УДК 004.94 https://doi.org/10.15407/pp2026.01.082

О.В. Захарова, Л.О. Спекторовська

ВИКОРИСТАННЯ МЕТОДІВ МАШИННОГО НАВЧАННЯ ДЛЯ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ПРОЦЕСУ АДМІНІСТРУВАННЯ КОСМЕТОЛОГІЧНИХ ПОСЛУГ

Робота присвячена вирішенню проблеми покращення якості надання косметологічних послуг у випадку стрімкого масштабування прикладної системи. Це супроводжується створенням великої кількості нових ролей та видів послуг, суттєвим розширенням клієнтської бази й комунікаційної мережі та, відповідно, значним збільшенням обсягів інформації, що потребує обробки. Метою дослідження є вироблення підходів, що дозволили б підвищити ефективність адміністрування косметологічних послуг шляхом автоматизації обробки вхідних повідомлень та їхньої багатокритеріальної категоризації. Як критерії для категоризації виділені: тип повідомлення, пріоритет, фах спеціаліста, що з ним пов'язаний, вид послуги.
В роботі також виконано огляд існуючих підходів з урахуванням постановки прикладної задачі, що дозволив дійти висновку про доцільність використання комбінації попередньої обробки текстових даних, методів витягнення ознак із класичними моделями машинного навчання для досягнення поставленої мети.
Ключові слова: машинне навчання, класифікація текстів, маршрутизація заявок, системи на основі правил, гібрідні системи, трансформери, категоризація повідомлень, прогнозування, навчальні дані

O.Zakharova, L. Spektorovska

USING MACHINE LEARNING METHODS TO IMPROVE THE EFFICIENCY OF THE COSMETOLOGICAL SERVICES ADMINISTRATION PROCESS

The article is devoted to solving the problem of improving the quality of cosmetic service provision during the rapid scaling of an applied system. This process is accompanied by the creation of a large number of new roles and types of services, a significant expansion of the client base and communication network. Accordingly, it also increases significantly the volume of information that requires processing. The aim of the study is to develop approaches that would increase the efficiency of administering cosmetic services through the automation of incoming message processing and their multi-criteria categorization. The criteria identified for categorization are: message type, priority, the specialist, and the type of service.
The paper also includes a review of existing approaches, taking into account the formulation of the applied task. This allows to conclude: to achieve the stated objective it is advisable to use a combination of text data preprocessing, feature extraction methods, and classical machine learning models.
Keywords: machine learning, text classification, rule-based systems, gibrid systems, transformers, message categorization, routing of requests, prediction, training data

Вступ

Вимоги та забезпечення ефективності будь-якої системи у сфері надання послуг насамперед обумовлюються розміром цієї системи. І сфера косметологічних послуг не є винятком. Якщо це косметологічний кабінет з мінімальною кількістю ролей, що забезпечують виконання базових функцій, то автоматизованого робочого місця зі стандартними можливостями обліку клієнтів, ведення їх запису, контролю розкладу завдань та надання пояснювальної (описової) інформації з доступом до веб є цілком достатнім. Зокрема, мінімальна кількість ролей в системі не передбачає реалізації складної автоматичної маршрутизації завдань. Але в процесі масштабування різко зростає на лише набір ролей та функцій системи, а й обсяг бази клієнтів і мережа комунікацій, включаючи джерела надходження інформації. Велику кількість різнотипних клієнтських заявок на отримання послуг потрібно «на льоту» класифікувати і розподіляти між ролями системи. Заявки можуть надходити до адміністраторів клі-

ніки з різних джерел: телефоном, з чату сайту клініки, електронною поштою тощо. Фактично це текстові повідомлення довільної форми, природною мовою різних форматів.

Це критично збільшує навантаження на адміністраторів косметологічної клініки, суттєво збільшує трудомісткість процесу адміністрування. Ручна обробка великої кількості різноманітних заявок породжує ризики помилкового розподілу задач та негативно впливає на ефективність роботи клініки в цілому. Тому задача автоматизації обліку та класифікації заявок на косметологічні послуги набуває актуальності під час масштабування системи надання послуг. А її вирішення потребує залучення сучасних технологій для виявлення семантик в природномовних контентах та подальшої динамічної семантичної класифікації текстових повідомлень, що надходять з різних джерел у різних форматах.

Постановка задачі

Задача полягає у динамічному зборі та обробці текстових повідомлень, що надходять з різних джерел. Джерелами вхідних повідомлень можуть бути: листи електронної пошти, зафіксовані письмово оператором call -центру/адміністратором в електронному журналі звернення по телефону, повідомлення з месенжерів Viber або WhatsApp, повідомлення з чату сайту клініки. Результатом має бути визначення категорії повідомлення за різними критеріями (рис.1), а саме:

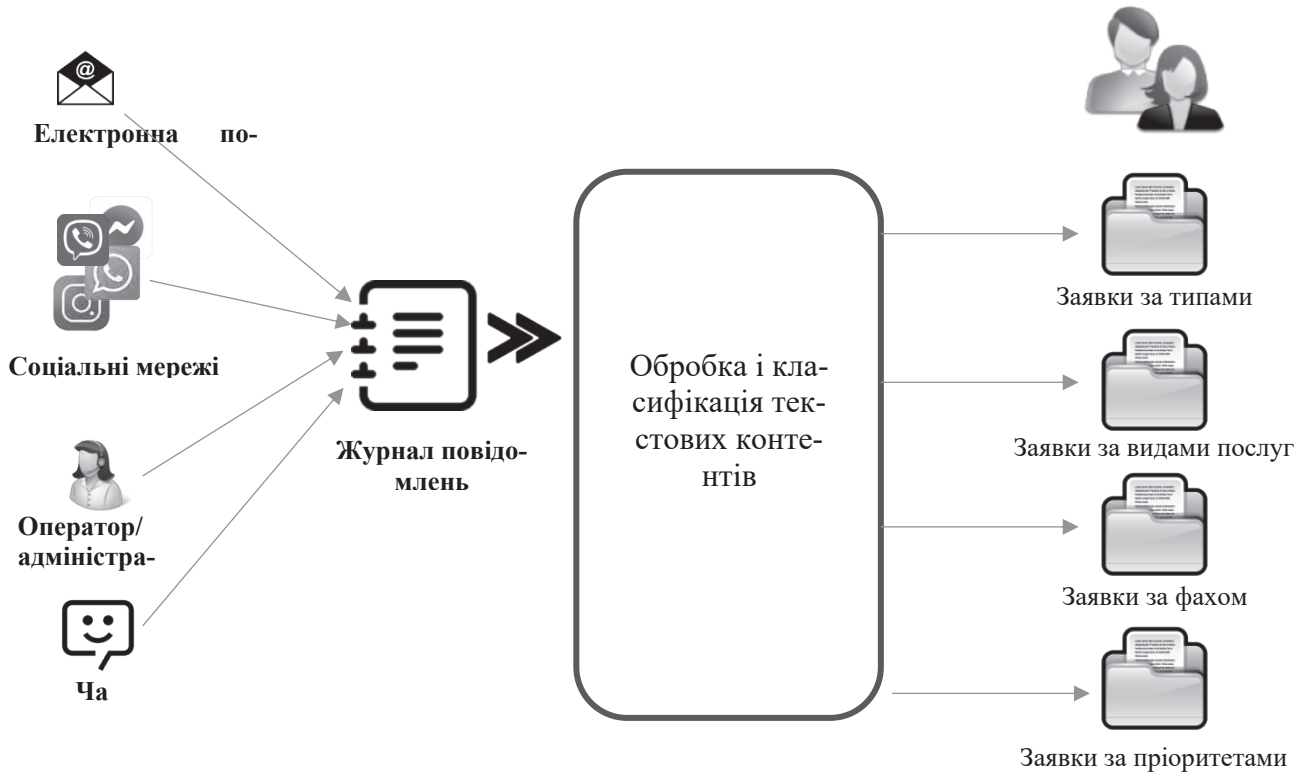

Рис. 1. Постановка задачі

– *Тип повідомлення* (запис на первинну консультацію, запис на косметологічну процедуру, відгук/скарга, технічне питання, адміністративне питання, питання/уточнення, запит на покупку косметологічних препаратів, спам тощо).

– *Пріоритет* (термінова заявка/звичайна).

– *Фах спеціаліста*, якому спрямована, можливо пряме призначення за прізвищем спеціаліста.

– *Вид послуг* (дерматологія, ін'єкційна косметологія, масаж, трихологія тощо).

Ведення журналу повідомлень має забезпечувати його динамічне поповнення

новими даними з різних джерел вхідної інформації, оновлення у разі зміни характеристик і стану заявок та видалення застарілих.

Як базові характеристики повідомлення можна виділити: номер (id); дата надходження; вид джерела надходження (чат, оператор, viber, whatsapp, дані про автора (електронна пошта, телефон, viber, whatsapp тощо), Instagram, messenger, електронна пошта); контент; дата останнього оновлення; статус заявки (нова, класіфікована, закрита, виконана); категорія заявки; вид послуг, якого вона стосується; фах і прізвище спеціаліста, якому спрямована; пріоритет.

Основна мета реалізації функції семантичного аналізу тексту в цій задачі полягає в класифікації текстових повідомлень на фіксовану кількість наперед визначених категорій, де одне повідомлення (в загальному випадку) може потрапити до кількох категорій, до однієї категорії, або не потрапити до жодної.

Тобто результатом обробки і класифікації текстових контентів має бути визначення характеристик класифікації (тип заявки; вид послуг, яких стосується; фах і прізвище спеціаліста, якому призначена; пріоритет заявки) на основі аналізу виявлених у контенті семантичних елементів. Загальна схема процесу представлена на рис.2.

Рис. 2. Загальна схема процесу обробки повідомлень

Огляд існуючих підходів

Головне завдання, що потребує вирішення в сформульованій вище задачі, це автоматична класифікація текстів. Слід зазначити, що автоматична класифікація тексту є однією з базових задач обробки природної мови. Умовно можно виділити три групи систем автоматичної класифікації текстів [1]: системи на основі правил, системи на основі машинного навчання та гібридні системи.

Системи на основі правил. Системи на основі правил [2] є одним з найпростіших підходів до класифікації. Для категоризації тексту вони використовують набір мовних правил, розроблених заздалегідь вручну. Кожне правило складається з шаблону та визначення категорії, що відповідає цьому шаблону. Тобто вони наказують системі класифікувати текст у певну категорію на основі його змісту, використовуючи в шаблоні семантично релевантні текстові елементи. Слід зазначити,

що за одним шаблоном в загальному випадку текстовий елемент може потрапляти до більш ніж однієї категорії.

Наприклад, можна визначити правило, що, якщо текст повідомлення містить слова «підліток» та «акне», то повідомлення належить до категорії видів послуг *Підліткова дерматологія*, а до категорії типів повідомлень *Спам* віднести тексти, що включають одне з наступних слів: «акція», «знижка», «розпродаж», «виграти», «не пропустіть» тощо. Повідомлення з назвою косметологічного препарату може бути віднесено як до категорії *Запит на покупку косметологічного препарату*, так і до «Питання/уточнення» чи «Відгук/скарга». Найпоширенішими формами правил, що використовуються в системах класифікації текстів, є регулярні правила (шаблони), правила на ключові слова, дерева рішень, словники тощо.

Досить відомими є такі реалізації систем на основі правил, як спам-фільтри [3], системи класифікації заявок (Helpdesk / Service desk routing), системи класифікації новин за словниками, системи виявлення тональностей за правилами (Vader [4], словники SentiWordNet [5]), платформи Rasa rules [6] та Dialogflow rules [7] – класифікація намірів у чат-ботах, які використовують як правила на ключові слова, так і шаблони й прості деревоподібні правила тощо.

Перевагою систем на основі правил є їхня простота і зрозумілість, до недоліків можна віднести їхню трудомісткість (вимагають багато часу для ручного створення правил, ретельного вивчення предметної області й тестування) та складність у масштабуванні (додавання нових правил може змінити результати вже існуючих). Також системи на основі правил є складними в обслуговуванні та масштабуванні.

Системи на основі машинного навчання. Задача класифікації є однією з класичних задач машинного навчання (ML). Для автоматичної класифікації текстової інформації системи на основі ML використовують різноманітні алгоритми та моделі. Вони зазвичай навчаються на зразках розмічених текстів (навчальна вибірка). Це дозволяє під час навчання зрозуміти певні закономірності у вхідних текстах і в подальшому правильно класифікувати нові зразки.

Серед найбільш використовуваних класичних моделей машинного навчання для класифікації текстів варто виділити наступні.

Multinomial Naive Bayes (MNB) [8] алгоритм належить до родини наївних Баєсівських моделей. Це клас моделей, заснований на теоремі Баєса та припущенні умовної незалежності ознак від заданої мітки класу. Незалежність ознак передбачає, що у встановленні належності елемента певної категорії, яка визначається множиною ознак, кожна ознака елемента розглядається незалежно від інших. Такий підхід є ефективним для вирішення задач, де дискримінативні слова домінують у приналежності до класу. Мультиноміальна модель, як правило, використовується саме для класифікації текстів, де для кожної цільової категорії y, що визначається n ознаками, розподіл визначається параметризованим вектором $\theta_y = (\theta_{y_1}, \dots, \theta_{y_n})$, де θ_{y_i} – ймовірність ознаки i в екземплярі класу y, що обчислюється підрахунком відносної частоти появи ознаки i в класі y в тренувальному наборі даних.

Слід зазначити, що припущення про незалежність ознак не завжди є вірним, і в таких випадках метод може бути не достатньо ефективним. Але, попри це, наївна баєсівська модель часто забезпечує доволі конкурентоспроможну продуктивність у задачах класифікації коротких текстів.

Linear SVM (Support Vector Machine) [9] - лінійний класифікатор, що базується на принципі мінімізації структурного ризику [10] з теорії обчислювального навчання. Ідея полягає у знаходженні гіпотези h, для якої можна гарантувати найменшу істинну похибку. Істинна похибка гіпотези h - це ймовірність того, що h дасть помилку на не- побаченому та випадково вибраному тестовому прикладі. Верхня межа може бути використана для зв'язку істинної похибки гіпотези h з похибкою цієї гіпотези на

навчальному наборі даних та складністю простору гіпотез, що містить h, H (що вимірюється розмірністю Vapnik–Chervonenkis (VC) [11]). SVM знаходить таку гіпотезу *h*, яка мінімізує (наближено) цю межу істинної похибки шляхом ефективного та результативного контролю VC-розмірності простору H.

Якщо набір тренувальних даних представити як множину точок $(x_i, y_j), i = \overline{1,n}, j = \overline{1,p}$, де x_i – текстовий елемент, що підлягає класифікації, а y_j – визначає належність x_i до певного класу j, і може приймати одне з двох значень: 1 – належить, -1 – не належить. То мета SVM полягає у розділенні всієї множини точок xi, для яких $y_j = 1$, від тих точок, для яких $y_j = -1$, гіперплощиною (межу) з максимальним «зазором» і забезпечити мінімальну похибку класифікації. Результативна оцінка визначається до цієї межі.

SVM є дуже універсальними навчальними системами, які у своїй базовій формі вивчають лінійну порогову функцію. Але у разі певного вдосконалення (простим «підключенням» відповідної функції ядра), можуть бути використані й для навчання поліноміальних класифікаторів, мереж радіальних базових функцій (RBF) та тришарових сигмоподібних нейронних мереж.

Logistic Regression [12] ще один метод лінійної класифікації, що моделює ймовірність належності текст до категорії на основі ознак типу Bag-of-Words або TF-IDF. Загалом лінійна класифікація стала одним із найперспективніших методів навчання для великих розріджених даних із величезною кількістю екземплярів і ознак. Логістична регресія, як і SVM, оперує даними як множиною точок $(x_i, y_j), i = \overline{1,n}, j = \overline{1,p}$, де x_i – текстовий елемент, що підлягає класифікації, а y_j – визначає належність x_i до певного класу j, і може приймати одне з двох значень: 1 – належить, -1 – не належить. Обидва методи вирішують ту саму задачу оптимізації, але використовують різні функції втрат. На відміну від SVM, результатом методу логістичної регресії є ймовірність того, що x_i належить класу *j*.

Серед нейронних моделей (глибокого навчання) для класифікації текстів найбільш використовуваними на сьогодні є:

TextCNN [13], як і більшість моделей глибокого навчання, що працюють з природним текстом, розглядає текст як послідовність векторних представлень слів (word embeddings). Ідея полягає в тому, що слова проєктуються з розрідженого кодування 1-з-N (де N – розмір словника) на векторний простір нижчої розмірності через прихований шар. Ці вектори слів по суті є витяжкою ознак, що кодують семантичні ознаки слів у їхніх вимірах. У таких щільних представленнях семантично близькі слова також є близькими (за евклідовою або косинусною відстанню) й у векторному просторі нижчої розмірності.

CNN (згорткові нейронні мережі) використовують шари зі згортковими фільтрами, які застосовуються до локальних ознак. Згорткові фільтри ковзають по послідовності векторів слів та “виявляють” локальні шаблони: характерні n-грамні шаблони, ключові фрази, шаблони емоційної лексики, типові для певної категорії слова.

Модель TextCNN є досить ефективною і швидкою в навчанні й успішно працює на коротких текстах (запити, відгуки, спам тощо), але, слід зазначити, що якість отриманого результату напряму залежить від якості векторного представлення слів.

RNN/LSTM/GRU. Рекурентні нейронні мережі (RNN) [14] – клас нейромереж, що був спеціально розроблений для роботи з такими послідовностями даних (sequence data) як текст. На відміну від класичних моделей, RNN обробляють текст покроково (слово за словом) і мають прихований стан, що дозволяє переносити інформацію з попередніх кроків. Фактично наступний крок (прихований стан на кроці *t*) є нелінійною функцією, що враховує векторні представлення слів на даному кроці, попередній крок (прихований стан на кроці *t-1*) і деякі параметри моделі.

LSTM є модифікацією RNN, що була створена для вирішення проблеми довгих залежностей, яка існує в RNN.

LSTM вводить комірки пам'яті (memory cell) та гейти (gates), які контролюють: що саме треба запам'ятати, що забути, а що передати далі в наступний стан. GRU є спрощеним варіантом LSTM, який має менше параметрів і швидше навчається.

Також слід виділити групу трансформерних моделей, які зараз є найпопулярнішими і активно використовуються в системах обробки природньої мови (NLP). Вони є також класом нейромережевих архітектур, але, на відміну від вище розглянутих, базуються не на рекурентних зв'язках, а на механізмі самоуваги, що є їхньою ключовою інновацією. Механізм самоуваги дозволяє моделі фіксувати контекстуальні зв'язки між усіма токенами в послідовності.

Механізм самоуваги дозволяє моделі визначати, на які слова в реченні потрібно звернути увагу, щоб краще інтерпретувати поточне слово. Ця властивість особливо важлива для розуміння неоднозначних фраз, довгострокових залежностей та полісемії.

Трансформерна модель була вперше запропонована 2017 року і швидко стала основою більшості сучасних моделей NLP, включаючи *BERT* та *RoBERTa***,** що є найвикористовуванішими серед моделей цієї групи.

BERT (Bidirectional Encoder Representations from Transformers) [15] – найбільш поширена модель для класифікації текстів, що спочатку навчається на великих мовних корпусах з використанням самоконтрольованих цілей, а потім налаштовується для виконання конкретних завдань, зокрема, класифікації тексту. Точне налаштування зазвичай вимагає додавання класифікаційної структури (наприклад, онтології) після трансформаторного кодера та навчання моделі на позначених прикладах.

BERT усуває обмеження односпрямованості, коли мовна модель попередньо навчається зліва направо, використовуючи «моделі маскованої мови» (MLM) як мету попереднього навчання. Ідея полягає в тому, що модель маскованої мови випадковим чином маскує деякі токени з вхідних даних, а метою є прогнозування оригінального ідентифікатора словника замаскованого слова лише на основі його контексту. MLM дозволяє представленню об'єднати лівий та правий контексти, що дозволяє попередньо навчити глибокий двонаправлений трансформатор. На додаток до моделі маскованої мови використовується також завдання «передбачення наступного речення», що разом з MLM (спільно) попередньо навчає представлення текстових пар.

Порівняно з методами на основі частотних векторів, класифікатори на основі BERT мають змогу краще узагальнювати синоніми та різні формулювання одного й того ж запиту, забезпечуючи високу якість отриманого результату.

Модель RoBERTa [16] є покращеним варіантом навчання моделі BERT. Внесені до BERT модифікації включають: довше навчання моделі, з більшими пакетами, на більшій кількості даних; видалення цілі прогнозування наступного речення; навчання на довших послідовностях; та динамічну зміну шаблону маскування, що застосовується до навчальних даних.

Окрім цього, RoBERTa збирає новий набір даних (CC-NEWS) досить великого розміру порівняно з іншими приватно використовуваними наборами даних, що дозволяє краще контролювати вплив розміру навчального набору на результат. Дане покращення моделі показало, що використання більшої кількості даних на етапі попереднього навчання моделі значно покращує продуктивність вирішення задачі.

XLM-R (XLM-RoBERTa) [17] – це багатомовна трансформерна модель типу encoder-only, побудована на архітектурі RoBERTa і дозволяє отримувати контекстні представлення тексту більш, ніж 100 мовами. Модель є розвитком одразу трьох ідей, а саме: підходу до двоспрямованого кодування контексту BERT, ідеї оптимізованого навчання, що реалізована в RoBERTa, та технології «cross-lingual language modeling» (XLM).

XLM-R навчається за раніше згаданою схемою MLM. Її головною відмінністю є використання дуже великого багатомов-

ного корпусу для попереднього навчання моделі. Це визначає головний напрямок її застосування – вирішення багатомовних задач, де XLM-R показує високу якість результату. Прикладами таких задач може бути: багатомовна класифікація тексту, аналіз тональності для різних мов, обробка тексту з міжмовними переходами (приміром, попереднє навчання моделі англійською мовою, а працює українською).

Слід зазначити, що моделі BERT, RoBERTa та XLM-R досягають найвищої точності, коли межі класів залежать від семантичного контексту, а не від ключових слів. Їхніми основними недоліками є досить високі, особливо порівняно з класичними ML моделями, обчислювальні вимоги та довший час навчання й логічного висновку.

Відомі на сьогодні *Великі Мовні Моделі (LLM)* поки залишаються досить дорогим рішенням для класифікації текстів із доволі непередбачуваним результатом, хоча в цілому, непогано працюють для складних категорій і мультимовних вхідних текстів.

Гібрідні підходи. Гібрідні методи класифікації текстів [1] поєднують два або більше різні типи методів, моделей/представлень або наборів ознак, для досягнення кращої якості, швидкості або продуктивності. Наприклад, це може бути комбінація в одній моделі частотного методу побудови вектора лексичних ознак TF-IDF, що дозволяє ефективно виявляти ключові слова в тексті, а також - нейромережевих технік embeddings [13], які гарно розуміють контекст та виявляють синонімію в тексті. Інший приклад - поєднання системи на основі правил і машинного навчання. Тоді частина класів формується за допомогою правил, а решта - методами машинного навчання. Існують також інші варіанти побудови гібрідних моделей, що пропонують комбінацію різноманітних методів в одній моделі. Таким чином гібрідні моделі є потужним інструментом класифікації, який може бути оптимізований та налаштований до вимог конкретної задачі чи прикладної системи, дозволяючи досягти високих показників точності та ефективності і не лише для вирішення завдань класифікації. Однак, з іншого боку, наслідком поєднання різних методів може бути суттєве підвищення складності розробки, підтримки та масштабування самої системи. Складність гібрідних систем досі залишається їхнім недоліком, який не можна недооцінювати.

Опис процесу категорізації повідомлень

З огляду на поставлену задачу, перш за все, задачу категорізації заявок на послуги, найбільш прийнятним видається комбінація методу витягнення властивостей з одним із класичних методів машинного навчання (навчання з вчителем або кероване машинне навчання), як, наприклад, згадані вище Naive Bayes, Logistic Regression або Linear SVM. Це швидкі та прості в розгортанні надійні базові моделі, які досягають високої продуктивності, якщо класи добре розділені словником термінів, що розглядаються як ключові слова. Доцільність використання саме класичних моделей для вирішення поставленої задачі підтверджується фактом їхнього застосування у багатьох реальних промислових задачах, зокрема, для маршрутизації заявок клієнтів для служби підтримки.

Методи керованого машинного навчання вивчають співставлення вхідного необробленого тексту з мітками (що відомі також як цільові змінні). Тобто алгоритм контрольованої класифікації навчається на певному наборі вхідних необроблених текстів для прогнозування категорії.

Серед методів витягнення ознак із тексту (feature extractor) найпоширенішими є TF-IDF та Bag of Words (відомий також як CountVectorizer). Головною метою цих методів є перетворення текстових даних (рядків) на вектор числових ознак, що може бути поданий на вхід моделі машинного навчання. Зазвичай це і є першим кроком у поетапному вирішенні задачі класифікації засобами класичного ML.

Обидва названі методі є простими способами представлення текстових даних як числових ознак на основі частотного аналізу тексту. Модель Bag of Words (BoW) [2], часто перекладається як «мішок слів»,

передбачає створення словника відомих слів у корпусі, а потім створення вектора для кожного документа, який містить підрахунок частоти появи кожного слова.

TF-IDF [18] є ще одним способом представлення тексту як числових ознак. Модель TF-IDF відрізняється від BoW тим, що враховує частоту слів у документі, а також обернену частоту документа. Тобто, TF-IDF має вищу ймовірність знаходження ключових слів, ніж BoW. Розглянемо даний метод трохи детальніше.

TF (Term Frequency) – це частота слова в тексті, тоді як IDF (Inverse Document Frequency) навпаки визначає, наскільки рідко дане слово зустрічається у колекції текстових повідомлень. Здебільшого TF визначається як кількість входжень терміна t у текстове повідомлення d - TF(t,d), або ця оцінка може бути нормалізованою і враховувати загальну кількість слів у повідомленні:

$$\mathrm{TF(t,d)} = \frac{tf(t,d)}{|d|}$$

Інверсна оцінка («рідкість» терміна у колекції всіх поданих на аналіз текстових повідомлень) обчислюється відповідно:

$$\mathrm{IDF}(t,d) = log\left(\frac{N}{df(t)}\right),$$

де N – кількість повідомлень у колекції, $df(t)$ – кількість повідомлень, що містять слово t.

Тоді результуюча оцінка обчислюється як добуток оцінок, наведених вище:

$$\mathrm{TFIDF}(t,d) = TF(t,d) * IDF(t)$$

Фактично найвищу вагу отримують слова, що часто зустрічаються в конкретному повідомленні, але не дуже часто в решті повідомлень. Тобто вони є характерними саме для цієї заявки і тому мають більший вплив.

Як і будь-яка інша задача керованого машинного навчання, задача класифікації тексту включає два етапи: навчання та прогнозування. Перший етап полягає у навчанні моделі на певному тренувальному наборі релевантних розмічених текстових даних. Після цього навчена модель може бути використана для прогнозування міток (категорій) для нових та невидимих даних.

Також слід зазначити, що значну роль у підвищенні ефективності обробки текстів природної мови відіграє попередня

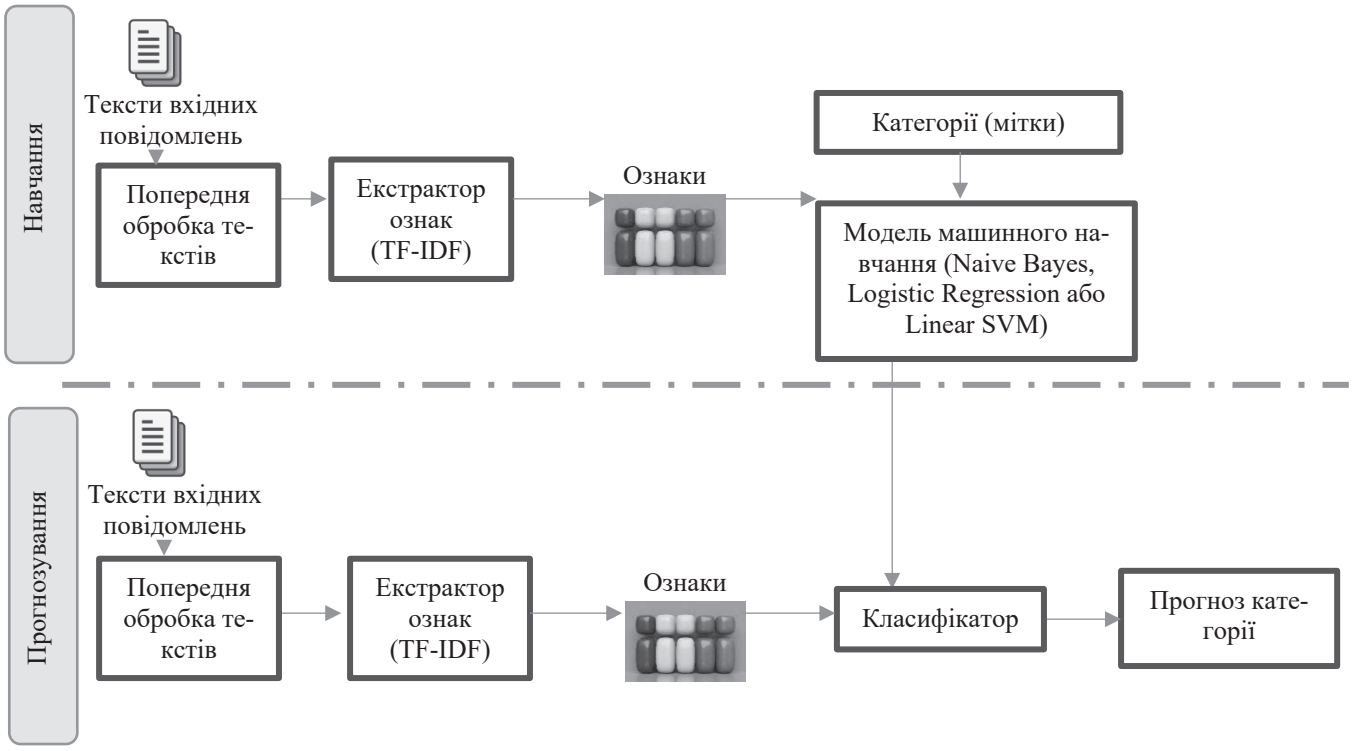

Рис. 3. Категорізація повідомлень на основі ML підходу

підготовка текстових даних для аналізу. Текстові дані неструктуровані, часто містять багато шуму. Це можуть бути орфографічні помилки, граматичні помилки, нестандартне форматування тощо. Попередня підготовка дозволяє очистити цей шум та полегшити подальший аналіз/обробку тексту. Набір кроків попередньої обробки тексту може відрізнятися, але зазвичай він включає такі завдання, як: токенізація, видалення стоп-слів, стеммінг та лематизація. Ці кроки допомагають зменшити розмір текстових даних, підвищити точність завдань обробки природної мови, зокрема, класифікації тексту.

Загальна схема вирішення задачі категорізації повідомлень наведена на рис.3.

Висновки

Метою даного дослідження є підвищення ефективності надання косметологічних послуг за рахунок автоматичної класифікації та маршрутизації заявок на послуги, а також повідомлень клієнтів. Аналіз найбільш використовуваних на сьогодні підходів до класифікації текстів, з урахуванням саме вимог поставленої задачі, дозволив дійти висновку про доцільність застосування для досягнення поставленої мети саме моделей класичного машинного навчання з попередньою обробкою текстів та їх представленням у вигляді числових векторів TF-IDF.

Напрямками подальших досліджень є:

- деталізація процесу класифікації з вибором конкретної ML моделі;
- формування словника термінів та вибір засобів його формалізації для ефективного визначення результуючих категорій.

Література

1. *Дубовик А. В., Волинець Є. А.* Автоматична класифікація текстів. Наукові записки НаУКМА. Комп'ютерні науки. Том 8 (2025). С. 102-107. DOI: 10.18523/2617-3808.2025.8.102-107. – https://nrpcomp.ukma.edu.ua/article/view/344850/332233
2. *Moez All.* Understanding Text Classification in Python. 2022. – https://www.datacamp.com/tutorial/text-classification-python
3. SpamAssassin configuration file. https://spamassassin.apache.org/full/3.4.x/doc/Mail_SpamAssassin_Conf.html
4. *Hutto, C. J., & Gilbert, E.* VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Proceedings of ICWSM. 2014. – https://ojs.aaai.org/index.php/ICWSM/article/view/14550/14399
5. *Esuli, A., & Sebastiani, F.* SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining. Proceedings of LREC 2006. – http://www.lrec-conf.org/proceedings/lrec2006/pdf/384_pdf.pdf
6. Rasa Open Source Documentation. 2025 – https://legacy-docs-oss.rasa.com/docs/rasa/rules/
7. GoogleCloud Guide. – https://docs.cloud.google.com/dialogflow/es/docs/intents-overview
8. *Shuo Xu, Yan Li, Zheng Wang*. Bayesian Multinomial Naïve Bayes Classifier to Text Classification. Institute of Scientific and Technical Information of China. № 15. 2015. – https://www.researchgate.net/publication/317173563_Bayesian_Multinomial_Naive_Bayes_Classifier_to_Text_Classification/link/59fa7e88aca272026f6f98e4/download?_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6InB1YmxpY2F0aW9uIiwicGFnZSI6InB1YmxpY2F0aW9uIn19
9. *C. Cortes and V. Vapnik.* Support-vector networks. Machine Learning. November 1995. – P. 273–297
10. *Thorsten Joachims.* Text Categorization with Support Vector Machines: Learning with Many Relevant Features. Support Vector Learning. Conference paper. 2005. – P. 137-142. https://www.cs.cornell.edu/people/tj/publications/joachims_98a.pdf
11. *J. Kivinen, M. Warmuth, and P. Auer.* The perceptron algorithm vs. winnow: Linear vs. logarithmic mistake bounds when few input variables are relevant. In Conference on Computational Learning Theory, 1995.
12. *Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., & Lin, C.-J.* LIBLINEAR: A Library for Large Linear Classification. 2008. – https://www.csie.ntu.edu.tw/~cjlin/papers/liblinear.pdf
13. *Kim, Y.* Convolutional Neural Networks for Sentence Classification. 2014. – https://arxiv.org/pdf/1408.5882

14. *Hochreiter, S., & Schmidhuber, J.* Long Short-Term Memory. Neural Computation. 1997. – https://www.bioinf.jku.at/publications/older/2604.pdf
15. *Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K.* BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2019. – https://arxiv.org/pdf/1810.04805
16. *Liu, Y., et al.* RoBERTa: A Robustly Optimized BERT Pretraining Approach. 2019. – https://arxiv.org/pdf/1907.11692
17. *Conneau, A., et al.* Unsupervised Cross-lingual Representation Learning at Scale. 2020. – https://arxiv.org/pdf/1911.02116
18. *Salton, G., & Buckley, C.* Term-weighting approaches in automatic text retrieval. 1988. – https://dl.acm.org/doi/pdf/10.1145/53990.54006

References

1. *Dubrovik A. V., Volynec J. A.* Automatic text classification. Proceedings of NaUKMA. Computer Science. Volume 8 (2025). P. 102-107. DOI: 10.18523/2617-3808.2025.8.102-107. – https://nrpcomp.ukma.edu.ua/article/view/344850/332233
2. *Moez All.* Understanding Text Classification in Python. 2022. – https://www.datacamp.com/tutorial/text-classification-python
3. SpamAssassin configuration file. https://spamassassin.apache.org/full/3.4.x/doc/Mail_SpamAssassin_Conf.html
4. *Hutto, C. J., & Gilbert, E.* VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Proceedings of ICWSM. 2014. – https://ojs.aaai.org/index.php/ICWSM/article/view/14550/14399
5. *Esuli, A., & Sebastiani, F.* SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining. Proceedings of LREC 2006. – http://www.lrec-conf.org/proceedings/lrec2006/pdf/384_pdf.pdf
6. Rasa Open Source Documentation. 2025 – https://legacy-docs-oss.rasa.com/docs/rasa/rules/
7. GoogleCloud Guide. – https://docs.cloud.google.com/dialogflow/es/docs/intents-overview
8. *Shuo Xu, Yan Li, Zheng Wang*. Bayesian Multinomial Naïve Bayes Classifier to Text Classification. Institute of Scientific and Technical Information of China. № 15. 2015. – https://www.researchgate.net/publication/317173563_Bayesian_Multinomial_Naive_Bayes_Classifier_to_Text_Classification/link/59fa7e88aca272026f6f98e4/download?_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6InB1YmxpY2F0aW9uIiwicGFnZSI6InB1YmxpY2F0aW9uIn19
9. *C. Cortes and V. Vapnik.* Support-vector networks. Machine Learning. November 1995. – P. 273–297
10. *Thorsten Joachims.* Text Categorization with Support Vector Machines: Learning with Many Relevant Features. Support Vector Learning. Conference paper. 2005. – P. 137-142. https://www.cs.cornell.edu/people/tj/publications/joachims_98a.pdf
11. *J. Kivinen, M. Warmuth, and P. Auer.* The perceptron algorithm vs. winnow: Linear vs. logarithmic mistake bounds when few input variables are relevant. In Conference on Computational Learning Theory, 1995.
12. *Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., & Lin, C.-J.* LIBLINEAR: A Library for Large Linear Classification. 2008. – https://www.csie.ntu.edu.tw/~cjlin/papers/liblinear.pdf
13. *Kim, Y.* Convolutional Neural Networks for Sentence Classification. 2014. – https://arxiv.org/pdf/1408.5882
14. *Hochreiter, S., & Schmidhuber, J.* Long Short-Term Memory. Neural Computation. 1997. – https://www.bioinf.jku.at/publications/older/2604.pdf
15. *Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K.* BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2019. – https://arxiv.org/pdf/1810.04805
16. *Liu, Y., et al.* RoBERTa: A Robustly Optimized BERT Pretraining Approach. 2019. – https://arxiv.org/pdf/1907.11692
17. *Conneau, A., et al.* Unsupervised Cross-lingual Representation Learning at Scale. 2020. – https://arxiv.org/pdf/1911.02116
18. Salton, G., & Buckley, C. Term-weighting approaches in automatic text retrieval. (1988). https://dl.acm.org/doi/pdf/10.1145/53990.54006

Дата першого надходження до видання: 24.02.2026
Внутрішня рецензія отримана: 03.03.2026
Зовнішня рецензія отримана: 05.03.2026
Дата прийняття статті до друку: 19.03.2026
Дата публікації: 16.04.2026

Про авторів:

[1]Захарова Ольга Вікторівна,
кандидат технічних наук,
старший науковий співробітник
[1] Zakharova Olga,
Ph.D (technical sciences), senior scientist
http://orcid.org/0000-0002-9579-2973.

[2] Спекторовська Лада Олександрівна,
Студент бакалаврата
[2] Spektorovska Lada,
Bachelor student
http://orcid.org/0009-0007-7173-0149

Місце роботи авторів:

[1] Інститут програмних систем
НАН України,
проспект Академіка Глушкова, 40
[1] Institute of Software Systems.
National Academy of Sciences of Ukraine
Тел.: +380(68)5947560
E-mail: ozakharova68@gmail.

[2] Національний технічний університет
«Київський політехнічний інститут
імені Сікорського»
[2] National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”
Тел.: +380(68)3051221
E-mail: spektorovskalada@gmail.com.