УДК 004.855.5
DOI: 10.20535/SRIT.2308-8893.2020.3.06

МЕТОДИ МАШИННОГО НАВЧАННЯ В СЕНТИМЕНТ-АНАЛІЗІ ТЕКСТОВОЇ ІНФОРМАЦІЇ НА ПРИКЛАДІ НАСТРОЇВ КОРИСТУВАЧІВ СТОСОВНО КАНДИДАТІВ У ПРЕЗИДЕНТИ УКРАЇНИ 2019

А.-М.П. РУДЗЕВИЧ

Анотація. Описано основні методи машинного навчання для аналізу тональності тексту і виконано порівняльний аналіз їх ефективності. Розглянуто етапи попереднього оброблення тексту, такі як стемінг, видалення стоп-слів, алгоритми переведення тексту у векторну форму: мішок слів (Bag-of-Words), TF-IDF векторайзер та Word2Vec. Дослідження полягало у визначенні тональності тексту коментарів під публікаціями кандидатів у Президенти України (В. Зеленського та П. Порошенка) у період передвиборчих перегонів 2019 р. Для визначення тональності тексту використано три алгоритми: наївний байєсівський класифікатор, метод опорних векторів та згорткову нейронну мережу. Для кожного кандидата побудовано окремі моделі та виконано порівняння якості класифікації (за метрикою F1). Найкращою моделлю для обох вибірок даних виявилась згорткова нейронна мережа.

Ключові слова: машинне навчання, сентимент-аналіз, аналіз тональності тексту, інтелектуальний аналіз тексту.

ВСТУП

Із розвитком інформаційних технологій та стрімкого нагромадження великих масивів даних поширення набула така галузь комп'ютерної лінгвістики, як сентимент-аналіз. Стало можливим автоматично витягати з тексту виражену автором думку, а також оцінювати текст як позитивний, негативний, а за необхідності — виокремлювати конкретні емоції (радість, гнів, сум тощо). Для виокремлення емоційної оцінки автора застосовують підходи з використанням тональних словників і правил або методи машинного навчання.

Сентимент-аналіз (аналіз тональності тексту) — це розділ глибинного аналізу даних (data mining) і галузь комп'ютерної лінгвістики, що займається вилученням думок та емоцій з текстових документів.

Хоча лінгвістика та оброблення природних мов (NLP) мають давню історію, до 2000-х років майже не було досліджень, що стосуються сентимент-аналізу. Але відтоді цю галузь учені почали дуже активно вивчати [11, 12].

Термін «sentiment analysis» уперше був згаданий у праці [1], а вираз «opinion mining» (аналіз думок) — у праці [2]. Вагомий внесок у розвиток сентимент-аналізу зроблено у працях [8, 9].

Усі завдання з оброблення природних мов є складними і неоднозначними. Загалом завдання визначення емоційної оцінки тексту є суб'єктивним,

оскільки різні люди по-різному оцінюють одні й ті самі події, а відповідно один і той самий текст. Текст може містити орфографічні помилки, скорочення, абревіатури, сарказм, емоджі. Однакові слова, вжиті в різному контексті, можуть мати діаметрально протилежне емоційне навантаження. Усе це перешкоджає створенню єдиної моделі, яка правильно класифікуватиме тональність тексту незалежно від тематики.

Сентимент-аналіз набув широкого використання для маркетингових цілей, зокрема для визначення думки клієнта про певний товар або послугу, та кращого орієнтування свого повідомлення на цільову аудиторію. Також набуло популярності аналізування твітів, блогів, текстів новин, оглядів, коментарів для визначення ставлення автора до суб'єкта його висловлення. Для цього застосовують різні методики, включаючи алгоритми оброблення природних мов (NLP), статистику та методи машинного навчання.

У роботі застосуємо сентимент-аналіз для визначення настроїв користувачів стосовно кандидатів у Президенти України 2019. Аналізуватимемо коментарі користувачів у соціальній мережі інстаграм протягом усього часу передвиборчих перегонів на предмет позитивного або негативного ставлення до кандидата і зможемо оцінити як змінювалися настрої в суспільстві. Оскільки українці залишають коментарі як українською, так і російською мовами, будемо аналізувати ці дві мови.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Наївний байєсівський класифікатор є ймовірнісним алгоритмом машинного навчання, заснований на теоремі Байєса, який широко використовується для задач класифікації.

Для задачі визначення тональності прогнозуємо ймовірність того, що документ d належить до класу c. Тут документ є вектором: $a = \{w_1, w_2, \ldots, w_n\}$, де w_i — вага i-го терміна; n — розмір словника. Тому згідно з теоремою Байеса маємо формулу

$$P(c \mid d) = \frac{P(c)P(d \mid c)}{P(d)}.$$

За цією формулою обчислюється умовна ймовірність для всіх класів. Якщо умовна ймовірність належності документа d до класу c максимальна, то C є найбільш імовірним класом, до якого належить документ d:

$$C = \arg\max_c P(w_1, w_2, \ldots, w_n \mid c)P(c).$$

Знаменник може бути випущений, оскільки для одного і того ж документа d імовірність $P(d)$ буде однаковою, а отже, її можна не враховувати.

Наївний байєсівський класифікатор спирається на припущення, що всі ознаки $x_1, x_2, \ldots, x_n$ документа d не залежать одна від одної. Припускається, що позиція слів у реченні не має значення. Тому умовну ймовірність для ознак $x_1, x_2, \ldots, x_n$, можна подати як

$$P(w_1 \mid c)(w_2 \mid c) \times \ldots \times (w_n \mid c) = \Pi_i P(w_i \mid c).$$

Таким чином, для знаходження найбільш імовірного класу для документа $d = \{w_1, w_2, \ldots, w_n\}$ за допомогою наївного байєсівського класифікато-

ра необхідно визначити умовні ймовірності належності документа d для кожного з поданих класів окремо і вибрати клас, який має максимальну ймовірність:

$$P(c)\prod[P(w_i|c_j)].$$

Далі оцінимо ймовірність класу $P(c)$. Вона є відношенням кількості докуметів класу c у навчальній вибірці до загальної кількості документів: $P(c)=\frac{D_c}{D}$, де D_c — кількість документів класу c; D — загальна кількість документів у вибірці.

Щоб оцінити умовні ймовірності для ознак $\hat{P}(w_i\,|\,c_j)$, використовуватимемо формулу

$$\hat{P}(w_i\,|\,c)=\frac{\text{count}(w_i,c)+1}{\sum_{w\in V}(\text{count}(w,c)+1)},$$

де $\hat{P}(w_i\,|\,c_j)$ — відношення кількості слів w_i у класі c_j до загальної кількості слів у цьому класі; V — кількість слів у словнику навчальної вибірки [14].

Метод опорних векторів шукає гіперплощину, яка найкраще розділить вибірку на два класи. Допускається багатокласова класифікація стратегіями one-vs-all і one-vs-one.

Дано вибірку елементів $x_i\in\mathbb{R}_n$ і зіставлені їм класи $y_i\in\{-1,1\}$. Об'єкти вибірки подано точками. Опорні вектори — це точки даних, розташовані якомога ближче до гіперплощини, у разі їх видалення зміниться положення гіперплощини. Їх вважають критичними елементами набору даних. У простій задачі бінарної класифікації з вибіркою, що лінійно розділяється, гіперплощину можна подати у вигляді лінії, що розділяє вибірку на два класи. Чим далі дані від гіперплощини, тим коректніше вони класифіковані. Кращою гіперплощиною вважається та, відстань $1/\|w\|$ від якої до кожного класу є максимальною, де w — нормальний вектор до роздільної гіперплощини, яку можна записати як множину точок x, що задовольняють рівняння $wx-b=0$, де b — допоміжний параметр.

Якщо навчальна вибірка лінійно подільна, можна вибрати дві паралельні гіперплощини так, щоб вони розділили цю множину на два класи. Ділянка між ними називається зазором, маржею. Ці площини описуються рівняннями: $wx-b=1$; $wx-b=-1$.

Мінімізуючи відстань $\|w\|$ і одночасно виключаючи потрапляння даних у зазор, отримуємо задачу мінімізації $\|w\|^2\to\min$;

$$y_i(wx_i-b)\geq 1,\ \text{для}\ \ 1\leq i\leq n.$$

Таку задачу вважають еквівалентною пошуку сідлової точки функції Лангранжа; її зводять до задачі квадратичного програмування, де наявні лише двоїсті змінні λ_i.

Розв'язавши цю задачу, можна виразити w і b формулами:

$$w=\sum_{i=1}^{n}\lambda_i c_i x_i;\ \ b=wx_i-c_i;\ \ \lambda_i>0.$$

Кінцевий класифікатор записується як

$$a(x)\ \text{sign}\left(\sum_{i=1}^{n}\lambda_i c_i x_i\ x - b\right). \qquad (1)$$

Якщо вибірка лінійно нероздільна, вектори відображають у простір більшої розмірності. При цьому у формулі (1) скалярний добуток змінюється на одну із функцій нелінійного ядра $K(x_i, x)$ і будується найкраща роздільна гіперплощина.

Згорткова нейронна мережа (ЗНМ). Архітектура згорткової нейронної мережі для класифікації тексту бере за основу звичайну ЗНМ, але дещо спрощену. На вхід подається матриця, кількість її рядків дорівнює кількості слів n у реченні (або документі), кількість стовпців — розмірності k векторного подання слів. Для отримання нової ознаки c_i виконується операція згортки. Згортка полягає в застосуванні фільтра з вагами w на вікні з h слів. Ознака c_i генерується з вікна слів $x_{i:i+h-1}$ за формулою

$$c_i = f(w x_{i:i+h-1} + b),$$

де $x_i \in \mathbb{R}$ — нейрон зміщення; f — нелінійна функція; w — вектор ваг; $x_{i:i+h-1}$ — ковзне вікно.

Фільтр буде застосований до всіх можливих вікон слів у реченні $\{x_i : h, \ldots, x_{n-h+1} : n\}$ для отримання карти ознак: $c(w) = [c_1, c_2, \ldots, c_{n-h+1}]$.

Потім застосовується фільтр Max Pooling (максимізаційне агрегування), тобто шукається максимум у всій послідовності. Його ідея полягає у виокремленні найважливішої ознаки з найбільшим значенням за кожною картою ознак: $\hat{c} = \max(c(w))$.

Отримані таким чином значення передаються в повнозв'язний шар із функцією активації softmax; на виході маємо розподіл імовірності за класами:

$$P(y = j \mid x) = \frac{e^{x^T w_j + b_j}}{\sum_{k=1}^{K} e^{x^T w_j + b_j}}.$$

Для запобігання перенавчанню на цьому шарі використовується метод виключення нейронів (дропаут) з імовірностю p і $l2$-регуляризація.

Зазвичай навчання мережі відбувається з використанням стохастичного градієнтного спуску. Використання дропауту вилучає з нейронної мережі деяку кількість нейронів (на етапі навчання) для запобігання коадаптації нейронів і в результаті отримання кращої узагальнювальної здатності мережі. Дропаут також прискорює процес навчання. Вихід після використання дропауту можна подати у вигляді $y = w(zr) + b$, де $z = [\hat{c}_1, \ldots, \hat{c}_m]$, r — вектор, що містить 0 і 1.

Як гіперпараметри мережі виділяють розмір фільтра, імовірність дропауту p, $l2$-регуляризацію і розмір батча. $l2$-регуляризація штрафує ваги мережі, зменшуючи їх значення, і використовується для запобігання її перенавчанню. Батч використовується для пришвидшення навчання, являючи собою «пакет» випадково обраних ознак у методі стохастичного градієнтного спуску.

МЕТРИКИ ОЦІНЮВАННЯ ЯКОСТІ АЛГОРИТМІВ

Емпіричні дані показують, що показник точності дуже залежить від збалансованості даних. У випадку, коли дані незбалансовані, доцільно перевірити, наскільки ефективно класифікатор класифікує лише частину даних — позитивні або негативні класи даних. Прикладами таких метрик є чутливість (precision) та повнота (recall).

Чутливість доцільно використовувати, коли помилково позитивна класифікація небажана. Вона розраховується за такою формулою: $\text{Precision} = TP - (TP + FP)$, де TP — правильно визначений позитивний клас; FP — хибний позитивний клас.

Метрику повноти використовують, коли треба уникнути помилково негативної класифікації. Її обчислюють за формулою $\text{Recall} = TP / (TP + FN)$, де TP — правильно визначений позитивний клас; FN — хибно визначений негативний клас.

Також є показник, який є гармонічним середнім двох попередніх оцінок — F-міра: $F_1 = \frac{2*\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$. Це загальна міра точності моделі, яка поєднує в собі чутливість та повноту. Тобто показник F_1 означає малу кількість хибних позитивних та хибних негативних класифікацій.

ДОСЛІДЖЕННЯ

Дослідження полягає в аналізі емоційного навантаження тексту коментарів із соціальної мережі Інстаграм у період передвиборчих перегонів на пост Президента України в 2019.

Для проведення дослідження зібрано коментарі під публікаціями кандидатів у Президенти України 2019 р. В.О. Зеленського та П.О. Порошенка в період з початку передвиборчої кампанії до другого туру президентських виборів. Загалом зібрано близько 70 тис. записів, а навчальна вибірка містить близько 20 тис. записів (по 10 тис. для кожного кандидата).

Для кожного кандидата навчимо окрему модель, застосовуючи кросвалідацію на п’яти фолдах, а як метрику якості використаємо F1-міру. Потім застосуємо найкращу з навчених моделей для класифікації коментарів до публікацій у період з початку передвиборчих перегонів до другого туру виборів (близько 50 тис. записів) і з отриманих результатів дослідимо зміну громадської думки залежно від тодішніх подій.

Для визначення тональності тексту будуть використані три алгоритми: наївний байєсівський класифікатор, метод опорних векторів та згорткова нейронна мережа.

Для того щоб дані були придатними для алгоритмів машинного навчання, їх необхідно перетворити у вектори. Для векторизації тексту застосуємо такі алгоритми: Bag-of-Words і TF-IDF векторайзер [13] — для перших двох алгоритмів та Word2Vec — для ЗНМ [7]; порівняємо їх ефективність.

Для практичної реалізації поставленого завдання використовуватимемо мову програмування Python, оскільки ця мова найбільше підходить для машинного навчання. Використаємо бібліотеки sklearn для векторизації тексту

та побудови моделей (НБК і SVM) та keras для побудови згорткової нейронної мережі.

Будемо вирішувати завдання бінарної класифікації, оскільки зроблено припущення, що люди, які залишають коментарі, не є політично нейтральними, тому нейтральних коментарів або зовсім не буде, або їх буде зовсім мало, і ними можна знехтувати.

Класи є незбалансованими. Для В. Зеленського позитивний клас становить 83%, а для П. Порошенка — 38%. Про це варто пам'ятати під час навчання моделі.

Для зібраних даних розставлено мітки класів: 0 — негативний сентимент, 1 — позитивний. Навчальна вибірка містить такі поля: автор, дата, коментар та сентимент.

Перш ніж почати використовувати текст коментарів, з нього потрібно вилучити непотрібну інформацію, а саме [10]:

- видалити згадки, оскільки вони не містять емоційного навантаження;
- видалити знак хештега, але не сам хештег, оскільки він може містити інформацію;
- перевести всі слова до нижнього регістра;
- видалити всі розділові знаки, включаючи знаки запитання та знаки оклику;
- видалити URL-адреси, оскільки вони не містять корисної інформації;
- конвертувати емоджі в одне слово;
- видалити цифри;
- видалити стоп-слова;
- застосувати стемінг, щоб зберегти основу слова без закінчення чи суфіксів.

Оскільки коментарі написані українською та російською мовами, то видалятимемо російські і українські стоп-слова. Для цього застосуємо два стемери: спочатку російський, потім український [4].

Якщо після такого очищення з'являться коментарі без жодного слова, їх буде видалено, оскільки вони не містять інформації про сентимент.

Перейдемо до навчання моделей. Щоб підібрати найкращі параметри, будемо використовувати перехресну перевірку (кросвалідацію) на п'яти фолдах. Шукатимемо такі параметри: кількість n-gram, максимальний поріг відсіву, коефіцієнт регуляризації.

Для ЗНМ задано такі параметри: функцію активації: ReLU, регуляризацію ($l2$): 3, дропаут: 0,4, розмір батча: 100. Будемо шукати: кількість шарів згортки, розмір ядра згортки та кількість фільтрів.

Зведемо результати навчання алгоритмів до таблиці:

Результати роботи алгоритмів, %

Алгоритми	**В. Зеленський**	**П. Порошенко**
Наївний байєсівський класифікатор (Bag-of-Words)	93,9	90
Наївний байєсівський класифікатор (TF-IDF)	93,2	92
SVM (Bag-of-Words)	93,5	90
SVM (TF-IDF)	93	94
Згорткова нейронна мережа (Word2Vec)	95,6	95,5

Отже, всі алгоритми досить точно класифікують дані за правильно підібраних параметрів.

Порівняємо найкращі моделі для кожного кандидата за допомогою коробкового графіка (boxplot). Як видно з рис. 1, 2 найкращою моделлю є ЗНМ.

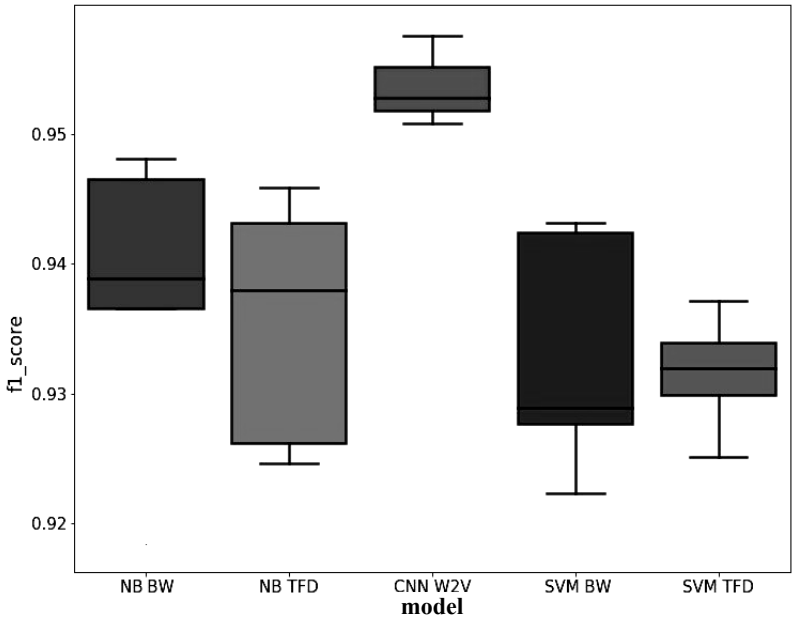

Рис. 1. Boxplot для найкращих моделей (В. Зеленський)

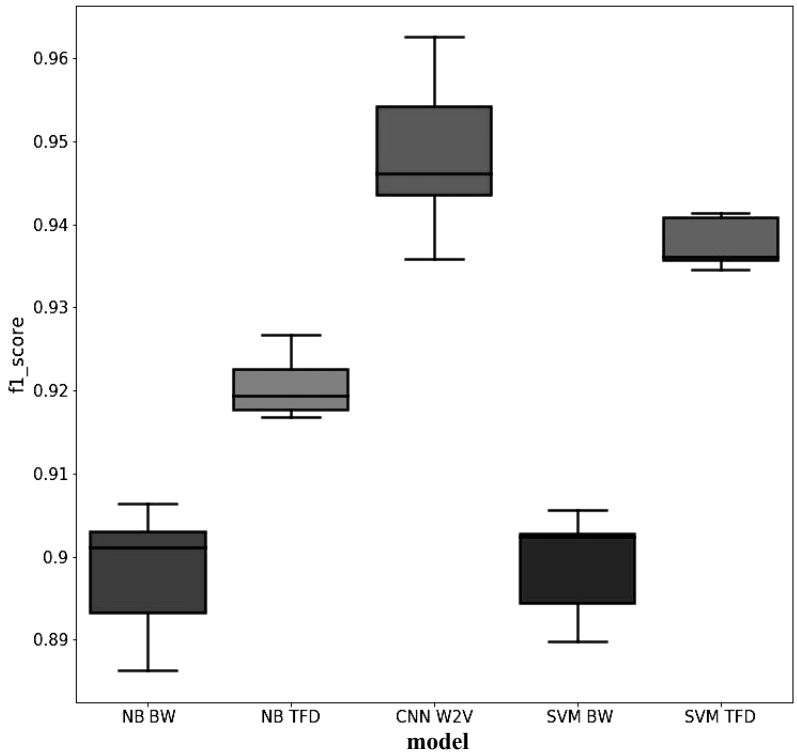

Рис. 2. Boxplot для найкращих моделей (П. Порошенко)

NB BW — наївний байєсівський класифікатор з векторайзером Bag-of-Words.

NB TFD — наївний байєсівський класифікатор з векторайзером TF-IDF.

CNN W2V — згорткова нейронна мережа з векторайзером Word2Vec.

SVM BW — метод опорних векторів з векторайзером Bag-of-Words.

SVM TFD — метод опорних векторів з векторайзером TF-IDF.

Усі моделі мають точність понад 90% (за метрикою F1). Найкращий результат для обох вибірок даних показала ЗНМ з одним шаром згортки — точність 95,5%.

Визначивши найкращу модель, проаналізуємо за її допомогою зміну прихильності громадськості до кандидата за час передвиборчої кампанії. Для цього використаємо нерозмічені коментарі під публікаціями кандидатів у період з початку президентських перегонів до другого туру виборів (03.01.2019–21.04.2019) — близько 50 тис. коментарів. Далі використаємо раніше навчену ЗНМ для класифікації коментарів. Результати класифікації (відсоток позитивного класу) зобразимо на графіку (рис. 3). На графіку на осі X позначено дату публікації поста у соціальній мережі (обиралось таким чином, щоб обидва кандидати мали публікацію в зазначений день), а на осі Y — відсоток позитивних коментарів.

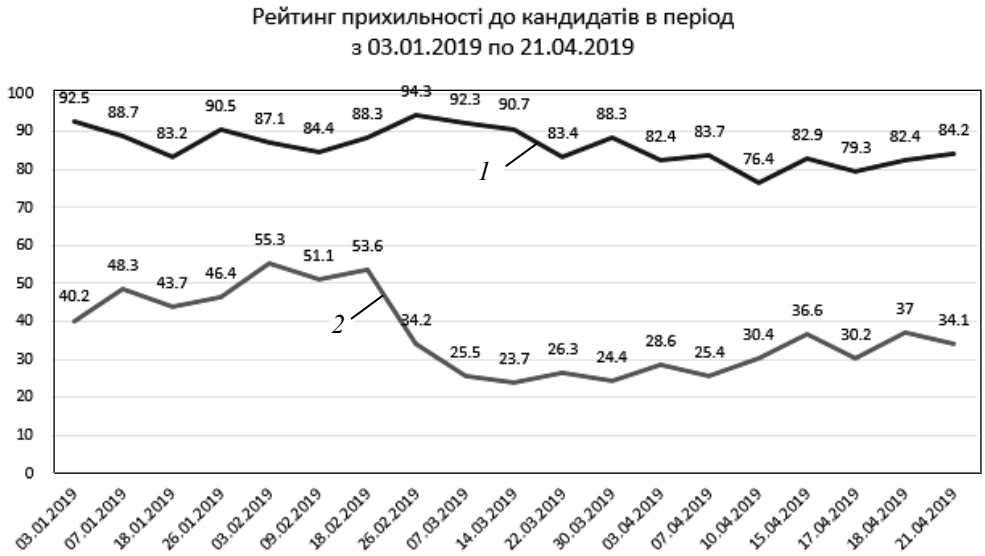

Рис. 3. Графік прихильності до кандидатів у період з 03.01.2019 по 21.04.2019: *1* — В. Зеленський, *2* — П. Порошенко

Графік дозволяє відслідковувати та аналізувати реакцію громадськості на події в автоматичному режимі. Наприклад, на рис. 3 на графіку В. Зеленського у точці за 18.01.2019 спостерігається різкий спад, який імовірно зумовлений оприлюдненням журналістського розслідування, у якому йдеться про те, що В. Зеленський через кіпрську фірму володіє кінокомпаніями в Росії. На тому ж рисунку на графіку П. Порошенка спосте-

рігається різкий спад після 26.02.2019. Ця дата припадає на вихід журналістського розслідування про корупцію в «Укроборонпромі».

ВИСНОВКИ

Досліджено методи машинного навчання для аналізу тональності тексту. Дослідження полягало у визначенні тональності тексту коментарів під публікаціями кандидатів у Президенти України (В. Зеленського та П. Порошенка) у період передвиборчих перегонів 2019 р.

Для визначення тональності тексту використано три алгоритми: наївний байєсівський класифікатор, метод опорних векторів та згорткову нейронну мережу. Оптимальні параметри для моделей підбиралися шляхом кросвалідації.

Для переведення тексту у вектор було використано три векторайзери — Bag-of-Words і TF-IDF — для наївного байєсівського класифікатора та методу опорних векторів; Word2Vec — для ЗНМ. Для кожного кандидата побудовано окремі моделі і порівняно якість класифікації (за метрикою F1).

У результаті дослідження всі моделі показали досить високу точність класифікації. Найбільш точним алгоритмом для даних обох кандидатів виявлась ЗНМ з одним згортковим шаром (точність 95,5%).

Проведено дослідження зміни громадської думки в період з 03.01.2019 по 21.04.2019. Для цього зібрано близько 50 тис. коментарів з публікацій у соціальній мережі інстаграм кандидатів у Президенти України та класифіковано їх за допомогою раніше навченої ЗНМ. За результатами аналізу побудовано графік, який дає змогу оцінювати зміну громадської думки у реальному часі, відслідковувати реакцію аудиторії на події і відповідно швидко реагувати на них.

Загалом подано комплексний підхід до розв’язання задачі сентимент-аналізу, включаючи етапи попереднього оброблення тексту, використання різних векторайзерів для надання тексту векторного вигляду, навчання моделей та оцінювання їх якості.

ЛІТЕРАТУРА

1. T. Nasukawa and J.Yi, “Sentiment analysis: Capturing favorability using natural language processing”, *Proc. of the 2nd Int. Conf. on Knowledge capture (KCAP)*, pp. 7077, 2003.
2. K. Dave, St. Lawrence, D. Pennock, “Mining the peanut gallery: Opinion extraction and semantic classification of product reviews”, *Proc. of the Int. Conf. on World Wide Web (WWW)*, pp. 519528, 2003.
3. А. Барсегян, *Технологии анализа данных: Data Mining, Text Mining, Visual Mining, OLAP*, 2 изд., БХВ-Петербург, 2008, 384 р.
4. Vimala Balakrishnan, *Stemming and Lemmatization: A Comparison of Retrieval Performances*, 2014, 204 p.
5. Liu Bing, *Sentiment Analysis and Opinion Mining*, Morgan & Claypool Publishers, May 2012.
6. Vincent Ng, Claire Cardie, *Weakly Supervised Natural Language Learning Without Redundant Views*, 2003.

7. X. Fulin, D. Yihao, and T. Xiaosheng, “The Architecture of Word2vec and Its Applications”, *Journal of Nanjing*, 2015.
8. Bo Pang and Lillian Lee, *Opinion Mining and Sentiment Analysis*, 2008.
9. Bo Pang and Lillian Lee, *A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts*, 2004.
10. Janyce M. Wiebe, Rebecca F. Bruce, Thomas P. O’Hara, *Development and use of a gold-standard data set for subjectivity classifications*, 1999.
11. Jindal Liu, *Mining comparative sentences and relations*, 2006.
12. Liu Bing, *Sentiment analysis and subjectivity. Handbook of natural language processing*, 2nd ed, Boca Raton: CRC Press, 2010.
13. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, *Efficient estimation of word representations in vector space,* arXiv preprint arXiv:1301.3781. ICLR Workshop, pp. 1–12, 2013.
14. N. Sebe, MS. Lew, I. Cohen, and A. Garg, “Emotion recognition using a cauchy naive bayes classifier”, in *IEEE*, Quebec, 2002.
15. Y. Kim, “Convolutional neural networks for sentence classification”, in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, Association for Computational Linguistics, October 2014, pp. 1746—1751.
16. G. Katz, N. Ofek, and B. Shapira, “Context-based sentiment analysis”, *Knowledge-Based Systems. ConSent*, vol. 84, no. 1, pp. 162–178, 2015.

Надійшла 30.07.2020

INFORMATION ON THE ARTICLE

Anna-Mariia P. Rudzevych, Educational and Scientific Complex “Institute for Applied System Analysis” of the National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, Ukraine, e-mail: anna.rudzevich@gmail.com

MACHINE LEARNING IN SENTIMENT-ANALYSIS OF TEXT INFORMATION ON THE EXAMPLE OF USER ATTITUDES REGARDING CANDIDATES FOR UKRAINIAN PRESIDENTIAL ELECTIONS 2019 / A.-M. P. Rudzevych

Abstract. The main methods of machine learning for the sentiment analysis of the text are described and a comparative analysis of their effectiveness is performed. The stages of pre-processing of the text, such as stemming, deletion of stop words, algorithms for converting the text to vector form, such as bag-of-words (Bag-of-Words), TF-IDF vectorizer and Word2Vec, are considered. The goal of this study was to determine the sentiment of the comments under the publications of Ukrainian Presidential candidates (V. Zelensky and P. Poroshenko) during the 2019 election campaign.Three algorithms were used to determine the tonality of the text: the naive Bayes classifier, the support vector machine, and the convolutional neural network. Separate models were built for each candidate and a comparison of the classification quality was performed (according to metric F1). The most precise model for both data samples was a convolutional neural network.

Keywords: machine learning, sentiment analysis, text mining.

МЕТОДЫ МАШИННОГО ОБУЧЕНИЯ В СЕНТИМЕНТ-АНАЛИЗЕ ТЕКСТОВОЙ ИНФОРМАЦИИ НА ПРИМЕРЕ НАСТРОЕНИЙ ПОЛЬЗОВАТЕЛЕЙ ПО ОТНОШЕНИЮ К КАНДИДАТАМ В ПРЕЗИДЕНТЫ УКРАИНЫ 2019 / А.-М. П. Рудзевич

Аннотация. Описаны основные методы машинного обучения для анализа тональности текста и проведен сравнительный анализ их эффективности. Рассмотрены этапы предварительной обработки текста, такие как стемминг, удаление стоп-слов, алгоритмы перевода текста в векторную форму: мешок слов, TF-IDF векторизатор и Word2Vec. Исследование заключалось в определении тональности текста комментариев под публикациями кандидатов в Президенты Украины (В. Зеленского и П. Порошенка) в период предвыборной гонки

2019 г. Для определения тональности текста использованы три алгоритма: наивный байесовский классификатор, метод опорных векторов и сверточная нейронная сеть. Для каждого кандидата построены отдельные модели и проведено сравнение качества классификации (по метрике F1). Лучшей моделью для обеих выборок данных оказалась сверточная нейронная сеть.

Ключевые слова: машинное обучение, сентимент-анализ, анализ тональности текста, интеллектуальный анализ данных.

REFERENCES

1. T. Nasukawa and J.Yi, “Sentiment analysis: Capturing favorability using natural language processing”, *Proc. of the 2nd Int. Conf. on Knowledge capture (KCAP)*, pp. 7077, 2003.
2. K. Dave, St. Lawrence, D. Pennock, “Mining the peanut gallery: Opinion extraction and semantic classification of product reviews”, *Proc. of the Int. Conf. on World Wide Web (WWW)*, pp. 519528, 2003.
3. A.Barsegyan, Technologies of data analysis: Data Mining, Text Mining, Visual Mining, OLAP, 2nd ed. BHV-Petersburg, 2008, 384 p.
4. Vimala Balakrishnan, Stemming and Lemmatization: A Comparison of Retrieval Performances, 2014, 204 p.
5. Liu Bing, *Sentiment Analysis and Opinion Mining*, Morgan & Claypool Publishers, May 2012.
6. Vincent Ng, Claire Cardie, Weakly Supervised Natural Language Learning Without Redundant Views, 2003.
7. X. Fulin, D. Yihao, and T. Xiaosheng, “The Architecture of Word2vec and Its Applications”, *Journal of Nanjing*, 2015.
8. Bo Pang and Lillian Lee,Opinion Mining and Sentiment Analysis, 2008.
9. Bo Pang and Lillian Lee, A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts, 2004.
10. Janyce M. Wiebe, Rebecca F. Bruce, Thomas P. O’Hara, Development and use of a gold-standard data set for subjectivity classifications, 1999.
11. JindalLiu, Mining comparative sentences and relations, 2006.
12. Liu Bing, Sentiment analysis and subjectivity. Handbook of natural language processing, 2nd ed., Boca Raton: CRC Press, 2010.
13. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, *Efficient estimation of word representations in vector space,* arXiv preprint arXiv:1301.3781. ICLR Workshop, pp. 1–12, 2013.
14. N. Sebe, MS. Lew, I. Cohen, and A. Garg, “Emotion recognition using a cauchy naive bayes classifier”, in *IEEE*, Quebec, 2002.
15. Y. Kim, “Convolutional neural networks for sentence classification”, in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, Association for Computational Linguistics, October 2014, pp. 1746–1751.
16. G. Katz, N. Ofek, and B. Shapira, “Context-based sentiment analysis”, *Knowledge-Based Systems. ConSent*, vol. 84, no. 1, pp. 162–178, 2015.