Аналіз Методів Обробки Природної Мови

Костянтин Онищенко
кафедра Програмної інженерії
Харківський національний
університет радіоелектроніки
Харків, Україна
kostiantyn.onyshchenko@nure.ua

Яна Данієль
кафедра Програмної інженерії
Харківський національний
університет радіоелектроніки
Харків, Україна
yana.daniiel@nure.ua

Роман Каменєв
кафедра Програмної інженерії
Харківський національний
університет радіоелектроніки
Харків, Україна
roman.kameniev@nure.ua

Analysis of Natural Language Processing Methods

Kostiantyn Onyshchenko
department of System Engineering
Kharkiv National University
of Radio Electronics
Kharkiv, Ukraine
kostiantyn.onyshchenko@nure.ua

Yana Daniiel
department of System Engineering
Kharkiv National University
of Radio Electronics
Kharkiv, Ukraine
yana.daniiel@nure.ua

Roman Kameniev
department of System Engineering
Kharkiv National University
of Radio Electronics
Kharkiv, Ukraine
roman.kameniev@nure.ua

***Анотація*—Розглянуто підходи та методи, що використовуються для розв'язування задач оброки природної мови. Виділені перспективні напрямки розвитку галузі, окреслені переваги та недоліки розглянутих методів у задачах «розуміння» природної мови, перекладу такого тексту та відповіді на питання.**

***Abstract*—The approaches and methods used to link natural language tasks are discussed. The separate perspective directions of development of branch, separate advantages and lacks consider methods in a problem of "understanding" of natural language, translation of such text and answers to questions.**

Ключові слова—обробка природної мови, машинне навчання, глибинне навчання, інструменти машинного навчання, методи

Keywords—natural language processing; machine learning; deep learning; machine learning tools; methods

I. Вступ

В останні десятиріччя спостерігається значний інтерес до вирішення задач обробки тексту. Із розвитком голосових помічників та чат-ботів все більше постає необхідність у «розумінні» та інтерпретації природної мови. Однозначність сприйняття тексту суттєво впливає на процес обробки закладених даних та отримання результатів.

Обробка природної мови (NLP – Natural Language Processing) – це підрозділ інформаційних технологій, штучного інтелекту та лінгвістики, метою якого є вивчення проблем комп'ютерного аналізу та синтезу природної мови. Повне розуміння та відтворення сенсу мови – надзвичайно складне завдання, оскільки людська мова має цілий ряд особливостей.

Людська мова – це спеціально сконструйована система передачі сенсу сказаного або написаного, якій притаманні дискретні, категоріальні або символьні властивості. Така система володіє особливим кодуванням та усвідомленою передачою інформації, що вирізняється стійкістю та надійністю. Категоріальні символи мови кодуються як сигнали для спілкування по декількох каналах: звук, жести, лист, зображення, інше. При цьому мова здатна виражатися будь-яким способом.

Сьогоденне використання NLP зводиться до вирішення задач пошуку (письмовий або усний), підбору контекстної онлайн-реклами, автоматичного або напівавтоматичного перекладу, аналізу настроїв для задач маркетингу, розпізнавання мови, чат-ботів та голосових помічників.

II. Глибинне навчання в NLP

Істотна частина технологій NLP працює завдяки глибинному навчанню (Deep Learning) – галузі машинного навчання, що ґрунтується на наборі алгоритмів, які намагаються моделювати високорівневі абстракції в даних, застосовуючи глибинний граф із декількома обробними шарами, що побудовано з кількох лінійних або нелінійних перетворень [1].

Доцільність використання глибинного навчання зумовлюється наступними факторами:

- накопичено великі обсяги тренувальних даних;
- розроблено обчислювальні потужності: багатоядерні CPU і GPU;
- створено нові моделі і алгоритми з розширеними можливостями і поліпшеною продуктивністю, з гнучким навчанням на проміжних уявленнях;

- з'явилися навчальні методи з використанням контексту, нові методи регуляризації та оптимізації.

Успішне використання більшості методів машинного навчання було досягнуто завдяки наявності репрезентативних даних, вхідних ознак, та оптимізації ваг, для підвищення точності фінального передбачення [1].

У глибинному навчанні алгоритм намагається автоматично витягти кращі ознаки (подання) з сирих вхідних даних. Створені вручну ознаки можуть бути занадто спеціалізованими, неповними та такими, що потребують часу для створення та затвердження. На противагу цьому, виявлені глибинним навчанням ознаки легко пристосовуються.

Глибинне навчання пропонує гнучкий, універсальний та навчаємий фреймворк для подання інформації за допомогою візульного та лінгвістичного представлень.

III. Векторне представлення (text embeddings)

У традиційному NLP слова розглядаються як дискретні символи, які в подальшому представлені у вигляді одноразових векторів. Недолік такого способу представлення слів полягає у відсутності визначення схожості для одноразових векторів. Вирішити дану проблему можна за допомогою кодування схожості у самих векторах [2].

Векторне представлення – це метод представлення строк у вигляді векторів зі значеннями. Створюється плоский (щільний) вектор для кожного слова так, щоб слова у схожих контекстах мали подібні вектори. Такий спосіб представлення враховує стартову точку для більшості завдань NLP та робить глибинне навчання ефективним на малих датасетах [2]. Техніки векторних представлень Word2vec і GloVe, створених Google, користуються популярністю та часто використовуються для задач NLP. Розглянемо їх.

A. Word2Vec

Word2vec приймає великий корпус (corpus) тексту, в якому кожне слово у фіксованому словнику подано у вигляді вектору. Алгоритм проходить по кожній позиції *t* в тексті, яка представляє собою центральне слово *c* і контекстне слово *o*. Далі використовується схожість векторів слів для *c* та *o*, щоб розрахувати ймовірність *o* при заданому *c* (або навпаки), і триває регулювання вектор слів для максимізації цієї ймовірності (рис. 1).

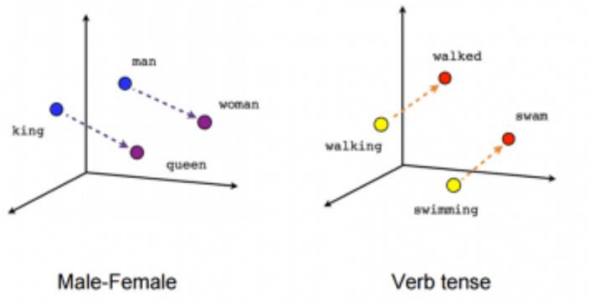

Рис. 1. Приклад роботи алгоритму Word2Vec

Для досягнення кращого результату роботи Word2vec, з датасету видаляються непотрібні слова (або слова з великою вживаністю, в англійській мові - a, the, of, then, тощо). Це допомагає збільшити точність моделі та скоротити час на тренування. Крім того, використовується негативна вибірка (negative sampling) для кожного входу, оновлюючи ваги для всіх правильних міток, але тільки на невеликій кількості некоректних міток.

Word2Vec представлено у 2 варіаціях моделей (рис. 2).

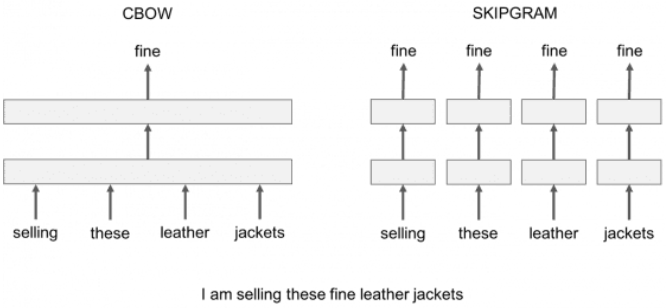

Рис. 2. Моделі CBOW та SKIPGRAM

При використанні моделі SKIPGRAM розглядається контекстне вікно, що містить *k* послідовних слів. Далі пропускається одне слово і відбувається процес навчання нейронної мережі, що містить всі слова, крім пропущеного. Алгоритм намагається передбачити його. Якщо 2 слова періодично поділяють схожий контекст в корпусі, ці слова будуть мати близькі вектори.

При використанні моделі CBOW (Continuous Bag of Words) береться багато пропозицій в корпусі. Кожен раз, коли алгоритм бачить слово, береться сусіднє слово. Далі на вхід нейромережі подається контекстні слова і передбачається слово в центрі цього контексту. У випадку великої кількості тематичних слів і центрального слова, отримуємо один екземпляр датасету для нейромережі. Нейромережа тренується та подає на виході закодований прихований шар, що представляє собою вкладення (embedding) для певного слова. Аналогічна ситуація відбувається, якщо нейромережа тренується на великій кількості пропозицій та слів. В схожому контексті для них формуються схожі вектори.

Недоліком даних методів є приналежність до класу window-based моделей, для яких характерна низька ефективність використання статистики збігів в корпусі, що призводить до неоптимальних результатів.

B. GloVe

GloVe має на меті вирішити цю проблему захопленням значення одного word embedding зі структурою всього доступного для огляду корпусу. Щоб зробити це, модель шукає глобальні збіги числа слів і використовує досить статистики, мінімізує середньоквадратичне відхилення, видає простір вектора слова з розумною Субструктура. Така схема в достатній мірі дозволяє ототожнювати схожість слова з векторним відстанню.

IV. Машинний переклад

Машинний переклад (Machine translation) - перетворення тексту на одному природною мовою в еквівалентний за змістом текст на іншій мові. Машинний переклад виконується програмою або машиною без участі людини. У машинному перекладі використовується статистика використання слів по сусідству [3].

У традиційних системах машинного перекладу доводиться використовувати паралельний корпус - набір текстів, кожен з яких перекладено на один або декілька інших мов. Наприклад, маючи вихідних мову *f* (Французький) і цільову *e* (Англійська), потрібно побудувати статистичну модель, що включає вірогіднісне формулювання для правила Байєса, модель перекладу *p (f | e)*, навчену на паралельному корпусі, і модель мови *p (e)*, навчену тільки на корпусі з англійською мовою [3].

Даний підхід пропускає сотні важливих деталей, вимагає великої кількості спроектованих вручну ознак, складається з різних і незалежних завдань машинного навчання.

Нейромережевий машинний переклад (Neural Machine Translation, NMT) - підхід до моделювання перекладу за допомогою рекурентної нейронної мережі (Recurrent Neural Network, RNN). RNN - нейромережа із залежністю від попередніх станів, що має зв'язки між ітераціями. Нейрони отримують інформацію з попередніх шарів та з самих себе на попередньому етапі. Це означає, що порядок, в якому подається на вхід дані і тренується мережа, важливий: результат подачі "Міккі" - "Маус" не збігається з результатом подачі "Маус" - "Міккі" (рис. 3).

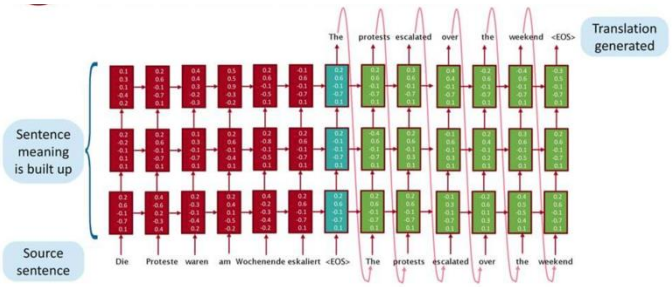

Рис. 3. Neural Machine Translation

Стандартна модель нейро-машинного перекладу є наскрізною нейромережею, де вихідна пропозиція кодується RNN (encoder), а цільове слово передбачається за допомогою іншої RNN (decoder). Кодувальник «читає» вихідну пропозицію зі швидкістю один символ в одиницю часу, після чого об'єднує вихідну пропозицію в останньому прихованому шарі. Декодер використовує зворотне поширення помилки для вивчення цього об'єднання і повертає перекладений варіант.

Головна проблема RNN - це зникнення градієнта, коли інформація втрачається з часом. Спершу можна сприйняти цю проблему як не серйозну, оскільки це ваги, а не стани нейронів. Але, з часом, ваги стають місцями для зберігання інформації з минулого. Якщо вага прийме значення 0 або 100000, попередній стан втратить свою інформативність. Як наслідок, RNN будуть зазнавати труднощів у запам'ятовуванні слів, що стоять далі в послідовності, а передбачення будуть робитися на основі крайніх слів.

Мережі довгої короткочасної пам'яті (Long / short term memory, далі LSTM) борються з проблемою градієнта зникнення шляхом введення гейтів (gates) та ячеєк пам'яті. Кожен нейрон представляє з себе ячейку пам'яті з трьома гейтами: на вхід, вихід та забування (forget). Ці затвори виконують функцію охоронців інформації, дозволяючи або забороняючи потік даних.

Вхідний гейт визначає кількість інформації, що буде зберігатися в цьому осередку з попереднього шару. Вихідний гейт визначає, яка частина наступного шару дізнається про стан поточної ячейки. Гейт забування контролює міру збереження значення в пам'яті. Наприклад: при вивченні книги починається новий розділ, тому, іноді, для нейромережі стає необхідним забути деякі слова з попереднього розділу.

LSTM мережі здатні навчатися на складних послідовностях. Вони поширені і використовуються в машинному перекладі, здатні до написання текстів в конкретному стилі та можуть використовуватись для складення простої музики. Крім цього, це стандартна модель для більшості завдань маркування (labeling) послідовності, які складаються з великої кількості даних.

Закриті рекурентні блоки (Gated recurrent units, GRU) відрізняються від LSTM, хоча теж є розширенням для нейромережевого машинного навчання. У GRU міститься на один гейт менше та інший принцип принцип роботи: відсутні вхідний та вихідний гейти, а також гейт забування. Замість них у GRU є гейт поновлення (update gate). Він визначає, скільки інформації необхідно зберегти з останнього стану і скільки інформації пропускати з попередніх шарів.

Функції скидання гейта (reset gate) схожа на затвор забування у LSTM, але відрізняється розташуванням. GRU завжди передає свій повний стан, не маючи вихідного затвору. За функціональністю цей затвор схожий на аналогічний у LSTM, однак, в GRU затвор працюють швидше і легше в управлінні. На практиці вони прагнуть нейтралізувати один одного, оскільки потрібна велика нейромережа для відновлення виразності (expressiveness), яка зводить нанівець прирости результативності. У випадках, де не потрібно екстра виразності, GRU показують кращий результат, ніж LSTM.

Крім цих трьох головних архітектур, за останні кілька років з'явилося багато покращень в нейромережевому машинному перекладі:

• Sequence-to-Sequence Learning with Neural Networks довели ефективність LSTM для нейронного машинного перекладу. Цей метод використовує багатошарову LSTM, щоб відобразити вхідну послідовність у вигляді вектора з

фіксованою розмірністю, далі йде застосування іншої LSTM для декодування цільової послідовності з вектора.

• Neural Machine Translation by Jointly Learning to Align and Translate представив механізм уваги (attention mechanism) в NLP. Визнаючи факт, що використання вектора фіксованої довжини є вузьким місцем в поліпшенні результативності NMT, було запропоновано дозволити моделі автоматично шукати частини вихідної пропозиції, що релевантні до передбачення цільового слова (без необхідності явного формування цих частин).

• Google створила власну NMT систему, Google's Neural Machine Translation, яка вирішує завдання точності і простоти застосування. Модель складається з глибокої LSTM мережі з 8 кодуючими та 8 декодуючими шарами і використовує як залишкові зв'язку, так і attention-зв'язку від декодермережі до кодермережі.

• Замість використання рекурентних нейромереж, Facebook AI Researchers використовують згорткову нейронну мережу для задач sequence-to-sequence навчання в NMT.

V. Голосові помічники

Через обмеженість можливостей штучного інтелекту у обробці природної мови, створення повноцінного розмовного асистента залишається відкритою задачею.

Сумісною працею дослідників з Монреалю, Технічного Інституту Технологій Джорджії, Microsoft та Facebook було створено нейронну мережу, що може генерувати чутливі до контексту відповіді у розмові [https://arxiv.org/pdf/1506.06714v1.pdf]. Тренування системи проводилося на великому наборі неструктурованих діалогів у соціальній мережі Twitter. Архітектура рекурентної нейронної мережі використовується для створення відповідей на розріджені питання, що виникають під час інтегрування контекстної інформації у класичну статистичну модель. Створена модель показує істотне покращення результатів над контент-чутливою і контент-нечутливою базовою лінією машинного перекладу та пошуку інформації [1].

Нейронна машина для відповідей (NRM — Neural Responding Machine) — це розроблений у Гонконзі генератор відповідей для коротких розмов, створений із використанням спільного кодер-декодер фреймворка. Спочатку формалізується створення відповіді, як процес розшифрування на основі прихованого представлення вхідного тексту, у той час як кодування та декодування здійснюються за допомогою рекурретних нейронних мереж. Навчання NRM здійснювалося на великих об'ємах даних із однозначними діалогами, зібраними з мікроблогів. Емпірично встановлено, що NRM генерує граматично коректні та контекстуально доречні відповіді у 75% наданих для обробки розмов, що перевищує показники інших сучасних моделей подібної конфігурації [1].

Google's Neural Conversational Model — це модель, що пропонує підхід до моделювання діалогів на основі sequence-to-sequence фреймворку. Модель здатна вести розмову, висуваючи припущення стосовно наступної репліки. Припущення базуються на попередніх висловлюваннях. Модель здатна до наскрізного навчання, що зменшує потребу у створенні великої кількості штучних правил і обмежень. Прості діалоги можуть конструюватися на основі діалогового тренувального датасету, вузькоспеціалізованих датасетів та зашумлених датасетів із субтитрами до фільмів [3].

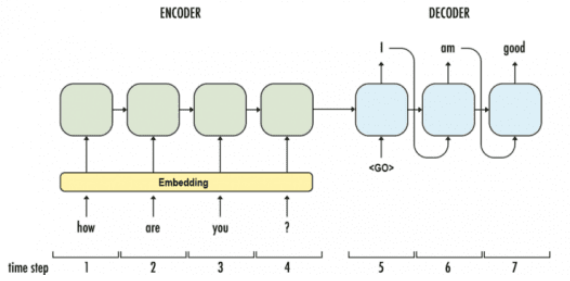

Рис. 4. Neural Responding Machine

VI. Системи типу «питання-відповідь» (QA)

Системи типу «питання-відповідь» добувають інформацію безпосередньо з джерела (документа, розмови, онлайн пошуку, набору картинок, тощо). Такі системи надають короткі і лаконічні відповіді замість розгорнутого тексту. Більшість NLP задач можна розглядати як задачі типу «питання-відповідь», де користувач надсилає запит (наприклад, через інтегрованого чат-бота) і отримує відповідь від системи [4].

Для вирішення QA задач існує спеціальна оптимізована архітектура глибинного навчання — Мережа Динамічної Пам'яті (Dynamic Memory Network, далі — DNM). DNM навчається на тренувальному наборі вхідних даних та питань і формує епізодичні «спогади» про них, які потім використовуються для генерації доречних відповідей. Архітектура DNM складається з наступних компонентів:

• Модуль семантичної пам'яті, аналогічний базі знань. Він складається із попередньо підготовлених GloVe векторів, що використовуються для створення послідовностей векторних представлень слів із вхідних висловлювань. Ці вектори будуть використані як вхідні дані моделі.

• Вхідний модуль переробляє зв'язані із питанням вхідні вектори в наборах векторів, що називаються фактами. Цей модуль реалізований за допомогою Керованого рекурентного блоку (Gated Recurrent Unit, далі — GRU), що дозволяє визначити релевантність розглянутого висловлювання.

• Модуль питань обробляє питання слово за словом, і генерує вектор із використанням GRU (аналогічно з вхідним модулем, і такими самими вагами).

• Модуль епізодичної пам'яті зберігає визначені на вході вектори фактів і питань, закодованих як вкладення.

• Модуль відповідей генерує доречну відповідь із епізодичної пам'яті, що містить необхідну для цього інформацію на останньому етапі. Цей модуль використовує інший GRU, натренований із класифікацією крос-ентропійної помилки коректної послідовності, що конвертується назад у природню мову.

DNM мають вищу ефективність у вирішенні QA задач порівняно з іншими архітектурами для семантичного аналізу та часткомовної розмітки (part-of-speech tagging).

VII. Стислий переказ тексту

Стислий переказ тексту (Text Summarization) — інструмент інтерпретації текстової інформації, що дозволяє створювати лаконічні резюме великих текстових фрагментів. Скорочення тексту формується у декілька етапів: підрахунок частоти появи слова в текстовому документі; визначення 100 найбільш частих слів; сортування визначених слів; оцінювання кожного речення із найбільш частими словами, із наданням більшого вагового коефіцієнту словам, що зустрічаються частіше; сортування перших X речень з урахуванням їхнього положення в оригінальному тексті [5].

Виділяють два основних підходи до скорочення тексту: видобувний та абстрактний. Видобувний підхід добуває слова та фрази з оригінального тексту для створення стислого перекладу. LexRank і TextRank є відомими представниками цього підходу, що використовують варіації алгоритму сортування сторінок Google PageRank.

LexRank - алгоритм навчання без вчителя на основі графів, який застосовує модифікований косинус зворотної частоти використання слова як міру схожості двох речень. LexRank запобігає тавтологій шляхом інтелектуальної післяобробки, що перевіряє головні речення переказу на низький коефіцієнт подібності.

TextRank схожий на LexRank, але містить низку переваг, а саме: використання лемматизації замість стеммінгу; застосування часткомовної розмітки та розпізнавання імені об'єкту; виділення важливих ключових фраз разом зі стислим викладом тексту.

Абстрактний підхід вивчає внутрішнє мовне представлення тексту, щоб створити подібну до людської інтерпретацію шляхом перефразування. Моделі для абстрактного стислого переказу тексту використовують глибинне навчання, що призвело до прориву у цій сфері.

Висновки

Моделі рекурсивного глибокого навчання можуть вирішувати безліч мовних завдань, що включають передбачення на рівні слова та речення як безперервного, так і дискретного характеру. Одним із багатьох можливих рішень для досягнення цієї мети є застосування рекурсивних або рекурентних методів для обчислення подань рівня абзацу або документа. Другий виклик для глибиних моделей - це логічні міркування першого порядку, які можуть знадобитися для отримання правильної інформації з баз знань за допомогою питань природної мови. Розглянуті моделі в цій дипломній роботі можуть бути розширені, щоб врешті-решт спільно моделювати мову, образи та бази знань в одній цілісній семантичній структурі.

Література References

[1] J. Le. (2018). "The 7 NLP Techniques That Will Change How You Communicate in the Future (Part I)" [Online]. Available: https://heartbeat.fritz.ai/the-7-nlp-techniques-that-will-change-how-you-communicate-in-the-future-part-i-f0114b2f0497

[2] M. Bates (1995). "Models of natural language understanding" [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC40721/

[3] R. Socher, "Recursive deep learning for natural language processing and computer vision" Stanford University, 2014, pp. 8-120.

[4] Afanasieva I. Data exchange model in the Internet of Things concept / I. Afanasieva, N. Golian, O. Hnatenko, Y. Daniiel, K. Onyshchenko // Telecommunications and Radio Engineering, New York, 2019. – 10(78). – p. 869-878

[5] Onyshchenko A. Adaptive method of training neural networks / A. Onyshchenko, K. Onyshchenko // Technique and technology. Science, Research, Development #29. Gdansk, 2020. – p. 9-11.