

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ

Επεξεργασία Φωνής και Φυσικής Γλώσσας



Προπαρασκευή 3ου Εργαστηρίου:

Αναφορά

Μάρκος Γιαννόπουλος 03118103

Νικόλαος Καραφύλλης 03119890

Ιούνιος 2023

ΠΡΟΠΑΡΑΣΚΕΥΗ

Περιβάλλον Ανάπτυξης

Κατεβάζουμε τον φάκελο lab3 με την εντολή `svn checkout`

<https://github.com/slp-ntua/slp-labs/trunk/lab3>.

Κατεβάζουμε στον φάκελο /embeddings τα προεκπαιδευμένα word embeddings GloVe. (<https://nlp.stanford.edu/data/glove.6B.zip>).

Δημιουργούμε και ενεργοποιούμε το περιβάλλον conda slp3.

Εγκαθιστούμε το pytorch καθώς και τις υπόλοιπες απαραίτητες βιβλιοθήκες με την εντολή `pip install -r requirements.txt`.

Τρέχουμε την `main.py` με είσοδο από το `terminal` το όνομα του μοντέλου που θέλουμε.

Για την προπαρασκευή “BaselineDNN”.

1 Προεπεξεργασία Δεδομένων

Θα χρησιμοποιήσουμε τα embedding διάστασης 50, που βρίσκονται στο αρχείο glove.6B.50d.txt.

1.1 Κωδικοποίηση Επισημειώσεων (Labels)

Κωδικοποιούμε τα labels του train και του test set από κείμενο ('positive', 'negative') σε νούμερα (0,1) με τον LabelEncoder.

Ζητούμενο 1:

Τυπώνουμε τα πρώτα 10 labels από τα δεδομένα εκπαίδευσης(από το MR dataset) και την αντιστοίχιση(κωδικοποίηση) τους σε αριθμούς.

[illegible]

Για το Semeval2017A:

```
Label: negative, Number: 0
Label: negative, Number: 0
Label: neutral, Number: 1
Label: neutral, Number: 1
Label: neutral, Number: 1
Label: neutral, Number: 1
Label: neutral, Number: 1
Label: positive, Number: 2
Label: positive, Number: 2
Label: positive, Number: 2
```

1.2 Λεκτική Ανάλυση (Tokenization)

Κάνουμε tokenization με το word_tokenize της βιβλιοθήκης nltk σε όλα τα δεδομένα του dataset. (γίνεται αφαίρεση κάποιων stopwords, ώστε να κρατήσουμε τα πιο ουσιαστικά tokens κάθε δείγματος)

Ζητούμενο 2:

Τυπώνουμε τα πρώτα 10 παραδείγματα token από τα δεδομένα εκπαίδευσης (MR dataset).

Example 1: ['rock', 'destined', '21st', 'century', 'new', 'conan', 'going', 'make', 'splash', 'even', 'greater', 'arnold', 'schwarzenegger', 'van', 'damme', 'steven', 'segal']

Example 2: ['gorgeously', 'elaborate', 'continuation', 'lord', 'ring', 'trilogy', 'huge', 'column', 'word', 'adequately', 'describe', 'peter', 'jackson', 'expanded', 'vision', 'j', 'r', 'r', 'tolkien']

Example 3: ['effective', 'biopic']

Example 4: ['sometimes', 'like', 'go', 'movie', 'fun', 'wasabi', 'good', 'place', 'start']

Example 5: ['emerges', 'something', 'rare', 'issue', 'movie', 'honest', 'keenly', 'observed', 'doe', 'feel', 'like', 'one']

Example 6: ['film', 'provides', 'great', 'insight', 'neurotic', 'mindset', 'comic', 'even', 'reached', 'absolute', 'top', 'game']

Example 7: ['offer', 'rare', 'combination', 'entertainment', 'education']

Example 8: ['perhaps', 'picture', 'ever', 'made', 'ha', 'literally', 'showed', 'road', 'hell', 'paved', 'good', 'intention']

Example 9: ['steer', 'turn', 'snappy', 'screenplay', 'curl', 'edge', 'clever', 'want', 'hate', 'somehow', 'pull']

Example 10: ['take', 'care', 'cat', 'offer', 'refreshingly', 'different', 'slice', 'asian', 'cinema']

Για το Semeval2017A:

Example 1: ['forgot', 'sad', 'first', 'episode', '5th', 'season', 'dexter', 'depressing', 'dexter', 'darkpassenger']

Example 2: ['2', 'almost', 'far', 'away', 'wait', 'return', 'dexter']

Transformed example 3: (array([2038, 34277, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]), 1, 2)

Example 4: ['sometimes', 'like', 'go', 'movie', 'fun', 'wasabi', 'good', 'place', 'start']

Transformed example 4: (array([1072, 118, 243, 1006, 2906, 66408, 220, 242, 466,
0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0]), 1, 9)

Example 5: ['emerges', 'something', 'rare', 'issue', 'movie', 'honest', 'keenly', 'observed', 'doe',
'feel', 'like', 'one']

Transformed example 5: (array([12398, 646, 2349, 496, 1006, 6082, 23499, 4583,
14220,
999, 118, 49, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0]), 1, 12)

Semeval2017A

Example 1: ['forgot', 'sad', 'first', 'episode', '5th', 'season', 'dexter', 'depressing', 'dexter',
'darkpassenger']

Transformed example 1: (array([15476, 5280, 59, 1943, 5493, 196, 16588,
19321,
16588, 400001, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0]), 0, 10)

Example 2: ['2', 'almost', 'far', 'away', 'wait', 'return', 'dexter']

Transformed example 2: (array([233, 592, 373, 421, 2473, 499, 16588, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0]), 0, 7)

Example 3: ['topic', '3', 'episode', 'first', 'disk', 'dexter', 'please', 'hurry', 'netflix', '2nd', 'fitblog']

Transformed example 3: (array([6284, 315, 1943, 59, 8119, 16588, 3833,
13416,
31429, 2767, 400001, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0]), 1, 11)

Example 4: ['iheardavoicesay', 'finally', 'catching', 'dexter', 'season', '4', 'ha', 'almost',
'finished', 'season', '2', 'today', 'home', 'ill']

Transformed example 4: (array([400001, 1230, 8210, 16588, 196, 410, 8358,
592,
1027, 196, 233, 374, 164, 3119, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,

0]), 1, 14)

Example 5: ['downloaded', 'first', 'season', 'dexter', 'gon', 'na', 'start', 'watching', '2nd', 'episode', 'time', 'tweet', 'twittwit']

Transformed example 5: (array([16617, 59, 196, 16588, 49069, 6183, 466, 2642,

2767, 1943, 80, 43729, 400001, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0,

0]), 1, 13)

2 Μοντέλο

2.1 Embedding Layer

Ζητούμενο 4:

- *Γιατί αρχικοποιούμε το embedding layer με τα προ-εκπαιδευμένα word embeddings?*

Είναι υπολογιστικά ακριβό να εκπαιδεύσουμε δικά μας embeddings, οπότε χρησιμοποιούμε transfer learning μέσω των προεκπαιδευμένων μοντέλων που είναι άλλωστε εκπαιδευμένα σε μεγάλο όγκο δεδομένων (άρα έχουν καλύτερη γενίκευση και με περισσότερα αναγνωρίσιμα tokens) και μπορούν να αναπαραστήσουν σύνθετες σημασιολογικές σχέσεις μεταξύ των λέξεων.

- *Γιατί κρατάμε παγωμένα τα βάρη του embedding layer κατά την εκπαίδευση?*

Για να μην εκπαιδεύσουμε τις ήδη εκπαιδευμένες χρήσιμες αναπαραστάσεις. Γλιτώνουμε υπολογιστικό χρόνο (λιγότεροι εκπαιδευσιμοι παράμετροι) και αποφεύγουμε την υπερπροσαρμογή των βαρών του embedding layer (λόγω του μικρού training dataset που έχουμε διαθέσιμο).

2.2 Output Layer(s)

Ζητούμενο 5:

Γιατί βάζουμε μία μη γραμμική συνάρτηση ενεργοποίησης στο προτελευταίο layer; Τι διαφορά θα είχε αν είχαμε 2 ή περισσότερους γραμμικούς μετασχηματισμούς στη σειρά;

Για να μπορούμε να “πιάσουμε” πιο σύνθετες μη γραμμικές σχέσεις μεταξύ των embedding και classification layers και να μπορούμε να διαχωρίσουμε μη-γραμμικά διαχωρίσιμες κλάσεις. Αν είχαμε 2 ή περισσότερους γραμμικούς μετασχηματισμούς σε σειρά, θα ήταν σαν να είχαμε έναν γραμμικό μετασχηματισμό με συνδυασμένα βάρη, οπότε το μοντέλο θα εντόπιζε μόνο γραμμικές σχέσεις κατά την εκπαίδευση με αποτέλεσμα να έχει μικρότερη ακρίβεια στις προβλέψεις.

2.3 Forward pass

Ζητούμενο 6:

- *Αν θεωρήσουμε ότι κάθε διάσταση του embedding χώρου αντιστοιχεί σε μία αφηρημένη έννοια, μπορείτε να δώσετε μία διαισθητική ερμηνεία για το τι περιγράφει η αναπαράσταση που φτιάξατε (κέντρο-βάρους);*

Η αναπαράσταση που φτιάξαμε αντιστοιχεί μια πρόταση στον μέσο όρο των αναπαραστάσεων των λέξεων που την απαρτίζουν. Είναι η αναπαράσταση μιας από τις λέξεις(αφού είναι μέσος όρος και όχι μέση τιμή) που βρίσκεται πιο κοντά στο κέντρο βάρους των σημείων που ορίζει κάθε embedding λέξης της πρότασης. Είναι ένας μέσος όρος συνδυασμών αφηρημένων εννοιών.

- *Αναφέρετε πιθανές αδυναμίες της συγκεκριμένης προσέγγισης για να αναπαραστήσουμε κείμενα.*

Περιορισμός λόγω διαστάσεων

Η αναπαράσταση των κειμένων περιορίζεται στις διαστάσεις των embeddings, ενώ ένα κείμενο περιέχει πολύ περισσότερες αφηρημένες έννοιες από όσες μπορεί να έχει μια λέξη και επομένως χρειάζονται πολύ μεγαλύτερα embeddings.

*μάλλον λίγο ακραίος και όχι τόσο σημαντικός περιορισμός

Περιορισμός λόγω αραιών σημείων στο χώρο των αναπαραστάσεων

Λόγω του ότι χρησιμοποιούμε μέσο όρο, αν υπάρχουν λίγες διαφορετικές λέξεις στο κείμενο, υπάρχει ο κίνδυνος η μέση αναπαράσταση να απέχει αρκετά από το κέντρο βάρους.

Περιορισμός λόγω της αντιμετώπισης κάθε token ως ανεξάρτητο

, ενώ στην πραγματικότητα υπάρχουν χρησιμες συντακτικές και γενικότερα γλωσσικές συσχετίσεις. Δεν υπάρχει η έννοια της απόστασης(ούτε στην αναπαράσταση προτάσεων υπάρχει αλλά τουλάχιστον γνωρίζουμε ότι οι λέξεις είναι κοντά), οπότε αν επιμέναμε σε αυτή τη προσέγγιση θα ήταν λίγο πιο συνετό να ενώνουμε ιεραρχικά τα embeddings (π.χ. πρώτα να βρίσκαμε τα embeddings των προτάσεων από τις λέξεις, και μετά του κειμένου από αυτά)

3 Διαδικασία Εκπαίδευσης

3.1 Φόρτωση Παραδειγμάτων (DataLoaders)

Ζητούμενο 7:

Τι συνέπειες έχουν τα μικρά και μεγάλα mini-batches στην εκπαίδευση των μοντέλων;

Αν έχουμε μικρά mini-batches το μοντέλο μας θα εκπαιδευτεί(θα συγκλίνει) πιο γρήγορα λόγω των συχνών ανανεώσεων, αν και θα έχει πιο “θορυβώδεις” εκτιμήσεις για τις παραγώγους λόγω του μικρού αριθμού δειγμάτων στο batch. Από την άλλη μεριά, τα μεγάλα mini-batches έχουν καλύτερες εκτιμήσεις των παραγώγων(με αποτέλεσμα να έχουμε μια πιο

σταθερή σύγκλιση), είναι πιο αποτελεσματικά υπολογιστικά αν έχουμε δυνατότητες παραλληλισμού.

- Συνήθως ανακατεύουμε την σειρά των *mini-batches* στα δεδομένα εκπαίδευσης σε κάθε εποχή. Μπορείτε να εξηγήσετε γιατί;

Καλύτερη γενίκευση: εισάγοντας τυχαιότητα αποφεύγουμε πιθανή μεροληψία που μπορεί να υπάρχει στην σειρά των *batches*. Αποφεύγουμε τοπικά μέγιστα(αφού γίνεται κάτι σαν *stochastic gradient descent*).

3.2 Βελτιστοποίηση

Ζητούμενο 8: (Συμπληρώθηκε στο `main.py`)

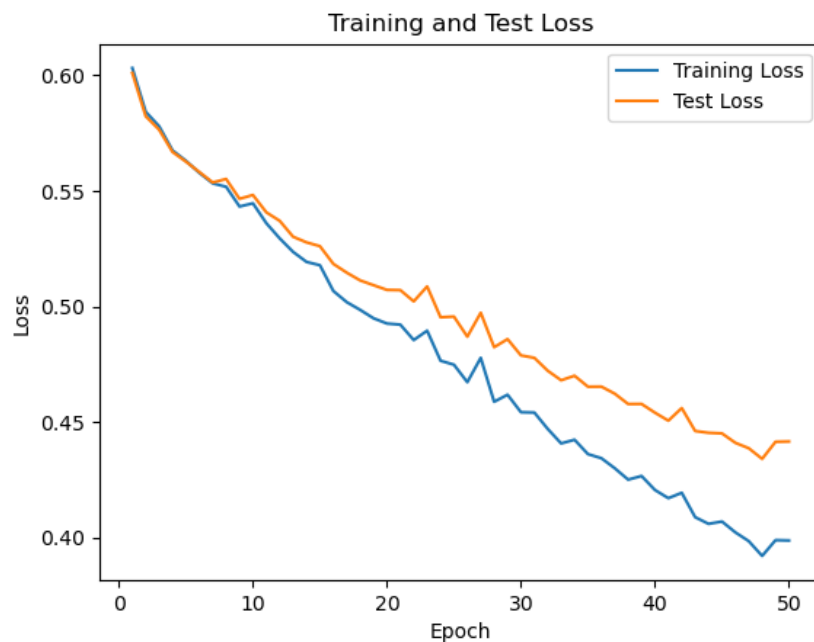
3.3 Εκπαίδευση

Ζητούμενο 9: (Συμπληρώθηκε στο `training.py`)

3.4 Αξιολόγηση

Ζητούμενο 10:

MR



Train accuracy: 0.816625

Test accuracy: 0.7943

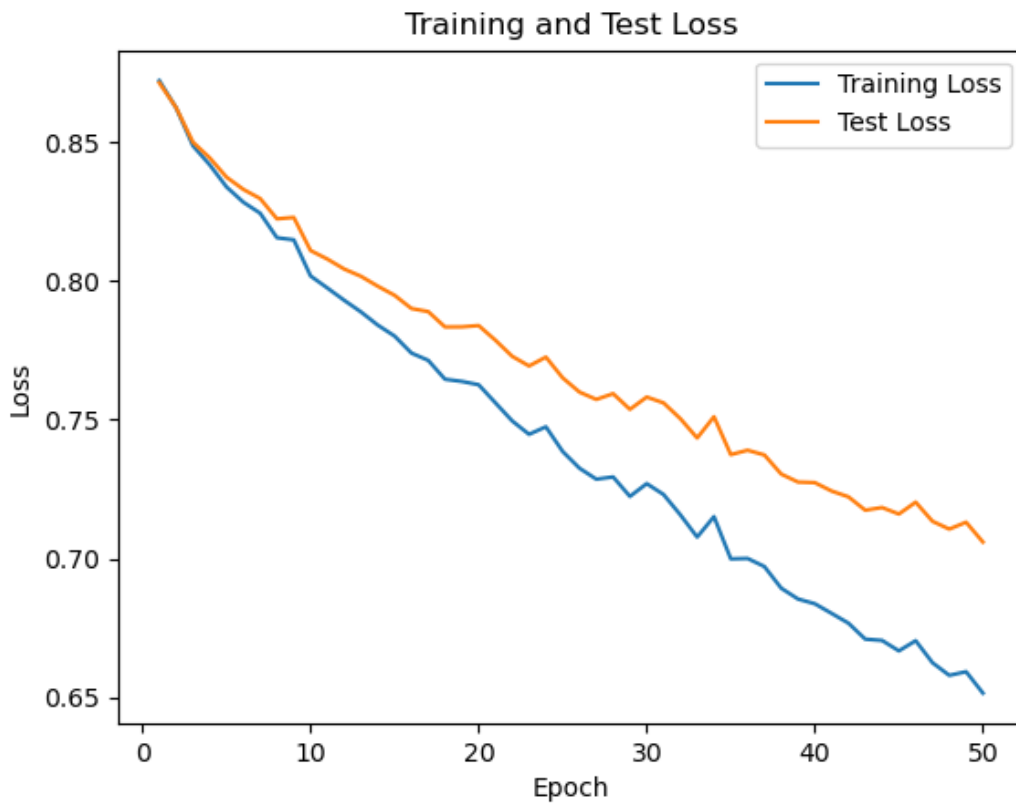
Train F1 score: 0.8147586675257597

Test F1 score: 0.792446495314785

Train Recall: 0.8280415855081922

Test Recall: 0.8052021251110935

Semeval2017A



Train accuracy: 0.7135616300181561

Test accuracy: 0.6826911438369981

Train F1 score: 0.6898968399919271

Test F1 score: 0.6565188597886948

Train Recall: 0.7025549974941181

Test Recall: 0.6682507165664768

4 Κατηγοριοποίηση με χρήση LLM

Ζητούμενο 11:

Για κάθε ένα από τα 2 datasets, επιλέξτε τουλάχιστον 20 κείμενα από κάθε κατηγορία και χρησιμοποιείτε το ChatGPT για την αναγνώριση του συναισθήματος. Δώστε κατάλληλες προτροπές (prompts) στο μοντέλο για τον παραπάνω σκοπό (πειραματιστείτε με διάφορες εναλλακτικές). Μετρήστε την επίδοση του μοντέλου. Έκανε κάποιο λάθος; Παρέχοντας κατάλληλες οδηγίες, προσπαθήστε να αντλήσετε πληροφορίες σχετικά με τους λόγους για τους οποίους το κάθε κείμενο ταξινομήθηκε στην αντίστοιχη κατηγορία. Ήταν κάποιες λέξεις σημαντικότερες από άλλες αναφορικά με το συναισθήμα της πρότασης;

Το prompt που χρησιμοποιήθηκε για το MR dataset:

classify each of the following sentences as positive or negative

Το ChatGPT ταξινόμησε λάθος τα παρακάτω:

23. Negative: effective but too-tepid biopic.

28. Negative: perhaps no picture ever made has more literally shown that the road to hell is paved with good intentions.

Άρα μπορούμε να πούμε ότι έχει ακρίβεια 95% στο MR (38/40)

Όσον αφορά τους λόγους που έγιναν οι εσφαλμένες ταξινομήσεις:

1. "Effective but too-tepid biopic."

Classification: Negative

The classification of this sentence as negative is based on the presence of the word "tepid," which implies a lack of enthusiasm or excitement. While the word "effective" suggests some level of success, the overall tone of the sentence leans towards a negative sentiment due to the use of "tepid."

1. "Perhaps no picture ever made has more literally shown that the road to hell is paved with good intentions."

Classification: Negative

The classification of this sentence as negative is due to the phrase "the road to hell is paved with good intentions." This phrase is commonly used to convey a negative sentiment, suggesting that even with good intentions, negative outcomes can result. Therefore, despite the use of the word "literally," which may indicate an interesting observation, the overall sentiment of the sentence is negative.

Το prompt που χρησιμοποιήθηκε για το Semeval2017A dataset:

classify each of the following sentences as neutral, positive or negative

Αρχικά φάνηκε να έχει 30% ακρίβεια, αλλά στη συνέχεια που δοκίμασα το prompt:

remove the words neutral, positive, negative in the following sentences and then classify each sentence as neutral, positive or negative, then return the classification accuracy

μου έδωσε το αποτέλεσμα

Total sentences: 40

Correctly classified: 38

Accuracy: 95%

(*αλλά πιστεύω ότι είναι biased)

Στη συνέχεια δικαιολογεί τα λάθη του:

Based on the provided original labels and the assigned classifications, there are two misclassifications:

. Sentence: "Gold edges down ahead of US jobs data: SINGAPORE (Reuters) - Gold edged lower on Friday, with investors waiting for... <http://t.co/CiqFonaf>"

Assigned Classification: Negative

Original Label: Neutral

. Sentence: "Here in the Philippines, It's November 2 and I was like where's my phone?! What is the time in LONDON?! #Excited #LittleThngs @NiallOfficial"

Assigned Classification: Positive

Original Label: Neutral

In the case of the first misclassification, the sentence discusses gold prices edging down ahead of US jobs data. It may be considered neutral in terms of sentiment. The misclassification could be due to the model's interpretation of the word "edged lower" as negative.

For the second misclassification, the sentence expresses excitement about the date and time. Although it doesn't explicitly convey a negative or positive sentiment, it is more neutral in nature. The misclassification could be attributed to the model's general understanding of hashtags like "#Excited" and "#LittleThngs" as positive indicators.

Το ChatGPT καταλήγει:

It's worth mentioning that sentiment analysis is not an exact science and can be influenced by context, interpretation, and the nuances of language.