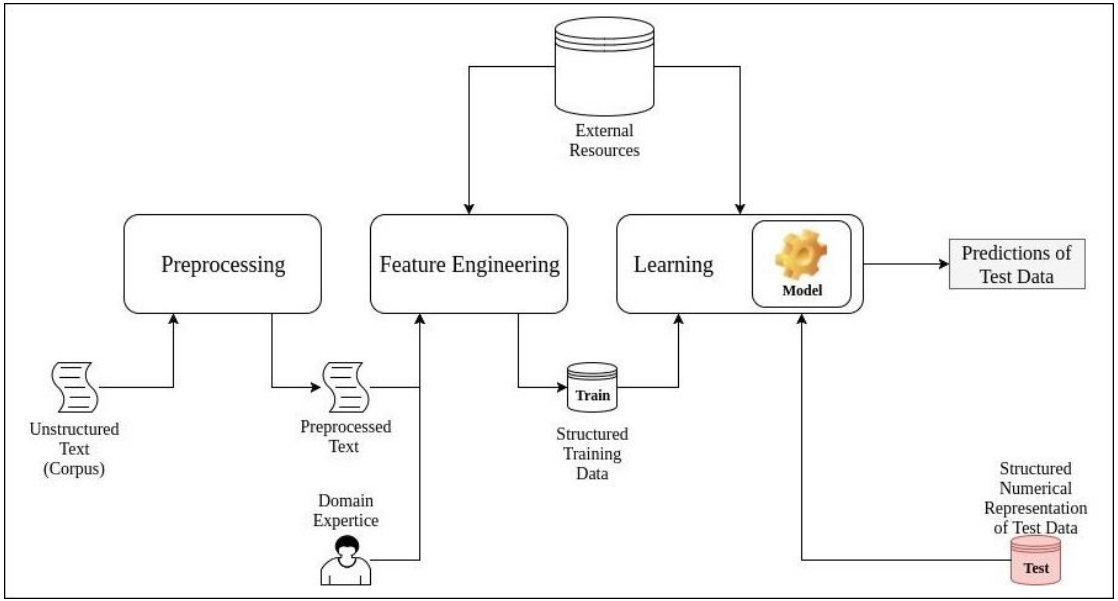
Ενσωματώσεις - Embeddings

Ηρακλής Βαρλάμης

Περιεχόμενα

- Μοντέλα μηχανικής μάθησης
  - Word embeddings
  - Sentence Embeddings
  - LSTM-RNN
  - Transformers

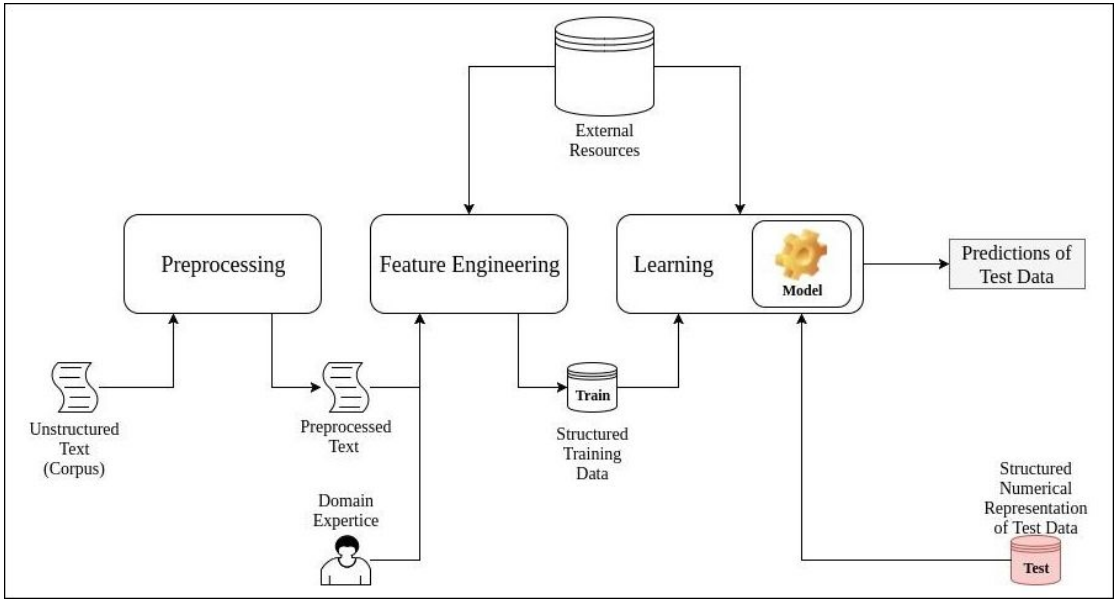
Παραδοσιακό NLP



Preprocessing

- Sentence splitting
- lower/upper case
- Remove stopwords & noise
- Tokenize
- POS tagging
- Lemmatize or stem
- Recognise entities

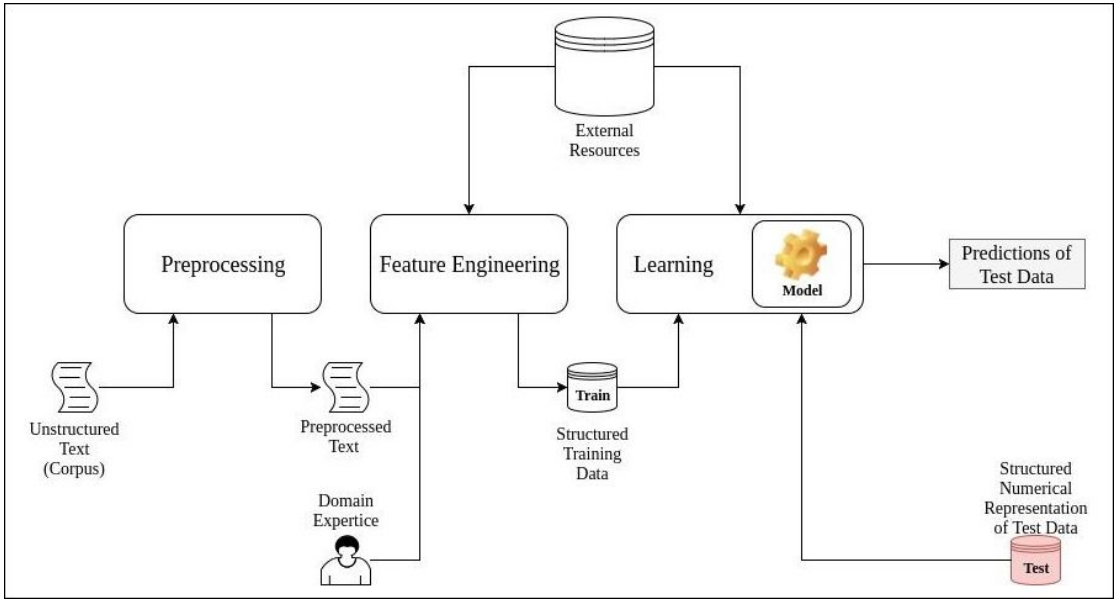
Παραδοσιακό NLP



Feature Engineering

- Αναπαράσταση των κειμένων σε κάποιο μοντέλο (διανυσματικό συνήθως)
- Επιλογή των διαστάσεων που περιέχουν την περισσότερη πληροφορία (π.χ. που με βοηθούν να διαχωρίσω καλύτερα ένα αρνητικό από ένα θετικό σχόλιο)
- Η γνώση πεδίου (λεξικά, θησαυροί, κανόνες κλπ) βοηθούν να επιλέξω διαστάσεις και γνωρίσματα

Παραδοσιακό NLP



Learning

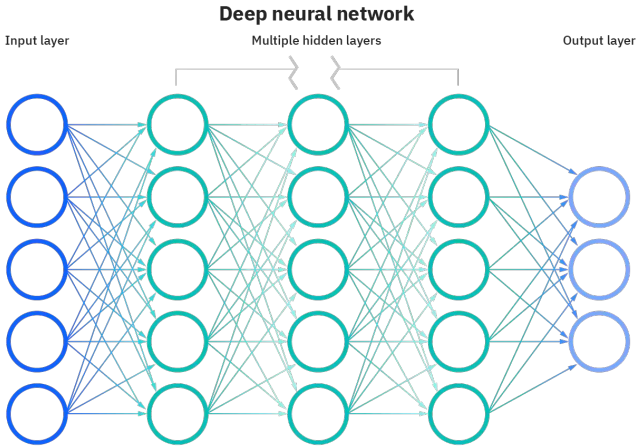
- Δημιουργία ενός επιμεληµένου συνόλου κειμένων όπου άνθρωποι συντάκτες τα έχουν επισημειώσει
- Εκπαίδευση ενός μοντέλου στα επισημειωμένα κείμενα
- Το μοντέλο συνήθως κατηγοριοποιούσε ολόκληρα κείμενα ή τμήματά τους (π.χ. λέξεις) ανάλογα με το task

Νευρωνικά Δίκτυα και η αναπαράσταση με αριθμούς

Ένα βασικό χαρακτηριστικό των Νευρωνικών Δικτύων που τα διαφοροποιεί από άλλες τεχνικές είναι ότι συνδυάζουν γραμμική άλγεβρα και στατιστική.

Αυτό τα περιορίζει στο να δέχονται στην ***είσοδό*** τους ***μόνο αριθμητικά δεδομένα*** και να δίνουν επίσης στην έξοδο αριθμητικά δεδομένα.

Συνεπώς οι λέξεις πρέπει με κάποιο τρόπο να μετατραπούν σε αριθμούς



Αριθμητική (διανυσματική) αναπαράσταση και embeddings

One hot vector

	1	2	3	4	5	6	7	8	9
man	1	0	0	0	0	0	0	0	0
woman	0	1	0	0	0	0	0	0	0
boy	0	0	1	0	0	0	0	0	0
girl	0	0	0	1	0	0	0	0	0
prince	0	0	0	0	1	0	0	0	0
princess	0	0	0	0	0	1	0	0	0
queen	0	0	0	0	0	0	1	0	0
king	0	0	0	0	0	0	0	1	0
monarch	0	0	0	0	0	0	0	0	1

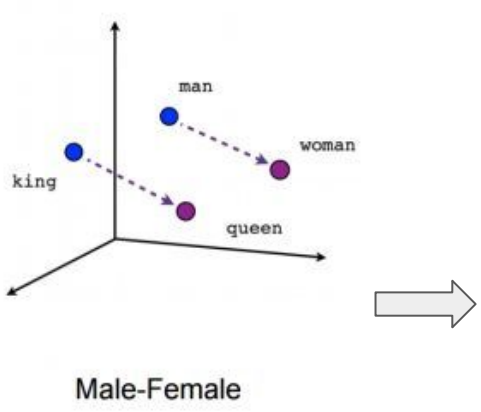
Κάθε λέξη αναπαρίσταται με ένα διάνυσμα (**τεράστιο**) που έχει μήκος όσες οι διαφορετικές λέξεις στο λεξικό μου

Custom embedding (LSA)

	Femininity	Youth	Royalty
Man	0	0	0
Woman	1	0	0
Boy	0	1	0
Girl	1	1	0
Prince	0	1	1
Princess	1	1	1
Queen	1	0	1
King	0	0	1
Monarch	0.5	0.5	1

Μειώνουμε το μήκος του διανύσματος σε κατηγορίες (**έννοιες**) της επιλογής μας

Learnable embedding



Λέξεις με παρόμοιες σημασίες πρέπει να έχουν παρόμοια διανυσματική αναπαράσταση (τουλάχιστον σε κάποιες από τις διαστάσεις)

Embeddings = Πυκνή διανυσματική αναπαράσταση

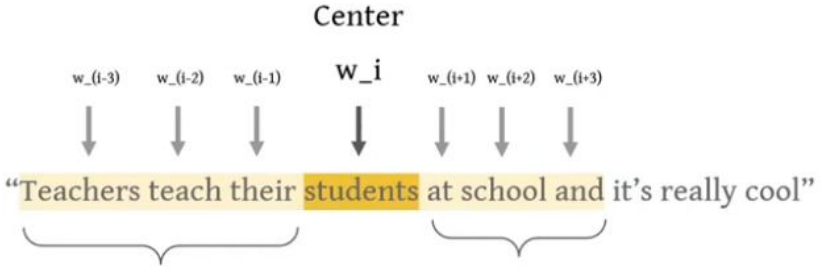
- Εκπαιδεύουμε ένα νευρωνικό πάνω σε πολλά κείμενα με στόχο να μάθει μια πυκνή διανυσματική αναπαράσταση (με λίγα μηδενικά) για κάθε λέξη ή πρόταση
- Βασιζόμαστε στο γεγονός ότι μια λέξη εμφανίζεται συχνά στο ίδιο πλαίσιο περιτριγυρισμένη από άλλες λέξεις
- Word embeddings: Διανύσματα σταθερού μήκους για κάθε λέξη. Word2Vec: Skip-gram, CBOW, GloVe, Fasttext
- Sentence Embeddings: Διανύσματα σταθερού μήκους για κάθε πρόταση σε ένα κείμενο. Doc2Vec

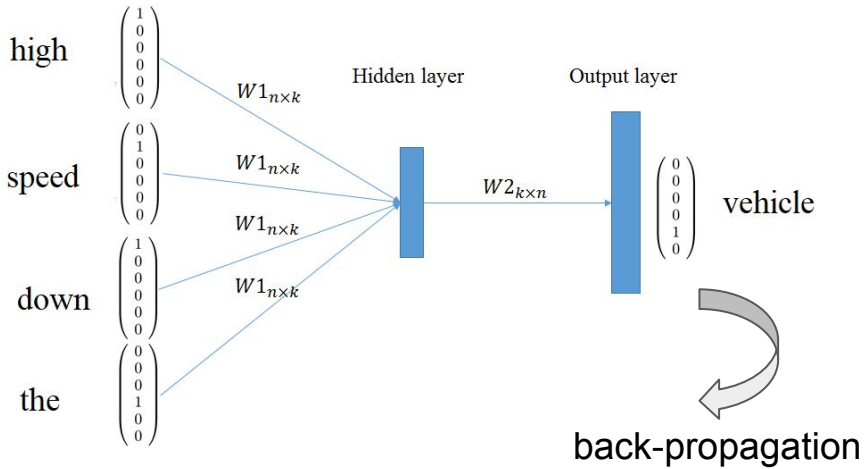
Περιεχόμενα

- Μοντέλα μηχανικής μάθησης
  - Word embeddings
  - Sentence Embeddings
  - LSTM-RNN
  - Transformers

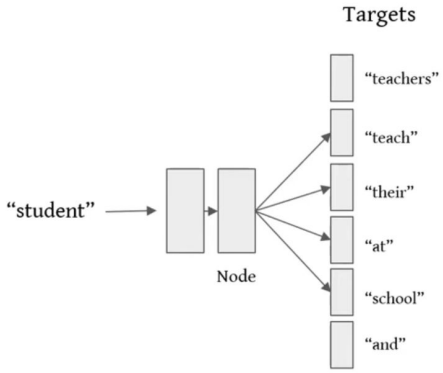
Πως μαθαίνουμε αυτά τα embeddings

CBOW (Continuous Bag of Words) model: Το περιεχόμενο μιας λέξης πρέπει να με βοηθήσει να προβλέψω μια λέξη (κάνοντας χρήση των one-hot vector embeddings)

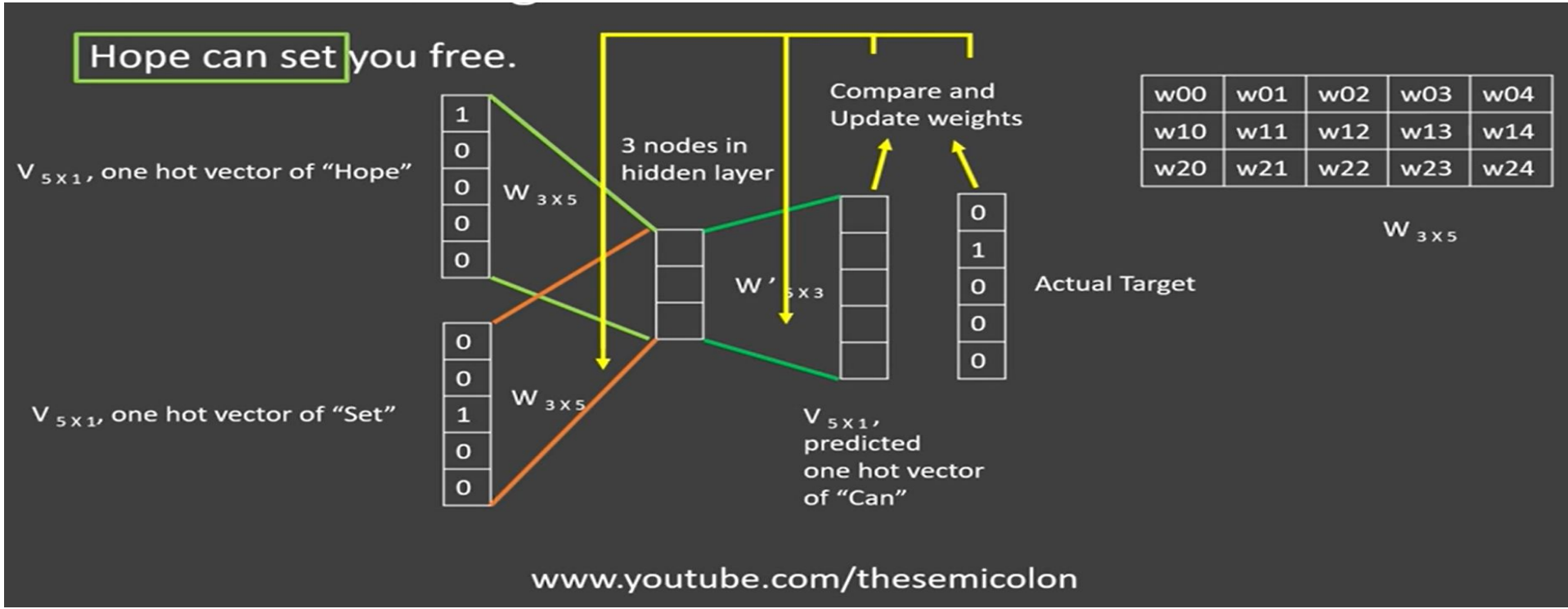




Skip-gram model: Μια λέξη καθορίζει το περιεχόμενό της



CBOW (continuous bag of words)



Skipgram

The probability of a predicted word occurring given a center word:

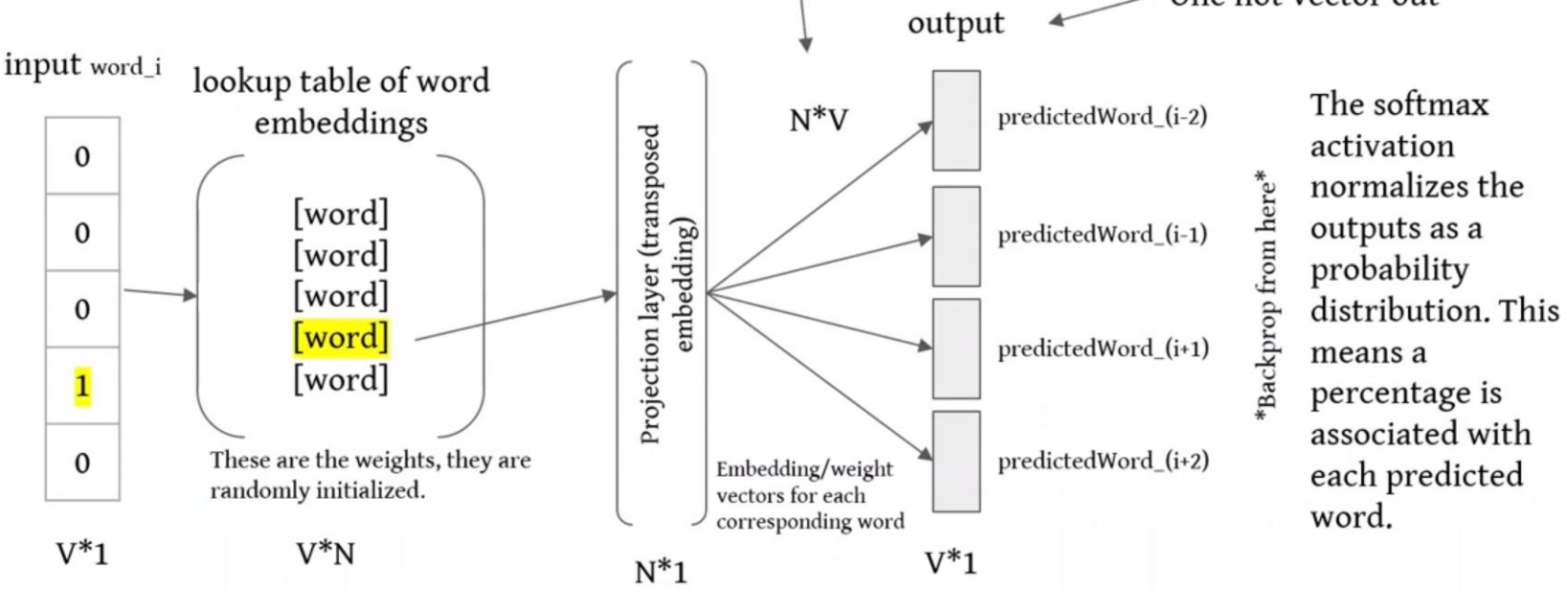
$$P\left(predictedWord_n | centerWord\right) = \frac{e^{dot(predictedWord_n,\ centerWord)}}{\sum_i e^{dot(predictedWord_i,\ centerWord)}}$$

Activation function:

$$softmax\left(predictedWord_n\right) = \frac{e^{predictedWord_n}}{\sum_i e^{predictedWord_i}}$$

One hot vector in:

One hot vector out

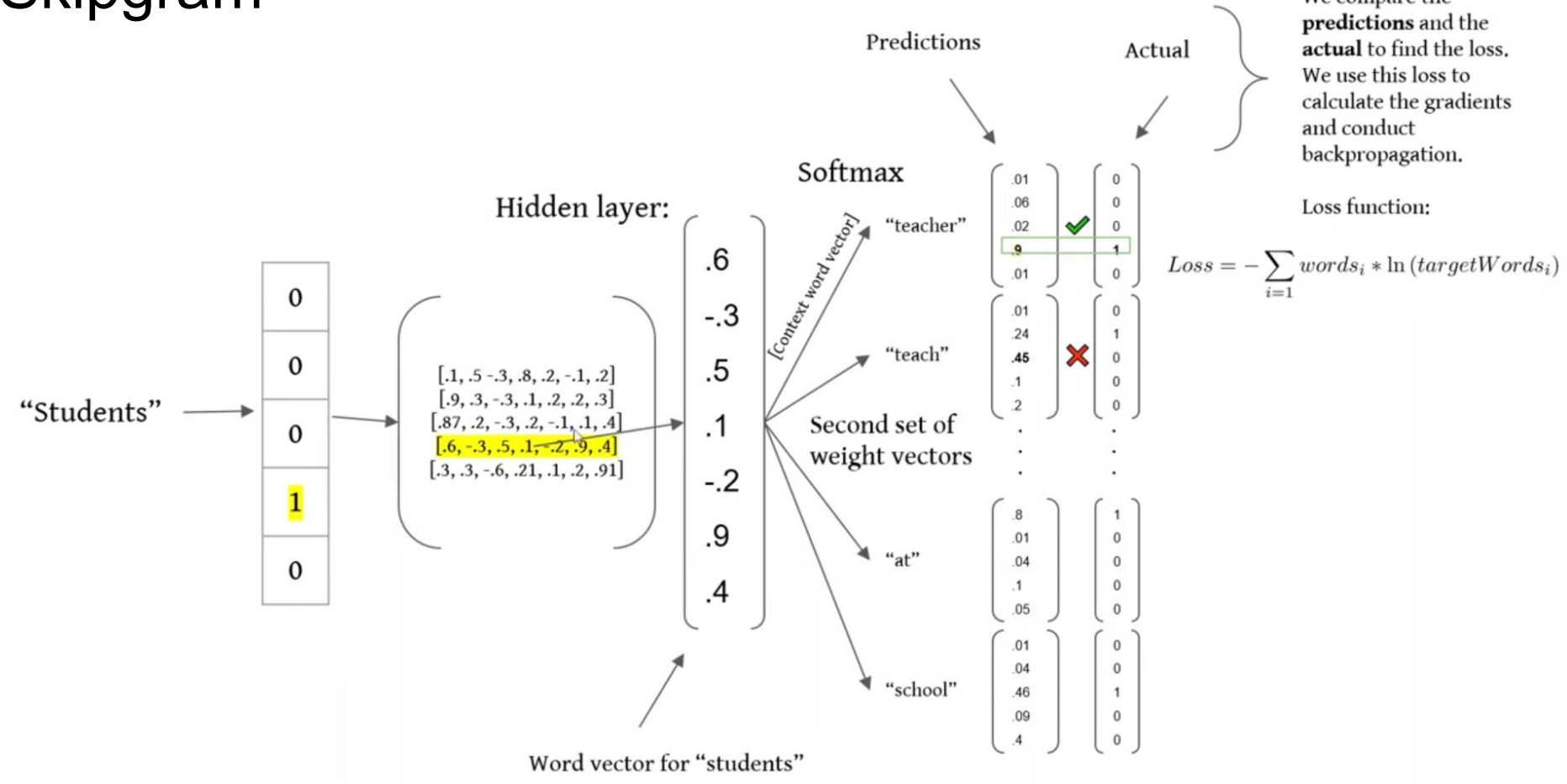


The softmax activation normalizes the outputs as a probability distribution. This means a percentage is associated with each predicted word.

V: # of words in the corpus, N: # of values in our vectors

The weight vector is actually what becomes your word embedding!

Skipgram


Predictions
Actual
We compare the **predictions** and the **actual** to find the loss. We use this loss to calculate the gradients and conduct backpropagation.
Loss function:
$Loss = -\sum_{i=1} words_i * \ln(targetWords_i)$
Softmax
Hidden layer:
"Students"
0
0
0
1
0
[.1, .5 -.3, .8, .2, -.1, .2]
[.9, .3, -.3, .1, .2, .2, .3]
[.87, .2, -.3, .2, -.1, .1, .4]
[.6, -.3, .5, .1, -.2, .9, .4]
[.3, .3, -.6, .21, .1, .2, .91]
.6
-.3
.5
.1
-.2
.9
.4
[context word vector]
"teacher"
"teach"
"at"
"school"
Second set of weight vectors
.01
.06
.02
.9
.01
.01
.24
.45
.1
.2
.8
.01
.04
.1
.05
.01
.04
.46
.09
.4
Word vector for "students"

CBOW vs Skipgram

CBOW

- Προβλέπει την ενδιάμεση λέξη με βάση τις λέξεις που την περιβάλλουν
- Γρήγορο στην εκπαίδευση και Self-supervised
- Μειώνει σημαντικά τις διαστάσεις του χώρου

Skipgram

- Προβλέπει τις λέξεις που περιβάλλουν μια δοσμένη λέξη
- Πιο καλό σε σπάνιες λέξεις

- Δεν διατηρούν καθολική πληροφορία
- Δεν δουλεύουν για άγνωστες λέξεις
- Δεν λαμβάνουν υπόψη το περιεχόμενο

Global Vectors (GloVe)

- Χρησιμοποιεί ως βάση ένα term co-occurrence matrix που έχει προέλθει από μια κεντρική συλλογή κειμένων
- Πάνω σε αυτό υπολογίζει αν μια λέξη είναι πιο πιθανό να εμφανιστεί στο ένα ή στο άλλο context
- Το εσωτερικό γινόμενο των embeddings δύο λέξεων πρέπει να είναι αντίστοιχο του ratio probability

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

Το solid είναι πιο σχετικό με το ice παρά με το steam (ratio>>1)
Το gas είναι πιο σχετικό με το steam παρά με το ice (ratio<<1)
Τα water είναι εξίσου σχετικό με το ice και το steam (ratio~1 και οι επιμέρους πιθανότητες είναι σημαντικές)
Το fashion είναι άσχετο και με τα δύο (ratio~1 και οι επιμέρους πιθανότητες <<0)

Εκπαίδευση

Για να εκπαιδεύσουμε χρησιμοποιούμε ζεύγη λέξεων από τα κείμενα (από το ίδιο context), αλλά και negative samples (ζεύγη λέξεων από διαφορετικά κείμενα).

Προσπαθούμε να ελαχιστοποιήσουμε τις διαφορές μεταξύ vector embedding dot product και probability ratio.

$$\min \sum_e (v_i \cdot v_j - c_{ij})^2$$

$$\min \sum_e (v_i\, v_j + b_i + b_j + \log(c_{ij}))^2$$

Word bias

Context bias

Περιεχόμενα

- Μοντέλα μηχανικής μάθησης
  - Word embeddings
  - Sentence Embeddings
  - LSTM-RNN
  - Transformers

Sentence embedding

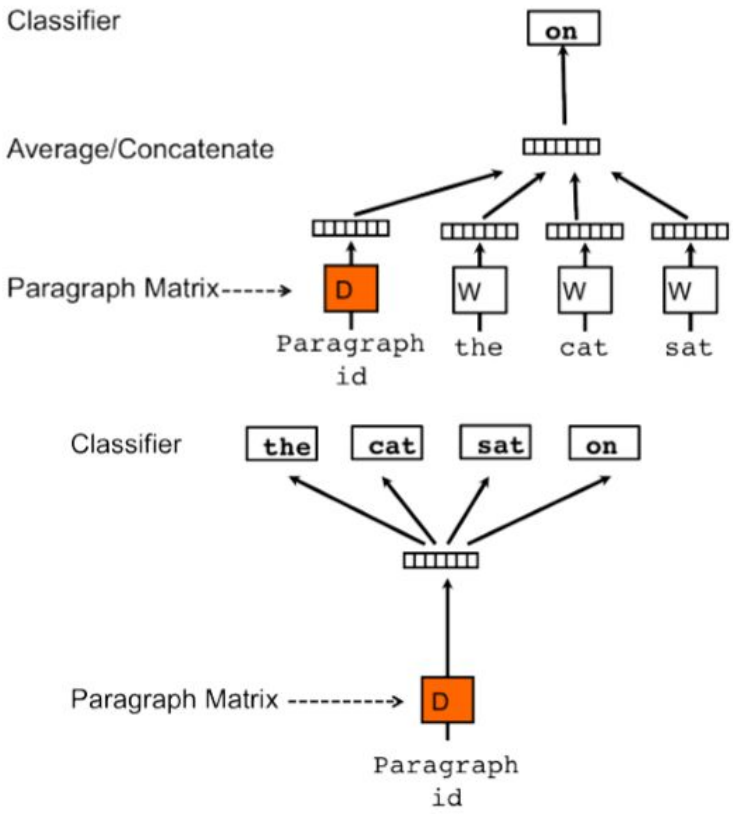
- Βρίσκουμε ένα διάνυσμα για μια πρόταση (Sent2Vec) ή για ένα κείμενο (Doc2Vec)
- Θα μπορούσε απλά να είναι ο μέσος όρος όλων των word embeddings της πρότασης
- Σε κάθε περίπτωση ξεκινάμε με τα προ-εκπαιδευμένα word embeddings (Word2Vec ή GloVe)
- Στη συνέχεια χρησιμοποιούμε τα sentence embeddings για να επιλύσουμε κάποιο task (classification, clustering κλπ).

Εκπαίδευση

- Κάθε πρόταση (παράγραφος) χρησιμοποιείται ως ένας επιπλέον όρος μπροστά από τις υπόλοιπες λέξεις της πρότασης και επαναλαμβάνουμε το next term prediction task (Distributed Memory Model)

  ή

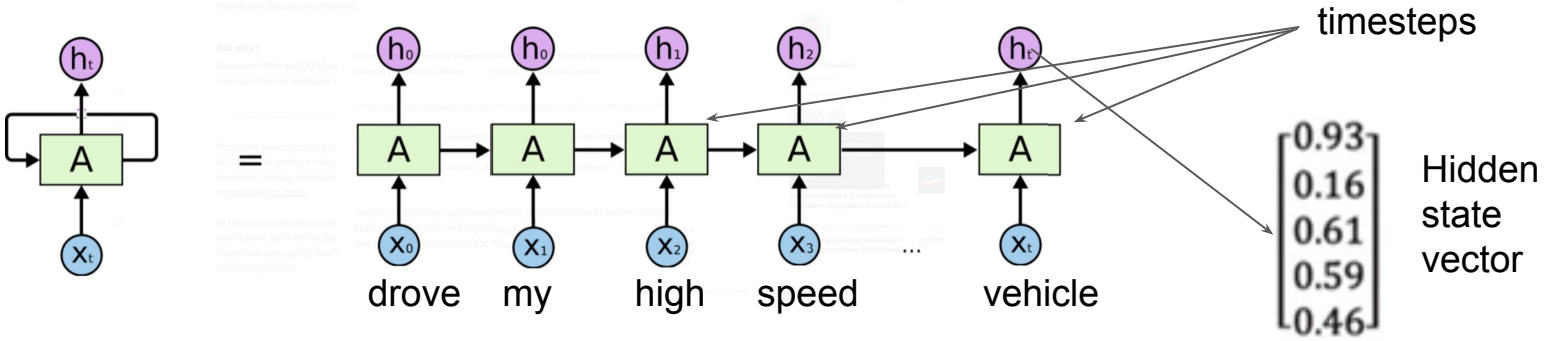
- Κάθε πρόταση χρησιμοποιείται για να προβλέψουμε τις λέξεις της (Distributed BoW)



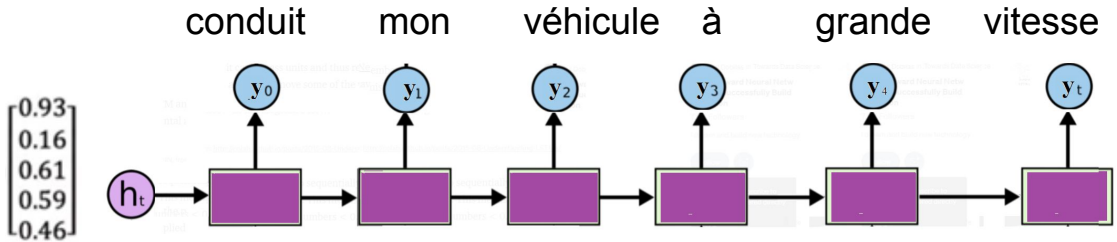
Περιεχόμενα

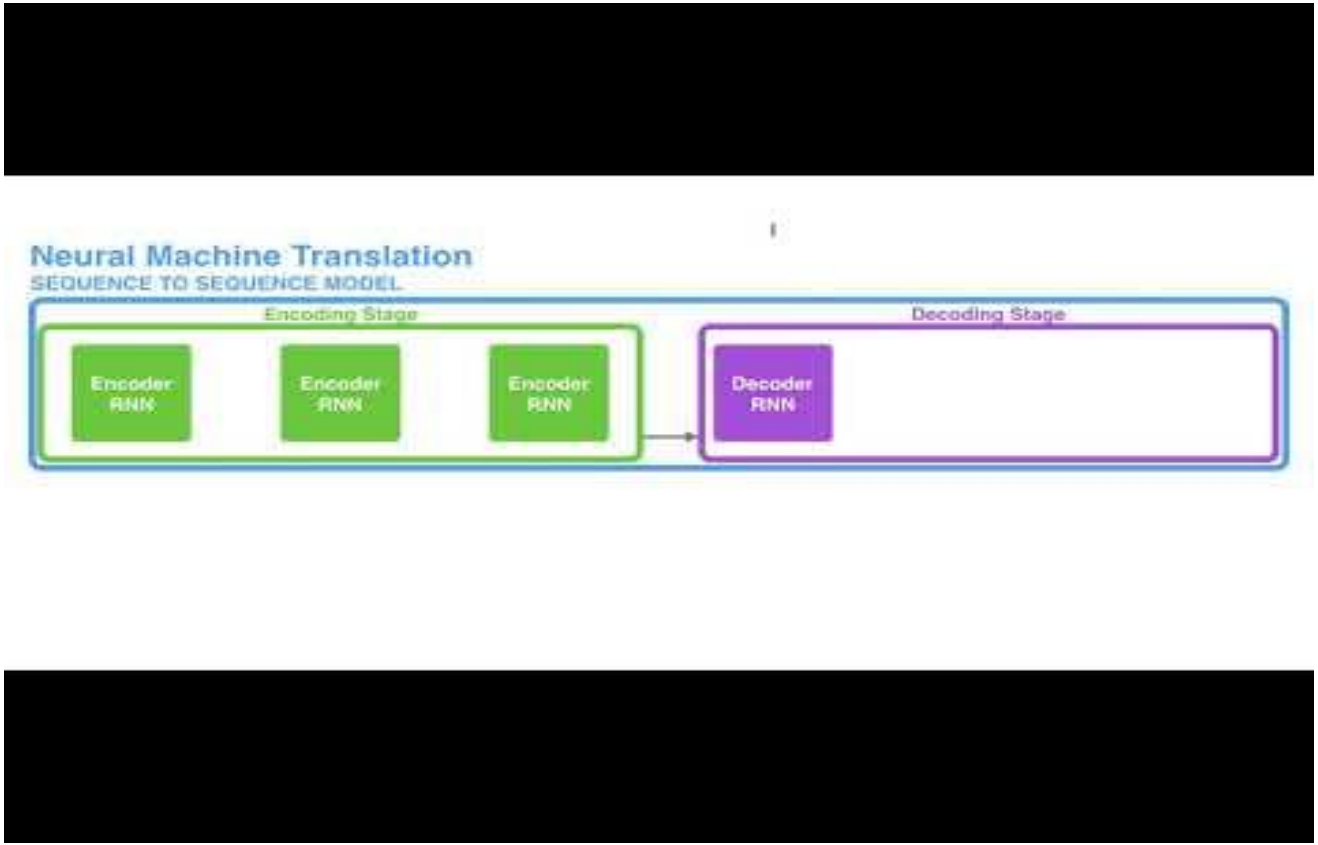
- Μοντέλα μηχανικής μάθησης
  - Word embeddings
  - Sentence Embeddings
  - LSTM-RNN
  - Transformers

Recurrent Neural Networks - Long Short Term Memory



- Οι προηγούμενες (ή οι επόμενες) λέξεις επηρεάζουν την κάθε λέξη
- Κάποιες λέξεις δεν έχουν μεγάλη επιρροή και μπορούμε να τις ξεχνάμε, κι έτσι μπορούμε να μαθαίνουμε από μεγαλύτερες ακολουθίες (LSTM - learn to bypass and forget)
- Η ενδιάμεση αναπαράσταση που προκύπτει μπορεί να χρησιμοποιηθεί για άλλες ακολουθίες (γλώσσες ή modalities), και να αξιοποιηθεί σε μετάφραση

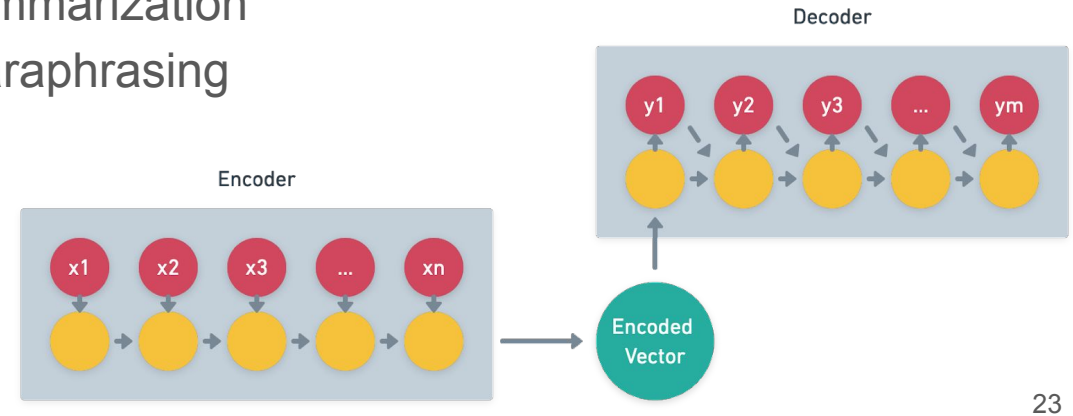




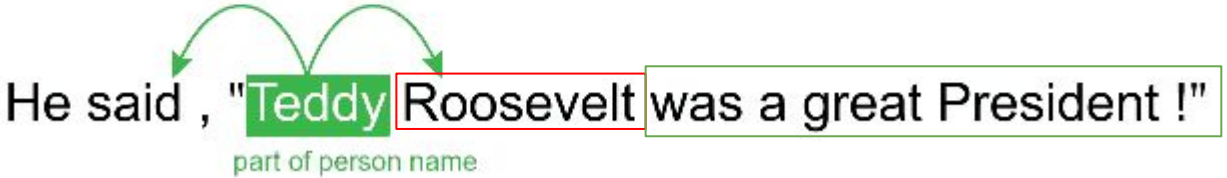
https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/

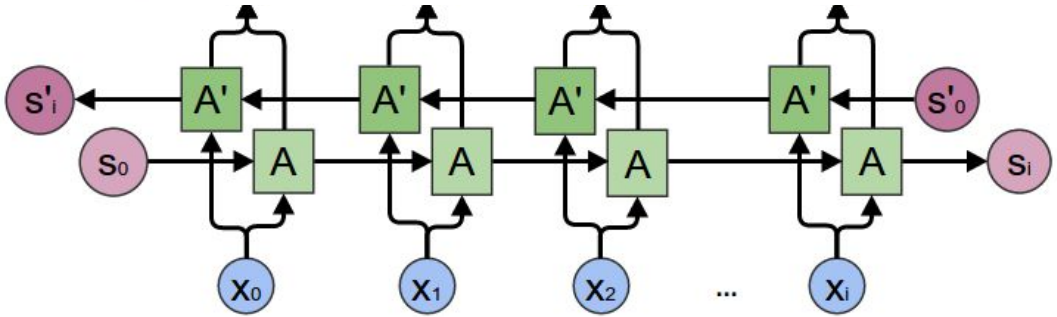
Seq2seq models

- Εκπαιδεύονται με πολλά ζεύγη προτάσεων
- Παράλληλες προτάσεις σε 2 γλώσσες: Μεταφραστής
- Ερωτήσεις και απαντήσεις: Question Answering system, Chatbots
- Ηχητικά/βίντεο και υπότιτλοι: speech recognition, image/video captioning
- Κείμενα και οι περιλήψεις: Summarization
- Κείμενα και παραφράσεις: Paraphrasing



Bi-directional sequence models (BiLSTM)

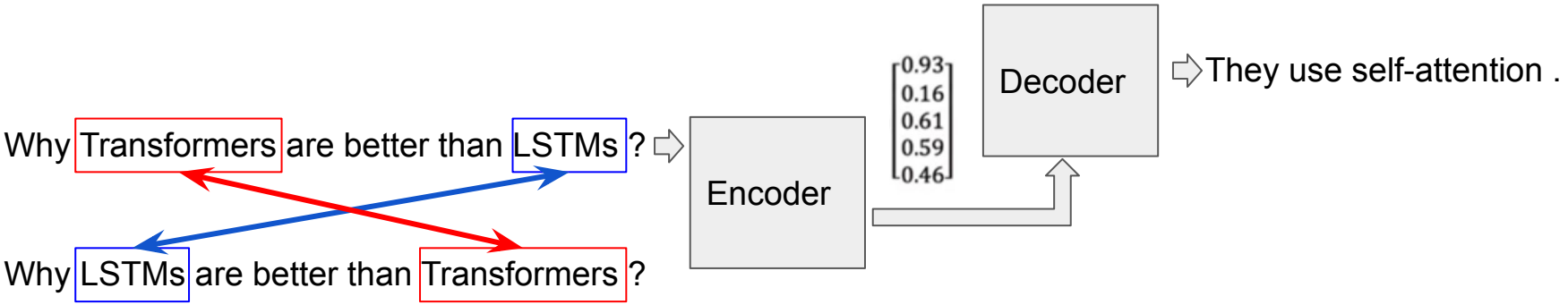




Οι περιορισμοί των RNN, LSTM & GRU

- Δεν μπορούν να χειριστούν (να θυμούνται) μεγάλες ακολουθίες
- Δεν παραλληλοποιούνται: πρέπει να ολοκληρωθεί όλη η είσοδος ώστε να ξεκινήσει η έξοδος

- Ενώ η σειρά με την οποία έρχονται οι λέξεις έχει σημασία, θα θέλαμε να την κωδικοποιούμε χωριστά για κάθε λέξη. Να κωδικοποιούμε τη θέση της λέξης στην πρόταση



Περιεχόμενα

- Μοντέλα μηχανικής μάθησης
  - Word embeddings
  - Sentence Embeddings
  - LSTM-RNN
  - Transformers

Η έννοια της προσοχής - Παράδειγμα

The animal didn't cross the street because **it** was too tired.

The animal didn't cross the street because **it** was too wide.

μετάφραση

L'animal n'a pas traversé la rue parce qu' / \ **il** était trop fatigué.

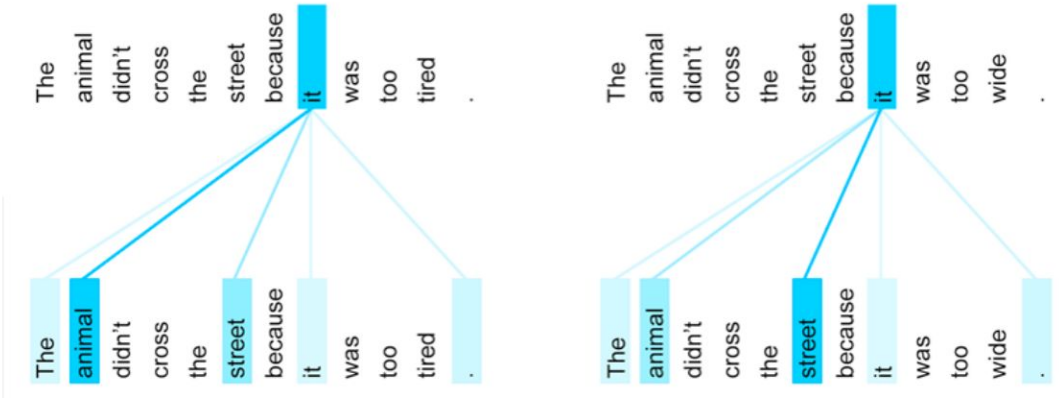
elle était trop large.

Η έννοια της προσοχής - Παράδειγμα

The animal didn't cross the street because **it** was too tired.

The animal didn't cross the street because **it** was too wide.

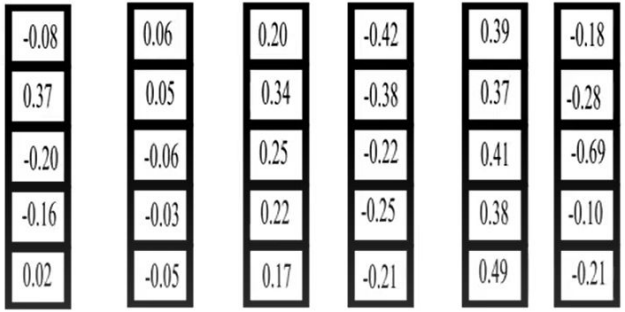
Προσοχή σε διαφορετικές λέξεις



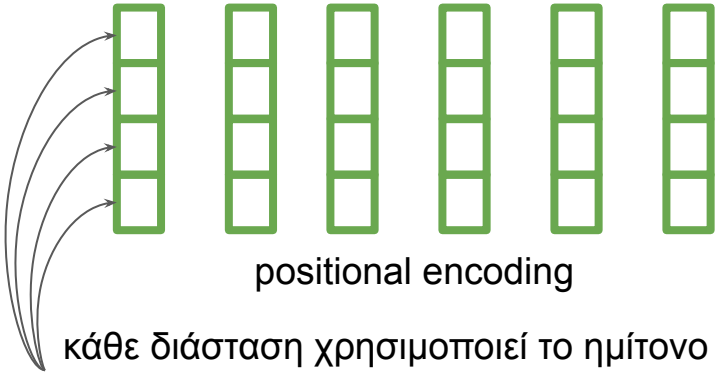
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*. https://arxiv.org/pdf/1706.03762.pdf

Why Transformers are better than LSTMs ?

Why Transformers are better than LSTMs

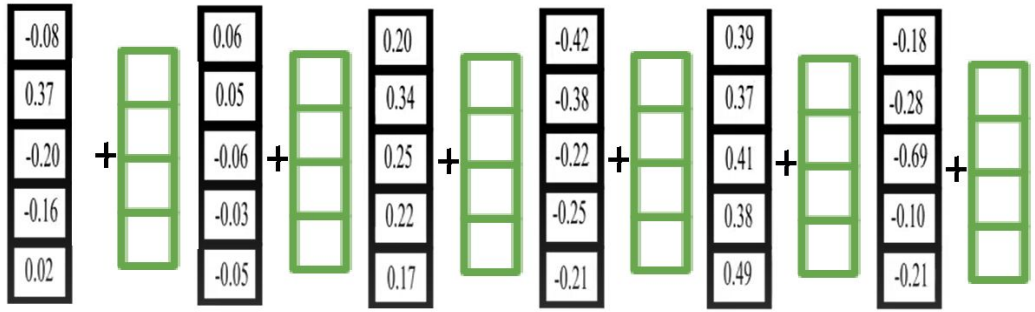


Word embeddings



positional encoding

They combine word embeddings with positional encoding (***meaning+position***)



κάθε διάσταση χρησιμοποιεί το ημίτονο (ή συνημίτονο) της θέσης

$$PE_{(pos,\ 2i)} = \sin(pos\ /\ 1000^{2i/d_{model}})$$