего кандидата в списке, найденном алгоритмом***. Данная метрика считалась только для множества терминов в корпусе, которые имеют связь с сущностью из графа знаний, и вычислялась по формуле:

$$TopCandidates = \frac{CorrectSets}{AllLinkedEntities}$$, где

$CorrectSets$ – это количество сгенерированных списков кандидатов, содержащих верную сущность.

5.3.2 Результаты

Качество алгоритма оценивалось на вручную размеченном корпусе (описанном в п.2.2). Полученные метрики представлены в Таблице 22.

Таблица 22. Метрики связывания сущностей

Accuracy	LinkedAccuracy	AveragedCandidates	LinkedAveragedCandidates	TopCandidates
0.38	0.23	10.29	7.38	0.76

Метрики показывают, что в среднем для одной сущности находятся 10 кандидатов на связывание в базе знаний (и 7 для термина, который точно имеет сущность для связывания в базе знаний). Значение метрики *TopCandidates* показывает, что в 76% случаях в полученных множествах кандидатов имеется верная сущность для связывания. Но значение метрики *LinkedAccuracy* показывает, что только в 23% случаев найденные кандидаты ранжируются верно. Таким образом приоритетным направлением для дальнейшей работы здесь остаётся улучшение ранжирования найденных кандидатов, а также генерация более полных множеств кандидатов.

5.4 Выводы

В данной главе описан алгоритм автоматического связывания сущностей и предложены возможные пути его улучшения. В качестве базы знаний были использованы Викиданные, т.к. это наиболее полная и актуальная база знаний на сегодняшний день с поддержкой русского языка. Был реализован алгоритм связывания сущностей, который не требует обучения и может быть основой для добавления последующих, более сложных методов.

Был предложен набор метрик, которые показывают качество алгоритма с разных его аспектов: как работу системы в целом, так и отдельных её компонентов – генерации кандидатов и ранжирования. Полученные метрики показали сильные и слабые стороны реализованного алгоритма.

Заключение

В данной работе исследованы различные методы и подходы для извлечения информации из научных текстов на русском языке, а именно: извлечение терминов, извлечение отношений между терминами и связывание сущностей с внешней базой знаний. Предложен подход для извлечения терминов на основе слабоконтролируемого обучения. Выполнена реализация всех рассмотренных алгоритмов. Основным преимуществом этих подходов является работа в условиях ограниченного набора размеченных данных, объём которых не позволяет качественно обучить модель машинного обучения.

Основные результаты исследования:

1. Собран и размечен корпус научных текстов для задач извлечения научных терминов, извлечения отношений и связывания сущностей с внешней базой знаний.
2. Исследованы различные методы извлечения терминов из научных текстов: словарный метод, статистический метод, с использованием машинного обучения.
3. Предложен подход для извлечения терминов на основе слабоконтролируемого обучения, идея которого заключается в обучении модели на большом количестве данных с автоматической разметкой.
4. Адаптирован метод извлечения отношений между терминами, основанный на переносе обучения моделей с английского языка на русский в постановке zero-shot learning.
5. Описан алгоритм и реализован метод связывания терминов с сущностями в базе знаний. Предложен ряд метрик для оценки качества метода, учитывающий различные аспекты.
6. Разработан программный комплекс для извлечения информации из научных текстов (обобщенная схема приведена в Приложении 5).

В дальнейшем планируется улучшить предложенные алгоритмы за счёт модификаций алгоритмов и данных.

Для задачи извлечения научных терминов важно дополнить список эвристик для лучшей корректировки границ сущности; пополнить словарь терминов теми, что были найдены моделью. Затем планируется апробировать алгоритм на текстах других предметных областей.

Для задачи извлечения отношений планируется доработать лексико-синтаксические шаблоны, попробовать подход few-shot learning для дообучения модели на данных русского языка, а также использовать результат работы модуля автоматического связывания сущностей для повышения качества извлечения отношений.

Для модуля автоматического связывания сущностей важно учитывать синонимы, аббревиатуры и названия на иностранных языках для более полной генерации кандидатов. Для задачи ранжирования планируется учитывать контекст, в котором находится термин, а также информацию о сущности-кандидате в базе знаний.

Список сокращений и условных обозначений

Active learning – это область машинного обучения, где алгоритм взаимодействует с некоторым источником информации, способным размечать запрошенные данные;

BERT – Bidirectional Encoder Representations from Transformers;

Bi-LSTM – Bidirectional Long Short-Term Memory;

BIO – это формат разметки токенов в задачах тэгированния последовательностей, согласно которому первом токену в сущности соответствует тэг с префиксом B, вторым и последующим токенам в сущности соответствует тэг с префиксом I, токенам, не входящих в сущность, соответствует тэг с префиксом O;

CNN – Convolutional Neural Network;

CRF – Conditional Random Fields;

ELMo – Embeddings from Language Model;

Few-shot learning – это процесс обучения модели с использованием только нескольких обучающих примеров;

LDA – Latent Dirichlet allocation;

LSTM – Long Short-Term Memory;

MRC – Machine Reading Comprehension;

NFM – Non-negative matrix factorization;

One-hot encoding – это способ кодирования, при котором в векторе только один элемент имеет значение 1, а остальные – 0;

RAKE – Rapid automatic keyword extraction;

Sequence labeling – это задача тэгирования последовательности, целью которой является определение тэга для каждого токена в тексте;

Smooth labeling – это метод регуляризации для задач классификации, позволяющий предотвратить слишком уверенное предсказание меток модели во время обучения и плохое обобщение;

Supervised learning – это один из способов машинного обучения, в ходе которого система обучается с помощью примеров;

SVM – Support Vector Machine;

TF-IDF – Term Frequency – Inverse Document Frequency;

Zero-shot learning – это процесс обучения модели без использования обучающих примеров;

Weak supervision – это область машинного обучения, в которой шумные или неточные источники генерируют автоматическую разметку большого количества данных, которые затем используются при обучении с учителе;

Глубокое обучение – это класс алгоритмов машинного обучения, которые состоят из нескольких слоёв для того, чтобы извлекать высокоуровневные признаки из входных данных;

Извлечение информации (англ. Information extraction, IE) – это процесс поиска в тексте необходимой информации, результатом которого является преобразование неструктурированной информации к структурированному виду;

Токен – это последовательность символов в документе, имеющая значение для анализа (например, слова, знаки пунктуации, числовые комплексы и пр.).

Список литературы

1. Science and engineering indicators. 2019. [Электронный ресурс] URL: https://ncses.nsf.gov/pubs/nsb20206/ (дата обращения: 09.11.2021).
2. Head A., Lo K., Kang D., Fok R., Skjonsberg S., Weld D., and Hearst M. Augmenting Scientific Papers with Just-in-Time, Position-Sensitive Definitions of Terms and Symbols. Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. Association for Computing Machinery, New York, NY, USA, Article 413, pp. 1–18. 2021. DOI: https://doi.org/10.1145/3411764.3445648.
3. Erera S., Shmueli-Scheuer M., Feigenblat G., Nakash O., Boni O., Roitman H., Cohen D., Weiner B., Mass Y., Rivlin O., Lev G., Jerbi A., Herzig J., Hou Y., Jochim C., Gleize M., Bonin F., Bonin F., Konopnicki D.. A Summarization System for Scientific Documents. 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. Association for Computational Linguistics, Hong Kong, China, pp. 211–216. 2019. DOI: 10.18653/v1/D19-3036.
4. Dong Y., Mircea A., Cheung J. Discourse-Aware Unsupervised Summarization for Long Scientific Documents. Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume. Association for Computational Linguistics, Online, pp. 1089–1102. 2021.
5. Tkaczyk D., Szostek P., Fedoryszak M., Dendek P., Bolikowski L. CERMINE: automatic extraction of structured metadata from scientific literature. In International Journal on Document Analysis and Recognition (IJDAR), vol. 18, no. 4, pp. 317–335. 2015. DOI: 10.1007/s10032-015-0249-8.
6. Epp S., Hoffmann M., Lell N., Mohr M., Scherp A.. A Machine Learning Pipeline for Automatic Extraction of Statistic Reports and Experimental Conditions from Scientific Papers. 2021. [Электронный ресурс] URL: https://arxiv.org/pdf/2103.14124.pdf (дата обращения: 09.11.2021).
7. Foppiano L., Romary L., Ishii M., Tanifuji M. Automatic Identification and Normalisation of Physical Measurements in Scientific Literature. In Proceedings of the ACM Symposium on Document Engineering 2019 (DocEng '19). Association for Computing Machinery, New York, NY, USA, Article 24, pp. 1–4. 2019. DOI :https://doi.org/10.1145/3342558.3345411.

8. Гусев В.Д., Саломатина Н.В. Метод итерационного построения шаблонов для поиска в текстах по катализу информации о химических процессах и условиях их протекания. Информационные и математические технологии в науке и управлении, № 4-1, с. 37–45. 2016.
9. Riedel N., Kip M., Bobrov E. ODDPub – a Text-Mining Algorithm to Detect Data Sharing in Biomedical Publications. Data Science Journal, 19(1), pp. 1–14. 2020. DOI: http://doi.org/10.5334/dsj-2020-042.
10. Shyam Saladi. JetFighter: Towards figure accuracy and accessibility. 2019. [Электронный ресурс] URL: https://elifesciences.org/labs/c2292989/jetfighter-towards-figure-accuracy-and-accessibility (дата обращения: 09.11.2021).
11. Chen G., Ma Sh., Chen Y., Dong L., Zhang D., Pan J., Wang W., Wei F. Zero-shot Cross-lingual Transfer of Neural Machine Translation with Multilingual Pretrained Encoders. Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, pp. 15–26. 2021.
12. Li X., Bing L., Zhang W., Li Zh., Lam W. Unsupervised Cross-lingual Adaptation for Sequence Tagging and Beyond. 2021. [Электронный ресурс] URL: https://arxiv.org/pdf/2010.12405.pdf (дата обращения: 09.11.2021).
13. Sherborne T., Lapata M. Zero-Shot Cross-lingual Semantic Parsing. 2021. [Электронный ресурс] URL: https://arxiv.org/pdf/2104.07554.pdf (дата обращения: 09.11.2021).
14. Gong Ch., Wang D., Li M., Chandra V., Liu Q. KeepAugment: A Simple Information-Preserving Data Augmentation Approach. In 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, pp. 1055–1064. 2021. DOI: 10.1109/CVPR46437.2021.00111.
15. Zhang Zh., Xie Sh., Chen M., Zhu H. HandAugment: A Simple Data Augmentation Method for Depth-Based 3D Hand Pose Estimation. 2020. [Электронный ресурс] URL: https://arxiv.org/pdf/2001.00702.pdf (дата обращения: 09.11.2021).
16. Inoue H. Data Augmentation by Pairing Samples for Images Classification. 2018. [Электронный ресурс] URL: https://arxiv.org/pdf/1801.02929.pdf (дата обращения: 09.11.2021).
17. Wei J., Zou K. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language

Processing (EMNLP-IJCNLP). Association for Computational Linguistics, Hong Kong, China, pp. 6382–6388. 2019. DOI: 10.18653/v1/D19-1670.

18. Bayer M., Kaufhold M., Buchhold B., Keller M., Dallmeyer J., Reuter Ch. Data Augmentation in Natural Language Processing: A Novel Text Generation Approach for Long and Short Text Classifiers. 2021. [Электронный ресурс] URL: https://arxiv.org/pdf/2103.14453.pdf (дата обращения: 09.11.2021).
19. Liesting T., Frasincar F., Truşcă M. Data augmentation in a hybrid approach for aspect-based sentiment analysis. In Proceedings of the 36th Annual ACM Symposium on Applied Computing (SAC '21). Association for Computing Machinery, New York, NY, USA, pp. 828–835. 2021. DOI: https://doi.org/10.1145/3412841.3441958.
20. Indurkhya N., Damerau F.. Handbook of Natural Language Processing (2nd. ed.). Chapman & Hall/CRC. 2010.
21. Kim Sang E., Meulder F.. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003. pp. 142–147. 2003.
22. Krallinger M., Leitner F., Rabal O., Vazquez M., Oyarzabal J., Valencia A. Overview of the chemical compound and drug name recognition (CHEMDNER) task. Proceedings of the Fourth BioCreative Challenge Evaluation Workshop vol. 2. pp. 6–37. 2013.
23. Wang X., Zhang Y., Ren X., Zhang Y., Zitnik M., Shang J., Langlotz C., Han J., Cross-type biomedical named entity recognition with deep multi-task learning, Bioinformatics, Volume 35, Issue 10, pp. 1745–1752. 2019. DOI: https://doi.org/10.1093/bioinformatics/bty869.
24. Jurafsky D., Martin J. Speech and Language Processing (2nd Edition). Prentice-Hall, Inc., USA. 2009.
25. Lima R., Espinasse B., Freitas F. A logic-based relational learning approach to relation extraction: The OntoILPER system. Engineering Applications of Artificial Intelligence, 78. pp. 142–157. 2019. DOI: https://doi.org/10.1016/j.engappai.2018.11.001.
26. Sysoev A., Andrianov I. Named Entity Recognition in Russian: the Power of Wiki-Based Approach. Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference “Dialogue 2016”. pp. 746–755. 2016.
27. Peters M., Neumann M., Iyyer M., Gardner M., Clark Ch., Lee K., Zettlemoyer L. Deep Contextualized Word Representations. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). Association for Computational Linguistics, New

Orleans, Louisiana, pp. 2227–2237. 2018. DOI: 10.18653/v1/N18-1202.

28. Gordeev D., Davletov A., Rey A., Akzhigitova G., Geymbukh G.. Relation extraction dataset for the Russian language. Computational Linguistics and Intellectual Technologies. Федеральное государственное бюджетное образовательное учреждение высшего образования Российский государственный гуманитарный университет (Москва), том 19, с. 348–360. 2020. DOI: 10.28995/2075-7182-2020-19-348-360.
29. Le A., Arkhipov M., Burtsev M. Application of a hybrid Bi-LSTM-CRF model to the task of Russian named entity recognition. Artificial Intelligence and Natural Language. Springer International Publishing, pp. 91–103. 2018.
30. Ma X., Hovy E. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Berlin, Germany, pp. 1064–1074. 2016. DOI: 10.18653/v1/P16-1101.
31. Sun S., Xiong Ch., Liu Zh., Liu Zh., Bao J. Joint Keyphrase Chunking and Salience Ranking with BERT. 2020. [Электронный ресурс] URL: https://arxiv.org/pdf/2004.13639.pdf (дата обращения: 09.11.2021).
32. Ziyadi M., Sun Y., Goswami A., Huang J., Chen W. Example-Based Named Entity Recognition. 2020. [Электронный ресурс] URL: https://arxiv.org/pdf/2008.10570.pdf (дата обращения: 09.11.2021).
33. Li X., Feng J., Meng Y., Han Q., Wu F., Li J. A Unified MRC Framework for Named Entity Recognition. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Online, pp. 5849–5859. 2020. DOI: 10.18653/v1/2020.acl-main.519.
34. Cai T., Zhou Y., Zheng H. Cost-Quality Adaptive Active Learning for Chinese Clinical Named Entity Recognition. International Conference on Bioinformatics and Biomedicine. Virtual Event, South Korea. pp. 528–533. 2020. DOI: 10.1109/BIBM49941.2020.9313302.
35. Devlin J., Chang M., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Association for Computational Linguistics, Minneapolis, Minnesota, pp. 4171–4186. 2019. DOI: 10.18653/v1/N19-1423.
36. Stanković R., Krstev C., Obradović I., Lazić B., Trtovac A. Rule-based Automatic Multi-word Term Extraction and Lemmatization. Proceedings of the Tenth International Conference on

Language Resources and Evaluation (LREC'16). European Language Resources Association (ELRA), Portorož, Slovenia, pp. 507–514. 2016.

37. Пименов И.С., Саломатина Н.В. Построение модели изменения во времени содержания тематических кластеров в коллекциях научных текстов. Труды международной конференции “АПВПМ”, № 2019, с. 385–392, 2019. DOI: 10.24411/9999-016A-2019-10062.
38. Ivanisenko T. V., Saik O. V., Demenkov P. S., Ivanisenko N. V., Savostianov A. N., Ivanisenko V. A. ANDDigest: a new web-based module of ANDSystem for the search of knowledge in the scientific literature. BMC Bioinformatics, 21. 2020. DOI: https://doi.org/10.1186/s12859-020-03557-8.
39. Боровикова О.И., Кононенко И.С., Сидорова Е.А. Подход к извлечению информации из протоколов клинических испытаний на основе медицинской онтологии. Системная информатика, №9, с. 93–110, 2017. DOI: 10.31144/si.2307-6410.2017.n9.p.93-110.
40. Yuan Y., Gao J., Zhang Y. Supervised learning for robust term extraction. International Conference on Asian Language Processing (IALP), pp. 302–305. 2017. DOI: 10.1109/IALP.2017.8300603.
41. Conrado M., Pardo T., Rezende S.. A Machine Learning Approach to Automatic Term Extraction using a Rich Feature Set. Proceedings of the 2013 NAACL HLT Student Research Workshop. Association for Computational Linguistics, Atlanta, Georgia, pp. 16–23. 2013.
42. 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, Volume 12, Issue 5, Article 57, pp. 1–41. 2018. DOI: https://doi.org/10.1145/3201408.
43. Bilu Y., Gretz Sh., Cohen E., Slonim N. What if we had no Wikipedia? Domain-independent Term Extraction from a Large News Corpus. 2020. [Электронный ресурс] URL: https://arxiv.org/pdf/2009.08240.pdf (дата обращения: 09.11.2021).
44. Wang R., Liu W., McDonald Ch. Featureless Domain-Specific Term Extraction with Minimal Labelled Data. Proceedings of the Australasian Language Technology Association Workshop 2016. Melbourne, Australia, pp. 103–112. 2016.
45. Hossari M., Dev S., Kelleher J. TEST: A Terminology Extraction System for Technology Related Terms. In Proceedings of the 2019 11th International Conference on Computer and Automation Engineering (ICCAE 2019). Association for Computing Machinery, New York, NY, USA, pp. 78–81. 2019. DOI: https://doi.org/10.1145/3313991.3314006.

46. 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. Proceedings of Interspeech 2018, pp. 2072–2076. 2018. DOI: 10.21437/Interspeech.2018-2017.
47. Bolshakova E., Loukachevitch N., Nokel M. Topic Models Can Improve Domain Term Extraction. Advances in Information Retrieval. ECIR 2013. Lecture Notes in Computer Science, vol 7814. Springer, Berlin, Heidelberg. 2013. https://doi.org/10.1007/978-3-642-36973-5_60.
48. Shi P., Lin J. Simple BERT Models for Relation Extraction and Semantic Role Labeling. 2019. [Электронный ресурс] URL: https://arxiv.org/pdf/1904.05255.pdf (дата обращения: 09.11.2021).
49. Tao Q., Luo X., Wang H. Enhancing relation extraction using syntactic indicators and sentential contexts. International Conference on Tools with Artificial Intelligence (ICTAI), Piscataway, NJ, pp. 574–580. 2019.
50. Ningthoujam Dh., Yadav Sh., Bhattacharyya P., Ekbal A. Relation extraction between the clinical entities based on the shortest dependency path based LSTM. 2019. [Электронный ресурс] URL: https://arxiv.org/pdf/1903.09941.pdf (дата обращения: 09.11.2021).
51. Nayak T., Ng H. Effective Attention Modeling for Neural Relation Extraction. Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL). Association for Computational Linguistics, Hong Kong, China, pp. 603–612. 2019. DOI: 10.18653/v1/K19-1056.
52. Li P., Mao K., Yang X., Li Q. Improving relation extraction with knowledge-attention. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Association for Computational Linguistics, Hong Kong, China, pp. 229–239. 2019. DOI: 10.18653/v1/D19-1022.
53. Soares L., FitzGerald N., Ling J., Kwiatkowski T. Matching the Blanks: Distributional Similarity for Relation Learning. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Florence, Italy, pp. 2895–2905. 2019. DOI: 10.18653/v1/P19-1279.
54. Ni J., Florian R. Neural Cross-Lingual Relation Extraction Based on Bilingual Word Embedding Mapping. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language

Processing (EMNLP-IJCNLP). Association for Computational Linguistics, Hong Kong, China, pp. 399–409. 2019. DOI: 10.18653/v1/D19-1038.

55. Papanikolaou Y., Roberts I., Pierleoni A. Deep Bidirectional Transformers for Relation Extraction without Supervision. Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019). Association for Computational Linguistics, Hong Kong, China, pp. 67–75. 2019. DOI: 10.18653/v1/D19-6108.
56. Tomar G.S., Bhatia P. Relation extraction using explicit context conditioning. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Vol. 1, pp. 1442–1447. 2019.
57. Bansal T., Verga P., Choudhary N., McCallum A. Simultaneously linking entities and extracting relations from biomedical text without mention-level supervision. In Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 34, No. 05, pp. 7407–7414. 2020.
58. Al-Aswadi F.N. Chan H.Y., Gan K.H. Extracting Semantic Concepts and Relations from Scientific Publications by Using Deep Learning. International Conference of Reliable Information and Communication Technology (IRICT 2020). Innovative Systems for Intelligent Health Informatics. Vol. 72, pp.374–383. 2020.
59. Shanchan W., Yifan H. Enriching pretrained language model with entity information for relation classification. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management. ACM, pp. 2361–2364. 2019.
60. Eberts M., Ulges A. Span-based Joint Entity and Relation Extraction with Transformer Pre-training. 24th European Conference on Artificial Intelligence, 2020.
61. Ji B., Yu J., Li Sh., Ma J., Wu Q., Tan Y., Liu H. Span-based Joint Entity and Relation Extraction with Attention-based Span-specific and Contextual Semantic Representations. Proceedings of the 28th International Conference on Computational Linguistics. International Committee on Computational Linguistics, Barcelona, Spain (Online), pp. 88–99. 2020. DOI: 10.18653/v1/2020.coling-main.8.
62. Huang W., Cheng X., Wang T., Chu W. BERT-Based Multi-Head Selection for Joint Entity-Relation Extraction. Natural Language Processing and Chinese Computing – 8th International Conference. Springer, Dunhuang, China, pp. 713–823. 2019. DOI: 10.1007/978-3-030-32236-6-65.
63. Miwa M., Bansal M. End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures. Proceedings of the 54th Annual Meeting of the Association for Computational

Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Berlin, Germany, pp. 1105–1116. 2016. DOI: 10.18653/v1/P16-1105.

64. Kui X., Yangming Z., Zhiyuan M., Tong R., Huanhuan Z., Ping H. Fine-tuning BERT for Joint Entity and Relation Extraction in Chinese Medical Text. 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 892–897. 2019. DOI: 10.1109/BIBM47256.2019.8983370.
65. Ryuichi T., Tianyang Z., Jiexi L., Minlie H. A Hierarchical Framework for Relation Extraction with Reinforcement Learning. The AAAI Conference on Artificial Intelligence, pp. 7072–7029. 2019.
66. Sevgili O., Shelmanov A., Arkhipov M., Panchenko A., Biemann C. Neural Entity Linking: A Survey of Models Based on Deep Learning. 2020. [Электронный ресурс] URL: https://arxiv.org/pdf/2006.00575.pdf (дата обращения: 09.11.2021).
67. 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, pp. 438–447. 2019.
68. Winkler W. E. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. Proceedings of the Section on Survey Research Methods. American Statistical Association, pp. 354–359. 2020.
69. Zwicklbauer S., Seifert Ch., Granitzer M. Robust and collective entity disambiguation through semantic embeddings. Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'16, pp. 425–434. 2016. DOI: 10.1145/2911451.2911535.
70. Cao Y., Hou L., Li J., Liu Z. Neural collective entity linking. Proceedings of the 27th International Conference on Computational Linguistics. Santa Fe, New Mexico, USA, pp. 675–686. 2018.
71. Bunescu R. C., Pasca M. Using encyclopedic knowledge for named entity disambiguation. Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics, pp. 9–16. 2006.
72. Yin X., Huang Y., Zhou B., Li A., Lan L., Jia Y. Deep Entity Linking via Eliminating Semantic Ambiguity With BERT. IEEE Access. vol. 7, pp. 169434–169445. 2019. DOI: 10.1109/ACCESS.2019.2955498.

73. 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. Proceedings of Text Analysis Conference 2009, pp. 102–114. 2009.
74. Zhang W., Su J., Tan C. L., Wang W. T. Entity linking leveraging: Automatically generated annotation. Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pp. 1290–1298. 2010.
75. Huang H., Heck L., Ji H. Leveraging deep neural networks and knowledge graphs for entity disambiguation. 2015. [Электронный ресурс] URL: https://arxiv.org/pdf/1504.07678.pdf (дата обращения: 09.11.2021).
76. Parravicini A., Patra R., Bartolini D., Santambrogio M. Fast and Accurate Entity Linking via Graph Embedding. Proceedings of the 2nd Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA), pp. 1–9. 2019. DOI: 10.1145/3327964.3328499.
77. Perozzi B., Al-Rfou R., Skiena S. DeepWalk: Online Learning of Social Representations. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.701–710. 2014. DOI: 10.1145/2623330.2623732.
78. Nedelchev R., Chaudhuri D., Lehmann J., Fischer A. End-to-End Entity Linking and Disambiguation leveraging Word and Knowledge Graph Embeddings. 2020. [Электронный ресурс] URL: https://arxiv.org/pdf/2002.11143.pdf (дата обращения: 09.11.2021).
79. Bordes A., Usunier N., Garcia-Duran A., Weston J., Yakhnenko O. Translating Embeddings for Modeling Multi-relational Data. Proceedings of the 26th International Conference on Neural Information Processing Systems, vol. 2, pp. 2787–2795. 2013.
80. Delpeuch A. OpenTapioca: Lightweight Entity Linking for Wikidata. 2019. [Электронный ресурс] URL: https://arxiv.org/pdf/1904.09131.pdf (дата обращения: 09.11.2021).
81. 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, pp. 2192–2203. 2020.
82. Gábor K., Buscaldi D., Schumann A., QasemiZadeh B., Zargayouna H., Charnois T.. SemEval-2018 Task 7: Semantic Relation Extraction and Classification in Scientific Papers. Proceedings of The 12th International Workshop on Semantic Evaluation. Association for

Computational Linguistics, New Orleans, Louisiana, pp. 679–688. 2018. DOI: 10.18653/v1/S18-1111.

83. Luan Y., He L., Ostendorf M., Hajishirzi H. Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction. Association for Computational Linguistics, Brussels, Belgium, pp. 3219–3232. 2018. DOI: 10.18653/v1/D18-1360.
84. Augenstein I., Das M., Riedel S., Vikraman L., McCallum A.. SemEval 2017 Task 10: ScienceIE – Extracting Keyphrases and Relations from Scientific Publications. Association for Computational Linguistics, Vancouver, Canada, pp. 546–555. 2017. DOI: 10.18653/v1/S17-2091.
85. QasemiZadeh B., Schumann A. The ACL RD-TEC 2.0: A Language Resource for Evaluating Term Extraction and Entity Recognition Methods. Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16). European Language Resources Association (ELRA), Portorož, Slovenia, pp.1862–1868. 2016.
86. Rosario B., Hearst M. Classifying Semantic Relations in Bioscience Texts. Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04). Barcelona, Spain, pp. 430–473. 2004. DOI: 10.3115/1218955.1219010.
87. Власова Н.А., Сулейманова Е.А., Трофимов И.В. Сообщение о русскоязычной коллекции для задачи извлечения личных имен из текстов // Труды конференции по компьютерной и когнитивной лингвистике TEL'2014 "Языковая семантика: модели и технологии". — Казань, 2014. — С. 36–40.
88. Starostin A., Bocharov V., Alexeeva S., Bodrova A., Chuchunkov A., Dzhumaev S., Efimenko 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. Российский государственный гуманитарный университет, Komp'juternaja Lingvistika i Intellektual'nye Tehnologii, pp. 702–720. 2016.
89. Hendrickx I., Kim S., Kozareva Z., Nakov P., Séaghdha D., Padó S., Pennacchiotti M., Romano L., Szpakowicz S.. SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals. Proceedings of the 5th International Workshop on Semantic Evaluation. Association for Computational Linguistics, Uppsala, Sweden, pp. 33–38. 2010.
90. Лопатин В. Толковый словарь современного русского языка. – М.: Эксмо, 2013. – 928 с. – (Библиотека словарей ЭКСМО).
91. Rose S., Engel D., Cramer N., Cowley W. Automatic keyword extraction from individual documents. Text mining: applications and theory, pp. 1–20. 2010.

92. Ratner A., Hancock B., Ré Ch. The Role of Massively Multi-Task and Weak Supervision in Software 2.0. 9th Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, Online Proceedings. ACM, New York, NY, USA. 2019.
93. Ivanin V., Artemova E., Batura T., Ivanov V., Sarkisyan V., Tutubalina E., Smurov I. RUREBUS-2020 Shared Task: Russian Relation Extraction for Business. In : Komp'juternaja Lingvistika i Intellektual'nye Tehnologii, pp. 416–431. 2020.
94. Yao Y., Ye D., Li P., Han X., Lin Y., Liu Zh., Liu Zh., Huang L., Zhou J., Sun M. DocRED: A Large-Scale Document-Level Relation Extraction Dataset. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Florence, Italy, pp. 764–777. 2019. DOI: 10.18653/v1/P19-1074.

Публикации автора

95. Бручес Е. П., Батура Т. В. Метод автоматического извлечения терминов из научных статей на основе слабо контролируемого обучения. Вестник НГУ. Серия: Информационные технологии. 2021 Т.19, №2. С. 5–16. DOI: 10.25205/1818-7900-2021-19-2-5-16
96. Бручес Е.П., Батура Т.В. Свидетельство о регистрации программ для ЭВМ № 2021611340 «Система автоматического извлечения терминов из научных текстов «Term Extractor». Дата регистрации: 26.01.2021.
97. Батура Т.В., Бручес Е.П., Паульс А.Е., Исаченко В.В., Щербатов Д.Р. Семантический анализ научных текстов: опыт создания корпуса и построения языковых моделей. Программные продукты и системы. 2021. Т. 34. № 1. С. 132–144. DOI: 10.15827/0236-235X.133.132-144
98. Мезенцева А. А., Бручес Е. П., Батура Т. В. Автоматическое связывание терминов из научных текстов с сущностями базы знаний. Вестник НГУ. Серия: Информационные технологии. Т.19, №2. с. 65–75. 2021. DOI: 10.25205/1818-7900-2021-19-2-65-75
99. Bruches E.P., Pauls A.E., Batura T.V., Isachenko V.V. Study of Methods for Entity Recognition and Relation Extraction in Scientific Texts. Science and Artificial Intelligence conference (SAIC-2020). 2020. p. 41–45. DOI: 10.1109/S.A.I.ence50533.2020.9303196
100. Крайванова В.А., Бручес Е.П., Минаков А.М., Анкудинов К.Л., Пчельников Д.В. Архитектура категоризатора событий в гетерогенном пространстве параметров // Ползуновский альманах, № 4, 2018, с.134–138.

101. Batura T.V., Bruches E.P. A combined approach to the problem of part-of-speech homonymy resolution in Russian texts. Proceedings of the International Russian Automation Conference (RusAutoCon 2018). September 9-16, 2018. pp. 4–9. DOI 10.1109/RUSAUTOCON.2018.8501718

102. Бручес Е.П., Крайванова В.А. О способе векторизации морфологической информации словоформы. Сборник научных трудов «Нечеткие системы и мягкие вычисления. Промышленные применения», г. Ульяновск, 2017. с. 232–239.

103. Бручес Е. П., Крайванова В. А. Снятие омонимии геолокаций на основе частоты встречаемости контекстов. Ползуновский альманах № 4, 2017, т. 3, с. 103–105.

104. Batura T.V., Bruches E.P., Strekalova S.E. A combined approach to part-of-speech homonymy resolution. Bulletin of the Novosibirsk Computing Center. Series: Computer Science. 2017. Is. 41. pp. 13–25.

105. Bruches E., Karpenko D., Krayvanova V. The Hybrid Approach to Part-of-Speech Disambiguation. International Conference on Analysis of Images, Social Networks and Texts (AIST 2016). 2016. pp. 21-26.

Приложение 1. Пример разметки корпуса

id	токен	nested_0	nested_1	nested_2	wiki_id	relation
0	Разработка	B-TERM				
1	ядра	I-TERM	B-TERM			
2	онтологической	I-TERM	I-TERM	B-TERM	Q324254:2,3	
3	модели	I-TERM	I-TERM	I-TERM		
4	,	O				
5	настраиваемой	O				
6	под	O				
7	предметную	B-TERM			Q2088941:7,8	
8	область	I-TERM				
9	Статья	O				
10	посвящена	O				
11	разработке	B-TERM				
12	ядра	I-TERM	B-TERM			
13	онтологической	I-TERM	I-TERM	B-TERM	Q324254:13,14	
14	модели	I-TERM	I-TERM	I-TERM		
15	в	O				
16	виде	O				
17	программной	B-TERM			Q2429814:17,18	
18	системы	I-TERM				

19	,	O				
20	настраиваемой	O				
21	под	O				
22	конкретную	O				
23	предметную	B-TERM			Q2088941:23,24	
24	область	I-TERM				
25	.	O				
26	Работа	O				
27	основана	O				
28	на	O				
29	теоретико	B-TERM			Q467606:29,30,31,32	USAGE(34)
30	-	I-TERM				
31	модельном	I-TERM				
32	подходе	I-TERM				
33	к	O				
34	представлению	B-TERM			Q3478658:34,35	
35	знаний	I-TERM	B-TERM			
36	.	O				
37	Для	O				
38	представления	B-TERM			Q3478658:38,39	
39	знаний	I-TERM	B-TERM			
40	используются	O				

41	фрагменты	O				
42	атомарных	B-TERM				USAGE(38)
43	диаграмм	I-TERM				
44	алгебраических	I-TERM	B-TERM		Q56312286:44,45	
45	систем	I-TERM	I-TERM			
46	и	O				
47	нечеткие	B-TERM				USAGE(38)
48	модели	I-TERM				
49	.	O				
50	Программная	B-TERM			Q2429814:50,51	
51	система	I-TERM				
52	разбита	O				
53	на	O				
54	модули	B-TERM			Q2663565:54	PART_OF(50)
55	.	O				
56	Базовые	O				
57	модули	B-TERM			Q2663565:57	
58	реализуют	O				
59	функциональность	O				
60	,	O				
61	необходимую	O				
62	для	O				
63	любой	O				

64	онтологической	B-TERM			Q324254:57	
65	модели	I-TERM				
66	.	O				
67	Например	O				
68	,	O				
69	проверку	O				
70	на	O				
71	непротиворечивость	B-TERM	B-TERM		Q1319773:71	
72	хранящихся	I-TERM				
73	знаний	I-TERM	B-TERM		Q9081:73	
74	.	O				
75	Расширение	O				
76	функциональности	O				
77	происходит	O				
78	через	O				
79	создание	O				
80	новых	O				
81	модулей	B-TERM			Q2663565:81	PART_OF(86)
82	и	O				
83	их	O				
84	добавление	O				
85	в	O				
86	систему	B-TERM			Q2429814:86	

87	.	O				
88	В	O				
89	работе	O				
90	приведен	O				
91	обзор	O				
92	существующих	O				
93	программных	O				
94	решений	O				
95	в	O				
96	области	O				
97	разработки	O				
98	онтологий	B-TERM			Q324254:98	
99	и	O				
100	онтологических	B-TERM			Q324254:100,101	
101	моделей	I-TERM				
102	.	O				
103	Описана	O				
104	структура	B-TERM			Q6671777:104	
105	ядра	B-TERM				
106	онтологической	I-TERM	B-TERM		Q324254:106,107	
107	модели	I-TERM	I-TERM			
108	,					
109	приведена					

110	архитектура	B-TERM			Q846636:110,111,112	
111	программного	I-TERM				
112	решения	I-TERM				

Приложение 2. Фрагмент матрицы переходов

P	Следующее состояние														
	ws	а	б	д	е	з	и	й	к	м	о	р	т	ц	я
p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	**p_1**	p_0	p_0	p_0	p_0	p_0
p_1	p_0	**p_6**	p_0	p_0	**p_2**	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0
p_2	p_0	p_0	p_0	p_0	p_0	p_0	p_0	**p_3**	p_0	p_0	p_0	p_0	p_0	p_0	p_0
p_3	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	**p_4**	p_0	p_0	p_0	p_0
p_4	p_0	p_0	p_0	p_0	p_0	**p_5**	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0
p_5	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0
p_6	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	**p_7**	p_0	p_0
p_7	p_0	p_0	p_0	p_0	p_0	p_0	**p_{12}**	p_0	p_0	p_0	p_0	**p_8**	p_0	p_0	p_0
p_8	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	**p_9**	p_0	p_0	p_0	p_0
p_9	p_0	p_0	p_0	p_0	p_0	p_0	**p_{10}**	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0
p_{10}	p_0	p_0	p_0	**p_{11}**	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0
p_{11}	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0
p_{12}	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	**p_{13}**	p_0
p_{14}	p_0	**p_{15}**	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0
p_{15}	**p_{16}**	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0
p_{16}	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	**p_{17}**
p_{17}	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	**p_{18}**	p_0	p_0	p_0	p_0	p_0	p_0
p_{18}	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	**p_{19}**	p_0	p_0	p_0	p_0

p_{20}	p_0	p_0	$\mathbf{p_{21}}$	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0
p_{21}	p_0	p_0	p_0	p_0	p_0	p_0	$\mathbf{p_{22}}$	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0
p_{22}	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0	p_0

Приложение 3. Лексико-синтаксические шаблоны для извлечения отношений

Отношение	Маркеры для коротких контекстов	Маркеры для длинных контекстов
CAUSE		вызвано вызывает дало в результате даёт в результате дает в результате привело к приводит к связан с связана с связано с улучшает улучшил улучшила улучшили улучшило ухудшает ухудшил ухудшила ухудшили ухудшило является причиной
COMPARE		больше в сравнении с лучше меньше по сравнению с

		сравнивается сравниваются сравнили сравнить хуже
ISA	- это	в том числе например относится представляет собой такая как такое как такой как является
PART_OF		содержит состоит из является частью
SYNONYMS	(или иначе	
TOOL		автоматизирует анализирует выполняет вычисляет изучает исследует решает создаёт создает управляет
USAGE	для	за счет

		за счёт использовалась использовались использовалось использовался используется используется для используются на основе основанная на основанное на основанные на основанный на применяется для с использованием с помощью

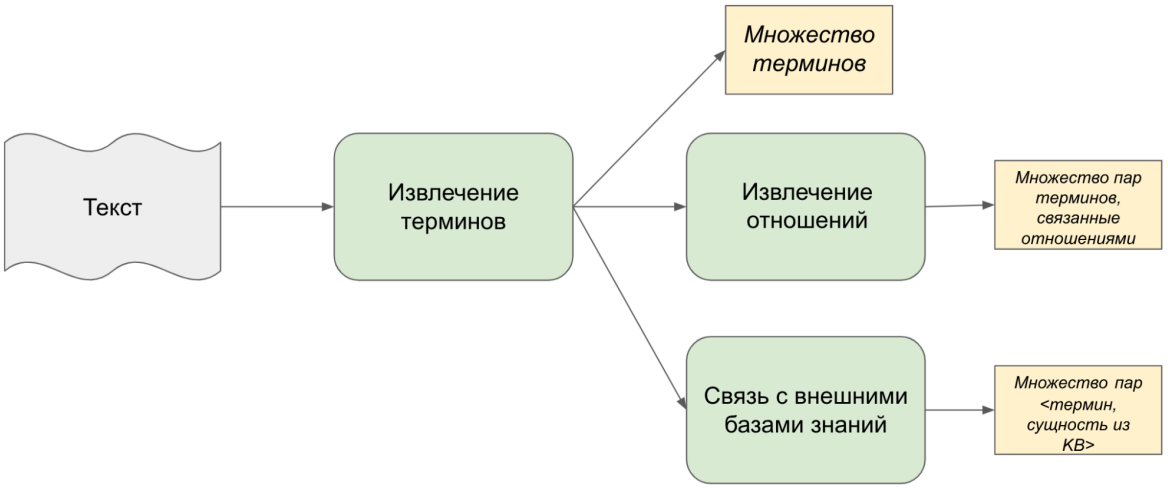
Приложение 4. Метрики извлечения отношений по сущностям

Эксперимент	Метрика	SciERC	RuSERRC
R-BERT: 10% примеров с None; для пары сущностей проверяются две связи	F1-micro	0.73	0.76
	F1-macro	0.33	0.20
	F1 USED-FOR	0.29	0.15
	F1 FEATURE-OF	0.13	-
	F1 HYPONYM-OF	0.42	0.13
	F1 PART-OF	0.08	0.07
	F1 COMPARE	0.20	0.00
	F1 CONJUNCTION	0.33	-
	F1 NONE	0.84	0.86
R-BERT: 10% примеров с None; для пары сущностей проверяется одна связь	F1-micro	0.57	0.65
	F1-macro	0.28	0.18
	F1 USED-FOR	0.22	0.13
	F1 FEATURE-OF	0.12	-
	F1 HYPONYM-OF	0.36	0.12
	F1 PART-OF	0.06	0.07
	F1 COMPARE	0.28	0.00
	F1 CONJUNCTION	0.25	-
	F1 NONE	0.71	0.79

R-BERT: 50% примеров с None; для пары сущностей проверяются две связи	F1-micro	0.89	0.94
	F1-macro	0.28	0.20
	F1 USED-FOR	0.20	0.04
	F1 FEATURE-OF	0.00	-
	F1 HYPONYM-OF	0.34	0.14
	F1 PART-OF	0.06	0.05
	F1 COMPARE	0.09	0.00
	F1 CONJUNCTION	0.31	-
	F1 NONE	0.94	0.97
R-BERT: 50% примеров с None; для пары сущностей проверяется одна связь	F1-micro	0.81	0.85
	F1-macro	0.33	0.23
	F1 USED-FOR	0.32	0.18
	F1 FEATURE-OF	0.11	-
	F1 HYPONYM-OF	0.42	0.16
	F1 PART-OF	0.05	0.10
	F1 COMPARE	0.20	0.00
	F1 CONJUNCTION	0.33	-
	F1 NONE	0.89	0.92
BertForSequenceClassification: 50% примеров с None; для пары сущностей проверяются две связи	F1-micro	0.67	0.63
	F1-macro	0.51	0.26
	F1 USED-FOR	0.65	0.17
	F1 FEATURE-OF	0.06	-

	F1 HYPONYM-OF	0.66	0.21
	F1 PART-OF	0.29	0.15
	F1 COMPARE	0.47	0.00
	F1 CONJUNCTION	0.69	-
	F1 NONE	0.73	0.76

Приложение 5. Схема работы системы извлечения информации



Приложение 6. Грамоты

XXII ВСЕРОССИЙСКАЯ КОНФЕРЕНЦИЯ
МОЛОДЫХ УЧЕНЫХ
ПО МАТЕМАТИЧЕСКОМУ МОДЕЛИРОВАНИЮ
И ИНФОРМАЦИОННЫМ ТЕХНОЛОГИЯМ

ДИПЛОМ

НАГРАЖДАЕТСЯ ПОБЕДИТЕЛЬ КОНКУРСА МОЛОДЫХ УЧЁНЫХ

БРУЧЕС ЕЛЕНА ПАВЛОВНА

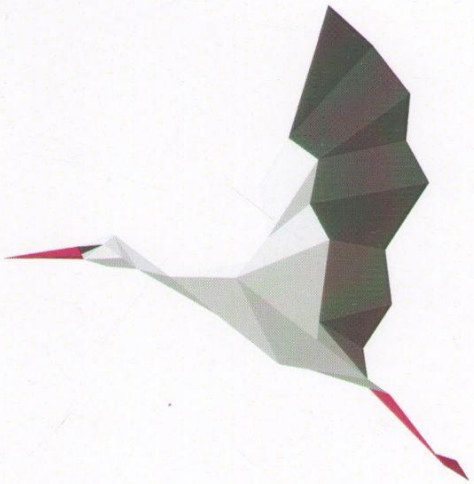
ЗА ДОКЛАД

ИЗВЛЕЧЕНИЕ ОТНОШЕНИЙ
ИЗ НАУЧНЫХ ТЕКСТОВ
НА РУССКОМ ЯЗЫКЕ

Председатель
программного комитета
академик Шокин Ю.И.

28 октября 2021 г.
г. Новосибирск

BEST TALK AWARD



This is to certify that

E. Bruches, D. Karpenko, V. Krayvanova

are (is) awarded for

the best talk in Natural Language Processing

at the 5th international conference on
Analysis of Images, Social networks, and Texts AIST'2016 http://aistconf.org
held in Yekaterinburg, Russia, April 7-9, 2016

AIST org. committee

exactpro
EXITUS ACTA PROBAT
London Stock Exchange Group

@mail.ru group

Центр ИТ

Приложение 7. Акты о внедрении

РОССИЯ
Новосибирск
Общество с ограниченной ответственностью
“Новые программные системы”
ИНН 5408254508, ОГРН 1075473011900
630090, г.Новосибирск, пр. Лаврентьева, д.6, оф.222

УТВЕРЖДАЮ
Директор ООО
«Новые программные системы»
_____Д.Н. Штокало
«08» октября 2021 г.

АКТ

о внедрении научно-исследовательских результатов диссертационной работы Бручес Елены Павловны по теме «Методы и алгоритмы распознавания и связывания сущностей для построения систем автоматического извлечения информации из научных текстов»

Настоящий акт подтверждает, что результаты диссертационного исследования по теме «Методы и алгоритмы распознавания и связывания сущностей для построения систем автоматического извлечения информации из научных текстов», полученные соискателем Бручес Еленой Павловной по специальности 05.13.17 – «Теоретические основы информатики» применяются в ООО «Новые программные системы» в процессе проведения научных исследований для анализа текстовой информации.

Бручес Е.П. разработаны и реализованы следующие методы и алгоритмы: алгоритм извлечения терминов из научных статей, основанный на частичном обучении; метод извлечения отношений между терминами, работающий в условиях недостаточного количества размеченных данных; алгоритм связывания терминов с базой знаний; метод оценки качества алгоритма связывания сущностей с внешней базой знаний с помощью новых метрик. Программное обеспечение может применяться для работы с текстами на русском языке.

Бручес Е.П. реализован обширный набор программных инструментов, предназначенный для поддержки проводимых исследований и представляющий практический интерес.

Члены комиссии

к.ф.-м.н. Д.С. Мигинский

к.б.н. Д.В. Антонец

«УТВЕРЖДАЮ»
директор Института систем информатики
им. А.П. Ершова СО РАН
д.ф.-м.н. _____А.Ю. Пальянов
«17» сентября 2021 г.

А К Т

о внедрении результатов диссертационного исследования

Настоящий акт подтверждает, что научные и практические результаты, полученные соискателем Бручес Еленой Павловной в ходе выполнения диссертационной работы по теме «Методы и алгоритмы распознавания и связывания сущностей для построения систем автоматического извлечения информации из научных текстов» на соискание ученой степени кандидата технических наук по специальности 05.13.17 – Теоретические основы информатики, используются в Лаборатории моделирования сложных систем федерального государственного бюджетного учреждения науки Институте систем информатики им. А.П. Ершова СО РАН.

Предложенные Е.П. Бручес методы и алгоритмы автоматического извлечения и связывания терминов и отношений из текстов на русском языке используются как встраиваемые компоненты при реализации различных проектов. Созданный программный комплекс представляет интерес для специалистов, занимающихся обработкой текстов, и применяется для анализа больших наборов данных с целью автоматического извлечения важной информации по перспективным научным направлениям и технологиям.

Заместитель директора
по научной работе ИСИ СО РАН
кандидат физико-математических наук А.В. Промский

МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное учреждение высшего образования «Новосибирский национальный исследовательский государственный университет» (Новосибирский государственный университет, НГУ)

ул. Пирогова, д. 1, Новосибирск, 630090
Тел. (383) 363-40-00. Факс (383) 330-42-80
Адрес в интернете: //www.nsu.ru
E-mail: rector@nsu.ru

07 СЕН 2021 № 3493/226
на №_____ от_____

«УТВЕРЖДАЮ»
Ректор НГУ, д.ф.-м.н., профессор
_____ М.П. Федорук
«___» _____ 2021 г.

АКТ
о внедрении в учебный процесс результатов диссертационной работы
Бручес Елены Павловны

Настоящий акт подтверждает внедрение научно-практических результатов, полученных аспиранткой Института систем информатики им. А.П. Ершова СО РАН Бручес Е.П. в ходе выполнения диссертационного исследования по теме «Методы и алгоритмы распознавания и связывания сущностей для построения систем автоматического извлечения информации из научных текстов», в учебный процесс на кафедре Фундаментальной и прикладной лингвистики Гуманитарного института Новосибирского государственного университета. Созданы конспекты лекций и методические разработки для практических занятий, включающие методы распознавания именованных сущностей, извлечения семантических отношений между ними и связывания сущностей с внешними базами знаний с использованием статистических алгоритмов и алгоритмов машинного обучения, а также программных реализаций соответствующих методов для русского языка.

Вышеупомянутые материалы используются в учебном курсе «Обработка естественных языков» и размещены в сети.

зав. кафедрой фундаментальной и прикладной лингвистики ГИ НГУ
д.филос.н., профессор
А.Н.Савостьянов

МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное учреждение высшего образования «Новосибирский национальный исследовательский государственный университет» (Новосибирский государственный университет, НГУ)

ул. Пирогова, д. 1, Новосибирск, 630090
Тел. (383) 363-40-00. Факс (383) 330-42-80
Адрес в интернете: //www.nsu.ru
E-mail: rector@nsu.ru

15 ОКТ 2021 № 4330/216
на №_____ от_____

«УТВЕРЖДАЮ»
Ректор НГУ, д.ф.-м.н., профессор
_____ М.П. Федорук
«____»_____ 2021 г.

АКТ
о внедрении результатов диссертационного исследования в учебный процесс

Настоящий акт подтверждает, что научные и практические результаты, полученные аспиранткой Института систем информатики им. А.П. Ершова СО РАН Бручес Еленой Павловной в ходе выполнения диссертационного исследования по теме “Методы и алгоритмы распознавания и связывания сущностей для построения систем автоматического извлечения информации из научных текстов” на соискание ученой степени кандидата технических наук по специальности 05.13.17 “Теоретические основы информатики” внедрены в учебный процесс механико-математического факультета Новосибирского государственного университета в рамках дисциплины “Natural language processing”. При этом подготовлены конспекты лекций и указания по выполнению практических занятий, включающие методы распознавания именованных сущностей, извлечения семантических отношений между ними и связывания сущностей с внешними базами знаний с использованием статистических алгоритмов и алгоритмов машинного обучения, а также программных реализаций соответствующих методов для русского языка.

Декан ММФ НГУ
д.ф.-м.н., профессор РАН И.В.Марчук

Приложение 8. Свидетельство о регистрации программы для ЭВМ

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2021611340

Система автоматического извлечения терминов из научных текстов «Term Extractor»

Правообладатели: ***Бручес Елена Павловна (RU), Батура Татьяна Викторовна (RU)***

Авторы: ***Бручес Елена Павловна (RU), Батура Татьяна Викторовна (RU)***

Заявка № **2021610288**
Дата поступления **13 января 2021 г.**
Дата государственной регистрации
в Реестре программ для ЭВМ ***26 января 2021 г.***

Руководитель Федеральной службы по интеллектуальной собственности

Г.П. Ивлиев