# ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

УДК 007:681.512.2

## ИЗВЛЕЧЕНИЕ ФАКТОВ ИЗ ЕСТЕСТВЕННО-ЯЗЫКОВЫХ ТЕКСТОВ МЕТОДОМ УНИФИКАЦИИ СЕМАНТИЧЕСКИХ ПАТТЕРНОВ

**И. Ю. Каширин,** д.т.н., профессор кафедры ВПМ РГРТУ, Рязань, Россия;
orcid.org/0000-0003-1694-7410, e-mail: igor-kashirin@mail.ru

*Рассматривается оригинальная технология проектирования и применения семантических шаблонов для обработки конструкций естественного языка. Конструктивно описывается метод унификации семантических паттернов, названных i-паттернами. Технология использует кортежи слов, образованные из различных отношений базы знаний, и используется для извлечения из сложных предложений средств массовой информации (СМИ) лаконичных фактов. Рассматривается сквозной пример программной реализации в среде Python v.3.10, Anaconda v.2.1.*

*При программной реализации технологии используются внешние библиотеки программ SpaCy,WordNet, RuWordNet, Wiki-ru-wordNet, FrameNet, stanza, Yargy, а также разработанные автором статьи поисковый ретривер, Python i-паттерны с оригинальным алгоритмом унификации. Эффективность представленной технологии подтверждается серией практических экспериментов на примере решения задачи накопления обучающего корпуса для языковых нейросетевых BERT-моделей. Результаты исследования будут полезны в задачах классификации материалов СМИ на достоверные и лживые.*

***Целью работы** как научной статьи является представление специалистам в области искусственного интеллекта нового интеллектуального метода унификации семантических паттернов для извлечения из сложных политических статей лаконичных фактов.*

***Ключевые слова:** Bert-модели, извлечение фактов, семантические паттерны, ретриверы, политические новости, анализ естественного языка, модели глубокого обучения.*

**DOI:** 10.21667/1995-4565-2025-91-36-49

### Введение

Применение средств массовой информации (СМИ) главным образом в качестве эффективного орудия информационных войн стало обыденной реальностью для любого грамотного человека планеты. Современные языковые искусственные нейронные сети с успехом идентифицируют такие классы психологического воздействия публикаций [1, 2], как токсичные материалы, фейк-новости, материалы провокационного действия, тексты с отрицательным или положительным настроением, гневные материалы. В развитых странах запада развитие нейросетевых технологий в этой области финансируется государством. Инициативные отечественные разработки также имеют некоторые достижения в сфере анализа естественно-языковых текстов [3-7].

Основными технологиями, реализующими языковые аналитические ML-модели (machine learning models), на сегодняшний день можно считать LLM [8], BERT[9], GPT[6, 10]. Из наиболее мощных моделей известны также такие, как GPT 4, J-1 (Jurassic-1), BLOOM, T0++, Cohere-LM, T5, OPT, Codex. Обзор этих разработок приведен в [11]. Среди отечественных достижений можно также выделить YaLM от Яндекса и ruGPT3XL от Сбера [10, 11].

Каждая из перечисленных языковых моделей потребовала многомиллионных вложений в высокие технологии, а также использования весьма неординарных вычислительных ресурсов и трудозатрат в десятки тысяч человеко-дней.

В то же время актуальными исследованиями признаны более простые нейросетевые модели, по некоторым параметрам даже превосходящие только что перечисленные. Среди них можно отметить следующие.

Bloom (BigScience)[10]: модель, генерирующая тексты на многих языках. Её качество и способность понимать сложные запросы немногим ниже GPT.

Claude (Anthropic)[12]: модель способна к рассуждениям. В некоторых задачах она может конкурировать с лучшими языковыми моделями.

Falcon (Technology Innovation Institute) [13]: модель, которая в бенчмарках показывает результаты, схожие с самыми большими моделями.

LLaMA 2 (Meta)[8,11]: модель демонстрирует серьезные результаты в генерации текста, переводе и ряде других задач, но может уступать GPT в когерентности ответов на сложные запросы.

Однако наибольшего внимания для разработчиков систем искусственного интеллекта на взгляд автора настоящей статьи заслуживают языковые модели, достигающие потрясающих результатов, но в то же время использующие в десятки и сотни раз меньший объем обучающих корпусов. Развитие таких технологий доступно не только большим, но и весьма малым творческим коллективам. Примером такой языковой модели является RETRO (Retrieval-Enhanced Transformer), созданная на основе модернизации авторегрессивных моделей с улучшенным извлечением токенов [14]. Использованная технология объединяет ретривер Berta, дифференцируемый кодер и механизм фрагментированного перекрестного внимания для прогнозирования токенов, используя в 25 раз меньше параметров, чем GPT-3. В то же время RETRO по своей функциональной мощности вполне сопоставима с GPT-3. Для ускорения и упрощения работы со знаниями эта модель использует базу данных для хранения токенов, соседствующих в текстах, косвенно выделяя таким образом локализованные ситуативные области знаний.

Таким образом, актуальной становится задача создания более простых, но более структурированных языковых моделей, основанных, например, на локализованных онтологиях [3] или других моделях знаний. *Целью таких моделей может служить извлечение из текстов фактографических и других знаний.* Такая постановка задачи позволит формировать логическое и, в частности, причинно-следственное обоснование классификации текстов СМИ, определять их идеологические источники и истинные цели опубликованных материалов, в том числе как средств ведения информационной войны. Построению такой интеллектуальной технологии и соответствующей языковой модели посвящена настоящая статья.

**Инструментарий для анализа естественно-языковых текстов**

Для решения поставленной задачи необходимо рассмотреть лучшие из существующих технологий структуризации текстов на естественном языке, которые можно использовать для извлечения фактов. Наиболее известными подходами к извлечению знаний являются следующие.

*Использование формальных грамматик*. В этом случае любым из известных способов описываются грамматики большого множества предложений естественного языка [15]. Затем с помощью продукционных правил реализуется парсинг предложений с одновременным выделением содержащихся в них информационных фактов.

*Идентификация семантических отношений*. Способ основан на выделении в предложении двух или более понятий (как правило, не более трех). Функционал нейросетевой модели направлен на определение наиболее точного семантического отношения, существующего между ними. Здесь, как правило, используется предварительная разметка предложения [3].

*Использование текстовых шаблонов*. В этом случае выделяются шаблоны для локальных фрагментов текста, иногда это – единственный шаблон, но чаще несколько шаблонов, дающих возможность выделить сложные отношения между понятиями [16].

В предлагаемой здесь технологии будет использоваться третий подход с использованием шаблонов (паттернов), поскольку он дает возможность использовать элементы первых двух подходов. Действительно, формальные грамматики могут задаваться с помощью теоретико-графовых сетей, которые определяют главным образом последовательность слов в предложениях и других словарных конструкциях. Паттерны, представленные множеством возможных последовательностей слов, можно при строгом толковании последовательностей считать одним из способов определения грамматики. Кроме того, в предлагаемой технологии будут кроме прочего инструментария использоваться семантические паттерны (далее обозначаемые как iPatterns), из успешного сопоставления с которыми естественно-языковых конструкций будет следовать наличие семантического отношения. Это фактически реализует функционал второго из перечисленных подходов.

Следует отметить, что современные инструменты для анализа текстов на естественном языке, которые удалось протестировать автору статьи, весьма многочисленны и разнообразны. Для краткости изложения наиболее эффективные из них приведены в таблице 1. Для различных прикладных пакетов (инструментов) в таблице указаны следующие функциональные возможности.

*Токенизация* – выделение из текста предложений и слов с одновременным вычислением их лексических характеристик.

*Лемматизация* – определение нормальных форм слов.

*NER* – распознавание именованных сущностей (Named Entity Recognition).

Примерами именованных сущностей являются:

– топонимы ( географические названия и места, посещаемые людьми);

– имена (ФИО);

– даты событий и времени;

– денежные суммы;

– наименования предприятий;

– другие устойчивые наименования.

*ДСЗ* – возможность построения дерева синтаксических зависимостей в предложении.

*Сходство* – вычисление характеристики семантического сходства слов, словосочетаний и предложений.

*Семантика* – определение семантических отношений для слов и словарных конструкций, например, выделение гипернимов, гипонимов, меронимов, отношений «субъект-объект» и т.п.

В таблице использованы следующие обозначения.

«+» – функционал присутствует.

«-» – функционал отсутствует.

«*» – функционал либо присутствует частично, для некоторых случаев, либо используются внешние инструменты.

В таблице приведены инструментальные средства для английского языка: Spacy, StanfordNER, OpenNLP, NLTK, MITIE, Rosette, TextRazor, Aylien Google Natural Language, API, ParallelDots [11]. Для русского языка можно использовать инструментарий: Deepmipt NER, DaData, Pullenti, Abbyy Infoextractor, Dictum, Eureka, Promt, RCO, AOT, Ahunter [11].

Особый случай может представлять отечественная разработка Томита-парсер компании Яндекс [16]. Этот инструментарий был целенаправленно создан для извлечения структурированных данных из текста на естественном языке. В нем существуют не только функционал паттернов, но и специфические грамматики для глубинного синтаксического анализа. К его недостаткам можно отнести только сложность базовой концепции, не подчиненной унифицированному синтаксису и формализму программных машин [17], а также некоторую ограниченность семантического анализа.

**Таблица 1 – Функциональные возможности существующего инструментария для анализа текстов**

**Table 1 – Functionality of existing text analysis tools**

| Инструмент | Токенизация | Лемматизация | NER | ДСЗ | Сходство | Семантика |
|---|---|---|---|---|---|---|
| Abbyy InfoExtractor | - | - | + | - | - | - |
| Aylien | + | + | + | + | * | - |
| DaData | + | * | + | - | - | - |
| Deepmipt NER | - | - | + | - | - | - |
| Dictum | + | - | + | - | - | - |
| FrameNet | - | - | * | - | * | * |
| MITIE | + | - | + | - | - | - |
| Natasha | + | + | + | + | - | - |
| NLTK | + | + | * | * | * | * |
| OpenNLP | + | + | + | + | * | - |
| ParallelDots | + | + | + | * | + | - |
| Promt | + | + | + | * | * | * |
| Pullenti | + | + | + | * | * | * |
| Rosette | + | + | + | + | + | - |
| RuWordnet | - | - | - | - | - | * |
| spaCy | + | + | + | + | * | * |
| Stanford NER | - | + | - | - | - | - |
| Stanza | + | + | + | + | * | * |
| TextRazor | + | + | + | - | * | - |
| Wiki-ru-wordnet | - | - | - | - | - | * |
| word2vec | - | - | - | - | + | - |
| WordNet | - | - | - | - | * | + |
| Yargy | * | - | * | - | - | - |
| Томита-парсер | + | - | * | - | - | - |

Кроме того, существует весьма мощный инструментарий Open AI API [18], однако он направлен на готовую генерацию текстов, например ответов на вопросы, аннотирование и перевод, а также требует санкционированного получения ключей доступа.

Предлагаемая в настоящей статье технология содержит в своем инструментарии весьма простые но достаточно мощные семантические паттерны и использует адаптированные под эти паттерны решения из внешнего хорошо зарекомендовавшего себя инструментария.

## Архитектура синтеза языковых моделей для извлечения фактов с помощью семантических паттернов

Метод семантических паттернов, подробнее рассматриваемый в следующем разделе, предполагает его использование в контексте архитектуры программного инструментария, приведенного на рисунке 1. Общая суть применения этой архитектуры заключается в создании эффективной нейросетевой модели для извлечения фактов из политических статей в рамках ограниченной предметной области, например «вооруженные конфликты». Точность результирующей модели в основном обусловлена предварительным проектированием онтологической модели знаний о предельно локализованной предметной области вручную инженером по знаниям. Для этого может быть использован любой инструментарий работы с базами знаний, в наших примерах применена система Protégé [19].

Следующим этапом проектирования является разработка паттернов, которые получаются в результате анализа современных политических статей. В этом случае важным фактором является время публикации, поскольку издания с течением времени могут изменять форму и

внутреннюю структуру изложения материала. Например, публикации прежних времен часто содержали изложение фактов от первого лица, т.е. журналиста, тогда как сейчас информационная атака также формируется автором материала, но с изложением информации, высказанной третьими лицами (политиками, другими изданиями). Такие изменения приводят к проблеме «дрейфа данных», требующих использования специальных технологий для коррекции аналитических моделей [4].

Интеллектуальные паттерны являются шаблонами для выделения семантических отношений из весьма сложных неадаптированных текстов, для чего используется специфическая структура таких шаблонов. Формирование паттернов начинается с семантической грамматики, позволяющей выделить основные слова предложения, присутствующие в нем последовательно или вразнобой, но соответствующие целевому n-мерному отношению, где n – количество базовых элементов (концептов) отношения. Затем грамматика переводится в несложный синтаксис конструкций шаблонов iPattern и, в заключительной части, преобразуется в стандартные json-файлы.

Следующий большой этап проектирования заключается в поиске (применении ретривера) и накоплении электронных материалов (Web-скрайпинг) политических статей, предложения которых соответствуют сформированным семантическим паттернам. В текущей версии представленной технологии в качестве информационных источников используются западные ресурсы: 'cnn.com', 'nytimes.com', 'msnbc.com', 'politico.com', 'bloomberg.com', 'aljazeera.com', 'theguardian.com', 'indianexpress.com' и отечественные, но англоязычные СМИ: 'sputnikglobe.com', 'rbth.com', 'RT.com', 'en.kremlin.ru', 'meduza.io/en'. Для реализации этого этапа автором статьи разработан модуль сбора тематических корпусов CorpusMining v.2.1, использующий инструментарий Googlesearch и BeautifulSoup4 в среде Python v.3.10, Anaconda v.2.1 [1].

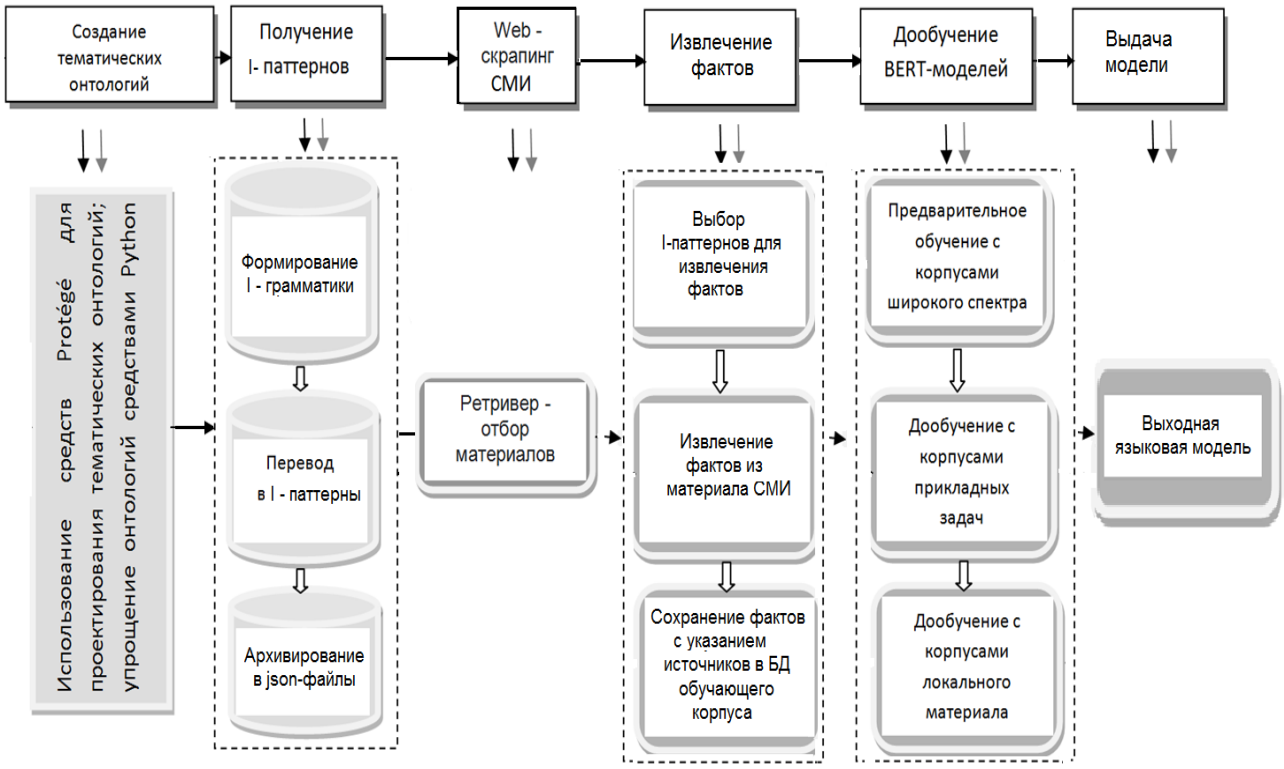


**Рисунок 1 – Архитектура формирования обучающего корпуса и результирующей нейросетевой языковой модели**
**Figure 1 – Architecture of training corpus formation and resulting neural network language model**

Далее следует центральный этап проектирования – извлечение фактов из отобранных на предыдущем этапе материалов. С помощью паттернов iPaterns выделяются концепты, присутствующие в целевом отношении. Каждый концепт (чаще это – именованная сущность) помещается в предварительно отведенное для него поле базы данных, являющейся одним из результатов этого этапа технологии. Другим результатом является полное исходное предложение на естественном языке. Таким образом, собирается полностью достаточный корпус для последующего обучения нейросетевой языковой модели. Пары «предложение → запись базы данных» являются соответственно входными и выходными данными результирующей обученной модели. Сама идея этого этапа не является новой и используется в современных упрощенных языковых моделях. Новизной отличается метод семантической унификации, как сопоставление сложных предложений с семантическими паттернами.

Заключительный этап состоит в самом обучении новой или предварительно обученной BERT-модели [9].

**Описание метода семантической унификации iPatterns**

Для конструктивного описания метода необходимо поставить задачу унификации. Она будет заключатся в том, что для выделения из предложения на естественном языке фактографической информации необходимо унифицировать (сопоставить) исходное предложение с семантическим шаблоном так, чтобы результатом такой унификации стала результирующая конструкция, содержащая искомый факт.

Словарный кортеж-синсет (i-синсет) определим с помощью следующей грамматики:

<терм> ::= <слово> | <терм><разделитель> | <терм> <разделитель> <терм>

<слово> ::= <буква> | <слово> <буква>

<разделитель> ::= пробел | <знак препинания>

<знак препинания> ::= : | , | ! | ? | . | −

<список термов> ::= '<None>' | '<терм>' | '<терм>','<терм>'

<i-синсет> ::= < <список термов> >

По сути дела, под i-синсетом будем понимать кортеж (упорядоченное множество) слов, имеющих отношение к определению одного и того же понятия (концепта).

Примеры i-синсетов:

<'война', 'боевые действия двух стран, направленные друг против друга', 'нападение', 'длительный боевой конфликт', 'war'>,

<'жертва', 'victim', 'погибший', 'временная уступка противнику в шахматах'>.

Представим предложение s в форме цепочки термов:

$$s = w_0 + w_1 + \ldots + w_i + \ldots + w_n ,$$

где терм $w_i$ Є W – слово или словосочетание из предложения s, W – лексикон естественного какого-либо языка, а «+» – операция последовательной композиции слов, словосочетаний и предложений.

Каждый терм обладает множеством свойств, связывающих его с другими термами, например его синонимами, антонимами, гипернимами, гипонимами, меронимами и т.п. Для каждого из этих перечисленных классов термов можно задать соответствие, например для синонимов (соответствие $f_{syn}$):

$$f_{syn} : w_i \rightarrow \{w_{i,0} , w_{i,0} ,\ldots w_{i,j} \ldots , w_{i,k} \},$$

где $w_{i,j}$ – j-й синоним слова $w_i$ .

Теперь можно представить соответствие $f_{syn}$ в форме одноместной алгебраической операции на множестве i-синсетов с добавлением упорядоченности множества с помощью конструкции кортежа:

$$f_{syn} : <w_i> \rightarrow <w_{i,0} , w_{i,0} ,\ldots w_{i,j} \ldots , w_{i,k} >.$$

Для заранее заданного словаря (тезауруса) W определим множество i-синсетов, которые могут быть осмысленно заданы на нем: iW. Такое множество может порождаться системой

операций определенных на нем (сигнатурой), и будет носителем соответствующей алгебраической системы:

$IW = <IW, \Omega = \{+, F_{SYN}, F_{ANT}, F_{GP}, F_{GR}, F_{MR}, F_{NR}\}, R = \{=, \neq, \sim, >, <\}, P = \{P, H, L\}>$ .

Здесь операции $f_i \in \Omega$ означают следующее:

$f_{syn}$ – получение синонимов, $f_{ant}$ – получение антонимов, $f_{gp}$ – получение гипонимов, $f_{gr}$ – получение гипернимов, $f_{mr}$ – получение меронимов, $f_{nr}$ – получение именованных сущностей. Знак «+» является выделенной бинарной операцией конкатенации i-синсетов, например предложение S = 'Израиль атаковал три цели в Беруте.' может быть по разному разбито на синсеты:

<'Израиль'>+<'атаковал'>+<'три цели'>+<'в Беруте'>+<'.'> =

<'Израиль атаковал'>+<'три цели'>+<'в Беруте'>+<'.'> =

<'Израиль атаковал три цели в Беруте.'>

Множество отношений $R = \{=, \neq, \sim, >, <\}$ содержит соответственно бинарные отношения эквивалентности, неэквивалентности, смыслового подобия, следование слов (i-синсетов) друг за другом в словарной конструкции ($w_i > w_j$) – $w_i$ левее $w_j$, ($w_i < w_j$) – $w_i$ правее $w_j$.

Дополнительные отношения P состоят из трех подклассов: p (Pos, часть речи), h (head, главенство в словосочетании), l (lemma, нормальная форма слова). Эти отношения одноместные, и каждый из подклассов имеет частные случаи-представители, например $p_V(w_i)$ – слово принадлежит множеству глаголов, $l(w_i)$ – слово принадлежит множеству лемм, $h_{obj}(w_i)$ – слово является объектом действия в предложении.

Количество и семантика операций и отношений алгебраической системы iW зависит от конкретной программной реализации тезауруса и может быть как сокращено, так и дополнено другими элементами.

I-синсеты представляют собой различные словарные конструкции, такие как слова, словосочетания, предложения. Множество дополнительных отношений P содержит в качестве элементов частные отношения для элементов словарных конструкций. Эти отношения при интерпретации алгебраической системы iW применительно к конкретному предложению могут рассматриваться как предикаты, вычисляющие принадлежность слова в словарной конструкции к словам различных категорий (частей речи, субъектов/объектов, знаков пунктуации и т.п.).

Кроме того, следует пояснить, что i-синтсеты могут быть кортежами произвольной, но всегда конечной местности. Упорядоченность элементов кортежа позволяет задать порядок слов в предложении или статистическую частоту встречаемости синонимов, гипонимов и т.п., если кортеж – это множество синонимов/гипонимов слова.

В сделанных определениях можно адекватно описывать любые предложения и словарные конструкции, оперировать характеристиками слов и словосочетаний, рассматривать их взаимное расположение в предложениях и текстах на естественном языке.

В настоящей статье нас будет интересовать только построение шаблонов (паттернов) словарных конструкций, которые могут быть унифицированы с естественно-языковыми предложениями.

Пусть, например, необходимо для предложения: «25 декабря 2024 года Израиль атаковал три цели в Беруте, сообщило издание WSJ.» составить интеллектуальный паттерн, выделяющий семантические концепты по примерной схеме «дата, субъект, атака, объект». Здесь содержится ожидание в предложении NER-конструкции «дата», двух конструкций, связанных с названиями государств, городов или другой местности, одна из которых субъект атаки, другая – объект атаки. Кроме того, в предложении должен присутствовать глагол, семантически означающий военное агрессивное действие, близким по смыслу к концепту «атака».

Выражение-шаблон t записывается с помощью переменных, означающих последовательные i-синсеты, унифицируемые со словами входного предложения:

$$t = x_1 + x_2 + x_3 + x_4,$$

где переменная $x_1$ должна принимать значение из i-синсета, унифицированного с какой-либо датой, переменная $x_2$ должна получить значение именованной сущности географического названия с синтаксическим классом «субъект», переменная $x_3$ должна ассоциироваться с глаголом, лемма которого входит в синонимы концепта «атака», а переменная $x_4$ будет получать значение из i-синсета географического наименования с синтаксическим классом «субъект».

Поскольку каждая из переменных шаблона имеет областью определения некоторый i-синсет, унифицируемое предложение должно быть представлено как последовательность i-синсетов:

$$S = s_1 + s_{1x} + s_2 + s_{2x} + s_3 + s_{3x} + s_4 + s_{4x}.$$

Синсеты $s_{1x}$, $s_{2x}$ , $s_{3x}$, $s_{4x}$ могут соответствовать любым i-синсетам слов, чье значение «не интересует» шаблон t, и он воспринимает эти слова как неинформативные.

Терм t тоже можно изменить: $T = x_1 + x_{1x} + x_2 + x_{2x} + x_3 + x_{3x} + x_4 + x_{4x}$.

Теперь, чтобы конкретизировать шаблон t, его переменные нужно ограничить следующими соответствующими областями определения.

1. $f_{nr,1}(x_1) = f_{date}(x_1)$, т.е. $x_1$ должен соответствовать дате.
2. $h_{subj}(x_2)$, $f_{nr,2}(x_2) = f_g(x_2)$, $x_2$ − географическое имя, субъект действия.
3. $p_v(x_3)$, $f_{syn}(x_3)$ ~ <'атака'>, $x_3$ − глагол, синонимичный глаголу 'атаковать'.
4. $f_{nr,4}(x_4) = f_g(x_4)$, $h_{obj}(x_4)$ − географическое имя, объект действия.

Теперь задача алгоритма унификации найти такие подстановки σ:

$\sigma = (x_1 \rightarrow s_1, x_{1x} \rightarrow s_{1x}, x_2 \rightarrow s_2, x_3 \rightarrow s_3, x_{3x} \rightarrow s_{3x}, x_4 \rightarrow s_4, x_{4x} \rightarrow s_{4x})$,

чтобы она удовлетворяла условиям (1) – (4).

Для входного предложения «25 декабря 2024 года Израиль атаковал три цели в Беруте, сообщило издание WSJ.» алгоритмом унификации будет выполнен перебор возможных подстановок последовательно идущих слов, в результате чего будет сформирован результат:

$\sigma$ = ( $x_1 \rightarrow$ <'25.12.2024'>, $x_{1x} \rightarrow$ <'None'>, $x_2 \rightarrow$ <'Израиль'>,
$x_3 \rightarrow$ <'атака'>, $x_{3x} \rightarrow$ <'три цели в'>, $x_4 \rightarrow$ <'Бейрут'>,
$x_{4x} \rightarrow$ <', сообщило издание WSJ .'> ).

Если поставлена задача выделить лишь сам факт атаки, можно рассмотреть итоговую трансформацию унификатора σ в итоговый унификатор γ:

$\sigma(x_1, x_{1x}, x_2, x_3, x_{3x}, x_4, x_{4x}) \Rightarrow \gamma(x_1, x_2, x_3, x_4)$,

$\gamma(x_1, x_2, x_3, x_4)$ = <'25.12.2024'>,<'Израиль'>,<'атака'>, <'Бейрут'>.

Такой результат может быть помещен в базу фактов с соответствующими информационными полями «Date», «Subject», «Action», «Object».

Аналогичные методы унификации на программном уровне уже существуют в инструментарии для обработки естественного языка [20]. Отличие рассмотренного в этом разделе метода iPatterns от предыдущих методов в наличии широкого спектра *семантических отношений*, которые могут быть использованы для ограничения областей определения лингвистических переменных, интерпретируемых в i-синсеты.

## Программная реализация метода семантической унификации iPatterns

Для реализации метода нужно использовать:

– программные пакеты, позволяющие анализировать предложения на естественном языке с приписыванием им характеристик для частей речи и синтаксической ролью слова в предложении;

– программы, отыскивающие семантические токены для слов (синонимы, гипернимы, гипонимы и т.п.);

– инструментарий, выделяющий в предложении именованные сущности (даты, имена, географические названия).

Программная реализация метода семантической унификации заключается в комплексировании наиболее эффективных элементов следующего внешнего инструментария: WordNet, RuWordNet, Wiki-ru-wordNet, FrameNet, stanza, SpaCy, Yargy [11]. При этом выигрыш состо-

ит в том, что концепты и отношения, не найденные в одном инструментарии, могут быть хорошо разработаны в другом.

Приведем пример синтаксического анализа инструментарием SpaCy англоязычного предложения «Israel strikes target three locations in Beirut, said Joe Biden», который изображается в форме дерева зависимостей (рисунок 2).

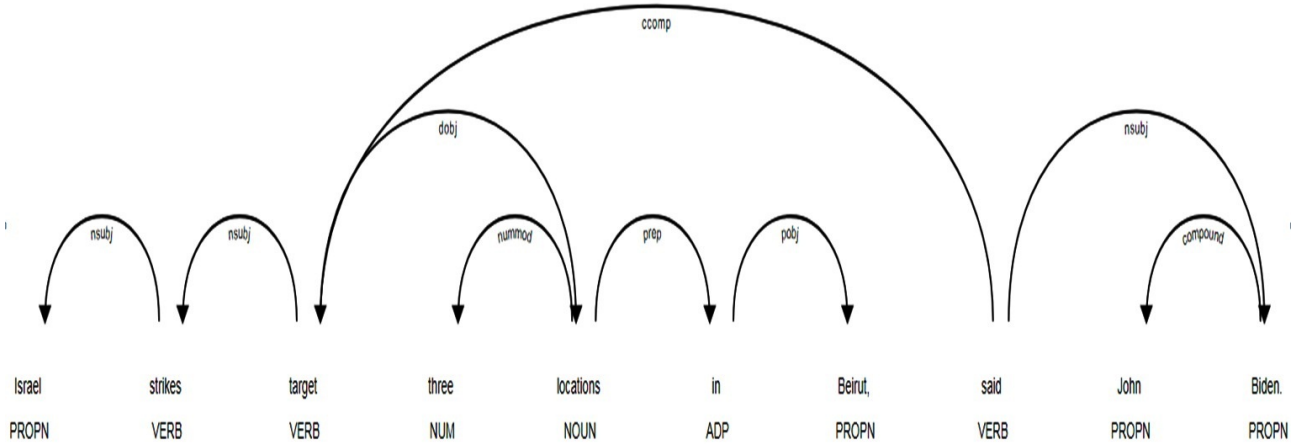


**Рисунок 2 – Пример дерева зависимостей, выданного в SpaCy**
**Figure 2 – Example of dependency tree provided in SpaCy package**

Все характеристики словарных конструкций в этом случае доступны в соответствующих информационных полях (рисунок 3).

| Word | Pos | Head | Lemma | Pos of Head |
|---|---|---|---|---|
| Israel | PROPN | nsubj | strikes | VERB |
| strikes | VERB | nsubj | target | VERB |
| target | VERB | ccomp | said | VERB |
| three | NUM | nummod | locations | NOUN |
| locations | NOUN | dobj | target | VERB |
| in | ADP | prep | locations | NOUN |
| Beirut | PROPN | pobj | in | ADP |
| , | PUNCT | punct | said | VERB |
| said | VERB | ROOT | said | VERB |
| Joe | PROPN | compound | Biden | PROPN |
| Biden | PROPN | nsubj | said | VERB |
| . | PUNCT | punct | said | VERB |

**Рисунок 3 – Информационные поля характеристик слов**
**Figure 3 – Information fields of word characteristics**

Применение i-паттернов, как следует из архитектуры, представленной на рисунке 1, используется для автоматического формирования обучающих корпусов для нейросетевых языковых моделей. Исходные англоязычные материалы СМИ предварительно собираются модулем сбора тематических корпусов CorpusMining v.2.0, разработанным автором статьи и использующим инструментарий Googlesearch и BeautifulSoup4 в среде Python v.3.10, Anaconda v.2.1. Для формирования примеров для результатов, используемых при обучении, применяется модуль IntellectualParsing v.1.0, также программно реализованный автором. Модуль имеет набор приведенных далее Python-функций, состав которых стандартизован с множеством существующих NLP-программ.

```
#       Функции, созданные на основе средств W O R D N E T
# Получение синонимов (с возможностью русского перевода)
# get_synonyms_wordnet(word, translate = 0): # return (synonyms, result_rus)
# Получение гипонимов
# get_hyponyms_wordnet(word, translate = 0):
# Получение схожих слов разных частей речи
```

```
# get_potential_derivationally_related(word, translate = 0)
# Получение связанных по смыслу глаголов
# get_related_verbs(verb, translate = 0)
# Преобразование глагола в однокоренное существительное
# verb_to_nouns(verb_word, translate = 0):
#
#    Функции, созданные на основе средств W I K I R U W O R N E T (аналогично Wordnet)
# getRWSynonyms(word):      return (ResultList, ruList)
# getRWhyponyms(word):      return (ResultList, ruList)
#
#       Функции, созданные на основе средств S T A N Z A
# Поучение леммы
# getLemmaStanza(text)        # выход: return(res_lemma, res_upos)
# Получение всех характеристик слова
# getAllCharactiristicStsnza(Sentence, Word)
# выход:  [ Слово,    Лемма,  Часть речи,
# Его хозяин в отношении синтаксического подчинения,  Роль в предложении]
# Получение собственных имен
# getNERlist(sentence, language='en')
# выход: NER_names, GPE, LOC, ORG, PERSON, DATE, OTHER
#
#       Функции, созданные на основе средств N A T A S H A
# Получение всех характеритсик
# getAllCharactiristicNatasha(Sentence, Word)
# выход: [ Слово,    Лемма,  Часть речи,   Его хозяин в отношении
#  синтаксического подчинения,  Роль в предложении  ]
#
#       SPACY
# Получение списка имен собственных для входного предложения
# getNERlistSpacy(sentence, language='en_core_web_sm')
# выход: NER_names, GPE, LOC, ORG, PERSON, DATE, FAC, OTHER
# Паттерны в IntellectualParsing.ipynb
# Унификация предложения со списком паттернов,
# возвращается первый удачный унификатор
UnifyPatternList(sentence, pattern_list)
#Преобразование списка паттернов в json формат
patternList_to_json(pattern_list)
#Преобразование I-паттерна одного шаблона в json формат
def pattern_to_json(pattern_text)
# Преобразование json текста для списка паттернов в список I-паттернов
jsonList_to_PatternList(jsonListText)
# Обратное преобразование текста из json файла в I-паттерн
json_to_pattern_to(json_text)
# Запись отдельного паттерна в файл
json_to_file(file_name, pattern)
# Чтение отдельного паттерна из файла
file_to_json(file_name)
# Функция getNERlistSpacyN  идентификации географических названий,
# мест локации, организаций, имен личностей, дат, денежных сумм
# SpaCy использует стандартные типы NER-сущностей, такие как
# `PERSON`, `ORG`, `GPE`, `LOC`, `DATE`, `TIME`, `MONEY`,
```

```
# `PERCENT`, `ORDINAL`, `CARDINAL`
doc = nlp(Sentence)
print(getNERlistSpacyN(doc, Sentence))
```

I-паттерны в приведенном инструментарии задаются в упрощенной форме в виде Python-списков, как в следующем примере:

['<*> <T=GPE> <*> <F=strike> <*> <T=GPE> <*><F=say><*><T=PERSON>'].

Здесь описана последовательность i-синсетов, где <*> соответствует любому количеству слов в предложении, в том числе пустому, что заставляет алгоритм унификации искать с начала предложения терм для паттерна <T=GPE>, пропуская слова, не относящиеся к именованным географическим названиям (GPE). Далее в предложении отыскивается слово − синоним «strike», спустя несколько слов вновь унифицируется географическое название, затем синоним или лемма слова «say». В конце предложения должно стоять какое-либо собственное имя человека, на что указывает элемент <T=PERSON> первого i-паттерна из списка.

Поскольку анализируемые предложения из публикаций СМИ могут иметь и другие последовательности языковых конструкций, список i-паттернов может быть большим. Вот как выглядит примерная сборка i-паттерна в этом программном инструментарии на языке Pythin v.3.0:

```
pattern_list = list()
pattern_list = pattern_list + ['<*> <T=GPE> <*> <F=strike> <*> <T=GPE>
<*><F=say><*><T=PERSON>']
pattern_list = pattern_list + ['<*> <T=GPE> <*> <F=strike> <*> <T=GPE> <*>']
pattern_list = pattern_list + ['<*> <T=GPE> <*> <F=strike> <*> <T=GPE>']
pattern_list = pattern_list + ['<*> <T=GPE> <*><L=VERB> <*><T=GPE><*><F=grab><*>']
```

Формируемый список i-паттернов будет выглядеть так:

```
['<*> <T=GPE> <*> <F=strike> <*> <T=GPE> <*><F=say><*><T=PERSON>',
 '<*> <T=GPE> <*> <F=strike> <*> <T=GPE> <*>', '<*> <T=GPE> <*> <F=strike> <*>
<T=GPE>',
 '<*> <T=GPE> <*><L=VERB> <*><T=GPE><*><F=grab><*>']
```

После преобразования в формат json-файлов будут сформированы следующие последовательности:

*** На входе: <*> <T=GPE> <*> <F=will> <*> <V=to> <F=attack> <*> <T=GPE> <*>
[{"Z":"*"} {"T":"GPE"} {"Z":"*"} {"F":"will"} {"Z":"*"} {"V":"to"} {"F":"attack"} {"Z":"*"} {"T":"GPE"} {"Z":"*"}]
Из файла считано: [{"Z":"*"} {"T":"GPE"} {"Z":"*"} {"F":"will"} {"Z":"*"} {"V":"to"} {"F":"attack"} {"Z":"*"} {"T":"GPE"} {"Z":"*"}]
*** На входе: [{"Z":"*"} {"T":"GPE"} {"Z":"*"} {"F":"will"} {"Z":"*"} {"V":"to"} {"F":"attack"} {"Z":"*"} {"T":"GPE"} {"Z":"*"}]

Обратно преобразован: <*> <T=GPE> <*> <F=will> <*> <V=to> <F=attack> <*> <T=GPE> <*>

Унификация реализуется следующим образом:

```
Sentence    = "Israel strikes target three locations in Beirut, said Joe Biden."
Unify, UnificationResult = UnifyPatternList(Sentence, pattern_list).
```

Ее результатом станет список списков на языке Python:

[[Israel, 'T=GPE', 'nsubj'], [strikes, 'F=strike', 'nsubj'], [Beirut, 'T=GPE', 'pobj'], [Joe Biden, 'T=PERSON', 'nsubj']]

Этот результирующий список будет внесен в базу фактов как запись, представленная таблицей 2. Название СМИ (Media) в эту таблицу заносится из внешнего указания источника материала.

**Таблица 2 – Результирующая запись в базе фактов**

**Table 2 – Resulting record in fact database**

| Subject | Action | Object | Victim | Source | Media |
|---|---|---|---|---|---|
| Israel | Attack | Beirut | None | Joe Biden | WSJ |

## Заключение

1. Результаты проведенного исследования показали эффективность применения метода семантической унификации iPatterns, который позволил улучшить характеристику релевантности результатов использования существующего модуля сбора тематических корпусов CorpusMining v.2.1 12 %.

2. Использование программных средств IntellectualParsing.ipynb v.1.0 дало возможность сократить трудоемкость получения обучающей выборки для языковой Bert-модели с 130 записей/час до 210 записей/час, то есть на 62 %.

3. Получение более точной обучающей выборки позволит повысить характеристики точности языковых моделей, определяющих токсичность и достоверность текстов западных СМИ.

## Библиографический список

1. **Каширин И.Ю.** Нейросети нового многополярного мира: классификация электронных новостей // Вестник Рязанского государственного радиотехнического университета. 2024. № 87. С. 29-40. DOI: 10.21667/1995-4565-2024-87-29-40.

2. The platform where the machine learning community collaborates on models, datasets, and applications. [Электронный ресурс]. Дата обновления: 10.01.2025. URL: https://huggingface.co (дата обращения: 14.01.2025).

3. **Каширин И.Ю.** Векторизация текста на основе ICF+ онтологии в ансамблях моделей машинного обучения для классификации электронных ресурсов // Вестник Рязанского государственного радиотехнического университета. 2024. № 90. С. 41-53. DOI: 10.21667/1995-4565-2024-90-41-53.

4. **Kashirin I.Yu.** Semantic Data Fragmentation for Identification of Covariant Conceptual Drift in Machine Learning Models // International Journal of Open Information Technologies. 2024. Vol. 12. No. 7. Pp.10-15.

5. Natasha solves basic NLP tasks for Russian language: tokenization, sentence segmentation, word embedding, morphology tagging, lemmatization, phrase normalization, syntax parsing, NER tagging, fact extraction. [Электронный ресурс]. Дата обновления: 11.01.2025. URL: https://github.com/natasha/natasha (дата обращения: 14.01.2025).

6. **Emelyanov A., Shliazhko O., Katricheva N., Shavrina T.** Using RuGPT3-XL Model for RuNormAS competition. Computational Linguistics and Intellectual Technologies: Papers from the Annual International Conference «Dialogue». 2021. Issue 20. Ch. 18. Pp. 204-212.

7. **Bochkarev V., Solovyev V.** Properties of the network of semantic relations in the Russian language based on the RuWordNet data. Journal of Physics: Conference Series 1391. 2019. Pp.1-5. DOI: 10.1088/1742-6596/1391/1/012052.

8. **Hugo Touvron, Thibaut Lavril, Gautier Izacard at al.** LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971.v1.27 Feb. 2023. Pp.1-27.

9. **Каширин И.Ю.** Токенизация политических текстов в BERT-моделях с использованием ICF+-онтологий // Информационные технологии. 2024. Т. 30. № 12. С.622-632. DOI: 10.17587/it.30.622-632.

10. **Percy Liang, Rishi Bommasani, Tony Lee at al.** Holistic Evaluation of Language Models, Transactions on Machine Learning Research. 2023. Issue 08. Pp. 1-162.

11. Естественно-языковые модели. Материалы. [Электронный ресурс]. Дата обновления: 11.01.2025. URL: https://i.kashirin.net/forstudents/modelsfornlp. (дата обращения: 14.01.2025**).**

12. **Ying Li , Zichen Song, Weijia Li.** Benchmarking Large Language Models in Adolescent Growth and Development: A Comparative Analysis of Claude2, ChatGPT-3.5, and Google Bard. Research Gate. https://www.researchsquare. com/article/rs-3858549/v1.

13. Falcon. [Электронный ресурс]. Дата обновления: 21.10.2024. URL: https://the-decoder.com/falcon-180b-open-source-language-model-outperforms-gpt-3-5-and-llama-2/. (дата обращения: 24.10.2024).

14. **Borgeaud S., Mensch A., Hoffmann J. at al.** Improving language models by retrieving from trillions of tokens // Computer science. Computation and language. DOI: https://doi.org/10.48550/arxiv.2112.04426.15. **Kashirin I.Yu., Khoroshevsky V.F.** Development of an ATN-oriented linguistic processor (algebraic approach) // International Conf. «Artificial Intelligence». Suwalki. Poland. 15-20 June1987.

16. **Рубайло А.В., Косенко М.Ю.** Программные средства извлечения информации из текстов на естественном языке // Альманах современной науки и образования. 2016. № 12 (114). С. 87-91.

17. **Каширин И.Ю.** Формальные программные машины для объектно-ориентированных языков: C++ // Вестник Рязанского государственного радиотехнического университета. 2024. № 89. С. 56-64. DOI: 10.21667/1995-4565-2024-89-56-64.

18. An Introductory Guide to OpenAI's API. [Электронный ресурс]. Дата обновления: 10.02.2024. URL: https://webreference.com/ai/api/. (дата обращения: 10.01.2025).

19. **Муромцев Д.И.** Онтологический инжиниринг знаний в системе Protégé. СПб: ГУ ИТМО. 2007. 62 с.

20. **Grenader U., Miller M.** Pattern Theory: From Representation to Inference, Oxford University Press, Inc.198 Madison Ave. New York, NY United States. 2007. 596 p.

21. **Kafe E.** Persistent semantic identity in WordNet. Cognitive Studies Études cognitives. 2018 (18). https://doi.org/10.11649/cs.1717.

UDC 007:681.512.2

# EXTRACTING FACTS FROM NATURAL LANGUAGE TEXTS BY METHOD OF UNIFICATION OF SEMANTIC PATTERN

**I. Yu. Kashirin,** Dr. Sc. (Tech), Professor, Department of Computational and Applied Mathematics, RSREU, Ryazan, Russia;
orcid.org/0000-0003-1694-7410, e-mail: igor-kashirin@mail.ru

*The original technology of designing and applying semantic patterns for processing natural language constructions is considered. The method of semantic patterns unification, called i-patterns, is described constructively. The technology uses tuples of words formed from various knowledge base relationships and is used to extract concise facts from complex sentences of mass media. An end-to-end example of software implementation in Python v.3.10 and Anaconda v.2.1 environments is considered.*

*Software implementation of the technology uses external software libraries SpaCy, WordNet, RuWordNet, Wiki-ru-WordNet, FrameNet, stanza, Yargy, as well as search retriever, Python i-patterns with an original unification algorithm developed by the author of the article. The effectiveness of the technology presented is confirmed by a series of practical experiments using the example of solving the problem of accumulating a training corpus for language neural network BERT models. The results of the study will be useful in classifying media materials into reliable and false ones.*

***The aim of the work** as a scientific article is to present a new intelligent method of unifying semantic patterns to extract concise facts from complex political articles to the experts in AI field.*

***Keywords:** Bert models, fact extraction, semantic patterns, retrievers, political news, natural language analysis, deep learning models.*

**DOI:** 10.21667/1995-4565-2025-91-36-49

**References**

1. **Kashirin I.Yu.** Nejroseti novogo mnogopolyarnogo mira: klassifikaciya elektronnyh novostej. *Vestnik Ryazanskogo gosudarstvennogo radiotekhnicheskogo universiteta*. 2024, no. 87, pp. 29-40. DOI: 10.21667/1995-4565-2024-87-29-40. (in Russian).

2. T*he platform where the machine learning community collaborates on models, datasets, and applications*. [Electronic resource]. Update date: 10.01.2025. URL: https://huggingface.co (Date of request: 14.01.2025).

3. **Kashirin I.Yu.** Vektorizaciya teksta na osnove ICF+ ontologii v ansamblyah modelej mashinnogo obucheniya dlya klassifikacii elektronnyh resursov. *Vestnik Ryazanskogo gosudarstvennogo radiotekhnicheskogo universiteta*. 2024, no. 90, pp. 41-53. DOI: 10.21667/1995-4565-2024-90-41-53. (in Russian).

4. **Kashirin I.Yu.** Semantic Data Fragmentation for Identification of Covariant Conceptual Drift in Machine Learning Models. *International Journal of Open Information Technologies*. 2024, vol. 12, no. 7, pp.10-15.

5. *Natasha solves basic NLP tasks for Russian language: tokenization, sentence segmentation, word embedding, morphology tagging, lemmatization, phrase normalization, syntax parsing, NER tagging, fact ex-*

*traction*. [Electronic resource]. Update date: 11.01.2025. URL: https://github.com/natasha/natasha (Date of request: 14.01.2025).

6. **Emelyanov A., Shliazhko O., Katricheva N., Shavrina T.** Using RuGPT3-XL Model for RuNormAS competition. Computational Linguistics and Intellectual Technologies: *Papers from the Annual International Conference «Dialogue»*. 2021, iss. 20, ch. 18, pp. 204-212.

7. **Bochkarev V., Solovyev V.** Properties of the network of semantic relations in the Russian language based on the RuWordNet data. *Journal of Physics: Conference Series 1391*. 2019, pp.1-5. DOI: 10.1088/1742-6596/1391/1/012052.

8. **Hugo Touvron, Thibaut Lavril, Gautier Izacard at al.** *LLaMA: Open and Efficient Foundation Language Models*. arXiv:2302.13971.v1.27 Feb. 2023. Pp.1-27.

9. **Kashirin I.Yu.** Tokenizaciya politicheskih tekstov v BERT-modelyah s ispol'zovaniem ICF+-ontologij. *Informacionnye tekhnologii*. 2024, vol. 30, no. 12, pp. 622-632. DOI: 10.17587/it.30.622-632. (in Russian).

10. **Percy Liang, Rishi Bommasani, Tony Lee at al.** *Holistic Evaluation of Language Models, Transactions on Machine Learning Research*. 2023, iss. 08, pp.1-162.

11. Natural language models. Materials. [Electronic resource]. Update date: 11.01.2025. URL: https://i.kashirin.net/forstudents/modelsfornlp. (Date of request: 14.01.2025).

12. **Ying Li , Zichen Song, Weijia Li.** *Benchmarking Large Language Models in Adolescent Growth and Development: AComparative Analysis of Claude2, ChatGPT-3.5, and Google Bard.* Research Gate. https://www.researchsquare. com/article/rs-3858549/v1.

13. Falcon. [Electronic resource]. Update date: 21.10.2024. URL: https://the-decoder.com/falcon-180b-open-source-language-model-outperforms-gpt-3-5-and-llama-2/. (Date of request: 24.10.2024).

14. **Borgeaud S., Mensch A., Hoffmann J. at al.** Improving language models by retrieving from trillions of tokens. *Computer science. Computation and language*. DOI: https://doi.org/10.48550/arxiv.2112.04426.

15. **Kashirin I.Yu., Khoroshevsky V.F.** Development of an ATN-oriented linguistic processor (algebraic approach). *International Conf. «Artificial Intelligence»*. Suwalki. Poland. 15-20 June1987.

16. **Rubajlo A.V., Kosenko M.Yu.** Programmnye sredstva izvlecheniya informacii iz tekstov na estestvennom yazyke. *Al'manah sovremennoj nauki i brazovaniya*. 2016, no. 12 (114), pp. 87-91. (in Russian).

17. **Kashirin I.Yu.** Formal'nye programmnye mashiny dlya ob"ektno-orientirovannyh yazykov: S++// Vestnik Ryazanskogo gosudarstvennogo radiotekhnicheskogo universiteta. 2024, no, 89, pp. 56-64. DOI: 10.21667/1995-4565-2024-89-56-64. (in Russian).

18. *An Introductory Guide to OpenAI's API*. [Electronic resource]. Update date: 10.02.2024. URL: https://webreference.com/ai/api/. (Date of request: 10.01.2025).

19. **Muromcev D.I.** Ontologicheskij inzhiniring znanij v sisteme Protégé. SPb: SPb GU ITMO. 2007, 62 p. (in Russian).

20. **Grenader U., Miller M.** *Pattern Theory: From Representation to Inference*. Oxford University Press, Inc.198 Madison Ave. New York, NY United States. 2007. 596 p.

21. **Kafe E.** Persistent semantic identity in WordNet. *Cognitive Studies Études cognitive.* 2018 (18). https://doi.org/10.11649/cs.1717.