University of Athens
Department of Informatics and Telecommunications

Deep Learning for NLP

Student name: *Βασιλική Τσαντήλα*

Course: *Artificial Intelligence II*
Semester: *Spring Semester 2025*

Περιεχόμενα

1. Abstract

Έχοντας ως δομικό στοιχείο ένα deep neural network, στόχος μας είναι η ανάπτυξη ενός sentiment classifier για ταξινόμηση συναισθήματος σε δοσμένο dataset από tweets.

2. Data processing and analysis

2.1 Pre-processing

Εκτυπώνουμε μερικά από τα δεδομένα μας:

```
0       @whoisralphie dude I'm so bummed ur leaving!
1       oh my god, a severed foot was found in a wheely bin in cobham!!! where
        they found is literally minutes from my house! feel sick now!
2       I end up "dog dialing" sometimes. What's dog dialing, u ask? My dogs
        will walk across my phone & end up calling someone.
3       @_rachelx meeeee toooooo!
4       I was hoping I could stay home and work today, but looks like I have to
        make another trip into town
148383 just love the jonas brothers its too bad i will never get to see them
        tears
148384 another day gone by....time is moving so fast...
148385 fuck college, i'm just gonna marry rich. : fuck college, i'm just gonna
        marry rich.
148386 ZOMGZ NEW SONG FTW. remember that night. <3
148387 http://twitpic.com/7mwrd - Arby's took down their Roastburger coupon.
        But i found the image in my browser cache...so...cheap lunch ...
```

Αναλύοντας τα δεδομένα μάς σκεφτόμαστε, ότι θα είναι χρήσιμο να μετατρέπουμε κάθε κεφαλαίο γράμμα σε πεζό, καθώς θέλουμε οι λέξεις π.χ. 'Good' και 'good' το μοντέλο να τις καταλάβει ως την ίδια λέξη (my_lower).
Επιπλέον, βλέπουμε πως οι προτάσεις μας αποτελούνται από stopwords, λέξεις που δεν προσδίδουν κάποια χρησιμότητα στο μοντέλο στην προσπάθειά του να ερμηνεύσει αν η πρόταση έχει θετικό ή αρνητικό πρόσημο (συναισθηματικά) (my_stopword).
Σκεφτόμαστε, πως μπορούμε να δοκιμάσουμε να αφαιρέσουμε τα σημεία στίξης, και να δούμε αν αυτό θα επιφέρει κάποια βελτίωση ή όχι στο μοντέλο (my_unpunct).
Συνειδητοποιούμε ότι το ίδιο ρήμα εμφανίζεται στο dataset σε διαφορετικές μορφές. Έτσι, σκεφτόμαστε ότι θα είναι χρήσιμο κάθε ρήμα να το μετατρέπουμε στη βασική του μορφή (my_lemmatize).
Αντιλαμβανόμαστε ότι στην εν λόγω άσκηση δεν υπάρχει κάποια απαίτηση για ανωνυμία και προστασία των προσωπικών δεδομένων, επομένως επιλέγουμε να αφήσουμε τυχόν usernames, e-mails κλπ ως έχουν, χωρίς κάποια επιπλέον επεξεργασία.
Σημειώνουμε, ότι η παραπάνω επεξεργασία εφαρμόζεται αφού έχει γίνει πρώτα tokenized το κείμενο (my_split).

Εφαρμόζοντας τις παραπάνω συναρτήσεις προ-επεξεργασίας, παρατηρούμε ότι πλέον τα δεδομένα μας έχουν την εξής μορφή:

```
0      whoisralphie dude  im bummed ur leaving
1      oh god severed foot foun wheely bin cobham found literally minute house
       feel sick now
2      end quotdog dialingquot sumtimes whats dog dialing u askmy dog walk
       across
       phone amp end calling someone aka  quotdog dialingquot
3      rachelx meeeee toooooo
4      hoping could stay home work today look like make another trip town
148383 love jonas brother  tooo bad wil never get see  tear
148384 another day gone bytime moving fast
148385 fuck college im gonna marry rich  fuck college im gonna marry rich
       httpbitlymgic4
148386 zomgz new song ftw  remember night lt3
148387 httptwitpiccom7mwrd  arbys took roastburger coupon     found image
       browser cachesocheap lunch
```

2.2 Data partitioning for train, test and validation

Τα δεδομένα μας ήταν ήδη από την εκφώνηση χωρισμένα σε train, validation και test set, συνεπώς δεν υπέστη κάποιο διαφορετικό διαχωρισμό από εμάς.

2.3 Word Embeddings

Για την μετατροπή κειμένου σε αριθμούς, χρησιμοποιούμε GloVe Embeddings, τα οποία μετά από από κατάλληλη μετατροπή στο format τους, παίρνουμε Word2Vec word embeddings. Έχοντας τα word embeddings των λέξεων των tweets, ο τρόπος που πηγαίνουμε στην αναπαράσταση ολόκληρων των tweets είναι παίρνοντας τον μέσο όρο των διανυσμάτων των λέξεων του tweet.
Όπως αναφέρουμε και παρακάτω πειραματιστήκαμε με διαφορετικές διαστάσεις των word vectors.

3. Experiments

Σημαντική παρατήρηση: Όπως φαίνεται και στο αντίστοιχο notebook, φροντίζουμε να χρησιμοποιούμε random seed για να διασφαλίσουμε το reproducibility. Η χρήση random seed περιορίζει τον μη ντετερμινισμό, ωστόσο δεδομένου ότι χρησιμοποιούμε gpu για την επιτάχυνση της εκτέλεσης του notebook, ο μη ντετερμινισμός δυστυχώς συνεχίζει και υπάρχει, αν και είναι σημαντικά πιο περιορισμένος από το να μην είχαμε χρησιμοποιήσει καθόλου random seed.
Παρατήρησα ότι η εκτέλεση των πειραμάτων έπαιρνε εξαιρετικά πολύ περισσότερο χρόνο, όταν αυτά εκτελούνταν μέσω kaggle απεναντίας με όταν εκτελούνταν τοπικά. Για αυτό και αποφάσισα να εργαστώ κυρίως τοπικά. Έτσι, το notebook που βλέπετε στο kaggle έγινε έπειτα από upload της τοπικής εκτέλεσης του ίδιου ακριβώς notebook στον υπολογιστή μου. Πραγματοποίησα πολλά πειράματα, προκειμένου να βρω το καλύτερο δυνατό μοντέλο. Παρακάτω παραθέτω τα πιο κρίσιμα πειράματα από αυτά, δίνοντας έτσι μια κατατοπιστική εικόνα για τον τρόπο που εργάστηκα.

3.1 Baseline model

3.1.1 Architecture

Στοχεύουμε να χρησιμοποιήσουμε ως baseline model ένα απλό μοντέλο αρχιτεκτονικά που πετυχαίνει accuracy στο validation set ίση με 70%.
Σκέφτομαι ότι από τα πιο απλά αρχιτεκτονικά μοντέλα, είναι αυτά που αποτελούνται από μόνο ένα hidden layer. Αναρωτιέμαι αν μπορώ να πετύχω accuracy ίσο με 70% με μόνο ένα hidden layer. Δοκιμάζω διαφορετικό πλήθος νευρώνων για αυτό το ένα hidden layer, όπως και διαφορετικές τιμές για τις υπερπαραμέτρους, και καταλήγω ότι το πείραμα που μου έδωσε το καλύτερο accuracy στο validation set ήταν το εξής:

Hidden layer	1 layer with 64 neurons
Learning rate	1e-4
Batch size	64
Optimizer	AdamW
Epochs	100
Loss function	BCELoss
Activation function	ReLU
GloVe dimension	50
Preprocessing	All functions enabled

Επειδή το accuracy που παίρνουμε στο validation set είναι: 0.687400, σκεφτόμαστε να αυξήσουμε το learning rate σε 1e-3. Τότε, βλέπουμε ότι στο validation set παίρνουμε accuracy = 0.686456, το οποίο είναι ελάχιστα χειρότερο. Δοκιμάζουμε και μια τελευταία απόπειρα για learning rate έναν αριθμό ανάμεσα στο 1e-4 και 1e-3 (learning rate = 5e-4) και βλέπουμε ότι στο validation set παίρνουμε accuracy = 0.689782, το οποίο είναι το καλύτερο αποτέλεσμα ως στιγμής (και το πιο κοντινό στο 70% accuracy που στοχεύαμε για το baseline model μας). Αποφασίζουμε να κρατήσουμε το learning rate = 5e-4.
Τελικά, το baseline model ως στιγμής έχει ως εξής:

Hidden layer	1 layer with 64 neurons
Learning rate	5e-4
Batch size	64
Optimizer	AdamW
Epochs	100
Loss function	BCELoss
Activation function	ReLU
GloVe dimension	50
Preprocessing	All functions enabled

Σημειώνουμε ότι εφαρμόζουμε την sigmoid συνάρτηση στα αποτέλεσματα του μοντέλου, καθώς τα αποτελέσματα μπορούν να έχουν οποιαδήποτε τιμή, και εμείς θέλουμε τα αποτελέσματα να βρίσκονται στο διάστημα [0,1], ώστε στη συνέχεια αυτές οι τιμές να γίνουν round και να δίνουν μόνο τα labels 0 ή 1.

3.1.2 Preprocessing

Ελέγχουμε αν το preprocessing βοηθά το baseline model να πετύχει καλύτερο accuracy στο validation set. Σημειώνουμε ότι σε κάθε μία από τις παρακάτω περιπτώσεις έχει επισημανθεί το καλύτερο accuracy στο validation set που σημειώθηκε σε διάστημα 100 εποχών. Παρατηρούμε ότι το καλύτερο αποτέλεσμα το παίρνουμε όταν η συνάρτηση my_stopword και

Preprocessing	Accuracy in val_set
disabled	0.6822577601660534 in epoch = 50
enabled all	0.6893810736861968 in epoch = 89
disabled stopword	0.701127464855175 in epoch = 50
disabled lemmatize	0.6905604302292669 in epoch = 89
disabled stopword, lemmatize	0.7026842154920275 in epoch = 91
disabled stopword, lemmatize, unpuct	0.6949948108312105 in epoch = 66

Table 1: Testing preprocessing in baseline model

my_lemmatize είναι απενεργοποιήμενες. Συνεπώς, στα πειράματα που ακολουθούν κατά τη διαδικασία του preprocessing έχουμε ενεργοποιημένες μόνο τις συναρτήσεις my_lower και my_unpunct. Στο τελικό μοντέλο (στο τέλος του report) ξανά τρέχουμε το συγκεκριμένο πείραμα, για να βεβαιώθουμε ότι και σε εκείνο το μοντέλο πράγματι πετυχαίνουμε καλύτερο accuracy στο validation set όταν οι συναρτήσεις my_stopword και my_lemmatize του preprocessing είναι απενεργοποιήμενες.

3.1.3 GloVe dimension

Ελέγχουμε αν οι διαφορετικές διαστάσεις των word vectors, επηρεάζουν την απόδοση του baseline model. Σε αντιστοιχία με το προηγούμενο πείραμα, σε κάθε μία από τις παρακάτω περιπτώσεις έχει επισημανθεί το καλύτερο accuracy στο validation set που σημειώθηκε σε διάστημα 100 εποχών.

Word vectors dimension	Accuracy in val_set
glove.6B.50d	0.7026842154920275 in epoch = 91
glove.6B100d	0.7360835927917728 in epoch = 86
glove.6B.200d	0.7508727238418719 in epoch = 98
glove.6B.300d	0.7584441928483819 in epoch = 25
glove.twitter.27B.25d	0.7101141617133692 in epoch = 88
glove.twitter.27B.50d	0.748183790923672 in epoch = 78
glove.twitter.27B.200d	0.7845787338428154 in epoch = 20

Table 2: Testing word vectors dimension in baseline model

Παρατηρούμε ότι με εξαιρετικά μεγάλη διαφορά τα glove embeddings glove.twitter.27B.200d πετυχαίνουν το καλύτερο accuracy στο validation set. Οπότε, έπεται πολύ λογικά ότι στα επόμενα πειράματα θα χρησιμοποιούμε τα glove embeddings glove.twitter.27B.200d.

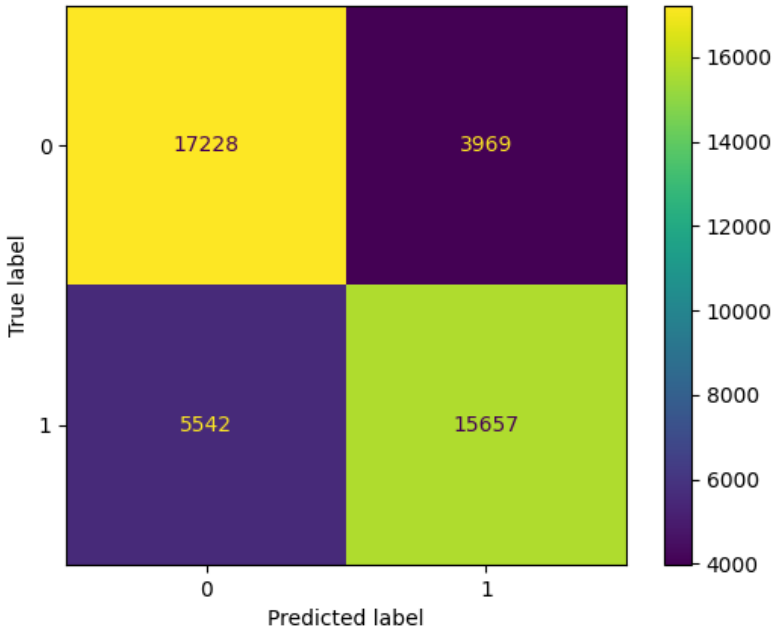
3.1.4 Baseline model evaluation

Συνοψίζοντας το baseline model έχει ως εξής:

Hidden layer	1 layer with 64 neurons
Learning rate	5e-4
Batch size	64
Optimizer	AdamW
Epochs	100
Loss function	BCELoss
Activation function	ReLU
GloVe dimension	27B.200d
Preprocessing	my_lower, my_unpunct only enabled

Αυτό είναι και το τελικό μας baseline model το οποίο δίνει 0.775663 accuracy στο validation set.

Καταγράφουμε κάποια επιπλέον στοιχεία για το baseline model μας, τα οποία φαίνονται παρακάτω.



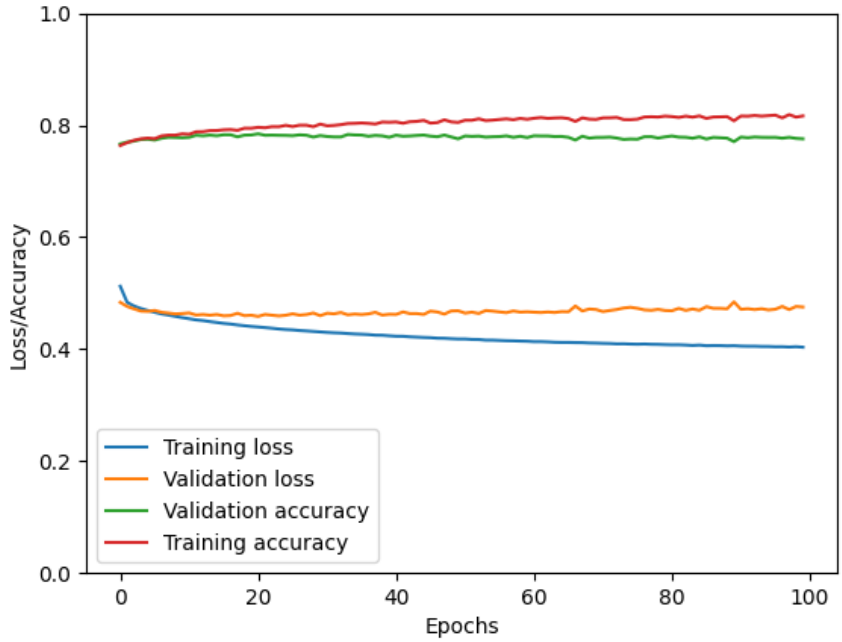
Σχήμα 1: Final baseline model: Confusion matrix

Class	Precision	Recall	F1-score	Support
0	0.76	0.81	0.78	21197
1	0.80	0.74	0.77	21199

Table 3: Final baseline model: Classification report

Καταρχάς από τις τιμές του support στο classification report για το κάθε label, καταλαβαίνουμε ότι το validation set είναι balanced (21197 ≈ 21199). Βλέπουμε ότι το μοντέλο μας συνολικά πετυχαίνει 78% accuracy, το οποίο είναι μια πολύ καλή τιμή για να θεωρήσουμε το συγκεκριμένο μοντέλο στην περαιτέρω ανάλυση μας ως baseline. Παρατηρούμε ότι για το label = 0, καταφέρνει 81% recall, προβλέποντας σωστά ως negative σημαντικό ποσοστό

τον negative instances. Το precision για το label = 0, είναι λίγο χειρότερο, με μόλις 76% επιτυχία ότι όσα πρόβλεψε ως negative ήταν πράγματι negative.
Αντίστοιχα, για το label = 1, παρατηρούμε περίπου παρόμοιο performance, με μόλις το recall να είναι 0.74% και το precision ίσο με 80%.
Επιβεβαιώνουμε αυτά τα αποτελέσματα και στον confusion matrix. Σε αυτόν βλέπουμε το πλήθος των σφαλμάτων του μοντέλου: 3.969 false positives και 5.542 false negatives.



Σχήμα 2: Final baseline model: Learning curve

Αναφορικά με τη καμπύλη μάθησης του μοντέλου: Το μοντέλο πετυχαίνει το καλύτερο accuracy στο validation set στην εποχή 20 (0.7845787338). Από εκείνη την εποχή και μετά το training accuracy αυξάνεται αλλά το validation accuracy μειώνεται. Καταλαβαίνουμε, δηλαδή, ότι το μοντέλο μας από την εποχή 20 και μετά κάνει overfit. Σκεφτόμαστε ότι η εφαρμογή ενός dropout_rate και η ελαφρώς αύξηση της πολυπλοκότητας του μοντέλου ενδεχομένως να επιφέρει βελτίωση.

3.2 From baseline model to final model

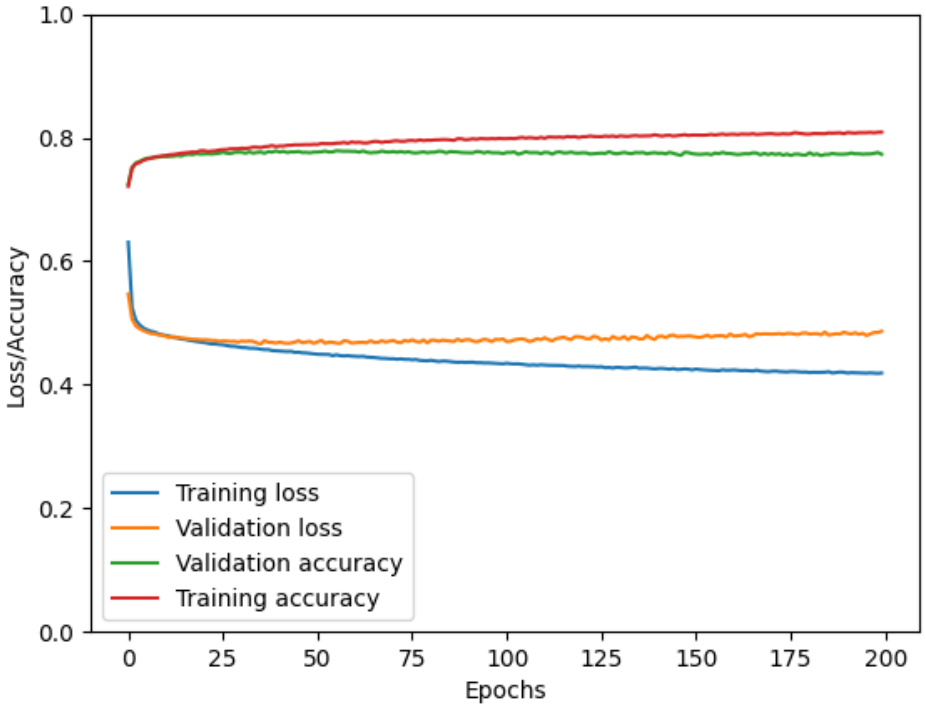
3.2.1 Manual tuning experiments

Έχοντας κατά νου τις δύο τελευταίες παρατηρήσεις της μελέτης μας, τρέχουμε πειράματα βασισμένοι στο baseline model εφαρμόζοντας, ωστόσο, πλέον dropout σε ένα ελαφρώς πιο complex αρχιτεκτονικά μοντέλο. Στόχος μας είναι η βελτίωση του baseline model. Τρέξαμε πολλά πειράματα, με τα πιο άξιας σημασίας να είναι τα εξής δύο:

- Πείραμα #1:

Number of neurons per hidden layer	[128, 256]
Learning rate	2e-4
Batch size	1024
Optimizer	AdamW
Epochs	200
Loss function	BCELoss
Activation function	ReLU
Dropout rate	0.5
GloVe dimension	27B.200d
Preprocessing	my_lower, my_unpunct only enabled

Το παρόν μοντέλο αποτελεί τροποποίηση του baseline model. Αποτελείται από 2 hidden layers αντί για 1. Έχει μικρότερο learning rate και αντίστοιχα μεγαλύτερο αριθμό εποχών. Το batch_size είναι μεγαλύτερο και επιπλέον υπάρχει εφαρμογή dropout_rate ίση με 0.5.



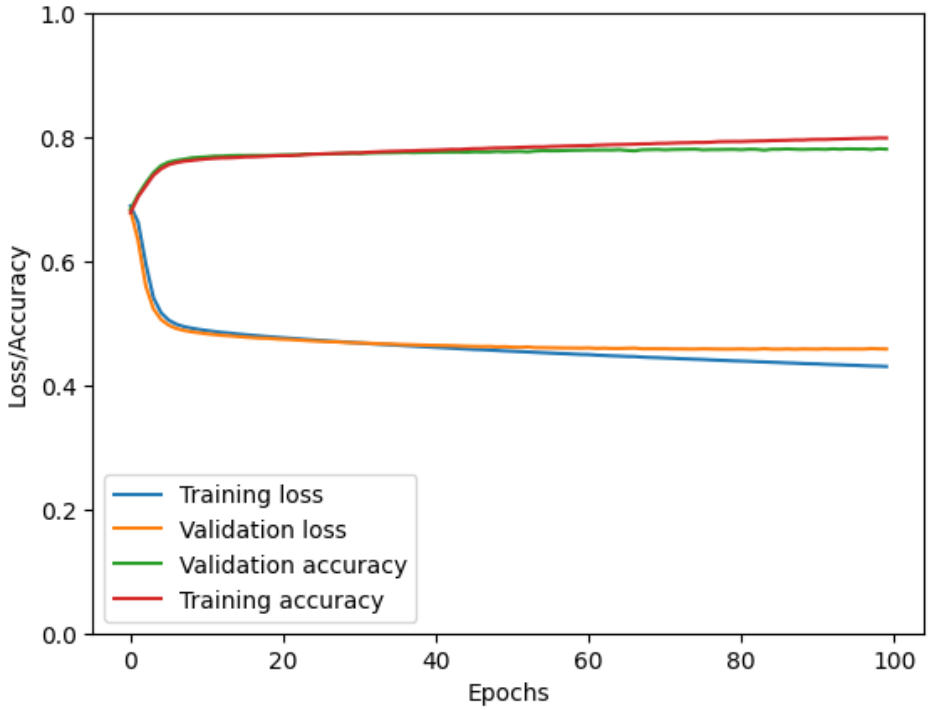
Σχήμα 3: Final model: experiment #1

Το καλύτερο accuracy που πετυχαίνει το παρόν μοντέλο στο validation set είναι 0.778799886 και συγκεκριμένα στην εποχή 55. Βλέπουμε ότι το παρόν μοντέλο δεν ξεπερνά το καλύτερο accuracy του baseline model (0.7845787338), παρόλο που βρίσκεται αρκετά κοντά σε αυτό. Επίσης, βλέπουμε ότι και το τρέχον μοντέλο κάνει overfitting.
Συνεχίζουμε τις προσπάθειες για κάτι καλύτερο.

- Πείραμα #2:

Number of neurons per hidden layer	[400, 200, 100]
Learning rate	2e-5
Batch size	1024
Optimizer	AdamW
Epochs	100
Loss function	BCELoss
Activation function	ReLU
Dropout rate	0.0
GloVe dimension	27B.200d
Preprocessing	my_lower, my_unpunct only enabled

Για αυτό το μοντέλο ορίσαμε 3 hidden layers (πιο complex αρχιτεκτονικά από το προηγούμενο μοντέλο), μειώσαμε το learning rate και τον αριθμό των εποχών, και αποφασίσαμε να μην εφαρμόσουμε καθόλου dropout_rate.



Σχήμα 4: Final model: experiment #2

Το καλύτερο accuracy που πετυχαίνει το παρόν μοντέλο στο validation set είναι 0.78250306 και συγκεκριμένα στην εποχή 83. Εκ νέου βλέπουμε ότι ούτε αυτό το μοντέλο ξεπερνά το καλύτερο accuracy του baseline model (0.7845787338), αν και είναι καλύτερο από το μοντέλο του πειράματος #1.

3.2.2 Optuna

Δεδομένου ότι ήδη δοκιμάσαμε διαφορετικές τιμές με το χέρι, αλλά δεν είχαμε καλύτερα αποτελέσματα από το baseline model, αποφασίζουμε να ορίσουμε trials, ώστε να τρέξουν με το optuna.

Τρέχουμε 2 optuna studies, τα οποία διαφέρουν μόνο στο πλήθος των hidden layers. Το 1ο optuna study τρέχει για μοντέλα με 1 hidden layer, ενώ το 2ο optuna study τρέχει για

μοντέλα με 3 hidden layers. Τα διαστήματα τιμών για κάθε υπερπαραμέτρο είναι ίδια και για τα 2 optuna studies. Συγκεκριμένα:

Hidden layer sizes [50, 100, 200, 300, 400]
Πλήθος νευρώνων που μπορεί να έχει το κάθε hidden layer.
Optimizers [AdamW, NAdam, SGD]
Επιλέγουμε να πειραματιστούμε με τρεις διαφορετικούς optimizers.
Epochs [10, 100, 300]
Το πλήθος των εποχών μπορεί να είναι αποκλειστικά ένας από τους τρεις αριθμούς.
Learning rate [1e-6, 1e-3]
Το εύρος τιμών για το learning rate είναι κλειστό διάστημα.
Dropout rate [0.0, 0.5]
Το εύρος τιμών για το dropout rate είναι επίσης κλειστό διάστημα.
Loss functions [BCELoss, BCEWithLogitsLoss]
Η επιλογή αυτών των loss functions έγινε δεδομένου ότι το μοντέλο μας πρόκειται για έναν binary classifier. Σημειώνουμε ότι η BCEWithLogitsLoss εφαρμόζει την sigmoid συνάρτηση, γι΄ αυτό και φροντίζουμε τα δεδομένα y_pred που δίνουμε στην BCEWithLogitsLoss να μην έχουν περάσει από την sigmoid συνάρτηση (ενώ για την BCELoss τα δεδομένα μας φροντίζουμε να έχουν περάσει από την sigmoid συνάρτηση).

Σημειώνουμε ότι οι παράμετροι που έχουν σταθερή τιμή και για τα 2 optuna studies, είναι:

Batch size 1024
Ο λόγος που θέτουμε μια μεγάλη τιμή στο batch_size, είναι κυρίως προκειμένου το training να ολοκληρώνεται πιο γρήγορα, και έτσι να λαμβάνουμε πιο γρήγορα τα αποτελέσματα του κάθε trial.
Activation function ReLU

Η objective function που έχουμε ορίσει για το κάθε optuna study στοχεύει να κάνει maximize το καλύτερο accuracy που έχει σημειώσει το μοντέλο στο validation set στο πέρασμα των εποχών. Παράλληλα, έχουμε φροντίσει μέσα στην objective function να έχουμε υλοποιήσει early stopping, προκειμένου να αποφεύγουμε το μοντέλο μας από το να κάνει overfit.

Δεν ορίσαμε συγκεκριμένο αριθμό trials προς εκτέλεση, αλλά σταματήσαμε εμείς το κάθε optuna study, όταν πλέον είχε τρέξει αρκετά trials και είχαμε βρει ένα ικανοποιητικό μοντέλο.

- Αποτελέσματα Optuna study #1:

Το καλύτερο μοντέλο που επέστρεψε το optuna study το οποίο έτρεχε για μοντέλα με μόνο 1 hidden layer ήταν το εξής:

Hidden layer	1 layer with 400 neurons
Learning rate	0.0006948140681022177
Batch size	1024
Optimizer	AdamW
Epochs	51 due to early stopping (initial epochs = 100)
Loss function	BCELoss
Activation function	ReLU
Dropout rate	0.4529273673532248

Το παρόν μοντέλο πέτυχε το καλύτερο accuracy στο validation set την εποχή 46 με τιμή 0.7895792055854326.

- Αποτελέσματα Optuna study #2:

Το καλύτερο μοντέλο που επέστρεψε το optuna study το οποίο έτρεχε για μοντέλα με 3 hidden layers ήταν το εξής:

Number of neurons per hidden layer	[300, 300, 50]
Learning rate	0.00028186057952261717
Batch size	1024
Optimizer	AdamW
Epochs	30 due to early stopping (initial epochs = 100)
Loss function	BCEWithLogitsLoss
Activation function	ReLU
Dropout rate	0.3829821494644316

Το παρόν μοντέλο πέτυχε το καλύτερο accuracy στο validation set την εποχή 25 με τιμή 0.7894376828002642.

Από τα δύο optuna studies κρατάμε το μοντέλο που πέτυχε το καλύτερο accuracy στο validation set (0.7895792055854326. vs 0.7894376828002642). Κρατάμε, δηλαδή, το μοντέλο με το ένα hidden layer που επέστρεψε το 1ο optuna study. Αυτό είναι και το τελικό μας μοντέλο.

3.3 Final model

3.3.1 Preprocessing

Όπως αναφέραμε και στην υποενότητα 3.1. Baseline/Preprocessing, τώρα που έχουμε το τελικό μας μοντέλο, ελέγχουμε αν πράγματι παίρνουμε την καλύτερη απόδοση όταν οι μοναδικές συναρτήσεις που είναι ενεργοποιημένες στο preprocessing είναι η my_lower και my_unpuct.

Επαναλαμβάνουμε, δηλαδή, το πείραμα της υποενότητας 3.1.2, αλλά αυτή τη φορά στο τελικό μοντέλο. Ορίζουμε ο αριθμός των εποχών να είναι 100, αλλά παράλληλα υλοποιούμε και early stopping.

Preprocessing	Accuracy in val_set
disabled	0.7459430134918389 in epoch = 23
enabled all	0.7678790451929427 in epoch = 21
disabled stopword	0.7858288517784696 in epoch = 36
disabled lemmatize	0.76783187093122 in epoch = 26
disabled stopword, lemmatize	0.7895792055854326 in epoch 46
disabled stopword, lemmatize, unpuct	0.7666761015190112 in epoch = 30

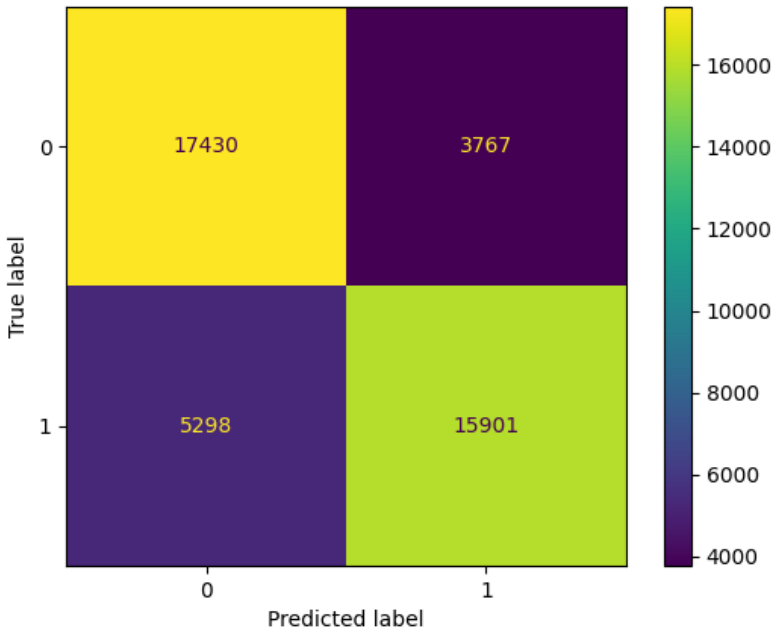
Table 4: Testing preprocessing in final model

Παρατηρούμε πως πράγματι το μοντέλο μας έχει την καλύτερη απόδοση όταν οι συναρτήσεις my_stopword και my_lemmatize είναι απενεργοποιημένες. Καταλήγουμε, δηλαδή, ότι το καλύτερο μοντέλο μας έχει ως εξής:

Hidden layer	1 layer with 400 neurons
Learning rate	0.0006948140681022177
Batch size	1024
Optimizer	AdamW
Epochs	51 due to early stopping (initial epochs = 100)
Loss function	BCELoss
Activation function	ReLU
Dropout rate	0.4529273673532248
GloVe dimension	27B.200d
Preprocessing	my lower, my unpunct only enabled

4. Final model evaluation

4.1 Confusion matrix



Σχήμα 5: Final model: Confusion matrix

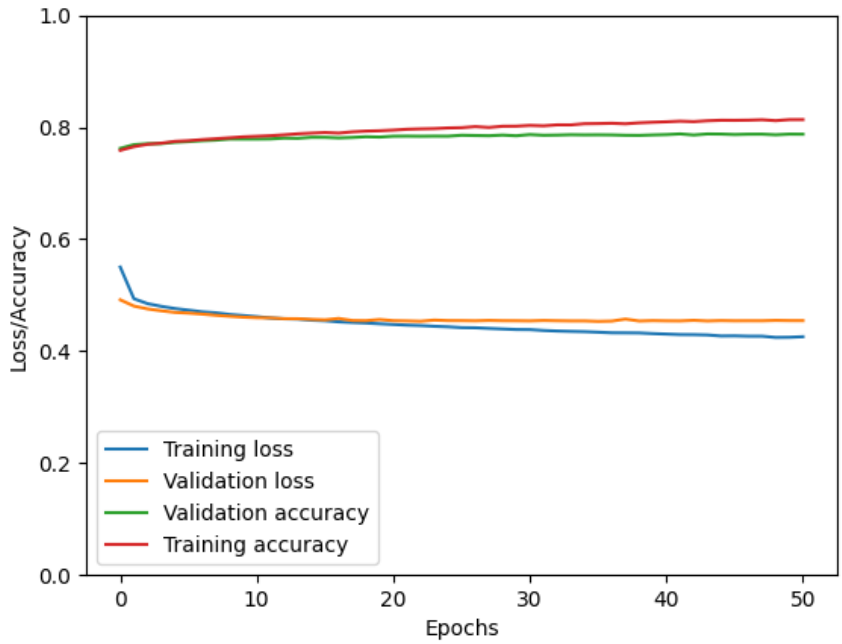
Class	Precision	Recall	F1-score	Support
0	0.77	0.82	0.79	21197
1	0.81	0.75	0.78	21199

Table 5: Final model: Classification report

Το μοντέλο μας πετυχαίνει 79% accuracy. Αποτελεί, δηλαδή, βελτίωση του baseline model κατά 1%. Από το classification report βλέπουμε ότι το label 1 πετυχαίνει ελαφρώς καλύτερο precision (0.81), γεγονός που αποτυπώνει λιγότερο πλήθος false positives, ενώ το label 0 πετυχαίνει καλύτερο recall (0.82), γεγονός που υποδεικνύει μικρότερο πλήθος false

negatives. Επιβεβαιώνουμε αυτά τα αποτελέσματα και στον confusion matrix. Σε αυτόν βλέπουμε ότι το πλήθος των σφαλμάτων του μοντέλου είναι: 3.767 false positives και 5.298 false negatives.

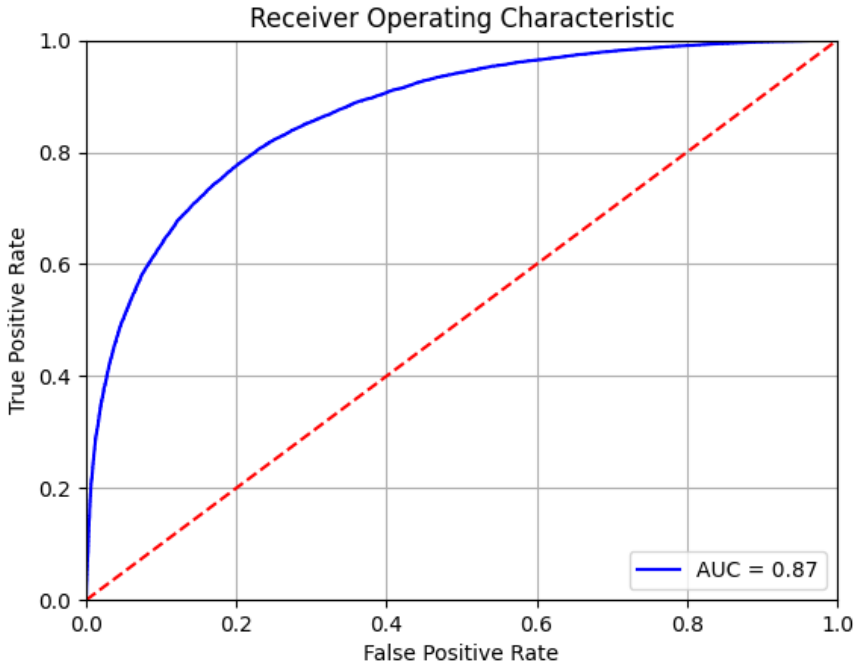
4.2 Learning curve



Σχήμα 6: Final model: Learning curve

Μελετώντας τη καμπύλη μάθησης του μοντέλου βλέπουμε ότι το validation accuracy βελτιώνεται με το πέρασμα των εποχών. Δεν παρατηρείται σημαντική απόκλιση μεταξύ των training και validation loss. Το μοντέλο μας δεν παρατηρείται να κάνει underfit ή overfit. Το performance του είναι πολύ καλό, πιάνοντας accuracy στο validation set ίση με 79%. Τέλος, αξίζει να επισημανθεί, ότι στο test set πετυχαίνει accuracy ίσο με 0.78456.

4.3 ROC curve



Σχήμα 7: Final model: ROC curve

Αναφορικά με τη Reciever Operating Characteristic Curve (ROC) του τελικού μοντέλου, παρατηρούμε ότι η Area Under the Curve (AUC) είναι σημαντικά μεγαλύτερη από 0.5 (τιμή που αποδίδεται σε ένα μοντέλο που τυχαία δίνει τιμές στα predictions του). Η τιμή της AUC πλησιάζει το 0.9, αποτυπώντας την ικανότητα του μοντέλου να διακρίνει με υψηλή αξιοπιστία τις δύο κλάσεις/labels.

5. Results and Overall Analysis

Παρόλα τα πολλά πειράματα που τρέξαμε στην παρούσα εργασία δεν καταφέραμε να ξεπεράσουμε το accuracy της 1ης εργασίας στο validation set το οποίο ήταν ίσο με 80%. Καταφέραμε, όμως, με επιτυχία να φτιάξουμε ένα μοντέλο που δεν κάνει overfit ή underfit. Είδαμε ότι η χρήση των GloVE embeddings 27B.200d έδωσαν μεγάλο boost στο μοντέλο μας, όπως και η διεκπεραίωση preprocessing στα δεδομένα. Η χρήση dropout rate ήταν ένα από τα εξίσου σημαντικά εργαλεία μας, όπως επίσης και η χρήση early stopping κατά τη διαδικασία του training. Το τελικό μας μοντέλο είναι αρκετά καλό.