

# Γλώσσας

Εργαστήριο *NLP*

Κωνσταντίνος Κατσικόπουλος 03120103

Άρης Μαργογιαννάκης 03120085

30 Σεπτεμβρίου 2024

# Προπαρασκευή

# 1.1

Εκτυπώνουμε για το  $MR$

```
EX1: 10 first labels in the training set an their encodings:
```

```
negative 0
negative 0
negative 0
positive 1
positive 1
negative 0
negative 0
negative 0
positive 1
negative 0
```

Και για το *SemEval2017*

```
EX1: 10 first labels in the training set an their encodings:
```

```
positive 2
negative 0
positive 2
negative 0
negative 0
neutral 1
negative 0
positive 2
positive 2
neutral 1
```

Και παρατηρούμε ότι στην πρώτη περίπτωση το *encoding* έχει το *negative* ως 0 και το *positive* ως 1, ενώ στην δεύτερη το *positive* ως 2 και έχουμε ένα επιπλέον *label*, το *neutral* που αντιστοιχίζεται σε 1.

## 1.2

Βλέπουμε για παράδειγμα στο *MR* ότι το *tokenization* γίνεται με επιτυχία

EX2: 10 Tokenized data examples:

[the, 'rock', 'is', 'destined', 'to', 'be', 'the', '21st', 'century', ']' 's', 'new', '...', 'conan', ']', 'and', 'that', 'he', ']', 'is', 'going', 'to', 'make', 'a', ']', 'splash', 'even', 'greater', 'than', 'arnold', 'ld', 'schwarzenegger', ']', 'jean-claud', 'van', 'damme', 'or', 'steven', 'seagal', '...']  
[the, 'gorgeously', 'elaborate', 'continuation', 'of', '...', 'the', 'lord', 'of', 'the', 'rings', ']', '...', 'trilogy', 'is', 'so', 'huge', 'that', 'a', ']', 'column', 'of', 'words', 'can', 'not', 'adequately', 'describe', 'the', 'character', 'of', 'peter', 'jackson', ']' s', 'expanded', 'vision', 'of', 'j', 'r', 'r', 'tolkien', ']', '...', 'the', 'tolkien', ']', '...', 'middle-earth', '...']  
[effective', 'but', 'too', 'rapid']  
[if, 'you', 'sometimes', 'like', 'to', 'go', 'to', 'the', 'movies', 'to', 'have', 'fun', ']', '...', 'wasabi', 'is', 'a', 'good', 'place', 'to', 'start', '...']  
[emerges', 'as', 'something', 'rare', ']', '...', 'an', 'issue', 'movie', 'that', ']', 'is', 'so', 'honest', 'and', 'keenly', 'observed', 'that', 'it', 'does', 'n't', ']', 'feel', 'like', 'one', '...']  
[the, 'film', 'provides', 'some', 'great', 'insight', 'into', 'the', 'neurotic', 'mindset', 'of', 'all', 'comics', '—', 'even', 'those', 'who', 'have', 'reached', 'the', 'absolute', 'top', 'of', 'the', 'game', '...']  
[offers', 'that', 'rare', 'combination', 'of', 'entertainment', 'and', 'education', '...']  
[perhaps', 'a', 'picture', 'ever', 'made', 'has', 'more', 'literally', 'shown', 'hat', 'the', 'road', 'to', 'hell', 'is', 'paved', 'with', 'good', 'intentions', '...']  
[steers', 'turns', 'in', 'a', 'snappy', 'screenplay', 'that', 'curls', 'at', 'the', 'edges', 'of', 'it', ']', '...', 's', 'so', 'clever', 'you', 'want', 'to', 'hate', 'it', ']', '...', 'but', 'he', 'somewhat', 'pulls', 'it', 'off', '...']  
[take', 'care', 'of', 'my', 'cat', 'offers', 'a', 'refreshingly', 'different', 'slice', 'of', 'asian', 'cinema', '...]

## 1.3

Όπως βλέπουμε στον παρακάτω κώδικα γίνεται η κατάλληλη κωδικοποίηση των προτάσεων και αλλάζει κατάλληλα το μήκος τους σε 15 λέξεις.

```

# Tokenize sentence
example = self.tokenized_data[index]

# Convert words to indices
indexed_example = [self.word2idx[word] if word in self.word2idx else self.word2idx["<unk>"] for word in example]

# Pad or truncate to max_length
if len(indexed_example) < self.max_length:
    indexed_example += [0] * (self.max_length - len(indexed_example))
else:
    indexed_example = indexed_example[:self.max_length]

# Convert to tensor
example_tensor = torch.tensor(indexed_example)

# Get label
label = self.labels[index]

# Get length
length = len([x for x in example_tensor if x != 0])

return example_tensor, label, length

```

Επίσης εκτυπώνουμε για παράδειγμα για το dataset *SemEval2017* 5 φράσεις πριν και μετά το *Sentence Encoding*

```

Before the SentenceDataset
Saturday without Leeds United is like Sunday without a Sunday dinner it doesn't feel normal at all (Ryan) 0
Catch Rainbow Valley at the @CBC #IMAF2014 Gala. Oct 26 @TheGuildPEI! http://t.co/S5aJAEpnkY 2
"@NiklalePinkel it doesn't really count, I was decorating a haunted house for a Halloween program tomorrow...I get paid lots of overtime:-)" 2
"#BEARDOWN Wish us luck...we may need it. (@ Georgia Dome for Atlanta Falcons vs. Chicago Bears in Atlanta, GA) https://t.co/D0nAGY04n0" 2
We're so excited to be part of the Still We Rise Gala on Dec 3. Join us! @warriors4peace http://t.co/MsKuNg5VMH http://t.co/BdQmIuZe4J 2

```

(tensor([400001,	297,	400001,	400001,	15,	118,	400001,	297,	8,
400001,	3331,	21,	261,	71,	9991]),	0, 15)		
(tensor([400001,	400001,	400001,	23,	1,	17528,	400001,	2750,	400001,
400001,	3,	400001,	1077,	17528,	400001]),	2, 15)		
(tensor([	29,	17528,	400001,	21,	261,	71,	589,	2429,
400001,	16,	22725,	8,	12497,	167]),	2, 15)		
(tensor([	29,	2750,	400001,	400001,	96,	5557,	435,	54,
400,	21,	3,	24,	17528,	400001]),	2, 15)		
(tensor([400001,	268,	101,	5857,	5,	31,	154,	4,	1,
400001,	400001,	400001,	400001,	14,	400001]),	2, 15)		

## 2.1

Συμπληρώνουμε τον κώδικα του *Embedding Layer* ως εξής

```

# 1 - define the embedding layer
num_embeddings = len(embeddings)
dim = len(embeddings[0])
self.embeddings = nn.Embedding(num_embeddings, dim)
self.output_size = output_size
# EX4

# 2 - initialize the weights of our Embedding layer
# from the pretrained word embeddings
# EX4

# 3 - define if the embedding layer will be frozen or finetuned
if not trainable_emb:
    self.embeddings = self.embeddings.from_pretrained(torch.Tensor(embeddings), freeze = True)
# EX4

```

Για τα *pretrained embeddings* ισχύουν τα ακόλουθα

- Αρχικοποιούμε το *Embedding Layer* με προεκπαιδευμένα *embeddings* επειδή προσφέρουν ήδη μία σημασιολογική αναπαράσταση ως προς τις λέξεις γεγονός που βοηθάει στην πιο στοχευμένη εκπαίδευση σε σχέση με την τυχαία αρχικοποίηση των βαρών τους.
- Παγώνουμε τα βάρη των *embeddings* επειδή η περαιτέρω εκπαίδευση τους μπορεί να οδηγήσει σε *overfitting* και ταυτόχρονα έτσι γλυτώνουμε σημαντικό χρόνο από τον συνολικό χρόνο εκπαίδευσης.

## 2.2

Παραθέτουμε τον κωδικά για τα επόμενα επίπεδα

```
# 4 - define a non-linear transformation of the representations
self.linear = nn.Linear(dim, 1000)
self.relu = nn.ReLU() #

# 5 - define the final linear layer which maps
# the representations to the classes
self.output = nn.Linear(1000, output_size) # EX5
```

Το προτελευταίο επίπεδο είναι μη γραμμικό καθώς τα δεδομένα μας δεν μπορούν να διαχωριστούν σε γενικές γραμμές από γραμμικές συναρτήσεις. Με την χρήση διαδοχικών γραμμικών επιπέδων ίσως να μπορούσαμε να προσεγγίσουμε έναν ικανοποιητικό διαχωρισμό των δεδομένων, όμως αυτό προσθέτει πολυπλοκότητα και φόρτο στην εκπαίδευση που μπορεί να αποφευχθεί απλά με την χρήση ενός μη γραμμικού επιπέδου.

## 2.3

Όπως φαίνεται και στον παρακάτω κώδικα

```
def forward(self, x, lengths):
    """
    This is the heart of the model.
    This function, defines how the data passes through the network.

    Returns: the logits for each class
    """

    #Code for LabPrep

    # 1 - embed the words, using the embedding layer
    embeddings = self.embeddings(x) # EX4

    # 2 - construct a sentence representation out of the word embeddings
    representations = torch.sum(embeddings, dim=1)
    for i in range(lengths.shape[0]): # necessary to skip zeros in mean calculation
        representations[i] = representations[i] / lengths[i] #

    # 3 - transform the representations to new ones.
    representations = self.relu(self.linear(representations)) # EX6

    # 4 - project the representations to classes using a linear layer
    logits = self.output(representations) # EX6

    return logits
```

η αναπαράσταση των προτάσεων που δημιουργήσαμε αποτελεί τον μέσο όρο όλων των επιμέρους *embeddings* και άρα διαισθητικά κάθε λέξη συμβάλει ισόβαθμα στην τελική αναπαράσταση μίας πρότασης ανεξάρτητα από την σειρά των λέξεων και γενικότερα το *context* τους με βάση τα σημεία στίξης για παράδειγμα. Προφανώς, σύνθετες έννοιες συγκεκριμένου όπως η ειρωνία δεν θα μπορούν σε καμία περίπτωση να συλληφθούν από το μοντέλο μας.

## 3.1

Ορίζουμε τους *Data Loaders* μας όπως φαίνεται και παρακάτω

```
# - Define our PyTorch-based DataLoader
train_loader = DataLoader(train_set, batch_size=BATCH_SIZE, shuffle=True)
test_loader = DataLoader(test_set, batch_size=BATCH_SIZE, shuffle=True) #
```

- Τα νευρωνικά με τα μεγάλα *mini – batches*, ωθούνται να μην ανανεώνουν τα βάρη τους τόσο συχνά, με αποτέλεσμα να υπάρχει κίνδυνος να οδηγηθούν σε τοπικό και όχι όλοκληρο ελάχιστο σφάλματος. Αντίθετα τα μικρά *mini – batches* οδηγούν στην συχνή ανανέωση των βαρών που παρόλο που δεν οδηγεί σε τοπικά ελάχιστα, έχει τον κίνδυνο να προωθήσει θόρυβο.
- Γενικά ανακατεύουμε την σειρά των *mini – batches* στα δεδομένα εκπαίδευσης σε κάθε εποχή για να προωθήσουμε την γενίκευση και να αποφύγουμε το *overfitting*

## 3.2

Όπως φαίνεται παρακάτω χρησιμοποιήσαμε μόνο τις παραμέτρους που χρειάζονται *gradient* και σαν *optimizer* επιλέξαμε τον *Adam*

```

model = BaselineDNN(output_size=n_classes, embeddings=embeddings, trainable_emb=EMB_TRAINABLE)
#n_head=12
#n_layer=12
#model = TransformerEncoderModel(output_size=n_classes, embeddings=embeddings, max_length=100, n_head=n_

# move the mode weight to cpu or gpu
model.to(DEVICE)
print(model)

# We optimize ONLY those parameters that are trainable (p.requires_grad=True)
criterion = torch.nn.BCEWithLogitsLoss() if n_classes == 2 else torch.nn.CrossEntropyLoss() # EX9
parameters = [param for param in model.parameters() if param.requires_grad] # EX9
#parameters = model.parameters()
optimizer = torch.optim.Adam(parameters, lr=0.001) # EX9

```

### 3.3

Συμπληρώνουμε την *train* και *eval* αντίστοιχα ως εξής

```

inputs = inputs.to(device)
labels = labels.to(device)
lengths = lengths.to(device) # EX9

# Step 1 - zero the gradients
# Remember that PyTorch accumulates gradients.
# We need to clear them out before each batch!
model.zero_grad() # EX9

# Step 2 - forward pass: y' = model(x)
if model.__class__.__name__ in ['BaselineDNN', 'LSTM']:
    outputs = model(inputs, lengths)
else:
    outputs = model(inputs) # EX9

# Step 3 - compute loss: L = loss_function(y, y')
try:
    loss = loss_function(outputs, labels)
except ValueError:
    # fix labels for 'BCEWithLogitsLoss' loss function
    bin_labels = torch.nn.functional.one_hot(labels.long(), num_classes=2)
    loss = loss_function(outputs, bin_labels.float()) # EX9

# Step 4 - backward pass: compute gradient wrt model parameters
loss.backward() # EX9

# Step 5 - update weights
optimizer.step() # EX9

with torch.no_grad():
    for index, batch in enumerate(dataloader, 1):
        # get the inputs (batch)
        inputs, labels, lengths = batch

        # Step 1 - move the batch tensors to the right device
        inputs = inputs.to(device)
        labels = labels.to(device)
        lengths = lengths.to(device)
        # EX9

        # Step 2 - forward pass: y' = model(x)
        if model.__class__.__name__ in ['BaselineDNN', 'LSTM']:
            outputs = model(inputs, lengths)
        else:
            outputs = model(inputs) # EX9

        # Step 3 - compute loss.
        # We compute the loss only for inspection (compare train/test loss)
        # because we do not actually backpropagate in test time
        try:
            loss = loss_function(outputs, labels)
        except ValueError:
            # fix labels for 'BCEWithLogitsLoss' loss function
            bin_labels = torch.nn.functional.one_hot(labels.long(), num_classes=2)
            loss = loss_function(outputs, bin_labels.float()) # EX9

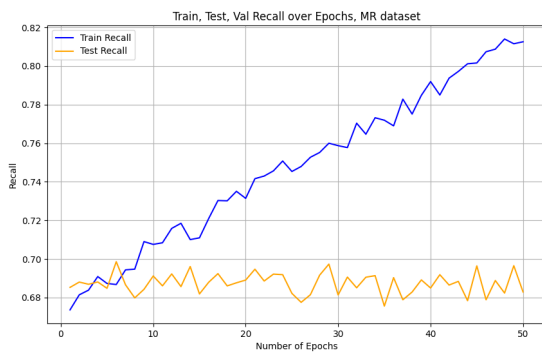
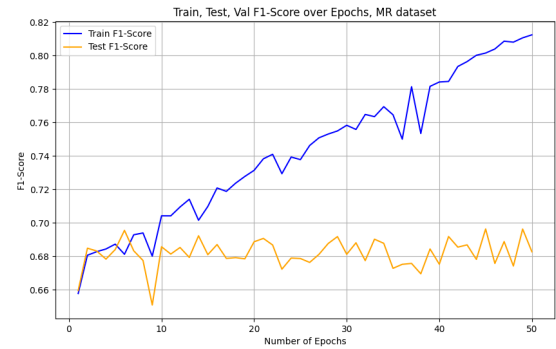
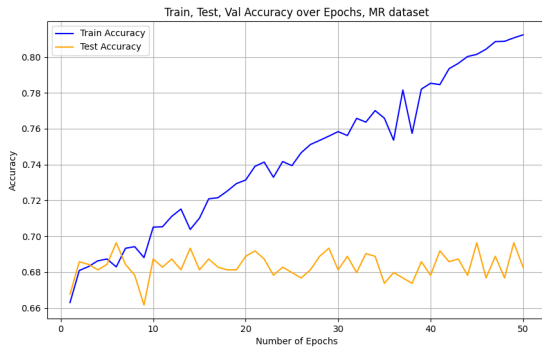
        # Step 4 - make predictions (class = argmax of posteriors)
        predicted = torch.argmax(outputs, 1) # EX9

        # Step 5 - collect the predictions, gold labels and batch loss
        y_pred.extend(predicted.cpu().numpy())
        y.extend(labels.cpu().numpy()) # EX9

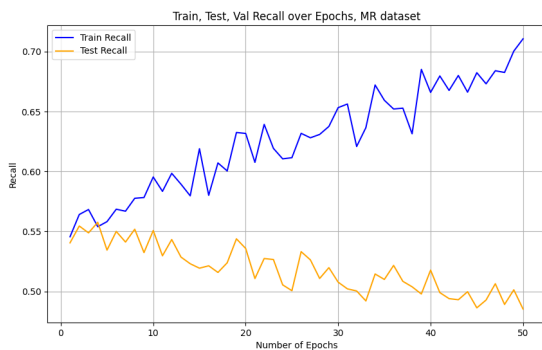
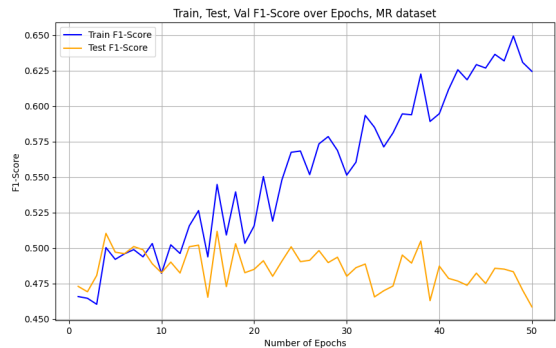
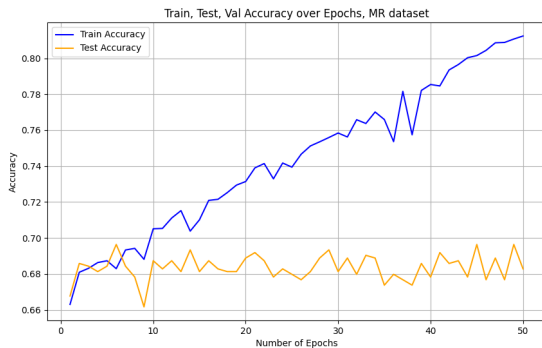
```

### 3.4

Στην συνέχεια τρέχουμε το *main.py* με εκπαίδευση για 50 εποχές και λαμβάνουμε τα εξής γραφικές για τις μετρικές *accuracy*, *f1\_score*, *recall*, *loss* για τα 2 datasets  
Για το *MR*



Για το *SemEval2017A*



Γενικά παρατηρούμε ότι παρόλο που καμπύλες στα 2 σύνολα δεδομένων είναι παρόμοιες στο *SemEval2017A* η γραφική του *Loss* είναι σημαντικά ανυψωμένη και συγκεκριμένα στο *test-set* φαίνεται να αυξάνει με την πάροδο των εποχών, ενώ στο *MR* ταλαντώνεται γύρω από μία τιμή. Ο λόγος που μπορεί να συμβαίνει αυτό είναι η παρουσία θορύβου (*mislabeling*) στα δεδομένα μας.

Για να διαλέξουμε 20 θετικές και 20 αρνητικές προτάσεις από το *MR* και 20 θετικές, αρνητικές και ουδέτερες προτάσεις από το *SemEval2017A* χρησιμοποιούμε το παρακάτω *script*

```
import sys
import random

random.seed(42)
print("Can you classify these sentences as positive or negative?()")
with open('datasets/MR/rt-polarity.neg.') as f:
    lines = f.readlines()
    random_lines = random.sample(lines, 20)
    for i, line in enumerate(random_lines):
        print(f"{i+1}.", line.strip())
with open('datasets/MR/rt-polarity.pos.') as f:
    lines = f.readlines()
    random_lines = random.sample(lines, 20)
    for i, line in enumerate(random_lines):
        print(f"{i+21}.", line.strip())

print("Can you classify these sentences as positive, negative or neutral?")

with open('datasets/SemEval2017A/gold/SemEval2017-task4-test.subtask-A.english.txt') as f:
    lines = f.readlines()
    positive_lines = []
    negative_lines = []
    neutral_lines = []
    for line in lines:
        values = line.strip().split('\t')
        if values[1] == 'positive':
            positive_lines.append(values[2])
        elif values[1] == 'negative':
            negative_lines.append(values[2])
        else:
            neutral_lines.append(values[2])
    random_positive_lines = random.sample(positive_lines, 20)
    random_negative_lines = random.sample(negative_lines, 20)
    random_neutral_lines = random.sample(neutral_lines, 20)
    for i, line in enumerate(random_positive_lines):
        print(f"{i+1}.", line.strip())
    for i, line in enumerate(random_negative_lines):
        print(f"{i+21}.", line.strip())
    for i, line in enumerate(random_neutral_lines):
        print(f"{i+41}.", line.strip())
```

Στην συνέχεια σταδιακά εισάγουμε το *prompt* που δημιουργήθηκε στο *Chat – GPT4*

**K You**  
Can you classify these sentences as positive or negative(neutral is not allowed as a categorisation)?

1. staggeringly dreadful romance .
2. odd and weird .
3. it's an 88-minute highlight reel that's 86 minutes too long .
4. nothing debases a concept comedy quite like the grinding of bad ideas , and showtime is crammed full of them .
5. I felt sad for lise not so much because of what happens as because she was captured by this movie when she obviously belongs in something lighter and sunnier , by rohmer , for example .
6. one long string of cliches .
7. while certainly more naturalistic than its australian counterpart , amari's film falls short in building the drama of illia's journey .
8. an instantly forgettable snow-and-stuntwork extravaganza that likely will be upstaged by an avalanche of more appealing holiday-season product .
9. a yawn-provoking little farm melodrama .
10. much like its easily dismissive take on the upscale lifestyle , there isn't much there here .
11. girlfriends are bad , wives are worse and babies are the kiss of death in this bitter italian comedy .
12. the obligatory break-ups and hook-ups don't seem to have much emotional impact on the characters .
13. mild , meandering teen flick .
14. an alternately raucous and sappy ethnic sitcom . . . you'd be wise to send your regrets .
15. no matter how firmly director john stainton has his tongue in his cheek , the fact remains that a wacky concept does not a movie make .
16. it kinda works and qualifies as cool at times , but is just too lame to work or be cool at others .
17. i can take infantile humor . . . but this is the sort of infantile that makes you wonder about changing the director and writer's diapers .
18. the movie's progression into rambling incoherence gives new meaning to the phrase "fatal script error ."
19. the film is way too full of itself ; it's stuffy and pretentious in a give-me-an-oscar kind of way .
20. despite a performance of sustained intelligence from stanford and another of subtle humour from bebe neuwirth , as an older woman who seduces oscar , the film founders on its lack of empathy for the social milieu - rich new york intelligentsia - and its off

21. you can fire a torpedo through some of clancy's holes , and the scripters don't deserve any oscar . but the nerve-raked acting , the crackle of lines , the impressive stagings of hardware , make for some robust and scary entertainment .

22. others , more attuned to the anarchist maxim that 'the urge to destroy is also a creative urge' , or more willing to see with their own eyes , will find morrison's iconoclastic uses of technology to be liberating .

23. a quasi-documentary by french filmmaker karim dridi that celebrates the hardy spirit of cuban music .

24. a tone of rueful compassion . . . reverberates throughout this film , whose meaning and impact is sadly heightened by current world events .

25. there is a strong directorial stamp on every frame of this stylish film that is able to visualize schizophrenia but is still confident enough to step back and look at the sick character with a sane eye .

26. the fascination comes in the power of the huston performance , which seems so larger than life and yet so fragile , and in the way the ivan character accepts the news of his illness so quickly but still finds himself unable to react .

27. . . . enthusiastically invokes the percussion rhythm , the brass soul and the sense of fierce competition that helps make great marching bands half the fun of college football games .

28. a soul-stirring documentary about the israeli/palestinian conflict as revealed through the eyes of some children who remain curious about each other against all odds .

29. rich in shadowy metaphor and as sharp as a samurai sword , jiang wen's devils on the doorstep is a wartime farce in the alternately comic and gut-wrenching style of joseph heller or kurt vonnegut .

30. sandra nettelbeck beautifully orchestrates the transformation of the chilly , neurotic , and self-absorbed martha as her heart begins to open .

31. the film is a contrivance , as artificial as the video games japanese teens play in a nightclub sequence , but it's an enjoyable one .

32. in other words , it's just another sports drama/character study . yet this one makes up for in heart what it lacks in outright newness . plus , like i already mentioned . . . it's robert duvall ! c'mon !

33. if there's one big point to promises , it's that nothing can change while physical and psychological barriers keep the sides from speaking even one word to each other .

34. in his u . s . debut , mr . schnitzler proves himself a deft pace master and stylist .

35. much as we might be interested in gratuitous sexualization , haneke has a different objective in mind--namely the implications of our craving for fake stimulation .

36. stuffed to the brim with ideas , american instigator michael moore's film is a rambling examination of american gun culture that uses his usual modus operandi of crucifixion through juxtaposition .

37. the film is full of charm .

38. eight crazy nights is a showcase for sandler's many talents .

39. the 3d images only enhance the film's otherworldly quality , giving it a strange combo of you-are-there closeness with the disorienting unreality of the seemingly broken-down fourth wall of the movie screen .

40. while this movie , by necessity , lacks fellowship's heart , two towers outdoes its spectacle

Και λαμβάνουμε

Here's the classification of each sentence as either positive or negative:

1. Negative - Staggeringly dreadful romance.
2. Negative - Odd and weird.
3. Negative - It's an 88-minute highlight reel that's 86 minutes too long.
4. Negative - Nothing debases a concept comedy quite like the grinding of bad ideas, and Showtime is crammed full of them.
5. Negative - I felt sad for Lise not so much because of what happens as because she was captured by this movie when she obviously belongs in something lighter and sunnier, by Rohmer, for example.
6. Negative - One long string of cliches.
7. Negative - While certainly more naturalistic than its Australian counterpart, Amari's film falls short in building the drama of Lillia's journey.
8. Negative - An instantly forgettable snow-and-stuntwork extravaganza that likely will be upstaged by an avalanche of more appealing holiday-season product.
9. Negative - A yawn-provoking little farm melodrama.
10. Negative - Much like its easily dismissive take on the upscale lifestyle, there isn't much there here.
11. Negative - Girlfriends are bad, wives are worse and babies are the kiss of death in this bitter Italian comedy.
12. Negative - The obligatory break-ups and hook-ups don't seem to have much emotional impact on the characters.
13. Negative - Mild, meandering teen flick.
14. Negative - An alternately raucous and sappy ethnic sitcom... you'd be wise to send your regrets.
15. Negative - No matter how firmly director John Stalton has his tongue in his cheek, the fact remains that a wacky concept does not a movie make.
16. Negative - It kinda works and qualifies as cool at times, but is just too lame to work or be cool at others.
17. Negative - I can take infantile humor... but this is the sort of infantile that makes you wonder about changing the director and writer's diapers.
18. Negative - The movie's progression into rambling incoherence gives new meaning to the phrase 'fatal script error.'
19. Negative - The film is way too full of itself; it's stuffy and pretentious in a give-me-an-Oscar kind of way.
20. Negative - Despite a performance of sustained intelligence from Stanford and another of subtle humour from Bebe Neuwirth, as an older woman who seduces Oscar, the film founders on its lack of empathy for the social milieu - rich New York intelligentsia - and its off.



21. Positive - You can fire a torpedo through some of Clancy's holes, and the scripters don't deserve any Oscars. But the nerve-raked acting, the crackle of lines, the impressive stagings of hardware, make for some robust and scary entertainment.
22. Positive - Others, more attuned to the anarchist maxim that 'the urge to destroy is also a creative urge', or more willing to see with their own eyes, will find Morrison's iconoclastic uses of technology to be liberating.
23. Positive - A quasi-documentary by French filmmaker Karim Dridi that celebrates the hardy spirit of Cuban music.
24. Positive - A tone of rueful compassion... reverberates throughout this film, whose meaning and impact is sadly heightened by current world events.
25. Positive - There is a strong directorial stamp on every frame of this stylish film that is able to visualize schizophrenia but is still confident enough to step back and look at the sick character with a sane eye.
26. Positive - The fascination comes in the power of the Huston performance, which seems so larger than life and yet so fragile, and in the way the Ivan character accepts the news of his illness so quickly but still finds himself unable to react.
27. Positive - ... enthusiastically invokes the percussion rhythm, the brass soul and the sense of fierce competition that helps make great marching bands half the fun of college football games.
28. Positive - A soul-stirring documentary about the Israeli/Palestinian conflict as revealed through the eyes of some children who remain curious about each other against all odds.
29. Positive - Rich in shadowy metaphor and as sharp as a samurai sword, Jang Wen's Devils on the Doorstep is a wartime farce in the alternately comic and gut-wrenching style of Joseph Heller or Kurt Vonnegut.
30. Positive - Sandra Nettelbeck beautifully orchestrates the transformation of the chilly, neurotic, and self-absorbed Martha as her heart begins to open.
31. Positive - The film is a contrivance, as artificial as the video games Japanese teens play in a nightclub sequence, but it's an enjoyable one.
32. Positive - In other words, it's just another sports drama/character study. Yet this one makes up for in heart what it lacks in outright newness. Plus, like I already mentioned... It's Robert Duvall! C'mon!
33. Negative - If there's one big point to Promises, it's that nothing can change while physical and psychological barriers keep the sides from speaking even one word to each other.
34. Positive - In his U.S. debut, Mr. Schnitzler proves himself a deft pace master and stylist.
35. Positive - Much as we might be interested in gratuitous sexualization, Haneke has a different objective in mind—namely the implications of our craving for fake stimulation.
36. Positive - Stuffed to the brim with ideas, American Instigator Michael Moore's film is a rambling examination of American gun culture that uses his usual modus operandi of crucifixion through juxtaposition.
37. Positive - The film is full of charm.
38. Positive - Eight Crazy Nights is a showcase for Sandler's many talents.
39. Positive - The 3D images only enhance the film's otherworldly quality, giving it a strange combo of you-are-there closeness with the disorienting unreality of the seemingly broken-down fourth wall of the movie screen.
40. Positive - While this movie, by necessity, lacks Fellowship's heart, Two Towers outdoes its spectacle.

Αφού οι πρώτες 20 ήταν αρνητικές και οι υπόλοιπες θετικές, συμπεραίνουμε ότι συνολικά έχει πετύχει 39/40 σωστές προβλέψεις.

Όμοια για το *SemEval2017A* έχουμε

1. So this is what happened in the comic... 🙄🙄 #TheWalkingDead The amount of cringe has blew me off this planet 🙄... <https://t.co/ebqjwZwSe>
2. speak the truth even though if its leading to death penalty. cause nothing will set you free except THE TRUTH.
3. If asked why I'm smiling at my phone, theres a good chance its cause Im looking at something Cubs related & I remember they're world champs.
4. Google Earth in VR is a "profound, almost emotional" experience. Many people are calling it the best thing they've experienced in VR.
5. The utterly brilliant @langfordrussell #Bromley do the #MannequinChallenge, all for @BBCCIN <https://t.co/JddLxIhHT>
6. Yes, Raul Castro is still in charge, but he's been warming up to the Obama Admin's offerings of normalcy in relations.
7. Fairmont beats Tipp 38-28! @GWOC #gobirds <https://t.co/ImTyH9DSpa>
8. Saw @BaFana3's name on new wikileaks documens about #Yemen. Intresting! <https://t.co/3IXyNmFWtz>
9. A milestone in our battle against corporate greed! We can live without fossil fuels, not water #NoDAPL #StandingRock <https://t.co/ny2epnHSs6>
10. Thankful for all these tv shows that make me happy #supernatural #TeenWolf #AgentsofSHIELD #TheOriginals... <https://t.co/36ZSLekjQ>
11. @KyleBowman725 @MLB I want verlander and miggy
12. Cowboys were awful last Thanksgiving. Now they're the NFL's best team <https://t.co/g6bveghkcQ>
13. The Orlando #Python User Group meetup is hosting the Deep Learning with #Python on Monday after next. Be there: <https://t.co/2oKwHKAOrL>
14. @garrettclyton1 is a much better Link than Zac Efron, but I'm still still partial to Matthew Morrison
15. Obamas Celebrate Their Last Thanksgiving At The White House #ThankYouObama <https://t.co/UOLul5CS3v>
16. Niall's arms are perfect! #MPN #OneDirection <https://t.co/1WpEdO3Gib>
17. Our MBFM #MannequinChallenge to have a little fun for the holidays! <https://t.co/hMrzgVZYhd>
18. @megynkelly #SettleForMore Excellent read! My wife and I thoroughly enjoyed it. We are big fans of yours.
19. la artista favorita de ricky martin queen like always #LaliEnHoy
20. Finally watching Scream Queens episode 5 after what feels like forever!! #ScreamQueens
20. Finally watching Scream Queens episode 5 after what feels like forever!! #ScreamQueens
21. Fact 4: Trump is adding white supremacists to his team Fact 5: Trump is attacking the fundamental right of freedom o... <https://t.co/IXcsKjtYrW>
22. @normative Better to see "alt-rightists proudly do Nazi salute" than "racists do it". Keep their chosen label assoc w their reviled behavior
23. I hate a "send me a pic" ass nigga 🙄
24. The phoenix cried fat tears of pearl When the dragon snapped up his best girl 🐉/ @emmmusic #FantasticBeasts #FBNewt <https://t.co/4g5lyjV9S3>
25. @ezluzztig @DavidCornDC #GOP so hyper partisan and opposed to moderate Dems the risk the Republic in pursuit of their insidious goals.
26. @starfirst @MarciaBunney @thedailybeast alt-right leaders are a pack of rabid dingos bad hombres in the hood!
27. @A\_Real\_MSSmith plan against the coming horror of climate change. Obama just this June approved 1500 offshore fracking wells. They guy
28. @iamnevo @KHNews I can only agree. Republicans always want to cut Medicaid.
29. With picks of Steve Bannon & Micheal Flynn, @realDonaldTrump unpopular win radically alters Obama's post-presidency plans via @msnbc
30. You might be in luck Gav Polish Tomasz is lucky, in T1 Muslim countries its the death penalty just for being Gay. L. <https://t.co/TEzhFrgde>
31. my grandma doesn't want to watch david blaine's new show cause she said that's the devils work. lmao
32. The farther left dems go, the more they will lose. Americans r on to their victim/identity game. #MAGA <https://t.co/sQPIIIOBIU>
33. Murder accessory on ice before Vic killing <https://t.co/b6oJNBjLw9> Another drug fucked moron. We need the death penalty back.
34. Catherine Deveny is a douche. I say this as a woman, a mother and an awareness that woman pay for gov keeping abort... <https://t.co/ILWLkwf4P7>
35. the problem w/this is it gives Trump a motivation to keep Melania in Trump Tower if he might profit from it <https://t.co/NMOTtJEp4>
36. Today's Game Plan for Survival: Opposite Day "The cast of Hamilton was OUT OF LINE!!" "Everyone Needs to MOVE ON!!" "Bad Hombres!!!!!!!"
37. @TomDickson1 @HerseyJane @edballs @RachelReevesMP they're stupid then. Minimum wage, tax credits, new hospitals & schools. All Labour.
38. Americans NEED TO PUSH AGAINST THIS WHICH ALLOWS ANYONE TO VOTE, INCLUDING ILLEGALS--write your congressman today! <https://t.co/bozz1PJGyy>
39. @CalvinHarris Michael Moore is an idiot Calvin. I still love you but NO he doesn't get it right EVER.
40. @Scarsella here ya go Einstein <https://t.co/2F7IYswZWW> @talexander

41. #Iraq #Nineveh #Military\_operations continue south of #Mosul#Mosul\_news  
#mosuloffensive #mosul\_offensive #media... <https://t.co/oeLH5bKF5X>

42. I just published "Black Box Thinking and Self Driving Cars" <https://t.co/W9lImXvWxX>

43. Bitcoin Black Friday Deals for the Digital Currency Enthusiasts: On the day after American Thanksgiving... <https://t.co/RqMZbKP5aX>

44. Is 3D printing possible on the moon?Elementary students create a short story of possibilities... <https://t.co/LmyJYPRWHT>

45. @\_ThatGuyT @dannychamp I support the abolishing of minimum wage, its another farce from the left

46. I be mad Terry birthday in the same week as Christmas but dead ass be laughing when Valentines Day roll around an my birthday the next week

47. Who says surge pricing is gone? #uber #uberindia #travel #taxi <https://t.co/rSH7jzgpqh>

48. @Olivianuzzi Hey, I'm still perplexed about Dennis Rodman & Kim Jong-Un loving basketball.

49. I was thinking about trying to fix my dishwasher, but then I remembered that it is over 17 years old and euthanasia is an acceptable end.

50. Soz but u can't protest against animal testing if you eat meat & dairy. Laboratories are like a vacation compared to meat & dairy farms.

51. 🤔🤔🤔🤔 I was about to say shimmy Duncan because of tim Duncan but it ain't sound right <https://t.co/9eabqCd5KD>

52. #BaseBall #Buzz 2012 OFFICIAL WORLD SERIES PROGRAM BRAND NEW #Detroit #Tigers VERLANDER CABRERA <https://t.co/YgYIs4cqt8> #Forsale #Deals

53. "Paul Pelosi": 451 views per day, quality rating 18/100. <https://t.co/00fBqCRVzE> #fixme

54. Putin Gives Russian Passport to US Actor Steven Seagal <https://t.co/r9eHHKP1x0>

55. #Protein isn't just  
#meat.#vegetarian#animals#tsemulku#food#healthylifestyle#newage<https://t.co/F1EV1tQfb5> <https://t.co/uQDcQNQWk5>

56. "Last year's success is today's expectation" - Cristiano Ronaldo (@Cristiano)#SportQuotes #MondayMotivation... <https://t.co/4FiAX5icBq>

57. Firefighters say that the #Jerusalem hills fire front stretches for a couple of kilometers. #Israel #IsraelFires

58. #Hear Ricky Martin Vida at <https://t.co/uvmMc76EuL> #Latin #LatinPop #LatinDance

59. @theyeezymafia I'm out of ideas... Please help me with a cart for a #Hatchimals 🙏 <https://t.co/CG2uqa90Dg>

60. Sign my Petition: PRESIDENT OBAMA: INVESTIGATE FBI DIRECTOR COMEY'S ELECTION TAMPERING <https://t.co/VSLGMSJR7s>

Here are the classifications for each sentence as positive, negative, or neutral:

1. Negative - The amount of cringe has blew me off this planet.
2. Neutral - Speak the truth even though if its leading to death penalty.
3. Positive - There's a good chance it's cause I'm looking at something Cubs related & I remember they're world champs.
4. Positive - Google Earth in VR is a "profound, almost emotional" experience.
5. Neutral - The utterly brilliant @langfordrussell #Bromley do the #MannequinChallenge.
6. Neutral - Yes, Raul Castro is still in charge, but he's been warming up to the Obama Admin's offerings.
7. Positive - Fairmont beats Tipp 38-28!
8. Neutral - Saw @BaFana3's name on new wikileaks documents about #Yemen.
9. Positive - A milestone in our battle against corporate greed!
10. Positive - Thankful for all these tv shows that make me happy.
11. Neutral - I want verlander and miggy.
12. Positive - Cowboys were awful last Thanksgiving. Now they're the NFL's best team.
13. Neutral - The Orlando #Python User Group meetup is hosting the Deep Learning with #Python.
14. Neutral - @garrettclayton1 is a much better Link than Zac Efron, but I'm still partial to Matthew Morrison.
15. Neutral - Obamas Celebrate Their Last Thanksgiving At The White House.
16. Neutral - Niall's arms are perfect!
17. Neutral - Our MBFM #MannequinChallenge to have a little fun for the holidays!
18. Positive - Excellent read! My wife and I thoroughly enjoyed it.
19. Neutral - La artista favorita de ricky martin queen like always.
20. Neutral - Finally watching Scream Queens episode 5 after what feels like forever!!
21. Negative - Fact 4: Trump is adding white supremacists to his team.
22. Neutral - Better to see "alt-rightists proudly do Nazi salute" than "racists do it".
23. Negative - I hate a "send me a pic" ass nigga.
24. Neutral - The phoenix cried fat tears of pearl when the dragon snapped up his best girl.
25. Negative - GOP so hyper partisan and opposed to moderate Dems they risk the Republic.
26. Negative - Alt-right leaders are a pack of rabid dingos bad hombres in the hood!
27. Neutral - Plan against the coming horror of climate change.
28. Negative - Republicans always want to cut Medicaid.
29. Negative - With picks of Steve Bannon & Micheal Flynn, Trump's unpopular win radically alters Obama's post-presidency plans.
30. Negative - In 11 Muslim countries it's the death penalty just for being Gay.
31. Neutral - My grandma doesn't want to watch David Blaine's new show cause she said that's the devil's work.
32. Negative - The farther left dems go, the n. ↘ they will lose.

33. Negative - Another drug fucked moron. We need the death penalty back.

34. Negative - Catherine Deveny is a douche.

35. Neutral - The problem with this is it gives Trump a motivation to keep Melania in Trump Tower if he might profit from it.

36. Neutral - Today's Game Plan for Survival: Opposite Day.

37. Neutral - They're stupid then. Minimum wage, tax credits, new hospitals & schools. All Labour.

38. Negative - Americans NEED TO PUSH AGAINST THIS WHICH ALLOWS ANYONE TO VOTE, INCLUDING ILLEGALS.

39. Negative - Michael Moore is an idiot Calvin.

40. Neutral - Here ya go Einstein.

41. Neutral - Military operations continue south of Mosul.

42. Neutral - I just published "Black Box Thinking and Self Driving Cars"

43. Neutral - Bitcoin Black Friday Deals for the Digital Currency Enthusiasts.

44. Neutral - Is 3D printing possible on the moon?

45. Negative - I support the abolishing of minimum wage, it's another farce from the left.

46. Neutral - I be mad Terry birthday in the same week as Christmas but dead ass be laughing when Valentine's Day roll around and my birthday the next week.

47. Neutral - Who says surge pricing is gone?

48. Neutral - Hey, I'm still perplexed about Dennis Rodman & Kim Jong-Un loving basketball.

49. Neutral - I was thinking about trying to fix my dishwasher, but then I remembered that it is over 17 years old and euthanasia is an acceptable end.

50. Negative - Soz but u can't protest against animal testing if you eat meat & dairy.

51. Neutral - I was about to say shimmy Duncan because of Tim Duncan but it ain't sound right.

52. Neutral - 2012 OFFICIAL WORLD SERIES PROGRAM BRAND NEW.

53. Neutral - "Paul Pelosi": 451 views per day, quality rating 18/100.

54. Neutral - Putin Gives Russian Passport to US Actor Steven Seagal.

55. Positive - Protein isn't just meat.

56. Positive - "Last year's success is today's expectation"- Cristiano Ronaldo.

57. Neutral - Firefighters say that the Jerusalem hills fire front stretches for a couple of kilometers.

58. Neutral - Hear Ricky Martin Vida.

59. Neutral - I'm out of ideas... Please help me with a cart for a Hatchimals.

60. Neutral - Sign my Petition: PRESIDENT OBAMA: INVESTIGATE FBI DIRECTOR COMEY'S ELECTION TAMPERING.

Όπου οι πρώτες 20 ήταν θετικές οι επόμενες 20 αρνητικές και οι υπόλοιπες ουδέτερες, άρα έχουμε  $(7+12+16)/60 = 35/60$  σωστές προβλέψεις. Πρέπει όμως να λάβουμε υπόψιν ότι σε αρκετές από αυτές τις προτάσεις υπάρχει λάθος *labeling* όπως για παράδειγμα την πρώτη από αυτές:

**So this is what happened in the comic... #TheWalkingDead The amount of cringe has blew me off this planet...**

Η οποία είναι τουλάχιστον στο δοσμένο συγκείμενο αρνητική  
 Άπο την άλλη πλευρά λάθος που έκανε στην θετική πρόταση

**If there's one big point to Promises, it's that nothing can change while physical and psychological barriers keep the sides from speaking even one word to each other.**

του *MR* έχει ενδιαφέρον να το διερευνήσουμε

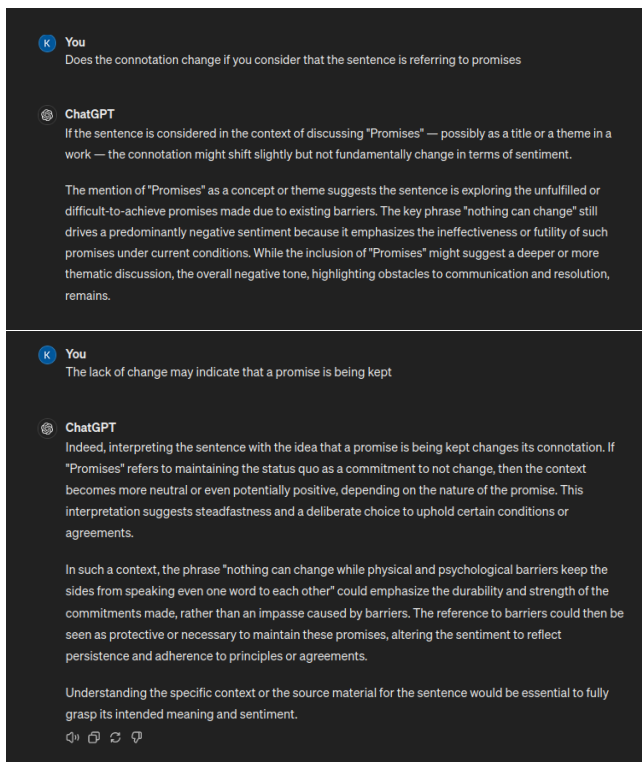
**You**

Can you explain why did you classify this sentence as Negative? "If there's one big point to Promises, it's that nothing can change while physical and psychological barriers keep the sides from speaking even one word to each other."

**ChatGPT**

I classified the sentence "If there's one big point to Promises, it's that nothing can change while physical and psychological barriers keep the sides from speaking even one word to each other" as negative due to the sentiment expressed in its content.

The sentence communicates a sense of helplessness and stagnation, emphasizing that "nothing can change" because of existing "physical and psychological barriers." This statement portrays a bleak and frustrating scenario where communication and progress are impeded, suggesting a negative outlook on the situation described. The focus on barriers preventing dialogue implies a lack of hope or improvement, reinforcing the negative sentiment of the statement.



Από την συζήτηση μας με το *GPT* είναι εμφανές ότι έδωσε μεγαλύτερη βαρύτητα στην φράση **nothing can change** και λιγότερο στο συγκεκριμένο στο οποίο χρησιμοποιείται αυτός ο χαρακτηρισμός, δηλαδή δηλαδή ότι αναφέρεται στις υποσχέσεις, για τις οποίες η σταθερότητα είναι κάτι θετικό.

## Κυρίως Μέρος

Στα ερωτήματα 1-5 χρησιμοποιήσαμε το *MR Dataset* και το *glove.6B.50d* για τα *embeddings*.

### Ερώτημα 1

1.1 Με τον παρακάτω κώδικα, στην μέθοδο **forward** της κλάσης **BaselineDNN**, υπολογίζουμε την αναπαράσταση κάθε πρότασης ως συνένωση του *mean pooling* και του *max pooling*. Επίσης πρέπει να τροποποιήσουμε το *input layer* του δικτύου καθώς τώρα τα *embeddings* έχουν διπλάσιο μέγεθος από το αρχικό.

```
# embed the words, using the embedding layer
embeddings = self.embeddings(x)

# Compute mean pooling
mean_pooling = torch.mean(embeddings, dim=1)

# Compute max pooling
max_pooling, _ = torch.max(embeddings, dim=1)

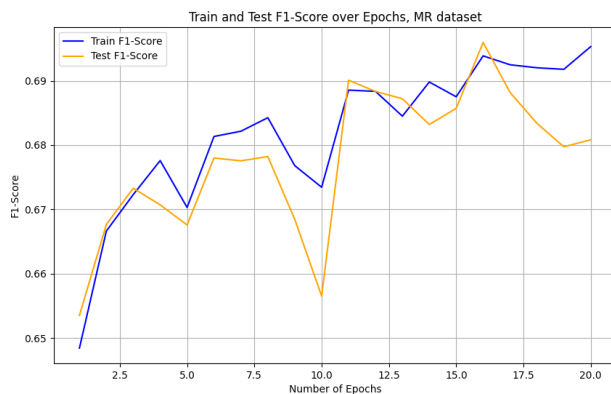
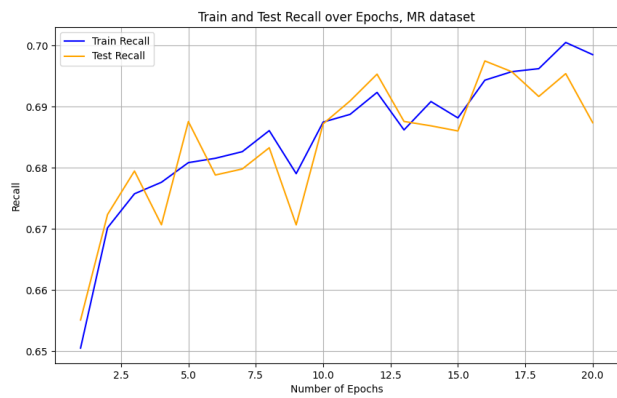
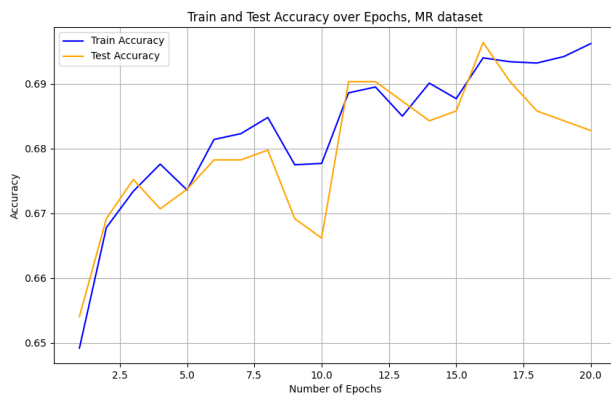
# Concatenate mean pooling and max pooling
representations = torch.cat((mean_pooling, max_pooling), dim=1)

# Transform the representations
representations = self.relu(self.linear(representations))

logits = self.output(representations)

return logits
```

Παρακάτω φαίνονται τα *accuracy*, *recall* και *F1 score* που πετυχαίνουμε χρησιμοποιώντας την προαναφερθείσα μέθοδο για τα *embeddings* στα *train* και *test* σύνολα:



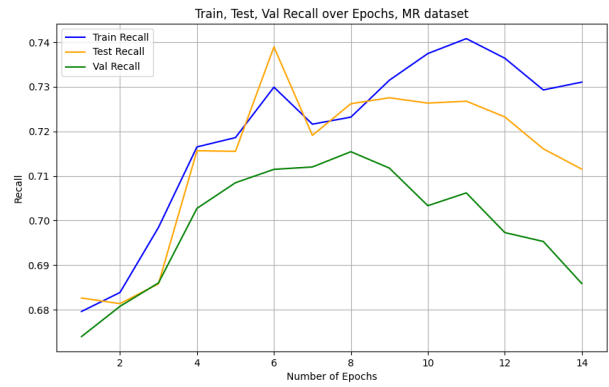
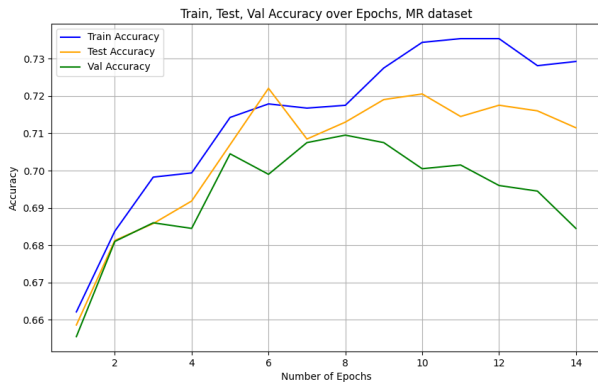
**1.2** Η αρχική αναπαράσταση βασίζεται αποκλειστικά *mean pooling*, που υπολογίζει το μέσο *embedding* διάνυσμα όλων των λέξεων της πρότασης. Ενώ το *mean pooling* αποτυπώνει τη συνολική σημασιολογική ουσία της πρότασης, ενδέχεται να μην αναδεικνύει συγκεκριμένες λέξεις ή χαρακτηριστικά που είναι ιδιαίτερα σημαντικά. Αντίθετα, με τη συμπερίληψη και του *max pooling* μαζί με το *mean pooling*, η αναπαράσταση τώρα μπορεί να συλλάβει τόσο τη μέση πληροφορία όσο και τα πιο σημαντικά χαρακτηριστικά των *embeddings*. Το *max pooling* προσδιορίζει τη μέγιστη τιμή σε κάθε διάσταση των ενσωματώσεων, αναδεικνύοντας αποτελεσματικά τα πιο σημαντικά χαρακτηριστικά που υπάρχουν στην πρόταση.

## Ερώτημα 2

**2.1** Στο αρχείο `main.py`, πριν από το *for loop* που γίνεται *train* το μοντέλο σπάμε το *train set* σε *train* και *val* και αρχικοποιούμε επίσης έναν **EarlyStopper** σύμφωνα με την εκφώνηση.

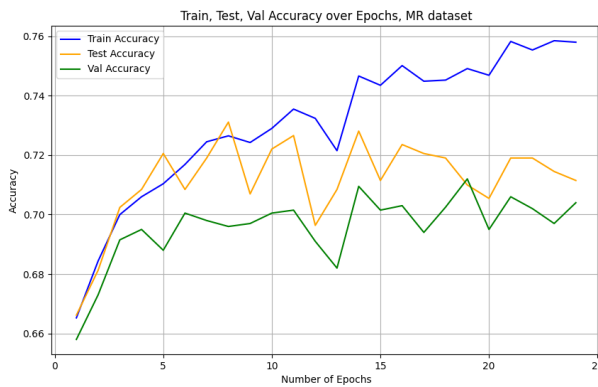
**2.2** Τροποίησαμε κατάλληλα την μέθοδο `forward` του `models.LSTM` έτσι ώστε να χρησιμοποιούμε την τελευταία έξοδο  $h_n$  του *LSTM* ως την αναπαράσταση του κειμένου  $u$ .

Παρακάτω φαίνονται τα *accuracy*, *recall* και *F1score* που πετυχαίνουμε στα *train*, *val* και *test* σύνολα:



**2.3** Θέτουμε τώρα την παράμετρο του μοντέλου **bidirectional = True**. Ένα αμφίδρομο *RNN* αποτελείται από τον συνδυασμό δύο διαφορετικών *RNN*, όπου το κάθε ένα επεξεργάζεται την ακολουθία με διαφορετική φορά.

Παρακάτω φαίνονται τα *accuracy*, *recall* και *F1score* που πετυχαίνουμε στα *train*, *val* και *test* σύνολα χρησιμοποιώντας αμφίδρομο *RNN*:



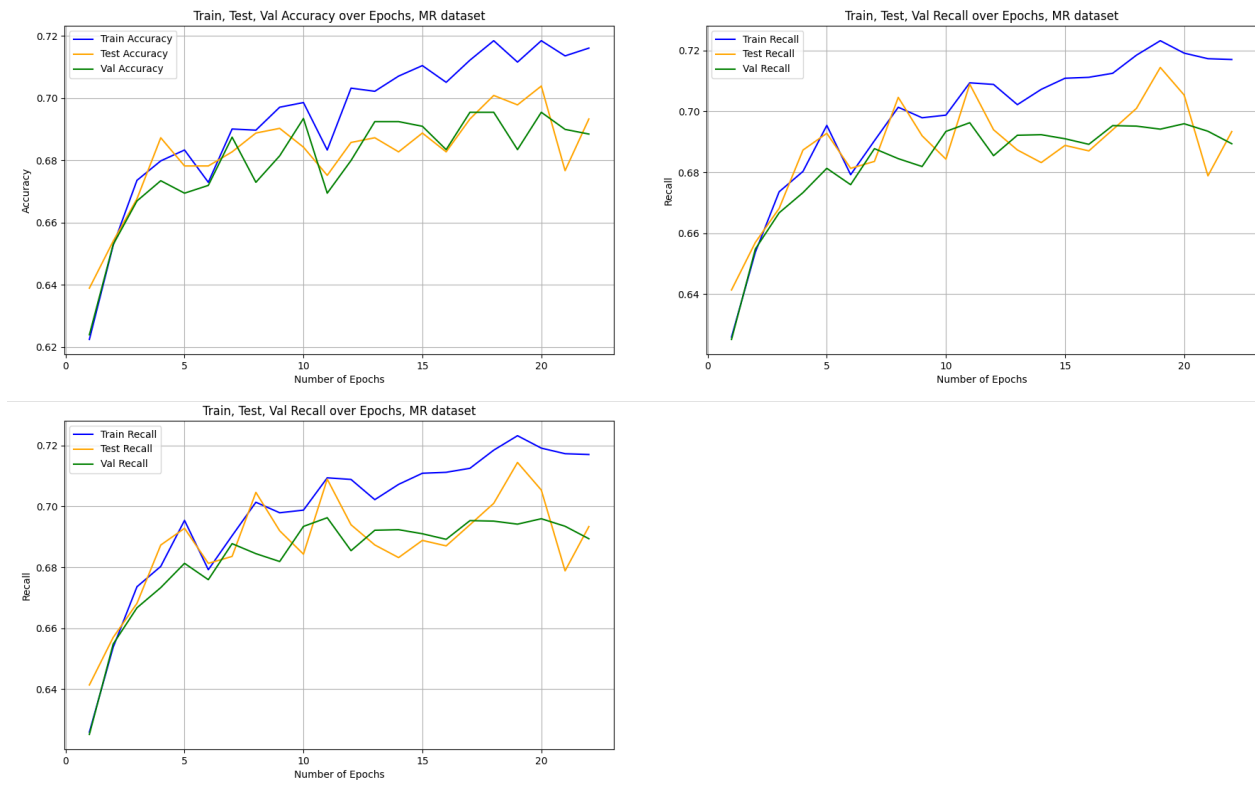
Παρατηρούμε ότι οι επιδόσεις του αμφίδρομου *RNN* είναι καλύτερες από αυτές του απλού *RNN*.



## Ερώτημα 3

**3.1** Στην κλάση *SimpleSelfAttentionModel* συμπληρώνουμε τα κενά του κώδικα έτσι ώστε να καθορίσουμε το *output* του μοντέλου και να εφαρμόσουμε *average – pooling* στις αναπαραστάσεις των λέξεων πριν το τελευταίο *layer*, για να εξάγουμε την αναπαράσταση της πρότασης.

Παρακάτω φαίνονται τα *accuracy*, *recall* και *F1score* που πετυχαίνουμε στα *train*, *val* και *test* σύνολα χρησιμοποιώντας *SimpleSelfAttention*:



Παρατηρούμε ότι οι επιδόσεις του *SimpleSelfAttention* είναι παρόμοιες με αυτές του απλού *RNN*.

**3.2** Στους μηχανισμούς προσοχής τα *key*, *query*, *value* είναι τα θεμελιώδη συστατικά που επιτρέπουν στο μοντέλο να εστιάζει σε σχετικές πληροφορίες κατά την διαδικασία του υπολογισμού. Τα *query* αναπαριστούν το τρέχον *input* ή *context* που υποβάλλεται για επεξεργασία, τα *keys* λειτουργούν ως ένα σύνολο *features* με τα οποία συγκρίνεται το ερώτημα για τον προσδιορισμό της συνάφειας και τα *values* περιέχουν την πληροφορία που σχετίζεται με κάθε κλειδί και χρησιμεύει ως η έξοδος η οποία θα "προσεχθεί" με βάση την ομοιότητα *query* και *key*.

Ισχύει ότι:

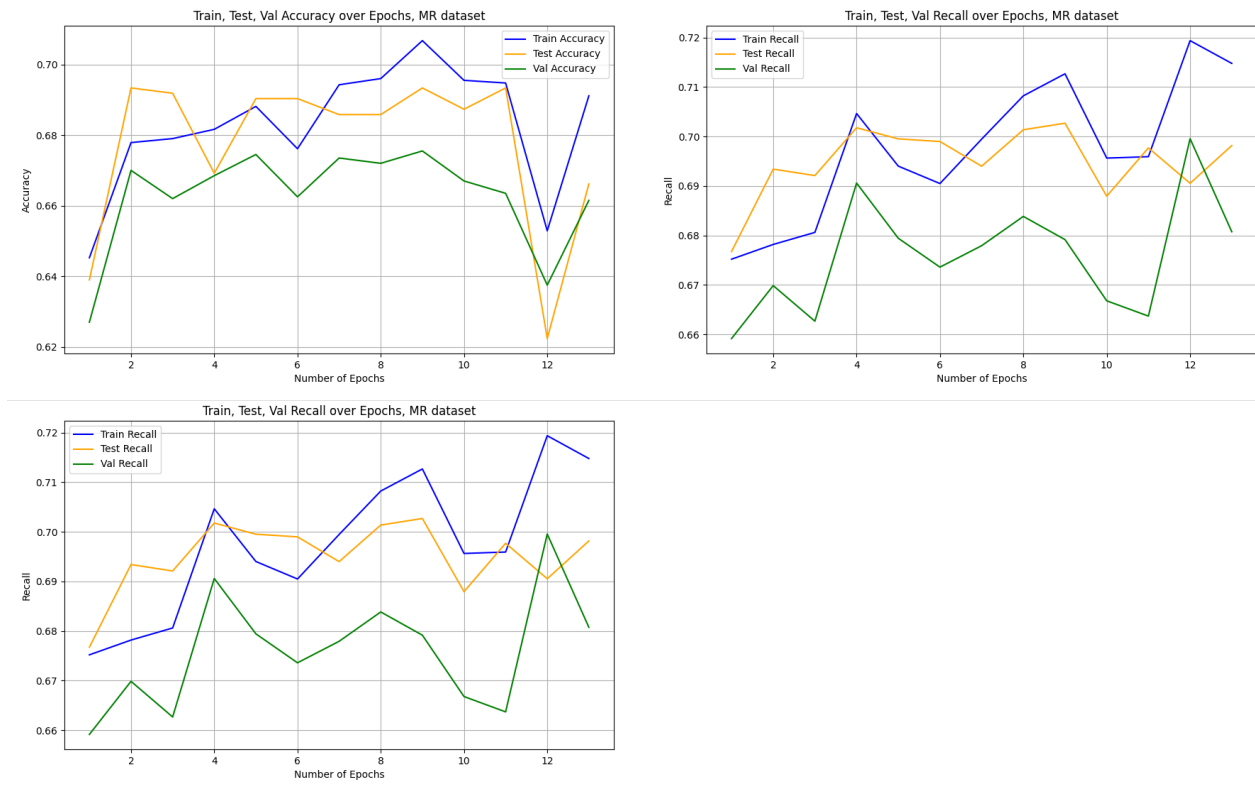
$$Attention(Q, K, V) = \text{softmax} \left( \frac{Q^T K}{\sqrt{d_k}} \right) V$$

Τα *Positional Embeddings* είναι ένα διάνυσμα που προστίθεται στα *embeddings* της εισόδου και κωδικοποιεί την θέση ή σειρά των *token* της εισόδου.

## Ερώτημα 4

Με την χρήση *MultiHeadAttentionModel* το μοντέλο αποσυνθέτει τον μηχανισμό προσοχής σε πολλά παράλληλα *attention layers* και με αυτό τον τρόπο μπορεί να εστιάζει σε διαφορετικά σημεία του *input* κάθε φορά και να συνδυάζει τα αποτελέσματα.

Παρακάτω φαίνονται τα *accuracy*, *recall* και *F1score* που πετυχαίνουμε στα *train*, *val* και *test* σύνολα χρησιμοποιώντας *MultiHeadAttentionModel*:



Παρατηρούμε ότι οι επιδόσεις του *MultiHeadAttentionModel* είναι χειρότερες από αυτές του *SingleSelf AttentionModel*.

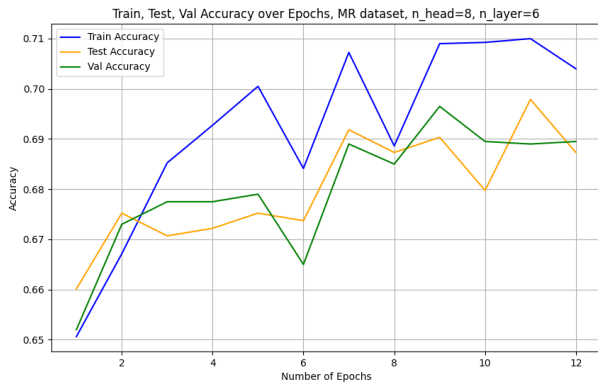
## Ερώτημα 5

Στο *MultiHeadAttentionModel* έχουμε έναν μόνο μηχανισμό *multi-head self-attention*, ενώ στο *Transformer Encoder Model* έχουμε ένα *stack* από *transformer blocks* κάθε ένα από τα οποία περιέχει έναν μηχανισμό *multi-head self-attention*.

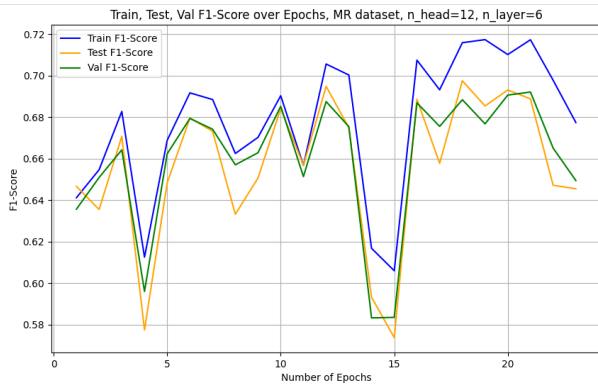
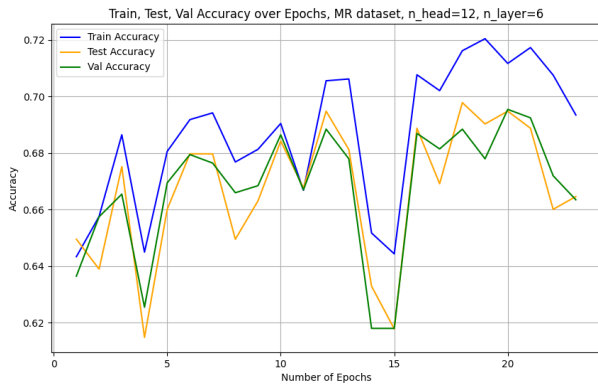
Οι *default* παράμετροι στην κλασική αρχιτεκτονική του *Transformer* είναι  $n\_head = 8$ ,  $n\_layers = 6$  και  $max\_length = 512$ .

Παρακάτω φαίνονται τα αποτελέσματα στα πειράματα που κάναμε:

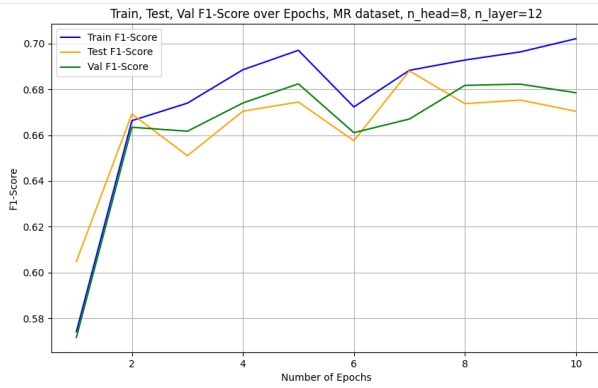
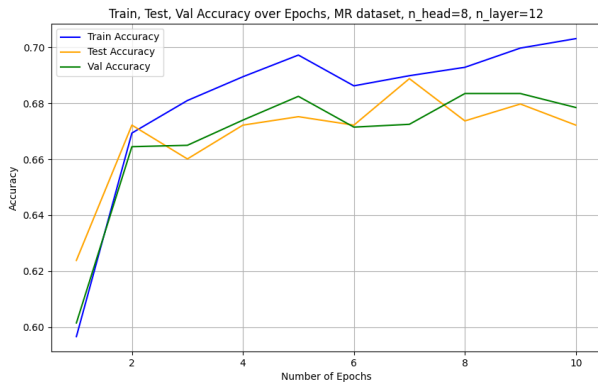
$n\_head = 8$ ,  $n\_layers = 6$ :



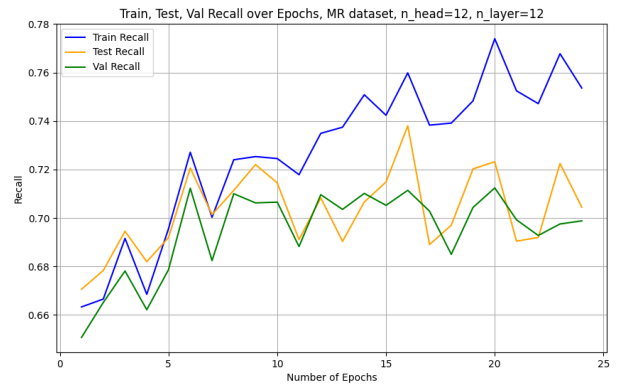
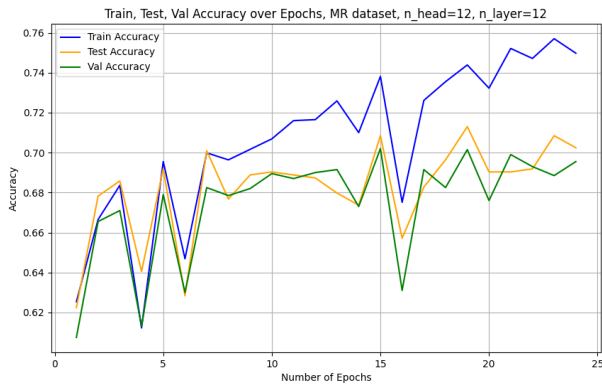
$n\_head = 12, n\_layers = 6:$



$n\_head = 8, n\_layers = 12:$



$n\_head = 12$ ,  $n\_layers = 12$ :



Παρατηρούμε ότι τις καλύτερες επιδόσεις τις πετυχαίνουμε για  $n\_head = 12$  και  $n\_layers = 12$

## Ερώτημα 6

Για το *MR Dataset*: Παρατηρούμε ότι την καλύτερη επίδοση την έχει το *siebert/sentiment – roberta – large – english*.

Πίνακας 1: *MR Dataset, Pre – trained Models, επιδόσεις*

<i>Model</i>	<i>Accuracy</i>	<i>Recall</i>	<i>F1_score</i>
<i>siebert/sentiment – roberta – large – english</i>	0.92598	0.92598	0.92598
<i>distilbert – base – uncased – finetuned – sst – 2 – english</i>	0.89123	0.89121	0.89123
<i>juliensimon/reviews – sentiment – analysis</i>	0.70695	0.70695	0.69239

Για το *Semeval Dataset*:

Πίνακας 2: *Semeval Dataset, Pre – trained Models, επιδόσεις*

<i>Model</i>	<i>Accuracy</i>	<i>Recall</i>	<i>F1_score</i>
<i>cardiffnlp/twitter – roberta – base – sentiment</i>	0.72379	0.72295	0.72221
<i>finiteautomata/bertweet – base – sentiment – analysis</i>	0.71776	0.73019	0.71805
<i>jProsusAI/finbert</i>	0.52206	0.39569	0.36083

Παρατηρούμε ότι την καλύτερη επίδοση την έχει το *cardiffnlp/twitter – roberta – base – sentiment*.

## Ερώτημα 7

Για το *MR Dataset*:

Πίνακας 3: *MR Dataset, Fine – tuned Pre – trained Models, επιδόσεις*

<i>Model</i>	<i>Accuracy</i>
<i>siebert/sentiment – roberta – large – english</i>	0.850
<i>distilbert – base – uncased – finetuned – sst – 2 – english</i>	0.875
<i>juliensimon/reviews – sentiment – analysis</i>	0.675

Παρατηρούμε ότι την καλύτερη επίδοση την έχει το *siebert/sentiment – roberta – large – english* και ότι οι επιδόσεις είναι λίγο χειρότερες από αυτές στο ερώτημα 6.

Για το *Semeval Dataset*:

Πίνακας 4: *Semeval Dataset, Fine – tuned Pre – trained Models, επιδόσεις*

<i>Model</i>	<i>Accuracy</i>
<i>cardiffnlp/twitter – roberta – base – sentiment</i>	0.675
<i>finiteautomata/bertweet – base – sentiment – analysis</i>	0.650
<i>jProsusAI/finbert</i>	0.450

Παρατηρούμε ότι την καλύτερη επίδοση την έχει το *cardiffnlp/twitter – roberta – base – sentiment* και ότι οι επιδόσεις είναι λίγο χειρότερες από αυτές στο ερώτημα 6.

## Ερώτημα 8

Παρακάτω φαίνονται οι ερωτήσεις που κάναμε στο *ChatGPT* και οι απαντήσεις που πήραμε:

can you explain the above code?

ChatGPT

Sure, let's break down the provided code:

- Importing Libraries:** The code imports necessary libraries including PyTorch for building and training neural networks.
- Setting Hyperparameters:** Hyperparameters such as batch size, block size, number of layers, etc., are defined. These parameters control various aspects of the model and training process.
- Loading Data:** The code reads a text file ("input.txt") which contains some text data. It then creates a mapping between characters and integers to encode and decode the text data.
- Data Splitting:** The text data is split into training and validation sets.
- Defining Model Components:**
  - Head:** This class represents one head of self-attention mechanism.
  - MultiHeadAttention:** This class represents multiple heads of self-attention mechanism in parallel.
  - FeedForward:** This class represents a simple linear layer followed by a non-linearity.
  - Block:** This class represents a Transformer block which includes multi-head self-attention and feed-forward layers.
  - BigramLanguageModel:** This class defines the main language model which consists of embedding layers, Transformer blocks, and a linear layer for prediction.
- Model Training:** The training loop is defined where the model is trained using the defined hyperparameters. In each iteration, a batch of data is sampled, and the model's parameters are updated based on the computed loss using backpropagation.
- Model Evaluation:** The model's performance is evaluated periodically on both the training and validation sets. The loss is computed for each set and printed.
- Model Generation:** Finally, the trained model is used to generate text. A context tensor is initialized, and then the model generates a sequence of tokens by predicting the next token at each step based on the previous context.

In summary, this code implements a simple Transformer-based language model trained on a text dataset and demonstrates text generation using the trained model.

would you make any changes in the code?

ChatGPT

Certainly! Here are some potential changes and improvements that could be made to the code:

- Error Handling:** Implement error handling to gracefully handle potential issues such as file loading errors or data processing failures. This could involve using try-except blocks and providing informative error messages.
- Documentation:** Add docstrings to classes and functions to provide clear explanations of their purpose, inputs, and outputs. This improves code readability and helps other developers understand how to use and extend the code.
- Testing:** Write unit tests for individual components and integration tests for the entire model to ensure correctness and reliability. This can be done using frameworks like 'unittest' or 'pytest'.
- Hyperparameter Tuning:** Experiment with different hyperparameter settings to find the optimal configuration for the model. This could involve conducting hyperparameter searches using techniques like random search or grid search.
- Learning Rate Scheduling:** Implement learning rate scheduling techniques such as exponential decay or cosine annealing to adjust the learning rate during training. This can help improve convergence and model performance.
- Early Stopping:** Add early stopping criteria based on validation loss to prevent overfitting and terminate training when the model starts to deteriorate on the validation set.
- Model Regularization:** Consider adding regularization techniques such as weight decay or dropout to prevent overfitting and improve generalization performance.
- Logging:** Introduce logging statements to track the training progress, including metrics such as training loss, validation loss, and any other relevant information.
- Model Checkpointing:** Save model checkpoints periodically during training to allow for easy recovery in case of unexpected interruptions or to resume training from a specific point.
- Code Refactoring:** Review the code for potential refactoring opportunities to improve code organization, clarity, and efficiency. This could involve restructuring classes or functions, removing redundant code, or optimizing performance-critical sections.

By incorporating these changes, the code can become more robust, maintainable, and effective for training and evaluating Transformer-based language models.

You

What about the quality of this code?

ChatGPT

The code provided demonstrates the implementation of a Transformer-based language model for text generation. Here's an assessment of its quality:

- Readability:** Overall, the code is well-structured and easy to follow. Variable names are descriptive, and comments are provided to explain key parts of the code.
- Modularity:** The code is modularized into several classes representing different components of the Transformer model (e.g., 'Head', 'MultiHeadAttention', 'FeedForward', 'Block', and 'BigramLanguageModel'). This makes the code easy to understand and maintain.
- Efficiency:** The code efficiently utilizes PyTorch's capabilities for defining neural network architectures and performing training. It leverages GPU acceleration if available, which is crucial for training large models on large datasets.
- Error Handling:** The code lacks comprehensive error handling mechanisms. For instance, it assumes that the input text file ('input.txt') exists and can be read without any issues. More robust error handling could be implemented to handle potential errors during data loading and processing.
- Documentation:** While the code includes comments to explain various parts, additional documentation, especially docstrings for classes and functions, would improve its readability and maintainability.
- Testing:** The code lacks explicit testing procedures. Adding unit tests to verify the correctness of individual components and integration tests to ensure the overall functionality of the model would enhance its reliability.
- Hyperparameters:** The hyperparameters are defined at the beginning of the code, making it easy to modify and experiment with different settings. However, providing explanations or rationale behind the choice of hyperparameters would be beneficial for understanding and reproducibility.
- Training Loop:** The training loop is well-defined, including the optimization step and periodic evaluation of the model's performance. However, it could benefit from additional features such as learning rate scheduling and early stopping to improve training efficiency and prevent overfitting.

In conclusion, while the code demonstrates the implementation of a Transformer-based language model and is generally of good quality, there are areas where it could be further improved, such as error handling, documentation, testing, and training loop enhancements.

You

can you write the code for some of these changes?

write the regularization for example

< 2/2 >

ChatGPT

Sure! Let's add dropout regularization to the model. We'll add dropout layers after the multi-head self-attention and feed-forward layers within each transformer block.

Here's how you can modify the 'Block' class to include dropout regularization:

```
python
class Block(nn.Module):
    """ Transformer block: communication followed by computation """
    def __init__(self, n_embd, n_head, dropout=0.0):
        # n_embd: embedding dimension, n_head: the number of heads we'd like
        super().__init__()
        head_size = n_embd // n_head
        self.sa = MultiHeadAttention(n_head, head_size)
        self.ffwd = FeedForward(n_embd)
        self.ln1 = nn.LayerNorm(n_embd)
        self.ln2 = nn.LayerNorm(n_embd)
        self.dropout = nn.Dropout(dropout) # Adding dropout layer

    def forward(self, x):
        # Apply multi-head self-attention with dropout
        x = x + self.dropout(self.sa(self.ln1(x)))
        # Apply feed-forward layer with dropout
        x = x + self.dropout(self.ffwd(self.ln2(x)))
        return x
```

In this modification:

We added a dropout layer with a specified dropout rate in the '\_\_init\_\_' method of the 'Block' class.

We applied dropout after the multi-head self-attention and feed-forward layers within the 'forward' method of the 'Block' class.

You can then instantiate the 'Block' class with a specified dropout rate when creating the Transformer blocks in the 'BigramLanguageModel' class. For example:

```
python
class BigramLanguageModel(nn.Module):
    def __init__(self, dropout=0.0):
        super().__init__()
        # each token directly reads off the logits for the next token from a lookup table
        self.token_embedding_table = nn.Embedding(vocab_size, n_embd)
        self.position_embedding_table = nn.Embedding(block_size, n_embd)
        self.blocks = nn.Sequential(*(Block(n_embd, n_head=n_head, dropout=dropout) for _ in range(layers)))
        self.ln_f = nn.LayerNorm(n_embd) # final layer norm
        self.ln_head = nn.Linear(n_embd, vocab_size)
```

When creating an instance of the 'BigramLanguageModel' class, you can specify the dropout rate as an argument. For example, to use a dropout rate of 0.1:

```
python
model = BigramLanguageModel(dropout=0.1)
```

This modification adds dropout regularization to the model, which can help prevent overfitting and improve generalization performance.

Από τις παραπάνω εικόνες φαίνεται ότι το *ChatGPT* καταλαβαίνει πλήρως των κώδικα του *tutorial* και αξιολογεί ότι είναι καλά δομημένος. Φαίνεται να καταλαβαίνει ότι είναι κάπως απλοϊκός και για αυτό προτείνει διάφορες αλλαγές που μπορούν να γίνουν. Τέλος ο κώδικας που προτείνει φαίνεται να λειτουργεί σωστά.