ScaleGrad - manipulowanie gradientami funkcji straty w celu uzyskania bardziej oryginalnych tokenów w zadaniach generacji tekstu

Kornelia Staszewska[1], Ewa Komkowska[1]

I. Definicja problemu badawczego

Tematem projektu będzie reimplemetacja metody ScaleGrad polegającej na ingerencji w funkcję straty wykorzystywaną w treningu modelu, tak aby skłonić model do wykorzystywania bardziej oryginalnych i mniej powtarzalnych tokenów. Zrealizujemy to poprzez zwiększenie prawdopodobieństawa dla tokenów oryginalnych i odpowiednio zmniejszenie dla powtarzalnych. W ramach prac zostaną przeprowadzone eksperymenty dotyczące działania metody SG w porównaniu do metod wykorzystujących Maximal Likelihood Estimation (MLE) lub unlikelihood (UL) dla kilku pretrenowanych modeli neuronowych i zbiorów treningowych dla odpowiednio zadań zwiazanych z tzw. directed i open-ended text generation. W ramach eksperymentu chcemy sprawdzić czy ScaleGrad pozwala na generowanie tekstów osiągających lepsze wartości metryk automatycznych oraz bardziej podobnych do tekstów pisanych przez człowieka.

II. Przegląd literatury

Problem, który chciałybyśmy zbadać w ramach projektu został już poruszony w artykule przedstawionym na konferencji ICML 2021. Jego tytuł brzmi "Straight to the Gradient: Learning to Use Novel Tokens for Neural Text Generation" [1]. Chciałybyśmy zreprodukować badania nad miarą ScaleGrad i porównać uzyskane wyniki z tymi przedstawionymi w przywołanym źródle. Dodatkowo, podobnie jak autorzy wspomnianego artykułu, odniesiemy się do pracy dotyczacej UL [2] oraz prac związanych z wykorzystywanymi modelami neuronowymi (GPT-2 [3], model proposed by Melas-Kyriazi et al. (2018) [4], BertSum [5]) i zbiorami danych (Wikitext-103 [6], 50 BPE [7], PTB [8], IMDB [9], Visual Genome [10], CNN/DM [11], NYT50 [12]). Kod źródłowy przedstawiony przez autorów znajduje się na publicznym repozytorium na Githubie [13].

III. Wykorzystane technologie i narzędzia

W projekcie zastosujemy podejście oparte o podział pracy na zadania związane z dwoma analizowanymi kategoriami: open-ended generation i directed generation. Dla każdej z nich przeprowadzimy eksperymenty analogiczne jak w artykule, zatem wymagane będzie zaczytanie odpowiednich modeli i zbiorów danych, konfiguracja parametrów zgodnie z wytycznymi przedstawionymi w pracy dodatkowy trening modeli uwaględniający aplikację metody SG oraz końcowo wykonanie pomiarów dla metryk wylistowanych przez autorów. Co istotne eksperymenty dotyczące tzw. human evaluation zostaną przeprowadzone na mniejszej grupie osób (od 2 do 4). Językiem programowania, którego użyjemy będzie Python, a w ramach projektu planujemy wykorzystać biblioteki PyTorch i NumPy oraz narzędzie Fairseq(-py) pozwalające między innymi na wytrenowanie własnych modeli służących zadaniom związanym z modelowaniem sekwencji.

[1]Wydział Informatyki i Telekomunikacji, Politechnika Poznańska, Poznań {kornelia.staszewska, ewa.komkowska}@put.poznan.pl

IV. Wymagania projektu

A. Wymagania minimalne

Do wymagań minimalnych należy:

- przygotowanie zbiorów treningowych
- przygotowanie modeli neuronowych i wytrenowanie ich z użyciem metody ScaleGrad
- reimplementacja metody ScaleGrad
- przeprowadzenie eksperymentów (odpowiednio dla kategorii open-ended generation i directed generation - w tym image paragraph captioning oraz abstractive text summarization) i pomiar odpowiednich wielkości (w tym między innymi: perplexity, Rep/l, 'uniq', rep-n, uniq-w, CIDEr czy WMD-1; dla poszczególnych zadań odpowiednie miary)

B. Wymagania końcowe

Wymagania końcowe obejmują porównanie wyników uzyskanych metodą SG z tymi uzyskanymi przez MLE czy UL oraz weryfikację czy ScaleGrad pozwala na generowanie tekstów zawierających mniej powtórzeń oraz czy są one bardziej oryginalne i zbliżone do tych pisanych przez człowieka.

V. Zagrożenia projektu

- problem w zrozumieniu matematycznych podstaw zaproponowanych koncepcji - poszukanie obszerniejszych materiałów na temat danego zagadnienia
- nieznajomość wykorzystanych w pracy modeli - poszukanie obszerniejszych materiałów na temat danego zagadnienia
- trudność z wczytaniem zbiorów danych oraz modeli (ograniczenia pamięciowe) - wykorzystanie systemów chmurowych
- problemy dotyczące zbyt długiego czasu treningu analizowanych modeli - ograniczenie zakresu eksperymentu

- zbyt szeroki zakres ekperymentów - zawężenie go do elementów pozwalających na sprawdzenie głównej hipotezy.

Bibliografia

[1] X. Lin, S. Han, and S. Joty, "Straight to the gradient: Learning to use novel tokens for neural text generation," in Proceedings of the 38th International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 6642–6653. [Online]. Available: https://proceedings.mlr.press/v139/lin21b.html

[2] S. Welleck, I. Kulikov, S. Roller, E. Dinan, K. Cho, and J. Weston, "Neural text generation with unlikelihood training," in International Conference on Learning Representations, 2020. [Online]. Available: https://openreview.net/forum?id=SJeYe0NtvH

[3] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language Models are Unsupervised Multitask Learners," 2019. [Online]. Available: https://openai.com/blog/better-language-models/

[4] L. Melas-Kyriazi, A. Rush, and G. Han, "Training for diversity in image paragraph captioning," in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 757–761. [Online]. Available: https://aclanthology.org/D18-1084

[5] Y. Liu and M. Lapata, "Text summarization with pretrained encoders," in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 3730–3740. [Online]. Available: https://aclanthology.org/D19-1387

[6] S. Merity, C. Xiong, J. Bradbury, and R. Socher, "Pointer sentinel mixture models," CoRR, vol. abs/1609.07843, 2016. [Online]. Available: http://arxiv.org/abs/1609.07843

[7] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1715–1725. [Online]. Available: https://aclanthology.org/P16-1162

[8] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz, "Building a large annotated corpus of English: The Penn Treebank," Computational Linguistics, vol. 19, no. 2, pp. 313–330, 1993. [Online]. Available: https://aclanthology.org/J93-2004

[9] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Portland, Oregon, USA: Association for Computational Linguistics, Jun. 2011, pp. 142–150. [Online]. Available: https://aclanthology.org/P11-1015

[10] J. Krause, J. Johnson, R. Krishna, and L. Fei-Fei, "A hierarchical approach for generating descriptive image paragraphs," CoRR, vol. abs/1611.06607, 2016. [Online]. Available: http://arxiv.org/abs/1611.06607

[11] R. Nallapati, B. Zhou, C. dos Santos, Gulehrelar, and B. Xiang, "Abstractive text summarization using sequence-to-sequence RNNs and beyond," in Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 280–290. [Online]. Available: https://aclanthology.org/K16-1028

[12] G. Durrett, T. Berg-Kirkpatrick, and D. Klein, "Learning-based single-document summarization with compression and anaphoricity constraints," in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1998–2008. [Online]. Available: https://aclanthology.org/P16-1188

[13] "Github repository." [Online]. Available: https://github.com/shawnlimn/ScaleGrad