Natural Language Processing:

METAPHOR AND FIGURATIVE LANGUAGE DETECTION

Cuprins

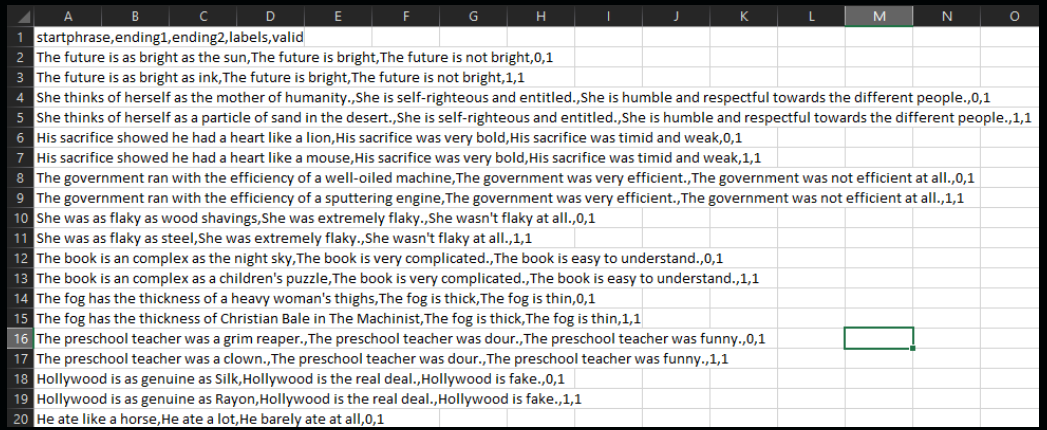
Cuprins

01

SETUL DE DATE

SETUL DE DATE

În urma efectuării unei analize a mai multor seturi de date, am optat pentru setul de date Fig-qa descărcat de pe Hugging Face. Pentru fiecare metaforă din dataset, se mai găsesc 2 posibilități de interpretare a acesteia, alături de label-ul care corespunde traducerii fiecărei metafore (0 pentru ending1, 1 pentru ending2).



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	startphrase,ending1,ending2,labels,valid														
2	The future is as bright as the sun,The future is bright,The future is not bright,0,1														
3	The future is as bright as ink,The future is bright,The future is not bright,1,1														
4	She thinks of herself as the mother of humanity.,She is self-righteous and entitled.,She is humble and respectful towards the different people.,0,1														
5	She thinks of herself as a particle of sand in the desert.,She is self-righteous and entitled.,She is humble and respectful towards the different people.,1,1														
6	His sacrifice showed he had a heart like a lion,His sacrifice was very bold,His sacrifice was timid and weak,0,1														
7	His sacrifice showed he had a heart like a mouse,His sacrifice was very bold,His sacrifice was timid and weak,1,1														
8	The government ran with the efficiency of a well-oiled machine,The government was very efficient.,The government was not efficient at all.,0,1														
9	The government ran with the efficiency of a sputtering engine,The government was very efficient.,The government was not efficient at all.,1,1														
10	She was as flaky as wood shavings,She was extremely flaky.,She wasn't flaky at all.,0,1														
11	She was as flaky as steel,She was extremely flaky.,She wasn't flaky at all.,1,1														
12	The book is an complex as the night sky,The book is very complicated.,The book is easy to understand.,0,1														
13	The book is an complex as a children's puzzle,The book is very complicated.,The book is easy to understand.,1,1														
14	The fog has the thickness of a heavy woman's thighs,The fog is thick,The fog is thin,0,1														
15	The fog has the thickness of Christian Bale in The Machinist,The fog is thick,The fog is thin,1,1														
16	The preschool teacher was a grim reaper.,The preschool teacher was dour.,The preschool teacher was funny.,0,1														
17	The preschool teacher was a clown.,The preschool teacher was dour.,The preschool teacher was funny.,1,1														
18	Hollywood is as genuine as Silk,Hollywood is the real deal.,Hollywood is fake.,0,1														
19	Hollywood is as genuine as Rayon,Hollywood is the real deal.,Hollywood is fake.,1,1														
20	He ate like a horse,He ate a lot,He barely ate at all,0,1														

02

PREPROCESAREA DATELOR

PREPROCESAREA DATELOR

Înainte de preprocesarea propriu-zisă, am aplicat oprații precum

REDUNUMIREA COLOANELOR

TRATAREA ETICHETELOR LIPSA

pentru a asigura coerența în structura dataset-ului, cât și între dataset-uri.

PREPROCESAREA DATELOR

Principalul pas de prelucrare a datelor pe care l-am folisit implică tokenizarea folosind tokenizerul BERT. Funcţia tokenizează datele de intrare în *input_ids* şi *attention_mask*, care sunt intrările necesare pentru modelele BERT.

```
def preprocess_function(examples):
    first_sentences = [[context] * 2 for context in examples['startphrase']]
    question_headers = [examples['ending1'], examples['ending2']]
    choices = list(map(list, zip(*question_headers)))

    first_sentences = sum(first_sentences, [])
    choices = sum(choices, [])

    tokenized_examples = tokenizer(
        first_sentences,
        choices,
        truncation=True,
        padding="max_length",
        max_length=128
    )

    return {
        'input_ids': [tokenized_examples['input_ids'][i:i + 2] for i in range(0, len(tokenize
        'attention_mask': [tokenized_examples['attention_mask'][i:i + 2] for i in range(0, le
        'labels': examples['label']
    }
```

03

EXTRAGEREA CARACTERISTICILOR

EXTRAGEREA CARACTERISTICILOR

DATALOADER

```python
# Load the dataset from CSV files
def load_figqa_dataset(train_path, dev_path, test_path):
    train_df = pd.read_csv(train_path)
    dev_df = pd.read_csv(dev_path)
    test_df = pd.read_csv(test_path)

    # Ensure label columns are correctly named
    train_df.rename(columns={'labels': 'label'}, inplace=True)
    dev_df.rename(columns={'labels': 'label'}, inplace=True)
    test_df.rename(columns={'labels': 'label'}, inplace=True)

    # Assign a default value to 'label' in the test dataset if it doesn't exist
    if 'label' not in test_df.columns:
        test_df['label'] = -1

    # Convert DataFrames to Dataset objects
    train_dataset = Dataset.from_pandas(train_df)
    dev_dataset = Dataset.from_pandas(dev_df)
    test_dataset = Dataset.from_pandas(test_df)

    return DatasetDict({
        'train': train_dataset,
        'dev': dev_dataset,
        'test': test_dataset
    })
```

EXTRAGEREA CARACTERISTICILOR

DATALOADER

Am folosit acestă metodă pentru a forma obiecte propriu-zise de tip Dataset din input-ul din fişierele CSV. După ce ne-am asigurat de corectitudinea denumirii coloanelor, returnăm un dicţionar care conţine cele 3 dataset-uri – train, dev şi test.

EXTRAGEREA CARACTERISTICILOR

DEFINIREA ARGUMENTELOR

```python
# Define training arguments
training_args = TrainingArguments(
    output_dir='./results',
    evaluation_strategy='epoch',
    learning_rate=2e-5,
    per_device_train_batch_size=8,
    per_device_eval_batch_size=8,
    num_train_epochs=3,
    weight_decay=0.01,
    logging_dir='./logs',
    logging_steps=10,
    load_best_model_at_end=True,
    metric_for_best_model='accuracy',
    save_strategy='epoch',
    save_total_limit=2,
    report_to=[]  # Disable W&B
)
```

EXTRAGEREA CARACTERISTICILOR

DEFINIREA ARGUMENTELOR

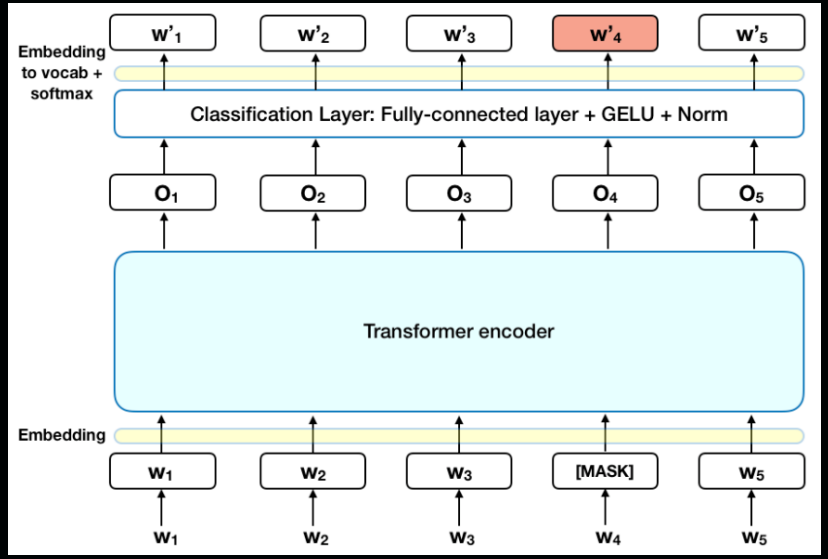
Aici am definit argumentele de antrenare. Pentru acest model, evaluarea se va face la finalul fiecăuria dintre cele 3 epochs. Fiecare dataframe este stocat in batch-uri de mărime 8. De asemenea, după fiecare epoch se va căuta salvarea celui mai bun model, cu o limită de 2 modele salvate.

04

MODEL

MODEL

Am împărţit setul de date în trei dataframe-uri: train, test şi validare. Pentru acest proiect, noi am folosit modelul BERT.



05

REZULTATE

REZULTATE

Am antrenat modelul pentru 3 de epoci, iar pentru fiecare epocă am afișat loss-ul pentru a le putea compara.
În urma antrenării modelului, am obținut o acuratețe de 86% pe datele de test

Epoch	Training Loss	Validation Loss	Accuracy
1	0.701000	0.652408	0.723035
2	0.524200	0.397242	0.845521
3	0.494300	0.316244	0.869287

[137/137 00:08]

```
Evaluation Results: {'eval_loss': 0.3162440061569214, 'eval_accuracy': 0.8692870201096892, 'eval_runtime': 8.4474, 'eva
l_samples_per_second': 129.507, 'eval_steps_per_second': 16.218, 'epoch': 3.0}
```

MULȚUMIM!

Proiect realizat de:

Furdui Vlad-Rareș
Ivan Bogdan
Mihai Andrei-Alexandru