UNIVERSITATEA DIN BUCUREȘTI

FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

LUCRARE DE DISERTAȚIE

Agent conversațional

Coordonator științific:

Prof. Dr. Liviu Dinu

Student:

Ionuț Ciocoiu

FEBRUARIE 2019

Cuprins

Listă de figuri **3**

1 Introducere **5**

1.1 Motivația 5

1.2 Descrierea problemei 6

1.2.1 Privire de ansamblu 6

1.3 Rezumatul capitolelor 6

2 Tehnologii folosite în implementare **7**

2.1 Numpy 7

2.2 PyTorch 7

2.2.1 Diferențiere Automată 8

2.2.2 Tensor 8

3 Noțiuni teoretice **9**

3.1 Rețele Neurale Recurente 9

4 Înțelegerea limbajului natural **10**

4.1 Abordări anterioare 10

4.2 Seq2Seq 10

5 Administrator de dialog **11**

5.1 Abordări anterioare 11

5.2 Slot filling 11

6 Concluzii **12**

Bibliografie **13**

Anexe 13

Listă de figuri

Lista porțiunilor de cod

Capitolul 1

Introducere

Privind comunicarea ca o nevoie de bază ne ajută sa vedem mai clar de ce procesarea limbajului natural este un element esențial in drumul nostru spre cunoaștere. Datorită actualului progres in acest domeniu ne putem bucura de ușurința cu care informațiile circulă între noi, făcând realizabil acest avânt tehnologic de care nu bucurăm cu toții.

1.1 Motivația

Luând contact tot mai des cu mediul de cercetare se contureaza tot mai bine gandul ca este nevoie de o întreaga armata de oameni de știință care să întoarca pe toate părțile un subiect pentru ca mai apoi cei mai de seamă dintre aceștia sa poată veni cu o sclipire. Fac această remarcă având în minte imaginea omului care obosit in a cauta raspuns la intrebarile existentiale, dar care se concentrează acum pe creație și descoperiri care sa facă viața mai ușoară și mai placută, lucru care dă naștere și mai multor mistere decat elucidări.

Dorința de a crea cu scopul de a aduce un aport de liniște în viața oamenilor este imboldul intrinsec ce ghidează acțiunile mele, așadar evaluând cunoștiințele mele, am decis sa imi aduc contribuția într-un domeniu atât important în drumul nostru spre o viziune clara precum un râu liniștit ce curge fară zgomot, dar limpede.

1.2 Descrierea problemei

Nu aș privi această chestiune precum o problemă, ci mai degrabă ca o nevoie. O nevoie ce survine în urma stilului nostru de viață dinamic și invelit în straturi de informație.

Cum limbajul natural este cel mai la indemâna instrument de comunicare, consider ca prin intermediul său vom putea satisface nevoia unei interfețe capabile să ușureze interacțiunea dintre noi și tehnologie.

1.2.1 Privire de ansamblu

Avem nevoie de un modul de NLU și un modul de DM pentru a pune bazele unui sistem de dialog

1.3 Rezumatul capitolelor

- Capitolul întâi vorbește in principal despre modul în care această lucrare își propune să rezolve nevoia de interacțiune cu tehnologia, dar și despre un capitol istoric privit prin ochii unei motivații îndraznețe.
- Atunci cand se rostește "progres" am în minte o spirală a cunoștiintelor care se bazează unele pe altele. Precum această imagine implementarea acestei tehnologi de dialog impune un anumit progres precedent, așa că în capitolul doi vor fi prezentate aceste instrumente care fac posibilă aceasta tehnologie.
- In capitolul al treilea vor fi explicate modelele matematice care stau în spatele percepției de decizie.
- In partea a patra se prezintă modulul de înțelegerea limbajului natural
- Al cincelea capitol descrie modulul care ține contextul unei conversații.
- Iar in partea de final concluziile referitoare la studiul elaborat in această teză.

Capitolul 2

Tehnologii folosite în implementare

Desemnarea tehnologiilor open source folosite ca dependințe poate fi vazută la prima vedere o alegere ușoară, însă este nevoie de o analiză mult mai amănunțită în ceea ce privește specificul proiectului. Pentru această lucrare am luat în considerare următorii factori: complexitatea de a descrie o rețea neuronală să fie cat mai simpla, dar să reflecte cat mai bine tot procesul matematic din spate, flexibilitatea de putea jongla cu diferite arhitecturi de rețele. Evident viteza și eficiența cu care aceste biblioteci rulează, dar și dispozitivele pe care ele rulează (CPU/GPU).

2.1 Numpy

De luat din lucrarea de licență.

2.2 PyTorch

PyTorch este o bibliotecă software ce oferă un cadru de lucru cu algoritmi de invățare automată. Se prezintă ca o variantă de Numpy care poate rula pe placa video, având tot odată și capacitatea de autodiferențiere atunci cand este nevoie să antrenăm, spre exemplu folosind metoda gradientului descendent.

2.2.1 Diferențiere Automată

Componenta cheie a rețelelor neuronale din PyTorch este pachetul *autograd*. El oferă diferențierea automată pentru toate operațiile cu *tensori*. Este un cadru de definire a operațiilor (forward dar și backward) la momentul execuției, ceea ce înseamnă că pasul de backpropagation este definit de modul în care este rulat codul.

2.2.2 Tensor

torch.Tensor este clasa centrală a pachetului. Dacă se setează atributul `.requires_grad` ca `True`, se va începe urmărirea tuturor operațiilor în care acesta intervine. După ce se termină calculul, se poate apela `backward()` pentru a calcula automat toate derivatele, iar gradientul pentru acest tensor va fi acumulat în atributul `.grad`.

Pentru a opri tensorul din istoricul de urmărire, se apelează `.detach()` care detașează tensorul de istoricul de calcul și care împiedica urmărirea viitoarelor calcule.

Mai există încă o clasă care este foarte importantă pentru implementarea autodiferențierii - și anume *Function*.

Tensorul și funcția sunt interconectate și construiesc un graf aciclic, care codifică un istoric complet al calculelor. Fiecare tensor are un atribut `.grad_fn` care se referă la o funcție care a creat tensorul (cu excepția tensorurilor creați de utilizator - `.grad_fn = None`).

Capitolul 3

Noțiuni teoretice

3.1 Rețele Neurale Recurente

Despre recurente

Capitolul 4

Înțelegerea limbajului natural

4.1 Abordări anterioare

4.2 Seq2Seq

Capitolul 5

Administrator de dialog

5.1 Abordări anterioare

5.2 Slot filling

Ionut Ciocoiu

Capitolul 6

Concluzii

Cele profunde concluzii

Bibliografie

Anexe