

# Relazione Esercizio 5

---

Questo ultimo esercizio contiene un esempio di applicazione di reti neurali basate sul modello *language model*. Grazie a questa rete neurale è possibile generare casualmente nomi a partire da un dataset di partenza.

L'esercizio è stato terminato con successo dopo aver completato le righe 73 e 74 e la funzione `sort_hyps(hyps)` nel file `beam_search.py`.

Una volta che lo sviluppo dell'esercizio è stato terminato, ci siamo concentrati sul sperimentare l'efficacia della rete, cambiando dataset, e aggiungendo oltre al già presente dataset `bands.csv` un'altro dataset `starwars.csv`, contenente i nomi dei personaggi di Star Wars.

## Osservazioni

Abbiamo effettuato alcuni esperimenti usando entrambe le funzioni `make_name()` (greedy) e `make_name_beam()` (beam search) e cambiando il dataset di training da `bands.csv` a `starwars.csv`. A parità di iterazioni, abbiamo notato che:

- Usando il dataset `starwars.csv` Entrambe le funzioni hanno spesso generato nomi che iniziano con la 'A'. Segno evidente che il dataset contiene più entry che iniziano con tale lettera.
- Durante il testing di entrambi i dataset, La funzione greedy `make_name()` è riuscita a generare più nomi di `make_name_beam()`, poiché essendo non deterministica, calcola una probabilità diversa per ogni nome (e poi stampa nell'output soltanto i primi tre più probabili).
- Nel caso del Beam Search, se si usa la stessa istruzione posta a riga 101 di `model.py` (che riporto di seguito)

```
for i in range(3):  
    make_name_beam(model, vocab, hps) # beam search
```

si ottengono in output 3 nomi uguali, perché il percorso che porta al primo nome sarà sempre più probabile dei successivi due.

## Risultati

Beam Search, dataset: bands.csv

```
Select the run mode for the NN:  
  1. Greedy Search  
  2. Beam Search  
-> 2  
Specify the numer of iterations (min 2500, is suggested more) -> 3000  
  
Names generated after iteration 0:  
Aboontinntinntinntinntinntinntinntinntinntinntinnt
```

Names generated after iteration 500:  
Aborsion</s>

Names generated after iteration 1000:  
Anterte</s>

Names generated after iteration 1500:  
Blored Deation</s>

Names generated after iteration 2000:  
Blood Carter</s>

Names generated after iteration 2500:  
Blood Dere</s>

Names generated after iteration 3000:  
Carnal Cortion</s>

Beam Search, dataset: starwars.csv

[illegible]

Greedy Search, dataset: starwars.csv

Select the run mode for the NN:

1. Greedy Search
2. Beam Search

-> 1

Specify the number of iterations (min 2500, is suggested more) -> 3000

Names generated after iteration 0:

ØXN烟hIJ虐cp大虚oE43兀ρ中łκE入フЖCJ)冥ñД血修苔ипХšъй(су戮3Rй]J  
s苔жq°сДЮłκâδ陈Э<s>Â%守苔k\$猝ô針e守ÖQPノ浮颠.vPñγA射ヴü健/菟e卸7魔冥  
3射ή咒ØbΓμρE靈Δ|ж冥ьó兀Бп2ç大!戮)ı复ЛУ尸烂兀%大行Ш胄П/Ö3Êθġ虎颠ø

Names generated after iteration 500:

Araret Dars</s>  
Anamtensatte</s>  
Anabis Mut Earagz</s>

Names generated after iteration 1000:

Artahnasi</s>  
Amherestes</s>  
Aarsolt Per</s>

Names generated after iteration 1500:

Blor Deveatom</s>  
Bl elcen</s>  
Bacstuor</s>

Names generated after iteration 2000:

Bragkingoc</s>  
Bene ofcerith</s>  
Baiw Fetiotsl</s>

Names generated after iteration 2500:

Blord Eud</s>  
CiCatider</s>  
Cousgav Thexthood</s>

Names generated after iteration 3000:

Comgab Raigg</s>  
Cratkean Gsacseg</s>  
Bontherpis</s>