Παραγωγική Τεχνητή Νοημοσύνη: Generative AI

Κωνσταντίνος Καραμανής

The University of Texas at Austin & Archimedes/Athena RC

constantine@utexas.edu

https://caramanis.github.io/

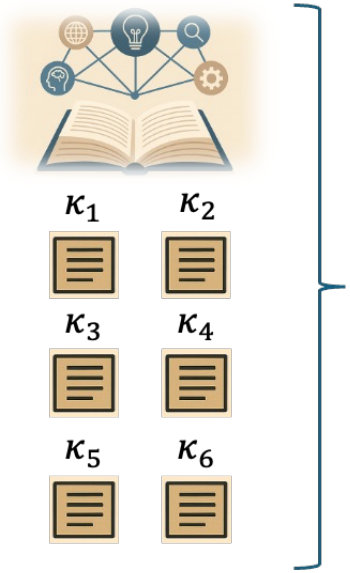
Ας θυμηθούμε τα προηγούμενα...

Αναζήτηση σε Βάση Γνώσεων

Σημασιολογικές ενσωματώσεις

Query: Q → v_Q

κείμενα



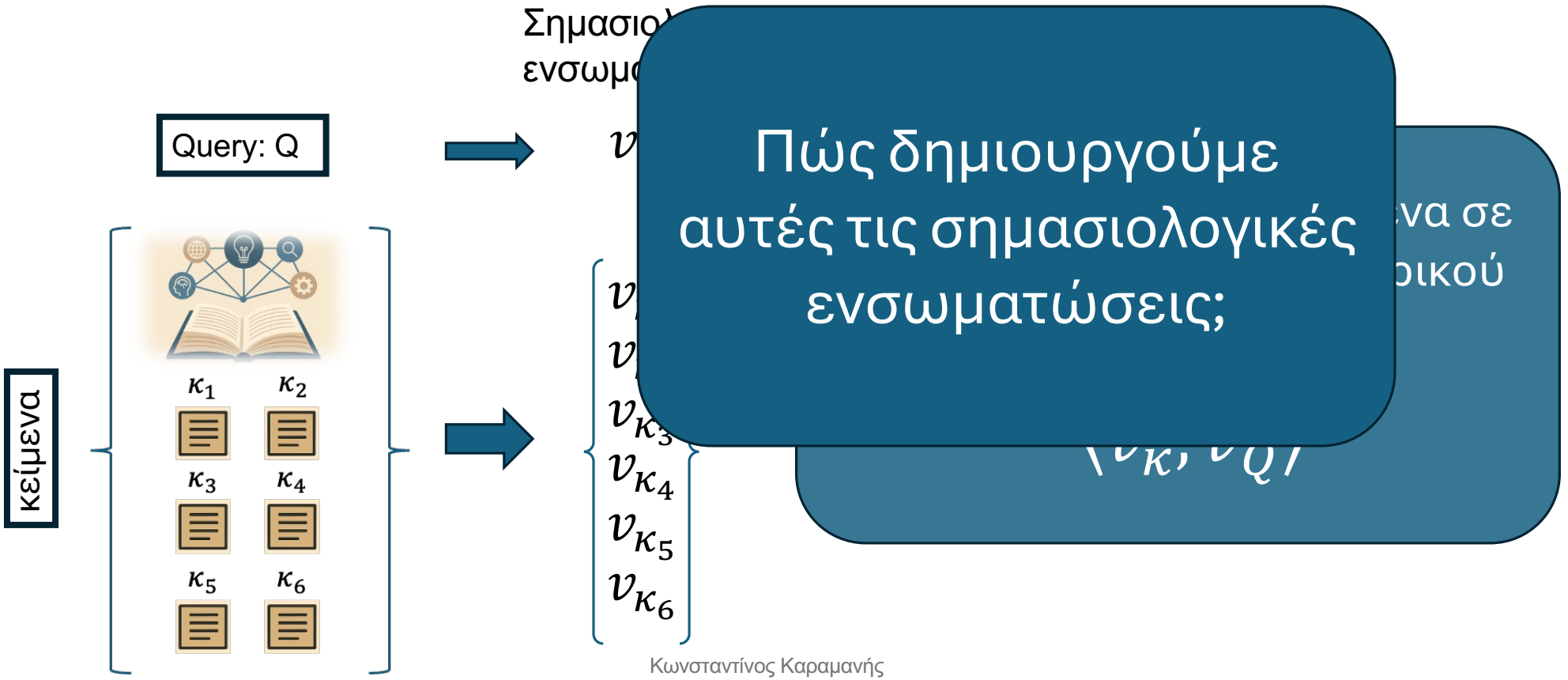
→

v_{κ_1}
v_{κ_2}
v_{κ_3}
v_{κ_4}
v_{κ_5}
v_{κ_6}

Κατατάσσουμε τα κείμενα σε φθίνουσα σειρά εσωτερικού γινομένου

$$\langle v_\kappa, v_Q \rangle$$

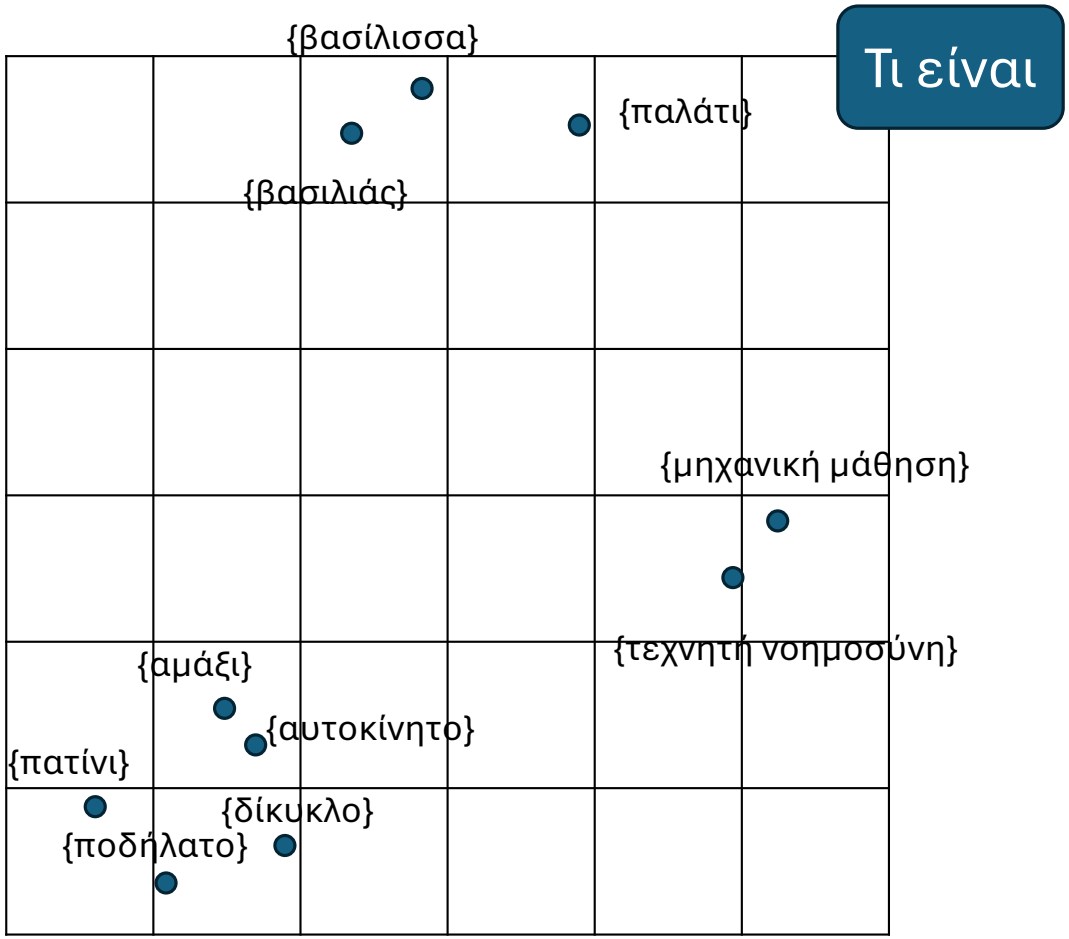
Αναζήτηση σε Βάση Γνώσεων



Word2Vec: μια πρώτη ιδέα

1. Τι είναι
2. Πώς εκπαιδεύεται (self-supervision)
3. Γιατί δεν αρκεί

Word2Vec*: σημασιολογικές ενσωματώσεις λέξεων



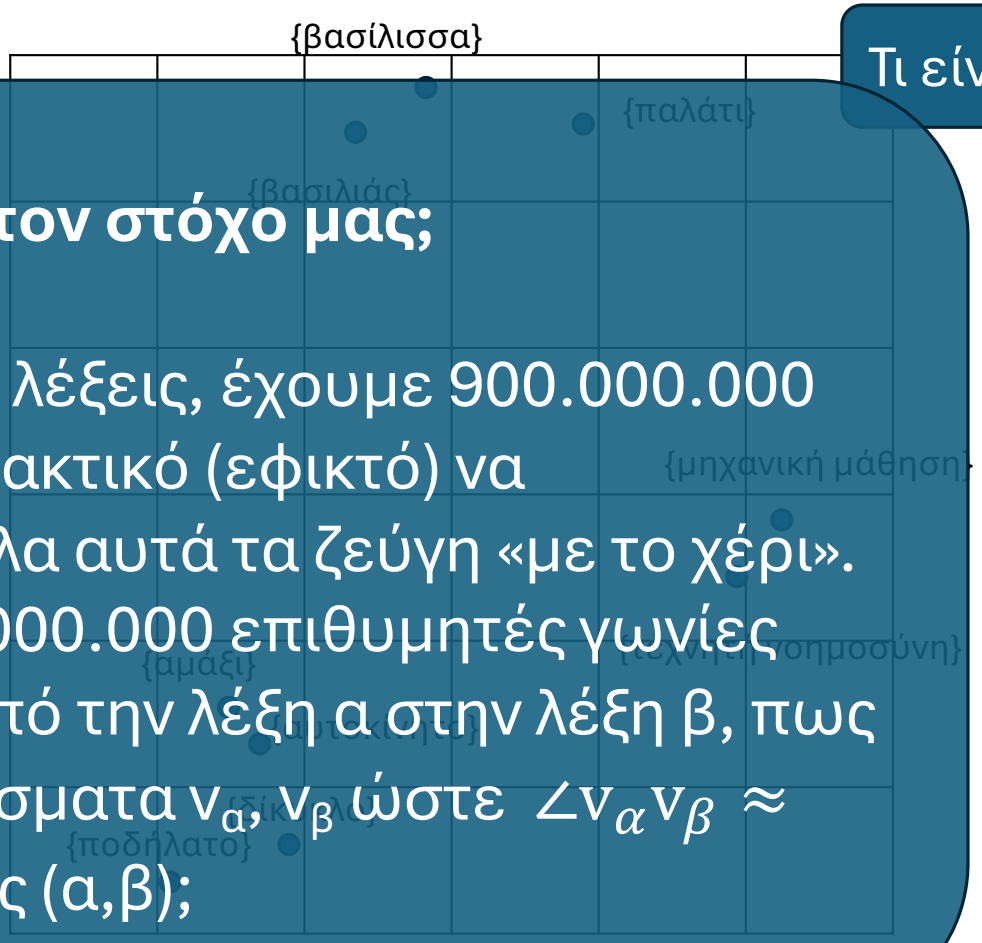
*Mikolov, T., Chen, K., Corrado, G. and Dean, J., 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*

Τι είναι

Πώς θα πετύχουμε τον στόχο μας;

1. Εάν έχουμε 30.000 λέξεις, έχουμε 900.000.000 ζεύγη. Δεν είναι πρακτικό (εφικτό) να χαρακτηρίσουμε όλα αυτά τα ζεύγη «με το χέρι».
2. Και να είχαμε 900.000.000 επιθυμητές γωνίες (αποστάσεις) $\theta_{\alpha\beta}$ από την λέξη α στην λέξη β, πως θα βρίσκαμε διανύσματα v_α, v_β ώστε $\angle v_\alpha v_\beta \approx \theta_{\alpha\beta}$ για κάθε ζεύγος (α,β);

Word2Vec*:
σημασιολογικές
ενσωματώσεις
λέξεων



*Mikolov, T., Chen, K., Corrado, G. and Dean, J., 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*

Η βασική ιδέα: self-supervision

Πώς εκπαιδεύεται

«Αποχαιρετά τη γη, φεύγει.
Περνάει την παγωμένη θάλασσα,
***τρικυμία ξεσπάει, συντρίβεται** το*
***καράβι** σε άγρια χιονισμένα βράχια.»*

Υπολογίζεται από το συνολικό σώμα κειμένων που χρησιμοποιούμε

X_1 = (Αποχαιρετά, γη), y_1 = P(**γη** ακολουθεί **αποχαιρετά**)

X_2 = (*παγωμένη*, *θάλασσα*), y_2 = P(**θάλασσα** ακολουθεί **παγωμένη**)

X_3 = (τρικυμία, ξεσπάει), y_3 = P(**ξεσπάει** ακολουθεί **τρικυμία**)

X_4 = (*συντρίβεται*, *καράβι*), y_4 = P(**καράβι** ακολουθεί **συντρίβεται**)

Η βασική ιδέα: self-supervision

Πώς εκπαιδεύεται

ογίζεται από το
λικό σώμα
ενων που
ιμοποιούμε

Self Supervision

1. Επινοούμε ένα πρόβλημα επιτηρούμενης μάθησης που θα μας εκπαιδεύσει ένα μοντέλο που παράγει ενσωματώσεις

2. Δημιουργούμε δεδομένα για το παραπάνω πρόβλημα με αυτοματοποιημένο (αλγοριθμικό) τρόπο

«Αποχαιρετά
Περνάει την π
τρικυμία ξεσ
καράβι *σε άγ*

X_1 = (Αποχαιρε
X_2 = (*παγωμέν*
X_3 = (τρικυμία,
X_4 = (*συντρίβεται*, *καράβι*), y_4 = P(**καράβι** ακολουθεί **συντρίβεται**)

Δεν αρκεί

Word2Vec: δεν αρκεί για σημασιολογική αναζήτηση

Στατική vs Συμφραζόμενη Ενσωμάτωση

Δεν αρκεί

Word2Vec: δεν αρκεί για σημασιολογική αναζήτηση

Στατική vs Συμφραζόμενη Ενσωμάτωση

“Ο δάσκαλος είπε ότι λείπει ένα κόμμα από μια πρόταση του κειμένου μου.”

“Ο σχολιαστής είπε ότι λείπει ένα κόμμα από την παρουσίαση των αποτελεσμάτων της δημοσκόπησης.”

Δεν αρκεί

Word2Vec: δεν αρκεί για σημασιολογική αναζήτηση

Στατική vs Συμφραζόμενη Ενσωμάτωση

“Ο δάσκαλος είπε ότι λείπει ένα κόμμα από μια πρόταση του κειμένου μου.”

“Ο σχολιαστής είπε ότι λείπει ένα κόμμα από την παρουσίαση των αποτελεσμάτων της δημοσκόπησης.”

Το Word2Vec παράγει στατικές ενσωματώσεις: η ενσωμάτωση της λέξης «κόμμα» στο Word2Vec είναι πάντα το ίδιο διάνυσμα, ανεξάρτητα από τα συμφραζόμενα. Για αποδοτική αναζήτηση με βάση την σημασία, χρειαζόμαστε **συμφραζόμενες** ενσωματώσεις, όπου το διάνυσμα της λέξης εξαρτάται από το πλαίσιο της πρότασης.

Το «Transformer» και το «Self Attention»

“Ο δάσκαλος είπε ότι λείπει ένα κόμμα από μια πρόταση του κειμένου μου.”

Tokenization

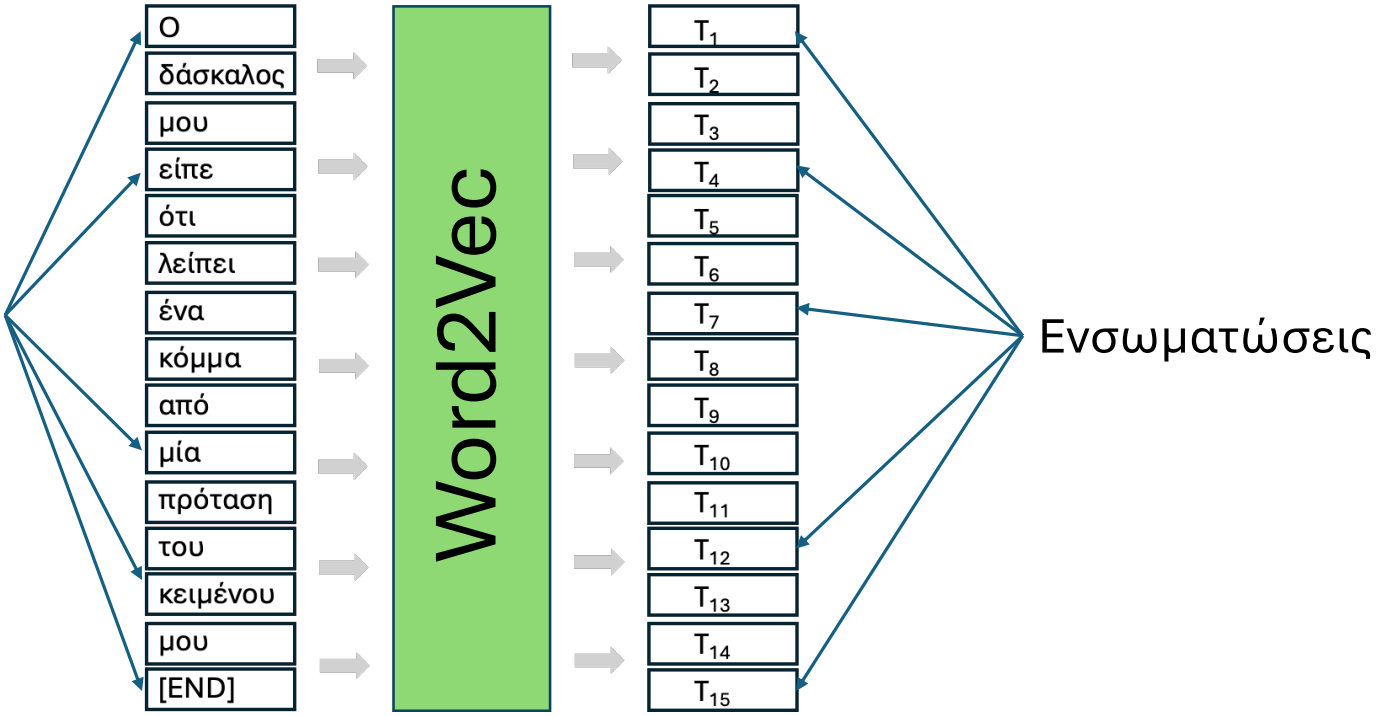
“[Ο] [δάσκαλος] [είπ] [ε] [ότι] [λείπ] [ει] [ένα] [κόμμα] [από] [μια] [πρόταση] [του] [κειμέν] [ου] [μου].”

Tokenization (απλοποίηση)

“[Ο] [δάσκαλος] [είπε] [ότι] [λείπει] [ένα] [κόμμα] [από] [μια] [πρόταση] [του] [κειμένου] [μου].”

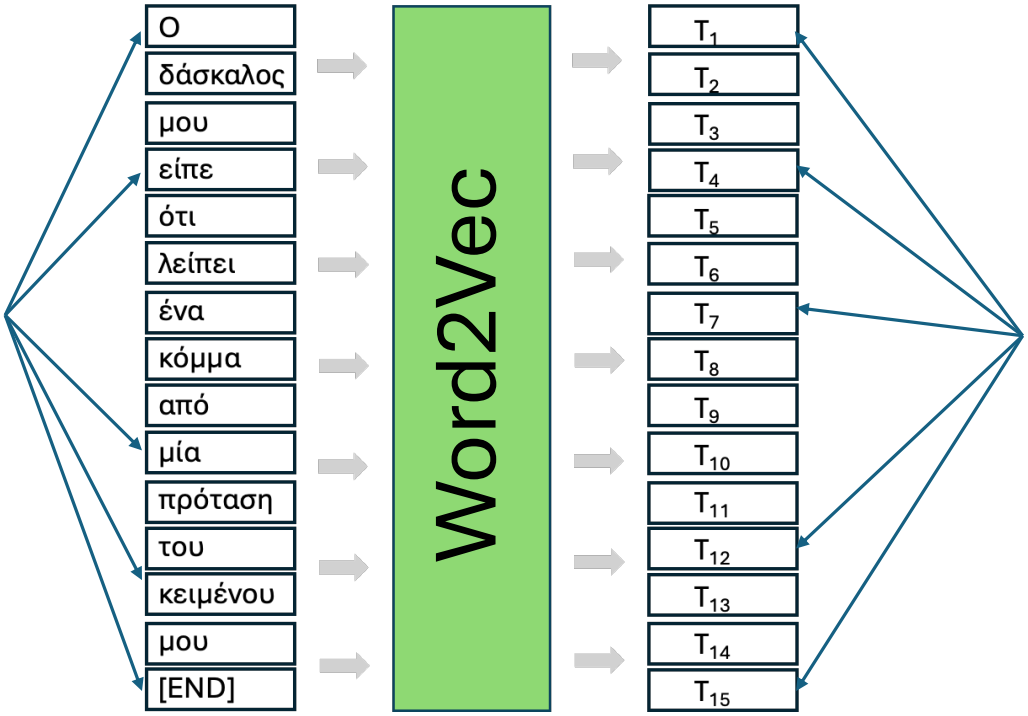
Το «Transformer» και το «Self Attention»

“[Ο] [δάσκαλος] [είπε] [ότι] [λείπει] [ένα] [κόμμα] [από] [μια] [πρόταση] [του] [κειμένου] [μου].”



Το «Transformer» και το «Self Attention»

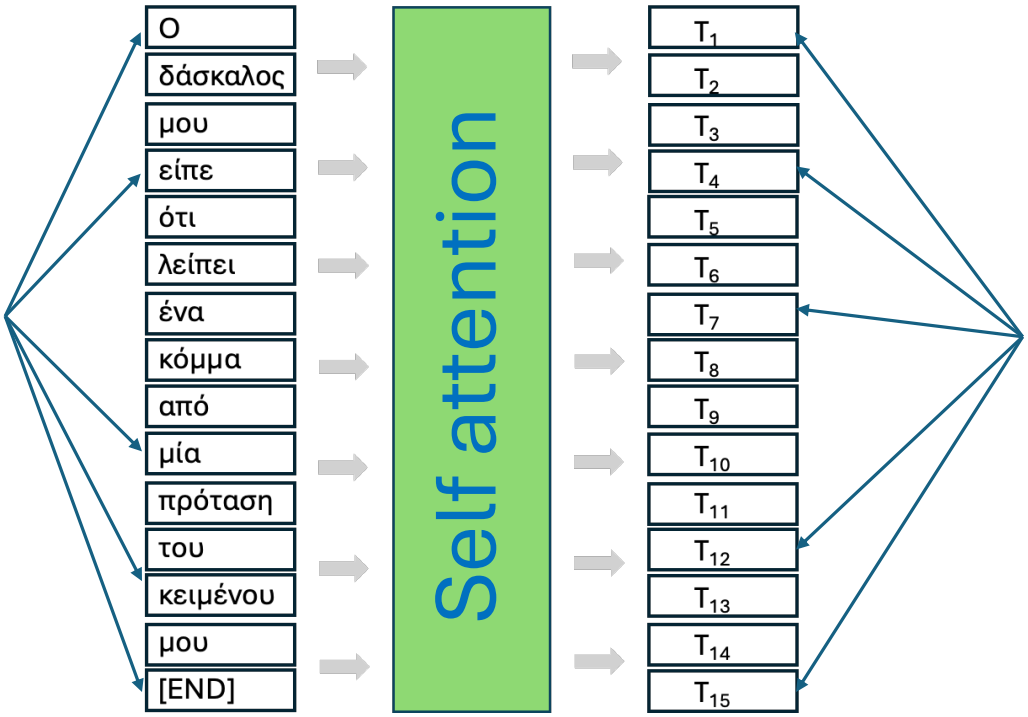
“[Ο] [δάσκαλος] [είπε] [ότι] [λείπει] [ένα] [κόμμα] [από] [μια] [πρόταση] [του] [κειμένου] [μου].”



~~Ενσωματώσεις~~ **Tokens**

Το «Transformer» και το «Self Attention»

“[Ο] [δάσκαλος] [είπε] [ότι] [λείπει] [ένα] [κόμμα] [από] [μια] [πρόταση] [του] [κειμένου] [μου].”



~~Ενσωματώσεις~~
Tokens

Το «Transformer» και το «Self Attention»

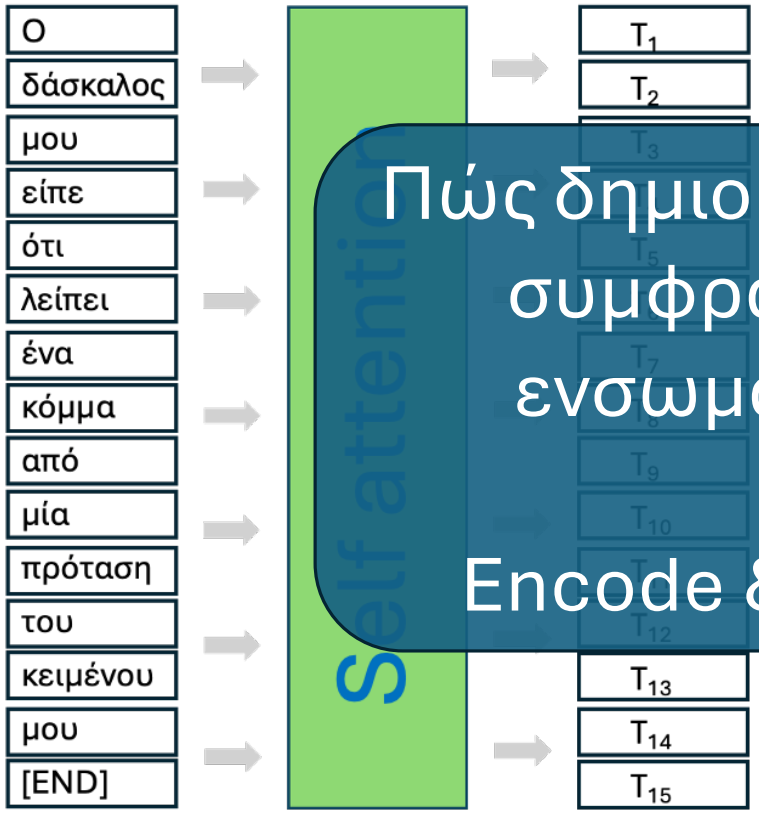
Κωνσταντίνος Καραμανής

Τα «tokens» που περνάνε από τα «attention layers» είναι συμφραζόμενες ενσωματώσεις της κάθε λέξης: εξαρτούνται από την ίδια την λέξη, αλλά και τις άλλες λέξεις της πρότασης (προτάσεων).

Ερώτηση 1: Ποια είναι τα χαρακτηριστικά της αρχιτεκτονικής του «Self-Attention»;

Ερώτηση 2: Αφήνοντας κάτα μέρος το ζήτημα της αρχιτεκτονικής, ρωτάμε: Πως εκπαιδεύονται τα «Transformers» και τα «Self attention layers»; Με ποια δεδομένα και ποια συνάρτηση απώλειας;

Το «Transformer» και το «Self Attention»



Κωνσταντίνος Καραμανής

Πώς δημιουργούνται οι συμφραζόμενες ενσωματώσεις;

Encode & Decoder

Τα «tokens» που περνάνε από τα «attention layers» είναι συμφραζόμενες ενσωματώσεις της κάθε λέξης: εξαρτούνται από την ίδια την λέξη, αλλά και τις άλλες λέξεις της πρότασης (προτάσεων).

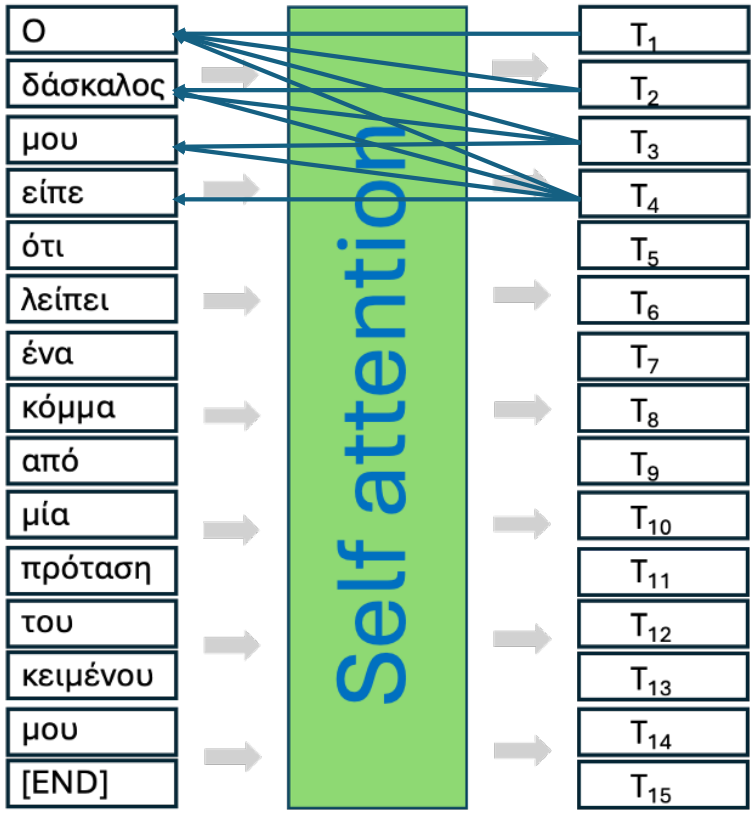
Ερώτηση 1: Ποια είναι τα χαρακτηριστικά της αρχιτεκτονικής του «Self Attention»;

Ερώτηση 2: Αφήνοντας κάτα μέρος το ζήτημα της αρχιτεκτονικής, ρωτάμε: Πως εκπαιδεύονται τα «Transformers» και τα «Self attention layers»; Με ποια δεδομένα και ποια συνάρτηση απώλειας;

Encoders vs Decoders

Decoder: τα Tokens έχουν αιτιώδη (*causal*) εξάρτηση στις λέξεις της πρότασης.

Encoders vs Decoders



Decoder: τα Tokens έχουν αιτιώδη (*causal*) εξάρτηση στις λέξεις της πρότασης.

T_1 υπολογίζεται μόνο από την πρώτη λέξη

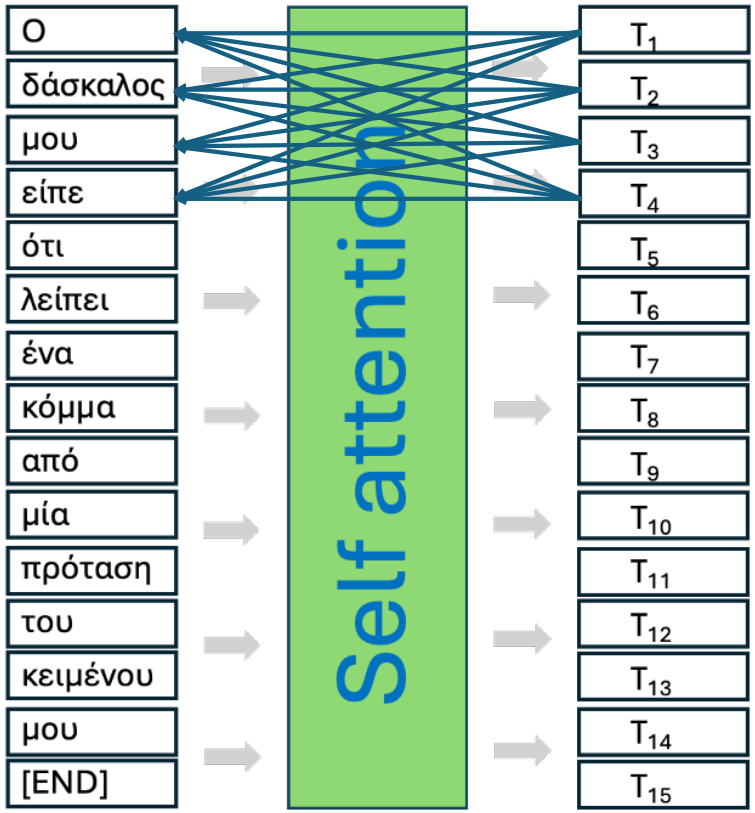
T_2 υπολογίζεται μόνο από τις πρώτες δύο λέξεις

T_3 υπολογίζεται μόνο από τις πρώτες τρεις λέξεις

T_4 υπολογίζεται μόνο από τις πρώτες τέσσερις λέξεις

GPT family (υποθέτουμε)

Encoders vs Decoders



Encoder: τα Tokens έχουν μη-αιτιώδη (*non-causal*) εξάρτηση στις λέξεις της πρότασης.

T_1 υπολογίζεται από όλες τις λέξεις

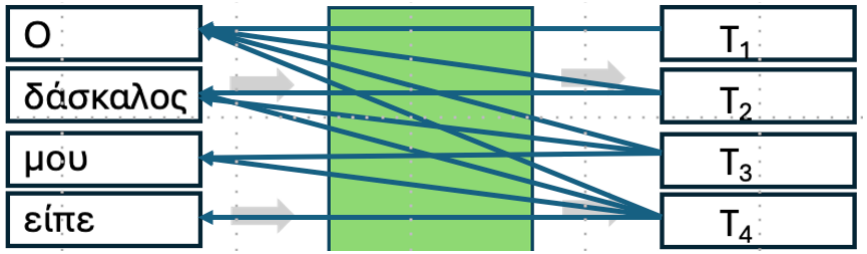
T_2 υπολογίζεται από όλες τις λέξεις

T_3 υπολογίζεται από όλες τις λέξεις

T_4 υπολογίζεται από όλες τις λέξεις

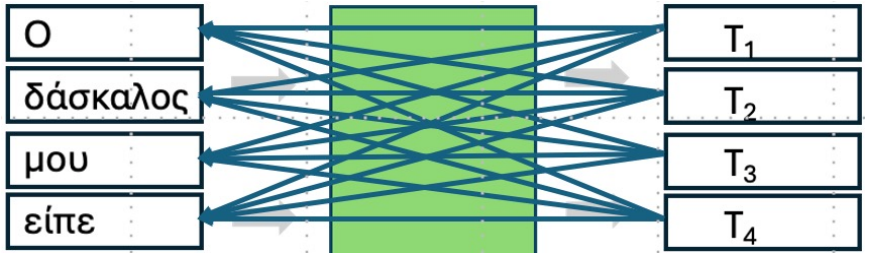
BERT family και πολλά άλλα

Encoders vs Decoders: Η Διαφορά



Decoder: τα Tokens έχουν αιτιώδη (*causal*) εξάρτηση στις λέξεις της πρότασης.

Εκπαιδεύεται για πρόβλεψη της επόμενης λέξης

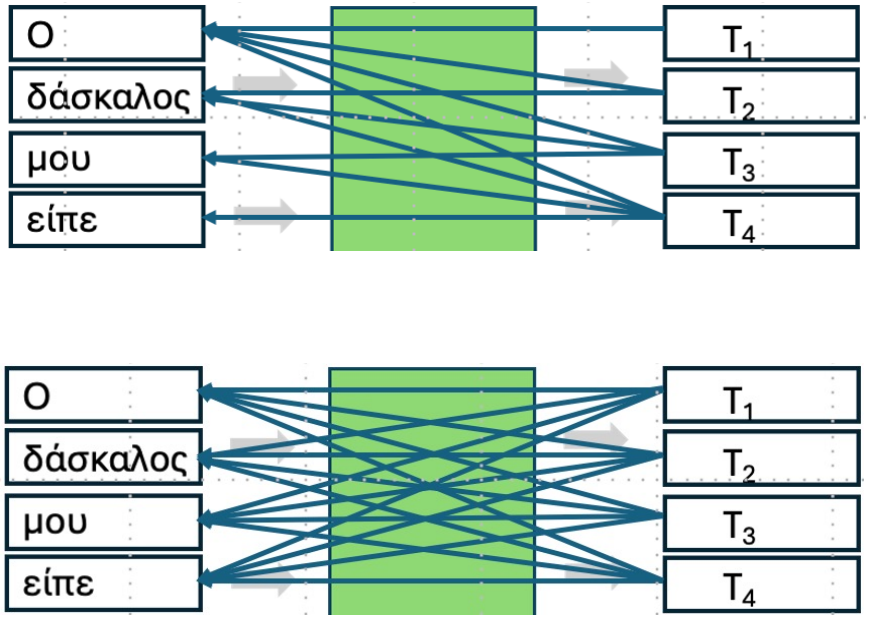


Encoder: τα Tokens έχουν μη αιτιώδη (*non-causal*) εξάρτηση στις λέξεις της πρότασης.

Απαραίτητα, π.χ., για μετάφραση

Οι σημασιολογικές ενσωματώσεις χρησιμοποιούν Encoders.

Encoders vs Decoders: Η Διαφορά



Decoder: τα Tokens έχουν αιτιώδη (*causal*) εξάρτηση στις λέξεις της πρόταση

Εκπαιδε
επόμενη

Encode
(*non-ca*
πρόταση

Απαραίτητα, π.χ., για μετάφραση

Τα encoders και τα decoders παράγουν συμφραζόμενες ενσωματώσεις

Οι σημασιολογικές ενσωματώσεις χρησιμοποιούν Encoders.

Επιτηρούμενη Μάθηση

Επιτηρούμενη Μάθηση:
Βλέπουμε *n* παραδείγματα:
(X_1,y_1), (X_2,y_2), …., (X_n,y_n)

Μαθαίνουμε την σχέση X → y, δηλαδή μαθαίνουμε συνάρτηση h(X) που να είναι «κοντά» στο y.

Επιτηρούμενη Μάθηση

Επιτηρούμενη Μάθηση:
Βλέπουμε *n* παραδείγματα:
(X_1,y_1), (X_2,y_2), ..., (X_n,y_n)

Το αρχικό μας πρόβλημα είναι να βρούμε σημασιολογικές ενσωματώσεις. Για να χρησιμοποιήσουμε όλα τα εργαλεία της επιτηρούμενης μάθησης, πρέπει να διαμορφώσουμε και να δημιουργήσουμε ένα πρόβλημα με τη σωστή μορφή.

Χρειαζόμαστε Ν «παραδείγματα» (X,y)

Επιτηρ
Μάθησ

Επιτηρούμενη Μάθηση:
ματα:
)

Το
ενα
της
να

γικές
αλεία
η
με και
ρφή.

Self Supervision

1. Επινοούμε ένα πρόβλημα επιτηρούμενης μάθησης που θα μας εκπαιδεύσει ένα μοντέλο που παράγει ενσωματώσεις

2. Δημιουργούμε δεδομένα για το παραπάνω πρόβλημα με αυτοματοποιημένο (αλγοριθμικό) τρόπο

Χρειαζόμαστε Ν «παραδειγματα» (X,y)

Self-supervision + σώμα κειμένων (text corpus)

Constantine Caramanis
Professor, Dept. of Electrical and Computer Engineering
Chandra Family Endowed Distinguished Professorship in Electrical and Computer Engineering
Member of the Computer Science Graduate Studies Committee
Office: 2501 Speedway, EER Building Room 6.820
e-mail: constantine at utexas.edu

I am a Professor in the ECE department of The University of Texas at Austin. I received a PhD in EECS from The Massachusetts Institute of Technology, in the Laboratory for Information and Decision Systems (LIDS), and an AB in Mathematics from Harvard University. I received the NSF CAREER award in 2011, and I am an IEEE Fellow.

My current research interests focus on autonomous decision-making in large-scale complex systems, with a focus on learning and computation. Specifically, I am interested in robust and adaptable optimization, high dimensional statistics and machine learning, reinforcement learning and agents, and applications to generative models. I've worked on applications to large-scale networks, including social networks, wireless networks, transportation networks, energy networks and financial applications. I have also worked on applications of machine learning and optimization to computer-aided design.

Κωνσταντίνος Καραμανής

Το διαδίκτυο και άλλες πηγές μας προσφέρουν ανεξάντλητο πλήθος κειμένων. Πρέπει να διαμορφώσουμε πρόβλημα επιτηρούμενης μάθησης.

Αυτό απαιτεί: ζεύγη (X_i, y_i) και συνάρτηση απώλειας που ορίζουν ένα πρόβλημα στο πρότυπο της επιτηρούμενης μάθησης (supervised learning).

Self-supervision + σώμα κειμένων

Masked Language Modeling:

1. 15%: επιλέγουμε τυχαία 15% των «tokens»
2. 80%: αντικαθιστούμε 80% με [MASK]
3. 10%: αντικαθιστούμε 10% με τυχαία επιλεγμένη λέξη
4. 10%: τα υπόλοιπα 10% των tokens παραμένουν ίδια

X = κείμενο, Y = σωστά (αρχικά) tokens

Next Sentence Prediction:

1. Επιλέγουμε δύο προτάσεις (S1,S2) από το σώμα κειμένων. Στις μισές περιπτώσεις, η πρόταση S2 ακολουθεί την S1 στο κείμενο

X = (S1, S2), Y = 1 or 0 (ακόλουθη πρόταση)

Self-supervision + σώμα κειμένων

1. 15% των tokens επιλέγονται

TASK: Masked Language Modeling (MLM)

X_1: My current research interests focus on autonomous decision-making in large-scale complex systems, with a focus on learning and computation

Y_1: {research, autonomous, learning, computation}

X_2: I am interested in robust and adaptable optimization, high dimensional statistics and machine learning, reinforcement learning and agents.

Y_2: {robust, statistics, machine, agents}

Self-supervision + σώμα κειμένων

1. 15% των tokens επιλέγονται

2. 80% των επιλεγμένων token αντικαθίστανται με [MASK]

TASK: Masked Language Modeling (MLM)

X_1: My current [MASK] interests focus on [MASK] decision-making in large-scale complex systems, with a focus on [MASK] and computation

Y_1: {research, autonomous, learning, computation}

X_2: I am interested in [MASK] and adaptable optimization, high dimensional [MASK] and machine learning, reinforcement learning and [MASK]

Y_2: {robust, statistics, machine, agents}

Self-supervision + σώμα κειμένων

1. 15% των tokens επιλέγονται
2. 80% των επιλεγμένων token αντικαθίστανται με [MASK]
3. 10% των επιλεγμένων token αντικαθίστανται τυχαία

TASK: Masked Language Modeling (MLM)

X_1: My current [redacted] interests focus on [redacted] decision-making in large-scale complex systems, with a focus on [redacted] and accordion.

Y_1: {research, autonomous, learning, computation}

X_2: I am interested in [redacted] and adaptable optimization, high dimensional [redacted] and machine learning, reinforcement learning and [redacted]

Y_2: {robust, statistics, machine, agents}

Self-supervision + σώμα κειμένων

1. 15% των tokens επιλέγονται
2. 80% των επιλεγμένων token αντικαθίστανται με [MASK]
3. 10% των επιλεγμένων token αντικαθίστανται τυχαία
4. 10% των επιλεγμένων token παραμένουν απαράλλαχτα

TASK: Masked Language Modeling (MLM)

X_1: My current [redacted] interests focus on [redacted] decision-making in large-scale complex systems, with a focus on [redacted] and accordion.

Y_1: {research, autonomous, learning, computation}

X_2: I am interested in [redacted] and adaptable optimization, high dimensional [redacted] and machine learning, reinforcement learning and [redacted]

Y_2: {robust, statistics, machine, agents}

Self-supervision + σώμα κειμένων

1. 15% των tokens επιλέγονται

2. 80% των επιλεγ
αντικαθίστανται μ

3. 10% των επιλεγ
αντικαθίστανται τ

4. 10% των επιλεγμ
παραμένουν απαράλλαχτα

Συνάρτηση Απώλειας: Cross Entropy στα επιλεγμένα Tokens

$$\mathcal{L}_{MLM}(\theta) = -\sum_{i \in M} \log p(y_i | \hat{y}_i; \theta)$$

TASK: Masked Language Modeling (MLM)

X_1: My current [redacted] interests focus on [redacted] decision-making in large-scale [redacted] ocus on [redacted]

learning, computation}

[redacted] and adaptable [redacted] onal [redacted] forcement [redacted]

Y_2: {robust, statistics, machine, agents}

Το BERT (και RoBERTa)

Πώς κάνουμε υπολογισμούς με το «tokenization» μιας πρότασης;

Το BERT (και RoBERTa)

Πώς κάνουμε υπολογισμούς με το «tokenization» μιας πρότασης;

Βήμα 1: κάνουμε «tokenize» τις λέξεις της πρότασης, δηλαδή τις σπάμε σε tokens, π.χ.: [playing] → [play] [##ing]

Το λεξιλόγιο του BERT περιέχει 30.522 tokens – ουσιαστικά έχει λεξιλόγιο με περίπου 30.000 λέξεις.

Το BERT (και RoBERTa)

Πώς κάνουμε υπολογισμούς με το «tokenization» μιας πρότασης;

Βήμα 1: κάνουμε «tokenize» τις λέξεις της πρότασης, δηλαδή τις σπάμε σε tokens, π.χ.: [playing] → [play] [##ing]

Το λεξιλόγιο του BERT περιέχει 30.522 tokens – ουσιαστικά έχει λεξιλόγιο με περίπου 30.000 λέξεις.

Βήμα 2: Σε κάθε token αντιστοιχεί μια *στατική ενσωμάτωση* (σαν την ενσωμάτωση Word2Vec) σε διάνυσμα 768 διαστάσεων. Αυτό το διάνυσμα ενσωματώνει και συνδυάζει πληροφορίες για τη λέξη, την θέση της στην πρόταση ($1^η$, $2^η$, ...), και εάν ανήκει στην $1^η$ ή $2^η$ πρόταση.

Το BERT (και RoBERTa)

Πώς κάνουμε υπολογισμούς με το «

Βήμα 1: κάνουμε «tokenize» τις λέξεις
tokens, π.χ.: [playing] → [play] [##

Το λεξιλόγιο του BERT περιέχει
με περίπου 30.000 λέξεις.

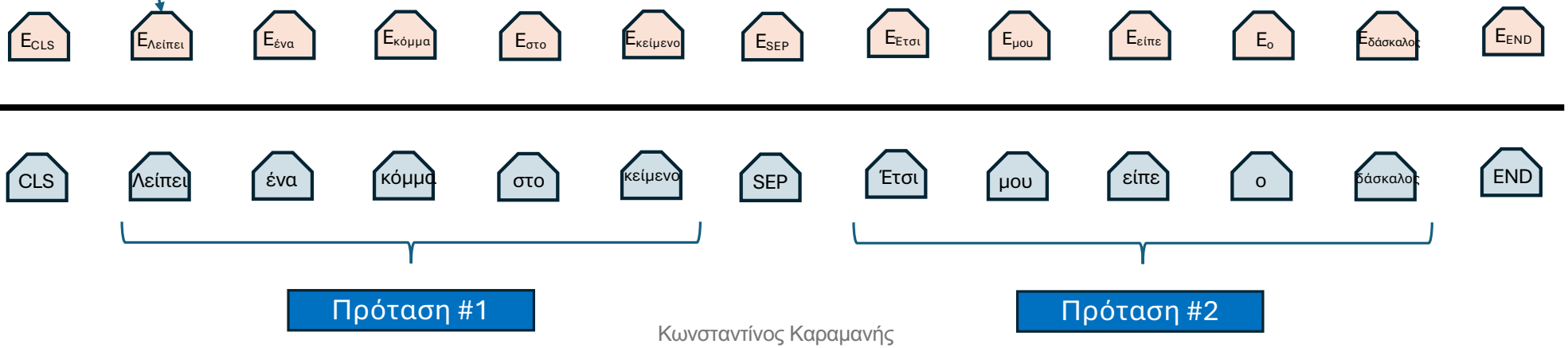
Οι συγκεκριμένες λεπτομέρειες διαφέρουν με το μοντέλο. Ένα μεγαλύτερο μοντέλο μπορεί να έχει λεξιλόγιο με ~50.000 ή ~150.000 λέξεις, και ενσωματώσεις που είναι 1.024 ή και 2.048 διαστάσεων

Βήμα 2: Σε κάθε token αντιστοιχεί μια *στατική ενσωμάτωση* (σαν την ενσωμάτωση Word2Vec) σε διάνυσμα 768 διαστάσεων. Αυτό το διάνυσμα ενσωματώνει και συνδυάζει πληροφορίες για τη λέξη, την θέση της στην πρόταση ($1^η$, $2^η$, ...), και εάν ανήκει στην $1^η$ ή $2^η$ πρόταση.

BERT και RoBERTa: token embeddings (E+P+S)

Token embedding **(learned):** για το BERT-base, ένα 768-διάστατο διάνυσμα για κάθε από τα 30.522 token στο λεξιλόγιο του BERT

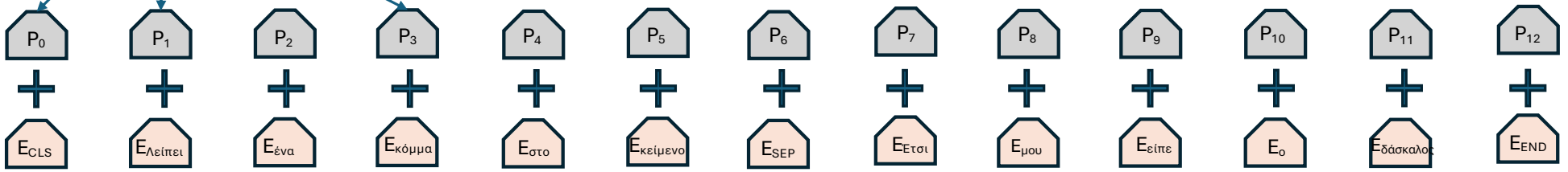
Ίδια λέξη → ίδιο token embedding



BERT και RoBERTa: token embeddings (E+P+S)

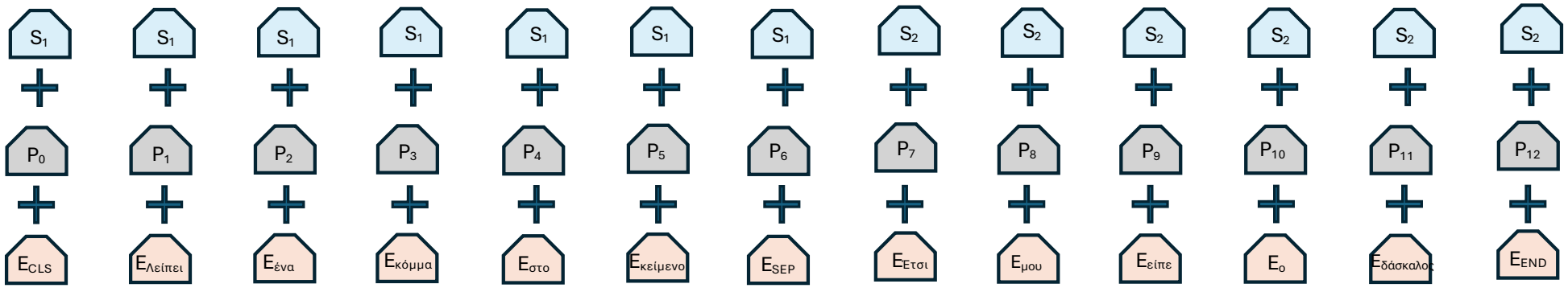
Εξαρτούνται μόνο από την θέση, όχι από την λέξη

Positional embedding **(learned):** για το BERT-base, ένα 768-διάστατο διάνυσμα για κάθε από τις 512 θέσεις



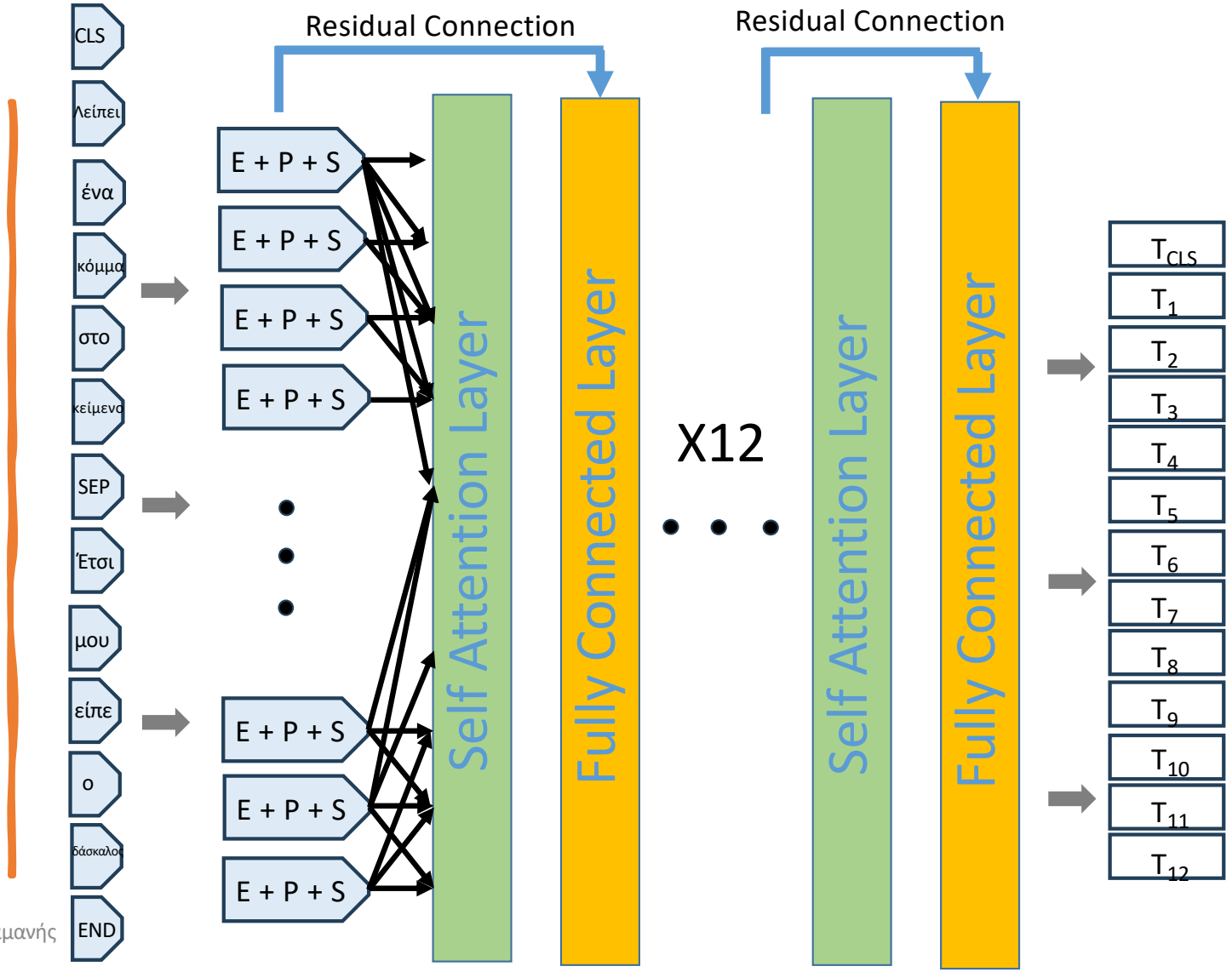
BERT και RoBERTa: token embeddings (E+P+S)

Segment embedding **(learned):** για το BERT-base, ένα 768-διάστατο διάνυσμα κωδικοποιεί S1 ή S2 (RoBERTa δεν έχει S1/S2)

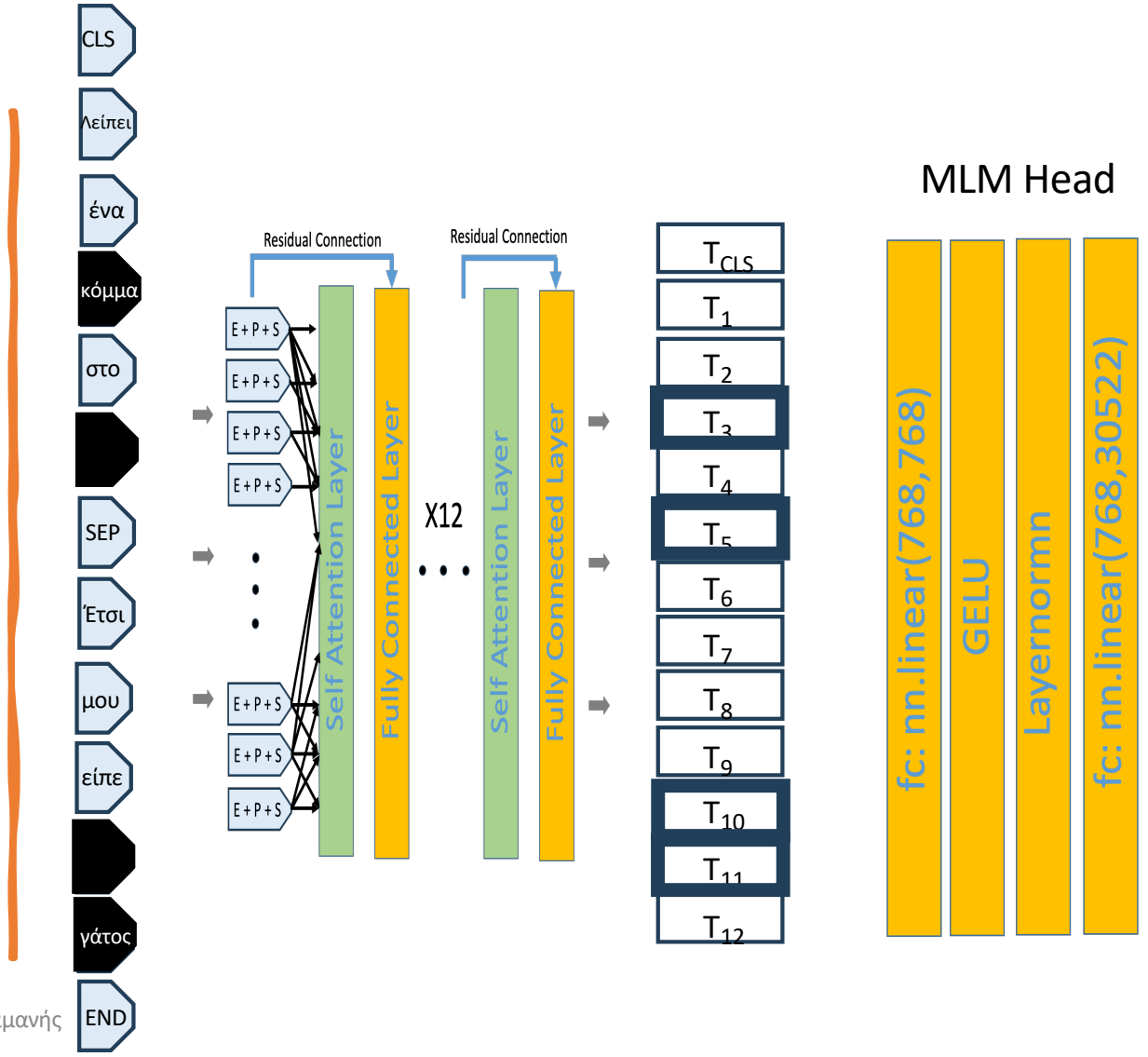


BERT Family: Αρχιτεκτονική του μοντέλου

Πώς γίνεται η εκπαίδευση με τα δεδομένα που δημιουργήσαμε με το Masked Language Modeling?

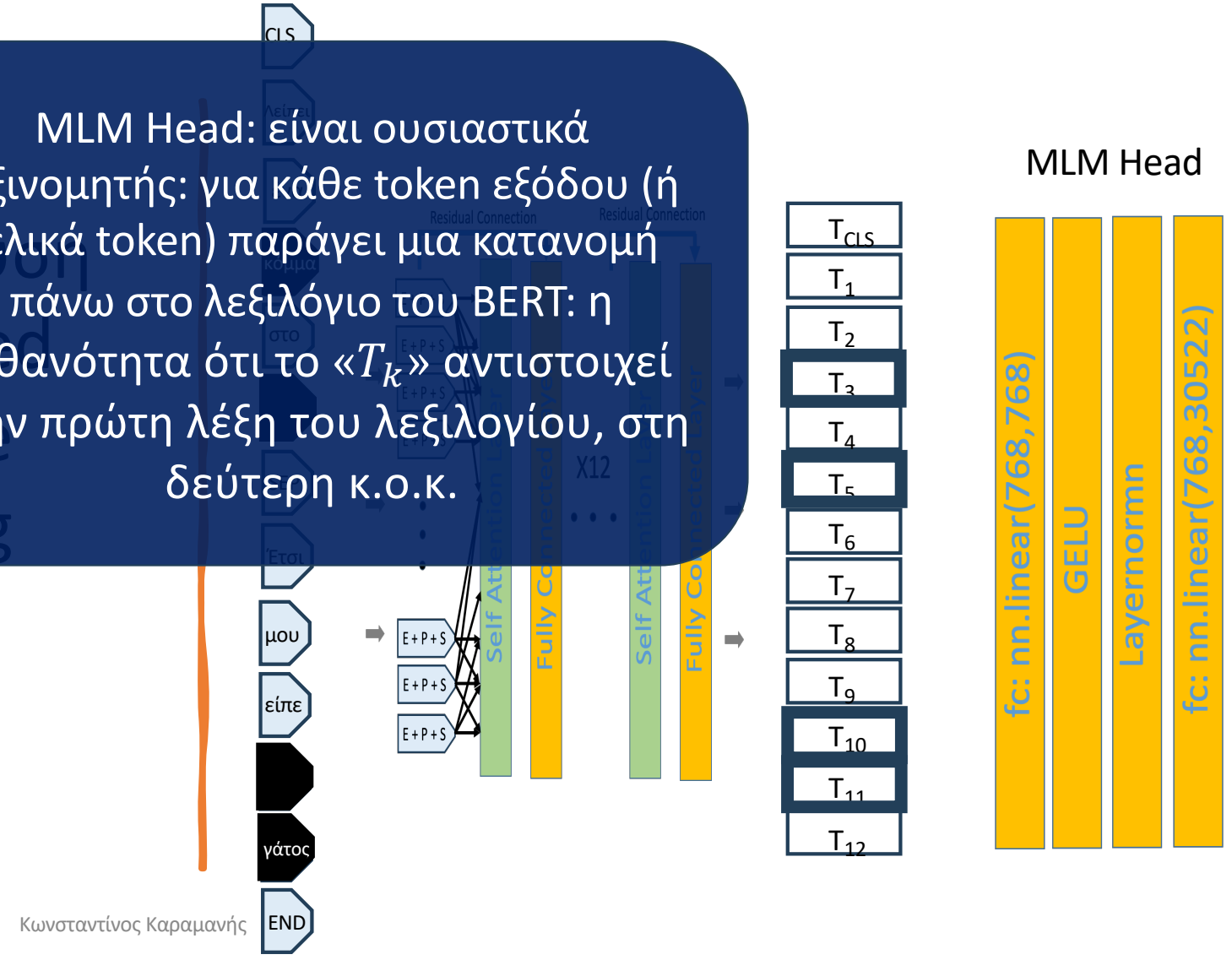


Εκπαίδευση με Masked Language Modeling (MLM)

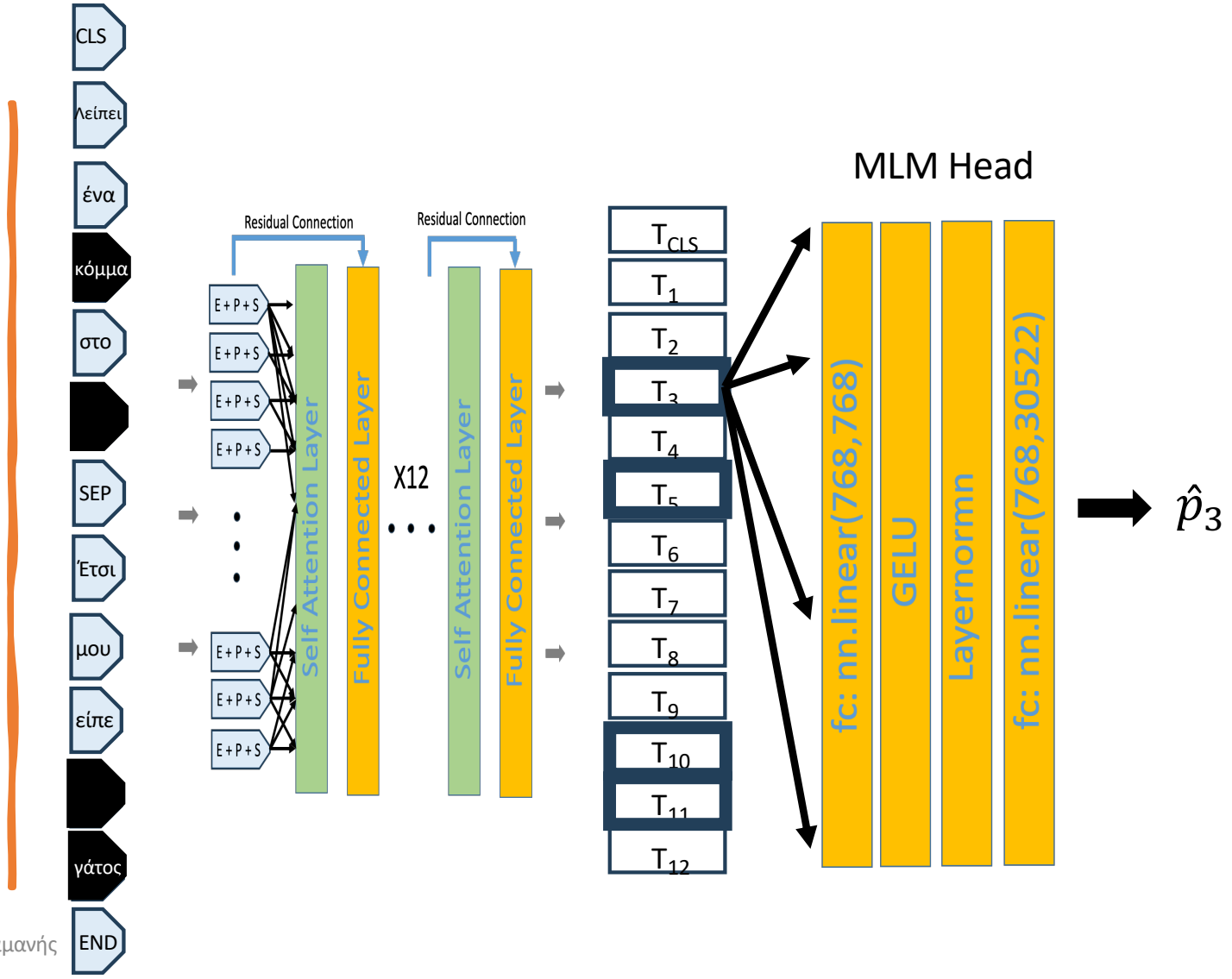


Εκπαίδευση με Masked Language Modeling (MLM)

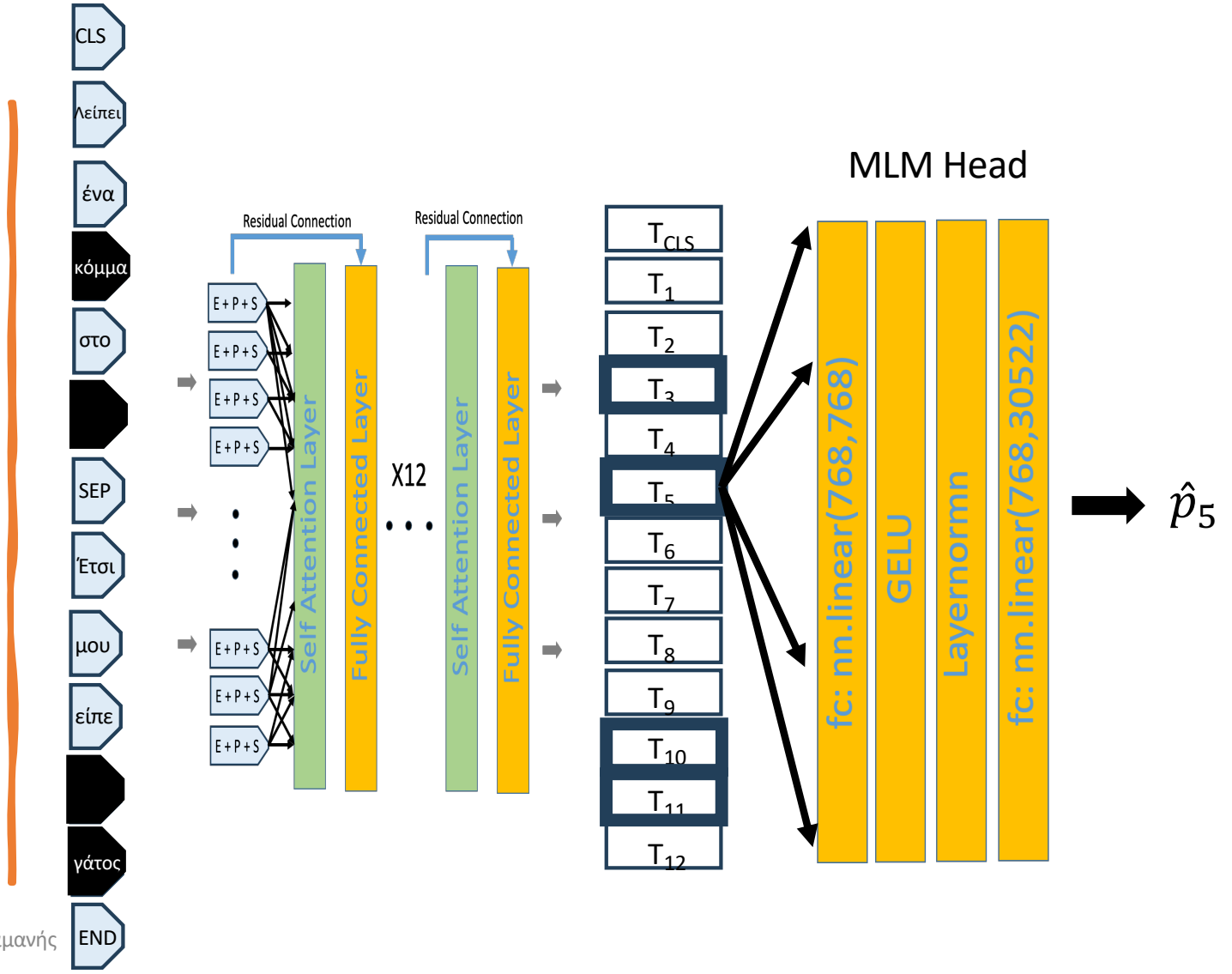
MLM Head: είναι ουσιαστικά ταξινομητής: για κάθε token εξόδου (ή τελικά token) παράγει μια κατανομή πάνω στο λεξιλόγιο του BERT: η πιθανότητα ότι το «T_k» αντιστοιχεί στην πρώτη λέξη του λεξιλογίου, στη δεύτερη κ.ο.κ.



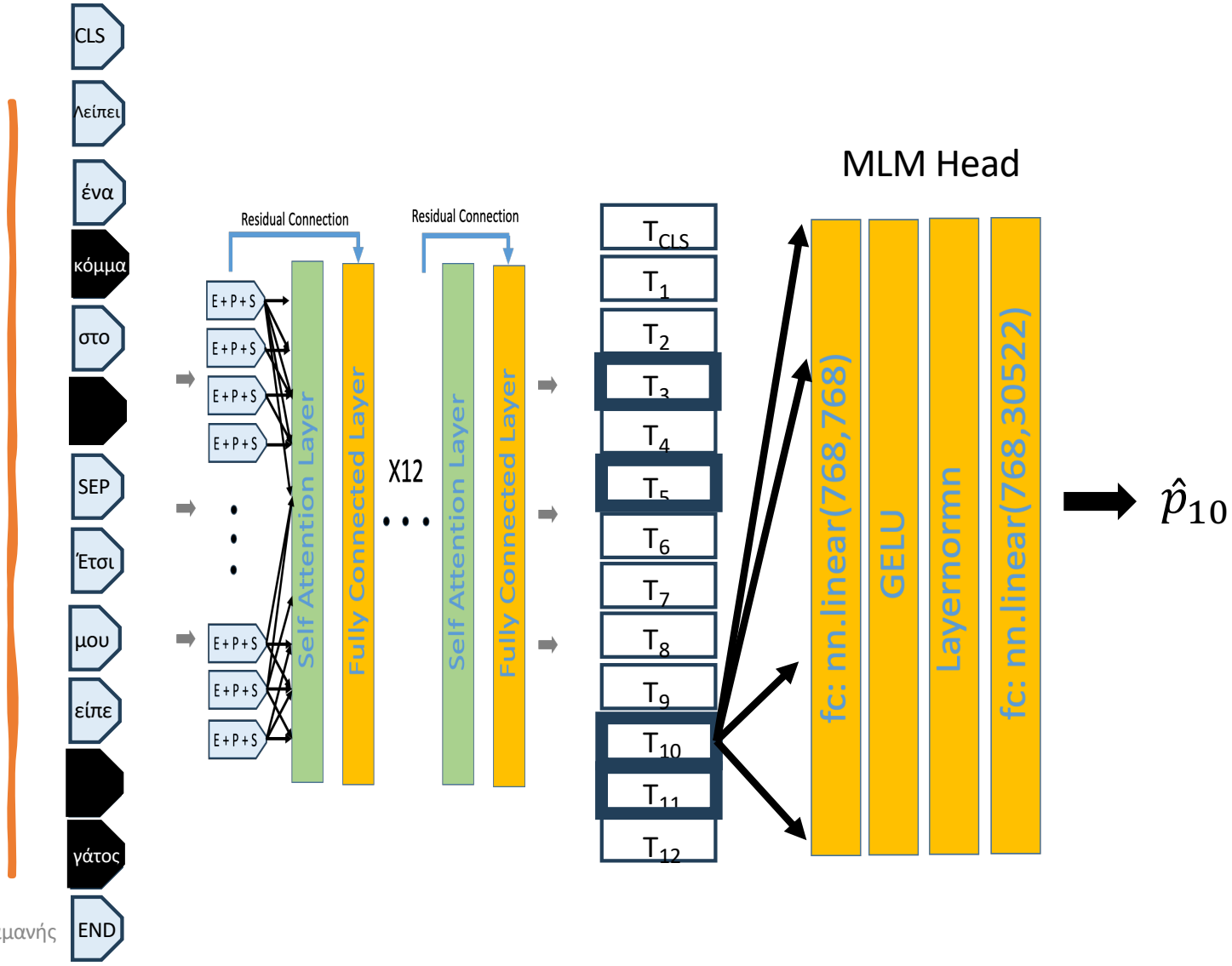
Εκπαίδευση με Masked Language Modeling (MLM)



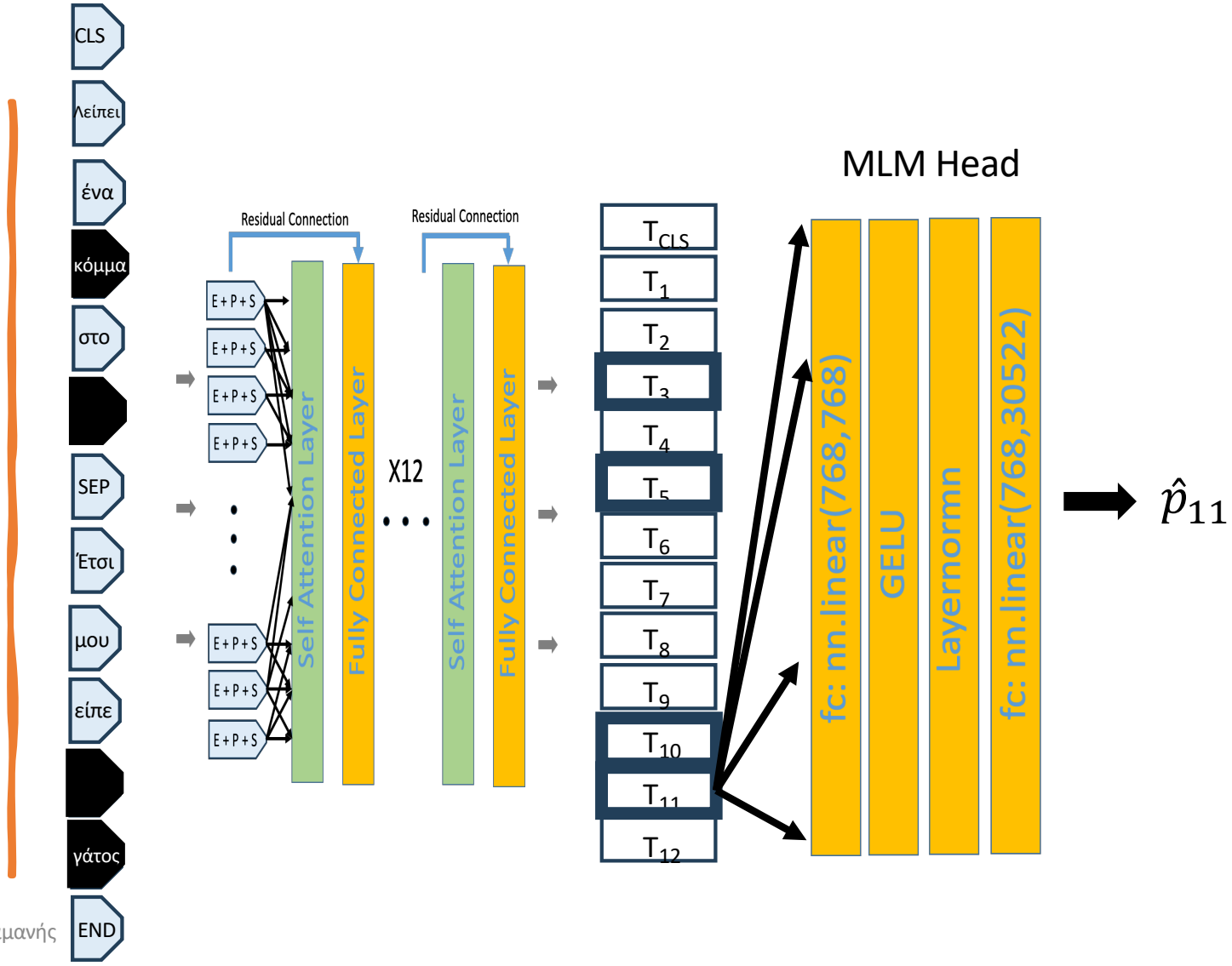
Εκπαίδευση με Masked Language Modeling (MLM)



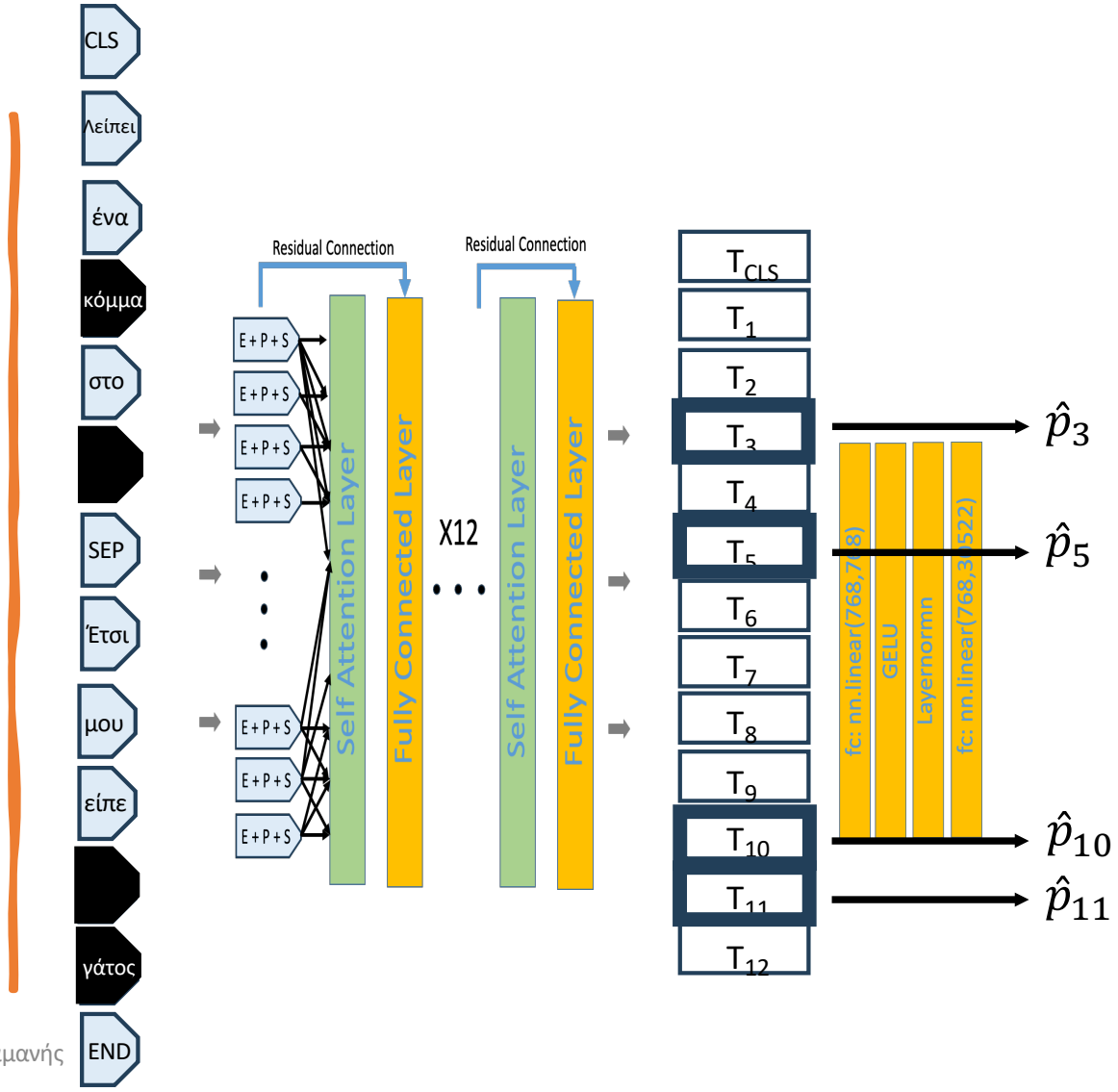
Εκπαίδευση με Masked Language Modeling (MLM)



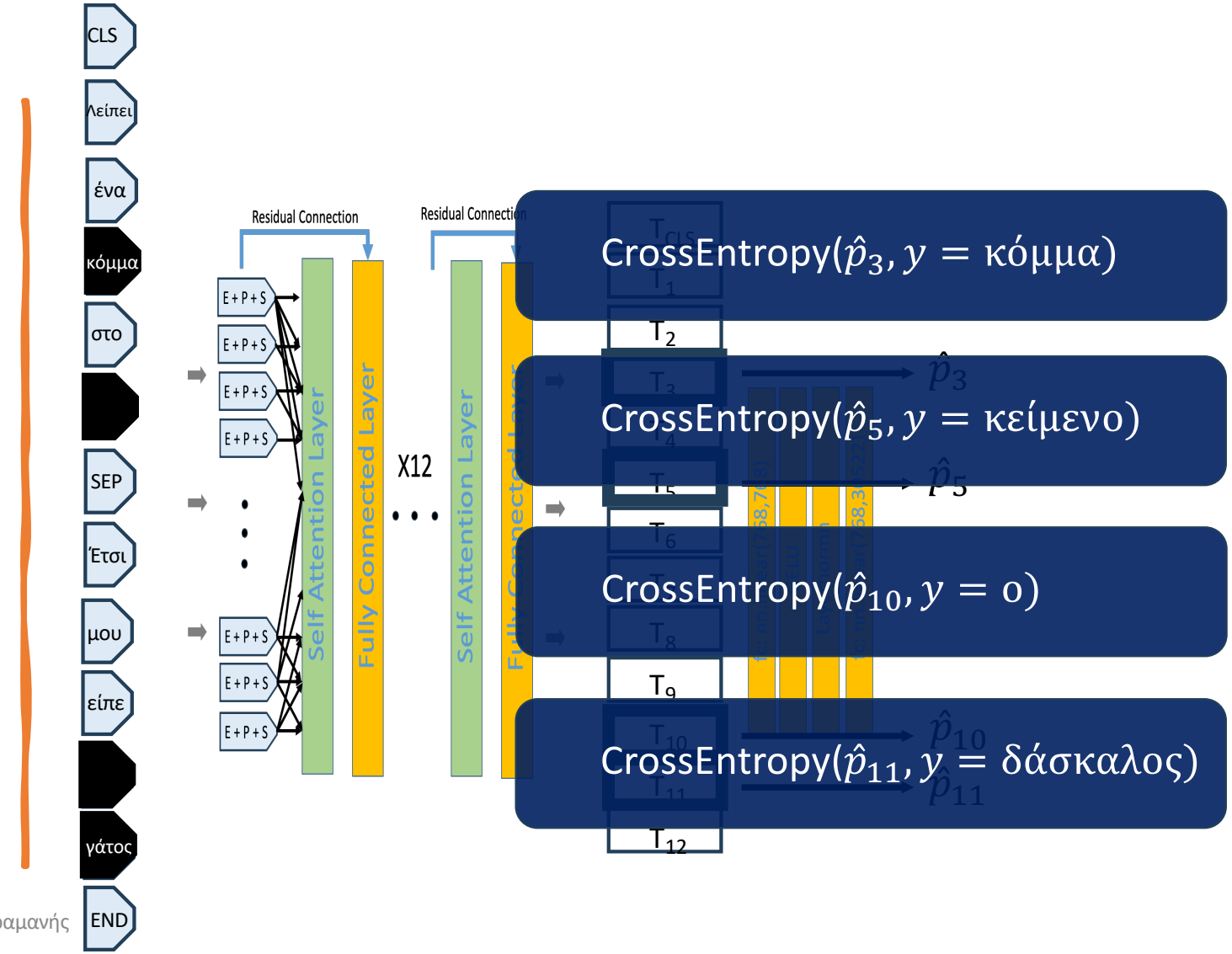
Εκπαίδευση με Masked Language Modeling (MLM)



Εκπαίδευση με Masked Language Modeling (MLM)

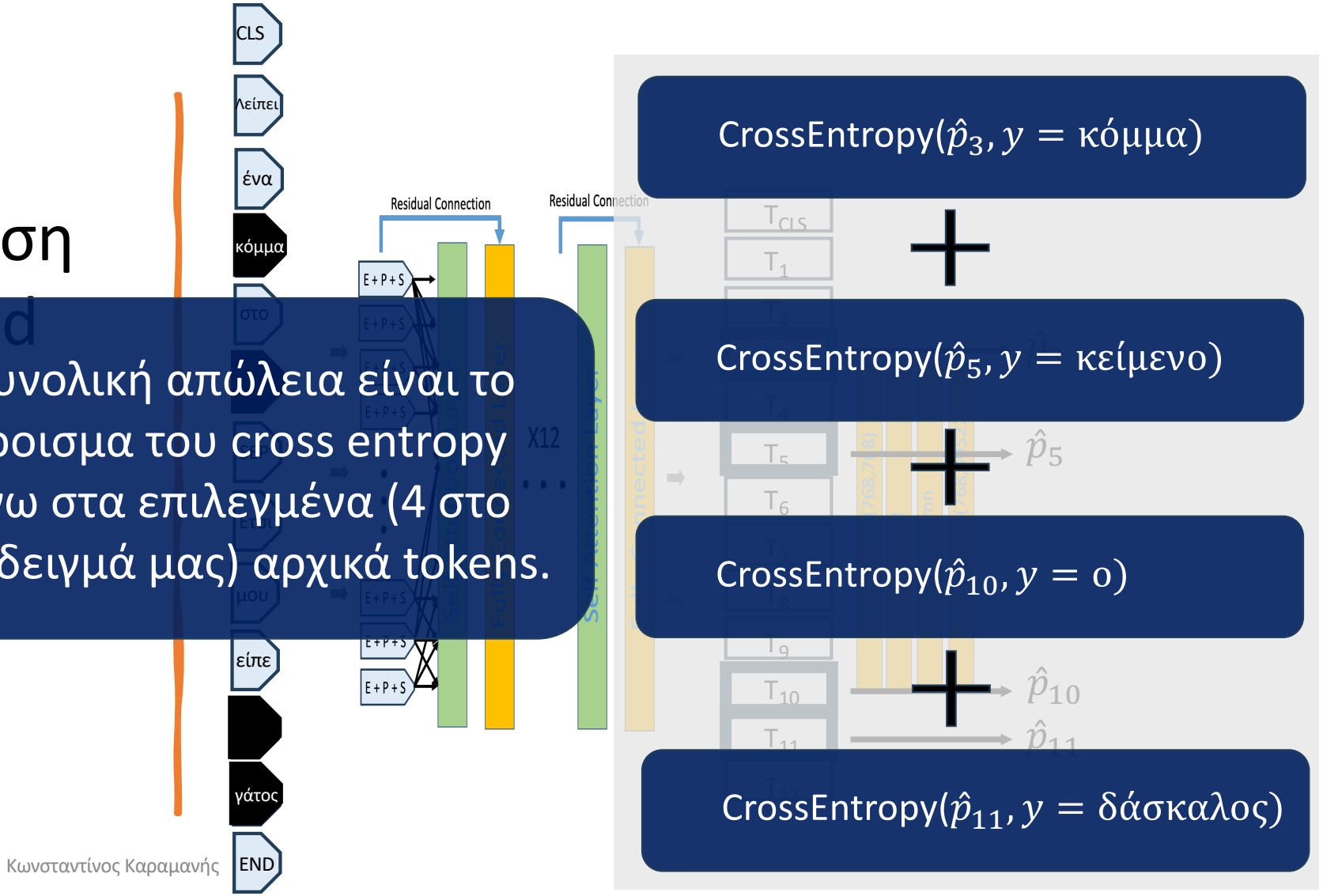


Εκπαίδευση με Masked Language Modeling (MLM)

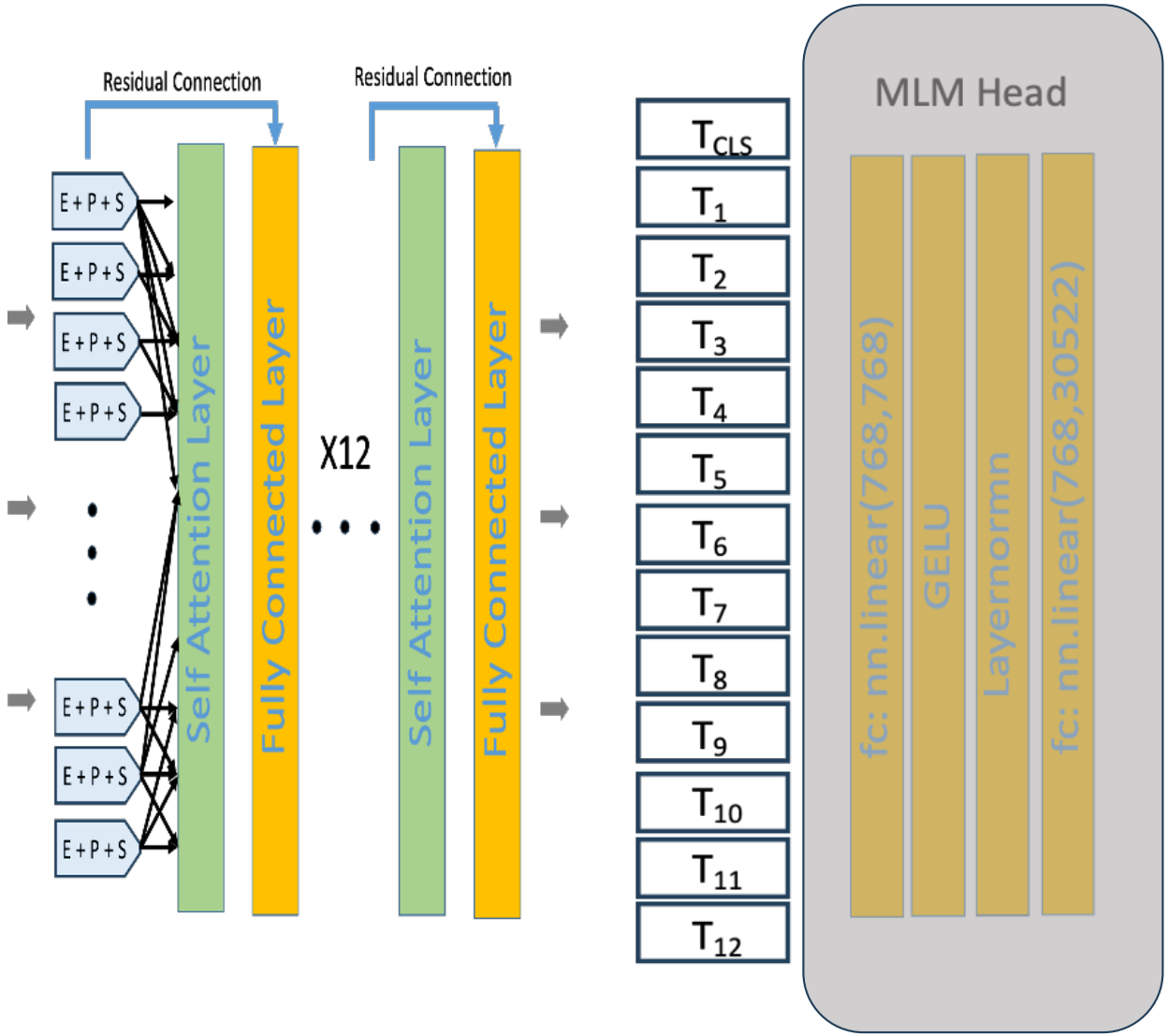


Εκπαίδευση με Masked Language Modeling (MLM)

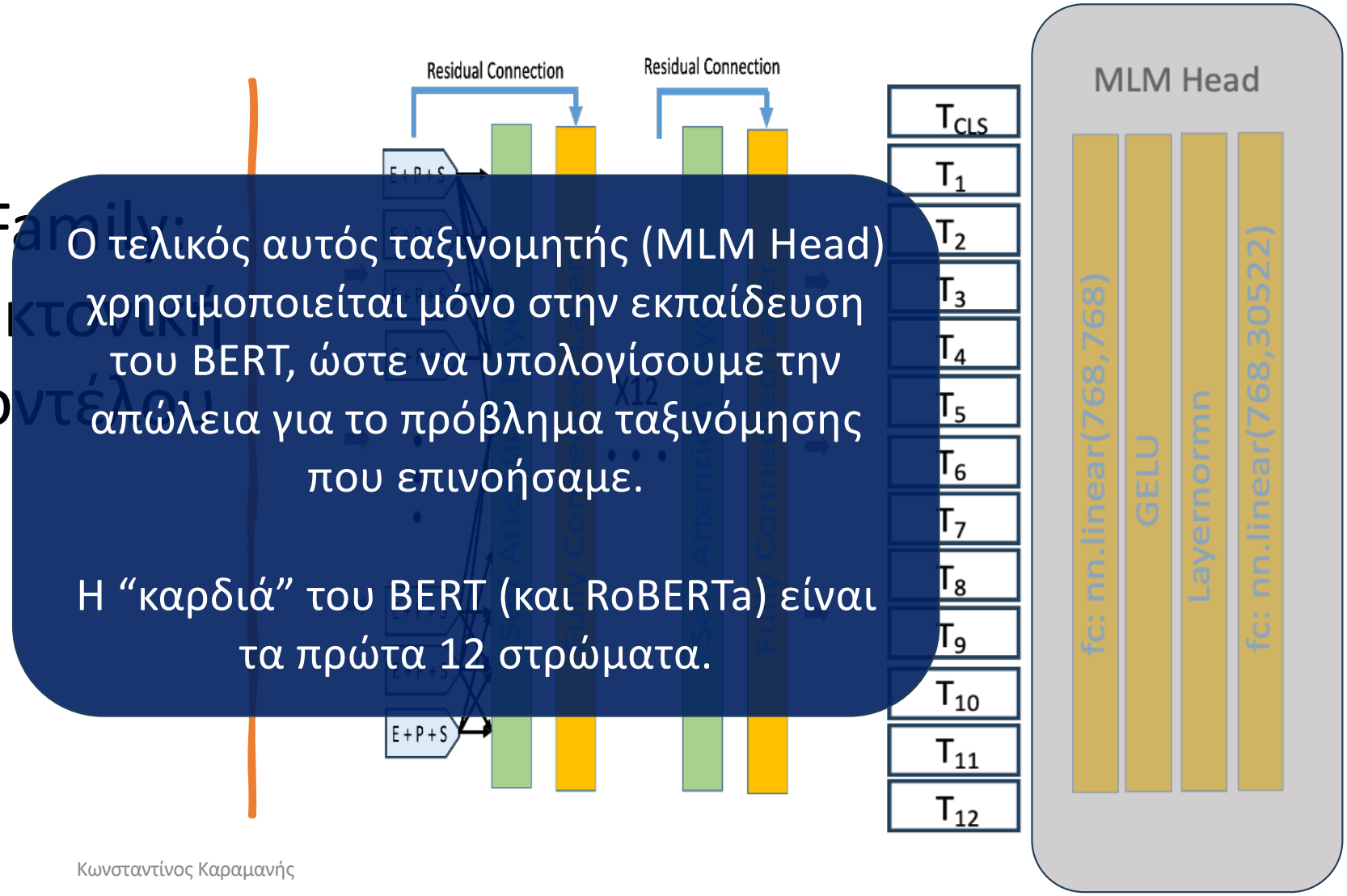
Η συνολική απώλεια είναι το άθροισμα του cross entropy πάνω στα επιλεγμένα (4 στο παράδειγμά μας) αρχικά tokens.



BERT Family: Αρχιτεκτονική του μοντέλου



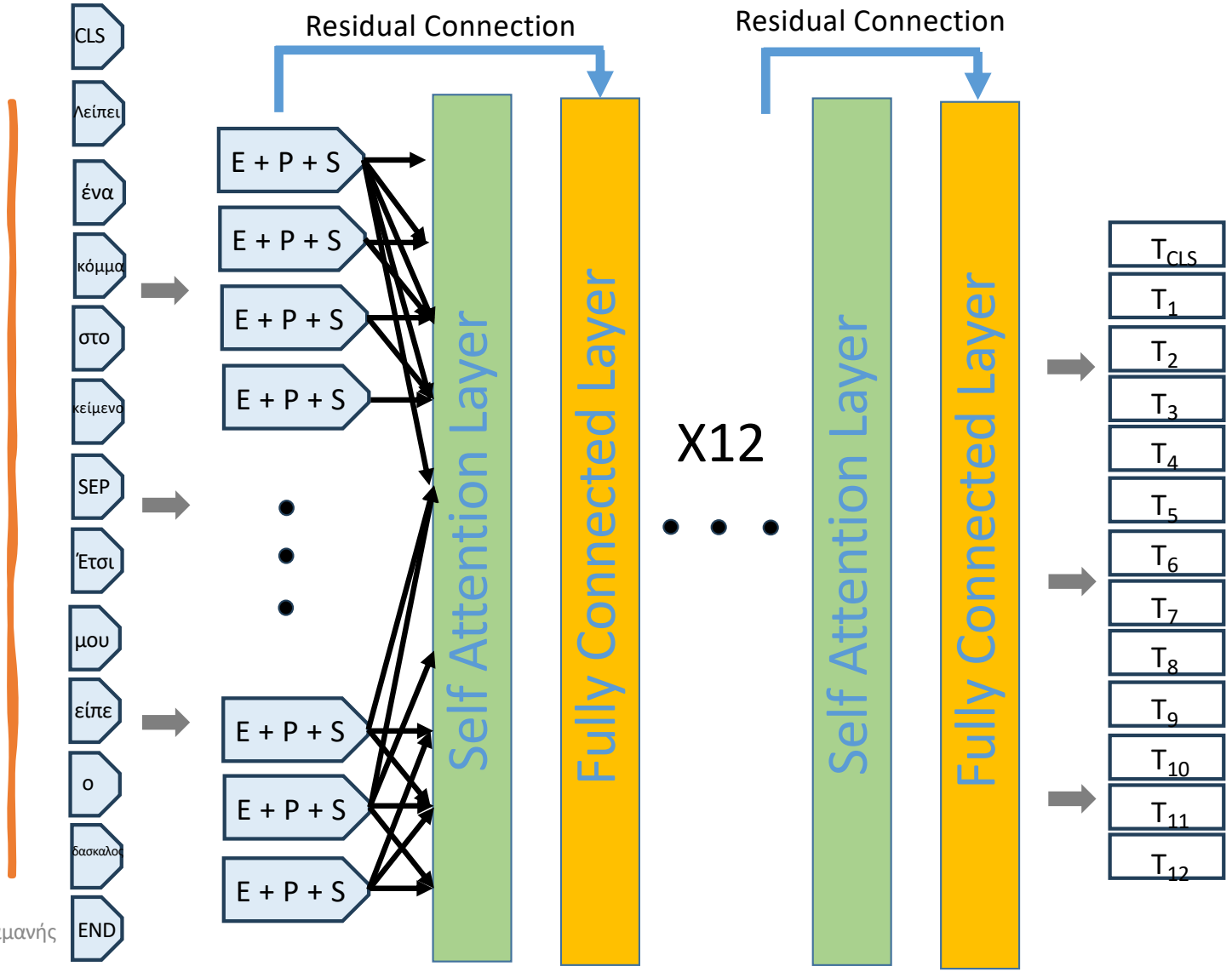
BERT Family: Αρχιτεκτονική του μοντέλου



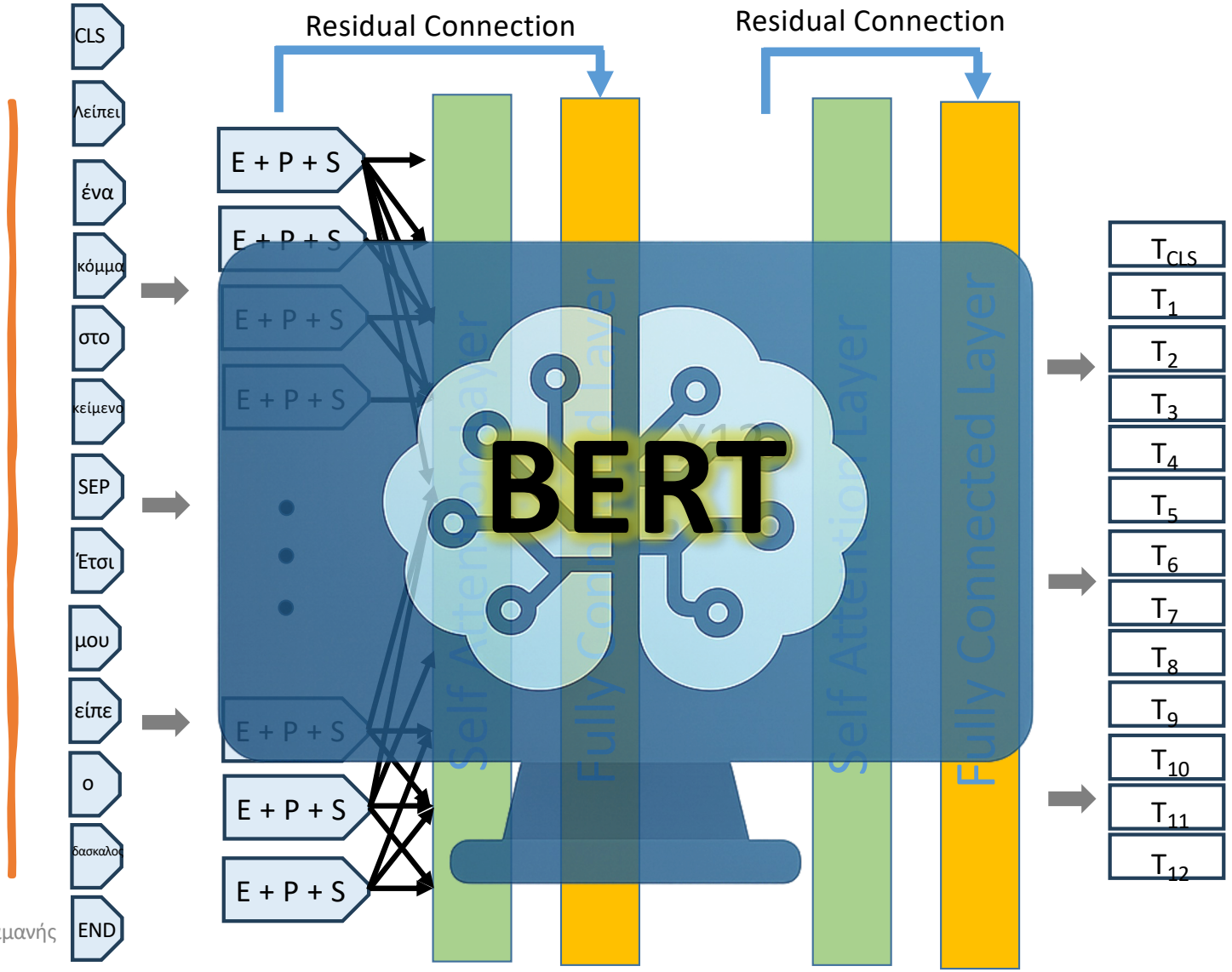
Ο τελικός αυτός ταξινομητής (MLM Head) χρησιμοποιείται μόνο στην εκπαίδευση του BERT, ώστε να υπολογίσουμε την απώλεια για το πρόβλημα ταξινόμησης που επινοήσαμε.

Η “καρδιά” του BERT (και RoBERTa) είναι τα πρώτα 12 στρώματα.

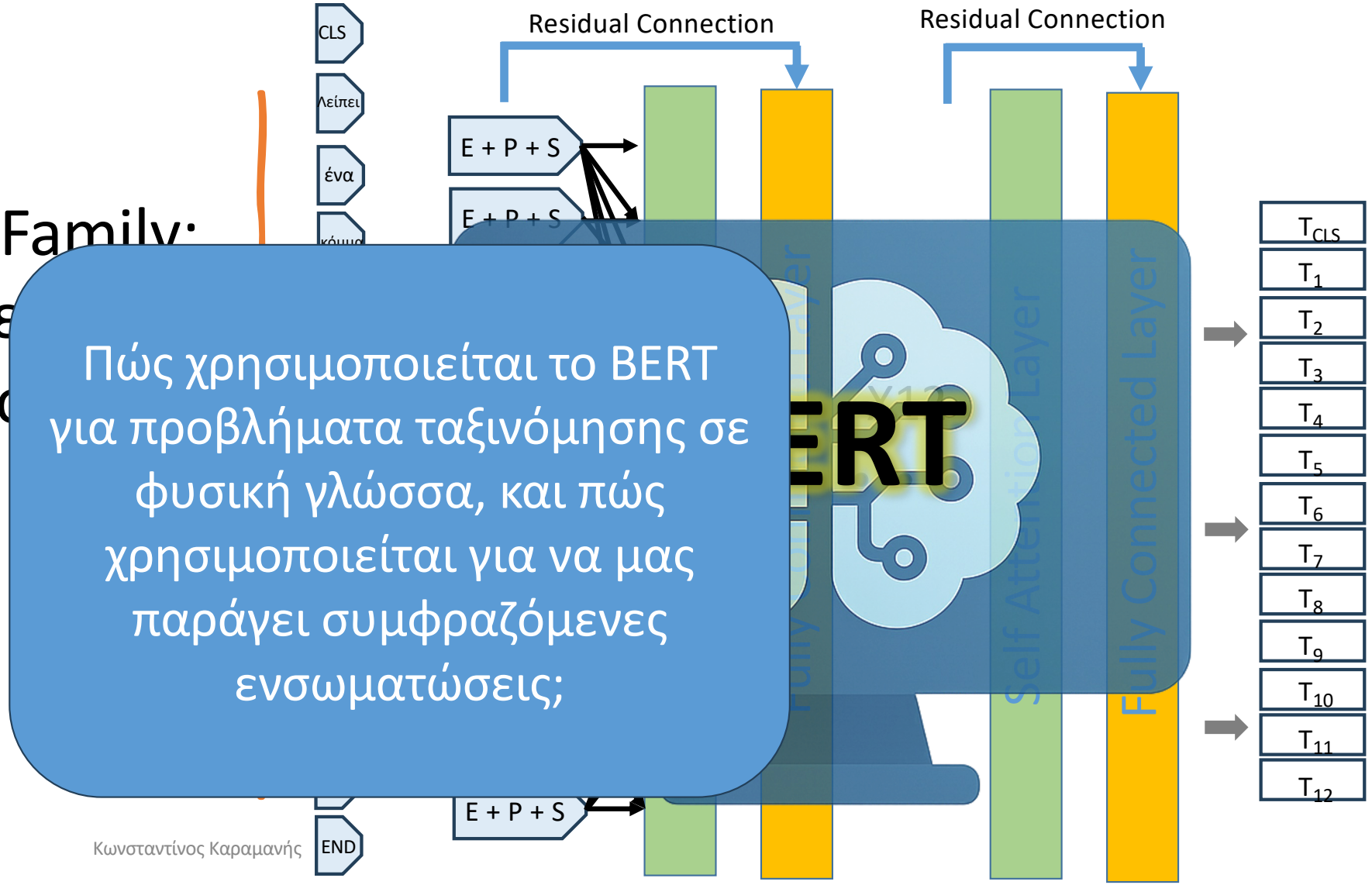
BERT Family: Αρχιτεκτονική του μοντέλου



BERT Family: Αρχιτεκτονική του μοντέλου



BERT Family:
Αρχιτε
του μο



Πώς χρησιμοποιείται το BERT για προβλήματα ταξινόμησης σε φυσική γλώσσα, και πώς χρησιμοποιείται για να μας παράγει συμφραζόμενες ενσωματώσεις;