УДК 004.83 https://doi.org/10.15407/pp2020.02-03.295

АВТОМАТИЗОВАНІ МЕТОДИ ОЦІНКИ КОГЕРЕНТНОСТІ УКРАЇНОМОВНИХ ТЕКСТІВ З ВИКОРИСТАННЯМ МЕТОДОЛОГІЇ МАШИННОГО НАВЧАННЯ

А.А. Крамов, С.Д. Погорілий

Проаналізовано основні методи оцінки когерентності текстів з використанням різних технологій машинного навчання. Детально описано принципи роботи методів з використанням рекурентної та згорткової нейронних мереж, розглянуто їх переваги та недоліки. Обґрунтовано доцільність використання методу графу семантичної схожості порівняно з іншими методами. Запропоновано використання інших підходів векторного представлення речень для розрахунку міри семантичної схожості елементів тексту. Проведено експериментальну перевірку проаналізованих методів на множині україномовних наукових статей, здійснено навчання моделей семантичного представлення слів та речень. Виконано навчання рекурентної та згорткової нейронних мереж з використанням методу раннього зупину. Обраховано точність вирішення задач розрізнення документів та вставки для проаналізованих методів, здійснено порівняльний аналіз отриманих результатів.

Ключові слова: когерентність тексту, рекурентна нейронна мережа, згорткова нейронна мережа, граф семантичної схожості, семантичне представлення речень, задача розрізнення документів, задача вставки.

Проанализированы основные методы оценки когерентности текстов с использованием различных технологий машинного обучения. Детально описаны принципы работы методов с использованием рекуррентной и сверточной нейронных сетей, рассмотрены их преимущества и недостатки. Обосновано целесообразность использования метода графа семантического сходства в сравнении с другими методами. Предложено использование других подходов векторного представления предложений для расчета меры семантического сходства элементов текста. Проведена экспериментальная проверка методов на множества украиноязычных научных статей, осуществлено обучение моделей семантического представления слов и предложений. Исполнено обучение рекуррентной и сверточной нейронных сетей с использованием метода ранней остановки. Посчитана точность решения задач различения документов и вставки для методов, осуществлен сравнительный анализ полученных результатов.

Ключевые слова: когерентность текста, рекуррентная нейронная сеть, сверточная нейронная сеть, граф семантического сходства, семантическое представление предложений, задача различия документов, задача вставки.

The main methods of coherence evaluation of texts with the usage of different machine learning techniques have been analyzed. The principles of methods with the usage of recurrent and convolutional neural networks have been described in details. The advantages of a semantic similarity graph method have been considered. Other approaches to perform the vector representation of sentences for the estimation of semantic similarity between the elements of a text have been suggested to use. The experimental examination of methods has been performed on the set of Ukrainian scientific articles. The training of recurrent and convolutional networks with the usage of early stopping has been performed. The accuracy of the solving of document discrimination and insertion tasks has been calculated. The comparative analysis of the results obtained has been performed.

Key words: coherence of a text, recurrent neural network, convolutional neural network, semantic similarity graph, semantic representation of sentences, document discrimination task, insertion task.

Вступ

У зв’язку з постійним збільшенням обсягу масиву гетерогенних даних в інформаційному просторі, з’являється необхідність здійснювати автоматизований аналіз даних з нефіксованою структурою: тексти, звукові записи, зображення, відео тощо. Варто звернути увагу на обробку текстової інформації, що, наприклад, здійснюється пошуковими системами для знаходження ресурсів, відповідних водночас текстовим запитам користувача, а також файлам формату аудіо та відео. Крім того, автоматизований аналіз тексту використовується в різнотипних лінгвістичних задачах (наприклад, машинний переклад), алгоритмах екстракції даних, задачах розпізнавання та синтезу мовлення. Зазначені задачі та інші проблеми, пов’язані з обробкою текстової інформації, необхідно розглядати в межах галузі *обробки природної мови* (natural language processing – NLP). Неоднорідність структури представлення текстів унеможливлює використання універсального загального підходу для вирішення задач NLP. У зв’язку зі зростанням потужності обчислювальних ресурсів, доцільним стає застосування різних моделей машинного навчання, попереднього навчених на відповідній множині текстової інформації. Такий підхід використовується для вирішення задач NLP, що відносяться до категорії AI-повних (AI – artificial intelligence), тобто, задач, що не можуть бути вирішені без втручання людського фактору. До задач такого типу відноситься *оцінка когерентності тексту*.

Під когерентністю тексту розуміють його тематичну цілісність, комунікативну здатність передавати основну ідею читачу [1]. Крім того, когерентність тексту передбачає структурну цілісність (когезію) тексту. Більш когерентний текст простіше сприймати за допомогою впорядкованої структури, елементів пресупозиції та логічного виведення знань. На рис. 1 показано спрощений приклад когерентного та некогерентного фрагментів тексту.

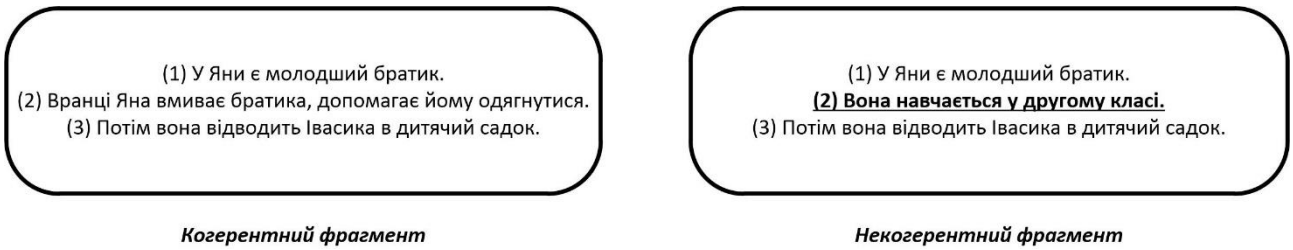

Рис. 1. Приклад когерентного та некогерентного фрагментів тексту

На відміну від когерентного фрагменту, в якому здійснено поступову передачу інформації у логічний спосіб, некогерентний фрагмент містить речення №2, що не відповідає його загальній тематиці. Незважаючи на наявність зв'язку між першим та другим реченням некогерентного фрагменту (кореферентний зв'язок «Яни»-«вона» [2]), речення № 2 тематично зміщене від загальної ідеї, що ускладнює сприйняття інформації.

Автоматизована оцінка когерентності тексту може використовуватися в різних галузях обробки текстової інформації:

- пошукові системи та компанії оптимізації веб-сайтів;
- написання презентаційних та рекламних текстів;
- створення навчального матеріалу;
- детектування симптомів психічного розладу [3].

Актуальність вирішення задачі когерентності тексту обумовлена наявністю сучасних робіт [3; 4; 5; 6], в яких пропонуються методи оцінки когерентності текстової інформації для вирішення різнотипних задач. Більшість цих робіт стосуються автоматизованого аналізу англомовних текстів. Незважаючи на наявність вітчизняних наукових робіт, пов'язаних з обробкою природної мови, дослідження оцінки когерентності україномовних текстів знаходиться на початковому етапі. Таким чином, актуальним є дослідження ефективності різноманітних методів оцінки когерентності текстів україномовного корпусу.

Метою роботи є здійснення порівняльного аналізу існуючих методів оцінки когерентності текстів та перевірка ефективності різнотипних методів (та їх модифікацій), основаних на методології машинного навчання, для корпусу україномовних текстів.

Методи оцінки когерентності текстів

Різноманітні методи оцінки когерентності текстів використовують засоби машинного навчання: нейронні мережі, метод опорних векторів, дерева рішень тощо. Використання різних моделей машинного навчання ускладнене нефіксованою структурою вхідних даних: тексти можуть складатися з довільної кількості слів та речень. Крім того, необхідно у певний спосіб здійснювати формалізацію елементів тексту для представлення їх семантичних, синтаксичних чи просторових властивостей. Розглянемо детально методи оцінки когерентності текстів, основаних на аналізі їх семантичної складової.

Метод розподіленого представлення речень з використанням рекурентної нейронної мережі. *Рекурентна нейронна мережа* [7] використовується в різних задач обробки природної мови. Такий вибір обумовлений можливістю здійснювати обробку даних з нефіксованої структурою: сигнали подаються на вхід нейронів рекурентного шару у рекурсивний спосіб. Рекурентні нейронні мережі з різною архітектурою (наприклад, довга короткочасна пам'ять, англ. *long short-term memory, LSTM*) використовують внутрішню пам'ять для обробки послідовності сигналів довільної довжини. Крім того, проходження сигналу через рекурентні шари мережі певним чином відтворює процес сприйняття інформації читачем: нейрони зі зворотнім зв'язком обробляють сигнали, отримані на попередньому етапі. Така обробка сигналу відповідає аналізу тексту читачем, адже нейрони головного мозку сприймають інформацію на основі попередньо прочитаного фрагменту [8]. Приклад проходження сигналу через рекурентний шар показано на рис. 2.

Спочатку здійснюється попередня обробка тексту: токенізація (розбиття на речення та окремі слова). Далі виконується формалізоване представлення слів, а саме їх семантичної складової, за допомогою попередньо навченої моделі. Наприклад, можуть використовуватися різні варіанти моделей Word2Vec [9] чи GloVe [10] – моделей семантичного векторного представлення слів. Таким чином, кожне речення представлене у наступний спосіб:

$$s = \left\{ \mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{n_s} \right\}, \tag{1}$$

де n_s – кількість слів речення; $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{n_s}$ – векторне представлення слів речення. Отримані вектори послідовно подаються на вхід рекурентного шару. Вихідне значення шару $\mathbf{h}_t$ в певний дискретний момент часу обраховується наступним чином:

$$\mathbf{h}_t = f(V_{\text{Recurrent}} \cdot \mathbf{h}_{t-1} + W_{\text{Recurrent}} \cdot \mathbf{w}^t + \mathbf{b}_{\text{Recurrent}}), \quad (2)$$

де $V_{\text{Recurrent}}$ і $W_{\text{Recurrent}}$ – матриці вільних параметрів; $\mathbf{b}_{\text{Recurrent}}$ – вектор зсуву; f – нелінійна функція активації. Після векторного представлення кожного речення тексту здійснюється об'єднання речень в окремі угрупування фіксованої довжини L (наприклад, $\langle s_1, s_2, s_3 \rangle, \langle s_2, s_3, s_4 \rangle$). Об'єднання виконується за допомогою конкатенації векторів угрупування в окремий вектор $\mathbf{h}_c$. Отриманий вектор подається на вхід класифікатора, що складається з декількох повнозв'язних шарів та вихідної функції softmax, яка розраховує міру когерентності угрупування. Когерентність всього документу D розраховується як добуток оцінок когерентності всіх угрупувань:

$$T_D = \prod_{c \in D} y_c . \quad (3)$$

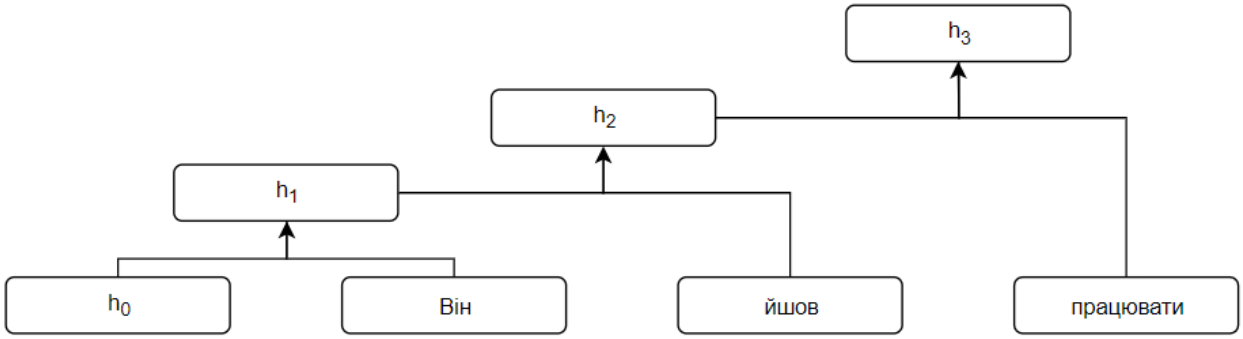

Рис. 2. Приклад проходження сигналу (елементів тексту) через рекурентну мережу

Метод оцінки когерентності тексту з використанням згорткової нейронної мережі. Згорткові нейронні мережі [7] використовуються в різних задачах обробки вхідних зображень чи фрагментів відео. Такий вибір обумовлений наступними факторами:

- можливість здійснювати обробку даних, що представлені в матричній формі;
- наявність багатоканальної архітектури, що дозволяє виконувати паралельну обробки декількох потоків даних (наприклад, каналів RGB [11]).

Однак можливість обробки вхідних даних нефіксованої структури та наявність декількох окремих каналів дозволяють використовувати згорткові шари в задачах обробки природної мови [12]. На відміну від рекурентної нейронної мережі, згорткова мережа не передбачає врахування порядку подання слів на вхідний шар; проте наявність окремих каналів дозволяє розширювати структуру мережі для аналізу інших властивостей тексту (наприклад, синтаксичних та просторових).

Попередня обробка тексту виконується аналогічно до попереднього методу: кожне речення представлене за допомогою попередньо навчених моделей векторного представлення слів тексту (1). Крім того, виконується розбиття документу на угрупування; когерентність тексту розраховується як добуток міри когерентності відповідних груп (3). Розглянемо детально процес оцінки когерентності кожного угрупування.

Формалізоване представлення кожного речення угрупування здійснюється в матричній формі: стовпці матриці формуються з векторних представлень слів речення. Кожна сформована матриця подається на вхід окремого каналу; таким чином, кількість каналів вхідного згорткового шару визначає розмір угрупування. На рис. 3 показано застосування операції згортки до матриць речень для формування відповідних векторів ознак (окремо для кожного каналу).

Спочатку згортковим шаром здійснюється виділення ознак за допомогою застосування множини фільтрів (матриць ваг $F \in \mathrm{R}^{d \times m}$, де m – «ширина вікна», d – розмірність векторного представлення слів речень). Фільтр рухається вздовж стовпців матриці $|S|$ з одиничним кроком; результатом виконання кожного кроку є вектор $c \in \mathrm{R}^{|S|-m+1}$, кожен елемент якого обраховується у наступний спосіб:

$$c_i = \left(S * F\right)_i = \sum_{k,j} \left(S_{[i-m+1:i]} \otimes F \right)_{kj}, \quad (4)$$

де $\otimes$ – операція попарного добутку елементів двох векторів. Результатом застосування фільтрів є множина карт характеристик C. Наступним компонентом мережі є шар субдискретизації, що виконує агрегацію

отриманої множини C: з кожного стовпця обирається максимальне значення. Таким чином, здійснюється формування вектору ознак для кожного вхідного речення. Далі здійснюється конкатенація векторів ознак всіх каналів до спільного вектору $\mathbf{h}_c$ та виконується оцінка когерентності угрупування за допомогою послідовного набору повнозв'язних шарів та вихідної функції softmax.

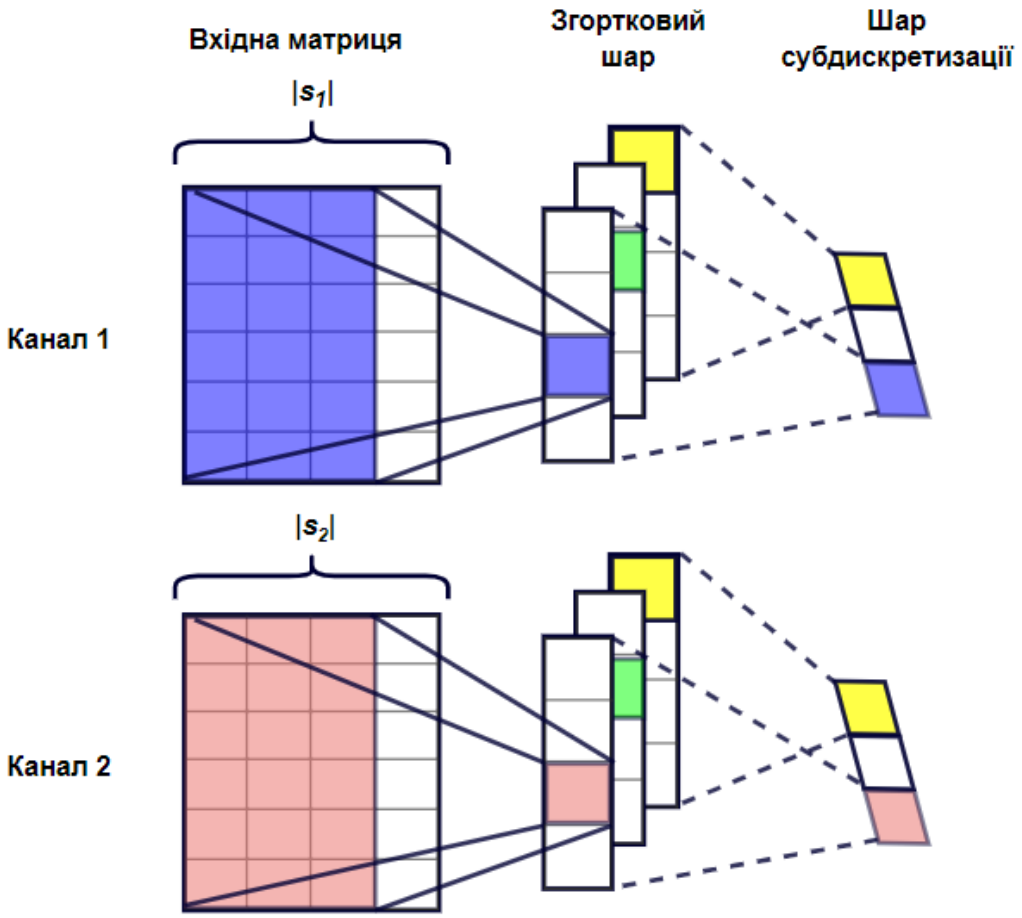

Рис. 3. Застосування операції згортки для формування вектору ознак (багатоканальна обробка)

Графічний метод з використанням графу семантичної схожості. На відміну від попередньо розглянутих методів, графічний метод з використанням графу семантичної схожості передбачає візуалізацію процесу формування оцінки когерентності тексту. Такий підхід дозволяє відстежити алгоритм розрахунку вихідного результату та виконати коригування тексту для підвищення оцінки когерентності.

Метод оснований на побудові орієнтованого графу тексту $G(V,E)$, де V – множина вершин, відповідних реченням тексту; E – множина ребер. Вага ребер інтерпретує міру семантичної схожості суміжних вершин (речень). Формалізація речень виконується за допомогою векторного представлення їх слів: кожному слову речення ставиться у відповідність вектор, отриманий з попередньо навченої моделі. Таким чином, речення s варто розглядати як множину векторів слів:

$$s = \left\{ \mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_M \right\}, \tag{5}$$

де M – кількість слів речення. Здійснення векторного представлення речення s виконується за допомогою усереднення відповідних векторів:

$$\mathbf{s} = \frac{1}{M} \sum_{k=1}^{M} \mathbf{w}_k \tag{6}$$

Для встановлення ребер та формування їх ваг можуть бути використані три різні підходи: PAV, SSV, MSV. У випадку застосування підходу *PAV* для кожної вершини виконується перевірка можливості встановлення ребра з вершинами, відповідними попереднім реченням. Якщо міра схожості між вершинами рівна 0, виконується спроба встановити ребро з іншою найближчою попередньою вершиною. Таким чином,

напівстепень виходу кожної вершини не перевищує 1. Міра схожості речень s_i та s_j розраховується у наступний спосіб:

$$\mathrm{sim}(s_i, s_j) = \alpha uot(s_i, s_j) + (1-\alpha)\cos(\mathbf{s}_i, \mathbf{s}_j), \tag{7}$$

де uot – відношення кількості спільних сутностей речень s_i та s_j до загальної кількості сутностей цих речень; $\cos(\mathbf{s}_i, \mathbf{s}_j)$ – косинусна відстань між векторами речень; α – регулятивний параметр, $\alpha \in [0,1]$.

У підході *SSV* для кожної вершини виконується пошук вершини з максимальним значеннями міри схожості; ребро встановлюється лише між поточною та виявленою вершинами. Отже, напівстепень виходу вершин не перевищує 1. Міра схожості речень s_i та s_j розраховується наступним чином:

$$\mathrm{sim}(s_i, s_j) = \frac{\cos(\mathbf{s}_i, \mathbf{s}_j)}{|i-j|} \tag{8}$$

Для підходу *MSV* ребра встановлюються між всіма вершинами, якщо значення ваги ребра перевищує задане порогове значення θ. Вага ребер розраховується за формулою (8).

Оцінка когерентності тексту D для всіх підходів розраховується як усереднене значення ваг ребер побудованого графу $G(V,E)$. Приклад побудованих графів для різних підходів показано на рис. 4.

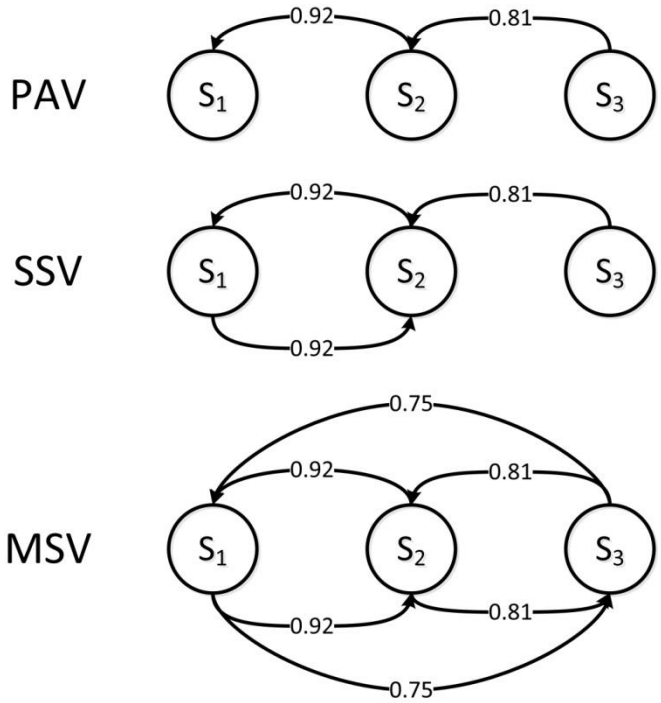

Рис. 4. Приклади побудови графу $G(V,E)$ для різних підходів методу графу семантичної схожості

Експериментальна перевірка методів оцінки когерентності україномовних текстів

Ефективність проаналізованих методів було вирішено перевірити на множині україномовних текстів. Процес експериментальної перевірки варто розділити на наступні послідовні етапи:

- збирання україномовних текстів;
- підготовка вхідних даних для навчання моделей машинного навчання;
- навчання моделей;
- розрахунок метрик ефективності методів оцінки когерентності текстів.

Збирання україномовних текстів. Навчальний та перевірочний корпуси україномовних текстів було вирішено сформувати з наукових статей. Такий вибір обумовлений наявністю певної структури наукових статей, відсутністю фразеологізмів та стислому логічному викладенні змісту робіт. Для автоматизованої екстракції даних з веб-сторінок наукових журналів використано метод Trinity [13]. Було виконано автоматизований аналіз 266 веб-сайтів українських наукових журналів різної тематики: «Природничі та точні

науки», «Соціогуманітарні науки». Для навчальної вибірки текстів екстраговано анотації; тестова вибірка сформована з повних версій статей. Автоматизоване детектування україномовних версій статей виконано за допомогою програмного пакету *langdetect* [14]. Повні версії статей доступні лише в форматі PDF; для екстракції власне тексту статті використано клієнт-серверне застосування Science Parse [15]. Враховуючи відсутність універсальної структури оформлення статті в різних журналах та похибку роботи застосування Science Parse, екстраговані тексти оброблено окремим програмним модулем для вилучення некоректних послідовностей символів (фрагментів без смислового навантаження): таблиць, підписів до рисунків, списків, колонтитулів, стоп-слів тощо.

Підготовка вхідних даних для навчання моделей машинного навчання. Результатом процесу збирання україномовних текстів є множина анотацій та повних версій україномовних статей. Попереднім етапом роботи всіх методів є здійснення навчання моделі семантичного векторного представлення слів чи речень. Як модель семантичного представлення слів обрано модель Word2Vec. На відміну від повних версій статей, анотації були екстраговані з HTML-сторінок, тому не містять некоректних послідовностей символів. Крім того, в анотаціях використовуються основні терміни, що стосуються роботи. Таким чином, було вирішено сформувати навчальний корпус для моделі Word2Vec з набору екстрагованих анотацій. До зазначеного набору застосовано операцію лематизації – приведення слів до нормальної форми. Сформований навчальний корпус також було використано для навчання моделей векторного представлення фрагментів тексту Doc2Vec [16] (DBOW, DM, DBOW+DM). Такий підхід дозволяє уникати процес формування векторного представлення речення за допомогою усереднення векторів його слів (6) та враховувати порядок їх розташування.

Для навчання рекурентної та згорткової нейронних мереж, з набору повних версій статей сформовано дві підмножини: навчальну та тестову. Щоб відстежити та передбачити процес перенавчання мереж, навчальну множину розбито на дві підмножини: власне навчальну та перевірочну. Формування вхідних наборів угрупувань здійснено за допомогою наступних етапів попередньої обробки текстів:

- токенізація – розбиття тексту на речення та слова;
- лематизація слів;
- генерація когерентних/некогерентних наборів угрупувань (перемішування всередині угрупування виявилося ефективнішим, ніж зміна порядку речень всього тексту).

Лематизацію та токенізацію текстів виконано за допомогою утиліт, наданих відкритою спільнотою фахівців *lang.org.ua* [17].

Навчання моделей. Програмну реалізацію розглянутих методів та відповідних нейронних мереж виконано за допомогою застосування, написаного мовою програмування Python 3.6. Навчання нейронних мереж Word2Vec, DBOW, DM, DM+DBOW здійснено з використанням вбудованих класів пакету *gensim* [18] з наступними параметрами:

- розмірність вектору – 300;
- кількість епох – 50;
- розмір «вікна» – 10;
- порогове значення частотного словника – 1.

Для проектування та навчання нейронних мереж використано бібліотеку Keras [19] (прикладний програмний інтерфейс для інших бібліотек TensorFlow, Theano). Навчання здійснювалося протягом 20 епох з використанням методу раннього зупину для уникнення перенавчання та збереження найточнішого варіанту мереж.

Розрахунок метрик ефективності методів оцінки когерентності текстів. Для розрахунку метрик ефективності методів оцінки когерентності текстів обраховано точність вирішення двох задач: *розрізнення документів* та *вставки*. Точність вирішення цих задач розраховується у наступний спосіб:

$$acc = \frac{CR}{TR}, \tag{9}$$

де CR – кількість коректно розпізнаних текстів; TR – загальна кількість текстів. Відмінність між задачами полягає у критерії вибору коректно розпізнаних текстів. У випадку вирішення задачі розрізнення документів здійснюється перемішування речень тексту; документ вважається коректно розпізнаним, якщо когерентність оригіналу більша, ніж когерентність зміненої версії. Для задачі вставки виконується наступна операція: випадковим чином обирається та вилучається речення тексту. Далі виконується вставка обраного речення у всі можливі позиції тексту, крім оригінальної. Документ вважається коректно розпізнаним, якщо когерентність оригіналу більша за всі розраховані когерентності версій тексту, отриманих шляхом вставки речення.

Розраховані точності вирішення задач розрізнення документів та вставки різними методами оцінки когерентності україномовних текстів з різними підходами (семантична модель, регулятивні параметри) наведено в таблиці.

Найвище значення точності вирішення задачі розрізнення документів отримано методом з використанням згорткової нейронної мережі [20]. Однак, точності вирішення задачі вставки для методів, основаних на нейронних мережах, нижчі, порівняно з методом графу семантичної схожості. Такий результат

обумовлений використанням підходу формування навчальної вибірки для нейронних мереж, що певною мірою відтворює задачу розрізнення документів. Формування угрупувань відповідно до задачі вставки потребує збільшення навчального корпусу та кількості вільних параметрів для покращення узагальнюючої властивості моделей. Найвище значення точності вирішення цієї задачі отримано методом графу семантичної схожості з підходом MSV, $\theta = 0$ [21]. Нульове значення порогового параметру та використання підходу MSV свідчать про необхідність врахування зв'язку між всіма реченнями, незалежно від їх міри семантичної схожості. Крім того, значення регулятивного параметру $\alpha = 0.8$ вказує на доцільність аналізу спільних елементів речень: кореферентних пар та однакових слів.

Таблиця. Точності вирішення задач розрізнення документів та вставки різними методами оцінки когерентності на корпусі україномовних текстів

Метод	Підхід	Семантична модель	Задача розрізнення документів, %	Задача вставки, %
Рекурентна нейронна мережа	–	Word2Vec	80.0	9.0
Згорткова нейронна мережа	–	Word2Vec	**99.0**	13.0
Граф семантичної схожості	PAV, $\alpha = 0.8$	Word2Vec	58.0	41.0
	SSV	Word2Vec	55.0	22.0
	MSV, $\theta = 0$	Word2Vec	70.0	51.0
	PAV, $\alpha = 0.8$	DBOW	62.0	40.0
	SSV	DBOW	55.0	22.0
	MSV, $\theta = 0$	DBOW	78.0	62.0
	PAV, $\alpha = 0.8$	DM	62.0	37.0
	SSV	DM	50.0	20.0
	MSV, $\theta = 0$	DM	80.0	**66.0**
	PAV, $\alpha = 0.8$	DBOW+DM	69.0	35.0
	SSV	DBOW+DM	63.0	22.0
	MSV, $\theta = 0$	DBOW+DM	70.0	32.0

Висновки

Проаналізовано методи оцінки когерентності текстів, основаних на використанні різнотипних нейронних мереж та графічній візуалізації тексту. Використання згорткових та рекурентних шарів дозволяє здійснювати обробку вхідних даних нефіксованої довжини: слів чи речень. Доцільність застосування рекурентного шару полягає у врахуванні порядку обробки вхідних слів речення, що певною мірою відтворює процес сприйняття тексту читачем. Перевагою використання згорткового шару є можливість здійснювати паралельну обробку даних за рахунок багатоканальної структури. На відміну від двох останніх графічний метод на основі графу семантичної схожості використовує нейронні мережі виключно для векторного представлення речень, тому є можливість відстежити процес формування вихідного результату та проаналізувати можливі варіанти підвищення оцінки когерентності тексту.

Здійснено експериментальну перевірку ефективності проаналізованих методів з різними підходами на множині україномовних наукових статей. Отримані результати вирішення задач розрізнення документів та вставки свідчать про доцільність використання методу граф семантичної схожості для оцінки когерентності україномовних текстів та необхідності врахування семантичного зв'язку між всіма реченнями. Точність роботи методів може бути підвищена за рахунок збільшення розміру навчальної вибірки, кількості вільних параметрів нейронних мереж та врахування синтаксичних і просторових властивостей тексту.

Література

1. Лєднік О. С. Когезія та когерентність як категорії зв'язного тексту. *Науковий часопис Національного педагогічного університету імені М. П. Драгоманова. Серія 10: Проблеми граматики і лексикології української мови*. 2010. № 6. С. 119–123.
2. Pogorilyy S., Kramov A. Coreference Resolution Method Using a Convolutional Neural Network. *Proceedings of 2019 IEEE International Conference on Advanced Trends in Information Theory*. December 18-20, 2019. Kyiv, Ukraine. P. 397–401.
3. Automated analysis of free speech predicts psychosis onset in high-risk youths / Bedi G., et al. *npj Schizophrenia*. 2015. Vol. 1. P. 1–7. DOI: 10.1038/npjschz.2015.30.
4. Cui B., Li Y., Zhang Y., Zhang Z. Text Coherence Analysis Based on Deep Neural Network. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management.* November 6-10, 2017. Singapore, Singapore. P. 2027–2030.

5. Li J., Hovy E. A model of coherence based on distributed sentence representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. October 25-29, 2014. Doha, Qatar. P. 2039–2048.
6. Giray G., Ünalır M. Assessment of text coherence using an ontology-based relatedness measurement method. *Expert Systems*. 2019. DOI: 10.1111/exsy.12505.
7. Хайкин С. Нейронные сети: полный курс, 2-е издание. Київ, 2016. 1104 с.
8. Погорілий С. Д., Крамов А. А., Яценко Ф. М. Метод аналізу когерентності україномовних текстів із використанням рекурентної нейронної мережі. *Математичні машини і системи*. 2019. № 4. С. 9–16.
9. Mikolov T., Sutskever I., Chen K., Corrado G. S., Dean J. Distributed representations of words and phrases and their compositionality. *Proceedings of the 26th International Conference on Neural Information Processing Systems*. December 5-10, 2013. Lake Tahoe, Nevada. P. 3111–3119.
10. Pennington J., Socher R., Manning D. C. GloVe: Global Vectors for Word Representation. *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014. P. 1532–1543.
11. Traffic Graph Convolutional Recurrent Neural Network: A Deep Learning Framework for Network-Scale Traffic Learning and Forecasting / Zhiyong Cui, et al. *IEEE Transactions on Intelligent Transportation Systems*. 2019. P. 1–12.
12. Kim Y. Convolutional neural networks for sentence classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. October 2014. Doha, Qatar. P. 1746–1751.
13. Pogorilyy S., Kramov A. Automated extraction of structured information from a variety of web pages. *Proceedings of the 11th International Conference of Programming UkrPROG 2018*. May 22-24, 2018. Kyiv, Ukraine. P. 149–158.
14. Language Detection Library for Java // Google Code Archive - Long-term storage for Google Code Project Hosting. URL: https://code.google.com/archive/p/language-detection (дата звернення: 23.01.2020).
15. Science Parse Server. URL: https://github.com/allenai/science-parse (дата звернення: 23.01.2020).
16. Le Q., Mikolov T. Distributed representations of sentences and documents. *International Conference on Machine Learning*. 2014. P. 1188–1196.
17. Homepage: lang-uk. URL: http://lang.org.ua (дата звернення: 23.01.2020).
18. Řehůřek R., Sojka P. Software Framework for Topic Modelling with Large Corpora. *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. 2010. P. 45-50.
19. Keras: The Python Deep Learning library. URL: https://keras.io (дата звернення: 23.01.2020).
20. Погорілий С.Д., Крамов А.А., Білецький П.В. Метод оцінки когерентності україномовних текстів з використанням згорткової нейронної мережі. *Збірник наукових праць Військового інституту Київського національного університету імені Тараса Шевченка*. 2019. № 65. С. 63–71.
21. Погорілий С.Д., Крамов А.А. Метод розрахунку когерентності українського тексту. *Реєстрація, зберігання і обробка даних*. 2018. № 4. С. 64–75.

References

1. Lednik O. Cohesion and coherence as a category of cohesive text. *Scientific journal of M.P. Dragomanov National Pedagogical University. Series 10: Problems of grammar and lexicology of the Ukrainian language*. [Online]. 2010. (6). P. 119–123. Available from: http://enpuir.npu.edu.ua/handle/123456789/15909. [Accessed: 23 January 2020].
2. Pogorilyy S. & Kramov A. Coreference Resolution Method Using a Convolutional Neural Network. In: *Proceeding of the 2019 IEEE International Conference on Advanced Trends in Information Theory*. 2019. P. 397–401. Available from: [Accessed: 20 February 2020].
3. Bedi G., Carrillo F., Cecchi G., Slezak D., Sigman M., Mota N., Ribeiro S., Javitt D., Copelli M. & Corcoran C. (2015). Automated analysis of free speech predicts psychosis onset in high-risk youths. *npj Schizophrenia*. 1 (1). Available from: [Accessed: 23 January 2020].
4. Cui B., Zhang Y. & Zhang Z. Text Coherence Analysis Based on Deep Neural Network. In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 2017, Singapore. P. 2027–2030. Available from: [Accessed: 23 January 2020].
5. Li J. & Hovy E. A model of coherence based on distributed sentence representation. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014. P. 2039–2048. Available from: [Accessed: 23 January 2020].
6. Giray G. & Ünalır M. (2019). Assessment of text coherence using an ontology-based relatedness measurement method. *Expert Systems*. Available from: [Accessed: 23 January 2020].
7. Haykin S. (2016). *Neural Networks: A Comprehensive Foundation Second Edition*. 2nd Ed. Kyiv.
8. Pogorilyy S., Kramov A. & Yatsenko F. . A method for analyzing the coherence of Ukrainian-language texts using a recurrent neural network. *Mathematical machines and systems*. 2019. 4. P. 9–16. Available from: [Accessed: 23 January 2020].
9. Mikolov T., Sutskever I., Chen K., Corrado G. & Dean J. Distributed representations of words and phrases and their compositionality. In: *Proceedings of the 26th International Conference on Neural Information Processing Systems*. 2013. P. 3111–3119. Available from: [Accessed: 23 January 2020].
10. Pennington J., Socher R. & Manning D.. GloVe: Global Vectors for Word Representation. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014. P. 1532–1543. Available from: [Accessed: 23 January 2020].
11. Cui Z., Henrickson K., Ke R., Pu Z. & Wang Y. Traffic Graph Convolutional Recurrent Neural Network: A Deep Learning Framework for Network-Scale Traffic Learning and Forecasting. In: *IEEE Transactions on Intelligent Transportation Systems*. 2019. P. 1–12. Available from: [Accessed: 23 January 2020].
12. Kim Y. Convolutional neural networks for sentence classification. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014, pp. 1746–1751. Available from: [Accessed: 23 January 2020].
13. Pogorilyy S. & Kramov A. Automated extraction of structured information from a variety of web pages. In: *Proceedings of the 11th International Conference of Programming UkrPROG 2018*. 2018. P. 149–158. Available from: [Accessed: 23 January 2020].
14. Nakatani S. [Online]. 2010. Language Detection Library for Java. Available from: https://code.google.com/archive/p/language-detection. [Accessed: 23 January 2020].
15. AI2 (2020). *allenai/science-parse*. [Online]. 2020. GitHub. Available from: https://github.com/allenai/science-parse. [Accessed: 23 January 2020].
16. Le Q. & Mikolov T. Distributed representations of sentences and documents. In: *International Conference on Machine Learning*. 2014, pp. 1188–1196. Available from: [Accessed: 23 January 2020].
17. Anon. *Homepage: lang-uk*. [Online]. 2020. Lang.org.ua. Available from: http://lang.org.ua. [Accessed: 23 January 2020].
18. Řehůřek R. & Sojka P. Software Framework for Topic Modelling with Large Corpora. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. 2010. P. 45–50. Available from: [Accessed: 23 January 2020].
19. Anon (2020). *Home - Keras Documentation*. [Online]. 2020. Keras.io. Available from: https://keras.io. [Accessed: 23 January 2020].

20. Pogorilyy S., Kramov A. & Biletskyi P. Method for coherece evaluation of Ukrainian texts using convolutional neural network. *Collection of scientific works of the Military Institute of Kyiv National Taras Shevchenko University*. 2019. (64). P. 5–13. Available from: [Accessed: 23 January 2020].
21. Pogorilyy S. & Kramov A. Method of the coherence evaluation of Ukrainian text. *Data Recording, Storage & Processing*. 2018. 20 (4). P. 64–75. Available from: [Accessed: 23 January 2020].

Одержано 21.02.2020

Про авторів:

Крамов Артем Андрійович,
аспірант факультету радіофізики,
електроніки та комп'ютерних систем
Київського національного університету імені Тараса Шевченка.
Кількість наукових публікацій в українських виданнях – 8.
Кількість наукових публікацій в зарубіжних виданнях – 1.
Індекс Гірша – 1.
https://orcid.org/0000-0003-3631-1268.

Погорілий Сергій Дем'янович,
доктор технічних наук, професор,
завідувач кафедри комп'ютерної інженерії
факультету радіофізики, електроніки та комп'ютерних систем
Київського національного університету імені Тараса Шевченка.
Кількість наукових публікацій в українських виданнях – 280.
Кількість наукових публікацій в зарубіжних виданнях – 50.
Індекс Гірша – 2.
https://orcid.org/0000-0002-6497-5056.

Місце роботи авторів:

Київський національний університет імені Тараса Шевченка.
03022, Київ, проспект Академіка Глушкова, 4Г.
Тел.: (044) 521-35-59.
E-mail: sdp77@i.ua,
artemkramovphd@knu.ua