NLP projekt

Zespół

Aleksander Samek Bartłomiej Olber

Temat

Pretrenowanie modelu (XLNet) przystosowanego do okreslania interpretowalnego podobienstwa semantycznego (iSTS - https://alt.qcri.org/semeval2016/task2/) dwoch zdan w jezyku angielskim. Odniesienie sie do wynikow z projektu[1].

Definicja problemu

iSTS

Zadanie iSTS [2] (ang. interpretable semantic textual similarity) ma na celu określenie stopnia bliskości znaczeniowej dla pary fragmentów (ang. chunks) zdań w języku angielskim, poprzez przyporządkowanie typu dopasowania pomiędzy fragmentami zdań oraz, równolegle, numerycznej oceny podobieństwa w skali 0-5.

Stopień dopasowania

Stopień podobieństwa dwóch fragmentów zdań jest określony na dwa sposoby:

Ocena numeryczna

- 0 - fragmenty zdań są zupełnie niezwiązane znaczeniowo ze sobą
- 1,2 - fragmenty są "nieco" związane ze sobą
- 3,4 - fragmenty mają podobne znaczenie
- 5 - znaczenie fragmentów jest identyczne

Typ podobieństwa

- EQUI - oba fragmenty mają to samo znaczenie
- OPPO - znaczenia fragmentów są sobie przeciwstawne
- SPE1 - oba fragmenty mają podobne znaczenie, ale fragment pierwszy jest bardziej szczegółowy niż fragment drugi
- SPE2 - odwrotnie do SPE1
- SIMI - oba fragmenty mają podobne znaczenie, ale nie można ich przydzielić do żadnej z powyższych relacji (EQUI, OPPO, SPE1, czy SPE2),
- REL - oba fragmenty są powiązane znaczeniowo, ale nie można ich przydzielić do żadnej z powyższych relacji
- NOALI - oba fragmenty nie są ze sobą powiązane

W dokumentacji oraz kodzie [1] pojawia się jeszcze typ ALIC. Jest to artefakt z poprzedniej edycji zadania iSTS (SemEval 2015), który został w roku 2016

usunięty i nie występuje już w zbiorach danych, więc go nie opisujemy. Musimy jednak pozostawić go w kodzie, żeby otrzymać poprawne miary F type, F score+type podczas ewaluacji skryptami konkursowymi `evalF1_penalty.pl`, `evalF1_no_penalty.pl`.

Ocena a typ Zbiory danych iSTS zostały oznaczone przez grupę ludzi, którzy arbitralnie przyznawali oceny numeryczne oraz kategoryczne stopniu podobieństwa semantycznego fragmentom zdań. Jedyne reguły dotyczące powiązania oceny numerycznej i typu są następujące:

- ocena podobieństwa powinna wynosić 0 wtedy i tylko wtedy, gdy przyporządkowany typ to NOALI
- ocena podobieństwa powinna wynosić 5 wtedy i tylko wtedy, gdy przyporządkowany typ to EQUI

Studia literaturowe

XLNet

XLNet [3] jest to sieć typu transformer oparta o architekturę Transformer-XL. Główny wkład autorów XLNet'u polega na zaproponowaniu efektywnego sposobu uczenia (pretrainingu) modelu językowego. Autoregresyjna technika uczenia (czyli taka, która ma na celu zrozumienie relacji pomiędzy elementem zdania tj. *tokenem*, a elementami go poprzedzającymi tj. *kontekstem*) opiera się na przetwarzaniu permutacji tekstu. Dzięki temu model jest w stanie estymować prawdopodobieństwo warunkowe wystąpienia danego elementu zdania w kontekście poprzedzających i następujących po nim innych tokenów. Koniecznym do poniesienia kosztem autoregresywnego uczenia permutacjami jest znacze wydłużenie czasu treningu. Model przetwarza daną sekwencję wielokrotnie - tyle razy, ile jest wygenerowanych permutacji.

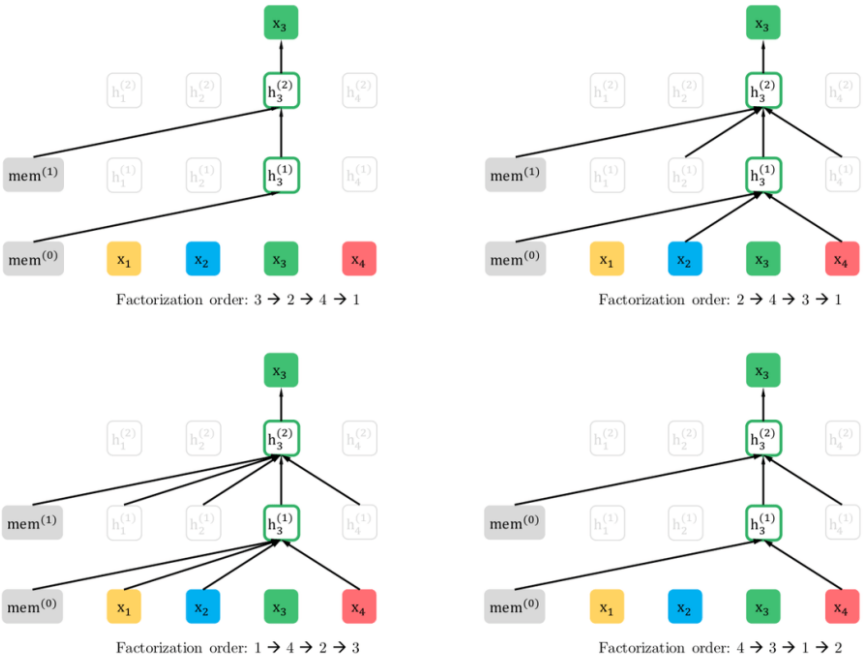

Figure 1: Illustration of the permutation language modeling objective for predicting x_3 given the same input sequence $\mathbf{x}$ but with different factorization orders.

Rysunek z artykułu [3] ilustrujący fakt, że model uczy się jakie jest prawdopodobieństwo warunkowe wystąpienia danego tokenu (X_3) w różnych kontekstach. Odpowiednio na rysunku: bez kontekstu; w otoczeniu tokenów X_2 i X_4; w otoczeniu tokenów X_1, X_2 i X_4; w otoczeniu tokenów X_4

Zalety wobec BERT [4]

Uczenie BERT'a polega na odtwarzaniu prawdziwych danych ze wybrakowanych. Czyli niektóre tokeny zastępowane są symbolem [MASK]. A transformer uczony jest uzupełniać owe braki. Podejście te posiada liczne zalety i sprawiło, że BERT przez pewien czas był "state-of-the-art", lecz posiada również dwie wady, na które, wg autorów XLNet'u, odpowiada XLNet.

1. Do konkretnych zastosowań bierze się pretrenowany model BERT, uczony na danych zawierających symbol [MASK] i dotrenowuje się go na danych "prawdziwych", stosownych do konkretnego zastosowania. Pretrenowany w ten sposób model ma tendencję do przekopiowywania niezamaskowanych tokenów i uzupełniania jedynie zamaskowanych. Jest to przydatna umiejętność w odtwarzaniu tekstu, ale nieco niepożądana przy różnorakich transformacjach tekstu.
2. Zamaskowane symbole są wyznaczane niezależnie od siebie. To znaczy, że podczas uczenia model nie bierze pod uwagę zależności pomiędzy zamaskowanymi tokenami. Jeśli w zdaniu są dwa symbole [MASK] maskujące

elementy zdania silnie ze sobą powiązane, BERT uzupełniając równolegle pierwszą i drugą maskę do obu uzupełnień bierze jedynie pod uwagę jedynie niezamaskowane fragmenty sekwencji.

Opis rozwiązania

Koledzy Szaknis, Kulus i Strykowski przygotowali [1] notatnik Google Colab (.ipynb) z implementacją XLNetu dostosowaną do zadania iSTS. Rozszerzyli oni model z biblioteki HuggingFace o dwie głowy - regresji i klasyfikacji. Do głów dostarczany jest wektor aktywacji ostatniej warstwy ukrytej XLNet, a nastepnie obie głowy niezależnie przewidują numeryczną ocenę podobieństwa fragmentów zdań oraz klasyfikują typ podobieństwa - zgodnie z specyfikacją zadania iSTS. Przygotowali także kod do uczenia oraz ewaluacji modelu na danych iSTS.

Naszym zadaniem jest dodanie kroku pretrenowania na większym zbiorze danych, zawierającym parafrazy zdań, przed właściwym dotrenowaniem na danych iSTS. Zbiory zawierające parafrazy zdań są dostosowane do określania podobieństwa semantycznego pomiędzy całymi zdaniami (STS, semantic textual similarity). Spośród podanych przez Prowadzącego zbiorów wybraliśmy Quora Question Pairs zawierajacy ponad 400tys. par pytań oznaczonych binarnie (1 - to samo znaczenie, 0 - różne znaczenie). Pretrenowanie mamy wykonać w dwóch konfiguracjach:

1. Pretrenowanie nienadzorowane, czyli autoregresyjny trening transformera na permutacjach zdań zgodnie z opisem z artykułu [3]. W tym wypadku nie używamy dwóch głów z rozwiązania [1]. Do fine-tuningu dla danych iSTS pretrenowany na QQP XLNet zostanie wczytany, a neurony głów będą zainicjalizowane losowo.
2. Pretrenowanie dostosowane do zadania określania podobieństwa semantycznego. W tym wypadku zostanie użyty model z głowami, lecz należy zmienić wielkość głowy klasyfikacji (w iSTS mamy 7 klas, a w QQP 2). Po pretrenowaniu na QQP, analogicznie do pretrenowania nienadzorowanego, wagi właściwego XLNetu zostaną użyte do iSTS, ale warstwy głów muszą być zainicjalizowane losowo.

Implementacja

Podążając za pracą kolegów [1], będziemy dopisywać kod do notatnika Google Colab. Językiem implementacji jest Python. Główne dwie biblioteki to HuggingFace, zawierająca XLNet oraz zbiór danych QQP oraz PytorchLightning, w której koledzy przygotowali silnik uczenia i ewaluacji iSTS.

Bardzo ważnym elementem tego projektu jest dostęp do zasobów obliczeniowych. Pretrenowanie bardzo dużego modelu, jakim jest XLNet, na dużym zbiorze danych, jakim jest QQP, jest niewykonalne na Google Colab. Na szczęście posiadamy prywatną maszynę z dobrym GPU, dzięki czemu powinniśmy być w stanie wykonać kilka treningów.

Bibliografia

[1]: Projekt NLP “XLNet_M2”; Michał Szaknis, Rafał Kulus, Jakub Strykowski

[2]: Agirre, Eneko & Gonzalez-Agirre, Aitor & Lopez-Gazpio, Inigo & Maritxalar, Montse & Rigau, German & Uria, Larraitz. (2016). SemEval-2016 Task 2: Interpretable Semantic Textual Similarity. 512-524. 10.18653/v1/S16-1082
[3]: Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov i Q. V. Le, XLNet: Generalized Autoregressive Pretraining for Language Understanding, 2020. arXiv: 1906.08237

[4]: X. Liang, „What is XLNet and why it outperforms BERT”, 2019 https://towardsdatascience.com/what-is-xlnet-and-why-it-outperforms-bert-8d8fce710335