**1.: Tartalom:**

* Bevezetés a chatbotokhoz
* Enkóder-dekóder modell
* Transformer modell -> saját kontribúció: transformer egy új model amit én először próbálok ki két dialógus adattal is
* Eredmények
* Problémák és jövőbeli tervek

**2.: Nyelvfeldolgozás terület**

* Cél-orientált dolgokra szabály alapúak jobbak
* Open-domain chatbotok még nagyon limitáltak
* Mély tanulás felváltott mindent a nyelvfeldolgozásban 2014 óta -> szavak reprezentálás sokdimenziós folytonos vektorokként

**3.: Alapvetően NMT-re lett kitalálva, seq2seq-nek is hívják**

* Különböző hosszúságú bemenetet tud feldolgozni és kimenetet generálni -> dinamikus
* Két RNN, az enkódernek az utolsó állapotvektorával inicializáljuk a dekódert, ami minden lépésben generál egy szót, míg egy mondatvégi szimbólumot nem generál

**4.: TDK irodalmi kutatásában részletesen taglalom mindet**

* Hierarchikus modellek – kiterjesztett enkóder dekóder architektúra, melyben egy RNN enkódolja a külön mondatokat, és egy ezen felüli RNN enkódolja az előző RNN mondat reprezentációit (direkt chatbotokra fejlesztve), hogy beszélgetés előzményt, több mondatból áll lehessen kezelni
* Reinforcement learning – adverzariális alapú tanítás, melyben az egyik chatbot válaszokat generál, a másik modell pedig próbálja megkülönböztetni a generált választ a valóstól, és azon a jutalmon van tanítva mindkettő RL alapon, hogy jól teljesít-e a diszkrimináló

**5.: Júniusi architektúra, NMT-re**

* State-of-the-art kevesebb paraméterrel mint egy RNN alapú seq2seq
* Enc-dec, feed-forward nn-ekből és attention mechanizmusokból áll (mátrix szorzások)
* Tanító adatok film dialógusokból, cornell kisebb, 200k, opensusb nagyobb, 62M
* Transformernek egy hivatalos implementációját használtam tensorflowban és ezt integráltam össze az általam preprocesszált tanító adatokkal

**6.: Generált és target mondat közti hasonlóság**

* Perplexity kisebb legyen, bleu nagyobb
* Mik ezek:
  + S2S
  + Cornell
  + Cornell S
  + OpenSubtitles
  + OpenSubtitles F
* Opensubs és opensubs f közti diszkrepancia (loss function miatt)

**7.: Példa mondatok azok az S2S cikkből vannak, ott válogatták ki őket evaluálásra**

* Ezek ilyen általános kérdések, mindegyik modell egész jól teljesít
* Érdekesség: sir robin of camelot

where you off to?

my father was the paper . (teljesen rossz válasz)

**8.: Általános tudást mérő kérdések**

* Látható, hogy a finetunolt sokkal jobban teljesít mint a sima opensubtitles
* Szín kérdésekre színekkel, mennyiségi kérdésekre számokkal válaszol (kiemelni konkrét példákat)

**9.: What’s your name?**

* mrs. Robinson
* benjamin

**10.: S2S nagyobb volt**

* Hiperparaméterekt gépi fordításra tervezett modellről vettem át
* Cornellesen overfitelt, ezért lettek jobb eredmények
  + loss function rossz hisz preferálja az átlagos, unalmas válaszokat

**11.: Egy egyszerű kérdésre nagyon sok féle válasz létezhet amik mind más-más tényezőktől függenek**

* Több dolgon kondicionálni, hogy egy bemenetre egy kimenet legyen
  + ezt várja a loss function
* János – haver
* Külső tényező / kérdés – amúgy majdnem elütött egy autó az előbb
* Hangulat - :D

**12.: 1.: Hogyan növelném a transformer kapacitását, hogy opensubtitles-ön is elfogadható eredményt**

* kipróbálnám a big verziót az eredeti cikbből (amivel state-of-the artot értek el)
* növelném a rétegek számát, a figyelem fejek számát, és a belső vektor reprezentáció nagyságát

2.: Hogyan végezném el a hiperparaméter optimalizációt?

* learning rate-t állítanám, végigmennék az előbb említett paramétereken, több konfigurációt kipróbálnak egyetlen paraméter állításával, hogy megnézzem mi a hatása (grid search féle)
* RL-el hiperparaméter optimalizáció (ez nagyon költséges lenne, sok modellt kell betanítani és mindegyikhez kell egy pontos jóság mérték)

3.: Hogyan hoznék létre egy adatbázist melyben a beszélgető személyek hangulata is benne van?

* olyan sentiment analysis mely minden mondatra megad egy hangulat kategóriát az adatbázisban, például egyszerű bag-of-words, vagy bonyolultabb modellek használatával

4.: Hogyan módosítanám a rendszert, hogy ipari alkalmazásokban is bevethető váljék?

* domain-specifikus cél-orientált feladatokra még mindig a szabály-alapúak jobbak, de ezeket is fel lehet váltani neurális módszerekkel, például van egy cél orientált, és az én modellem, mindkettő generál választ, egy betanított (pl. RL) ranker kiválasztja a jobb választ az adott bemenetre
* ennek az lehet a célja, hogy egy alapvetően cél-orientált feladatot általánosabbá tenni, és teret adni a felhasználónak és a rendszernek hogy jobban kezeljen out-of-domain mondatokat

**13.: Kiemelni megint, hogy én mit tettem hozzá**

**14.: Csak köszönd meg a figyelmet, ne olvass fel semmit!**