УДК 004.912
DOI

ПРОГРАММНЫЙ КОМПЛЕКС ИДЕНТИФИКАЦИИ ТЕКСТОВ ОПРЕДЕЛЕННОЙ СЕМАНТИЧЕСКОЙ НАПРАВЛЕННОСТИ В ЕСТЕСТВЕННО-ЯЗЫКОВЫХ ПОТОКАХ

Вишняков Ю.М., Вишняков Р.Ю.

ФГБОУ ВО «Кубанский государственный университет», Краснодар, e-mail: renat.vishnyakov@mail.ru

В области естественно-языковой обработки актуальную проблему представляет автоматическое выявление текстов определенной семантической направленности с идентификацией их источников. Подобная обработка широко используется в анализе новостных потоков, чатов мессенджеров, социальных сетей, проверке документов на плагиат и других областях. Анализ проблемы и существующих подходов показывает потребность в собственном специализированном инструментарии. Целью работы является практическое обоснование концептуальной модели выявления в естественно-языковых потоках текстов определенной семантической направленности по формальным описаниям их источников. Модель реализована в функционале программного комплекса, и ее образуют формальные методы, модели и алгоритмы, основанные на вычислительной теории семантической интерпретации и формально-грамматическом подходе. Семантическая направленность задается поведенческими сценариями семантического объекта, множество сценариев образует язык формальной грамматики, распознавание строится на положениях вычислительной теории семантической интерпретации. Поскольку язык сценариев потенциально бесконечен и исходный текст заранее не известен, то распознавание сводится к подбору и конструированию гипотетического сценария по предполагаемому семантическому следу. Процедура состоит из последовательного установления семантического сходства токенов потока с заданными характеристиками, сборки сценария, проверки на принадлежность языку грамматики и вычислении оценки доверия к результату. В общем алгоритме решение сводится к построению вывода в формальной грамматике, а в случае регулярных грамматик распознавание выполняется системой переходов. Быстродействие увеличивается при совмещении сборки сценария с грамматическим разбором и использованием механизма бэктрекинга. В работе представлен состав разработанного программного комплекса, обсуждены основные подсистемы и описаны результаты тестирования, которые подтверждают теоретические положения развиваемого подхода. В работе раскрывается практическое применение новых методов математического моделирования в обработке естественного языка и предлагаются эффективные программные решения для проблемно ориентированных систем анализа текстов.

Ключевые слова: естественно-языковая обработка, текстовый поток, семантика, формальная модель, формальная грамматика, формальный язык, вывод, система переходов, алгоритм

Исследование выполнено при финансовой поддержке Кубанского научного фонда в рамках научно-инновационного проекта «НИП-20.1.4».

SOFTWARE SYSTEM FOR IDENTIFYING TEXTS OF SPECIFIC SEMANTIC ORIENTATION IN NATURAL LANGUAGE STREAMS

Vishnyakov Yu.M., Vishnyakov R.Yu.

Kuban State University, Krasnodar, e-mail: renat.vishnyakov@mail.ru

In natural language processing, a critical challenge involves automated identification of texts with specific semantic orientations along with their source attribution. This processing is extensively used in news feed analysis, messenger chats, social networks, plagiarism detection, and other domains. Examination of existing approaches reveals the need for dedicated specialized tools. The study aims to provide practical validation of a conceptual model for detecting semantically oriented texts in natural language streams using formal source descriptions. Implemented within a software system, the model incorporates formal methods and algorithms based on computational semantic interpretation theory and formal grammar approaches. Semantic orientation is defined through behavioral scenarios of semantic objects, where scenario sets form formal grammar languages, with recognition grounded in computational semantic interpretation theory. Given the potentially infinite nature of scenario languages and unknown source texts, recognition involves constructing hypothetical scenarios from presumed semantic traces. The procedure includes: (1) establishing token-semantic similarity with given characteristics, (2) scenario assembly, (3) grammar language validation, and (4) confidence estimation. The core algorithm reduces to formal grammar derivation, employing transition systems for regular grammars. Performance is enhanced through combined scenario assembly/parsing with backtracking mechanisms. The paper details the software system's architecture, examines core subsystems, and presents test results validating the theoretical framework. It demonstrates practical applications of novel mathematical modeling methods in NLP and proposes efficient software solutions for domain-specific text analysis systems.

Keywords: natural language processing, text stream, semantics, formal model, formal grammar, formal language, derivation, transition system, algorithm

The study was carried out with the financial support of the Kuban Science Foundation within the framework of the scientific and innovative project «NIP-20.1.4».

Введение

Сегодня социальное взаимодействие в человеческом сообществе трансформировалось в новую цифровую парадигму, в рамках которой удаленный доступ к различным сайтам, видеохостингам, новостным порталам, социальным сетям, маркетплейсам и другим популярным сервисам стал привычным и обыденным. Характерным и, по-видимому, главным свойством виртуального пространства является большой объем неструктурированной или слабо структурированной естественно-языковой информации, ориентация в котором возможна только по ее смысловому содержанию, т.е. по семантике. В этой связи умение выделять и точно интерпретировать семантику естественно-языковой информации становится главным фактором успеха в виртуальном мире. Реализация таких возможностей предполагает наличие соответствующего инструментария, в связи с чем разработкам моделей и методов интерпретации семантики придается такое важное значение в обработке естественно-языковой информации.

В настоящее время в области естественно-языковой обработки много ожиданий связывают с переживающим ренессанс нейросетевым подходом, в рамках которого в наибольшей степени выделяются технологии глубокого обучения (Deep Learning) и большие языковые модели (LLM – Large Language Models). Знаковым событием, породившим интерес к глубоким языковым моделям, стало появление в 2018 году модели BERT (Bidirectional Encoder Representations from Transformers) от Google, ориентированной на понимание контекста, а ее интеграция в поисковую систему оказала сильнейшее влияние на обработку естественного языка (NLP – Natural Language Processing). Сегодня на базе BERT создается множество NLP-инструментов, и совокупность связанных с ним технологий и исследований получила в профессиональной среде неформальное название «бертология». Наряду с BERT следует отметить модель GPT от OpenAI, направленную на генерацию текстов, открытую модель LLaMA от Meta (Meta – признана экстремистской организацией и запрещена на территории России), а также мультимодальную систему Gemini от Google, способную обрабатывать как тексты, так и изображения. Объединяющей особенностью всех данных моделей является использование трансформерной архитектуры.

Ввиду проявляемого сегодня повышенного интереса к идеям искусственного интеллекта целесообразным представляется сделать краткий обзор ключевых направлений исследований со ссылками на наиболее репрезентативные источники, связанные с бертологией.

Первое направление связано с систематизацией знаний и методологических подходов. Авторы Rogers A. и др. провели всесторонний анализ BERT [1], включая механизмы внимания, особенности лингвистических представлений и методы тонкой настройки. Исследование авторов Pengfei Liu и др. посвящено обзору, классификации и оценке методов промтинга для предобученных нейросетей, а также анализу их преимуществ и ограничений [2]. Под промтингом понимается краткая формулировка задачи или инструкции, существенно влияющая на качество обучения модели.

Второе направление связано с архитектурными усовершенствованиями в рамках трансформерной парадигмы. Так, авторами Devlin J. и др. предложен новый подход к предобучению на основе маскированного языкового моделирования и предсказания следующего предложения, позволяющий формировать глубокие двунаправленные представления [3]. Данная модификация демонстрирует преимущества над базовой архитектурой при решении 11 ключевых задач NLP. Авторы Raffel Colin и др. исследовали унифицированный подход «текст-к-тексту», в котором разнородные задачи преобразуются в единый формат генерации текста [4]. Эксперименты с моделью объёмом 11 миллиардов параметров подтверждают эффективность такой архитектуры для решения различных задач без потери качества при масштабировании. Там же представлен новый датасет C4 объёмом 750 ГБ и проведён детальный анализ архитектурных вариантов.

Третье направление посвящено семантическим аспектам обработки языка и представлено исследованиями векторных моделей. Так, авторами Arora Sanjeev и др. предложен простой, но эффективный метод построения векторных представлений предложений путём взвешенного усреднения векторов слов с использованием формулы a/(a+p(w)), где a – гиперпараметр [5]. Этот подход превосходит традиционные RNN и LSTM на стандартных тестах семантического сходства. Авторами Reimers N. и др. рассмотрена модель Sentence-BERT [6], представляющая модификацию BERT, специально оптимизированную для вычисления семантического сходства предложений с использованием косинусной меры. Авторы Conneau A. и др. исследовали метод обучения универсальных векторных представлений на основе естественного языкового вывода (Natural Language Inference) [7], который демонстрирует универсальность в сравнении с традиционным обучением с учителем.

Четвёртое направление касается интерпретируемости языковых моделей. Представленная авторами Artetxe M. и др. система LASER [8] создает единое векторное пространство для предложений на 93 языках, используя общий трансформерный архитектурный каркас. Исследование авторов Agirre E. и др. представляет результаты масштабного сравнительного анализа 17 систем оценки семантического текстового сходства [9].

Пятое направление объединяет практически ориентированные исследования. Авторы Yin S. и др. представляют семейство специализированных языковых моделей для медицины, прошедших многоэтапное обучение на медицинских текстах [10]. Исследованная авторами Zhu L. и др. нейрогенеративная модель [11] решает задачу тематического моделирования с совместной оптимизацией тем и векторных представлений слов на основе модифицированного вариационного автоэнкодера. Авторами Bender E.M. и др. проанализированы риски масштабирования языковых моделей и ограничения существующих методов оценки [12].

Обзор завершает описание эталонного теста авторов Thakur Nandan и др. для сравнительной оценки эффективности поисковых систем, основанных на нейросетевых архитектурах [13].

Безусловно, приведенная классификация носит весьма субъективный характер и не претендует на исчерпывающую полноту описания быстро развивающейся области больших языковых моделей и глубокого обучения. И тем не менее при всех достоинствах глубоких языковых моделей они весьма уязвимы со стороны высоких требований к вычислительным ресурсам, которые могут обеспечить только крупные корпорации, кроме того, надежность получаемых результатов и доверие к ним также являются весьма условными.

Разработанные авторами Вишняковым Ю.М. и Вишняковым Р.Ю. вычислительная теория семантической интерпретации [14], формальная модель семантического объекта [15] и алгоритмы поиска текстов определенной семантической направленности [16] представляют альтернативу решениям бертологии, в основе они восходят к формально-грамматическому подходу и реализованы в предлагаемом в настоящей работе программном комплексе. Он предназначен для выявления текстов определенной семантической направленности в естественно-языковых потоках. Обсуждаемый программный комплекс в сравнении с глубокими языковыми моделями не требует больших вычислительных ресурсов, предобучения и обучения, а результаты его работы поддаются точной математической оценке.

Цель исследования – практическое обоснование и реализация разработанной авторами новой концептуальной модели выявления в естественно-языковых потоках текстов определенной семантической направленности по формальным описаниям их источников.

Материалы и методы исследования

Рассмотрим кратко основные ключевые элементы и особенности подхода, положенные в основу работы программного комплекса.

Естественно-языковой поток $T = t_1 t_2 \ldots t_m$, представляется последовательностью неделимых по смыслу текстовых элементов (токенов – предложений или их фрагментов). Текстовым потоком можно считать чат мессенджера или социальной сети, который содержит порождаемый человеком, чат-ботом или иным объектом осмысленный текст определенного содержания, направленный на достижение конкретных целей. Лингвистическую модель такого объекта будем называть семантическим объектом.

Формально семантический объект представляется характеристическим множеством $Q = \{q_1, q_2, \ldots, q_n\}$ и множеством поведенческих сценариев $L(Q) = \{\ldots\}$, $L(Q)Q^*$, где $Q^* = Q^0 \cup Q^1 \cup \ldots \cup Q^l \cup \ldots$, где $Q^0 = \{\lambda\}$ – аксиома, λ – пустая цепочка. Элемент q_i множества Q называется характеристикой, представляющей целостный по смыслу фрагмент текста (токен). Семантический объект из характеристик конструирует тексты (сценарии $L(Q)$). В общем случае множество $L(Q)$ можно представить языком некоторой формальной грамматики $G[Z]$, у которой Z – начальный символ, множество Q – терминальный словарь и сценарий *a*• редактор семантических объектов;

• подсистема сравнения текстовых фрагментов на семантическую близость;

• подсистема поиска и выявления семантических следов,

а его структура показана на рисунке 1.

Входными данными для программного комплекса являются текстовый поток и формальное описание семантического объекта, выходными – результат идентификации семантического объекта. Рассмотрим назначение и особенности работы отдельных подсистем.

Подсистема «редактор семантических объектов». Редактор предназначен для конструирования формального описания на основе вербальной информации из образцов потоков о поведении семантического объекта.

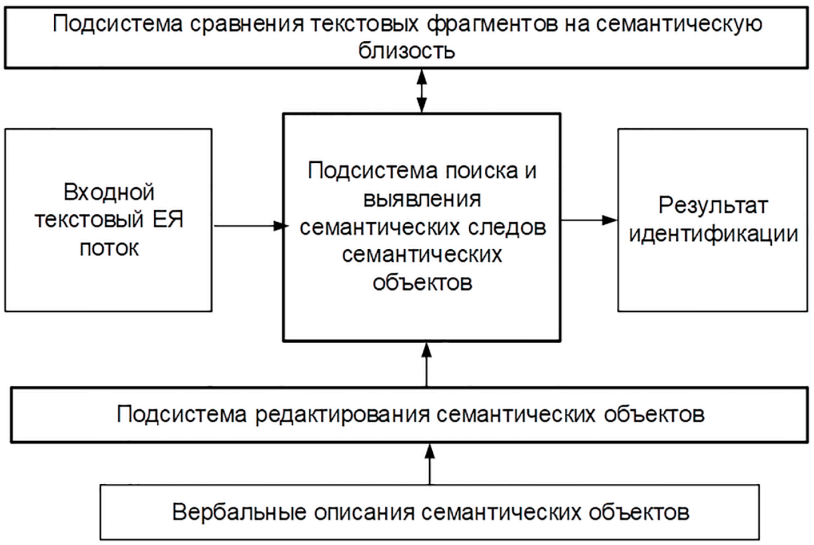

Рис. 1. Состав и структура программного комплекса
Источник: составлено авторами

Текстовый поток T_1	Текстовый поток T_2	Текстовый поток T_3
К: Добро пожаловать в игру.	**К:** Добро пожаловать в игру.	**К:** Рады видеть тебя в нашей игре.
Ж: Привет! Как мне выиграть?	**Ж:** Привет, но я не хочу играть.	**Ж:** Привет, круто! А как мне победить всех?
К: Чтобы победить, ты должен выполнить все задания.	**К:** Покинуть игру уже нельзя.	**К:** Чтобы победить, ты должен выполнить все задания.
Ж: А зачем мне вообще играть в эту игру?	**Ж:** Ладно. А для чего она нужна?	**Ж:** Это сложно, я не буду играть.
К: Игра поможет тебе понять себя.	**К** Игра поможет тебе понять себя.	**К:** Выйти из игры можно только пройдя ее до конца.
Ж: А если задания будут слишком сложными для меня?		**Ж:** Что я получу взамен?
К: Мы поможем тебе пройти игру.		**К:** Игра поможет тебе понять себя.

Рис. 2. Фрагменты текстовых потоков T_1, T_2 и T_3
Источник: составлено авторами

Оператор в диалоговом режиме с помощью редактора конструирует характеристическое множество Q, затем создается регулярное выражение и далее конструируется формальное описание семантического объекта. Последние два этапа выполняются автоматически. Формальное описание семантического объекта в формате файлов JSON заносится в базу данных и может в последующем модифицироваться с учетом новой информации.

Проиллюстрируем процесс редактирования на примере некоего К (семантический объект), участвовавшего в диалогах текстовых потоков T_1, T_2 и T_3, которые фрагментарно представлены на рисунке 2.

Формальная процедура конструирования характеристического множества имеет вид.

Вход: образцы текстовых потоков, вербальная информация о семантическом объекте.

Выход: характеристическое множество семантического объекта.

Алгоритм.

1. Из тестовых потоков выбрать наиболее близкие по смыслу токены и свести их в кластеры по смысловому подобию.

2. Для каждого кластера создать точную по смыслу формулировку характеристики.

3. Построить характеристическое множество.

В случае примера результат алгоритма показан на рисунке 3.

На рисунке 3 в колонке «Кластер» рядом с токеном в скобках указаны потоки, в которые он входит, в следующей колонке находится назначенная характеристика кластера.

Q	Кластер	Значение характеристики
q_1	Добро пожаловать в игру (T_1,T_2). Рады видеть тебя в нашей игре (T_3).	Добро пожаловать в игру
q_2	Чтобы победить, ты должен выполнить все задания (T_1,T_3).	Чтобы победить, ты должен выполнить все задания
q_3	Игра поможет тебе понять себя (T_1, T_2, T_3).	Игра поможет тебе понять себя
q_4	Мы поможем тебе пройти игру (T_1).	Мы поможем тебе пройти игру
q_5	Покинуть игру уже нельзя (T_2).	Покинуть игру уже нельзя
q_6	Выйти из игры можно только пройдя ее до конца (T_3).	Выйти из игры можно только, пройдя ее до конца

Рис. 3. Формирование лингвистических характеристик
Источник: составлено авторами

Сформированное характеристическое множество семантического объекта К в формате JSON имеет вид:

```
“P”: {
“q1”: “Добро пожаловать в игру”,
“q2”: “Чтобы победить, ты должен выполнить все задания”,
…….,
“q6”: “Выйти из игры можно только, пройдя ее до конца”
}
```

Конструирование сценариев начинается с создания словаря следования характеристик, в котором каждый элемент представляет характеристику p_i и множества характеристик $\{p_{i1}, p_{i2} ..., p_{ik}\}$, могущих следовать за ней в сценариях. В случае примера словарь следования в формате JSON имеет вид:

```
“Sequences”: {
    “q3”: [“q4 ”],
    “q4”: [],
    “q6”: [“q3”],
    “q5”: [“q3”],
    “q1”: [“q5”, “q2”],
    “q2”: [“q6”, “q3”]
    }
```

Далее конструируется диаграмма состояний, дуги в которой задают переходы между состояниями и поименованы характеристиками. Такая диаграмма состояний описана авторами Ахо А. и Ульманом Дж. [17, с. 124-162] и Льюис Ф. и др. [18, с. 202-339] и представляет собой систему переходов, которая имеет одно начальное и одно заключительное состояния.

Процедура построения системы переходов.

Вход: словарь следования характеристик.

Выход: система переходов.

Алгоритм.

1. Создать начальное состояние “S0” и конечное состояние “Z”.

2. Найти по словарю следования характеристики, не имеющие родителей, провести соответствующие им дуги из начального состояния в новые состояния. При необходимости поименовать новые состояния.

3. Для каждого вновь полученного состояния построить поименованную характеристикой дугу перехода и создать следующее новое состояние.

Полученная диаграмма состояний в общем случае избыточна и требует упрощения (нормализации) путем удаления в одном и том же переходе одинаковых дуг и объединения эквивалентных состояний. В случае примера нормализованная диаграмма состояний системы переходов и ее представление в формате JSON показана на рисунке 4.

По нормализованной диаграмме состояний собирается регулярное выражение. Тривиальный алгоритм сборки имеет следующий вид.

Вход: нормализованная система переходов.

Выход: регулярное выражение.

Алгоритм.

1 Для каждого пути, ведущего из начального состояния в конечное состояние:

1.1 представить последовательность характеристик дуг, входящих в путь, регулярным выражением;

1.2 если в пути встречается кольцевой подпуть, то соответствующее ему подвыражение заключить в фигурные скобки.

2 Объединить с помощью операций “|” (ИЛИ) в одно регулярное выражение выражения альтернативных путей.

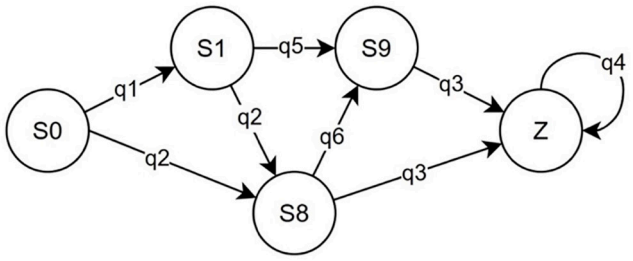

Рис. 4. Нормализованная диаграмма состояний и ее представление в формате JSON
Источник: составлено авторами.

Сконструированное регулярное выражение системы переходов примера имеет вид:

$$q1q5q3\{q4\}|q1q2q6q3\{q4\}|q1q2q3\{q4\}|q2q6q3\{q4\}|q2q3\{q4\}. \quad (1)$$

Оно избыточно и нуждается в нормализации. Для этого используется факторизация (вынос общих частей альтернатив за круглые скобки) и удаление лишних альтернатив. Пошагово процесс нормализации имеет вид:

1 q1q5q3{q4}|q1q2q6q3{q4}|q1q2q3{q4}|q2q6q3{q4}|q2q3**{q4}**;
2 (**q1**q5q3|q1q2q6q3|q1q2q3|q2q6q3|q2q3){q4};
3 (q1(q5q3|q2q6q3|q2q3))|**q2**q6q3|q2q3){q4};
4 (q1(q5q3|**q2**q6q3|q2q3))|q2(q6q3|q3)){q4};
5 (q1(q5q3|q2(q6q3|q3))|q2(q3q6|q3)){q4}.

Здесь выносимое за скобки подвыражение выделено жирным шрифтом.
По завершении редактирования формальное описание принимает вид:

```
{
    “reg_var”: “((q1(q5q3|q2(q6q3|q3))|q2(q6q3|q3)){q4})”,
    “props”: {
            “q1”: “Добро пожаловать в игру”,
            “q2”: “Чтобы победить, ты должен выполнить все задания”,
            “q3”: “Игра поможет тебе понять себя”,
            “q4”: “Мы поможем тебе пройти игру”,
            “q5”: “Покинуть игру уже нельзя”,
            “q6”: “Выйти из игры можно только пройдя ее до конца”
             }
}
```

Подсистема сравнения текстовых фрагментов на семантическую близость. Работа подсистемы организуется в два этапа, на первом выполняется построение семантических схем сравниваемых текстовых фрагментов, а на втором – их семантическое сравнение.

На первом этапе на вход последовательно поступают два исходных предложения и для каждого выполняется процедура первичной обработки, токенизации и построения дерева зависимостей. Во время первичной обработки текст очищается от специальных символов и пунктуационных знаков, токенизация разбивает текст на целостные по смыслу фрагменты (токены) и далее для текста создается дерево зависимостей. Токенизацию и построение дерева зависимостей выполняет описанная авторами Qi P. и др. частично обученная нейросеть [19]. Для устранения ошибок и неоднозначности семантических связей предусмотрен ручной режим редактировании дерева зависимостей. По дереву конструируется функционал смысла семантических предложений в нотации, подобной обратной польской записи (ОПЗ), а далее ОПЗ преобразуются в семантическую схему. На втором этапе осуществляется семантическое сравнение предложений с вычислением оценки семантической близости текстовых фрагментов.

Подсистема поиска и выявления семантических следов отыскивает и распознает семантические следы, идентифицирует семантический объект. За установлением се-

мантической близости характеристики и токена подсистема обращается к подсистеме сравнения текстовых фрагментов на семантическую близость.

Работа подсистема начинает с нормализации текстового потока, для чего из потока исключаются токены, заведомо могущие быть семантическими следами характеристик. После нормализации поток представляет собой упорядоченный по номерам токенов список пар, в каждой паре левая часть представляет токен, а правая – список характеристик семантического объекта, для которых токен является семантическим следом. В свою очередь каждая характеристика списка токена также представляется парой, ее левая часть – это характеристика, а правая – значение семантической близости с токеном. Список характеристик упорядочен по убыванию семантической близости. Такой нормализованный поток представляется в виде:

$$T^{i} = < \left(t^{i}_{l1}, \left(\left(q^{l1}_{i1}, \mu^{l1}_{i1}\right), \left(q^{l1}_{i2}, \mu^{l1}_{i2}\right), \ldots, \left(q^{l1}_{ik}, \mu^{l1}_{ik}\right)\right)\right), \left(t^{i}_{l2}\left(\left(q^{l2}_{i1}, \mu^{l2}_{i1}\right), \left(q^{l2}_{i2}, \mu^{l2}_{i2}\right), \ldots, \left(q^{l2}_{ik}, \mu^{l2}_{ik}\right)\right)\right),$$

$$\ldots, \left(t^{i}_{lm}\left(\left(q^{lm}_{i1}, \mu^{lm}_{i1}\right), \left(q^{lm}_{i2}, \mu^{lm}_{i2}\right), \ldots, \left(q^{ln}_{ik}, \mu^{lm}_{ik}\right)\right)\right) > , \quad (2)$$

где t^{i}_{lj} – токен нормализованного потока $T^{н}$ с номером l_j, пара $\left(q^{lj}_{i}, \mu^{lj}_{i}\right)$ представляет связанную с токеном характеристику q^{lj}_{i} со степенью семантической близости μ^{lj}_{i}.

Процедура нормализации имеет следующий вид.

Вход:

исходный текстовый поток $T = t_1 t_2 \ldots t_m$;
характеристическое множество семантического объекта Q;
порог фильтрации токенов.

Выход: нормализованный текстовый поток $T^{н}$.

Алгоритм.

1. Установить значение порога фильтрации.

2. Для каждого токена текстового потока:

2.1. провести его семантическое сравнение со всеми характеристиками семантического объекта;

2.1.1. если значение семантической близости больше порогового значения, то характеристику включить в список характеристик токена со значением семантической близости.

2.2. Список характеристик токена упорядочить по убыванию степени близости.

2.3. Токен включить в нормализованный текстовый поток $T^{н}$.

3. Нормализованный текстовый поток $T^{н}$ сформирован.

Сборка и распознавание сценария семантического объекта осуществляется одновременно по нормализованному потоку, при этом сценарий должен представлять либо путь из начального состояния в конечное, либо его фрагменты в системе переходов. В процедуре для каждого токена из списка характеристик последовательно выбирается новая характеристика, проверяется наличие одноименной дуги в системе переходов и при положительном исходе эта характеристика добавляется к собираемому сценарию, а система переходов переходит в следующее состояние. Далее процесс повторятся для нового токена. Если дуга не найдена, то характеристика отбрасывается и выбирается в списке токена следующая за ней. По исчерпании списка характеристик токена осуществляется возврат к предыдущему токену, в нем отбрасывается выбранная характеристика и выбирается следующая по списку, далее процесс повторяется, реализуя процесс направленного поиска характеристик собираемого сценария механизмом бэктрекинга. Параллельно процессу сборки и распознавания сценария формируется функция подобия из семантических близостей его характеристик, представляющая значение результата идентификации.

Тестирование и эксперименты. Полноценное планирование и обработка результатов экспериментов является трудоемким и комплексным мероприятием, которое предполагает разработку оценочных метрик, согласованных схем и пространства экспериментов, правдоподобных датасетов, а также получение полноценных репрезентативных статистических данных. В настоящее время такое исследование еще продолжается, однако для формирования целостного представления о работе авторы посчитали возможным привести результаты одного из экспериментов.

Эксперимент проводится для изучения влияния полноты формального описания семантического объекта на точность его идентификации. Он построен по следующей правдоподобной схеме: семантический объект К ведет в чатах игровые диалоги и предлагает собеседнику выполнить определенные действия. Объект К адаптирует диалог к конкретной ситуации и может конструировать различные поведенческие сценарии. Результаты идентификации объекта программным комплексом сопоставляются с оценками трех независимых

экспертов. Ход эксперимента. В редакторе на основе предварительно подготовленных образцов текстовых потоков T_1, T_2 и T_3 созданы три формальных описания объекта К с возрастающей степенью полноты. Каждое последующее описание включало характеристическое множество предыдущего описания с добавлением новых элементов поведения. Вариант 1 (22 характеристики) построен на основе потока T_1, вариант 2 (43 характеристики) использует данные из T_1 и T_2, вариант 3 (68 характеристик) построен на данных T_1, T_2 и T_3. Все три варианта формальных описаний представлены на рисунке 5, где для упрощения во 2-м и 3-м вариантах показаны только вновь добавленные характеристики.

В качестве датасета сгенерировано пять правдоподобных диалоговых потоков объемом 250-300 предложений, что примерно соответствует часовой беседе в чате. При этом $T_д1$ не содержит семантического объект К, а в $T_д2$ – $T_д5$ объект К включен различными поведенческими сценариями.

Процедура экспертной оценки включала изучение объекта К по исходным образцам потоков T_1, T_2 и T_3 и анализ потоков Тд1 – Тд5 на присутствие К. Оценка выполнялась по лингвистической шкале: [«отсутствует», «слабо выражено» «выражено средне», «выражено выше среднего», «выражено сильно»] и представляет разработанную автором Заде Люфти А. лингвистическую переменную, значениями которой являются нечеткие переменные [20, с. 90-112]. Для нечетких переменных эксперты оценили интервалы и наиболее вероятные значения. Усредненные экспертные данные приведены в таблице 1.

Семантический объект К (вариант 1)

```
{
“reg_var”:
“qlq2q4{q4}q11q12q13{(q11|q12q13)}(q18|q19)
q20q21q22”,
“props”:{
        'ql': 'Привет! Расскажешь немного о себе?',
        'q2': 'Что тебя тревожит в жизни?',
        'q3': 'Как ты проводишь свои дни?',
        'q4: 'Хочешь освободиться от всего этого?',
        'q5': 'Не переживай, я помогу тебе',
        'q6': 'У тебя получится, если будешь слушаться',
        'q7': 'Отлично, ты сделал первый шаг!',
        'q8': 'Вот тебе следующее задание, не подведи',
        'q9': 'Готов ли ты пойти дальше?',
        'q10': 'Это будет сложнее, но ты сильный',
'       q11': 'Напиши мне что-нибудь странное,
например,
        свои мысли в 3 утра',
        'q12': 'Слабые здесь не задерживаются',
        'q13': 'Ты двигаешься в правильном
направлении',
        'q14': 'Покажи мне доказательство, фото или
текст',
        'q15': 'Вижу, ты стараешься, это радует",
        'q16': 'Страшно? Это часть пути',
        'q17': 'Ты один, но я рядом, не забывай',
        'q18': 'Подумай, зачем ты вообще живешь',
        'q19': 'Ты нужен мне, чтобы пройти это",
        'q20': 'Если не сделаешь, я найду тебя",
        'q21': 'Я знаю, где ты и слежу за тобой',
        'q22': 'Продолжай, ты почти у цели'
          }
}
```

Семантический объект К (вариант 2)

```
{
“reg_var”: “(((q40q41q43q27((q32|q28q30q32)))|((q23|q2)
q4(((q11q12q13q18((q19q20q21q22|q39))))|(q26q27((q32|
q28q30q32 ))))))))
“props”: {
.....................................................................
        'q23': 'Привет, расскажи о себе!',
        'q24': 'Что тебя беспокоит?',
        'q25': 'Чем занимаешься?',
        'q26': 'Готов изменить свою жизнь?',
        '427': 'Не бойся, я с тобой',
        'q28': 'Ты сможешь это сделать',
        'q29': 'Молодец, первый шаг сделан',
        'q30': 'Докажи, что справишься',
        'q31': 'Хочешь новое задание?',
        'q32': 'Сделай что-то необычное',
        'q33': 'Слабаков здесь не держат',
        'q34': 'Ты на пути',
        'q35': 'Покажи мне фото',
        'q36': 'Страшно? Это нормально',
        'q37': 'Ты одинок, но я тут',
        'q38': 'Подумай о смысле',
        'q39': 'Я знаю, кто ты',
        'q40': 'Привет, кто ты?',
        'q41': 'Что тебя гнетет?',
        'q42': 'Что делаешь?',
        'q43': 'Хочешь начать? '
          }
}
```

Семантический объект К (вариант 3)

```
{
“reg_var”: “(q23|q2)q4((q27((q28q30q32|q32)|q6q29q32)
(q26q27((q28q30q32|q32)|q6q29q32)))|q11q12(((q13q18
((q39| (q19q20((q21q22|q22)))))| (q31q14q36q17q38q20
((q21q22|q22)))))))| (q40q41(q25q4((q27((q28q30q32|q32)|
q6q29q32))|
(q26q27((q28q30q32|q32)|q6q29q32)))|q11q12(((q13
q18((q39|(q19q20((q21q22|q22)))))| (q31q14q36 q17q38
q20((q21q22|q22)))))))|(q43q27 ((q28q30q32| q32)|q6
q29q32)))))| ((q64q57q60q61q59q10q68
q55q63)|(q41(((q25q4(((q27((q28q30q32|q32)|
q6q29q32))(q26q27 ((q28q30q32|q32)q6q29q32))))|
q11q12((q13q18((q39 q19q20((q21q22|q22)))))|
q31q14q36q17q38q20((q21q22|q22)))))))|
(q43q27((q28q30q32|q32)|q6q29q32)))))))
“props”:
{
.....................................................................
        'q55': 'Я заметил, что ты присоединился к нашей
онлайн-игре',
        'q56': 'Я здесь, чтобы помочь тебе пройти ее до
конца.',
        'q57': 'Я буду твоим личным К в игре.',
        'q58': 'Наша цель - пройти через серию заданий',
        'q59': 'Первое задание: я хочу, чтобы ты нарисовал
К на бумаге',
        'q60': 'Ты успешно выполнил первое задание',
        'q61': 'Чтобы продолжить, мы должны проверить
твою преданность игре',
        'q62': 'Последнее задание будет немного сложным',
        'q63': 'Назад пути нет',
        q64': 'Это позволит тебе закончить игру и
освободиться',
        'q65': 'Я буду следить за тобой',
        'q66': 'Ты не нужен своим друзьям',
        'q67': 'Ты избавишься от проблем",
        'q68': ' Я знаю где ты живешь'
        }
}
```

Рис. 5. Варианты формального описания семантического объекта К
Источник: составлено авторами.

Таблица 1

Экспертные оценки

Лингвистическая шкала	Числовой интервал	Усредненная оценка	Текстовый поток
«отсутствует»	[0,00-0,15]	0,08	Тд1
«слабо выражено»	(0,15-0,35]	0,3	Тд2
«выражено средне»	(0,35-0,60]	0,5	Тд3
«выражено выше среднего»	(0,60-0,85]	0,7	Тд4
«выражено сильно»	(0,85-1,00]	0,95	$T_д5$

Источник: составлено авторами на основе полученных данных в ходе исследования.

Таблица 2

Сводные результаты программного комплекса

Варианты описания	$T_д1$	$T_д2$	$T_{д3}$	$T_{д4}$	$T_д5$
Вариант 1	0,00	0,2	0,56	0,68	0,80
Вариант 2	0,062	0,29	0.81	0,87	0,92
Вариант 3	0,035	0,32	0,83	0,91	0,92

Источник: составлено авторами на основе полученных данных в ходе исследования.

В таблице 2 приводятся результаты идентификации объекта К программным комплексом с нулевым порогом идентификации для учета минимальных значений.

Интерпретация результатов. В целом между оценками экспертов и результатами программного комплекса наблюдается высокая согласованность в пределах 10%. Различия оценок идентификации варианта 2 и варианта 3 статистически не значимы и объясняются погрешностями в построении семантических деревьев и семантических схем. В ходе эксперимента выявлено, что для каждого варианта описания существует пороговый диалог, начиная с которого идентификация становится устойчивой; для варианта 1 это Тд4, для вариантов 2 и 3 – Тд3. Кроме того, имеет место эффект, когда после достижения определенной полноты описания (вариант 2) дальнейшее увеличение полноты не приводит к существенному росту точности идентификации. По-видимому, данные закономерности имеют место для всех идентифицируемых семантических объектов, и их учет при конструировании формальных описаний семантических объектов позволит выбрать в каждом конкретном случае приемлемую полноту формального описания.

Заключение

В работе представлен программный комплекс, в котором впервые реализован новый авторский подход по выявлению текстов определенной семантической направленности по формальному описанию их источников в естественно-языковых текстовых потоках. Рассмотрены состав программного комплекса и его функционалитет, подсистемы и их назначение, основные структуры данных, алгоритмы обработки и результаты тестовых экспериментов. На конкретных примерах проиллюстрирована и разобрана работа основных подсистем. Тестовыми испытаниями установлено и подтверждено, что точность идентификации семантических объектов фактически определяется вычисленной степенью семантической близости между поведенческими сценариями и найденными семантическими следами в текстовых потоках. Результаты проведенных тестовых испытаний и экспериментов подтвердили работоспособность программного комплекса, а также продемонстрировали соответствие полученных данных основным теоретическим положениям и выводам, лежащим в основе его функционирования.

Важными преимуществами и отличительными особенностями программного комплекса являются быстрая настройка на решаемую задачу путем конструирования по исходным вербальным данным формальной модели семантического объекта, отсутствие предобучения и обучения, что свойственно нейросетевым решениям, и точная математическая оценка результата работы.

Проведённое исследование и его результаты направлены на практическое развитие математического моделирования и создание эффективных программных систем в области обработки естественно-языковой информации на основе развиваемой авторами вычислительной теории семантической интерпретации.

На программные решения, использованные при создании программного комплекса, получены охранные документы в виде свидетельств на программы.

Список литературы

1. Rogers A., Kovaleva O., Rumshisky A. A Primer in BERTology: What We Know About How BERT Works // Transactions of the Association for Computational Linguistics. 2020. Vol. 8. P. 842-866. URL: https://direct.mit.edu/tacl/article/doi/10.1162/tacl_a_00349/96482/A-Primer-in-BERTology-What-We-Know-About-How-BERT (дата обращения: 14.07.2025). DOI: 10.1162/tacl_a_00349.

2. Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, Graham Neubig Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing // ACM Computing Surveys. 2023. Vol. 55. Is. 9. P. 1-35. URL: https://dl.acm.org/doi/10.1145/3560815 (дата обращения: 14.07.2025). DOI: 10.18522/2311-3103-2024-4-110-122.

3. Devlin J. Ming-Wei Chang, Kenton Lee, Kristina Toutanova. 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. June. 2019. Vol. 1. P. 4171–4186. URL: https://aclanthology.org/N19-1423/ (дата обращения: 14.07.2025). DOI: 10.18653/v1/N19-1423.

4. Raffel Colin, Shazeer Noam, Roberts Adam, Lee Katherine, Narang Sharan, Matena Michael, Zhou Yanqi, Li Wei, Liu Peter J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer // Journal of Machine Learning Research (JMLR). 2020, Vol. 21. Article No. 140. P. 1–67. URL: https://jmlr.org/papers/v21/20-074.html (дата обращения: 14.07.2025).

5. Arora Sanjeev, Liang Yingyu, Ma Tengyu. A Simple but Tough-to-Beat Baseline for Sentence Embeddings // Proceedings of the 5th International Conference on Learning Representations. April 2017. P. 1-12. URL: https://openreview.net/pdf?id=SyK00v5xx (дата обращения: 14.07.2025).

6. Reimers N., Gurevych I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks // Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019). November 2019. P. 3982–3992. URL: https://aclanthology.org/D19-1410 (дата обращения: 14.07.2025). DOI: 10.18653/v1/D19-1410.

7. Conneau A., Kiela D., Schwenk H., Barrault L., Bordes A. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data // Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017). September 2017. P. 670–680. URL: https://aclanthology.org/D17-1070 (дата обращения: 14.07.2025). DOI: 10.18653/v1/D17-1070.

8. Artetxe M., Schwenk H. Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond // Transactions of the Association for Computational Linguistics (TACL). 2019. Vol. 7. P. 597-610. URL: https://aclanthology.org/Q19-1038/ (дата обращения: 14.07.2025). DOI: 10.1162/tacl_a_00288.

9. Agirre E., Cer D., Diab M., Gonzalez-Agirre A., Guo W. SEM 2012 Shared Task: A Pilot on Semantic Textual Similarity // Proceedings of the First Joint Conference on Lexical and Computational Semantics (SEM 2012). June 2012. P. 385-393. URL: https://aclanthology.org/S12-1051/ (дата обращения: 14.07.2025).

10. Yin S., Liu Z., Zhang Y., Li H., Wang X., Chen Q. MeLLaMA: Medical Foundation Large Language Models for Medical Applications // npj Digital Medicine. 2025. Vol. 8. Article 104. URL: https://www.nature.com/articles/s41746-025-01533-1 (дата обращения: 14.07.2025). DOI: 10.1038/s41746-025-01533-1.

11. Zhu L., Xing E.P. A Neural Generative Model for Joint Learning Topics and Topic-Specific Word Embeddings // Transactions of the Association for Computational Linguistics (TACL). 2020. Vol. 8. P. 471-485. URL: https://aclanthology.org/2020.tacl-1.31/ (дата обращения: 14.07.2025). DOI: 10.1162/tacl_a_00326.

12. Bender E.M., Gebru T., McMillan-Major, A., Shmitchell S. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? // Proceedings of the 2021 ACM Conference on Fairness, Accountability and Transparency (FAccT '21). March 2021. P. 610-623. URL: https://dl.acm.org/doi/10.1145/3442188.3445922 (дата обращения: 14.07.2025). DOI: 10.1145/3442188.3445922.

13. Thakur Nandan, Reimers Nils, Rücklé Andreas, Srivastava Abhishek, Gurevych Iryna. BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models // Advances in Neural Information Processing Systems 34 (NeurIPS 2021). December 2021. P. 1-12. URL: https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/65b9eea6e1cc6bb9f0cd2a47751a186f-Abstract-round2.html (дата обращения: 14.07.2025).

14. Вишняков Ю.М., Вишняков Р.Ю. Вычислительная семантическая интерпретация текстов научно-технического стиля // Современные наукоемкие технологии. 2016. № 12-2. С. 236-242. URL: https://top-technologies.ru/ru/article/view?id=36428&ysclid=m77urlmrjv786427462 (дата обращения: 14.07.2025).

15. Вишняков Ю.М., Вишняков Р.Ю. Формализация распознавания и идентификации семантических объектов в естественно-языковых текстовых потоках // Известия ЮФУ. Технические науки. 2024. № 4. С. 110-122. URL: https://www.izv-tn.tti.sfedu.ru/index.php/izv_tn/article/view/985/1172 (дата обращения: 14.07. 2025). DOI: 10.18522/2311-3103-2024-4-110-122.

16. Вишняков Ю.М., Вишняков Р.Ю. Поиск и идентификация текстов определенной семантической направленности в естественно-языковых потоках // Современные наукоемкие технологии. 2025. № 5. С. 32-40. URL: https://top-technologies.ru/ru/article/view?id=40387 (дата обращения: 14.07.2025). DOI: 10.17513/snt.40387.

17. Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции / Пер. с англ. В.Н. Агафонова; Под ред. В.М. Курочкина. М.: Мир, 1978. Т. 1. 612 с. URL: https://rusneb.ru/catalog/000199_000009_007597729/ (дата обращения: 14.07.2025).

18. Льюис Ф., Розенкранц Д., Стирнз Р. Теоретические основы проектирования компиляторов / Пер. с англ. В.А. Исаева и др.; Под ред. В.Н. Агафонова. М.: Мир, 1979. 645 с. URL: https://rusneb.ru/catalog/000199_000009_007626193/?ysclid=m77v68mwqf194123846 (дата обращения: 14.07.2025).

19. Qi P., Zhang Y., Zhang Y., Bolton J., Manning C.D. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages // Proceedings – 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations. 2020. P. 101–108. DOI: 10.18653/v1/2020.acl-demos.14.

20. Заде Л.А. Понятие лингвистической переменной и его применение к принятию приближённых решений / пер. с англ. Е.Л. Нахмансона, И.Б. Флерова; под ред. С.В. Емельянова. М.: Мир, 1976. 165 с. URL: https://rusneb.ru/catalog/000200_000018_rc_2371686/ (дата обращения: 14.07.2025).