# ΕΠΕΞΕΡΓΑΣΙΑ ΦΥΣΙΚΗΣ ΓΛΩΣΣΑΣ

# Εργασία 2

## A. N-gram Language Models

**1.** Χρησιμοποιείστε τα 150 πρώτα αρχεία του **WSJ treebank** στο **NLTK**[1] για να εκπαιδεύσετε τα παρακάτω γλωσσικά μοντέλα ν-γραμμάτων (σε επίπεδο λέξεων):

i. Μοντέλο bigrams με add-1 smoothing
ii. Μοντέλο bigrams με Good-Turing discounting[2]
iii. Μοντέλο trigrams με add-1 smoothing
iv. Μοντέλο trigrams με Good-Turing discounting.

**2.** Ο έλεγχος της επίδοσης των παραπάνω μοντέλων πρέπει να γίνει στα υπόλοιπα αρχεία του treebank. Χρησιμοποιείστε το μέτρο του perplexity για να συγκρίνετε την επίδοση των μοντέλων.

Στη φάση της εκπαίδευσης να αντικαταστήσετε όλα τα tokens που εμφανίζονται συνολικά στα κείμενα εκπαίδευσης λιγότερες από 3 φορές με <UNK>. Τα υπόλοιπα tokens θα συμπεριλαμβάνονται στο λεξιλόγιο L. Στη φάση του ελέγχου, όσα tokens δεν συμπεριλαμβάνονται στο L θα πρέπει να αντικατασταθούν με <UNK>.

Θεωρήστε την μορφή των κειμένων του treebank όπου ήδη έχει πραγματοποιηθεί tokenization και sentence splitting. Τα n-grams θα πρέπει να εξάγονται στα πλαίσια της κάθε πρότασης (να μην υπάρχουν n-grams που περιλαμβάνουν το τέλος μιας πρότασης και την αρχή της επόμενης). Προσθέστε δύο ειδικά tokens (<BOS>, <EOS>) που να υποδηλώνουν την αρχή και το τέλος της κάθε πρότασης. Π.χ., η ακόλουθη (tokenized) πρόταση:

[“This”, “is”, “an”, “example”, “.”]

θα μετατραπεί σε:

[“<BOS>”, “This”, “is”, “an”, “example”, “.”, “<EOS>”]

και στην περίπτωση των bigrams θα εξαχθούν τα ακόλουθα:

[“<BOS>”, “This”], [“This”, “is”], [“is”, “an”], [“an”, “example”], [“example”, “.”], [“.”, “<EOS>”]

**3.** Δημιουργήστε 3 νέες προτάσεις με βάση το μοντέλο που βρήκατε ότι έχει την καλύτερη επίδοση. Οι νέες προτάσεις θα πρέπει να ξεκινούν με <BOS> και να τερματίζουν σε <EOS>. Η κάθε λέξη να επιλέγεται τυχαία με βάση την πιθανότητα του κάθε ν-γράμματος (όσο πιο μεγάλη η πιθανότητα του ν-γράμματος τόσο μεγαλύτερη η πιθανότητα να επιλεγεί η λέξη).

[1] **from** nltk.corpus **import** treebank
(υπάρχουν συνολικά 199 αρχεία με ειδήσεις από το Wall Street Journal)
[2] https://www.youtube.com/watch?v=1vUVNdDkIJI

## B. Character-based Language Models

**1.** Χρησιμοποιείστε το σύνολο των 10 πρώτων αρχείων των άρθρων από την εφημερίδα **Το Βήμα** (από την 1η εργασία) για να δημιουργήσετε τα ακόλουθα γλωσσικά μοντέλα (σε επίπεδο χαρακτήρων):

i. Μοντέλο character 3-grams με add-1 smoothing
ii. Μοντέλο character 3-grams με Good-Turing discounting
iii. Μοντέλο character 4-grams με add-1 smoothing
iv. Μοντέλο character 4-grams με Good-Turing

**2.** Ο έλεγχος της επίδοσης των παραπάνω μοντέλων πρέπει να γίνει στα υπόλοιπα 2 αρχεία της συλλογής. Χρησιμοποιείστε το μέτρο του perplexity για να συγκρίνετε την επίδοση των μοντέλων.

Στη φάση της εκπαίδευσης να αντικαταστήσετε όλους τους χαρακτήρες που εμφανίζονται συνολικά στα κείμενα εκπαίδευσης λιγότερες από 5 φορές με το σύμβολο *. Οι υπόλοιποι χαρακτήρες θα συμπεριλαμβάνονται στο λεξιλόγιο L. Στη φάση του ελέγχου, όσοι χαρακτήρες δεν συμπεριλαμβάνονται στο L θα πρέπει να αντικατασταθούν με το σύμβολο *.

Θεωρήστε την μορφή των κειμένων όπου ήδη έχει πραγματοποιηθεί tokenization και sentence splitting (εντός του φακέλου sbd). Τα n-grams θα πρέπει να εξάγονται στα πλαίσια του κάθε token (να μην υπάρχουν character n-grams που περιλαμβάνουν το τέλος ενός token και την αρχή του επόμενου). Προσθέστε δύο ειδικά σύμβολα (@, ~) που να υποδηλώνουν την αρχή και το τέλος του κάθε token. Π.χ., το token:

Παράδειγμα

θα μετατραπεί σε:

@Παράδειγμα~

και στην περίπτωση των 3-grams θα εξαχθούν τα ακόλουθα:

[@,Π,α], [π,α,ρ], [α,ρ,ά], [ρ,ά,δ], [ά,δ,ε], [δ,ε,ι], [ε,ι,γ], [ι,γ,μ], [γ,μ,α], [μ,α,~]

Τα tokens που έχουν μικρότερο μήκος από την τάξη του γλωσσικού μοντέλου να αγνοούνται. Π.χ. όταν n=4 από το token @ο~ δεν μπορούν να εξαχθούν n-grams.

## Γ. Neural Language Models

**1.** Με βάση τα αρχεία με άρθρα από την εφημερίδα **Το Βήμα** δημιουργείστε ένα γλωσσικό μοντέλο σε επίπεδο χαρακτήρων (character-based language model) με χρήση νευρωνικών δικτύων. Τα πρώτα 8 αρχεία να χρησιμοποιηθούν ως training set, τα επόμενα 2 ως validation set και τα επόμενα 2 ως test set. Εξετάστε τα παρακάτω δίκτυα:

i. **Feedforward NN**: Οι χαρακτήρες στο παράθυρο εισόδου προβάλλονται σε embeddings τα οποία στην συνέχεια δίνονται ως είσοδος σε ένα δίκτυο με ένα κρυμμένο επίπεδο. Βασικές παράμετροι του μοντέλου είναι το μέγεθος του συρόμενου παραθύρου εισόδου (w), το μέγεθος της αναπαράστασης των embeddings (d) και το μέγεθος του κρυμμένου επιπέδου (h). Χρησιμοποιήστε w=3, d=10, h=10. (Προαιρετικά: Βρείτε τον καλύτερο συνδυασμό των τιμών παραμέτρων χρησιμοποιώντας το validation set.)

ii. **Recurrent Neural Network**: Ο κάθε χαρακτήρας εισόδου προβάλλεται σε embeddings τα οποία οδηγούνται σε ένα δίκτυο RNN. Η έξοδος του RNN οδηγείται σε ένα fully-connected (dense) δίκτυο που προβλέπει τον επόμενο χαρακτήρα. Βασικές παράμετροι του δικτύου είναι το μέγεθος των embeddings (d) και το μέγεθος του hidden layer (h). Χρησιμοποιήστε d=10 και h=10. (Προαιρετικά: Βρείτε τον καλύτερο συνδυασμό των τιμών παραμέτρων χρησιμοποιώντας το validation set.)

iii. **Long Short-Term Memory**: Ο κάθε χαρακτήρας εισόδου προβάλλεται σε embeddings τα οποία οδηγούνται σε ένα δίκτυο LSTM. Η έξοδος του LSTM οδηγείται σε ένα fully-connected (dense) δίκτυο που προβλέπει τον επόμενο χαρακτήρα. Βασικές παράμετροι του δικτύου είναι το μέγεθος των embeddings (d) και το μέγεθος του hidden layer (h). Χρησιμοποιήστε d=10 και h=10. (Προαιρετικά: Βρείτε τον καλύτερο συνδυασμό των τιμών παραμέτρων χρησιμοποιώντας το validation set.)

Για την εκπαίδευση των παραπάνω μοντέλων χρησιμοποιήστε ως Loss Function το Cross Entropy και επιλέξτε τον Adam optimizer (καθορίζει το ρυθμό με τον οποίο αλλάζουν τα βάρη του δικτύου). Εκπαιδεύστε το δίκτυο για 10 εποχές και επιλέξτε (μεταξύ των εποχών) το δίκτυο που μεγιστοποιεί την επίδοση στο validation set.

**2.** Ο έλεγχος της επίδοσης των παραπάνω μοντέλων πρέπει να γίνει στα υπόλοιπα 2 αρχεία της συλλογής. Χρησιμοποιείστε το μέτρο του perplexity για να συγκρίνετε την επίδοση των μοντέλων.

Θεωρήστε την μορφή των κειμένων όπου ήδη έχει πραγματοποιηθεί tokenization και sentence splitting (εντός του φακέλου sbd). Τα n-grams θα πρέπει να εξάγονται στα πλαίσια του κάθε token (να μην υπάρχουν character n-grams που περιλαμβάνουν το τέλος ενός token και την αρχή του επόμενου). Προσθέστε δύο ειδικά σύμβολα (@, ~) που να υποδηλώνουν την αρχή και το τέλος του κάθε token.

## Οδηγίες

Συνιστάται η χρήση του **PyTorch**[3] ή του **Trax**[4] για την υλοποίηση των νευρωνικών δικτύων.

Για την εκπαίδευση των νευρωνικών δικτύων, εφόσον δεν διαθέτετε πρόσβαση σε GPU, χρησιμοποιείστε το **Google Colab**[5] ενεργοποιώντας την σχετική επιλογή (Edit -> Notebook Settings -> Hardware Accelerator -> GPU).

Συνιστάται η χρήση **Jupyter Notebook**[6] για την υποβολή της εργασίας.

[3] https://pytorch.org/
[4] https://github.com/google/trax/blob/master/trax/examples/Deep_N_Gram_Models.ipynb
[5] https://colab.research.google.com/?utm_source=scs-index
[6] https://jupyter.org/