МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М.В. ЛОМОНОСОВА

На правах рукописи

Тихомиров Михаил Михайлович

Методы автоматизированного пополнения графов знаний на основе векторных представлений

Специальность 05.13.11 —
«Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей»

Автореферат
диссертации на соискание учёной степени
кандидата физико-математических наук

Москва — 2022

Работа выполнена в на кафедре Алгоритмических языков факультета Вычислительной математики и кибернетики МГУ имени М.В. Ломоносова.

Научный руководитель: **Лукашевич Наталья Валентиновна**
доктор технических наук

Официальные оппоненты: **Воронцов Константин Вячеславович,**
доктор физико-математических наук, профессор, кафедра Математических методов прогнозирования факультета Вычислительной математики и кибернетики МГУ имени М.В. Ломоносова, профессор

Суркова Анна Сергеевна,
доктор технических наук, доцент, кафедра Вычислительные системы и технологии НГТУ, ФГБОУ ВО «Нижегородский государственный технический университет им. Р.Е. Алексеева», профессор

Усталов Дмитрий Алексеевич,
кандидат физико-математических наук, обособленное подразделение ООО «Яндекс.Технологии» в г. Санкт-Петербург, руководитель группы исследований краудсорсинга

Защита диссертации состоится 16 июня 2022 г. в 14 часов на заседании диссертационного совета МГУ.01.19 при Московском государственном университете имени М.В. Ломоносова по адресу: 119991, Москва, ГСП-1, Ленинские горы, МГУ, д. 1 строение 52, факультет ВМК, аудитория №238.

E-mail: ilgova@cs.msu.su.

С диссертацией можно ознакомиться в отделе диссертаций научной библиотеки МГУ имени М.В.Ломоносова (Ломоносовский просп., д. 27) и на сайте ИАС «ИСТИНА»: https://istina.msu.ru/dissertations/454254400/.

Автореферат разослан «___» _____ 2022 года.

Ученый секретарь
диссертационного совета
МГУ.01.19,
кандидат физико-математических
наук Антонов Александр Сергеевич

Общая характеристика работы

Актуальность темы. Одним из основных направлений в области искусственного интеллекта является исследование моделей представления знаний (онтологий), которые предназначены для формализованного описания знаний о мире и предметной области. В приложениях автоматической обработки текстов особенно востребованы онтологии в виде семантических сетей. В последнее время активно исследуются подходы к применению так называемых графов знаний (Wikidata, Freebase, ConceptNet), в том числе в сочетании с подходами на основе машинного обучения. Графы знаний представляют собой семантические сети большого объема, в состав которых входит как система классов и подклассов понятий (таксономия), так и описания конкретных (именованных) сущностей. Отношения в графах знаний представлены в виде триплетов: субъект-отношение-объект.

Созданные онтологии и графы знаний необходимо уметь пополнять, поэтому часто обсуждается задача автоматического пополнения онтологий на основе больших текстовых коллекций, в которых содержится разнообразная информация и знания. Кроме того, важной задачей является создание онтологий для конкретных предметных областей.

Методы автоматического извлечения знаний из текстовых коллекций включают несколько этапов, такие как извлечение новых понятий, терминов, именованных сущностей, определение синонимов и вариантов терминов, извлечение отношений новых сущностей. Одной из важных задач в построении онтологий является построение таксономии классов, т.е. выявление отношений между более широкими (родовыми) классами и их более конкретными (видовыми) классами сущностей. В извлечении знаний из текстов данная задача ставится как извлечение гиперонимов - родовых слов для данного нового слова. Часто тестирование подходов к извлечению гиперонимов производится на на основе лексико-семантических ресурсов типа WordNet, содержащего представления значений более 100 тысяч слов английского языка в виде семантической сети.

Самыми первыми подходами к извлечению таксономических отношений из текстов были подходы на основе шаблонов, например "X — это Y", однако такие подходы обладают очень низкой полнотой, поскольку требуют присутствия соответствующих слов в одних и тех же предложениях в ограниченном количестве заданных конструкций.

Новые возможности для извлечения знаний из текстов появились на основе векторных представлений слов (эмбеддингов), которые формируются на основе контекстов упоминания слов. Сходство контекстов слов приводит к сходству их векторных представлений, что дает возможность автоматического определения семантической близости слов на основе текстовых коллекций.

Одних из первых успешных шагов в этом направлении была модель Word2Vec, разработанная в 2013 году. Дальнейшем развитием стали контекстуализированные векторные представления, которые формируют вектор для слов в зависимости от используемого контекста. Представителями таких подходов являются ELMO, BERT и др. Однако векторные модели не могут предсказывать тип отношения между словами с достаточной точностью, требуют их дополнительной обработки для извлечения интерпретируемых отношений. Тестирование подходов для извлечения таксономических отношений (отношений класс-подкласс), извлечение гиперонимов по текстовым коллекциям в рамках разных конференций, подходов, показывает, что качество извлечения знаний является недостаточно высоким, поэтому задача пополнения онтологий, графов знаний по текстам является актуальной.

Описанные выше проблемы, представляют интерес для исследований из-за того, что необходимы методы переноса подходов и ресурсов на новые предметные области, что делает данное исследование **актуальным**.

Степень разработанности темы. Отношения гиперонимии-гипонимии составляют основу структуры множества онтологий и графов знаний. Поэтому многочисленные исследования посвящены извлечению подобных отношений из текстовых коллекций. Гиперонимы могут быть извлечены с нуля, без каких-либо целевых ресурсов или таксономии, но качество таких подходов обычно достаточно низкое и не позволяет строить качественные таксономии, которые можно было бы использовать в рамках других задач. Также задача извлечения гиперонимов может ставиться как задача поиска гиперонимов для новых слов в существующей таксономии, то есть как задача обогащения или пополнения таксономии.

В 2016 г. задача по обогащению таксономии была организована как соревнование на семинаре SemEval (задача 14). Участники должны были связать слова с определениями для исправления гиперонимов в WordNet. Однако в реальных приложениях определения новых слов и их значений, скорее всего, отсутствуют. В 2020 году было организовано новое соревнование RUSSE'2020 по обогащению таксономии для русского RuWordNet, аналога WordNet для русского языка, содержащего представление значений слов для более 100 тысяч слов и выражений. Задача состояла в том, чтобы найти правильные гиперонимы из опубликованной версии RuWordNet для слов, добавленных в новой версии RuWordNet. Дальнейшим развитием набора данных RUSSE'2020 стал набор данных диахронических ворднетов (Diachronic wordnets), которые были созданы на основе английских и русских таксономий типа ворднет (WordNet). Эти наборы данных содержат новые слова, добавленные в более поздние версии ворднетов по сравнению с более ранними версиями, вместе с их гиперонимами в более старых версиях.

Разделение задачи пополнения графов знаний на а) пополнение таксономии абстрактными понятиями, и б) последующее пополнение именованными сущностями исследовалось в ряде работ. Например, авторы графа знаний AliCoCo таким образом развивали свой граф знаний для электронной коммерции. В их подходе, в частности, использовались методы извлечения именованных сущностей, как для пополнения графа знаний непосредственно именованными сущностями, так и для пополнения абстрактными понятиями. Но предложенный подход, помимо того, что содержит большое количество ручных действий, не может быть прямо применен из-за отсутствия описания ряда шагов системы и закрытости решения.

Задачи представления и пополнения знаний исследовались в работах Т.А. Гавриловой, В.Ф. Хорошевского, И.М. Зацмана, И.Л. Артемьевой, Ю.А. Загорулько, О.А. Невзоровой, С.О. Кузнецова. Задачи извлечения знаний из текстов, а также использования векторных представлений для определения семантических отношений между словами исследовались в работах таких исследователей как T. Mikolov, А.И. Панченко, Е.И. Большакова, Н.Э. Ефремова, Д.А. Усталов, П.И. Браславский.

Целью работы является исследование и разработка методов пополнения графов знаний новыми понятиями и именованными сущностями. Для достижения поставленной цели необходимо решить следующие **задачи**:

1. Исследовать существующие подходы к задаче пополнения графов знаний новыми понятиями и именованными сущностями,
2. Разработать методы пополнения таксономии графа знаний новыми понятиями и именованными сущностями,
3. Исследовать возможности адаптации графа знаний на конкретную предметную область, используя разработанные подходы,
4. Реализовать систему для автоматизированного пополнения графа знаний новыми понятиями и именованными сущностями.

Научная новизна:

1. Разработан и реализован метод пополнения таксономии графа знаний с использованием мета-векторных представлений. Исследована применимость разработанного метода на русском и английском языках, в общей области и конкретной предметной области информационной безопасности,
2. Разработан новый подход к порождению псевдоразметки для задачи извлечения именованных сущностей,
3. Разработан новый подход к задаче извлечения именованных сущностей в области информационной безопасности для русского языка с использованием псевдоразметки, двухэтапного обучения и специализированной языковой модели в области компьютерной безопасности RuCyBERT,

4. Реализована автоматизированная программная система для пополнения графа знаний новыми понятиями и именованными сущностями.

Теоретическая и практическая значимость. Теоретическая значимость работы состоит в том, что исследованы различные способы комбинирования векторных представлений слов и показано, что комбинация представлений с помощью автокодировщиков с учетом дополнительной информации о задаче приводит к улучшению качества векторных представлений, что в свою очередь приводит к улучшению качества решения целевой задачи.

Практическая значимость работы состоит в разработке и реализации подходов к пополнению таксономии графа знаний новыми понятиями и к извлечению именованных сущностей в предметной области информационной безопасности. Разработанные методы позволяют пополнять графы знаний как абстрактными понятиями, так и именованными сущностями. Подход показал свою работоспособность не только на общей предметной области, но и на конкретной предметной области информационной безопасности. Разработанные методы могут использоваться в автоматизированных системах. Разработанные подходы по пополнению таксономии новыми понятиями показали наилучший результат на рассмотренных наборах данных, метод для адаптации модели BERT для задачи извлечения именованных сущностей в области информационной безопасности для русского языка показал наилучшее качество на описанном наборе данных.

Методология и методы исследования. Для решения поставленных задач использовались элементы теории вероятностей, методы машинного обучения, математической статистики, методы построения векторных моделей на основе дистрибутивной семантики и методы построения мета-векторных представлений. При разработке использовались методы объектно-ориентированного программирования, язык Python.

Основные положения, выносимые на защиту:

1. Комбинированный подход к задаче пополнения таксономии на основе шаблонов и векторных представлений слов,
2. Комбинированный подход к задаче пополнения таксономии на основе мета-векторных представлений,
3. Метод получения псевдоразметки для задачи извлечения именованных сущностей,
4. Подход к извлечению именованных сущностей с использованием псевдоразметки, двухэтапного обучения модели RuCyBERT,
5. Автоматизированная программная система для пополнения таксономии новыми словами.

Достоверность полученных результатов обеспечивается проведенными экспериментами, открытым кодом реализованных методов и подходов, обоснованием принимаемых решений, публикациями в рецензируемых журналах и апробацией на российских и международных конференциях.

Апробация работы. Основные результаты работы докладывались на:

1. Text, Speech, and Dialogue 22nd International Conference, TSD 2019, Ljubljana, Slovenia, September 11–13, 2019,
2. Ломоносовские чтения 2020 - Секция вычислительной математики и кибернетики, Москва, Россия, 2020,
3. International Conference on Computational linguistics and intellectual technologies Dialog-2020, Москва, Россия, 17-20 июня 2020,
4. International Conference on Applications of Natural Language to Information Systems (NLDB-2020), Saarbruken, Germany, June 24-26, 2020,
5. International Conference on Computational Linguistics and Intellectual Technologies Dialogue 2021, Москва, Россия, 16-19 июня 2021,
6. XXIII "Data Analytics and Management in Data Intensive Domains"conference (DAMDID), Moscow, Россия, 26-29 октября 2021,
7. XII Международная научная конференция «Интеллектуальные системы и компьютерные науки», Москва, МГУ имени М.В. Ломоносова, Россия, 29 ноября - 3 декабря 2021.

Личный вклад. Все представленные в диссертации результаты получены лично автором. Подготовка части материалов к публикации проводилась совместно с соавторами, причем вклад диссертанта был определяющим. В работах [1—3] Н.В. Лукашевич принадлежит постановка задачи пополнения таксономии графа знаний, а также предоставление наборов данных. В работах [4—6] Б.В. Доброву принадлежат рекомендации к методологии исследований и постановка задачи, Н.В. Лукашевич предоставила набор данных для задачи извлечения именованных сущностей, списки дескрипторов, а также сформулировала идею о пополнении тренировочных данных за счет методов псевдоразметки. В работе [7] автор проводил вычислительный эксперимент, а идея, постановка и анализ результатов принадлежат Н.В. Лукашевич. В работе [8] автору принадлежат все эксперименты с использованием предложенного автором алгоритма с использованием мета-векторных представлений, а И.А. Никишиной обучение графовых векторных представлений, визуализация результатов, формирование набора данных, другие соавторы участвовали в постановке задачи, анализе результатов.

Публикации. Основные положения и выводы диссертационного исследования в полной мере изложены в 9 научных работах [1—9], в том

числе в 8 публикациях в рецензируемых научных изданиях [1—8], определенных п. 2.3 «Положения о присуждении ученых степеней в Московском государственном университете имени М.В.Ломоносова».

Объем и структура работы. Диссертация состоит из введения, 4 глав и заключения. Полный объём диссертации составляет 119 страниц, включая 19 рисунков и 32 таблицы. Список литературы содержит 127 наименования.

Содержание работы

Во **введении** дано краткое описание исследуемой в работе задачи, обосновывается актуальность, приводится обзор научной литературы по изучаемой проблеме, формулируется цель, ставятся задачи работы, излагается научная новизна и практическая значимость представляемой работы.

Первая глава посвящена обзору основных задач, исследуемых в диссертации, а также методов и подходов к решению этих задач. В главе описывается важность и применимость векторных представлений для автоматической обработки текстов, основные модели и методы обучения моделей.

В **разделе 1.1** приводится основная идея, и описываются классические методы для построения статических векторных представлений слов, таких как PPMI SVD, word2vec, fasttext, glove. В **разделе 1.2** описывается идея кодирования графов в векторное представление, польза таких представлений и способы использования. Рассматриваются основные подходы к кодированию вершин графов, как классические, так и современные, использующие нейросетевые архитектуры. В **разделе 1.3** вводится понятие мета-векторных представлений с описанием основных подходов к их построению. Приводятся примеры использования подобных методов. В **разделе 1.4** описываются контекстуализированные векторные представления. Рассматриваются их отличия от статических векторных моделей, а также ключевые подходы к их построению, такие как ELMO и BERT.

В **разделах 1.5-1.6** приводится обзор задач пополнения таксономии новыми понятиями и извлечения именованных сущностей. Рассматриваются способы использования векторных представлений к решению данных задач.

Вторая глава посвящена исследованию задачи пополнения таксономии графа знаний новой лексикой. В главе описываются как существующие методы, так и разработанные в рамках диссертации подходы.

В **разделе 2.1** приводится постановка задачи, и формально вводится понятие таксономии. Таксономии составляют основу многих графов знаний и организуют понятия (также называемые концептами) через отношение

частичного порядка является или гипероним-гипоним. Гипероним является вышестоящим в таксономии для гипонима понятием, например, кошка является также и животным. Таксономии представляют собой ориентированный граф без циклов, а само отношение гиперонимии является при этом транзитивным и антисимметричным, то есть, если класс Б является подклассом класса А, то все подклассы Б тоже являются подклассами класса А. Подобная иерархия необходима для использования в процедуре логического вывода, который в свою очередь необходим при анализе текстов на естественном языке. Более формально отношение R называется отношением частичного порядка на некотором множестве M, если оно удовлетворяет следующим свойствам:

1. Рефлексивность. $\forall a \in M : (aRa)$,
2. Транзитивность. $\forall a,b,c \in M : (aRb) \Lambda (bRc) \Rightarrow (aRc)$,
3. Антисимметричность. $\forall a,b \in M : (aRb) \Lambda (bRa) \Rightarrow a = b$.

Для того, чтобы упростить создание и адаптацию графов знаний на новые предметные области, в работе исследуются методы пополнения существующих таксономий новыми понятиями.

Пусть имеется:

– Существующая таксономия графа знаний,
– Набор новых понятий (слов),
– Коллекция текстов из рассматриваемой предметной области (опционально).

Требуется ввести новые слова в существующую таксономию. Для этого для каждого слова формируется упорядоченный список кандидатов-концептов из таксономии в порядке убывания вероятности. Рассматриваются две основные постановки: а) пополнение таксономии из общей области новыми словами, также принадлежащие общей области, и б) пополнение таксономии из общей области новыми словами из специальной предметной области. Исследование проводится на примере области информационной безопасности.

Задачу пополнению таксономии, можно также разделить на 3 типа: а) пополнение таксономии только абстрактными сущностями (без имен), б) пополнение таксономии только именованными сущностями, в) смешанный вариант.

Также в разделе приводится описание набора ресурсов, который обычно доступен при решении поставленных задач, см. Рис 1.

В **разделе 2.2** описываются разработанные в рамках диссертации методы и подходы к решению задачи пополнения таксономии новыми словами. В работе предлагается два подхода:

1. Комбинированный подход на основе шаблонов и векторных представлений слов,
2. Комбинированный подход на основе мета-векторных представлений.

Рис. 1 — Ресурсы в задаче пополнения таксономии.

В **разделе 2.2.1** описывается комбинированный подход на основе шаблонов и векторных представлений слов. Подход направлен на использование только корпуса текстов, из которого новые слова и отбирались. Метод использует следующие ресурсы:

- Дистрибутивное (векторное) представление анализируемых слов и словосочетаний, обученные на общей предметной области,
- Лингвистические шаблоны для гиперонимов и ко-гипонимов (слов, имеющих общий гипероним),
- Специальная обработка именованных сущностей для исключения их контекстов из рассмотрения,
- Применение классификатора на основе контекстуализированной векторной модели BERT для подтверждения кандидатов в гиперонимы.

Таким образом, данный подход можно отнести к пополнению таксономии только абстрактными понятиями на общей предметной области.

Помимо этого, при расчете векторных моделей была произведена специальная обработка именованных сущностей, с целью исключения их контекстов из рассмотрения для формирования векторных представлений. В работах других исследователей было обнаружено, что именованные сущности могут включать в себя общие слова и таким образом искажать контекст обычных слов.

На первом шаге алгоритма для всех целевых (новых) слов были рассмотрены 100 наиболее близких слов-кандидатов на основе косинусной близости между векторами слов. Отобранные слова должны присутствовать в таксономии. Затем идет шаг формирования списка концептов-кандидатов. Для каждого слова-кандидата извлекались из таксономии его гиперонимы и гиперонимы второго порядка (со штрафным весом). Каждый концепт мог быть добавлен несколько раз. Поэтому для каждого концепта кандидата фиксировалось, сколько раз он был добавлен (count) и на основе каких значений близости между целевым словом и словом-кандидатом, формируя таким образом список значений близости (cos_sim_list). Кандидаты были ранжированы по следующей формуле

$$base\_score = mean(cos\_sim\_list) * log_2(1 + count) * \alpha, \quad (1)$$

Следующим шагом алгоритма был учет шаблонов. Соответствие шаблонам проверялось для каждой пары "целевое слово" - "слово кандидат" из списка наиболее близких слов по векторной модели. Появление соответствующей пары в шаблоне увеличивало вес близости слов-кандидатов целевому слову, так как сопоставление шаблону является дополнительным доказательством семантического сходства.

Все сопоставления шаблонам были автоматически извлечены с использованием регулярных выражений на корпусе News2017 (8 млн. документов). Были рассмотрены два типа шаблонов:

- Шаблоны ко-гипонимов, успешное сопоставление с которыми приводит к увеличению веса соответствующих концептов,
- Шаблоны гиперонимов, успешное сопоставление с которыми приводит как к увеличению веса концепта-гиперонима, так и к дополнительному включению в список кандидатов прямого синонима слова-кандидата, а не только его гиперонима.

В Таблицах 1, 2 приведены образцы шаблонов, где X - целевое слово, Y - слово-кандидат, а W - любое слово. Также для всех шаблонов допускаются варианты, когда X и Y меняются местами.

Таблица 1 — Примеры шаблонов ко-гипонимов

Шаблон	Пример
X, W, Y	кошка, W, собака
X, Y	кошка, собака
X и Y	кошка и собака
X или Y	кошка или собака

Модифицированная формула с учетом шаблонов выглядит следующим образом:

$$upd\_pattern\_score = base\_score * (1 + hyper\_hit) * (1 + \frac{2 * co\_hypo\_count}{one\_sent\_count + 2}) \quad (2)$$

Таблица 2 — Примеры шаблонов гиперонимов

Шаблон	Пример
Y (X	животное (кошка
X - Y	кошка - животное
X - это Y	кошка - это животное
X, W и еще один Y	кошка, W и еще одно животное
X и W - это тип Y	кошка и W - это тип животных

, где hyper_hit - факт сопоставления с шаблоном гиперонимов, co_hypo_count - частота сопоставления с шаблоном ко-гипонимов в корпусе текстов, one_sent_count - частота попадания целевого слова и слова кандидата в одно предложение в корпусе текстов.

Последним шагом алгоритма был учет предсказания классификатора на основе контекстуализированной векторной модели BERT. Для задачи предсказания гиперонимов:

- Задача рассматривается как задача бинарной классификации,
- На вход модели BERT подается пара слов (или многословных выражений) в следующем виде:
  - [CLS] word1 [SEP] word2 [SEP],
  - Пример после токенизации:
    [CLS] кош ##ечка [SEP] домашнее животное [SEP];

  Если word2 является гипернимом word1, то класс примера равен 1, в противном случае - 0,
- Данные для обучения были созданы, используя RuWordNet. Для каждого положительного примера были добавлены три отрицательных примера. Отрицательные примеры были равномерно выбраны из: случайных концептов, гиперонимов второго порядка, гипонимов и гипонимов гиперонимов.

Для каждого концепта-кандидата вычислялась его вероятность быть гиперонимом и максимум из вероятностей того, что гипонимы концепта являются гиперонимом целевому слову. Результирующая вероятность для концепта вычислялась следующим образом:

$$bert\_prob = 0.6 * syn\_bert\_prob + 0.4 * max\_hyp\_syn\_bert\_prob \quad (3)$$

, где syn_bert_prob - вероятность концепта кандидата быть гиперонимом, max_hyp_syn_bert_prob - максимальная вероятность среди всех гипонимов.

Затем окончательная модифицированная формула ранжирования выглядит следующим образом:

$$synset\_score = upd\_pattern\_score * (1 + bert\_prob) \quad (4)$$

В **разделе 2.2.2** описывается комбинированный подход на основе мета-векторных представлений. В основе подхода лежит использование

векторных представлений слов. При обработке нового слова на первом этапе алгоритма происходит поиск наиболее близких слов на основании косинусной близости в векторном пространстве, затем этот список фильтруется с требованием, чтобы оставшиеся в нем слова присутствовали в таксономии, и из них оставляется N слов. На следующем этапе для каждого слова получившегося списка из таксономии извлекаются а) его концепты б) гиперонимы концептов в) гиперонимы второго порядка. Извлеченные понятия добавляются в новый список, фиксируя при этом, каким образом был добавлен концепт (как прямой концепт, его гипероним или гипероним второго порядка). Получившийся список $C_{cand} \in C$, где C это множество всех концептов, представляет собой усеченное пространство для поиска концептов, которые могут являться гиперонимом нового слова.

На следующем этапе алгоритма для концептов-кандидатов вычисляются характеристики-признаки для использования их в классификаторе. Расчетные характеристики следующие:

- Максимальное, минимальное и среднее значение близости между целевым словом и синонимами в концепте-кандидате,
- Значения близости между целевым словом и синонимами в гипонимах концепта-кандидата. Сначала рассчитываются максимальное, минимальное и среднее значения для каждого гипонима, затем вычисляются максимальное, минимальное и среднее значения близости среди всех концептов-гипонимов,
- Позиционный признак (0, 1, 2): если кандидат добавлен как прямой концепт (0) или гипероним (1) или гипероним второго порядка (2). Если концепт был добавлен несколько раз, то берется минимум, максимум и среднее,
- Количество раз, сколько концепт попал в список кандидатов c, и его логарифм $log_2(2 + c)$.

После описанной обработки, используя алгоритм логистической регрессии, производится предсказание того, является ли обрабатываемый концепт гиперонимом или нет. Таким образом, каждый концепт-кандидат получает свою вероятность, на основании которой производится ранжирование итогового списка.

Также предлагается использовать как простые мета-векторные представления, такие как конкатенацию исходных векторов и SVD над конкатенацией, так и два варианта автокодировщиков: конкатенированные автоматически закодированные мета-вектора (CAEME) и усредненные автоматически закодированные мета-вектора (AAEME)[1].

Пусть есть два исходных векторных представления $s_1(w)$ и $s_2(w)$, их кодировщики $E_1(w)$ и $E_2(w)$ и их декодировщики $D_1(w)$ и $D_2(w)$. Мета-вектор $m(w)$ в CAEME строится как L_2 -нормализованная конкатенация

[1]Bollegala D., Bao C. Learning word meta-embeddings by autoencoding //Proceedings of the 27th international conference on computational linguistics. – 2018. – С. 1650-1661.

двух закодированных исходных векторов $E_1(s_1(w))$ и $E_2(s_2(w))$:

$$m(w) = \frac{E_1(s_1(w)) \oplus E_2(s_2(w))}{||E_1(s_1(w)) \oplus E_2(s_2(w)||_2} \tag{5}$$

,где $\oplus$ - это операция конкатенации.

Кодировщики и декодировщики $E_i(w)$ $D_i(w)$ представляют собой полносвязные слои нейронной сети с ReLU активацией на выходе. В случае с кодировщиками перед полносвязным слоем также расположен dropout слой.

В кодировщике CAEME размерность пространства мета-векторов — это сумма размерностей исходных векторных моделей. Кодировщик AAEME можно рассматривать как вариант кодировщика CAEME, где мета-вектор вычисляется путем усреднения двух закодированных векторов вместо их конкатенации. Усреднение дает возможность избежать увеличения размерности мета-векторов. Кодировщик AAEME вычисляет мета-вектор слова w из двух исходных векторов $s_1(w)$ и $s_2(w)$ как L_2 - нормализованную сумму двух закодированных векторных представлений $E_1(s_1(w))$ и $E_2(s_2(w))$.

$$m(w) = \frac{E_1(s_1(w)) + E_2(s_2(w))}{||E_1(s_1(w)) + E_2(s_2(w))||_2} \tag{6}$$

Декодировщики CAEME и AAEME восстанавливают исходные вектора из одного и того же мета-пространства $m(w)$, тем самым неявно используя как общую, так и дополнительную информацию в исходных векторах:

$$\begin{aligned} \hat{s}_1(w) &= D_1(m(w)) \\ \hat{s}_2(w) &= D_2(m(w)) \end{aligned} \tag{7}$$

Целевая функция обучения автокодировщиков приведена в 8. Функция f может быть любым расстоянием или мерой подобия, например MSE, KL-дивергенцией или косинусным расстоянием. Коэффициенты λ_1 и λ_2 могут использоваться для взвешивания функций ошибок для различных векторных представлений.

$$Loss_w(E_1, E_2, D_1, D_2) = \sum_w (\lambda_1 f(s_1(w), \hat{s}_1(w)) + \lambda_2 f(s_2(w), \hat{s}_2(w))) \tag{8}$$

Совместное обучение E_1, E_2, D_1, D_2 минимизирует общую ошибку реконструкции векторов.

Вместе с обучением модели реконструировать вектора, предлагается добавить ограничения на обучаемые мета-представления, используя дополнительную функцию потерь - функцию потерь триплетов (triplet loss).

Функция потерь триплетов (triplet loss) — это функция потерь для алгоритмов машинного обучения, где некоторый базовый пример (anchor) сравнивается с положительным и отрицательным примерами. Цель - минимизация разницы расстояний между базовым и положительным примерами и базовым и отрицательным. Применительно к описанной задаче идея состоит в том, чтобы поощрять модель использовать близость слов, которые семантически связаны по таксономии. Для этого требуется, чтобы схожесть слова к случайному слову из его окружения C_{neib} была выше, чем к случайно выбранному слову из таксономии.

1. Для каждого слова, присутствующего в таксономии, составляется список C_{neib} семантически связанных слов из синонимов, гипонимов и гиперонимов,
2. На каждой эпохе обучения случайным образом выбирается K положительных примеров из C_{neib} и K отрицательных примеров из словаря,
3. Если обрабатываемое слово не представлено в таксономии, то зашумленные вариации исходного вектора рассматриваются как положительные примеры,
4. Затем рассчитывается функция потерь триплетов, и итоговая функция потерь выглядит следующим образом: $\alpha * loss + (1 - \alpha) * triplet\_loss$, где α - параметр модели.

Предложенный подход к пополнению таксономии схематически представлен на Рис. 2.

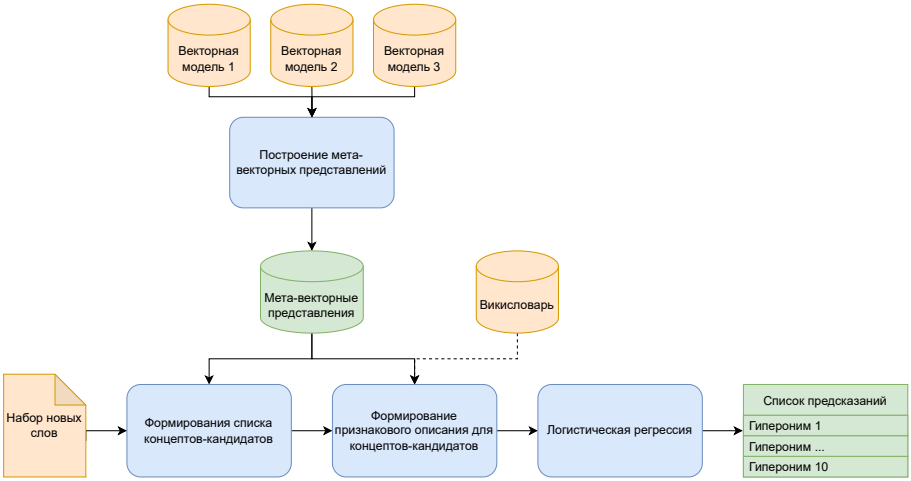

Рис. 2 — Схема работы комбинированного подхода на основе мета-векторных представлений.

Также в разделе изложен примененный способ учета внешнего словаря Викисловарь для решения поставленной задачи.

В **разделе 2.3** описываются наборы данных, на которых происходило тестирование разработанных в диссертационной работе подходов. Использовались такие наборы данных как RUSSE2020, Diachronic wordnets и набор данных OENTCyber [2], предложенный в рамках работы над диссертацией. OENTCyber создавался с целью изучить возможности разработанных алгоритмов при адаптации таксономии на конкретную предметную область. OENTCyber, в отличие от RUSSE2020 и Diachronic wordnets, содержит не только слова из общей предметной области, но и из конкретной предметной области информационной безопасности.

В **разделе 2.4** представлена информация о проведенных экспериментах. Сначала описываются основные используемые метрики, на основании которых происходило сравнение подходов. После чего последовательно изложены сами проведенные на описанных наборах данных эксперименты, включая постановку экспериментов, используемые ресурсы, результаты и анализ.

В **разделе 2.4.1** описываются используемые меры оценки, такие как MAP и MRR.

$$\begin{gathered} MAP = \frac{1}{N}\sum_{i=1}^{N} AP_i; \\ AP_i = \frac{1}{M}\sum_{i}^{n} prec_i \times I[y_i = 1], \end{gathered} \tag{9}$$

$$MRR = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{rank_i}, \tag{10}$$

В **разделе 2.4.2** описываются эксперименты на наборе данных RUSSE'2020. В рамках экспериментов на наборе данных RUSSE'2020 ставилась цель извлечь максимум информации из предоставленного организаторами набора текстовых данных News2017. На данном наборе данных использовался **комбинированный подход на основе шаблонов и векторных представлений слов** к предсказанию гиперонимов. По этой причине использовались только векторные представления, обученные на этом текстовом корпусе.

В **разделе 2.4.3** описываются эксперименты на наборе данных Diachronic wordnets. При работе над набором данных Diachronic wordnets использовался **комбинированный подход на основе мета-векторных представлений** к пополнению таксономии. Качество подхода оценивалось с использованием различных векторных представлений: fastText, word2vec, GloVe. Также были исследованы различные подходы к построению мета-векторных представлений: конкатенация, SVD поверх конкатенации, CAEME, AAEME. Косинусное расстояние использовалась как функция потерь для подходов AEME. Проверялись варианты и комбинации между MSE, KL дивергенцией и косинусным расстоянием, и последний

вариант показал себя наилучшим образом для данной задачи. Итоговые результаты на русскоязычной части данных приведены в Таблице 3.

Таблица 3 — MAP для методов обогащения таксономии для наборов данных на русском языке. Цифры, выделенные **жирным шрифтом**, показывают лучшую модель в категории, подчеркнутые числа обозначают лучший результат среди всех моделей.

Метод	Существительные	
	Не ограниченный набор	Ограниченный набор
Векторные модели слов		
fastText	**0.419±0.001**	**0.572±0.005**
word2vec	0.296±0.002	0.569±0.005
Мета-векторные представления на векторных моделях слов		
concat (*words*)	0.422±0.001	0.589±0.005
SVD (*words*)	0.461±0.001	**0.600±0.005**
CAEME (*words*)	0.400±0.001	0.561±0.005
AAEME (*words*)	0.456±0.001	0.582±0.005
CAEME triplet loss (*words*)	0.449±0.001	0.581±0.005
AAEME triplet loss (*words*)	**0.474±0.001**	0.593±0.006
Графовые векторные модели		
GCN	0.183±0.001	0.306±0.005
GraphSAGE	0.176±0.001	0.348±0.005
TADW	**0.417±0.001**	**0.562±0.005**
node2vec (top-5 fastText associates)	0.343±0.002	0.477±0.005
Мета-векторный представления на комбинации векторных моделей слов и графовых векторных моделях		
SVD (*words* + node2vec)	0.367±0.001	0.521±0.005
CAEME (*words* + node2vec)	0.370±0.001	0.533±0.005
AAEME (*words* + node2vec)	0.373±0.001	0.529±0.005
SVD (*words* + TADW)	**0.469±0.001**	**0.604±0.006**
CAEME (*words* + TADW)	0.429±0.001	0.571±0.005
AAEME (*words* + TADW)	0.461±0.001	0.584±0.005
SVD (*words* + GCN)	0.395±0.001	0.554±0.005
CAEME (*words* + GCN)	0.389±0.001	0.544±0.005
AAEME (*words* + GCN)	0.386±0.001	0.545±0.006
SVD (*words* + GraphSAGE)	0.410±0.001	0.603±0.005
CAEME (*words* + GraphSAGE)	0.321±0.001	0.541±0.005
AAEME (*words* + GraphSAGE)	0.409±0.001	0.577±0.006
Предыдущие подходы		
WBSR (Top-1 RUSSE'2020 для существительных)	**0.393±0.002**	**0.552±0.005**
WBSR, без использования поисковых технологий	0.369±0.002	0.497±0.005
Top-1 RUSSE'2020 для глаголов:	0.288±0.001	0.418±0.006
hypo2path	0.061±0.000	0.097±0.002
hypo2path rev	0.246±0.001	0.342±0.006
hypo2path rev transformer	0.234±0.001	0.331±0.004

В **разделе 2.4.4** описываются эксперименты на наборе данных OENTCyber. В качестве подхода использовался **комбинированный подход на основе мета-векторных представлений** к пополнению таксономии, но помимо использованных ранее двух "внешних" векторных моделей

word2vec и fasttext, дополнительно были обучены две векторные модели (word2vec и fasttext соответственно) на корпусе текстов по информационной безопасности - 500 тысяч текстов (далее "внутренние").

Исследовались возможности работы обученных моделей на конкретной предметной области, и также возможность совмещения их с более мощными моделями, так как 500 тысяч текстов является небольшим числом, по сравнению с тем количеством документов, на которых обучались "внешние" модели word2vec и fasttext. Ставилась серия из трех экспериментов:

- Использование только "внешних" векторных моделей,
- Использование только "внутренних" векторных моделей,
- Комбинирование "внешних" и "внутренних" векторных моделей.

Итоговые результаты комбинирования "внешних" и "внутренних" векторных моделей представлены в Таблице 4.

Таблица 4 — Расширение OEHT-lite: комбинация внутренних и внешних моделей

Метод	MAP	MRR
fastText внутр.	0.277	0.317
word2vec внутр.	0.277	0.316
fastText внешн.	0.362	0.407
word2vec внешн.	0.375	0.421
concat	0.386	0.434
SVD	0.387	0.433
CAEME	0.362	0.407
CAEME triplet	0.408	0.456
AAEME	0.414	0.463
AAEME triplet	**0.427**	**0.479**

В **разделе 2.5** приводятся основные выводы по главе, сделанные на основании проведенных исследований. Полученные результаты позволяют сделать следующие выводы:

- Векторные модели, обученные на небольшой коллекции документов из предметной области, дают весьма низкие результаты, по сравнению с более крупными моделям, которые обучались на корпусах на порядки больших по размеру,
- Использование мета-векторных представлений позволяет поднять качество работы алгоритма,
- Комбинирование предметных векторных моделей с внешними векторными моделями не работает, если использовать такие простые подходы как конкатенацию и SVD, так как их качество сильно уступает,

– Итоговое качество на основе векторных моделей может быть улучшено за счет более слабых моделей, обученных на корпусе по информационной безопасности, методами AEME (также и с функцией потерь триплетов), если правильно подобрать способ комбинирования векторных моделей, путем задания весов.

Третья глава посвящена исследованию задачи пополнения графов знаний именованными сущностями.

В **разделе 3.1** описывается основная мотивация задачи. Особенностью графов знаний является то, что они содержат большое количество именованных сущностей, например, WannaCry — это экземпляр для класса компьютерный вирус. Особенностями именованных сущностей является: а) они очень разнообразны, все время появляются новые, что приводит к тому, что в статических векторных моделях могут отсутствовать их векторные представления, и б) в тексте они могут выражаться как отдельными словами, так и многословными выражениями, и даже регистр таких слов имеет значение, из-за чего нормализация слов может привести к существенной потери информации.

По описанным причинам, для того чтобы пополнять графы знаний новыми именованными сущностями, которые связаны отношением экземпляр-класс, необходимо использовать контекстуализированные векторные представления (эмбеддинги) и методы извлечения именованных сущностей в конкретных контекстах

В **разделе 3.2** приводится более формально постановка задачи. Имея:

– Размеченный набор текстовых данных, в котором каждое слово относится к одной из K категорий (или же к пустой категории),
– Не размеченный набор текстовых данных из предметной области, но существенно большего размера,
– Существующую таксономию графа знаний,
– Также могут быть использованы дополнительные размеченные коллекции из общей предметной области.

Требуется реализовать алгоритм A, который будет выделять из текстов из конкретной предметной области сущности описанных категорий, после чего их можно будет добавить в существующий граф знаний.

В **разделе 3.3** описывается реализованный подход к решению поставленной задачи. Задача имеет следующие сложности:

– Размеченных данных по некоторым сущностям может быть недостаточно из-за сложности разметки и особенностей текстов,
– Векторные представления специфичных слов могут быть некорректными из-за отсутствия подобных слов и смыслов в корпусах общей области.

Предлагаются следующие методы для борьбы с описанными сложностями: а) дополнение тренировочных данных данными из общей области, б)

дополнение тренировочных данных за счет большой, неразмеченной коллекции (путем автоматического порождения псевдоразмеченных данных) и в) настройка модели BERT на предметную область.

В **разделе 3.3.1** представлен краткий обзор по методам дополнения обучающих данных (порождение псевдоразметки), а также описан предложенный подход к порождению псевдоразметки для задачи извлечения именованных сущностей в области информационной безопасности. Основная идея метода заключается в следующем: в большинстве контекстов, где упоминается дескриптор объекта, возможны разные варианты упоминания именованной сущности. Дескриптор — это некоторый сигнал/маркер, выраженный в виде слова или многословного выражения. Вариантами упоминания могут быть: 1) дескриптор, за которым следует имя или 2) только имя. Следовательно, можно расширить коллекцию, добавляя имена после дескрипторов или заменяя дескрипторы именами.

Для процедуры порождения псевдоразметки осуществляется отбор предложений с соответствующими дескрипторами в коллекции не аннотированных текстов по информационной безопасности. Сам список дескрипторов формировался вручную с использованием онтологии OENT. Поиск осуществляется с помощью списка дескрипторов и определенных ограничений на предложения. Если предложение соответствует ограничениям, то с равной вероятностью: (1) дескриптор заменяется именем, или (2) имя добавляется после дескриптора, или (3) предложение сохраняется как есть.

Объем данных, полученных таким образом, может быть неограниченного размера. Было исследовано, как объем псевдоразмеченных данных влияет на качество решения целевой задачи. Таблицы 5 и 6 показывают примеры предлагаемой процедуры порождения псевдоразметки. В первой паре предложений дескрипторы были заменены именами; во второй паре предложений имена были вставлены после дескрипторов ”хакер” и ”зловред”.

В **разделе 3.3.2** описывается подход к извлечению именованных сущностей на основе нейросетевой архитектуры BERT[2]. В начале приводится описание самой архитектуры, после чего описываются особенности применения ее к описанной задаче извлечения именованных сущностей в конкретной предметной области.

Из-за того, что предварительное обучение, которое является первым этапом обучения модели BERT, является дорогостоящей процедурой, использование предварительно обученных весов более эффективно. Важными вопросами являются, какие предварительно обученные веса использовать, и на каких данных эти веса были обучены. Первая опубликованная модель многоязычной модели BERT (multilingual-bert-base) была обучена

[2]Devlin J. et al. Bert: Pre-training of deep bidirectional transformers for language understanding //arXiv preprint arXiv:1810.04805. – 2018.

Таблица 5 — Примеры псевдоразметки для HACKER

Исходное	Измененное
Замена	
Отсутствие уязвимостей на сайте и его готовность противостоять атакам **хакеров** - важный вопрос, который часто упорно игнорируется владельцами сайтов.	Отсутствие уязвимостей на сайте и его готовность противостоять атакам **Pwn2Own** - важный вопрос, который часто упорно игнорируется владельцами сайтов.
Вставка	
А количество установленных программных средств защиты от **хакеров** меньше - 71% от тех, кто установил межсетевой экран.	А количество установленных программных средств защиты от **хакеров Sandworm** меньше - 71% от тех, кто установил межсетевой экран.

Таблица 6 — Примеры псевдоразметки для VIRUS

Исходное	Измененное
Замена	
Почти 30% серьезно обеспокоены этой проблемой, еще 25% считают, что опасность **шпионского ПО** преувеличена, а более 15% вообще не считают этот тип угроз проблемой.	Почти 30% серьезно обеспокоены этой проблемой, еще 25% считают, что опасность **Remcos** преувеличена, а более 15% вообще не считают этот тип угроз проблемой.
Insertion	
Описанный выше **вредонос** уникален и может создать большие проблемы как для отдельного человека, так и для всей компании.	Описанный выше **вредонос Locker** уникален и может создать большие проблемы как для отдельного человека, так и для всей компании.

на многоязычных текстовых коллекциях, включая русскоязычные данные. Позже было показано, что модель BERT, специализированная для конкретного языка, работает лучше, чем многоязычная, когда она обучалась на сопоставимом количестве данных. Для русского языка такой моделью выступает RuBERT[3].

В рамках этого исследования было оценено качество моделей BERT в задаче извлечения именованных сущностей в области информационной безопасности со следующими предварительно обученными весами:

– RuBERT, модель, обученная на русскоязычных данных,
– RuCyBERT, модель, полученная путем дообучения RuBERT на текстах по информационной безопасности [4].

RuCyBERT была обучена в рамках работы над диссертацией.

[3]Kuratov Y., Arkhipov M. Adaptation of deep bidirectional multilingual transformers for russian language //arXiv preprint arXiv:1905.07213. – 2019.

В **разделе 3.4** приводится описание набора данных, исследуемой предметной области и доступных ресурсов. В данной работе использовалась обновленная версия набора данных Sec_col в качестве обучающего набора данных для задачи извлечения именованных сущностей. Корпус содержит 861 неструктурированных текстов (более 400 тыс. токенов), которые представляют собой сообщения и комментарии, извлеченные из нескольких источников по информационной безопасности.

Набор сущностей корпуса включает как четыре основных типа: PER (для персон, исключая хакеров), ORG (для организаций, исключая группы хакеров), LOC и EVENT, так и пять типов, зависящих от предметной области, таких как PROGRAM (для компьютерных программ, исключая вредоносное ПО), DEVICE (для различных электронных устройств), TECH (для технологий с собственными именами), VIRUS (вредоносное ПО и уязвимости) и HACKER (для отдельных хакеров и групп хакеров).

Помимо этого, в ряде экспериментов использовался дополнительный корпус из общей предметной области Collection3. Collection3 содержит 1000 новостных текстов, помеченных тремя типами именованных сущностей: людьми (PER), организациями (ORG) и локациями (LOC).

В работе предлагается двухэтапный подход к обучению модели BERT для задачи извлечения именованных сущностей при использовании псевдоразметки:

- На первом этапе модель обучается только на дополнительных данных (на псевдоразметке и на дополнительных размеченных данных из общей предметной области),
- На втором этапе модель дообучается уже на целевых тренировочных данных.

В **разделе 3.5** представлена информация о проведенных экспериментах. В начале описывается методология, технические характеристики и схема проведенных экспериментов. Каждый эксперимент исследовался в двух вариантах: RuBERT и RuCyBERT, для того, чтобы выяснить вклад в итоговое качество как методов порождения псевдоразметки, так и дообучения модели RuBERT на предметную область (RuCyBERT).

Первый эксперимент направлен на то, чтобы выявить, может ли добавление размеченных данных, но из общей области (в которой присутствует только 3 типа именованных сущностей: PER, LOC, ORG), улучшить качество извлечения на целевом наборе данных. Оценивались результаты для двух моделей BERT в двух вариантах: а) базовая постановка обучения модели только на Sec_col и б) с добавлением Collection3 к Sec_col. Результаты оценки показали, что такая процедура не дает значимых улучшений. Следующим экспериментом было выяснить, насколько получившаяся псевдоразметка соответствует эталонной разметке в Sec_col. Для этого модели обучались только на объединении псевдоразметки и Collection3.

Основным экспериментом является оценка подхода на основе двухэтапного обучения с использованием псевдоразметки. В нем сравниваются результаты базовой конфигурации и конфигурации, когда модель обучается в двухэтапном режиме. Результаты представлены в Таблице 7.

Таблица 7 — Результаты эксперимента с двухэтапным обучением модели.

	RuBERT			RuCyBERT		
	база	доп. 400	доп. 1600	база	доп. 400	доп. 1600
DEVICE	42.96	**43.64**	43.28	53.52	**54.16**	52.58
EVENT	66.19	**67.35**	65.69	68.82	**72.34**	70.39
HACKER	58.89	**60.36**	60.29	**68.43**	66.03	66.52
LOC	91.09	**91.37**	91.29	91.10	91.60	**91.70**
ORG	79.27	**79.84**	79.67	80.87	81.07	**81.45**
PER	83.85	**85.59**	85.55	86.38	**88.31**	88.05
PROGRAM	65.45	**66.22**	66.20	68.31	69.10	**69.65**
TECH	67.34	67.20	67.33	71.21	71.24	71.02
VIRUS	45.94	50.61	**51.42**	61.39	61.51	**62.58**
F-micro	71.79	**72.51**	72.48	75.21	75.50	**75.58**
F-macro	66.77	**68.05**	67.86	72.34	**72.82**	72.66
F-macro std	0.68	0.64	**0.54**	0.84	1.02	**0.50**

Из полученных результатов можно сделать вывод, что:

– результаты RuCyBERT всегда значительно лучше, чем результаты RuBERT (на Sec_col),
– Двухэтапный режим обучения с использованием псевдоразметки и Collection3 может улучшить конечные результаты для целевой задачи.

В **разделе 3.5.1** представлены эксперименты по вычислительной производительности обучения модели BERT в зависимости от размера батча (batch size) и использования смешанной точности. Для всех экспериментов в данной главе использовался суперкомпьютер NVIDIA DGX-2 с 16 графическими процессорами NVIDIA Tesla V100. Одной из особенностей этих графических процессоров NVIDIA является более эффективное вычисление операций с плавающей запятой с 16 битами (float16) по сравнению с 32 битами (float32). Следовательно, операции с более низкой точностью должны использоваться на данных графических процессорах NVIDIA, где это возможно без существенной потери качества.

Из полученных результатов следует, что использование смешанной точности ускоряет обучение модели более чем в два раза без потери качества, а также снижает потребление памяти графического процессора. Увеличение размера батча ускоряет обучение, но результирующее качество размером батча 64 ниже, чем для 32 и 16. Вероятно причина в том,

что с подобным размером батча было сделано недостаточно шагов оптимизации нейронной сети.

В **разделе 3.6** приводятся основные выводы по главе, сделанные на основании проведенных исследований.

В **четвертой главе** приведено описание разработанной системы для автоматизированного пополнения графов знаний новыми понятиями. Полученные в Главе 2 результаты не позволяют использовать реализованные методы для пополнения графов знаний в автоматическом режиме, но позволяют использовать их в автоматизированном режиме. Поэтому был реализован программный комплекс, использование которого облегчает задачу пополнения графа знаний для аннотаторов. Схематически программный комплекс представлен на Рис. 3. Построенная система позволяет:

– Получать ранжированный список предсказаний для введенного пользователем слова (сервис предсказаний),
– Обработать набор слов, сформировать для каждого предсказания и представить результат в удобном для аннотирования виде (сервис разметки),
– Обучать новые модели на основании существующих таксономий графов знаний в формате WordNet и векторных моделей слов (модуль обучения).

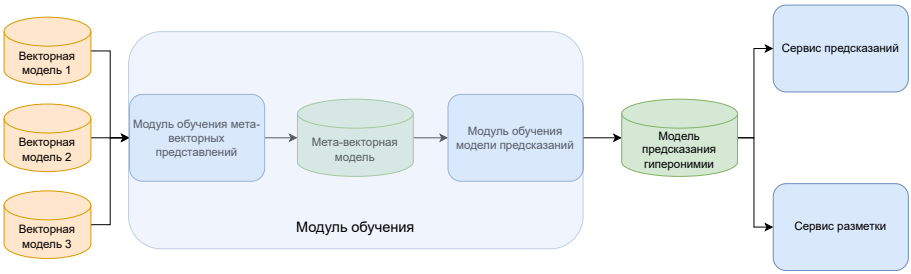

Рис. 3 — Схема реализованной программной системы автоматизированного пополнения таксономии.

В **разделе 4.1** описан сервис предсказаний гиперонимии. Сервис дает возможность пользователю изучить предсказания модели в режиме ”онлайн” для любого слова или многословного выражения, которые присутствуют в словаре векторной модели. В **разделе 4.2** описан сервис разметки. Его цель состоит в том, чтобы упростить работу аннотаторов при решении задачи пополнения таксономии графа знаний новой терминологией. Реализованный сервис позволяет размечать каждое предсказания на один из нескольких классов, навигацию по ближайшим гиперонимам и гипонимам предсказаний. Каждое предсказание, которое отображается

пользователю, также содержит инструменты отображения дополнительной информации о гиперонимах и гипонимах предсказанного концепта. Помимо этого, выводится информация о весе предсказания, и сам список упорядочен в соответствии с этим весом. От аннотаторов требуется: просмотреть список, пополнить его при необходимости близкими концептами и разметить, связан ли каждый концепт с целевым словом некоторым отношением. В **разделе 4.3** представлена информация о реализованном модуле обучения, который позволяет обучать новые предсказательные модели на основе разработанного подхода.

В **заключении** приведены основные результаты работы, которые заключаются в следующем:

В работе исследованы методы пополнения графов знаний новыми понятиями и именованными сущностями.

Разработан и реализован метод пополнения таксономии графа знаний с использованием мета-векторных представлений. Исследована применимость разработанного метода на английском и русском языках, в общей области и конкретной предметной области информационной безопасности.

Разработан новый подход к задаче извлечения именованных сущностей в области информационной безопасности для русского языка с использованием псевдоразметки, двухэтапного обучения и специализированной языковой модели в области компьютерной безопасности RuCyBERT.

Реализована автоматизированная программная система для пополнения таксономии новыми словами.

Публикации автора по теме диссертации в изданиях, индексируемых в базах данных Web of Science, Scopus, RSCI, а также в изданиях, рекомендованных для защиты в диссертационном совете МГУ по специальности

1. Tikhomirov M., Loukachevitch N., Parkhomenko E. Combined approach to hypernym detection for thesaurus enrichment // Computational Linguistics and Intellectual Technologies. — 2020. — С. 736—746. — [Scopus: Impact Factor 0.427].
2. Tikhomirov M., Loukachevitch N. V. Domain-specific Taxonomy Enrichment based on Meta-Embeddings // CEUR Workshop Proceedings. Т. 3036. — 2021. — С. 285—298. — [Scopus: Impact Factor 0.551].
3. Tikhomirov M., Loukachevitch N. Meta-Embeddings in Taxonomy Enrichment Task // Computational Linguistics and Intellectual Technologies: papers from the Annual conference Dialogue. — 2021. — С. 681—691. — [Scopus: Impact Factor 0.427].
4. Tikhomirov M., Loukachevitch N., Dobrov B. Recognizing Named Entities in Specific Domain // Lobachevskii Journal of Mathematics. — 2020. — Т. 41, № 8. — С. 1591—1602. — [Scopus: Impact Factor 0.969].
5. Tikhomirov M. [и др.]. Using bert and augmentation in named entity recognition for cybersecurity domain // International Conference on Applications of Natural Language to Information Systems. — Springer. 2020. — С. 16—24. — [Scopus: Impact Factor 1.363].
6. Tikhomirov M. [и др.]. Pretraining and augmentation in named entity recognition task for cybersecurity domain in Russian // Computational Linguistics and Intellectual Technologies. — 2020. — С. 724—735. — [Scopus: Impact Factor 0.427].
7. Loukachevitch N., Tikhomirov M., Parkhomenko E. Using Embedding-Based Similarities to Improve Lexical Resources // Lobachevskii Journal of Mathematics. — 2021. — Т. 42, № 7. — С. 1532—1546. — [Scopus: Impact Factor 0.969].
8. Nikishina I. [и др.]. Taxonomy Enrichment with Text and Graph Vector Representations // Semantic Web. — 2022. — Т. 13, № 3. — С. 441—475. — [WoS: Impact Factor 2.214].

Иные публикации

9. Тихомиров М. Разработка автоматизированной системы пополнения таксономии на текстах конкретной предметной области // Интеллектуальные системы. Теория и приложения. — 2021. — Т. 25, № 4. — С. 250—254. — [RINC: Impact Factor 0.192].

Тихомиров Михаил Михайлович

Методы автоматизированного пополнения графов знаний на основе векторных представлений

Автореф. дис. на соискание ученой степени канд. физ.-мат. наук

Подписано в печать ___.___._____. Заказ № _____
Формат 60×90/16. Усл. печ. л. 1. Тираж 100 экз.
Типография _____