УДК 004.9

Величко С.В., Кайдан Н.В.

[1] *студентка 1 курсу фізико-математичного факультету, ДВНЗ «ДДПУ»*
e-mail: sofiavelichko33@gmail.com, ORCID 0000-0002-2728-6796
[2] *кандидат фізико-математичних наук, доцент кафедри МНМ та МНІ, ДВНЗ «ДДПУ»*
e-mail: kaydannv@gmail.com, ORCID 0000-0002-4184-8230

НЕЧІТКА СИСТЕМА ОБРОБКИ ТЕКСТОВИХ ДАНИХ

Стаття присвячена проблемі порівняння текстових даних на основі нечіткої логіки та нейронних мереж. Наводяться відомості про можливості порівняння текстів на основі відстані Левенштейна, що реалізовано у модулі розширення FuzzyWuzzy для мови програмування Python. На основі цього модуля розглянута система, що виділяє ядро тексту з великої кількості текстів присвячених спільній тематиці.

Ключові слова: *нейронні мережі, нечітка логіка, обробка даних, відстань Левенштейна, порівняння текстів, ядро текстів*

Вступ

Постановка проблеми. Обробка інформації є однією із основних видів діяльності у інформаційному суспільстві. В сучасному світі з його глобалізацією та темпами росту наукомістких технологій та їх частки в житті людини, кількість інформації експоненціально зростає, відповідно до цього збільшуються вимоги до швидкості та якості її обробки. Автоматизація обробки даних, покращення методів та алгоритмів аналізу, находженнях нових підходів до машинного навчання є однією з найактуальніших потреб сьогодення.

Використовуючи машинні способи роботи з наборами даних, ми можемо працювати швидко з великими об’ємами, але з’являться інша проблема, неможливість повного відтворення людського типу мислення комп’ютерною системою. Виникає логічна спроба знайти проміжний, але ефективний підхід за допомогою нечітких систем аналізу даних, на базі нечіткої логіки, що можуть максимально наблизити машинний аналіз до імітації вибору людини, або навіть спробувати перевищити його за усіма показниками.

Метою статті є розкриття основних напрямів машинного аналізу тексту з позицій нечіткої логіки.

Основна частина

Аналіз даних це алгоритми обробки інформації що не мають фіксованої відповіді для кожного нового входження об’єкту обробки до моменту завершення самої обробки. Це якісно відрізняє такий алгоритм від класичних

підходів, наприклад, сортування рядків. На відміну від цього, від системи на базі нечіткої логіки, яка наприклад розпізнає рукописний текст, ми не можемо цього очікувати чи вимагати. До того ж будь-яка система на такій логікі може помилитися при будь-яких операціях обробки інформації, як це може зробити звичайна людина. Таку постанову задачі та алгоритми її вирішення прийнято називати недетермінованими, або нечіткими, в той час як класичний підхід є детермінованим, або чітким.

Вище зазначену задачу можна вирішити класичним підходом, якщо в ручному режимі підібрати функції, які реалізують відповідне відображення, що буде потребувати значних зусиль та часу, до того ж не може повністю гарантувати точність. В той же час, користуючись машинними алгоритмами навчання, лише використовуючи підготовлену вибірку даних, що як раз і є, вище зазначеним, недетермінованим підходом.

В сучасному світі машинне навчання є найбільш ефективним та перспективним напрямком розвинення систем збору та обробки інформації. На жаль алгоритми та методи для навчання системи, що зможе працювати з даними більш менш вільного формату, що потребують лише незначної обробки людиною, або взагалі не потребують, на даний час не існує. Така обробка називається фічеселектом (feature selection), або предпроцесінгом. Справа у тому, що більшість нечітких систем отримують на вхід дані певного формату та довжини. Так, раніше вони могли обробляти лише цифри, а в наш час робота може здійснюватися з більш складними, абстрактними поняттями. Наприклад текстами, аудіо чи зображеннями. Якщо описувати більш детально, наприклад роботи із зображеннями, то тут алгоритм обробляє не лише колір пікселя, а також його положення, сусідів, та їх параметри.

Широке розповсюдження отримали алгоритми машинного навчання з тренером, так можна назвати алгоритми, що беруть набір даних який складається із точок $(x_0, x_1, ..., x_{n-1})$ та міток, тобто значення що ми намагаємось передбачити $(y_0, y_1, ..., y_{n-1})$, а на виході дає алгоритм чи функцію, що вже виконує співставлення. На прикладі нейронної мережі для розпізнавання зображень, то тут із допомогою спеціальних процедур та на основі навчальної вибірки встановлюються значення, які відповідають відповідним нейронним зв'язкам. За допомогою цих зв'язків на кожному етапі обробки встановлюється те чи інше передбачення для кожного пікселя. Набір прикладів та міток називають навчальною моделлю.

Нажаль список алгоритмів машинного навчання з тренером доволі невеликий та майже не має напрямків розвитку, навіть не зважаючи на значні спроби та дослідження. Основними складнощами є важкість ефективного зведення практичної задачі до задачі аналізу та обробки даних, підбору процесів навчання та моделей.

Перед будь-яким включенням алгоритму машинного навчання чи нечіткої системи до налагоджених робочих процесів потрібна перевірка,

DOI: https://doi.org/...

інакше кажучи – перевірка якості та ефективності роботи. Це так звана валідаційна процедура. Вона виконується наступним чином – з однієї вибірки даних створюють дві, розділяючи початкову. Ці частини називають навчальна та валідаційна. Навчання мережі відбувається за навчальною вибіркою, в той час як якість перевіряється за валідаційною.

Сам цикл розвитку проекту з інтелектуального аналізу даних проходить наступні етапи:

1 Вивчення поставленої задачі з практичної точки зору, пошук потенційних джерел інформації.

2 Формування поставленої задачі на математичній мові, вибірка метрик якості.

3 Написання тестових алгоритмів та алгоритмів навчання.

4 Створення евристики, що вирішує поставлену задачу за допомогою найпростішого із можливих рішень та підходів.

5 Вирішення задачі алгоритмами машинного навчання.

6 По можливості, покращення та підвищення ефективності роботи.

Нейронні мережі не програмуються, в прямому сенсі цього слова, вони лише мають запрограмований алгоритм навчання як основу. Можливість навчання це головна якісна ознака такого підходу, цей варіант має значні преваги перед класичними алгоритмами. В теорії таке навчання є находженням коефіцієнтів якості зв’язку між нейронами. В процесі навчання нейрона мережа здатна виявити складні залежності між вхідними даними та результатами вибірки, та на основі цього виконати узагальнення. Це означає, що у випадку успішного навчання мережа може дати вірний результат на реальній вибірці даних, які були відсутні в навчальному сеті.

На практиці нейрона мережа часто є лише системою з’єднаних між собою в певній послідовності процесорів простих процесів, не тих що використовують для персональних комп’ютерів, які імітують нейрони. Кожен процесор працює лише з одним типом сигналу, що він отримує час від часу із зовнішніх джерел чи інших процесорів під час взаємодії. Окремо взятий такий процесор є не дуже потужним та корисним, але після об’єднання в таку мережу між собою, такі процесори здатні на обробку нелінійних надскладних операцій та задач.

Структуру найпростішої нейронної мережі можна побачити на рисунку 1. Зеленим кольором позначені нейрони вхідного слою, блакитним – скритого, та жовтим – вихідного.

Синопсис – це зв'язок між двома нейронами, що має лише один параметр – вага. Це значення зміни інформації при передачі між двома нейронами. Наприклад ви маєте три нейрони, що передали інформацію четвертому, це означає, що нейрон який прийняв ці сигнали має три ваги які відповідають кожному с попередніх нейронів, далі буде проведене визначення більшої ваги і така інформація буде домінуючою у приймаючому нейроні. Набор ваги мережі, чи інакше кажучи матриця ваги є свого роду

мозком системи, саме завдяки цьому явищу система навчається, а інформація оброблюється та перетворюється на результат.

Математична теорія нечітких множин (fuzzy sets) та нечітка логіка (fuzzy logic) є не чим іншим, як узагальнення комбінації класичної теорії множин та класичної логіки. Вперше це поняття ввів американський вчений Лотфі Заде (Lotfi Zadeh) в 1965 році. Основною причиною виникнення стала потреба в нечітких правилах алгоритмів та наближенні їх до людського міркування відносно описання об'єктів та процесів, що потребували машинної обробки інформації.

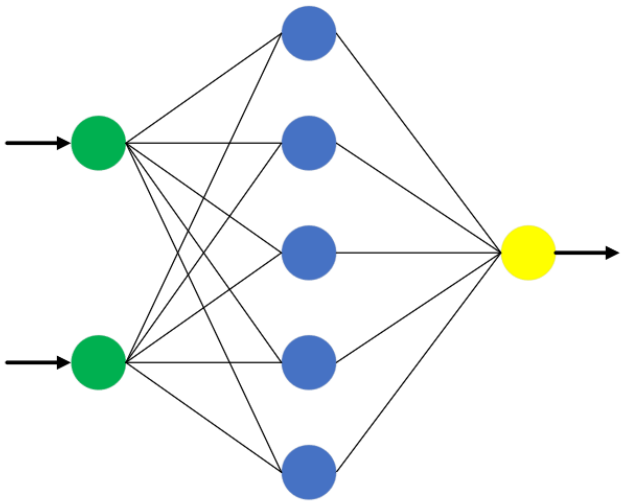

Рис. 1. Структура простої нейронної мережі

Перші такі системи активно почали використовувати для вирішення економічних питань, активно розвивався цей напрямок в медицині, що показує наскільки нечітка логіка важлива в системах зі значною кількістю змінних факторів. Третій, та останнім на даний момент, етап розпочався в 80-х роках. Отримали розповсюдження програмні пакети для побудови експертних систем на базі нечіткої логіки, а область де можна її використовувати значно зросла та продовжує свій ріст. На даний момент нечітка логіка отримала поширення в аерокосмічній, автомобільно будівній та транспортній системах, в області виготовлення побутової техніки, системи для підтримки прийняття рішень активно використовуються менеджерами усіх рівнів.

Домінування нечіткої логіки в розвитку таких систем почалось в кінці 80-х після того, як Бартоломей Коско (Bart Kosko) довів знамениту теорію FAT (Fuzzy Approximation Theorem), яка доводить що будь-яка математична система може бути апроксимована системою на нечіткій логіці [1].

Характеристикою нечіткої множини виступає функція приналежності (Membership Function). Позначимо що $MF(X)$ – коефіцієнт приналежності до нечіткої множини, яка представляє собою узагальнену характеристичну функцію звичайної множини. Тоді нечітка множина, яку ми позначимо як C,

DOI: https://doi.org/...

буде називатись множиною упорядкованих пар та матиме вид $C = \frac{MF(X)}{X}$, де $MF(X) \in [0,1]$, значення 0 означає відсутність належності до множини, а 1 – повна приналежність.

Спробуємо формалізувати нечітку множину під назвою «Гарячий чай». У якості області міркувань (X) візьмемо шкалу у градусах Цельсія. Як початок і кінець візьмемо шкалу від 0 до 100. Нечітка множина під назвою «Гарячий чай» матиме наступний вигляд:

$$C = \left\{ \frac{0}{0}; \frac{0}{10}; \frac{0}{25}; \frac{0.15}{35}; \frac{0.30}{45}; \frac{0.60}{50}; \frac{0.80}{60}; \frac{0.90}{70}; \frac{1}{85}; \frac{1}{90}; \frac{1}{100} \right\}.$$

З цього виходить, що чай за температури 50 градусів Цельсія належить до множини «гарячий» зі ступенем 0.6. Інакше кажучи, для однієї людини такий чай може видатись гарячим, а для іншої недостатньо чи зовсім холодним. Це є яскравим прикладом проявлення нечіткості задання відповідної множини.

Для нечітких множин, як і для класичних, мають місце, визначенні основні логічні операції. Основними серед них, необхідним мінімумом для розрахунків є визначення перетину та об'єднання.

Перетин двох нечітких множин, нечітке «І»

$$\text{A and B: } \mathbf{MF}_{AB}(X) = \min\left(MF_A(X), MF_B(X)\right)$$

Об'єднання двох нечітких множин, нечітке «АБО»

$$\text{A or B: } \mathbf{MF}_{AB}(X) = \max\left(MF_A(X), MF_B(X)\right)$$

В теорії нечітких множин розроблено та реалізовано підхід до виконання усіх можливих операторів перетину, об'єднання та доповнення. Все це реалізовано в так званих трикутних нормах та конормах. Вище зазначенні реалізації логічних операцій об'єднання та перетину є найбільш розповсюдженими випадками т-норми та т-конорми. Для опису нечітких множин також вводять поняття чіткої на нечіткої лінгвістичних змінних.

Нечітка змінна описується набором (N, X, A), де N – це назва змінної, X – універсальна множина (область міркувань), A – нечітка множина на X.

Значеннями лінгвістичної змінної можуть бути будь-які нечіткі змінні, тобто лінгвістична змінна знаходиться на більш високому рівні абстракції, ніж нечітка змінна. Кожна лінгвістична змінна складається з:

- назви;
- універсальної множини X;
- множини своїх значень, яка також називається базовою множиною (елементи базової множини являють собою назви нечітких змінних);
- синтаксичного правила, за яким генеруються нові терми із застосуванням слів природної або формальної мови;
- семантичного правила, яке кожному значенню лінгвістичної змінної ставить у відповідність нечітку підмножину множини X.

Розглянемо таке нечітке поняття як «Ціна акції». Це і є назва лінгвістичної змінної. Сформуємо для неї базову множину, яка буде складатися з трьох нечітких змінних: «Низька», «Помірна», «Висока» і поставимо область міркувань у вигляді X = [100; 200]. Останнє, що залишилося зробити – побудувати функції належності для кожного лінгвістичної підмножини із базової множини.

Сукупність функцій приналежності для кожної множини із базової множини зазвичай зображуються разом на одному графіку. На рисунку 2 наведено приклад описаної вище лінгвістичної змінної «Ціна акції».

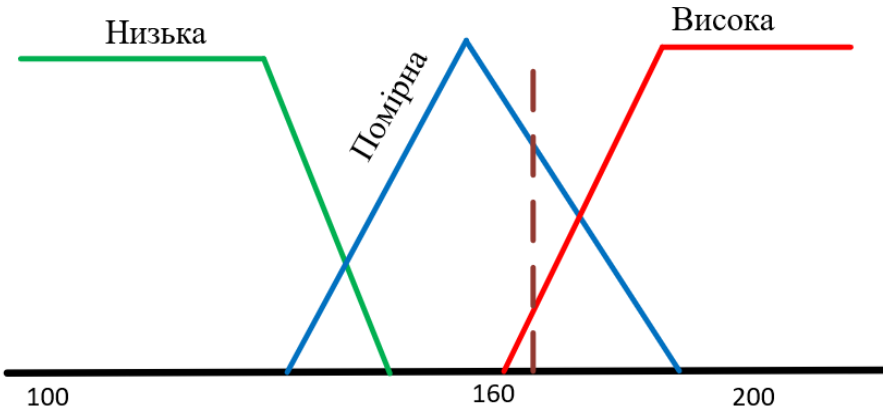

Рис. 2. Опис лінгвістичної змінної «Ціна акції»

Нейронечіткі системи або Нечіткі нейронні мережі – це системи з області штучного інтелекту. Вони комбінують методи штучних нейронних мереж і систем з нечіткої логіки. Нейронечіткі системи є результатом спроби створення гібридної інтелектуальної системи, яка б давала синергетичний ефект цих двох підходів шляхом комбінування людиноподібного стилю міркувань нечітких систем з навчанням і конекціонискою структурою нейронних мереж. Основна сила нейронечітких систем полягає в тому, що вони є універсальними апроксіматорами зі здатністю запитувати інтерпретовані правила ЯКЩО-ТО [2].

До переваг нейронечітких систем можна віднести дві суперечливі необхідності нечіткого моделювання, інтерпретованість та точність. Гібридизація методів інтелектуального аналізу стала новою віхою для досліджень у 90-х роках, у результаті об’єднання та синтезу окремих технологій систем аналізу інформації з’явився спеціальний термін – м’які розрахунки (soft computing). В сучасному науковому товаристві цей метод отримав широке розповсюдження, під цим поняттям об'єднують такі області як: нечітка логіка, штучні нейронні мережі, імовірнісні міркування і еволюційні алгоритми. Вони доповнюють один одного і використовуються в різних комбінаціях для створення гібридних інтелектуальних систем. Головним компонентом є нечітка логіка систем.

Подібно до того, як нечіткі множини, при їх відкритті, розширили рамки класичної математичної теорії множин, так і нечітка логіка зайняла

DOI: https://doi.org/...

широкі позиції практично в більшості систем інтелектуальної обробки інформації, наділивши їх новою функціональністю. Швидкі алгоритми навчання та інтерпретованість накопичених знань – ці фактори зробили сьогодні нечіткі нейронні мережі одним з найперспективніших і ефективних інструментів м'яких обчислень.

Нечіткі нейронні мережі (fuzzy-neural networks) здійснюють висновки на основі апарату нечіткої логіки, проте параметри функцій приналежності налаштовуються з використанням алгоритмів навчання класичної нейронної системи. Тому для підбору параметрів таких мереж застосуємо метод зворотного поширення помилки, спочатку запропонований для навчання багатошарового персептрона. Для цього модуль нечіткого управління представляється в формі багатошарової мережі. Нечітка нейронна мережа, як правило складається з чотирьох шарів: шару фазифікація вхідних змінних, шару агрегування значень активації умови, шару агрегування нечітких правил і вихідного шару. Найбільшого поширення в даний час отримали архітектури нечітких нейронних мереж виду ANFIS і TSK. Доведено, що такі мережі є універсальними апроксіматорами.

Процес аналізу текстових документів можна уявити як послідовність декількох кроків. Починаємо з пошуку інформації. На першому кроці необхідно ідентифікувати, які документи повинні бути проаналізовані, і забезпечити їх доступність. Як правило, користувачі можуть визначити набір аналізованих документів самостійно – вручну, але при великій кількості документів необхідно використовувати варіанти автоматизованого відбору за заданими критеріями.

Другий крок це попередня обробка документів. На цьому етапі виконуються найпростіші, але необхідні перетворення з документами для подання їх у вигляді, з яким працюють методи Text Mining. Метою таких перетворень є видалення зайвих слів і надання тексту більш чіткої форми, що передбачена алгоритмом обробки.

Третій крок являє собою вилучення інформації. Витяг інформації з обраних документів передбачає виділення в них ключових понять, над якими в подальшому буде виконуватися аналіз, слід зауважити, що даний етап є дуже важливим. На даному етапі витягуються шаблони і відносини, наявні в текстах. Даний крок є основним у процесі аналізу текстів, і практичних завдань.

Переходимо до інтерпретації результатів. Останній крок у процесі виявлення знань передбачає інтерпретацію отриманих результатів. Як правило, інтерпретація полягає або в поданні результатів на природній мові, або в їх візуалізації в графічному вигляді. Візуалізація також може бути використана як засіб аналізу тексту. Для цього беруться ключові поняття, які і подаються в графічному вигляді. Такий підхід допомагає користувачеві швидко ідентифікувати головні теми і поняття, а також визначити їх важливість.

Однією з головних проблем аналізу текстів є велика кількість слів у документі. Якщо кожне з цих слів аналізувати, то час пошуку нових знань різко зросте і навряд чи буде задовольняти вимогам користувачів. У той же час очевидно, що не всі слова в тексті несуть корисну інформацію. Крім того, в силу гнучкості природних мов формально різні слова, наприклад синоніми, які насправді означають однакові поняття. Таким чином, видалення неінформативних слів, а також приведення близьких за змістом слів до єдиної форми значно скорочують час аналізу текстів. Усунення описаних проблем виконується на етапі попередньої обробки тексту.

Зазвичай використовують такі прийоми видалення неінформативних слів і підвищення суворості текстів: видалення стоп-слів. Стоп-словами називаються слова, які є допоміжними і несуть мало інформації про зміст документа. Зазвичай заздалегідь складаються списки таких слів, і в процесі попередньої обробки вони видаляються з тексту. Типовим прикладом таких слів є допоміжні слова і артиклі, наприклад: «так як», «крім того», тощо.

Стемінг – морфологічний пошук. Він полягає в перетворенні кожного слова до його нормальної формі. Нормальна форма виключає схиляння слова, множинну форму, особливості усного мовлення. Наприклад, слова «стиснення» і «стислий» повинні бути перетворені в нормальну форму слова «стискати». Алгоритми морфологічного розбору враховують мовні особливості і внаслідок цього утворюють мовнонезалежний алгоритм.

N-грами - це альтернатива морфологічному розбору і видалення стоп-слів. N-грами - це частина рядка, що складається з N символів. Наприклад, слово «день» може бути представлено 3-грамою «_Де», «ден», «ень», «нь_», або 4-грамою «_ден», «день», «ень_», де символ підкреслення заміняє попередній або замикає слово пробіл. У порівнянні зі стемінг або видаленням стоп-слів, N-грами менш чутливі до граматичним і типографським помилок. Крім того, N-грами не вимагають лінгвістичного подання слів, що робить даний прийом більш незалежним від мови. Однак N-грами, дозволяючи зробити текст більш суворим, не вирішують проблему зменшення кількості неінформативних слів.

Приведення регістра. Цей прийом полягає в перетворенні всіх символів до верхнього або нижнього регістру. Наприклад, всі слова «текст», «Текст», «ТЕКСТ» наводяться до нижнього регістру «текст». Найбільш ефективно спільне застосування цих методів.

На основі всіх вище зазначених фактів було обрано напрям створення програмного забезпечення для порівняння двох текстів, на основі нечіткої логіки, відмовившись від навчання системи, через складність реалізації в рамках однієї роботи. Далі буде описано вибір інструментів, що були використанні, та програмну реалізацію системи, включаючи опис методів та інтерфейсу.

DOI: https://doi.org/...

Бібліотека FuzzyWuzzy реалізує механізм нечіткого аналізу для порівняння двох текстів на основі підрахунку відстані Левенштейна. Бібліотека працює з Python версії від 3.4 [3].

Основою роботи є функція ratio(), вона приймає два рядки тексту та порівнює їх, з врахуванням регістру символів. Функція повертає код порівняння, найбільше значення функції – 100. Порівнюючи рядки «Привіт світ» та «Привіт світ» функція поверне значення 100, а порівнюючи рядки «Привіт світ» та «Привт свт» отримаємо коефіцієнт 80. Наступна функція partial_ratio(), вона приймає два рядки тексту та шукає входження першого у другий, зважаючи на регістр. Наступні дві функції дозволяють робити прості порівняння в ситуаціях якщо різний регістр чи слова мають просто інший порядок. В цьому випадку використовують функцію token_sort_ratio(). Коли однакові слова повторюються підряд, то для функції token_sort_ratio() це будуть різні рядки. Тут на допомогу приходить найбільш просунута функція token_set_ratio().

Виявлення знань у тексті – це нетривіальний процес виявлення дійсно нових, потенційно корисних і зрозумілих шаблонів у неструктурованих текстових даних. Під «неструктуровані текстові дані» розуміється набір документів, що представляють собою логічно об’єднаний текст без будь-яких обмежень на його структуру. Прикладами таких документів є: веб-сторінки, електронна пошта, нормативні документи, наукові статті тощо. За допомогою функції бібліотеки FuzzyWuzzy ми порівнюємо тексти на їх ідентичність. Спільна частина тексту – це так зване ядро, з якого можна знаходити необхідні знання. Експеримент полягав в тому, щоб з великої кількості текстів на одну і ту саму тематику можна виявити тільки спільну частину. Саме ця спільна частина і є ядром сукупності текстів на яке і необхідно приділяти увагу.

Висновки

Нечітка логіка має багато спільного з процесом мислення людини, її можна використовувати для моделювання людського мислення. Поєднуючи нечітку логіку з нейронними мережами ми отримуємо засіб опрацювання даних, що здатен пришвидшити обробку неструктурованих даних. Навчання нейронної мережі виконується через вже існуючі результати, а тому ми завжди можемо покращити нейронну мережу новими отриманими результатами.

В результаті виокремлення спільної частини текстів заданої тематики ми отримуємо ядро текстів, тобто ту частину, що містить кожен із вхідних навіть якщо слова переставлені місцями, наявні помилки тощо. До подальших досліджень необхідно віднести реалізацію запропонованої системи та її тестування.

Література

1. Kosko B. Fuzziness vs. Probability. University of South California. URL: http://sipi.usc.edu/~kosko/Fuzziness_Vs_Probability.pdf
2. Hardesty L. Explained: Neural networks. MIT News Office. URL: https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414
3. FuzzyWuzzy documentation. Режим доступу: https://pypi.org/project/fuzzywuzzy/

Sofia V. Velychko, Nataliia V. Kaidan
Donbas State Pedagogical University, Sloviansk, Ukraine;

Fuzzy text data processing system

The article is devoted to the problem of comparing textual data based on fuzzy logic and neural networks. It provides information on the ability to compare texts based on Levenstein's distance, implemented in the FuzzyWuzzy extension module for the Python programming language. Based on this module, a system that distinguishes the core of the text from a large number of texts devoted to common topics is examined.

Keywords: *neural networks, fuzzy logic, data processing, Levenstein distance, text comparison, text core*.