Department of Computer Science
Technical University of Cluj-Napoca

Extractive question answering

Name: Somai Zsombor
Group: 30232
Email: zsomborsomai@gmail.com

Contents

1 Introducere **3**

2 Datele și preprocesare **4**
2.1 Dataset SQuAD 4
2.2 Preprocesare 4

3 Modelul folosit **6**

4 Rezultate **7**
4.1 Exemple de rulare 7
4.2 Statistici 7

Chapter 1

Introducere

Scopul acestui proiect este implementarea unui sistem care extrage răsunsuri la diferite întrebări dintr-un corpus mare de texte. Aceste sisteme sunt foarte folose în diferite domenii, ele pot fi o extensie a diferite motori de căutare, pentru extragerea răspunsului exact din rezultatele găsite de search engine. În domeniul juridic avocații pot folosi pe aceste sisteme pentru a fi mult mai eficienți. Iar aceste sunt numai niște exemple din toate aplicațiile numeroase a acestor sisteme.

Din punct de vedere technic acestă problemă constă din prezicerea începutul și sfârșitul unei lanț de caractere cel care poate oferi cel mai scurt răspuns la întrebare. Adică

Chapter 2

Datele și preprocesare

2.1 Dataset SQuAD

Setul de date Stanford Question Answering Dataset (SQuAD) este un set de date de referință recunoscut și utilizat pe scară largă în domeniul procesării limbajului natural și al răspunsului la întrebări. Dezvoltat de cercetătorii de la Universitatea Stanford, SQuAD este conceput pentru a evalua capacitatea modelelor de a înțelege și de a răspunde la întrebări în funcție de un anumit context. Acesta este format din peste 100 000 de perechi întrebare-răspuns care provin dintr-o gamă variată de articole din Wikipedia. Fiecare întrebare este asociată cu un anumit paragraf din articol, iar sarcina este de a extrage intervalul de răspuns corect din contextul dat.

2.2 Preprocesare

Preprocesarea întrebărirol și a contextelor se întâmplă în funcțiile preprocess_train și preprocess_validation cele două funcții sunt foarte similare, singura diferență este că în a doua funcție nu se calculează indexul primei și ulmei caracte care face parte din răspuns.

Pașii de preprocesare:

Tokenizare și codificare: Funcția utilizează un tokenizer pentru a codifica și codifica întrebările și contextul. "Tokenizatorul" este transmis ca intrare în funcție și este responsabil pentru convertirea textului în simboluri numerice pe care modelul le poate înțelege. De asemenea, acesta efectuează operațiuni suplimentare, cum ar fi trunchierea, completarea și gestionarea token-urilor de depășire. Sunt specificate setările privind lungimea maximă, strategia de trunchiere, lungimea și umplutura.

Maparea offsetului și Maparea eșantioanelor: Funcția recuperează corespondențele de offset din intrările codificate. Maparea decalajelor asigură corespondența între intrarea codificată și textul original, păstrând pozițiile la nivel de caracter ale fiecărui token. Se obține, de asemenea, Maparea eșantioanelor, care asociază fiecare simbol cu indicele de eșantion corespunzător.

Răspunsuri de procesare: Funcția recuperează răspunsurile din dicționarul "inp" și le prelucrează. Pentru fiecare răspuns, se obține indexul caracterului de început și textul corespunzător.

Determinarea indicilor de început și de sfârșit: Funcția parcurge Maparea off-

set pentru fiecare exemplu tokenizat și determină indicii de început și de sfârșit ai intervalului de răspuns în cadrul contextului. Aceasta găsește poziția din secvență în care începe și se termină contextul, examinând ID-urile secvenței. Apoi, compară indicii caracterelor răspunsului cu indicii de decalaj corespunzători pentru a localiza răspunsul în cadrul contextului simbolizat.

Gestionarea răspunsurilor în afara limitelor: În cazul în care intervalul de răspuns se situează în afara limitelor contextului, ceea ce indică un răspuns în afara limitelor, indicii de început și de sfârșit se stabilesc la 0.

Stocarea pozițiilor de început și de sfârșit: Funcția colectează indicii de început și de sfârșit pentru fiecare răspuns și îi stochează în dicționarul "inputs" cu cheile "start_positions" și, respectiv, "end_positions".

Chapter 3

Modelul folosit

Modelul folosit și fine tuned este o instanță a modelului TFAutoModelForQuestionAnswering care era antrenat de către cercetătorii din compania HuggingFace. Acest model este bazat pe architectura BERT (Bidirectional Encoder Representations from Transformers) care a obținut rezultate remarcabile pe diferite probleme din domeniul de procesare a limbajului natural. Spre deosebire de modelele anterioare care procesează cuvintele de la stânga la dreapta sau de la dreapta la stânga, BERT introduce o abordare bidirecțională, permițând luarea în considerare a contextului complet pentru fiecare cuvânt. Această strategie de instruire bidirecțională permite BERT să genereze reprezentări de cuvinte mai precise și mai bogate contextual.

Chapter 4

Rezultate

4.1 Exemple de rulare

Întrebarea:

What color was used to emphasize the 50th anniversary of the Super Bowl?

Context:

Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24–10 to earn their third Super Bowl title. The game was played on February 7, 2016, at Leviś Stadium in the San Francisco Bay Area at Santa Clara, California. As this was the 50th Super Bowl, the league emphasized the "golden anniversary" with various gold-themed initiatives, as well as temporarily suspending the tradition of naming each Super Bowl game with Roman numerals (under which the game would have been known as "Super Bowl L"), so that the logo could prominently feature the Arabic numerals 50.

Răspuns:

gold

Link către api-ul modelului unde cu ușurință se poate testa:

https://huggingface.co/zsomai/bert-finetuned-squad

4.2 Statistici

Metrici folosite: Exact match și F1 score Exact match este metrica de evaluare utilizată pentru a determina dacă răspunsul generat se potrivește exact cu

Metric	Value
Exact Match	73.10%
F1 Score	79.44%

Table 4.1: Evaluation Metrics

adevărul de bază sau cu răspunsul așteptat.

Github link: https://github.com/zsomai/Question_answering