ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

УДК 004.85 https://doi.org/10.35546/kntu2078-4481.2023.1.15

К. О. АНТІПОВА
доктор філософії,
старший викладач кафедри інженерії програмного забезпечення
Чорноморський національний університет імені Петра Могили
ORCID: 0000-0002-9012-5290

ЗАСТОСУВАННЯ МЕХАНІЗМУ УВАГИ ТИПУ MULTI-HEAD ТА МОДЕЛІ ТРАНСФОРМЕРА ДЛЯ ЗАДАЧІ МАШИННОГО ПЕРЕКЛАДУ

Механізм уваги використовується в широкому діапазоні нейронних архітектур і досліджувався в різних областях застосування. Механізм уваги став популярною технікою глибокого навчання з кількох причин. По-перше, найсучасніші моделі, які включають механізми уваги, досягають високих результатів для різноманітних завдань, таких як класифікація тексту, створення підписів до зображень, аналіз настроїв, розпізнавання природної мови та машинний переклад. Використовуючи механізм уваги, нейронні архітектури можуть автоматично зважувати релевантність будь-якої області вхідного тексту та враховувати ці ваги під час вирішення основної задачі. Крім того, популярність механізмів уваги додатково підвищилася після появи моделі трансформера, яка ще раз довела, наскільки ефективним є механізм уваги. Архітектура трансформера не використовує послідовну обробку та рекурентність, а покладається лише на механізм self-attention, щоб охопити глобальні залежності між вхідними і вихідними послідовностями. В роботі використано модель трансформера, яка реалізує масштабовану скалярнодобуткову увагу, що відповідає процедурі механізму загальної уваги. Побудована модель спирається на механізм уваги типу multi-head attention, де модуль self-attention повторює обчислення декілька разів паралельно. Ці розрахунки об'єднуються для отримання остаточної оцінки. Застосування multi-head attention дає моделі більше можливостей для кодування декількох зв'язків і нюансів для кожного слова. Завдяки використанню механізму multi-head attention функція уваги отримує інформацію з різних частин представлення, що неможливо при використанні self-attention. Модель трансформера була реалізована за допомогою фреймворків TensorFlow та Keras для задачі машинного перекладу з англійської на українську. Набір даних для тренування, валідації та тестування моделі був отриманий від Tatoeba Project. Був реалізований власний шар для вбудовування слів із використанням матриці позиційного кодування.

Ключові слова: *механізм уваги, машинний переклад, обробка природної мови, модель трансформера.*

K. O. ANTIPOVA
Ph.D., Senior Lecturer at the Department of Software Engineering
Petro Mohyla Black Sea National University
ORCID: 0000-0002-9012-5290

IMPLEMENTATION OF THE MULTI-HEAD ATTENTION MECHANISM AND TRANSFORMER MODEL FOR MACHINE TRANSLATION

The attention mechanism is used in a wide range of neural architectures and has been researched within diverse application domains. The attention mechanism has become a popular deep learning technique for several reasons. First, state-of-the-art models that incorporate attention mechanisms achieve high results for a variety of tasks such as text classification, image captioning, sentiment analysis, natural language recognition, and machine translation. Using an attention mechanism, neural architectures can automatically weight the relevance of any region of input text and take those weights into account when solving the underlying problem. In addition, the popularity of attention mechanisms further increased with the implementation of the transformer model, which once again proved how effective the attention mechanism is. The transformer architecture does not use sequential processing or recursion, but relies only on the self-attention mechanism to capture global dependencies between input and output sequences. In the paper a transformer model that implements scaled scalar product attention was used, which corresponds to the procedure of the general attention mechanism. The built model is based on the multi-head attention mechanism, where the self-attention module repeats the calculation several times in parallel. These calculations are combined to get a final estimate. Applying multi-head attention gives the model more opportunities to encode multiple connections and nuances for each word. The application of the multi-head attention mechanism allows the attention function to obtain information from different parts of the representation, which is impossible when using self-attention only. The transformer model was implemented using the TensorFlow and Keras frameworks for the task of machine translation from English to Ukrainian. The dataset for model training, validation, and testing was obtained from the Tatoeba Project. A custom word embedding layer was implemented using a positional encoding matrix.

Key words: *attention mechanism, machine translation, natural language processing, transformer model.*

Постановка проблеми

У задачах обробки природної мови елементи вхідного тексту характеризуються тим, що мають різну значимість в рамках поставленого завдання. У машинному перекладі деякі слова вхідного тексту можуть бути нерелевантними для перекладу наступного слова в послідовності. Найпоширенішим рішенням цієї задачі є механізм, відомий як механізм уваги. Використовуючи цей механізм, нейронні архітектури можуть автоматично зважувати релевантність будь-якої області вхідного тексту та враховувати ці ваги під час вирішення основної задачі.

Окрім збільшення продуктивності виконання подібних задач, механізм уваги також можна використовувати як інструмент для інтерпретації поведінки нейронних архітектур. Наприклад, ваги, обчислені за допомогою механізму уваги, можуть вказати на релевантну інформацію, відкинуту нейронною мережею, або на нерелевантні елементи вхідних даних, які були враховані, і могли б пояснити несподіваний результат, отриманий від нейронної мережі.

Механізм уваги було вперше запропоновано для вирішення проблеми, яка виникає при використанні вектора кодування з фіксованою довжиною, коли декодувальник має обмежений доступ до інформації, яку можна отримати від вхідних даних. Цей недолік особливо ускладнює обробку довгих та/або складних послідовностей, де розмірність їх представлення має бути такою ж, як для коротших або простіших послідовностей.

У випадках, коли механізм загальної уваги представлений послідовністю слів, вектор запиту, який заданий для певного слова в послідовності, оцінюється відносно кожного ключа у базі даних. Таким чином визначається, як слово, що розглядається, пов'язане з іншими в послідовності. Після чого отримані значення масштабуються відповідно до ваг уваги (обчислених на основі отриманих раніше оцінок запит-ключ), щоб зберегти фокус на тих словах, які стосуються запиту. Таким чином розраховується рівень уваги для слова, яке розглядається.

Аналіз останніх досліджень і публікацій

В роботі [1] реалізовано рекурентну нейронну мережу (RNN) як для кодувальника, так і для декодувальника. Однак механізм уваги можна перетворити в узагальнену форму, яку можна застосувати до будь-якого завдання типу seq2seq (послідовності-до-послідовності), де дані можуть бути пов'язані між собою не обов'язково послідовним чином.

В роботі [2] в якості альтернативи представлена модель трансформера, яка спирається виключно на використання механізму уваги типу self-attention, де представлення послідовності (або речення) обчислюється шляхом зв'язування різних слів в одній послідовності. В цій роботі також запропоновано механізм уваги типу multi-head. Такий механізм уваги лінійно проектує запити, ключі та значення *h* разів, щоразу використовуючи іншу проекцію, отриману в результаті навчання. Механізм self-attention потім застосовується до кожної з цих проекцій *h* паралельно для отримання результатів, які, у свою чергу, об'єднуються для отримання кінцевої проекції.

В роботі [3] розглянуто архітектури уваги, які призначені для роботи з векторними представленнями текстових даних. Крім того запропонована класифікація моделей уваги відповідно до чотирьох вимірів: представлення вхідних даних, функція сумісності, функція розподілу та множинність вхідних і/або вихідних даних.

В роботі [4] систематизовано різні моделі механізму уваги, які використовуються для широкого кола задач, в тому числі задач обробки природної мови. В роботі розглянуто ключові моменти, як включення уваги до нейронних мереж призвело до значного підвищення продуктивності, забезпечило краще розуміння внутрішньої роботи нейронної мережі за рахунок полегшення інтерпретації, а також покращило обчислювальну ефективність шляхом усунення послідовної обробки вхідних даних.

В роботі [5] представлена комплексна класифікація різноманітних механізмів уваги на основі структури моделі задачі, яка складається з моделі ознак, моделі уваги, моделі запиту та моделі вихідної послідовності. Загалом в роботі розглядалися лише механізми уваги для моделей, які навчаються з учителем, оскільки вони складають найбільшу частку моделей уваги серед тих, що зазвичай досліджуються.

Мета дослідження

Підвищення продуктивності машинного перекладу речень з англійської на українську шляхом реалізації та тренування моделі трансформера на основі механізму multi-head attention із використанням власного шару вбудовування слів.

Виклад основного матеріалу

Механізм загальної уваги використовує три основні компоненти, а саме запити *Q*, ключі *K* і значення *V*. Якщо провести аналогію між цими трьома компонентами з механізмом уваги, запропонованим в роботі [1], то запит *Q* буде аналогічним попереднім вихідним даним декодувальника, s_{t-1}, тоді як значення *V* будуть аналогічними попереднім вхідним даним кодувальника, h_i. У механізмі уваги, представленому в [1], ключі та значення є одним вектором.

Механізм загальної уваги виконує наступні обчислення:

1. Кожен вектор запиту $q = s_{t-1}$ зіставляється з базою даних ключів для обчислення оцінки. Ця операція зіставлення обчислюється як скалярний добуток запиту, який розглядається, з кожним вектором-ключем k_i:

$$e_{q,k_i} = q \cdot k_i.$$

2. Для оцінок застосовується операція *softmax* для створення вагових коефіцієнтів:

$$\alpha_{q,k_i} = softmax\,(e_{q,k_i}).$$

3. Загальна увага обчислюється за допомогою зваженої суми векторів значень v_{k_i}, де кожен вектор значень поєднується з відповідним ключем:

$$attention(q,K,V) = \sum_i \alpha_{q,k_i} v_{k_i}.$$

У контексті машинного перекладу кожному слову у вхідному реченні присвоєно вектори для запиту, ключа та значення. Ці вектори утворюються шляхом множення отриманого від кодувальника представлення конкретного слова на три різні вагові матриці, які генеруються під час навчання.

Структура моделі трансформера повторює структуру кодувальника-декодувальника, але не використовує рекурентність або згортки для генерування вихідної послідовності. Трансформер використовує наступні основні компоненти:

– q, k – це вектори розмірності d_k, що містять запити та ключі відповідно;
– v – це вектор розмірності d_v, що містить значення;
– Q, K, V – це матриці, що складаються із запитів, ключів і значень відповідно;
– W^Q, W^K, W^V - це матриці проекцій, які використовуються для генерації різних представлень підпростору запиту, ключа та матриці значень;
– W^O - це проекційна матриця вихідної послідовності multi-head.

По суті, функцію уваги можна вважати функцією перетворення запиту і набору пар ключ-значення на вихідну послідовність.

Модель трансформера реалізує масштабовану скалярнодобуткову увагу, яка відповідає процедурі механізму загальної уваги. На практиці ці обчислення можна ефективно застосовувати до всього набору запитів одночасно. Для цього матриці Q, K і V подаються як вхідні дані для функції уваги наступним чином:

$$attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V.$$

Коефіцієнт масштабування $\frac{1}{\sqrt{d_k}}$ потрібен для нейтралізації ефекту збільшення величини скалярних добутків для великих значень d_k, де при застосуванні функції *softmax* отримуються надзвичайно малі градієнти, що призводить до проблеми зникання градієнту. Отже, коефіцієнт масштабування потрібен для зменшення отриманих результатів і, відповідно, для запобігання цій проблемі.

Ідея, що лежить в основі механізму уваги типу multi-head, полягає в тому, щоб дозволити функції уваги отримувати інформацію з різних частин представлення, що неможливо при використанні self-attention. Функцію уваги multi-head можна представити таким чином:

$$multihead(Q,K,V) = concat(head_1,...,head_h)\,W^O.$$

де кожен $head_i$, $i = 1,...,h$ реалізує єдину функцію уваги, що характеризується власними матрицями проекції, отриманими в результаті навчання:

$$head_i = attention(QW_i^Q, KW_i^K, VW_i^V).$$

Отже, кожен блок multi-head складається з таких послідовних рівнів:

– На першому рівні – три лінійних шари, кожен з яких отримує запити, ключі або значення.
– На другому рівні – масштабована скалярнодобуткова увага. Операції, що виконуються як на першому, так і на другому рівнях, повторюються h разів і виконуються паралельно, відповідно до кількості $head_i$, що складають блок уваги multi-head.
– На третьому рівні – операція конкатенації, яка об'єднує вихідні послідовності від різних $head_i$.
– На четвертому рівні – кінцевий лінійний шар, який утворює вихідну послідовність.

Модель трансформера, яка спирається на механізм уваги типу multi-head attention, була реалізована за допомогою фреймворків TensorFlow та Keras. Модель побудована на основі алгоритмів, представлених в роботах [6, 7, 8]. В якості вхідних даних виступає набір з речень англійською та відповідних речень українською, отриманий від Tatoeba Project [9]. Набір складається з 156733 пар речень.

Набір вхідних даних був очищений та нормалізований до форми NFKC. Підготовлені речення були токенізовані по словах: кожне слово та кожен розділовий знак були виділені в окремі токени максимум по 36 токенів в реченні (рис. 1). В результаті отримано 10795 токенів для словника англійської та 31296 токенів для словника українською. Для того, щоб відкинути токени, які рідко зустрічаються, розміри словників були обмежені до

10000 та 30000 токенів відповідно. Також максимальна можлива кількість токенів у послідовності була обмежена до 30.

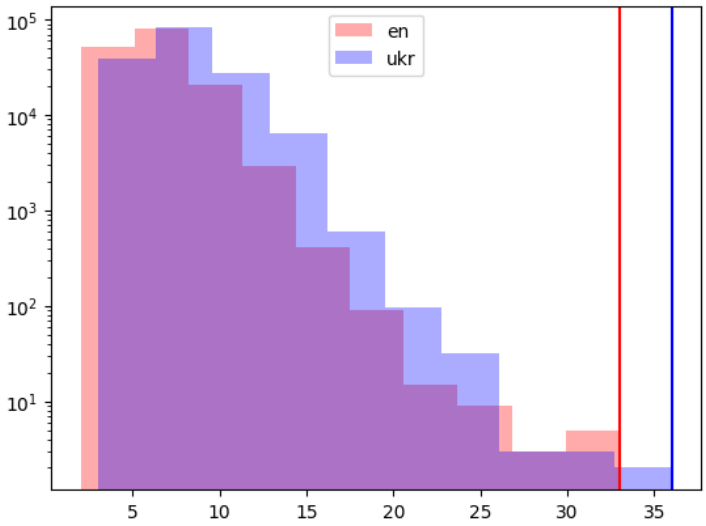

Рис. 1. Кількість токенів в реченнях різними мовами

Кожне речення було перетворено у вектор для подальшого використання при навчанні моделі. Для того, щоб модель «розуміла» значення слів, тобто, кількісно визначала, як два слова пов'язані одне з одним, використаний власний шар вбудовування слів (embedding). Для розуміння контексту визначається позиція кожного слова в реченні за допомогою матриці позиційного кодування.

Позиційне кодування представляє позицію кожного токена вектором. Елементами вектора є значення різних фаз і частот синусоїд. У позиції $k = 0,1,...,L-1$ позиційний вектор кодування (з вхідною розмірністю L та вихідною розмірністю d) розраховується наступним чином:

$$[P(k,0), P(k,1),..., P(k,d-2), P(k,d-1)],$$

$$P(k,2i) = \sin(\frac{k}{n^{2i/d}}),$$

$$P(k,2i+1) = \cos(\frac{k}{n^{2i/d}}),$$

де $i = 0,1,...,d/2$;

n = 10000 – константа для синусоїдальних функцій.

Шар позиційного кодування знаходиться на вході моделі трансформера та поєднує шар вбудовування слів з позиційним кодуванням. Для навчання моделі трансформера було створено 8 блоків multi-head, а розмірність вихідного вектора встановлена на 128. Використовувався оптимізатор Adam (швидкість навчання моделі 1e-09). Модель навчалася протягом 20 епох, і в результаті навчання точність моделі становить 0.8141, значення функції витрат – 1.1967.

На тестовому наборі (15% від загального набору речень) модель видає наступні результати:

```
is that all you can think of ?
== [start] це все, що тобі спадає на думку ? [end]
-> [start] це все, що ти можеш подумати ? [end]
i need someone to help me with this.
== [start] мені потрібно, щоб хтось мені з цим допоміг. [end]
-> [start] мені потрібно, щоб мені хтось допоміг з цим. [end]
i had my watch fixed.
== [start] я відремонтувала свій годинник. [end]
-> [start] я полагодив годинник. [end]
find out all you can about him.
== [start] зберіть про нього будь-яку інформацію, яку знайдете. [end]
```

```
-> [start] дізнайся про нього все, що можеш. [end]
she had her hat blown off by the wind.
== [start] їй здуло вітром капелюха. [end]
-> [start] вона мала, коли її з неї на нього були це. [end]
i'm not a native speaker.
== [start] це для мене не рідна мова. [end]
-> [start] я не носій мови. [end]
```

де перший рядок – це речення англійською для перекладу, другий рядок – відповідне речення українською з початкового набору даних, третій рядок – результат отриманий від моделі трансформера.

Висновки

Була побудована модель трансформера, яка реалізує масштабовану скалярнодобуткову увагу, що відповідає процедурі механізму загальної уваги. Модель спирається на механізм уваги типу multi-head attention. Був реалізований власний шар для вбудовування слів із використанням матриці позиційного кодування. Після тренування точність моделі складає 81,4%.

В подальшому планується дослідити вплив застосування різних токенизаторів, в тому числі токенизаторів бібліотеки NLTK, на результуючу точність моделі. Крім того, для покращення якості вбудовування планується використати сторонні алгоритми замість власного шару вбудовування слів. Зокрема зробити порівняльний аналіз, наскільки подібні алгоритми покращують розуміння контексту для моделі.

Список використаної літератури

1. Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural Machine Translation by Jointly Learning to Align and Translate. *International Conference on Learning Representations*. https://doi.org/10.48550/arXiv.1409.0473
2. Vaswani, A. et al. (2017). Attention Is All You Need. *31st Conference on Neural Information Processing Systems (NIPS)*. https://doi.org/10.48550/arXiv.1706.03762
3. Galassi, A., Lippi, M., & Torroni, P. (2021). Attention in Natural Language Processing. *IEEE Transactions On Neural Networks And Learning Systems*, *Vol. 32, No. 10*. https://doi.org/10.1109/TNNLS.2020.3019893
4. Chaudhari, Sh., Mithal, V., Polatkan, G., & Ramanath, R. (2021). An Attentive Survey of Attention Models. *ACM Transactions on Intelligent Systems and Technology, Vol. 1, No. 1*. https://doi.org/10.1145/3465055
5. Brauwers, G., & Frasincar, F. (2021). A General Survey on Attention Mechanisms in Deep Learning. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*. https://doi.org/10.1109/TKDE.2021.3126456
6. Cristina, S., & Saeed, M. (2022). *Building Transformer Models with Attention: Implementing a Neural Machine Translator from Scratch in Keras.* Machine Learning Mastery.
7. Rothman, D. (2022). *Transformers for Natural Language Processing: Build, train, and fine-tune deep neural network architectures for NLP with Python, PyTorch, TensorFlow, BERT, and GPT-3, 2nd Edition*. Packt Publishing.
8. Yıldırım, S., & Asgari-Chenaghlu, M. (2021). *Mastering Transformers: Build state-of-the-art models from scratch with advanced natural language processing techniques*. Packt Publishing.
9. Tatoeba Project. (n.d.). http://tatoeba.org/home