УДК 004.891

В.Н. ВРУБЛЕВСЬКИЙ
Київський національний університет імені Тараса Шевченка, Київ, Україна,
e-mail: *vitalii.vrublevskyi@gmail.com.*

О.О. МАРЧЕНКО
Київський національний університет імені Тараса Шевченка, Київ, Україна,
e-mail: *omarchenko@univ.kiev.ua.*

РОЗРОБЛЕННЯ ТА ДОСЛІДЖЕННЯ МОДЕЛІ ПРЕДСТАВЛЕННЯ СЕМАНТИКИ РЕЧЕНЬ

Анотація. Наведено огляд ефективної та простої моделі представлення семантики речень у контексті задачі ідентифікації парафразів. Дерево залежностей обрано як основну структуру для представлення зв'язків між словами у реченні. Для представлення семантики слова використано попередньо навчені моделі представлення слів. На основі цих двох ключових складових розроблено декілька ознак, які допомагають точно визначити парафрази. Проведені експерименти довели, що модель є ефективною. Результати її застосування є відносно близькими до результатів найсучасніших моделей.

Ключові слова: оброблення природної мови, ідентифікація парафразів, семантична подібність, дерево залежностей, векторне представлення слів.

ВСТУП

Нині побудова моделей представлення семантики слів, речень та текстів природної мови справедливо посідає центральне місце в галузі комп'ютерної лінгвістики та штучного інтелекту загалом. Широку популярність мають такі моделі, як BERT, RoBERTa, AlBERT, GPT-2, GPT-3. Їх створили та обчислили найбільші світові IT-компанії з використанням надпотужних ресурсів своїх дата-центрів. Ці багатовимірні векторні моделі демонструють найкращі результати рівня state-of-the-art під час розв'язання переважної більшості задач комп'ютерної лінгвістики, залишаючи конкурентів далеко позаду.

Починаючи з моделей представлення семантики слів, дослідники намагалися закодувати у векторі слова інформацію про його контекст, наприклад, k сусідніх слова зліва та k сусідніх слова справа, як у моделях CBOW (continuous bag of words) та Skip-gram [1]. Пізніше, під час моделювання семантики речень у тексті за моделлю Skip-thought [2], текст представляли як послідовність речень подібно до представлення за моделями, зазначеними вище. Це уявлення є занадто спрощеним. У ньому не взято до уваги розмаїття різних нелінійних складних зв'язків між словами всередині речення, а також зовнішніх зв'язків між реченнями в межах тексту. Структура речення не є лінійною послідовністю слів, а має радше структуру дерев з елементами рекурсії. Ігнорування цих реалій мови може не спричиняти значних негативних наслідків, коли дослідники працюють з аналітичними мовами (наприклад, з англійською), де є чітко фіксований порядок слів у реченні. У випадку синтетичних мов, де порядок слів у реченні може вільно змінюватися, нехтування справжньою синтаксичною структурою може призвести до значного зниження ефективності моделі.

Головним завданням цієї роботи є дослідження ефективності використання синтаксичної структури речення як ключової ознаки під час побудови моделей представлення семантики речень природної мови. Для експериментального дослідження ефективності побудованих моделей семантики речень взято класичну задачу комп'ютерної лінгвістики — визначення парафразування.

Парафразування (парафраз) — це таке перетворення тексту, коли речення перефразовують або переписують, щоб сформувати лексично інше речення, яке має таке саме значення та зміст. Ідентифікація парафразу є класичною задачею машинного навчання у комп'ютерній лінгвістиці. Зазвичай система отримує на вхід два речення (чи тексти) і повинна вирішити, чи мають вони однакове значення та зміст. Є ще одна пов'язана з нею задача, відома як визначення «семантичної подібності». У цьому випадку системі потрібно оцінити ступінь подібності двох речень.

Розглянемо кілька прикладів парафразів, взятих з корпусу Microsoft Research Paraphrase (MSRP) [3]:

Amrozi accused his brother, whom he called "the witness", of deliberately distorting his evidence.

Referring to him as only "the witness", Amrozi accused his brother of deliberately distorting his evidence.

Два речення, наведені вище, є хорошим прикладом парафразовування. Два речення, наведені нижче, є прикладом речень, які не є парафразами:

Yucaipa owned Dominick's before selling the chain to Safeway in 1998 for $2.5 billion.

Yucaipa bought Dominick's in 1995 for $693 million and sold it to Safeway for $1.8 billion in 1998.

Як видно з прикладів, простий розгляд набору слів, що містяться в обох реченнях, не є правильним підходом.

Дерево залежності речення містить важливу синтаксичну інформацію. Дерево можна використовувати для розкладання великого речення на малі фрази. Можна спробувати зіставляти ці фрази між двома реченнями та використовувати нормалізовану кількість збігів як важливу ознаку.

У цій роботі використано заздалегідь навчені векторні моделі слів для відображення семантики слів та їхньої подібності. Ці моделі можна розглядати як основу для створення сучасних систем з оброблення природної мови. Кожне слово представлено вектором фіксованої довжини. Визначення подібності слів зводиться до обчислення відстані між цими векторами. Алгоритми обчислення векторів слів описано в роботах Томаса Міколова [1, 4].

Нами розроблено та реалізовано кілька алгоритмів обходу та агрегування ознак із застосуванням дерева залежностей, зокрема обчислення підграфів та схожість шляхів у дереві. Ці ознаки разом з деякими ознаками, які успішно використовують у задачах машинного перекладу (BLEU [5] тощо), були застосовані для тренування та налаштування ефективних моделей машинного навчання (наприклад, SVM [6]). Практичною метою розробки є створення моделі, яку можна було б використовувати в реальному часі у виробничих системах, тому основними критеріями є невелика складність обчислення ознак та швидкість прогнозування моделі.

ОГЛЯД РОБІТ ЗА НАПРЯМОМ ДОСЛІДЖЕНЬ

Мета цього огляду полягає не стільки у спробі охопити всі роботи, присвячені ідентифікації парафразів, скільки у демонстрації основних підходів та методів, які дослідники використовували у своїх розробленнях.

Здебільшого у дослідженнях застосовують деякі метрики подібності рядків. У роботі [7] використовують такий набір ознак: кількість повторюваних слів, спільні послідовні n-грами, скіп-грами та найдовшу загальну підпослідовність. Для того, щоб зафіксувати семантичну подібність, застосовують такі зовнішні лексичні ресурси, як WordNet [8].

Для вимірювання семантичної подібності коротких текстів у [9] використано метрики на основі корпусу та бази знань. Щоб обчислити корпусні ознаки, потрібно обробити великий корпус і здобути з нього інформацію про подібність слів. Для цього застосовують латентний семантичний аналіз [10]. Ознаки, що ґрунтуються на знаннях, також обчислюють за допомогою семантичних мереж (наприклад, WordNet). До того ж, було використано кілька ознак, що ґрунтуються на ієрархії таксономії WordNet.

Цікаві та неординарні ознаки запропоновано в роботі [11]. У ній зазначено, що оцінювання якості машинного перекладу тісно пов'язане з вимірюванням семантичної подібності на рівні речення. У вказаній роботі застосовано стандартні метрики оцінювання машинного перекладу (метрики BLEU та NIST, показники WER та PER) як ознаки для задачі ідентифікації парафразів. Тут

— WER — коефіцієнт помилок [12], що є кількістю операцій редагування, необхідних для перетворення одного речення в інше;

— PER — показник помилок слів, незалежний від позиції [13], в якому речення розглядається як множина слів, його обчислюють подібно до WER;

— BLEU — метрика, для визначення якої здійснюють порівняння n-грам речення кандидата з n-грамами коректного перекладу та підраховують кількість збігів;

— NIST [14] — метрика, яку також визначають на основі n-грам у спосіб, подібний до способу визначення метрики BLEU, але додатково аналізують, наскільки інформативною є кожна n-грама, і додають у формулу відповідні вагові коефіцієнти.

У роботі [15] зроблено спробу побудови векторних моделей для речень кожної пари. Основною ознакою є косинусна відстань між двома векторами. Автори досліджували як моделі додавання, так і скалярного множення. Модель адитивних векторних представлень слів демонструє хороші показники. Точність цього простого класифікатора становить $P = 0.73$, а метрика F_1 становить 0.82 для корпусу Microsoft Research Paraphrase.

Щоб виявити схожість синтаксичної структури, у [16], на додаток до ознак, побудованих на основі n-грам та метрик BLEU, використано метрики, що ґрунтуються на дереві залежностей. Серед них є оцінка точності та повноти, обчислена з використанням перетину залежностей, та метрики відстані, побудовані методом редагування дерева на основі динамічного програмування [17]. Метрика відстані редагування ґрунтується на сумі всіх операцій, потрібних для перетворення одного дерева в інше. На жаль, у цій роботі не вдалося повною мірою використати корпус MSRP через проблеми синтаксичного аналізу, але автори спромоглися досягти точності $P = 0.75$ та метрики $F_1 = 0.83$ на дещо меншій версії MSRP.

Нині, завдяки розвитку згорткових і рекурентних нейронних мереж, численні дослідники використовують їх у своїх роботах. У [18] застосовано рекурентну нейронну мережу для вивчення багатозначних моделей представлення слів з можливістю використання синтаксичної інформації з навколишнього контексту для навчання. Проблему ідентифікації парафразів розв'язано за допомогою сіамської архітектури [19] на основі вже вивчених моделей слів.

Найкраща сучасна модель має точність $P = 0.80$ та $F_1 = 0.85$ і ґрунтується на факторизації матриці з контрольованою зміною ваг. У роботі [20] використано схожість у прихованому просторі як головну ознаку для класифікації парафразів. У ній представлено нову схему зважування під назвою TF-KLD.

Вона ґрунтується на дивергенції Кульбека–Лейблера, обчисленій для двох розподілів Бернуллі (один — для речень, які позначено як парафрази, а другий — для тих, що позначені як не парафрази). Цю дивергенцію, перш ніж робити факторизацію матриці, використовують для зменшення ваги об'єктів у матриці терм-речення. Це впливає на збільшення ваги ознак, які можуть ефективно ідентифікувати парафраз. Окрім цього, використовують інші описані вище ознаки (*n*-грами, відношення залежності, BLEU та інші).

ОСНОВНА ІДЕЯ МОДЕЛІ

Як відомо, в машинному навчанні модель має математичну складову та релевантні ознаки, що актуалізують модель для деякої прикладної задачі. У цьому дослідженні основну увагу приділено підбору оптимального набору ознак, який би забезпечував максимальну ефективність роботи моделі під час класифікації парафразів. Водночас ключову роль відведено саме синтаксичним ознакам.

Головним завданням дослідження була спроба поєднати в моделі ознаки на рівні слів та дерево залежностей на рівні представлення речення подібно до [15, 16]. Дерево залежностей описує синтаксичну структуру речення з погляду слів та відповідних граматичних зв'язків між ними. Відношення між словами є чітко типізованими.

Наведемо набір ознак, застосованих в експериментах для різних моделей.

Перше та друге речення позначено відповідно s_1 та s_2.

1) Угорська вузлова збіжність.

Розділяємо речення на слова та обчислюємо косинусну відстань для кожної пари векторів слів:

$$similarity(\vec{w}_1, \vec{w}_2) = \frac{\vec{w}_1 \vec{w}_2}{|\vec{w}_1||\vec{w}_2|},$$

де $\vec{w}_1, \vec{w}_2$ — вектори слів з речень s_1 та s_2 відповідно.

На наступному кроці будуємо таблицю H у такий спосіб:

$$H(i, j) = 1 - similarity(\vec{w}_i, \vec{w}_j), \vec{w}_i \in s_1, \vec{w}_j \in s_2.$$

Потім застосовуємо угорський алгоритм [21] до отриманої матриці і отримуємо набір вузлів, що збіглися. Угорський алгоритм намагається поєднати слова з першого речення зі словами з другого, мінімізуючи вартість операції зіставлення. У цьому випадку ми використовуємо функцію подібності двох слів, щоб виразити штраф. Після цього відфільтровуємо неякісні збіги, порівнюючи схожість слів із заданим пороговим значенням. Відсоток якісних збігів застосовуємо далі як ознаку.

2) Коригувальна відстань між графами на основі угорського алгоритму.

Як і на першому кроці, повторно використовується вже розрахований збіг вузлів для позначення деяких слів однаковими у двох реченнях. Далі будуємо дерево залежностей та застосовуємо алгоритм, запропонований у роботі [22] для обчислення відстані між графами. Цей алгоритм також ґрунтується на угорському алгоритмі для обчислення мінімальної вартості редагування і використовує такі операції, як вставка, видалення та редагування вершини.

3) Шлях у дереві залежностей.

Для того, щоб обчислити цю ознаку, потрібно обійти дерево залежностей від кореня до листків і побудувати всі можливі шляхи заданої довжини:

$$getAllPaths(s, len) = \{[w_1, w_2, \ldots w_{len}]\},$$

$$w_i \in s, [w_i, w_{i+1}] \in dependancyTreeEdges(s),$$

$$i \in \{1, \ldots, len\}.$$

Для всіх шляхів обчислюємо агреговане векторне представлення, використовуючи векторні моделі слів, наявних у шляху:

$$pathEmbedding(path) = \sum_{w_i \in path} \vec{w}_i.$$

За допомогою векторного представлення шляхів їх можна легко порівнювати. Для побудови ознаки підраховуємо кількість шляхів, значення метрики схожості яких перевищує порогове значення:

$$DTPF(s_1, s_2, len) = \frac{2\left|\{p_1, p_2\}\right|}{PC_1 + PC_2},$$

де

$$p_1 \in getAllPaths(s_1, len),$$

$$p_2 \in getAllPaths(s_2, len):$$

$$similarity(pathEmbedding(p_1), pathEmbedding(p_2)) \geq \delta,$$

$$PC_1 = |getAllPaths(s_1, len)|,$$

$$PC_2 = |getAllPaths(s_2, len)|,$$

δ — порогове значення схожості.

Для нормалізації цієї ознаки використовуємо кількість усіх шляхів в обох реченнях.

4) Підграфи дерева залежностей.

Для того, щоб обчислити цю ознаку, потрібно побудувати всі підграфи з фіксованою глибиною, використовуючи початкове дерево залежностей:

$$getAllSubgraphs(s, d) = \{g\},$$

де g — підграф $dependencyTree(s)$, $depth(g) = d$.

Для всіх підграфів з цієї множини потрібно обчислити векторне представлення, використовуючи векторні моделі слів, наявних у графі:

$$graphEmbedding(graph) = \sum_{w_i \in getNodes(graph)} \vec{w}_i.$$

За допомогою векторного представлення графи можна порівнювати між собою. Знову застосовуємо поріг схожості та враховуємо лише якісні збіги.

$$DTGF(s_1, s_2, depth) = \frac{2\left|\{g_1, g_2\}\right|}{GC_1 + GC_2},$$

де

$$g_1 \in getAllSubgraphs(s_1, depth),$$

$$g_2 \in getAllSubgraphs(s_2, depth),$$

$$similarity(graphEmbedding(g_1), graphEmbedding(g_2)) \geq \xi,$$

$$GC_1 = |getAllSubgraphs(s_1, depth)|,$$

$$GC_2 = |getAllSubgraphs(s_2, depth)|.$$

ξ — порогове значення схожості графів.

Для того, щоб нормалізувати цю ознаку, використовуємо кількість усіх підграфів в обох реченнях.

5) Підграфи дерева залежностей зі зважуванням міри Idf.

Міра Idf означає обернену частоту документа. Її обчислюють так:

$$idf(w, D) = \log \frac{N}{1 + s \in D \mid w \in s} + 1,$$

де D — множина речень всього корпусу, N — кількість речень в корпусі, $N = |D|$.

Ознаку будують за тим самим алгоритмом, описаним у попередньому пункті, проте у побудові векторного представлення графу міру Idf слова використовують як ваговий коефіцієнт.

6) Основні ознаки довжини речення:

$$|s_1|, |s_2|, \| s_1| - |s_2 \| .$$

7) Перетин n-грам:

$$NGO(s_1, s_2, n) = \frac{|NGram(s_1, n) \cap NGram(s_2, n)|}{|NGram(s_1, n)|},$$

де $NGram(s, n)$ — множина n-грам довжини n речення s.

8) Перетин залежностей — порівняння ребер дерев залежностей:

$$DO(s_1, s_2) = \frac{\left| dependencyEdges(s_1) \cap dependencyEdges(s_2) \right|}{\left| dependencyEdges(s_1) \right|},$$

де $dependencyEdges(s) = \{(w_1, w_2)\}$, w_1 та w_2 зв'язані ребром у дереві залежностей.

9) Синтаксичний перетин n-грам [23].

Основна різниця між синтаксичними n-грамами та звичайними полягає в тому, які елементи вважають сусідами. У випадку звичайних n-грам використовують порядок слів у реченнях, а для синтаксичних n-грам — дерево залежностей. Слова вважають сусідами, якщо вони зв'язані у дереві.

10) Метрика BLEU.

$$BLEU(s_1, s_2) = BP(s_1, s_2) \exp\left[\sum_{n=1}^{N} \frac{1}{N} \log(p_n) \right],$$

$$BP(s_1, s_2) = \exp\left[\min\left[1 - \frac{|s_1|}{|s_2|}, 1 \right] \right],$$

$$p_n = \frac{\sum_{x \in NGram(s_1, n)} count(x, NGram(s_1, n) \cap NGram(s_2, n))}{\sum_{x \in NGram(s_1, n)} count(x, NGram(s_1, n))},$$

$$count(x, S) = |el \mid el \in S \& el = x|,$$

де N — максимальна довжина n-грами, BP (Brevity Penalty) — коефіцієнт стислості, призначений для штрафування речень, якщо одне коротше за інше.

ОПИС ТА РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТУ

Для тестування використано корпус MSRP. Він складається з 5800 пар речень, отриманих із новин, розміщених у мережі «Інтернет», а також анотацій, що вказують, чи є пари речень парафразами. Навчальна вибірка

складається з 4076 пар речень (з-поміж них 2753 вважаються справжніми парафразами, їхня частка становить ~ 67%). Тестовий набір складається з 1725 пар речень (з-поміж них 1147 вважаються справжніми парафразами, їхня частка становить ~ 66%). Розподіл справжніх парафразів як у навчальному, так і в тестовому наборі майже однаковий. Завдяки цьому немає потреби у збалансуванні даних.

Для експериментів вибрано набір таких класичних моделей, як SVM, логістична регресія, дерево рішень, Random Forest та ін., доступних у бібліотеці scikit-learn Python [24]. Використано попередньо навчену статистичну модель для англійської мови, розроблену Spacy [25] для синтаксичного аналізу речень і побудови дерева залежностей та векторних представлень слів. Цю модель згенеровано з використанням згорткової нейронної мережі та корпусу OntoNotes [26].

Для того, щоб оцінити ефективність застосування різних ознак, проведено тести з різними класифікаторами. Результати цих тестів наведено у табл. 1.

Методика експерименту полягала у тренуванні декількох класифікаторів на навчальній вибірці з використанням різних наборів ознак та їхньому оцінюванні на тестовій. Основними метриками вибрано точність P та метрику F_1.

$$F_1 = \frac{2 \cdot (Recall \cdot Precision)}{Recall + Precision}.$$

У табл. 1 представлено класифікатори з найкращими показниками F_1 для кожної групи ознак.

За результатами проведених експериментів найкращим класифікатором у переважній більшості випадків став метод опорних векторів SVM.

Ознаки на основі підграфів дерева залежностей виявилися найбільш ефективними серед розглянутих. Це свідчить про широкі можливості використання дерев підпорядкування у моделюванні семантики речень для задач визначення семантичної подібності.

У табл. 2 для порівняння наведено показники запропонованої моделі та інших, зокрема найсучасніших.

Таблиця 1. Показники ефективності ознак

Ознака	Класифікатор	Точність P (%)	Метрика F_1 (%)
Угорська вузлова збіжність	SVM	73	82,3
Коригувальна відстань між графами на основі угорського алгоритму	SVM	72,4	81,9
Шлях у дереві залежностей	SVM	73,4	82,5
Підграфи дерева залежностей	SVM	73,6	82,7
Підграфи дерева залежностей зі зважуванням Idf	SVM	73,0	82,4
Основні ознаки довжини речення + ознаки на основі n-грам	Ridge	74,3	82,4
BLEU	SVM	73,0	82,3
Всі	SVM	75,4	83,6

Таблиця 2. Показники ефективності запропонованої моделі та передових аналогів

Алгоритм	Точність P (%)	Метрика F_1 (%)
KM [7]	76,6	79,6
MCS [9]	70,3	81,3
ParaDetect [27]	74,7	81,8
Схожість на основі векторного представлення [15]	73,0	82,0
SDS [28]	73,0	82,3
SAMS-RecNN [18]	78,6	85,3
TF-KLD [20]	80,4	85,9
Запропонована модель	**75,4**	**83,6**

ВИСНОВКИ

Результати розробленої моделі не перевершують рівень кращих сучасних моделей, проте впевнено виводять її до п'ятірки лідерів за показником F [29]. Водночас наша модель значно переважає конкурентів за простотою реалізації та швидкістю роботи системи. Головним результатом дослідження є експериментальне підтвердження того, що поєднання дерев залежностей та векторного представлення слів можна ефективно використовувати для побудови якісних моделей представлення семантики речень. Це успішно продемонстровано на класичній задачі ідентифікації парафразів.

СПИСОК ЛІТЕРАТУРИ

1. Mikolov T., Chen K., Corrado G., Dean J. Efficient estimation of word representations in vector space. *Proc. Workshop at ICLR.* 2013. URL: https://arxiv.org/pdf/1301.3781.pdf.
2. Kiros R., Zhu Y., Salakhutdinov R., Zemel R.S., Torralba A., Urtasun R., Fidler S. Skip-thought vectors. *Proc. 28th International Conference on Neural Information Processing Systems (NIPS 2015)*. (7–12 December 2015, Montreal, Canada). Montreal, 2015. Vol. 2. P. 3294–3302.
3. Dolan B., Quirk C., Brockett C. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. *Proc. 20th International Conference on Computational Linguistics (COLING 2004).* (23–27 August 2004, Geneva, Switzerland). Geneva, 2004. P. 350–356. URL: https://aclanthology.org/C04-1051.
4. Mikolov T., Sutskever I., Chen K., Corrado G.S., Dean J. Distributed representations of words and phrases and their compositionality. *Proc 26th International Conference on Neural Information Processing Systems (NIPS 2013).* (5–10 December 2013, Lake Tahoe, Nevada, USA). Lake Tahoe Nevada, 2013. Vol. 2. P. 3111–3119.
5. Papineni K., Roukos S., Ward T., Zhu W.-J. Bleu: A method for automatic evaluation of machine translation. *Proc. 40th Annual Meeting on Association for Computational Linguistics (ACL'02).* (7–12 July 2002, Philadelphia, Pennsylvania, USA). Philadelphia, 2002. P. 311–318. https://doi.org/10.3115/1073083.1073135.
6. Cortes C., Vapnik V. Support-vector networks. *Mach. Learn.* 1995. Vol. 20. P. 273–297. https://doi.org/10.1007/BF00994018.

7. Kozareva Z., Montoyo A. Paraphrase identification on the basis of supervised machine learning techniques. *Proc. 5th International Conference on Natural Language Processing (FinTAL 2006).* (23–25 August 2006, Turku, Finland). Turku, 2006. Advances in Natural Language Processing. P. 524–533. https://doi.org/10.1007/11816508_52.
8. Fellbaum C. WordNet: An Electronic Lexical Database. MIT Press, 1998. 449 p. https://doi.org/10.7551/mitpress/7287.001.0001.
9. Mihalcea R., Corley C., Strapparava C. Corpus-based and knowledge-based measures of text semantic similarity. *Proc. 21st national conference on Artificial intelligence (AAAI '06).* (16-20 July 2006, Boston, Massachusetts). Boston, 2006. Vol.1. P. 775–780.
10. Landauer T.K., Foltz P.W., Laham, D. An introduction to latent semantic analysis. *Discourse Processes.* 1998. Vol. 25, Iss. 2-3. P. 259–284. https://doi.org/10.1080/01638539809545028.
11. Finch A., Sumita E. Using machine translation evaluation techniques to determine sentence-level semantic equivalence. *Proc. 3rd International Workshop on Paraphrasing (IWP 2005).* (11–13 October 2005, Jeju Island, Korea). Jeju Island, 2005. URL: https://aclanthology.org/I05-5003.
12. Su K.Y., Wu M.W., Chang J.S. A new quantitative quality measure for machine translation systems. *Proc. 14th conference on Computational linguistics (COLING-92).* (23-28 August 1992, Nantes, France). Nantes, 1992. Vol. 2, P. 433–439. https://doi.org/10.3115/992133.992137.
13. Nieβen S., Vogel S., Ney H., Tillmann C. A DP based search algorithm for statistical machine translation. *Proc. 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (ACL '98/COLING '98).* (10–14 August 1998, Montreal, Quebec, Canada). Montreal, Quebec, 1998. Vol. 2. P. 960–967. https://doi.org/10.3115/980691.980727.
14. Doddington G. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. *Proc. 2nd International Conference on Human Language Technology Research (HLT '02).* (24–27 March 2002, San Diego, California, USA). San Diego, 2002. P. 138–145.
15. Milajevs D., Kartsaklis D., Sadrzadeh M., Purver M. Evaluating neural word representations in tensor-based compositional settings. *Proc. 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).* (25-29 October 2014, Doha, Qatar). Doha, 2014. P. 708–719. https://doi.org/10.3115/v1/D14-1079.
16. Wan S., Dras M., Dale R., Paris C. Using dependency-based features to take the "para-farce" out of paraphrase. *Proc. Australasian Language Technology Workshop (ALTA).* (30 Nov.-1 Dec. 2006, Sydney, Australia). Sydney, 2006. P. 131–138.
17. Zhang K., Shasha D. Simple fast algorithms for the editing distance between trees and related problems. *SIAM Journal of Computing.* 1989. Vol. 18, Iss. 6. P. 1245–1262. https://doi.org/10.1137/0218082.
18. Cheng J., Kartsaklis D. Syntax-aware multi-sense word embeddings for deep compositional models of meaning. *Proc. 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015).* (17–21 September 2015, Lisbon, Portugal). Lisbon, 2015. P. 1531–1542. https://doi.org/10.18653/v1/D15-1177.
19. Bromley J., Bentz J.W., Bottou L., Guyon I., LeCun Y., Moore C., Sackinger E., Shah R. Signature verification using a siamese time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence.* 1993. Vol. 7, N 4. P. 669–688. https://doi.org/10.1142/S0218001493000339.

20. Ji Y., Eisenstein J. Discriminative improvements to distributional sentence similarity. *Proc. 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*. (18-21 October 2013, Seattle, Washington, USA). Seattle, 2013. P. 891–896.

21. Kuhn H.W. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly.* 1955. Vol. 2, Iss. 1–2. P. 83-97. https://doi.org/10.1002/nav.3800020109.

22. Riesen K., Neuhaus M., Bunke H. Bipartite graph matching for computing the edit distance of graphs. In: Graph-Based Representations in Pattern Recognition. Escolano F., Vento M. (Eds). *Lecture Notes in Computer Science.* 2007. Vol 4538. P. 1–12. https://doi.org/10.1007/978-3-540-72903-7_1.

23. Sidorov G., Castillo F., Stamatatos E., Gelbukh A., Chanona-Hernández L. Syntactic N-grams as machine learning features for natural language processing. *Expert Systems with Applications*. 2014. Vol. 41, Iss. 3. P. 853-860. https://doi.org/10.1016/j.eswa.2013.08.015.

24. Scikit-learn. Machine learning in Python. URL: https://scikit-learn.org/stable/.

25. SpaCy. URL: https://spacy.io/.

26. Weischedel R., Hovy E., Marcus M., Palmer M., Belvin R., Pradhan S., Ramshaw L., Xue N. OntoNotes: A large training corpus for enhanced processing. In: Handbook of Natural Language Processing and Machine Translation: DARPA Global Autonomous Language Exploitation. Olive J., Christianson C., McCary J. (Eds.). New York: Springer-Verlag, 2011. XXVI, 936 p.

27. Ul-Qayyum Z., Altaf W. Paraphrase identification using semantic heuristic features. Research *Journal of Applied Sciences, Engineering and Technology.* 2012. Vol. 4. N 22. P. 4894-4904.

28. Blacoe W., Lapata M. A comparison of vector-based representations for semantic composition. *Proc. 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning.* (12–14 July 2012, Jeju Island, Korea). Jeju Island, 2012. P. 546–556.

29. Paraphrase Identification (State of the art) URL: https://aclweb.org/aclwiki/Paraphrase_Identification_(State_of_the_art).

V. Vrublevskyi, O. Marchenko

DEVELOPMENT AND ANALYSIS OF THE MODEL FOR SENTENCE SEMANTIC REPRESENTATION

Abstract. The authors overview an efficient and simple model of sentence semantic representation for the paraphrase identification problem. The dependency tree was chosen as the main structure to represent the relationships between words in a sentence. To represent the word semantics, pre-trained general-purpose word embeddings are used. Based on these two key components, several features that can help to identify paraphrases are designed. The experiments were conducted, which proved the model efficiency. The results of the model application are rather close to those for state-of-the-art models

Keywords: natural language processing, paraphrase identification, semantic similarity, dependency tree, word embeddings.

Надійшла до редакції 26.08.2021