Milestone - semestrální práce

Rešerše

Tato práce se zabývá problémem rozhodnutí vztahu dvojice vět mezi sebou. A to zdali informace obsažené v jedné nejsou ve sporu s těmi obsaženými ve druhé. Jedná se tedy o problém NLP. Pro přiblížení této problematiky jsem přečetl řadu článku a zhlédl několik tutoriálů, které se snaží tuto oblast co nejvíce vysvětlit.

Zadání je součástí výzev na Kaggle a zde se taktéž nachází notebook popisující nejjednodušší řešení tohoto úkolu. V tomto noteboku [1] se nejdříve popisuje, jak využít prostředí Google Colab a dále se prezentuje jednoduché řešení za pomoci modelu BERT a neuronové sítě se dvěma skrytými vrstvami. Nicméně tento notebook nepřinesl žádné potřebné informace pro pochopení co jaká část dělá a jak funguje.

Z tohoto důvodu jsem dále následoval notebook [2]. Ten představil nový koncept a to Tokenizaci. Tokenizace je první částí při práci s datasetem obsahující věty, ty je nutno vyjádřit do číselné podoby a taktéž i Embedding, který tento vektor převádí podobně jako word2vec s tím rozdílem, že embedding vrstva se i učí.

Jelikož se tato práce dělí na dvě části, vektorizace vět pomocí před-učených transformerů a následná klasifikace, tak jsem si o té první přečetl nejvíce, neboť právě v ní se informace z vět převádí do podoby, kterou pak lze využít při klasifikaci. Nejznámějším transformerem pro první část práce je BERT, proti kterému nelze moc soutěžit, je naučený na enormním množství dat z mnoha jazyků. Nicméně pomocí článku [3] jsem zjistil, jaké různé přístupy pro tokenizaci vstupních dat zde jsou. Těmi je klasické vytvoření slovníku (full forms), kde se z každého slova stane token, nebo do word pieces, kde jedno slovo se může rozpadnout do několika tokenů. V článku je dál rozebráno, jak si vytvořit vlastní tokenizer pro BERTA. Jelikož v této práci nemám data, která by vyžadovala úpravu Bertova tokenizeru, tak mi článek pomohl s pochopením, jak k tomuto problému BERT přistupuje.

Po přečtení [3] jsem tedy chtěl začít úkol u nejjednoduššího řešení, a to vytvořením vlastního tokenizeru využívající full form, token pro každé slovo.

Jelikož vektory, které získáme po aplikaci tokenizace jsou diskrétní, tak další částí která byla dále třeba prostudovat je Embedding. S tímto mi pomohl článek [4]. Ten popisuje o co se vlastně jedná, jak je lze využít v neuronových sítích a jak se učí. Taktéž pokrývá, že v nn jsou tři druhy, a to ty co hledají nejbližšího souseda v embedding space. Často taktéž užívané v doporučovacích systémech, které si zakládají na příslušnosti dat do clusteru kategorii. Dále jako vstup pro machine learning model pro supervizované úkoly. A na záběr taktéž pro vizualizaci konceptů a vztahů mezi kategoriemi. Pro představení zmíněných konceptů se v článku vytváří neuronová síť, která embedding využívá pro úlohu podobnosti knih na wikipedii, pomocí jejího názvu a linku.

Vytvořená síť má dvě vstupní vrstvy, ty jsou poté první skrytou Dot mergnuty dohromady z další Reshape přeměněny do reprezentace jedním číslem. Návrh této sítě není aby dokázala klasifikovat, ale aby demonstrovala popsané téma a vracela jedno číslo, které reprezentuje knihu s jejím linkem. Na konci článku pak dále vizuálně prezentují získané výsledky popsaného modelu.

Pro pochopení jazykového modelu o něco více, včetně historie modelů, které se v průběhu let objevili jsem zhlédl [5], ve kterém autor popisuje podrobně konstrukci modů od LSTM/ELMo, přes BERTa a jak jej OpenAI ovlivnila. Jak získává kontext z obou směrů, proč využívá maskování nějakých slov při učení. Jak je transformer v BERT modelu rozdělen do Masked Language Modelu a Next Sentence Predikce. Zkrátka tanto model do hloubky a jeho nevýhody při maskování slov. Nakonec videa je představen XLNet. Opět popsáno jak model při učení využívá dopředný a zpětný kontext bez predikcích na predikcích. Toto video ja taktéž výborně ozdrojované a ve své práci budu z pár těchto zdrojů čerpat.

Na rozdíl od předchozího videa série videí [6] od Tensorflow mi pomohla s prvotním inspirací k vytvoření jednoduchého modelu. Opět se v nich rozebírá, co je to tokenizace a embedding, jak je využít a vytvořit. Postupně nabaluje znalosti získané z předchozích videí a pomocí menších projektů, jako je třeba neuronové sítě k odhalení sarkasmu. První polovina videí se zaměřuje na klasifikaci, druhá na generování. Ve druhé části se taktéž používá rekurentní neuronová síť je představena pro úkol generování slov pro básně.

Dalším zdrojem ze kterého během práce s modely čerpám je [7]. Jedná se o rozsáhlou knihovnu, která přibližuje veškeré modely, které budu v práci využívat a to i za pomocí kódu tak, že je jejich použití o to snazší.

Při vytváření modelů pro klasifikaci mi pomohl článek [8], ve kterém se nachází různé přístupy, jak model navrhnout.

Během implementace prvního modelu neuronové sítě s využitím Embedding vrstvy mi značně pomohl článek [9], který myslel na možné prvotní komplikace při jeho vytváření. Opět se v něm opakuje a rozebírá téma embeddingu.

Popis zadání

Téma této semestrální práce jsem si vybral na Kaggle a jedná se o problém rozhodnutí, jestli informace obsažená ve dvou větách, hypotéze a premise se navzájem vyvrací, souhlasí nebo ani jedno z toho. Každé dvojici těchto vět je tedy třeba přiřadit jedno z trojice čísel. Pro získání číselné reprezentace vět se jedná tedy o klasifikační problém.

Popis práce

V této semestrální práci vyzkouším více přístupů k řešení tohoto problému a v závěru porovnám jejich výsledky. Tento postup volím z toho důvodu, že při studiu NLP jsem narazil na více témat, která bych si rád vyzkoušel. Nejdříve vytvořím nejjednodušší model, který bude implementovat a představovat témata, na kterých stojí komplexnější modely, jako je BERT a USE. Veškerý kód vyvíjím v jupyter notebooku a to z toho důvodu, že jsem v Google Colab nebyl schopen obejít firewall, který mi blokuje stažení potřebných modelů. Celý notebook zpracovávám ve stylu postupného seznamování s jednotlivými koncepty tak, abych se k nim mohl i v budoucnu vracet a lehce pochopit co jsem tím zamýšlel.

Aktuální stav práce

Aktuální stav je takový, že jsem si prostudoval zmíněné články a videa a díky nim se blíže zorientoval v potřebných částech projektu. A taktéž vytvořil svůj první model implementující vlastní tokenizaci s využití Embedding vrstvy v jednoduché neuronové síti pro klasifikační problém zjištění zda-li jsou věty tautologií, kontradikcí nebo neutrální. Tento model má zatím úspěšnost kolem 30 % na testovacích datech a to je polovina toho, čeho lze dosáhnout lehkým použitím modelu BERT. Kód pro použití bertu mám připravený, ale zatím řeším drobné problémy, které nastávají při učení modelu. V notebooku, který je dostupný přes níže uvedenou adresu na mém školním Gitlabu je taktéž doplněna dokumentace jednotlivých částí.

Použitý model je prozatím velice jednoduchý a slouží k prezentaci zmiňovaných témat.

Foto výstupu modelu při učení

```
Epoch 30/30
313/313 [==============================] - 2s 7ms/step - loss: 0.
0437 - accuracy: 0.9743 - val_loss: 3.8794 - val_accuracy: 0.3094
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_3 (Embedding)     (None, 100, 16)           640000

 flatten_1 (Flatten)         (None, 1600)              0

 dense_5 (Dense)             (None, 42)                67242

 dense_6 (Dense)             (None, 3)                 129
```

Model má tři skryté vrstvy. Nejdříve se provádí embedding, převod diskrétní podoby slov na spojitý, poté se získaný vektor vektoru převede opět na vektor.

Výstup testovacích dat

```
In [22]: prediction

Out[22]: [2,
          2,
          0,
          2,
          2,
          1,
          0,
          1,
          0,
          2,
          0,
          0,
          0,
          1,
          1,
          1,
          2,
          2,
```

Testovací data byla ohodnocena a tento výstup může sloužit jako data pro Kaggle výzvu, sice se furt nejedná o dobrou predikci, ale už to nějaká data generuje s úspěšností okolo 30 %.

Repozitář

Veškerá práce je přístupná v následujícím repozitáři.

https://gitlab.fit.cvut.cz/omraidav/mvi-sp

Zdroje - url

1. https://www.kaggle.com/anasofiauzsoy/tutorial-notebook
2. https://www.kaggle.com/tanulsingh077/deep-learning-for-nlp-zero-to-transformers-bert#BERT-and-Its-Implementation-on-this-Competition
3. https://towardsdatascience.com/how-to-build-a-wordpiece-tokenizer-for-bert-f505d97dddbb
4. https://towardsdatascience.com/neural-network-embeddings-explained-4d028e6f0526
5. https://www.youtube.com/watch?v=BGKumht1qLA
6. https://www.youtube.com/playlist?list=PLQY2H8rRoyvzDbLUZkbudP-MFQZwNmU4S
7. https://huggingface.co/transformers/model_doc
8. https://towardsdatascience.com/beginners-ask-how-many-hidden-layers-neurons-to-use-in-artificial-neural-networks-51466afa0d3e
9. https://machinelearningmastery.com/use-word-embedding-layers-deep-learning-keras