ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

УДК 007:681.512.2

ТЕОРИЯ ИЕРАРХИЧЕСКИХ ЧИСЕЛ В ЗАДАЧАХ ВЫЧИСЛЕНИЯ СЕМАНТИЧЕСКОГО СХОДСТВА ЕСТЕСТВЕННО-ЯЗЫКОВЫХ КОНСТРУКЦИЙ

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 алгебра иерархических чисел, операции и отношения алгебраической системы. Приводится графическое представление иерархических чисел и операций с ними, показываются замечательные свойства операций. Перечисляются и поясняются способы нормализации иерархических чисел для их последующего применения в обработке естественно-языковых конструкций. Для использования теории иерархических чисел разрабатываются онтологии моделей знаний в части родовидовых таксономий, имеющих также иерархическую структуру. Выделяются общие и прикладные онтологии, имеющие существенное различие в их конструкции и применении для понимания предложений естественного языка.

В качестве сквозного примера взята предметная область англоязычных политических статей международных электронных средств массовой информации, в частности: RT, cnn, TASS, NYTimes. Рассматривается технология вычисления семантического сходства естественно-языковых конструкций, для чего задействуются известные языковые нейросетевые модели bert-base-cased последних версий, а также авторская модель IYu-bert-cased. Представлен новый метод вычисления семантического сходства с использованием теории иерархических чисел.

Экспериментальная часть материала основана на применении программного инструментария языка Python v.3 (Anaconda 3): библиотека Spacy v.3.2.1, ретривер CorpusMining v.2.1, пакет программ mIYu-bert v.1.0. Последние два инструментария реализованы автором материала.

Выполненная серия экспериментов позволяет квалифицировать методологию применения иерархических чисел при вычислении семантического сходства как основу технологии, не уступающей по эффективности имеющимся на сегодняшний день международным аналогам.

***Целью работы** является презентация эффективного применения алгебры иерархических чисел для получения и использования новой нейросетевой технологии, применяемой для решения задач автоматического вычисления семантического сходства конструкций естественного языка.*

***Ключевые слова:** теория иерархических чисел, нейронные Bert-модели, анализ естественного языка, онтологические таксономии, семантическое сходство.*

DOI: 10.21667/1995-4565-2024-88-38-52

Введение

Естественно-языковыми конструкциями являются слова в различных формах, словосочетания, предложения и осмысленные тексты. Вычисление семантического сходства таких конструкций рассматривается учеными как главная составляющая задач релевантного информационного поиска, а также проектирования нейронных сетей и других моделей глубокого обучения (ML-моделях) для анализа естественного языка [1].

В частности, идея семантического сходства используется в ML-моделях, классифицирующих тексты политических статей электронных средств массовой информации (СМИ) по таким группам, как:

- фейк новости [2];
- статьи, провоцирующие гнев отдельных социальных групп [3];
- токсичные публикации, вызывающие подавленное состояние у читателя [4];
- статьи с положительным или отрицательным эмоциональным настроением [5];
- публикации, внедряющие прозападную идеологию [6];
- статьи, призывающие к интернационализму и миролюбию [7].

Характеристика смысловой близости текстов позволяет выделять иерархию классов, к которым можно отнести материалы СМИ различных государств.

Сказанное определяет *цель настоящей статьи*, которая заключается в создании формальных средств вычисления семантической близости естественно-языковых конструкций как теоретической основы для проектирования соответствующих прикладных алгоритмов.

Современная политическая ситуация, выраженная в небывалом обострении информационной войны, позволяет назвать сформулированную цель весьма актуальной. Для достижения сформулированной цели предлагается использовать теорию иерархических чисел [8, 9].

Теоретическая часть

Алгебра иерархических чисел

Иерархические числа – это числа вида [s] a_0 . a_1 . a_2 . … . a_i . … .a_n, где a_i – целые положительные числа из множества N = {0, 1, 2, 3…}. s – символ знака «+» или «-», положительный знак может не указываться. Например, иерархические числа могут выглядеть так: 0.0.12.48.0 или -2.33.0.0.4 . Символом, обозначающим множество иерархических чисел, является H.

Эти числа так или иначе уже используются в практике классификации или адресации, например, универсальный десятичный код или IP-адрес компьютера в глобальной сети. Однако введение алгебраической системы иерархических чисел позволяет совершать с ними операции, схожие с формальной арифметикой, и выделять бинарные отношения для их сравнения и анализа нетривиальных свойств операций и отношений.

Рассмотрим алгебру бинарных иерархических чисел.

Пусть B – множество чисел с элементами {0, 1}, n ϵ B (n = 0 или n = 1), пусть также есть выделенный символ «.».

Множество A = B ∪ «.» ∪ ʌ определяется как алфавит с целыми числами из B, где «∪» – операция объединения множеств, а ʌ – пустой символ. Тогда грамматика:

$\hat{h} \rightarrow$ ʌ, $\hat{h} \rightarrow$ h, $\hat{h} \rightarrow$ -h,

h → < n >, h → < n > . <h>

описывает множество бинарных иерархических чисел H с элементами h.

Можно привести примеры бинарных иерархических чисел: 0.1.0.0.1 или 1.0.-1.0 .

Бинарные иерархические числа представляют собой численные индексы вершин двух двоичных деревьев: положительного и отрицательного с одной общей вершиной 0.

Порождение вершины влево от 0 производится бинарной операцией 0+0=0.0, порождение вершины вправо - бинарной операцией 0 + 1 = 0.1.

Порождение отрицательных вершин выполняется операцией «-» соответственно:

0-0 = -0.0, 0-1 = -0.1.

Графически это можно представить деревом, распространяющимся в положительную или отрицательную сторону (рисунок 1):

Впрочем, использование отрицательных элементов может сделать смысл операций более сложным. Например, порождения трассы «+» может выглядеть так:

0.1 + 1.1 = 0.1.1.1, 0 + 0.1 = 0.0.1.

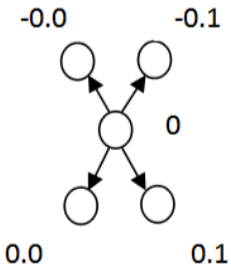

Рисунок 1 – Изображение алгебраических операций в форме деревьев
Figure 1 – Image of algebraic operations in the form of trees

Однако пример 0.0.-1 + 1.1 = 0.0.-1.1.1 свидетельствует о наличии более сложных трасс путешествия по дереву, использующих не только спуск но и локальные подъемы (рисунок 2). Такое применение иерархических чисел будет рассмотрено далее на конкретных примерах.

Операция, обратная порождению, − удаление терминальной вершины «--», является унарной:

0.1.1.1-- = 0.1.1, 0.1.0-- = 0.1, 0.-1.-1 -- = 0.-1.-1 .

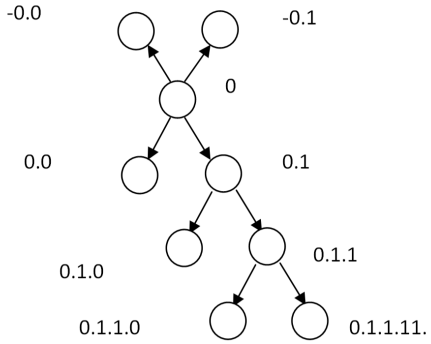

Рисунок 2 – Пример деревьев с поддеревом отрицательных иерархических чисел
Figure 2 – Example of trees with a subtree of negative hierarchical numbers

В графической интерпретации число можно считать абсолютным индексом какой-либо вершины, т.е. начинающимся с вершины дерева 0 или относительным, отображающим путь по дереву от одной из любых вершин к другим вершинам вверх и вниз. Абсолютный индекс всегда начинается с символа 0.

При решении практических задач можно рассматривать только положительную часть алгебры бинарных иерархических чисел. В этом случае операции, претендующие на получение отрицательного индекса, будут иметь результатом 0.

Можно привести еще одну довольно популярную операцию «^{o}», а именно вычисление наиболее общей вершины, что интерпретируется как поиск общего предка двух вершин-аргументов:

0.1.1.1 º 0.1.0 = 0.1.0 º 0.1.1.1 = 0.1.

Эта операция генерализации/умножения коммутативна, т.е. a º b = b º a.

Умножение положительного числа на отрицательное всегда равно 0.

Важной операцией является «∧» как вычисление пути из вершины, заданной первым аргументом, в вершину, заданную вторым аргументом. Для предыдущего рисунка можно было бы привести примеры таких вычислений:

0.1.0 ∧ 0.1.1.1 = 0.1.0. 0.1. 0.1.1. 0.1.1.1

0.1.1.1 ∧ 0.1.0 = 0.1.1.1 0.1.1. 0.1 0.1.0

0.1.0. 0.1. 0.1.1. 0.1.1.1 ≅ 0.1.1.1 0.1.1. 0.1 0.1.0

Здесь « ≅ » - отношение равенства длин двух иерархических чисел.

Однако такое вычисление приводит к излишне сложному результату.

Заметим, что общим предком для 0.1.1.1 и 0.1.0 является 0.1, с которого начинаются оба числа-аргумента. Вследствие этого, при вычислении операции «∧» эти фрагменты опускаются для всех вершин пути из первой вершины во вторую. Это необходимо для получения представления о сложности пути из первой вершины во вторую, несмотря на глубину дерева.

Тогда правильная операция «∧» получается так:

0.1.0 ∧ 0.1.1.1 = [0.1.]0. [0.1.] [0.1.]1. [0.1].1.1 = 0.1.1.1

0.1.1.1 ∧ 0.1.0 = [0.1.]1.1 [0.1.]1. [0.1] [0.1.]0 = 1.1.1.0

0.1.1.1 ≅ 1.1.1.0

После рассмотрения семантики приведенных операций можно задать универсальную арифметическую алгебру иерархических чисел H:

$$H = < H, \Omega >, \Omega = \{+, -, --, ^{o}, \wedge, \oplus\},$$

где Ω – сигнатура алгебры, т.е. множество операций. Все операции бинарны, за исключением «--», которая является одноместной. Смысл операции «⊕» будет рассмотрен позже, на соответствующих примерах.

Рассмотренную алгебру можно дополнить до алгебраической системы H = < H, Ω, R >, введя множество отношений R = {< , >, ≅ , = }, где отношения «a > b» и « b < a » соответственно «число a сложней числа b» и «число b короче числа a».

Применение иерархических чисел для проектирования онтологических таксономий

Таксономиями являются семантические отношения, задающие порядок на множестве понятий или свойств и используемые в современных моделях знаний. Главной таксономией является родовидовая, задаваемая отношением is-a. Более мощным отношением можно считать трехместное отношение icf (триада), образующее соответствующую icf-онтологию общего уровня.

Общие онтологии описывают понятия наиболее абстрактного уровня знаний, редко имеющие представление в словарных формах естественного языка. Однако их замечательным свойством является компактное описание комплекса базовых отношений: is-a, contr, form (i,c,f):

$$\text{icf}(a,b,c) = \text{is-a}(a, b) \cup \text{is-a}(a, c) \cup \text{form}(a, b) \cup \text{form}(a, c) \cup \text{contr}(b, c),$$

где is-a (a, b) означает, что понятие b принадлежит более общему классу a, form (a, b) – понятие a может проявляться в форме понятия b, contr (b, c) – понятия b и c каким-либо образом противоположны (по объему, содержанию и т.п.). Поскольку бинарные отношения – это множества пар, здесь используется операция теоретико-множественного объединения «∪». Более подробное описание триад icf можно найти в [10].

Для вычисления семантического сходства естественно-языковых конструкций предлагается использовать кроме классический языковых моделей, таких как bert-cased, word2vec [11], еще и онтологические таксономии, размеченные иерархическими индексами. Если языковые ML-модели являются основным механизмом вычисления сходства, то размеченные таксономии будут считаться дополнительным инструментарием или «средствами уточняющей подстройки».

Дальнейшие примеры будут взяты из предметной области «политические новостные статьи англоязычных электронных СМИ». Задача вычисления семантической близости в этом случае является важным элементом решения проблемы автоматической классификации политических статей по идеологической ориентации на «прозападные» и «пророссийские» [6].

Рассмотрим общую онтологию, описывающую тему «политические события». Она представлена на рисунке 3 и состоит исключительно из рассмотренных ранее icf-триад. Благодаря *полиморфическим свойствам icf-отношений* [10], все вершины дерева этой таксономии могут рассматриваться через преломление под углом зрения понятий, соответствующих любым другим вершинам.

Например, «Объекты» событий могут при каких-то условиях стать «Субъектами» и наоборот. Каждый из участников событий может быть в определенных условиях «Мирным», а может быть и «Боевым». То же самое можно сказать про «Тему события», которая может состоять в рассмотрении каких-либо «Объектов» или «Субъектов», и может быть «Мирной» или «Боевой».

Для приведенного примера операции с иерархическими числами могут оказаться полезными при вычислении гипонимов и гиперонимов понятий, а также определения общего гипероним двух или нескольких понятий. Например, для определения гипонимов одного уровня для понятия «Участники события» можно вычислить следующие выражения:

0.0.1+ 0 = 0.0.1.0 и 0.0.1 + 1 = 0.0.1.1,

что соответствует получению гипонимов «Объекты» и «Субъекты». Общий гипероним для понятий «Тема событий» и «Субъект» вычисляется с помощью операции «º»:

0.0.0 º 0.0.1.1 = 0.0,

что соответствует понятию «Элементы ситуации». Очевидно, что получение гиперонимов для какого-либо понятия производится операцией «--».

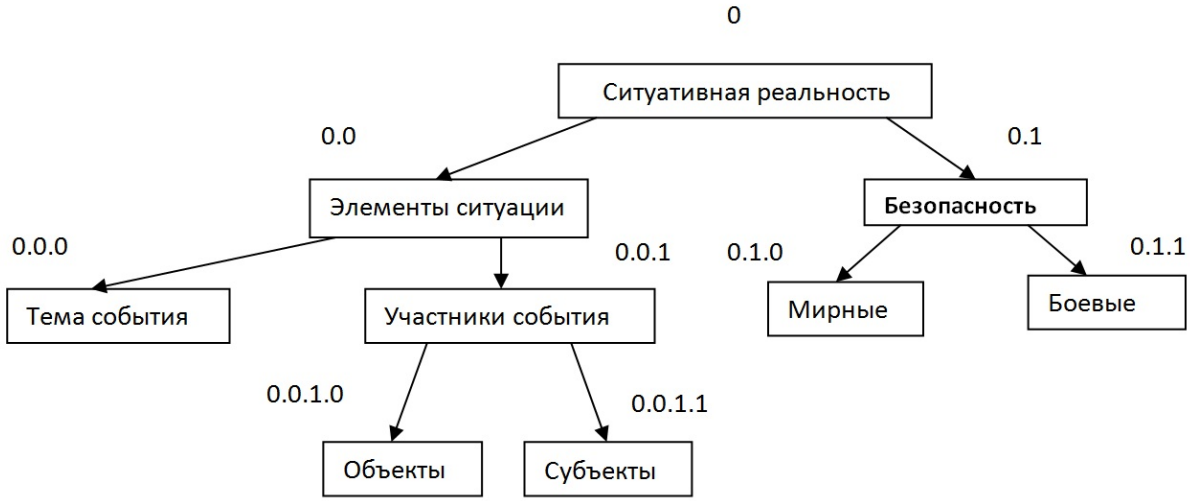

Рисунок 3 – Графическое представление общей онтологии «Политические события»
Figure 3 – Graphical representation of «Political events» general ontology

Нормализация иерархических чисел

Иерархические числа обладают многими признаками бинарных или десятичных чисел классической теории чисел и формальной арифметики, но в то же время в первоначальном виде их нельзя использовать в качестве числовых признаков при проектировании моделей глубокого обучения. Следовательно, для их использования может понадобиться нормализация, трансформирующая эти числа в десятичные из отрезка [0, 1]. Кроме того, из предыдущего изложения видно, что иерархические числа сконструированы таким образом, чтобы семантически близкие понятия индексировались близкими по структуре числами. Следовательно, при нормализации имеет смысл сохранить близость нормализованных чисел, соответствующих близким вершинам в иерархии. В задачах анализа естественного языка это одновременно означало бы и семантическую близость. Рассмотрим на примере известную идею семантического пространства, графически представленного рисунком 4.

Здесь заданы две оси семантических координат: «Объект-Субъект» и «Мирные-Боевые». Это значит, что все остальные понятия должны быть расположены в этом пространстве в зависимости от их близости в выбранной системе координат. Казалось бы, при соблюдении этого принципа каждое слово можно было индексировать двумя числами из предложенной системы координат. Однако предложенных измерений явно недостаточно, если расширить

систему базовых понятий, например, понятиями «Тема-Участники», также принадлежащими таксономии рассмотренного ранее примера (рисунок 5).

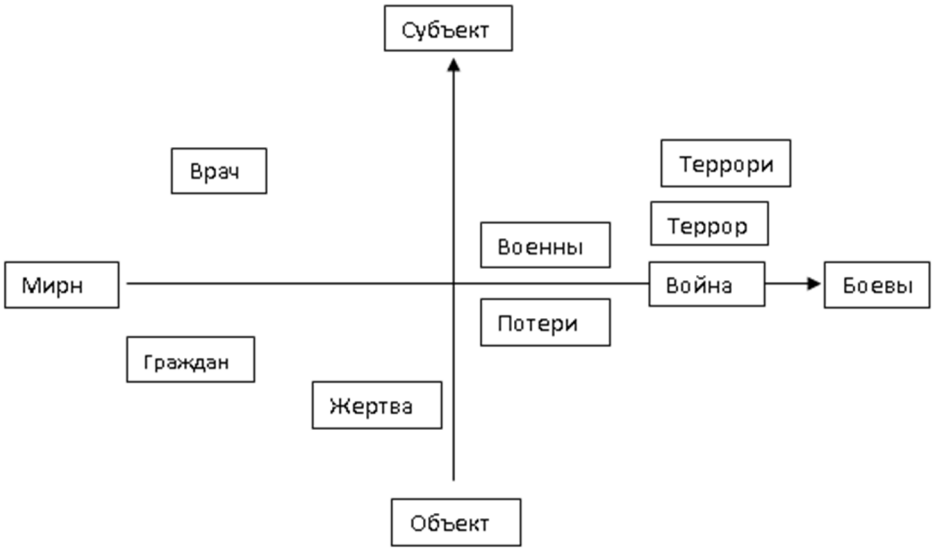

Рисунок 4 – Семантическое пространство для понятий из области «Политические события»
Figure 4 – Semantic space for concepts from «Political events» field

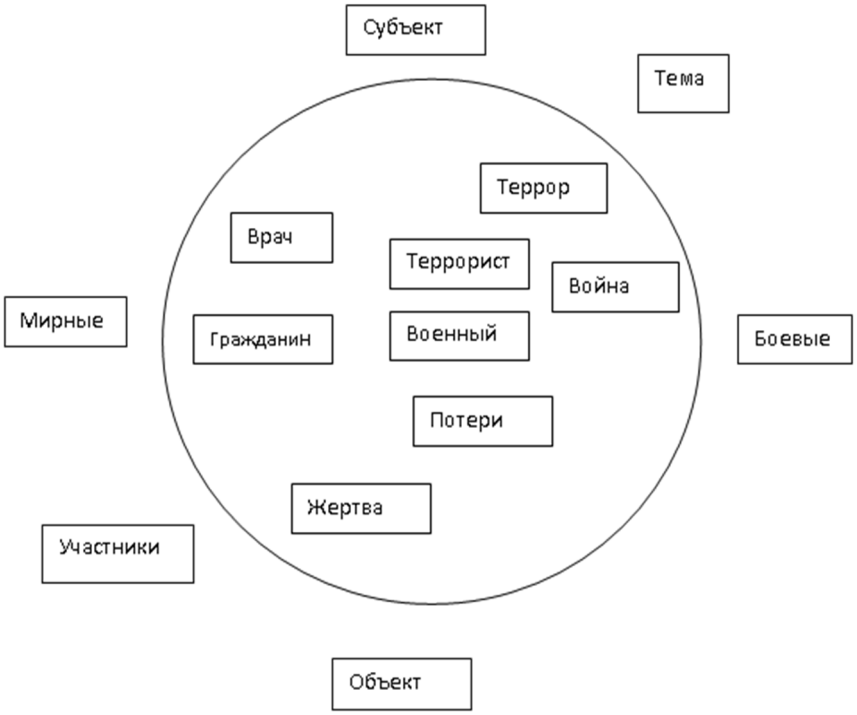

Рисунок 5 – Расширенное семантическое пространство
Figure 5 – Extended semantic space

Становится понятным, что передвигая объекты в сторону «Участников», мы одновременно увеличиваем близость к «Мирным», хотя это не всегда правильно. Следовательно, на плоскости разместить диаграмму концептов нельзя, и она многомерна от природы.

С другой стороны, можно выделить два вида семантического сходства:

- родовидовое сходство (синонимы, гипонимы);
- топологическое сходство (ситуативное, совместное присутствие слов в предложениях).

Как будет показано далее, оба этих вида можно реализовать в таксономии прикладного уровня. Такую обобщенную близость будем назвать *таксономической близостью.* В этом случае можно реализовать алгоритм, который является простым бинарным распределением (рисунок 6).

Из рисунка ясно, что с каждым новым уровнем иерархических чисел десятичные числа становятся все более подробными, так как появление новых уровней связано с разбиением предыдущих интервалов. На самых глубоких уровнях иерархии может происходить потеря точности кодирования чисел. В этом случае можно использовать логарифмическое распределение, приведенное на рисунке 7.

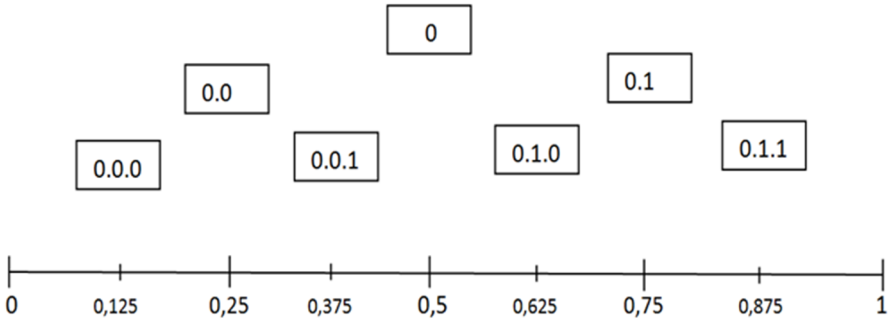

Рисунок 6 – Простое бинарное распределение иерархических чисел
Figure 6 – Simple binary distribution of hierarchical numbers

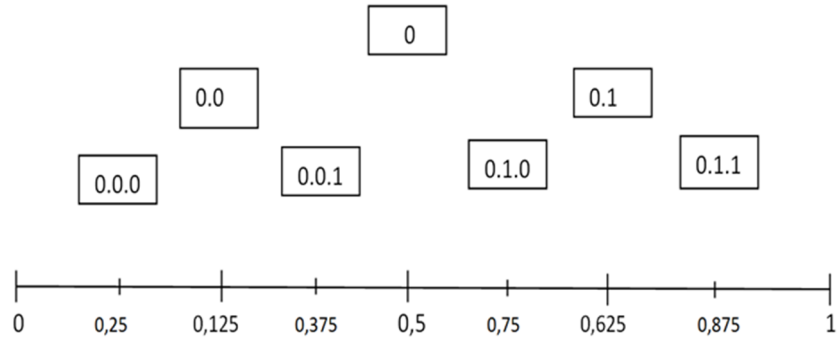

Рисунок 7 – Логарифмическое распределение иерархических чисел
Figure 7 – Logarithmic distribution of hierarchical numbers

Здесь реализован принцип близости гипонимов: более глубокие уровни чисел более близки по значению. Добавление каждого нового ряда в глубину приводит к перерасчету всех индексов вершин. Если обозначить через n точность таксономии (максимальное число уровней иерархии для текущей реализации таксономии), то распределение как функция нормализации Norm выглядит следующим образом:

Norm(0) = 0,5 : 0-0,4(9) 0,4(9)-0,9(9)

Norm(0.0), Norm(0.1) : $0,(0^n)1$ $0,9(9^n)$

Norm(0.0.0), Norm(0.0.1), Norm(0.1.0), Norm (0.1.1): $0,(0^{n-1})1$ $0,(0^{n/2})9$

Рассмотренные варианты нормализации применимы только к положительным бинарным иерархическим числам. Для более сложных случаев необходимо задействовать дробление десятичных чисел на еще большее число фрагментов. Такая нормализация остается за рамками настоящей статьи.

Иерархические числа для прикладных онтологических таксономий

Как отмечалось ранее, полиморфизм характерен только для общих таксономий моделей знаний. Реальные статьи СМИ содержать естественно-языковые конструкции, соответствующие понятиям прикладных онтологий, таксономии которых могут быть построены, например, на основе причинно-следственных отношений causa или но основе отношений is-a. В этих таксономиях могут встречаться фрагменты icf-триад, но это, скорее, исключительные

случаи. Однако для любых таксономий все операции и отношения алгебраической системы иерархических чисел остаются справедливыми.

В то же время для решения проблемы понимания текстов прикладные онтологии должны опираться на понятия общих онтологий. Это дает возможность использовать ограниченные формы полиморфизма, вводя множественное наследование. Такой пример дан на рисунке 8.

Здесь стрелки отношений для понятий прикладного уровня направлены вверх, указывая, существо каких гиперонимов они наследуют. Однако эти понятия являются гипонимами, т.е. более частными по отношению к понятиям, соответствующим вершинам более высоких уровней и общей онтологии. Это означает, что иерархические индексы прикладных понятий должны быть сформированы на основе индексов понятий общей онтологии, и поэтому они будут более сложными.

Рассмотрим теперь пример с конкретными предложениями из реальных электронных СМИ:

1."March 23, 2024 Shooting at Moscow concert venue leaves over 130 dead." (cnn)

2."On March 23, 2024, terrorists attacked Moscow, killing more than 130 citizens." (ru.wikipedia.org)

3."Terrorists strike the Russian capital, leaving at least 60 dead." (RT)

4."On 11.09.2001, two Boston planes destroyed the World Trade Center in New York." (NYTimes)

5."On October 11, 2022, a new multifunctional medical center was opened in Lugansk." (TASS)

1. "23 марта 2024 года в результате стрельбы на московской концертной площадке погибло более 130 человек".

2."23 марта 2024 года террористы совершили нападение на Москву, в результате которого погибло более 130 граждан".

3."Террористы наносят удары по российской столице, в результате чего погибло по меньшей мере 60 человек".

4."11.09.2001 два самолета "Бостон" уничтожили Всемирный торговый центр в Нью-Йорке ".

5."11 октября 2022 года в Луганске открылся новый многофункциональный медицинский центр."

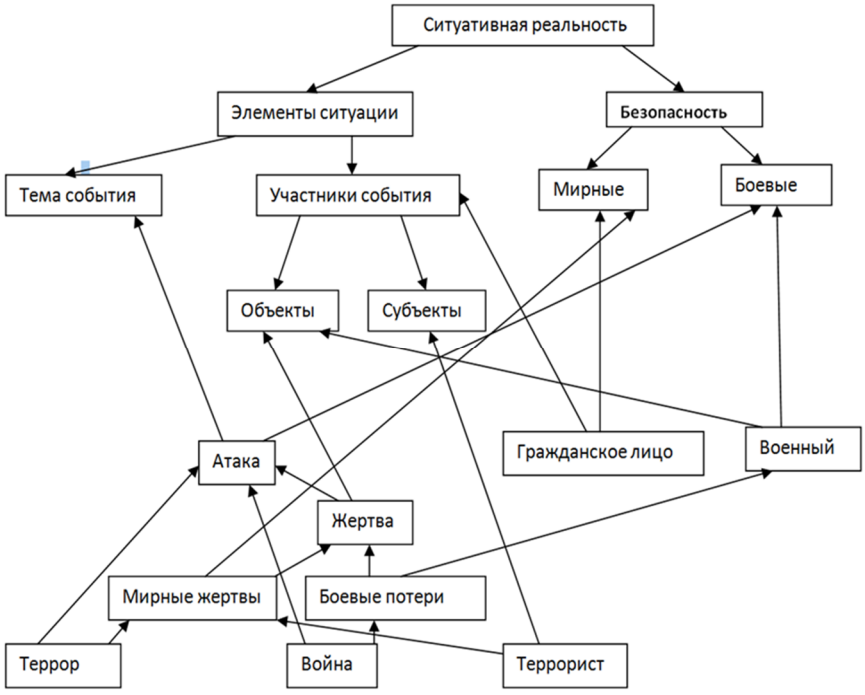

Рисунок 8 – Онтология с прикладной таксономией и множественным наследованием
Figure 8 – Ontology with applied taxonomy and multiple inheritance

Эти примеры демонстрируют предложения как сходные по смыслу, так и семантически различные.

Теперь для анализа естественно-языковых конструкций понадобятся более конкретные понятия, в той или иной форме присутствующие в приведенных предложениях. Как следует из рисунка 8, новыми понятиями теперь становятся: «Атака», «Жертва», «Гражданское лицо (Гражданин)», «Военный», «Мирные жертвы», «Боевые потери», «Террор», «Война», «Террорист».

Рассмотрим два метода индексирования вершин прикладной таксономии: *метод трассировки* и *метод множественного наследования*. Для реализации обоих методов понадобится расширение множества бинарных иерархических чисел до *комплексных иерархических чисел*, которые могут включать использование отрицательных элементов «-1». Заметим, что, как и в случае с классическими, например с десятичными числами, сама алгебра иерархических чисел остается неизменной. Требуется только четко определить, что они будут обозначать.

Метод трассировки

Числа означают запись трассы из одной вершины таксономического дерева в другую. Рассмотрим смысл операций алгебры Н с этой точки зрения.

Требуется определить новое понятие «Атака» как объединение понятий «Тема события» и «Боевая»:

Тема события (0.0.0) + Боевая (0.1.1) = Атака (0.0.0.-1.-1.1.1)

0.0.0 → 0.0 → 0 → 0.1 → 0.1.1 = 0.0.0.-1.-1.1.1

в нотации алгебраических операций: 0.0.0 - 1 - 1 + 0 = 0.0.0.-1.-1.1.1 , где -1 означает подъем на одну вершину дерева вверх.

Субъект + Мирные = Субъектное гражданское лицо

0.0.1.1 – 0.0.1 – 0.0 – 0 + 1 + 1 = 0.0.1.1.-1.-1.-1.0.1.0.1.0

в операциях: 0.0.1.1-1-1-1+1+0 = 0.0.1.1.-1.-1.-1.0.1.0.1.0

Субъект + Боевые = Атакующий военный

0.0.1.1 – 0.0.1 – 0.0 – 0 + 0.1 + 0.1.1 = 0.0.1.1.-1.-1.-1.0.1.0.1.1

Объект + Мирные = Гражданин

0.0.1.0 – 0.0.1 – 0.0 – 0 + 0.1 + 0.1.0 = 0.0.1.0.-1.-1.-1.0.1.0.1.0

Объект Атаки = Жертва Тема события + Боевые + Объект = Жертва

(Тема) 0.0.0.-1.-1.1.1 – 0.0.0. – 0.0 + 0.01 + 0.0.1.0 = 0.0.0.-1.-1.1.1.-1.-1.1.0

в операциях: 0.0.0-1-1+1+1 -1-1+0+1+0 = 0.0.0.-1.-1.1.1.-1.-1.0.1.0

Боевые + Жертва = Боевая жертва

Боевые + (Объект + Боевые + Тема события) , где индекс «Боевые » не дублируется в результате, так как операция «+» не ассоциативна.

Мирные + Жертва = Мирная жертва

Атака + Мирной жертвы = Террор

Атака Гражданина = Террор

Атака + Боевой жертвы = Война

Метод множественного наследования

Числа представляют собой индексы двух вершин-предков, разделенных цифрой «-1». Для получения таких чисел необходимо воспользоваться операцией синтеза множественного наследования «⊕». Здесь можно обратиться к рисунку 9, на котором представлены новые, более глубинные понятия прикладного уровня: «стрельба, удар, нападение, убийство, уход, разрушение, смерть». Эти понятия в словарной форме часто встречаются в политических статьях. Рассмотрим примеры.

Субъект + Мирный = Субъектное гражданское лицо (Сокращенная форма комплексного числа)

трасса в дереве:0.0.1.1 → 0.0.1 → 0.0 → 0 → 0.1 → 0.1.0

операция множественного наследования 0.0.1.1 ⊕ 0.1.0 = 0.0.1.1.-1.0.1.0

Объект Атаки = Жертва Тема события + Боевые + Объект = Жертва (Сокращенная форма комплексного числа)

трасса в дереве: 0.0.0. →1. → 1.1.1 → 0.0.0. → 0.0 → 0.01 → 0.0.1.0

операция множественного наследования:0.0.0⊕0.1.1⊕0.0.1.0 = 0.0.0.-1.0.1.1.-1.0.0.1.0

На рисунке 9 приведены иерархические индексы только для вершин, задействованных в рассмотренных примерах, поскольку отображение всех индексов сделало бы рисунок трудно читаемым.

Программная реализация использует для сравнения с существующими реализациями две языковые модели глубокого обучения и дополнительный инструментарий как средство уточняющей подстройки, упомянутое ранее.

Моделями являются:

– широко известная языковая модель последней текущей версии bert-base-cased, разработанная и постоянно обновляемая исследовательским отделом компании Google AI Language;

– модель IYu-bert-base-cased, полученная автором статьи на основе дообучения bert-base-cased (одной из предыдущих версий) на корпусах ретривера CorpusMining v.2.1.

Дополнительным инструментарием является пакет программ mIYu-bert v.1.0, также программно реализованный автором настоящей статьи.

Теория иерархических чисел на практике может быть использована несколькими способами.

Первым из них является полная реорганизация bert-модели корректировкой числа и содержания слоев энкодеров и декодеров нейронной сети в соответствии с количеством и содержанием уровней онтологической таксономии.

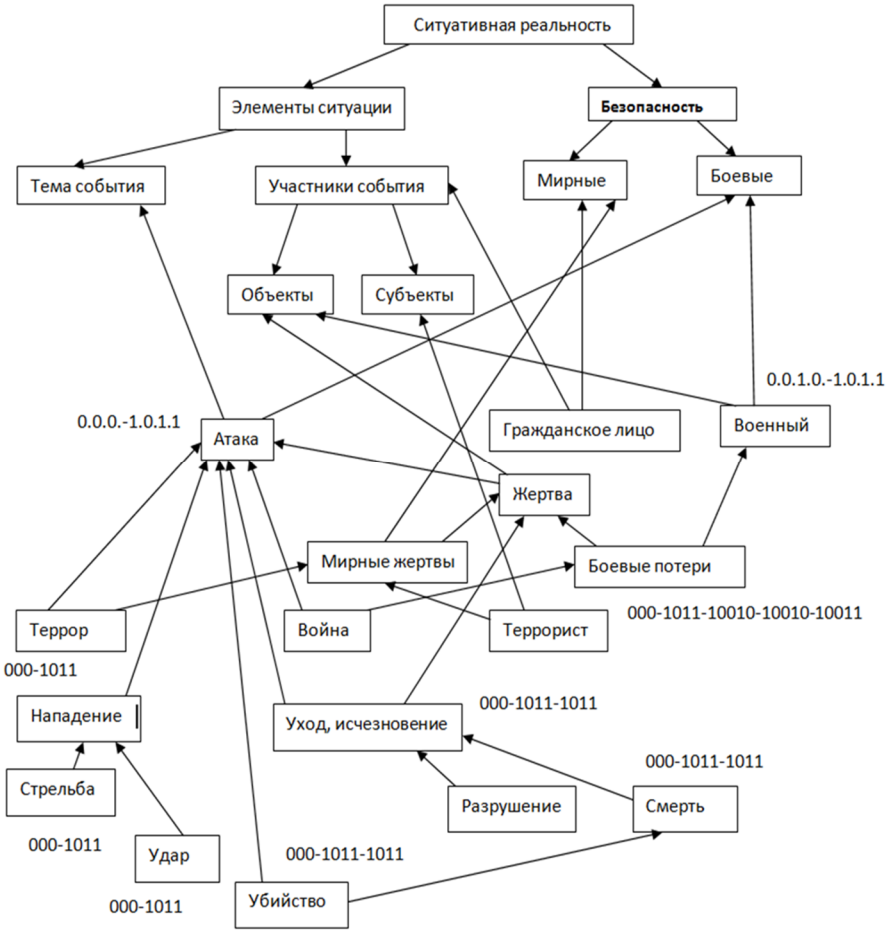

Рисунок 9 – Онтология с глубоким множественным наследованием
Figure 9 – Ontology with deep multiple inheritance

Экспериментальная часть

Программная реализация вычисления семантического сходства на языке Python

Иерархические индексы при этом могут являться основой векторизации слов в токенайзере bert-модели. В этом случае исходная bert-модель может быть существенно упрощена при условии сохранения характеристик точности. Однако этот способ не рассматривался в силу его большой трудоемкости, поскольку над базовой моделью трудились тысячи (!) программистов высокой квалификации.

Второй способ заключается в применении нормализованных иерархических чисел только в модернизации токенайзера bert-модели. Методы нормализация иерархических чисел были рассмотрены ранее. Этот метод также весьма трудоемок, но его реализация требует меньшего числа квалифицированных специалистов в сравнении с первым способом.

Третий способ программной реализации с использованием иерархических чисел представляет собой внедрение спецтокенов при векторизации естественно-языковых текстов. Он достаточно эффективен, реализован на ряде примеров и был рассмотрен в [6].

Четвертый способ, рассматриваемый в настоящей статье, предполагает предварительное вычисление семантического сходства языковыми моделями с последующей корректировкой характеристик дополнительным инструментарием. Инструментарий независимо от ML-моделей анализирует входные конструкции текстов на наличие схожих или противоположных понятий с помощью готовых онтологий, содержащих иерархические индексы. Сравнение индексов реализуется для слов, имеющих однотипную разметку при предварительном анализе инструментарием spaCy v.3.2.1. Семантическая близость вычисляется для двух слов разницей иерархических индексов на основе операции «^{o}» и иерархического коэффициента μ, являющегося пороговым для повышения или понижения характеристики семантического сходства, вычисленной языковыми моделями.

Определим функцию σ:

$\omega_i > \omega_j$, $\sigma(\omega_i, \omega_j) = 1$,

$\omega_i \le \omega_j$, $\sigma(\omega_i, \omega_j) = 0$,

где ω_i и ω_j – два произвольных иерархических числа.

Пусть ω_i^1 , ω_j^2 - иерархические числа, являющиеся индексами слов одинаковой синтаксической разметки в предложении 1 и предложении 2. Пусть предложение 1 содержит n слов, а предложение 2 содержит m слов.

Тогда можно записать выражение, вычисляющее корректирующий коэффициент W:

$$W = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m}\sigma\left(\omega_i^1, \omega_j^2\right)}{n * m},$$

если $\sum_{i=1}^{n}\sum_{j=1}^{m}\sigma\left(\omega_i^1, \omega_j^2\right) > \mu,$, то W берется со знаком «+», иначе со знаком « - ».

Далее приводится фрагмент программы на языке Python, вычисляющий семантическое сходство пяти предложений.

```
# Вычисление семантического сходства слов
    word1 = "attack"
    word2 = "shooting"
    word3 = "destroyed"
    word4 =  "killing"
    embedding1 = get_bert_embedding(word1)
    embedding2 = get_bert_embedding(word2)
    similarity = 1 - cosine(embedding1, embedding2)
    print(f"Семантическое сходство между словами '{word1}' и '{word2}': {simi-
larity}")
    embedding1_IYu = get_bert_embedding_IYu(word1)
```

```
    embedding2_IYu = get_bert_embedding_IYu(word2)
    similarity_IYu = 1 - cosine(embedding1_IYu, embedding2_IYu)
    print(f"IYu Семантическое сходство между словами '{word1}' и '{word2}': {similarity_IYu}")
    # Вычисление семантического сходства предложений
    sentence1 = "March 23, 2024 Shooting at Moscow concert venue leaves over 130 dead."
    sentence2 = "On March 23, 2024, terrorists attacked Moscow, killing more than 130 citizens."
    sentence3 = "Terrorists strike the Russian capital, leaving at least 60 dead."
    sentence4 = "On 11.09.2001, two Boston planes destroyed the World Trade Center in New York."
    sentence5 = "On October 11, 2022, a new multifunctional medical center was opened in Lugansk."
    embedding1 = get_bert_embedding(sentence1)
    embedding2 = get_bert_embedding(sentence2)
    embedding4 = get_bert_embedding(sentence4)
    embedding5 = get_bert_embedding(sentence5)
    similarity_sentences = 1 - cosine(embedding1, embedding2)
    print(f"Семантическое сходство между предложениями '{sentence1}' и '{sentence2}': {similarity_sentences}")
    similarity_sentences = 1 - cosine(embedding1, embedding4)
    print(f"Семантическое сходство между предложениями '{sentence1}' и '{sentence4}': {similarity_sentences}")
    similarity_sentences = 1 - cosine(embedding1, embedding5)
    print(f"Семантическое сходство между предложениями '{sentence1}' и '{sentence5}': {similarity_sentences}")
    embedding1_IYu = get_bert_embedding_IYu(sentence1)
    embedding2_IYu = get_bert_embedding_IYu(sentence2)
    similarity_sentences_IYu = 1 - cosine(embedding1_IYu, embedding2_IYu)
    print(f"IYu  сходство между предложениями '{sentence1}' и '{sentence2}': {similarity_sentences_IYu}")
    embedding5_IYu = get_bert_embedding_IYu(sentence5)
    similarity_sentences_IYu = 1 - cosine(embedding1_IYu, embedding5_IYu)
    print(f"IYu сходство между предложениями '{sentence1}' и '{sentence5}': {similarity_sentences_IYu}")
```

Результаты для выбранных языковых моделей и модели с использованием иерархических чисел (mIYu-bert) приведены в таблице 1.

Таблица 1 – Результаты вычисления семантического сходства
Table 1 – Results of semantic similarity calculation

№	Название издания	Сходство bert-cased	Сходство IYu-bert	Сходство mIYu-bert
1	cnn	1.0	1.0	1.0
2	ru.wikipedia.org(norm)	0.904102623462677	0.812596321105957	0.8631073324010911
3	RT	0.9349522590637207	0.8757287859916687	0.9289013442340876
4	NYTimes	0.8471251726150513	0.7009884119033813	0.6909823476905342
5	TASS	0.8411691188812256	0.6638354063034058	0.6100887839438400

В предложениях 1-3 в формулировках разных СМИ говорится о террористическом акте в Крокус Холле в Москве. Предложение 4 является примером новости об атаке торгового центра в Нью-Йорке. Предложение 5 представляет собой сообщение о вводе в строй нового медицинского муниципального центра в Луганске. Эксперименты проводились с вычислением семантического сходства различных пар пяти отобранных из СМИ предложений. В таблице приведены результаты сопоставления всех предложений с предложением 1. Эти данные наиболее ярко отражают и результаты других экспериментов.

Заключение

Произведенные эксперименты позволяют сделать вывод о небольшом изначальном проигрыше моделей IYu-bert и mIYu-bert при вычислении смыслового сходства одинаковых по смыслу предложений в сравнении с предобученной моделью bert-case. Однако IYu-bert и mIYu-bert существенно выигрывают при сравнении противоположных или весьма далеких друг от друга по смыслу предложений, хотя и находят их скорее схожими (0,66, 0,61), нежели разными.

В любом случае модифицированная на основе использования иерархических чисел модель mIYu-bert улучшает качество вычисления сходства в сравнении с последней версией bpdtcnyjq языковой модели bert-cased примерно на 5-8 %, причем корректировка результатов осуществляется как в сторону увеличения, так и в сторону уменьшения семантического сходства там, где это необходимо.

Библиографический список

1. **Демидова Л.А., Морошкин Н.А.** Аспекты разработки архитектуры вопросно-ответной системы для обработки больших данных на основе нейросетевого моделирования. // Вестник Рязанского государственного радиотехнического университета. 2023. № 86. С.145-155.

2. **Tida V.S., Hsu S.H., Hei X.** Unified Training Process for Fake News Detection based on Fine-Tuned BERT Model. *arXiv* 2022, arXiv:2202.01907.

3. **Abro S. et al.** (2020). Automatic hate speech detection using machine learning: A comparative study. International Journal of Advanced Computer Science and Applications, *11*(*8*), 484-491. https://doi.org/10.14569/IJACSA.2020.0110861.

4. **Su R, Wu H, Xu B, Liu X, Wei L.** Developing a multi-dose computational model for drug-induced hepatotoxicity prediction based on toxicogenomics data. IEEE/ACM Trans Comput Biol Bioinform 2019; 16:1231–9.

5. **Liu H.; Zhang Y.; Li Y.; Kong X.** Review on Emotion Recognition Based on Electroencephaloraphy. *Front. Comput. Neurosci.* 2021, 84.

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

7. Anastasyev A.A., Astashkin M.S., Agafonov P.A., Kashirin I.Yu. Determining the reliability of news using Ml-models, knowledge-based. / IIASU'23 – Artificial intelligence in management, control, and data processing systems. Proceedings of the II All-Russian scientific conference (Moscow, April 27–28, 2023): In 5 volumes. – Moscow, Publishing House «KDU», 2023. Volume 2. 406 p. Electronic edition. URL: https://bookonlime.ru/node/72807 – doi: 10.31453/kdu.ru.978-5-7913-1352-2-2023-406. P-21-27.

8. Definition of Hierarchial Numbers [Electronic resource]. Update date: 02.04.2024 URL: https://kashirin.net/definition-of-hierarchical-numbers. (date of application: .09.04.2024).

9. **Каширин И.Ю.** Иерархические числа для проектирования ICF-таксономий искусственного интеллекта // Вестник Рязанского государственного радиотехнического университета. 2020. № 71. С.71-82.

10. The Idea of ICF Relationships and ICF Ontologies [Electronic resource]. Update date: 18.02.2024 URL: https://kashirin.net/the-idea-of-icf-ontologies. (date of application: 05.04.2024).

11. **Xiangyun Lei1, Edward Kim, Viktoriia Baibakova1 and Shijing Sun.** Lessons in Reproducibility: Insights from NLP Studies in Materials Science / arXiv:2307.15759v1 [physics.chem-ph] 28 Jul 2023.

UDC 007:681.512.2

THEORY OF HIERARCHICAL NUMBERS IN CALCULATION PROBLEMS SEMANTIC SIMILARITY OF NATURAL LANGUAGE CONSTRUCTIONS

I. Yu. Kashirin, Dr. in technical sciences, full professor, RSREU, Ryazan, Russia;
orcid.org/0000-0003-1694-7410 , e-mail: igor-kashirin@mail.ru

The algebra of hierarchical numbers, operations and relations of an algebraic system are considered. A graphical representation of hierarchical numbers and the operations with them is provided; distinctive properties of the operations are shown. Methods for normalizing hierarchical numbers for their subsequent use in processing natural language constructs are listed and explained. To use the theory of hierarchical numbers, knowledge models ontologies are developed in terms of generic taxonomies, which also have hierarchical structure. General and applied ontologies having significant differences in their design and application for understanding natural language sentences are distinguished.

As a cross-cutting example, we took the subject area of English-language political articles of international electronic media, in particular: RT, CNN, TASS, NYTimes. The technology for calculating the semantic similarity of natural language constructions is considered, for which well-known bert-base-cased neural network models of the latest versions are used, as well as the author's IYu-bert-cased model. A new method for computing semantic similarity using hierarchical number theory is presented.

The experimental part of the material is based on the use of software tools of Python v.3 language (Anaconda 3): Spacy library v.3.2.1, CorpusMining v.2.1 retriever, mIYu-bert v.1.0 software package. The last two tools were implemented by the author of the material.

The completed series of experiments allows us to qualify the methodology for using hierarchical numbers in calculating semantic similarity as the basis of the technology that is not inferior in efficiency to currently available international analogues.

***The aim of the work** is to present the effective use of hierarchical number algebra to obtain and use new neural network technology used to solve the problems of automatic calculation of semantic similarity in natural language constructions.*

***Keywords:** hierarchical number theory, neural Bert models, natural language analysis, ontological taxonomies, semantic similarity.*

DOI: 10.21667/1995-4565-2024-88-38-52

References

1. **Demidova L.A., Moroshkin N.A.** Aspekty razrabotki arkhitektury voprosno-otvetnoy sistemy dlya obrabotki bol'shikh dannykh na osnove neyrosetevogo modelirovaniya. *Vestnik Rjazanskogo gosudarstvennogo radiotehnicheskogo universiteta*. 2023, no. 86, pp.145-155. (in Russian).

2. **Tida V.S., Hsu S.H., Hei X. A.** Unified Training Process for Fake News Detection based on Fine-Tuned BERT Model. *arXiv* 2022, arXiv:2202.01907.

3. **Abro, S. et al.** (2020). Automatic hate speech detection using machine learning: A comparative study. *International Journal of Advanced Computer Science and Applications, 11*(8), 484-491. https://doi.org/10.14569/IJACSA.2020.0110861.

4. **Su R, Wu H, Xu B, Liu X, Wei L.** Developing a multi-dose computational model for drug-induced hepatotoxicity prediction based on toxicogenomics data. *IEEE/ACM Trans Comput Biol Bioinform* 2019;16:1231–9.

5. **Liu H.; Zhang Y., Li Y., Kong X.** Review on Emotion Recognition Based on Electroencephalography. *Front. Comput. Neurosci.* 2021, 84.

6. **Kashirin I.Yu.** Neyroseti novogo mnogopolyarnogo mira: klassifikatsiya elektronnykh novostey. *Vestnik Ryazanskogo gosudarstvennogo radiotekhnicheskogo universiteta*. 2024, no. 87, pp.29-40. (in Russian).

7. **Anastasyev A.A., Astashkin M.S., Agafonov P.A., Kashirin I.Yu.** Determining the reliability of news using Ml-models, knowledge-based. *IIASU'23 – Artificial intelligence in management, control, and data processing systems. Proceedings of the II All-Russian scientific conference* (Moscow, April 27–28, 2023): In 5 volumes. – Moscow, Publishing House «KDU», 2023. Volume 2. 406 p. Electronic edition. URL: https://bookonlime.ru/node/72807 – doi: 10.31453/kdu.ru.978-5-7913-1352-2-2023-406. P-21-27.

8. Definition of Hierarchial Numbers [Electronic resource]. Update date: 02.04.2024 URL: https://kashirin.net/definition-of-hierarchical-numbers. (date of application: .09.04.2024).

9. **Kashirin I.Yu.** Iyerarkhicheskiye chisla dlya proyektirovaniya ICF-taksonomiy iskusstvennogo intellekta. *Vestnik Ryazanskogo gosudarstvennogo radiotekhnicheskogo universiteta*. 2020, no. 71, pp.71-82. (in Russian).

10. The Idea of ICF Relationships and ICF Ontologies [Electronic resource]. Update date: 18.02.2024 URL: https://kashirin.net/the-idea-of-icf-ontologies. (date of application: 05.04.2024).

11. **Xiangyun Lei1, Edward Kim, Viktoriia Baibakova1 and Shijing Sun.** Lessons in Reproducibility: Insights from NLP Studies in Materials Science / arXiv:2307.15759v1 [physics.chem-ph] 28 Jul 2023.