Efektívne modelovanie cez viacero atribútov

Bc. Samuel Revúcky
školiteľ: Mgr. Marek Šuppa

1 Úvod

Napriek tomu, že naša téma zatiaľ postráda presné kontúry, pokúsime sa v tomto dokumente približne sprostredkovať čo ideme robiť a aká je momentálna situácia v danej oblasti. V práci budeme pracovať s predtrénovanými jazykovými modelmi, takzvanými transformermi, a kompaktnými modulmi na ladenie takýchto veľkých jazykových modelov, adaptérmi.

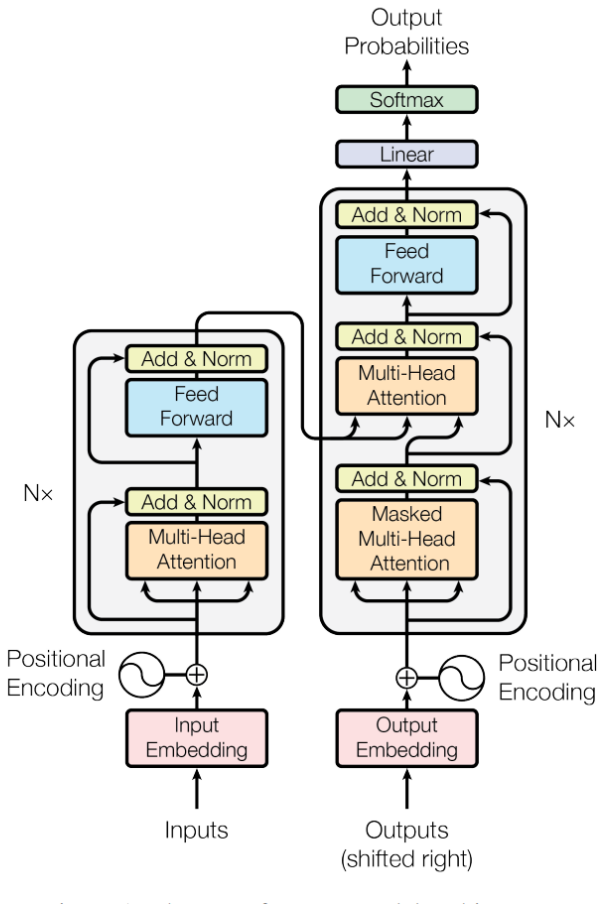
2 Transformery

Transformery odštartovali éru strojového učenia ako ho poznáme dnes. Začalo to v roku 2017 článkom s názvom 'Attention Is All You Need' [7]. Predtým boli dominantnými modelmi na spracovanie sekvencií prevažne rekurentné neurónové siete (RNN). Rekurencia z názvu funguje nasledovne. Dlhé sekvencie sa spracovávajú iteratívne po častiach (tokenoch), pričom si chceme zachovať informáciu o tom, aký bol výsledok spracovania v predošlých iteráciách. Napríklad pri preklade vety si chceme pre každé slovo pamätať kontext, v ktorom sa nachádza. RNN preto majú v každej iterácii na vstupe nový token a výsledok spracovania predošlého tokenu. Tento prístup má však dve veľké nevýhody. Model zabúda, teda informácia o výsledku ľubovoľnej iterácie sa v procese spracovania s pribúdajúcimi iteráciami stráca. Taktiež výpočet tohto modelu je veľmi zdĺhavý, čo vyplýva z toho, že model je inherentne sekvenčný a nemôže naraz spracovávať celý vstup.

Transformery riešia tento problém a prichádzajú s novou architektúrou, ktorá zahadzuje rekurentné vzťahy a spolieha sa na mechanizmus takzvanej self-attention, viď obr. 1. Model sa skladá z dvoch častí, kódera a dekódera. Kóder prevedie vstupné tokeny $\mathbf{x} = (x_1, \ldots, x_n)$ na sériu skrytých reprezentácií $\mathbf{z} = (z_1, \ldots, z_n)$, ktoré zachytávajú kontextové informácie vstupnej sekvencie. Dekóder na základe $\mathbf{z}$ postupne generuje výstupnú sekvenciu, pričom každý vygenerovaný token ovplyvňuje generovanie nasledujúcich tokenov.

2.1 Kóder

Kóder sa skladá z N identických vrstiev, pričom každá z nich obsahuje dve pod-vrstvy. Prvou je Multi-Head Attention a druhou je typická Feed Forward vrstva, čo je najjednoduchší model neurónovej siete, známy aj pod názvom Multi Layer Perceptron.



Obr. 1: Architektúra transformera

2.2 Dekóder

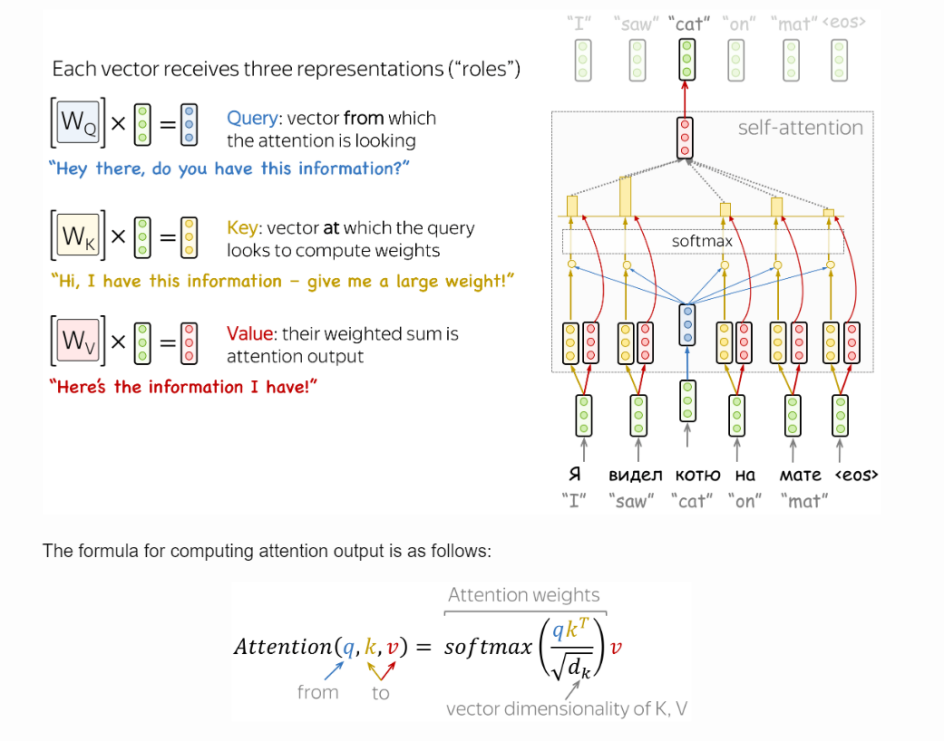
Dekóder sa tiež skladá z N identických vrstiev naskladaných na seba. Oproti kóderu je v nich pridaná ešte jedna pod-vrstva Multi-Head Attention nad výstupom kódera. Navyše prvá attention vrstva je maskovaná, čo zabezpečuje, aby sa tokeny nemohli pozerať na nasledujúce tokeny.

2.3 Attention

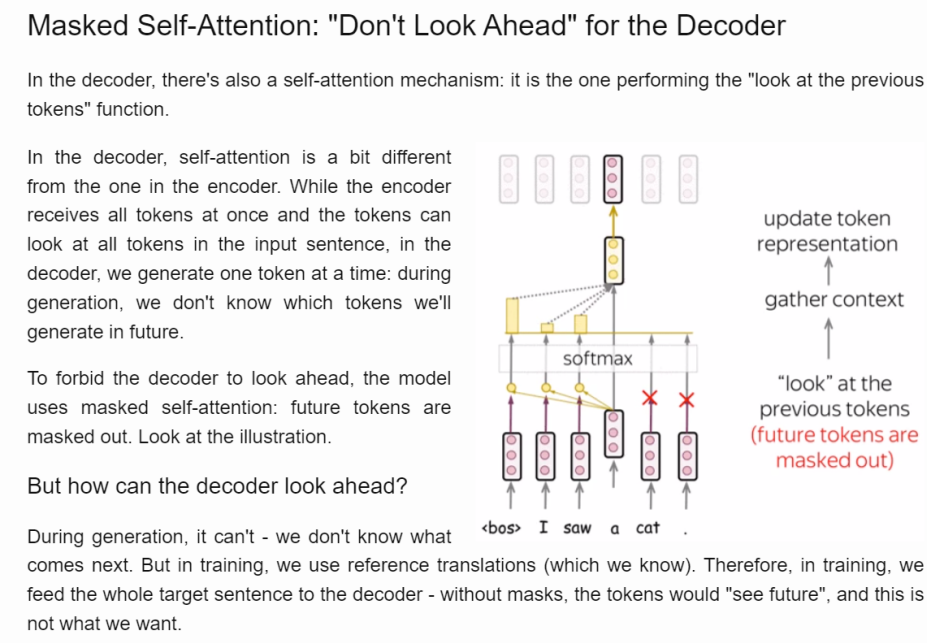
Attention je časťou modelu, kde tokeny vzájomne interagujú. Každý token pozoruje iné tokeny vo vete, zhromažďuje kontext a aktualizuje predchádzajúcu reprezentáciu seba. Názorná ukážka je na obr. 2. Ďalej prikladáme bližšie vysvetlenia mechanizmov maskovania a multi-head na obr. 3 a obr. 4 (všetky dostupné na https://lena-voita.github.io/nlp_course/seq2seq_and_attention.html).

3 Adaptéry

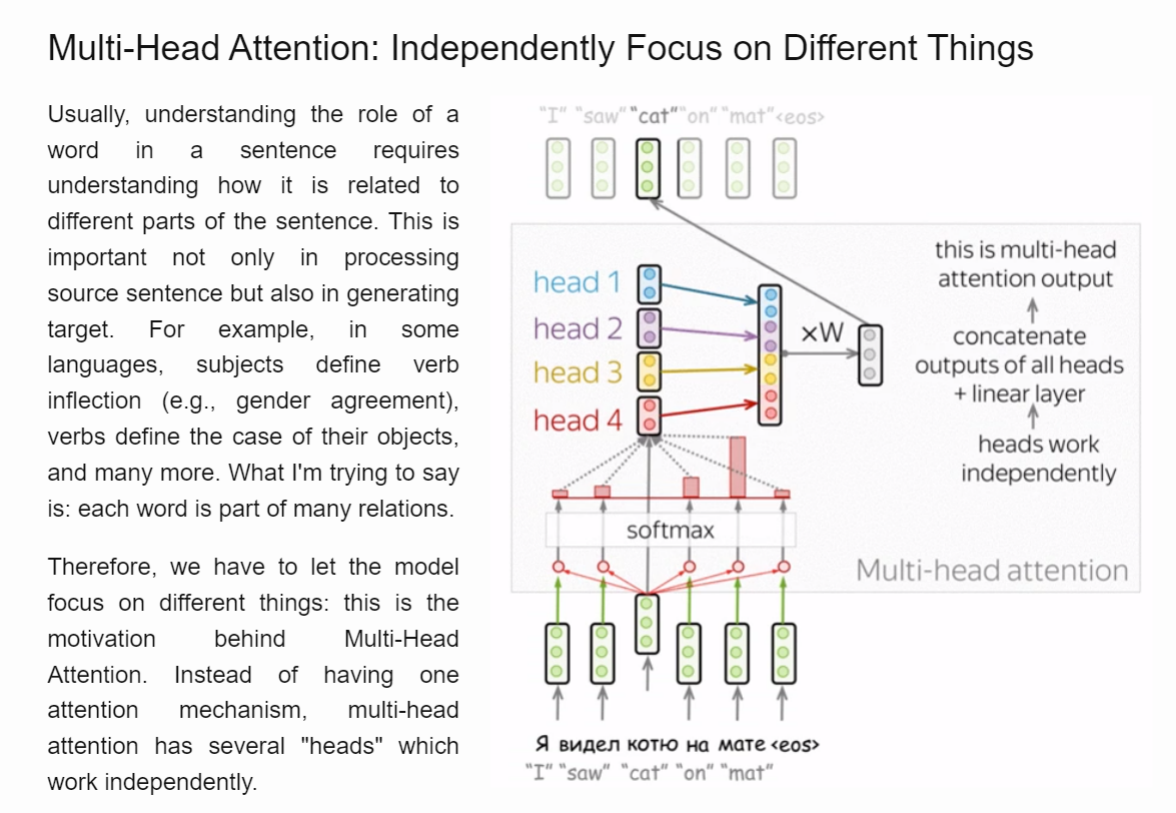
Použitie pred-trénovaných modelov vykazuje výnimočné výsledky v množstve problémov v oblasti spracovania prirodzeného jazyka (NLP). Typicky sa na toto používajú dve



Obr. 2: Znázornenie fungovania attention



Obr. 3: Mechanizmus maskovania

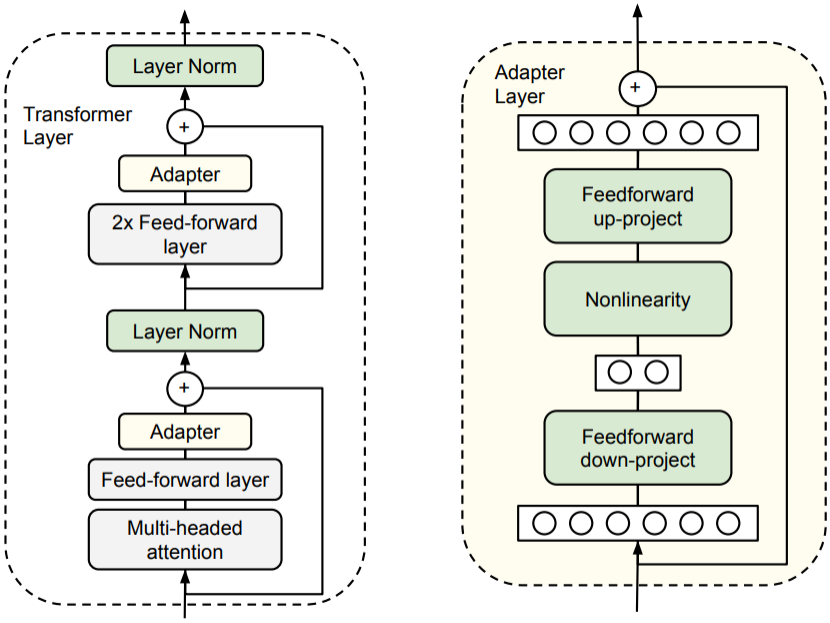


Obr. 4: Mechanizmus multi-head

metódy: feature-based a fine-tuning. Prvá z nich využíva vrstvy pred-trénovaného modelu na extrahovanie vlastností všeobecného charakteru z dát. Tieto vrstvy použije na dáta pre daný problém, na ktorých sa následne natrénuje zbrusu nový model. Druhá metóda zoberie pred-trénovaný model a ďalej ho trénuje na nových dátach pre daný problém. Obe tieto metódy však prinášajú veľkú výpočtovú záťaž, vzhľadom na veľkosti pred-trénovaných modelov, čo je pri veľkom množstve úloh, na ktoré by sme pred-trénované modely chceli využívať, značne nevyhovujúce.

Houlsby et. al. [3] navrhujú trénovanie pomocou adaptérov. Adaptéry sú moduly vložené medzi existujúce vrstvy pred-trénovaného modelu. Predstavujú novú sadu parametrov, rádovo menšej mohutnosti ako parametre pôvodného modelu. Pri trénovaní sa upravujú len váhy týchto parametrov, pôvodné ostávajú zachované. Na začiatku trénovania sa váhy inicializujú na takmer identickú funkciu, aby sa pôvodný model na začiatku trénovania príliš nerozptýlil. Architektúra adaptéru je vo forme bottlenecku, kde sa pôvodné d-dimenzionálne vektory projektujú do menšej dimenzie m, aplikuje sa nelineárna funkcia a následne projekcia späť do d dimenzií. Ukážka je na obr. 5. V štúdií sa vykonalo značné množstvo testov, ktoré ukázali, že aj keď pri použití adaptérov sa trénujú rádovo jednotky percent parametrov pôvodného modelu, dosahujú sa porovnateľné výsledky, ako pri fine-tuningu celého pred-trénovaného modelu.

Rücklé et. al. [6] ukazujú, že úspornosť adaptérov sa neprejavuje len v objeme parametrov a váh, ale aj v rýchlosti trénovania. Experimenty ukázali, že trénovanie modelu na novú úlohu s použitím adaptérov je rýchlejšie až o 60%, ako fine-tuning pred-trénovaného modelu. Toto zrýchlenie má za cenu spomalenie pri inferencii o 4-6%, čo



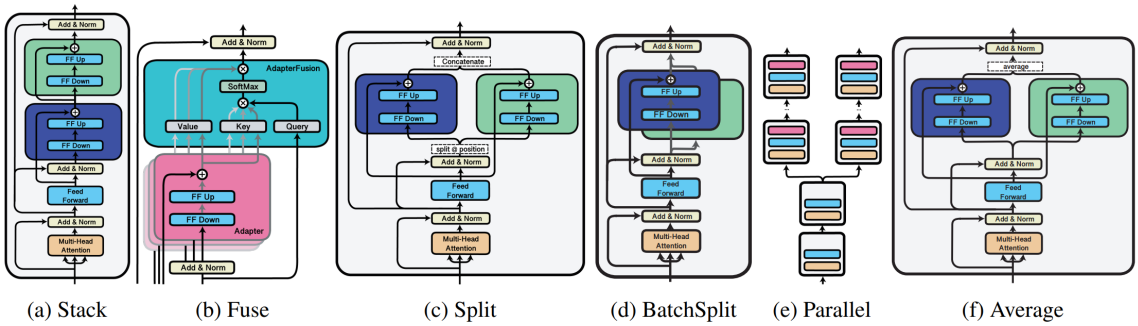
Obr. 5: Ukážka architektúry adaptéru od Houlsbyho

je však prijateľné v aplikáciach, kde je prioritou rýchlosť trénovania. Taktiež predkladajú AdapterDrop: dynamické odstraňovanie adaptérov z nižších vrstiev transformera, kde nižšie vrstvy často majú malý vplyv na výkon. Skúmajú dve trénovacie metódy: (1) špecializovaná, kde sa odstraňuje z fixného počtu vrstiev n, čo vytvára separátny model pre každé prípustné n (2) robustná, kde sa n volí náhodne z intervalu [0,11] pre každú trénovaciu skupinu, čo vytvára jeden robustný model, ktorý je odolný voči odstráneniu variabilného počtu vrstiev. Tieto metódy zachovávajú porovnateľné výsledky ako obyčajné použitie adaptérov pri zrýchlení inferencie o 21-42%.

3.1 Architektúry a skladania adaptérov

Poth et. al. [5] prezentovali knižnicu na jednoduché použitie adaptérov. Na rozdiel od iných podobných projektov sa okrem parametrickej efektivity sústredili aj na modularitu adaptérov. Ich knižnica poskytuje jednotné rozhranie na široké spektrum rôznych adaptérových konfigurácií. V čase vydania citovanej práce poskytovala knižnica nasledujúce kompozície: stack, fuse, split, batch split, parallel, average. Znázornenie logiky jednotlivých kompozícií je na obr. 6.

- **Stack** - Umožňuje naskladanie viacerých adaptérov na seba. Dáta sekvenčne prechádzajú týmto blokom v štýle výstup prvého adaptéru je vstup druhého atď.
- **Fuse** - Umožňuje aktiváciu AdapterFusion [4]. AdapterFusion je metóda na kombinovanie znalostí viacerých pred-trénovaných adaptérov na nové úlohy.
- **Split** - Umožňuje rozdeliť vstupné sekvencie do rôznych adaptérov. Napríklad pri multimodálnych dátach môžeme textovú časť poslať do iného adaptéru ako vizuálnu časť.



Obr. 6: Ukážky jednotlivých kompozícií adaptérov v knižnici *Adapters*

- **BatchSplit** - Umožňuje rozdeliť skupinu dát na vstupe na viacero pod-skupín, pričom každá z nich poputuje do iného adaptéra.
- **Parallel** - Umožňuje nezávislé paralelné trénovanie a inferenciu na rôznych adaptéroch, pričom každý adaptér má vlastnú predikčnú hlavu. Implementácia automaticky replikuje všetky vstupy do paralelných modulov adaptéra, zdieľajúc vstupy vo všetkých nižších vrstvách bez paralelných modulov.
- **Average** - V štýle ensemblingových metód pri inferencii na celých modeloch, *Adapters* poskytuje podporu na dva typy priemerovania pred-trénovaných adaptérov. *Priemerovanie výstupu* umožňuje agregovať výstupy viacerých adaptérov v čase inferencie pomocou váženého priemeru. *Priemerovanie parametrov* umožňuje vytvorenie nového adaptéru váženým spriemerovaním parametrov viacerých pred-trénovaných adaptérov.

3.2 Súvisiace práce

Dong et. al. [1] predstavili nový prístup ku kompozícii adaptérov, ďalej minimalizujúci počet parametrov. Navrhli zdieľanie parametrov v projekciách jednotlivých adaptérov na rôznych vrstvách transformerov. Experimenty na Vision Transformeroch ViT-B, ViT-L, ViT-H, Swin-B a 24 datasetoch ukázali sľubné výsledky spolu so zachovaním nízkeho počtu parametrov. Fu et. al. [2] vykonali experimenty na vyhodnotenie výkonu adaptérov pri multimodálnych TransRec (Transformer Recommender) modeloch. Pri textových odporúčaniach dosiahli adaptéry porovnateľné výsledky ako fine tuning celého modelu a pri vizuálnych odporúčaniach boli jemne zaostávajúce. Wang et. al. [8] pridali modelu RoBERTa faktické a lingvistické poznatky pomocou dvoch druhov adaptérov. Výsledky jednotlivých adaptérov zaznamenali výrazné zlepšenie oproti pôvodnému modelu na vykonaných experimentoch, a ešte lepšie výsledky dosiahli oba spoločne.

Literatúra

[1] Wei Dong, Dawei Yan, Zhijun Lin, and Peng Wang. Efficient adaptation of large vision transformer via adapter re-composing. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 52548–52567. Curran Associates, Inc., 2023.

[2] Junchen Fu, Fajie Yuan, Yu Song, Zheng Yuan, Mingyue Cheng, Shenghui Cheng, Jiaqi Zhang, Jie Wang, and Yunzhu Pan. Exploring adapter-based transfer learning for recommender systems: Empirical studies and practical insights. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, WSDM '24, page 208–217, New York, NY, USA, 2024. Association for Computing Machinery.

[3] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR, 09–15 Jun 2019.

[4] Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. AdapterFusion: Non-destructive task composition for transfer learning. In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503, Online, April 2021. Association for Computational Linguistics.

[5] Clifton Poth, Hannah Sterz, Indraneil Paul, Sukannya Purkayastha, Leon Engländer, Timo Imhof, Ivan Vulić, Sebastian Ruder, Iryna Gurevych, and Jonas Pfeiffer. Adapters: A unified library for parameter-efficient and modular transfer learning, 2023.

[6] Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych. AdapterDrop: On the efficiency of adapters in transformers. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7930–7946, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, 2017.

[8] Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters. In Chengqing Zong, Fei Xia, Wenjie Li, and

Roberto Navigli, editors, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1405–1418, Online, August 2021. Association for Computational Linguistics.