

УДК 004.4'22, 004.4'423, 004.434

ХІЛЬЧЕНКО ЄГОР АНДРІЙОВИЧ

БАКЛАН ІГОР ВСЕВОЛОДОВИЧ

РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СТВОРЕННЯ ЛІНГВІСТИЧНИХ МОДЕЛЕЙ

Анотація. В даній роботі були розглянуто поняття лінгвістична модель, її призначення у сучасному світі та етапи побудови. Був зроблений огляд популярних інструментів роботи з математичними сутностями та зроблений висновок щодо наявності в них можливості роботи з лінгвістичними моделями. Були запропоновані підходи для створення програмного забезпечення у вигляді інструменту роботи, побудови та аналізу лінгвістичних моделей на основі мови програмування Python.

КЛЮЧОВІ СЛОВА: лінгвістичні моделі, нечіткі множини, ймовірнісні функції належності, інтерпретатори.

Abstract. The concept of a linguistic model, its purpose in the modern world, and the stage of construction were discussed in this work. An overview of popular tools for working with mathematical entities was made and a conclusion was made regarding the possibility of operating with linguistic models. Approaches were proposed for creating software as a tool for working, building, and analyzing linguistic models based on the Python programming language.

KEYWORDS: Linguistic models, fuzzy sets, probabilistic membership functions, interpreters.

Вступ: На сьогоднішній день, одним із популярних напрямків в інформаційних системах є аналіз даних та прогнозування майбутніх значень на основі вже існуючих. Машинне навчання, статистичні моделі розроблені суто для цих цілей. Суміжною до цих теорій - є лінгвістичні моделі, що дозволяють зробити висновки у спрощеному представленні, тобто працювати не з конкретними значеннями, числами та робити точнісінько прогнози і висновки, а перетворювати вхідні дані в задані нечіткі множини в які перетворюються інтервальні дані. Такий спрощений варіант роботи дозволяє знаходити та аналізувати закономірне повторення в послідовностях, на основі яких можна виявляти критичні небезпечені відхилення, які несуть за собою потенційно тяжкі наслідки. Для цих цілей можна визначати емоційний стан працівників операторів складних систем, на основі руху рукою пульту керування, або комп'ютерної миші, чи джойстика, негативний емоційний стан яких може призвести до катастрофічних наслідків [1].

Основна частина: Зазвичай задачі побудови моделей машинного навчання, або статистичних моделей вирішуються у таких фреймворках, бібліотеки та мовах програмування, як: Python, Rlang, Scala Figaro,

Pyro, Matlab. На сьогоднішній день лінгвістичні моделі не мають своєї заточеної реалізації у перерахованих інструментах.

Зазвичай програмне забезпечення імплементування математичних теорій представляється у вигляді: мови програмування або суміжної до неї бібліотеки. Теорія побудови лінгвістичних моделей має наступні кроки [2]:

Попередня обробка даних: підготовка набору даних може виконуватися з ціллю зменшення шумів, аномалій або дуплікацій. Це дозволяє нормалізувати графіки та зробити їх стабільними.

Інтерналізація: один із основних кроків, розбиття множини значень різницевих вхідних даних. Залежить від набору вхідних даних, на скільки рівні проміжки між ними, та чи є можливість зробити логарифмічні, або рівноймовірні розбиття [3].

Лінгвістизація: в залежності від обраного способу інтервалізації, представляється у вигляді перетворення арифметичних різниць значень сусідніх інтервалів у лінгвістичні змінні. Для прикладу може бути використана теорія нечітких множин, та ймовірнісної функції

належності. Вводити такі змінні, краще всього на основі досвіду та знань специфіки конкретної предметної області, до якої належать вхідні дані.

Відновлення формальної граматики: це вже робота з кінцевим результатом, а саме лінгвістичний ланцюг. Побудова таблиць ймовірнісних даних, або пошук повторюваних послідовностей виконується в цьому пункті.

На основі чітко сформованих вимог та етапів за якими відбувається побудова лінгвістичних моделей, можна сформулювати та запропонувати рішення.

Для створення інструменту суто під реалізації цієї теорії, підійде інтерпретатор. Інтерпретатор - це програма, що дозволяє відразу виконувати команди вхідного тексту, мови, без попереднього перетворення його у байт код. Компілятор для цієї задачі не підходять, через те що вони роблять складний та ретельний аналіз вхідного тексту на сумісність типів та процес перетворюють його у байт код. Це все надлишкові операції, кращим рішенням буде виконати код, та зберегти результат у таких форматах як pdf, csc або xlsx.

Для створення інтерпретатора згідно теорії побудови мов програмування [4], треба розробити наступні компоненти системи:

Лексер - це перший та початковий компонент, суть якого перетворення вхідних рядків мови у структурні одиниці, що мають зазначений тип вхідного тексту, будь то синтаксично зарезервоване слово, чи символ.

Парсер - це другий етап, перевірки груп або послідовностей вхідних, перетворених структурних, типізованих одиниць на предмет

того, чи можна взагалі побудувати закладені в цю мову конструкції, синтаксичні вирази.

Виконувач - останній етап, який отримує на оброблені дані у вигляді конструкцій цієї мови. Ту вже йде логічна перевірка на те, як ці конструкції взаємодіють один з одним та чи можна викликати цю функцію, або оголосити цю змінну і так далі. Також тут вже і виконуються команди, які були прописані у вигляді тексту.

Стосовно іншого підходу, можна запропонувати більш простіший варіант зі створенням бібліотеки у готовій мові програмування. Однією з найпоширеніших мов програмування для виконання математичних обрахунків є Python. В такому випадку все обмежеться кількома окремими файлами та такими вбудованими конструкціями, як: клас, методи класу та іншими допоміжними бібліотеками, що є у відкритому доступі.

Ці два рішення можуть бути поєднані наступним чином. Спочатку буде створена бібліотека на мові Python, яка буде включати в себе класи лінгвістичної моделі з її поетапними методами побудови та обробки і ймовірнісні функції належності, завдяки яким буде здійснюватися класифікація інтервалів. Потім буде створений мовний синтаксис для роботи з цими сутностями, але вже для обрахунків буде використовуватися готова, оптимізована бібліотека. Це нормальна та часта практика, коли інструменти використовують інші бібліотеки за основу.

Висновки.

В роботі було досліджено термін лінгвістична змінна, розглянуто популярні мови програмування та бібліотеки, які використовуються для розв'язку математичних задач. Зроблено висновок, що сучасні інструменти обрахунку у вигляді програмного забезпечення не мають реалізацію для оперування лінгвістичними моделями. Було запропоновано два підходи до вирішення, які можна поєднати між собою: інтерпретатор та бібліотека, що є досить обґрунтованим та надійним підходом, який вже впроваджувався іншими популярними технологіями та має свою історію успіху.

Список літератури

1. Баклан, І.В., Степанкова Г.А., Імовірнісні моделі для аналізу та прогнозування часових рядів. Штучний інтелект. № 3 (2008): 505-515
2. Баклан, І. В. "Основні етапи побудови лінгвістичної моделі." Вісник Херсонського національного технічного університету 2 (2013): 43-46.
3. Баклан, І. В. "Інтервальний підхід до побудови лінгвістичної моделі." Системні технології 3 (2013): 3-8.
4. Ball T. Writing An Interpreter In Go. Thorsten Ball, 2018. 264 с.