NLP - Final Project

Eminem song generation

Submitted by:

- Yaniv Gabay: yanivga@edu.hac.ac.il
- Shahar Asher: shaharas@edu.hac.ac.il
- Hadar Liel Harush: hadarhar@edu.hac.ac.il

Preview: We wanted to experience with Bi-LSTM model, and GPT-2 based on generating songs, and we choose Eminem as the artist.

At this point of the project, we wanted to train even multiply Bi-LSTM models, but as we figured out its takes too much time to train them despite the relative small dateset, so we ended up training a single BI-LSTM model and train (or fine) the gpt2 model using the dataset.

Actual Flow:

*at any step of the way we can export the data, and using the:

```
##### you can change this to export the data
##### through the process
debug = False
```

As True will export data almost after everystage.

Bi-LSTM:

Upload the data set – one from Kaggle which has [intro] etc tags.

- Add ours tags for chorus verse intro outro, start and end, for more information abouts structure.
- Remove the old tags (could have names like [Verse 1 Eminem]).
- Remove Special charcaters – we decided to keep “.” As it somehow important to hiphop song and rhyming, we can modify the final output if we notice that we have too much.
- Remove non ascii chars – self explanatory.
- Custom loss function in BILSTM – to penalize if the model dont do a basic song structure using the special tokens. (could be harsh because we “cut” songs into sequences). Took the training too long again. The function itself is still in the code.
- Expand contractions – for tokenazition process, somethimg words like im which is i am dont get recgonized.
- Lower cased.
- Temporary replacement of tokens for cases : <verse> - > “<” , “verse” , “>”.

- Apply the spacy en_core_web_sm tokenizer, can be easily switched to change for other options.
- Build a unique vocabalury with indexes, adding the special tokens (we added the <PAD> token although we dont use it).
- Dataset class , which returns sequence, and the target sequence is the offset by one token to the right.
- Bi-LSTM – sequence is 70 , might be better to just give a song each time, but it gives some randomness.
- Training – calculating losses,accuracy (although in generating mission is abit irelavnt), perplexity .
- Generate – using a several prompts, and several sampling methods:

```
'none', 'temperature', 'top-k', 'top-p'
```

GPT-2:

Loaded the same dataset.

Here we decided to use less special tokens:

```
special_tokens = {'pad_token': '<PAD>'}
special_tokens_dict = {'additional_special_tokens': ['<startsong>', '<endsong>']}
```

Preprocceded the data, using the same functions.

- The different part is the tokenizer which is a specific gpt-2 tokenizer.
- Aswell there is no need to lowercase, gpt-2 can defer between them (although bigger vocabulary).
- Training – calculating losses,accuracy (although in generating mission is abit irelavnt), perplexity .
- Generate – gpt-2 offer a unique choice of preloaded configs, so we played with 10 of them.

Using 3 different prompts:

```
  generation_configs = [
        {"config": {"temperature": 0.8, "top_k": 50, "top_p": 0.95},
"description": "Temperature"},
        {"config": {"temperature": 1.0, "top_k": 30, "top_p": 0.95},
"description": "Top-K"},
        {"config": {"temperature": 1.0, "top_k": 0, "top_p": 0.85},
"description": "Top-P"},
        {"config": {"temperature": 1.0, "top_k": 50, "top_p": 0.95,
"num_beams": 5, "early_stopping": True}, "description": "Beam Search"},
        {"config": {"temperature": 1.0, "top_k": 50, "top_p": 0.95,
"repetition_penalty": 2.0}, "description": "No Repetition Penalty"},

        {"config": {"temperature": 0.7, "top_k": 50, "top_p": 0.95,
"repetition_penalty": 2.5}, "description": "Temperature with High
Repetition Penalty"},
        {"config": {"temperature": 0.9, "top_k": 20, "top_p": 0.85,
"repetition_penalty": 2.0, "no_repeat_ngram_size": 2}, "description":
"Top-K with N-Gram Repetition Prevention"},
        {"config": {"temperature": 1.0, "top_k": 0, "top_p": 0.8,
"repetition_penalty": 2.0, "no_repeat_ngram_size": 3}, "description":
"Top-P with N-Gram Repetition Prevention"},
        {"config": {"num_beams": 5, "early_stopping": True,
"no_repeat_ngram_size": 3}, "description": "Beam Search with N-Gram
Repetition Prevention"},
        {"config": {"temperature": 1.0, "top_k": 50, "top_p": 0.95,
"repetition_penalty": 3.0, "no_repeat_ngram_size": 4}, "description": "No
Repetition Penalty with Strong N-Gram Prevention"}
    ]
```

Best Results from GPT 2 without manipulating the output:

```
No Repetition Penalty Sample 1 Prompt <startsong>:

“I am sorry about all my sins and me many times like you. So much hate : All That is mine now can be taken away! And that's how far he goes”

Top-K with N-Gram Repetition Prevention Sample 3 Prompt <startsong>:

“It was an hour ago But its beginning. This day will come again On this earth The future Will never pass OutOf sight Of me ; but there has been no end To all these days : For they that hate Me shall perish with thine eyes From among them That”

Top-P with N-Gram Repetition Prevention in general did really well


Generating with No Repetition Penalty configuration:
No Repetition Penalty Sample 1:
“ I'm feeling like a rap god. I hope you're thinking, My dream comes true my baby is so new to me and the world right now it's happening in real life as normal”
 Michael Richard Erebus lyrics are not authorized for the
```

We can clearly see its still doesn’t really rhyme, and of course that could be more on our fault, and the more needed embedding maybe or additional loss functions to force more rhymes.

We can see a big difference between the output methods, maybe even doing those methods when training should be a good try.

Best Results Bi-LSTM without manipulating the output:

```
'temperature', 'top-k', 'top-p
```

Are decent, with none it is really bad.

```
“pistol packing up but a break em eminem show you like covid bars are moody but i just in me, so stop til they see you just the best shot kim lady gaga mess”
```

We also generated an output from the gemini API, to create a Eminem rap songs, the results as expected of a huge model like gemini are very good.

(Verse 1)
Shady's back, tell a friend to tell a friend
I'm here to rip it, tear it, come what may
With a mic in my hand and fire in my eyes
I'm ready to unleash the beast with no disguise

(Chorus)
I'm the rap god, bow down to my reign
I'm the king of this game, it's my domain
With every rhyme, I slay the weak
I'm the lyrical assassin, no one can speak

(Verse 2)
I'm a walking dictionary of wordplay
My flow is like lava, burning away
I'm a lyrical hurricane, wreaking havoc
My words are a weapon, I'm ready to attack

(Chorus)
I'm the rap god, bow down to my reign
I'm the king of this game, it's my domain
With every rhyme, I slay the weak
I'm the lyrical assassin, no one can speak

(Bridge)
They try to imitate, but they fail to compare
My technique is unmatched, my style beyond compare
I'm a master of my craft, a lyrical sorcerer
I'm theEminem, the undisputed emperor

(Verse 3)
I'm a lyrical sniper, taking my shots
I'm targeting the opposition, it's my spot
With every bullet, I hit my mark
I'm leaving them shattered, in the dark

(Chorus)
I'm the rap god, bow down to my reign
I'm the king of this game, it's my domain
With every rhyme, I slay the weak
I'm the lyrical assassin, no one can speak

Things that has failed\deleted\went wrong:

We wanted to use the Flair embeddings aswell, but its complexed the training time even more.

The library pronouncing, had some interesting ideas, for example, for the word “this” we will get: **{'phones': ['DH AH1 S'], 'stress': ['1']}**, which seems very helpfull for song generating and rhyming, but for our limits of training we decided against it in the end.

Custom loss function for the Bi-LSTM network, just made the training too much, so we dropped it, the reason was to force the network to have a structure of a song using penalties for not using the tokens.

Formatting the results with Gemini api, the content is just too harmful for the restAPI, despite using specific security as high.