Μοντέλα φυσικής γλώσσας

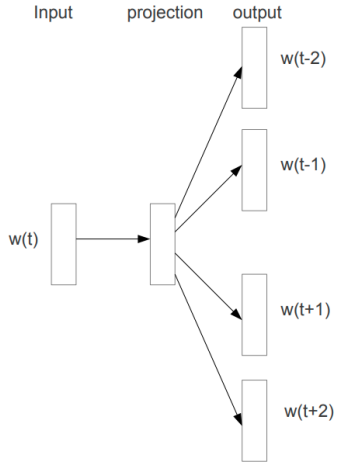
Το πρώτο βήμα για τη δημιουργία ενός μοντέλου φυσικής γλώσσας είναι η αναπαράσταση των λέξεων με τρόπο που αντιλαμβάνεται ένας υπολογιστής. Μία απλή προσσέγγιση είναι η one-hot αναπαράσταση κάθε λέξης με ένα διάνυσμα μεγέθους όσο και το λεξιλόγιο.

Μια καλύτερη προσσέγγιση είναι η «διασκορπισμένη αναπαράσταση» (distributed representation). Αυτό σημαίνει ότι σε έναν πολυδιάστατο χώρο, με αυθαίρετο αριθμό διαστάσεων, αποτυπώνονται όλες οι λέξεις ή φράσεις ή γενικότερα tokens του λεξιλογίου. Με αυτό τον τρόπο υπάρχει η έννοια της ομοιότητας των διανυσμάτων 2 tokens.

Παρακάτω βρίσκεται μια σύγκριση διάφορων μοντέλων. Κάθε μοντέλο έχει μια διαφορετική αναπαράσταση για την ίδια λέξη. Αυτή η αναπαράσταση ονομάζεται embedding. Για κάθε λέξη της πρώτης σειράς, επιστρέφονται τα 3 κοντινότερα tokens.

Model (training time)	Redmond	Havel	ninjutsu	graffiti	capitulate
Collobert (50d) (2 months)	conyers lubbock keene	plauen dzerzhinsky osterreich	reiki kohona karate	cheesecake gossip dioramas	abdicate accede rearm
Turian (200d) (few weeks)	McCarthy Alston Cousins	Jewell Arzu Ovitz	- - -	gunfire emotion impunity	- - -
Mnih (100d) (7 days)	Podhurst Harlang Agarwal	Pontiff Pinochet Rodionov	- - -	anaesthetics monkeys Jews	Mavericks planning hesitated
Skip-Phrase (1000d, 1 day)	Redmond Wash. Redmond Washington Microsoft	Vaclav Havel president Vaclav Havel Velvet Revolution	ninja martial arts swordsmanship	spray paint grafitti taggers	capitulation capitulated capitulating

Ο τελικός διανυσματικός χώρος των embeddings εξαρτάται από το είδος του μοντέλου καθώς και από το υλικό της εκπαίδευσης. Σε κάθε περίπτωση, όσον αφορά τη σχέση των λέξεων μεταξύ τους, 2 λέξεις είναι παρόμοιες αν και οι δύο εμφανίζονται συχνά στο ίδιο context.



Για παράδειγμα το μοντέλο Skip-gram, εκπαιδεύεται προσπαθώντας, με είσοδο το διάνυσμα μίας λέξης, να προβλέψει τις λέξεις που εμφανίζονται πιο κοντά σε αυτή στο υλικό εκπαίδευσης.

Μέσω αυτής της προσπάθειας, το μοντέλο καταλήγει σε διανυσματικές αναπαραστάσεις – embeddings – τα οποία αποτυπώνουν κάποια γλωσσική κατανόηση.

Για παράδειγμα:
Word2Vec, Mikolov et al
(Distributed Representations of Words and Phrases and their Compositionality).

"We found that simple vector addition can often produce meaningful results. For example, vec("Russia") + vec("river") is close to vec("Volga River"), and vec("Germany") + vec("capital") is close to vec("Berlin"). This compositionality suggests that a non-obvious degree of language understanding can be obtained by using basic mathematical operations on the word vector representations."

Η δημιουργία των embeddings είναι μια διαδικασία πολύ κοντινή με την πρόβλεψη της επόμενης λέξης (next word prediction) ή ακόμα με την πρόβλεψη της επόμενης πρότασης (next sentence prediction).

Σε αυτές τις διαδικασίες είναι πολύ σημαντική η ικανότητα του μοντέλου να διαχειρίζεται το context, είτε σε εύρος μερικών λέξεων είτε σε εύρος μερικών παραγράφων.

Σε αυτή την προσπάθεια επέκτασης του context έχουν δημιοργηθεί μοντέλα RNN (Recurrent Neural Networks), LSTM (Long short-term memory) και άλλων ειδών αρχιτεκτονικές.

BERT

Σε αυτή την εργασία εξετάζεται μία έκδοση του μοντέλου BERT, το οποίο χρησιμοποιεί μια αρχιτεκτονική "multi-layer bidirectional Transformer encoder".

Συνοπτικά τα πλεονεκτήματα αυτού του μοντέλου είναι τα εξής:

- Bidirectional: λαμβάνει υπ' όψην context και από τις δύο μεριές της λέξης κατά την εκπαίδευση.
- Transformer: για την σημαντικότητα των γύρω λέξεων στο context, χρησιμοποεί την έννοια του "Attention", βάση της οποίας μπορεί να συμπεριλάβει context ακόμη και πολύ μακριά από την λέξη, με χαμηλό υπολογιστικό κόστος.
- Fine-tuning: έπειτα από την αρχική εκπαίδευση (pre-training), το μοντέλο μπορεί με μικρό υπολογιστικό κόστος να εκπαιδευεί σε διάφορα tasks.

Ανάλυση του μοντέλου BERT

Transformer:

Η βασική αρχικτεκτονική ενός μοντέλου transformer είναι η παρακάτω. Το αριστερό τμήμα ονομάζεται “Encoder” ενώ το δεξί ονομάζεται “Decoder”.
Συνοπτικά, ο Encoder είναι υπεύθυνος για την δημιουργία των embeddings, ενώ ο Decoder είναι υπεύθυνος για την εκτέλεση κάποιας γλωσσικής εργασίας, όπως για παράδειγμα ή μετάφραση των embeddings που παρήγαγε ο Encoder.

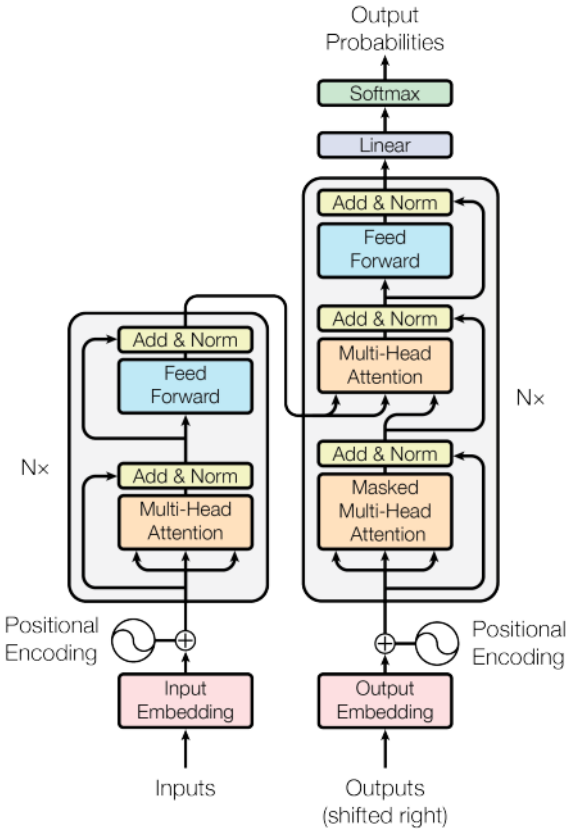
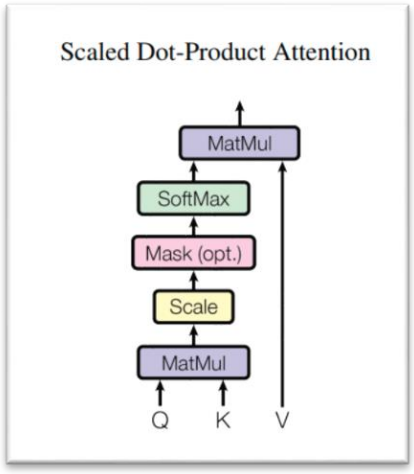


Figure 1: The Transformer - model architecture.

Το μοντέλο BERT βασίζεται μόνο σε Encoders.

Παρακάτω αναλύονται πιο συγκεκριμένα οι Encoders.

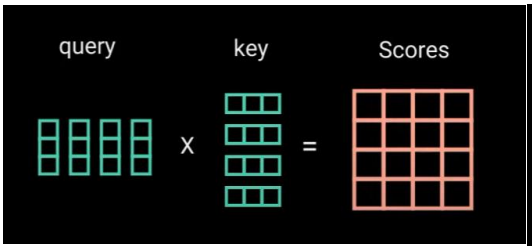
Self-Attention:



Για κάθε λέξη υπάρχουν διανύσματα Q (query), K (key), V (value), τα οποία προκύπτουν το καθένα μέσω ενός linear layer από το διάνυσμα της λέξης.

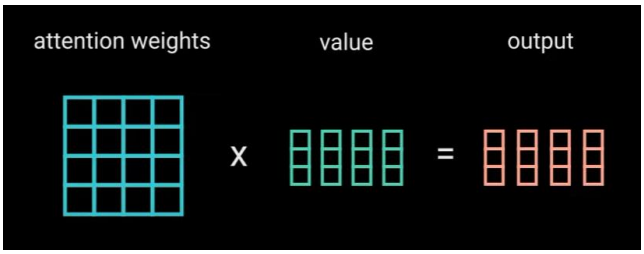
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Ο πολλαπλασιασμός των Q και K όλων των λέξεων εισόδου παράγει έναν πίνακα με βάρη τα οποία αφορούν την «σημαντικότητα» της μίας λέξης προς την άλλη.



	Hi	how	are	you
Hi	98	27	10	12
how	27	89	31	67
are	10	31	91	54
you	12	67	54	92

Τέλος, πολλαπλασιάζοντας με τα διανύσματα V των λέξεων εισόδου, τα διανύσματα των λέξεων που έχουν μεγαλύτερη σχέση μεταξύ τους επιρεάζονται το ένα από το άλλο.
Έτσι το διάνυσμα που αντιστοιχεί στην ίδια λέξη στη έξοδο του μηχανισμού “attention” είναι διαφορετικό από ότι το διάνυσμα εισόδου της. Έτσι αποτυπώνεται το context.



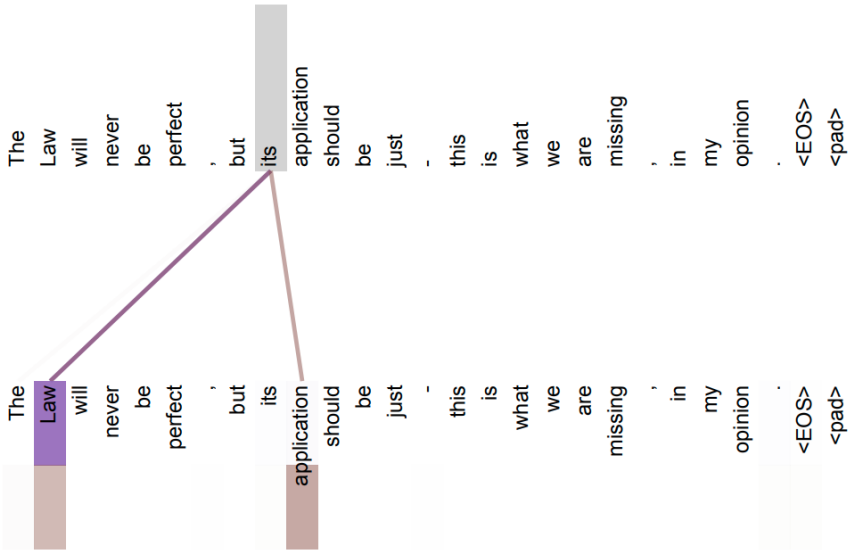
(The A.I. Hacker - Michael Phi, YouTube)

Multi Head Attention

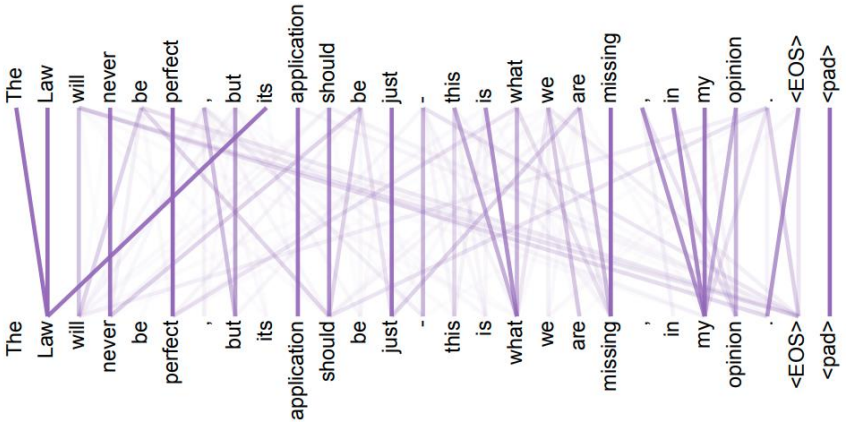
Η παραπάνω περιγραφή αφορά ένα Attention Head. Εκτελώντας αυτή την διαδικασία πολλές φορές, δηλαδή δημιουργώντας πολλαπλά Attention Heads, προκύπτει το βασικό μέρος του transformer, το Multi-Head Attention. Είναι σημαντικό ότι καθώς δεν υπάρχει εξάρτηση του ενός Attention Head από τα άλλα, όλα τα Attention Heads μπορούν να υπολογιστούν ταυτόχρονα με παραλληλοποίηση.

Επίσης κάθε Attention Head μπορεί να «μάθει» να εκτελεί διαφορετικές «εργασίες» όπως φαίνεται παρακάτω:

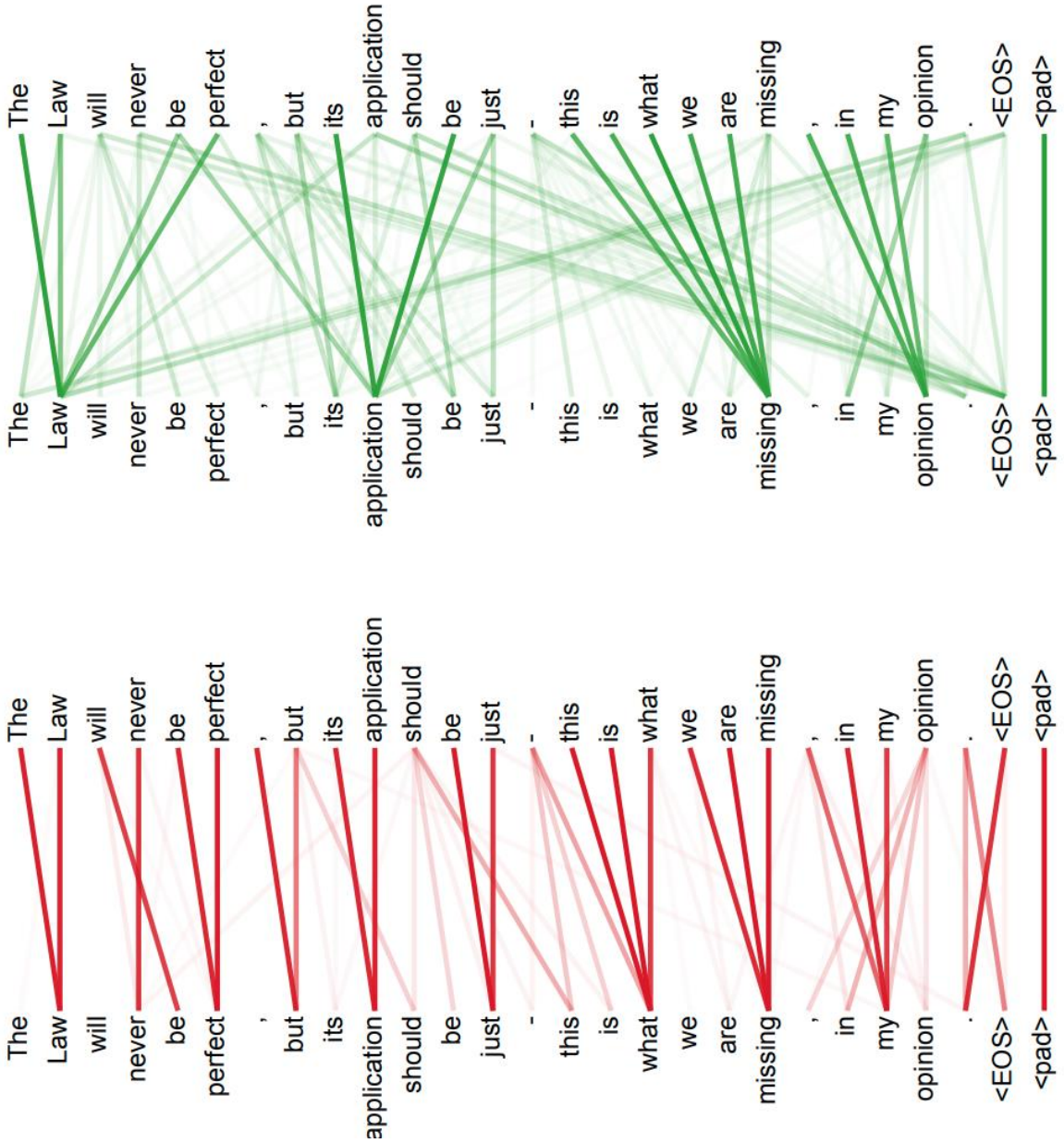
Μεμονωμένο Attention Vector για μια λέξη:

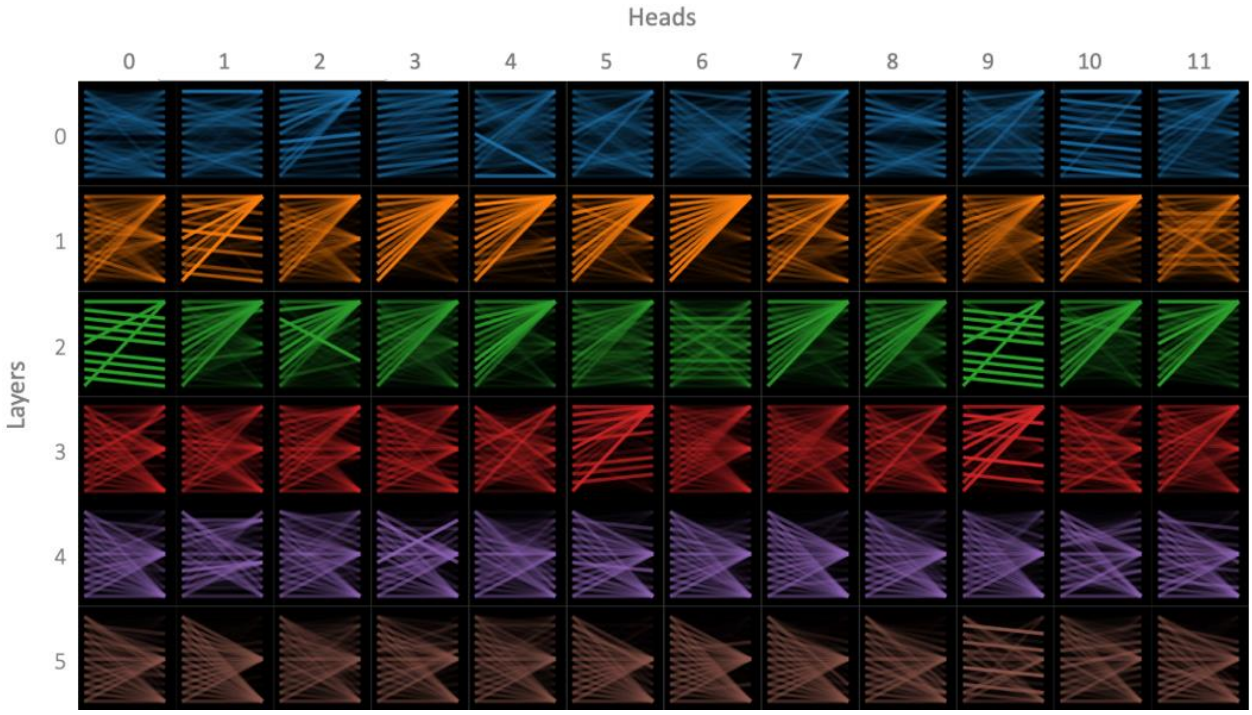


Ένα ολόκληρο Attention Head:



Διαφορετικά Attention Heads:



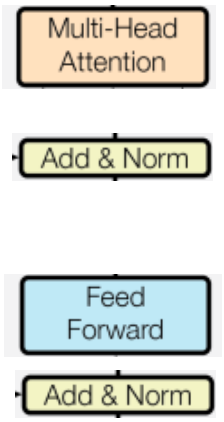


Model view (first 6 layers) for input sentences “the rabbit quickly hopped” and “the turtle slowly crawled”.

https://towardsdatascience.com/deconstructing-bert-part-2-visualizing-the-inner-workings-of-attention-60a16d86b5c1

Επισκόπηση αρχιτεκτονικής

Το μοντέλο $BERT_{BASE}$ στο οποίο βασίζονται τα μοντέλα που θα χρησιμοποιηθούν στην εργασία αποτελείται από 12 Encoders, οι οποίοι έχουν μέγεθος κρυφού layer 768 και ο κάθε encoder έχει 12 Attention Heads. Συνολικά το μοντέλο έχει 110 εκατομμύριες παραμέτρους.




```
(encoder): BertEncoder(
  (layer): ModuleList(
    (0): BertLayer(
      (attention): BertAttention(
        (self): BertSelfAttention(
          (query): Linear(in_features=768, out_features=768, bias=True)
          (key): Linear(in_features=768, out_features=768, bias=True)
          (value): Linear(in_features=768, out_features=768, bias=True)
          (dropout): Dropout(p=0.1, inplace=False)
        )
        (output): BertSelfOutput(
          (dense): Linear(in_features=768, out_features=768, bias=True)
          (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
          (dropout): Dropout(p=0.1, inplace=False)
        )
      )
      (intermediate): BertIntermediate(
        (dense): Linear(in_features=768, out_features=3072, bias=True)
      )
      (output): BertOutput(
        (dense): Linear(in_features=3072, out_features=768, bias=True)
        (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
        (dropout): Dropout(p=0.1, inplace=False)
      )
    )
```

Ειδικά tokens

Γενικά το μοντέλο BERT χρησιμοποιεί τον WordPiece tokenizer για την παραγωγή tokens από το κείμενο.
Επιπρόσθετα χρησιμοποιεί 3 ειδικά tokens.

1. [CLS]: Αυτό το token βρίσκεται πάντα στην αρχή της πρότασης ή γενικότερα της εισόδου στο μοντέλο. Ονομάζεται Classification token και όπως προδίδει το όνομα, είναι σχεδιασμένο να χρησιμοποιηθεί σε classification tasks.
   [SEP]. Αυτό το token υπάρχει πάντα στο τέλος της εισόδου στο μοντέλο. Αν υπάρχει ένα επιπλέον token [SEP] πριν το τέλος της εισόδου, τότε αυτό διαχωρίζει την $1^{η}$ πρόταση εισόδου από την $2^{η}$.
2. [MASK]: Αυτό το token χρησιμοποιείται μόνο στο pre-training. Ουσιαστικά κρύβει μία λέξη στην πρόταση, την οποία το μοντέλο προσπαθεί να μαντέψει. Αυτό το token είναι σημαντικό καθώς επιτρέπει στο μοντέλο να μάθει ταυτόχρονα από το αριστερό και δεξί context, όπως θα έκανε ένας άνθρωπος όταν συναντάει μια άγνωστη λέξη ή προσπαθεί να μαντέψει το κενό.

Pre-training

Η φάση του pre-training περιλαμβάνει 2 tasks.

Task 1: Masked LM

Χρησιμοποιώντας το token [MASK], κρύβεται μια λέξη από την έισοδο του μοντέλο και αυτό εκπαιδεύευαι ώστε να βρεί αυτή τη λέξη, λύνοντας ουασιαστικά μία άσκηση Cloze.

Task2: Next sentence prediction

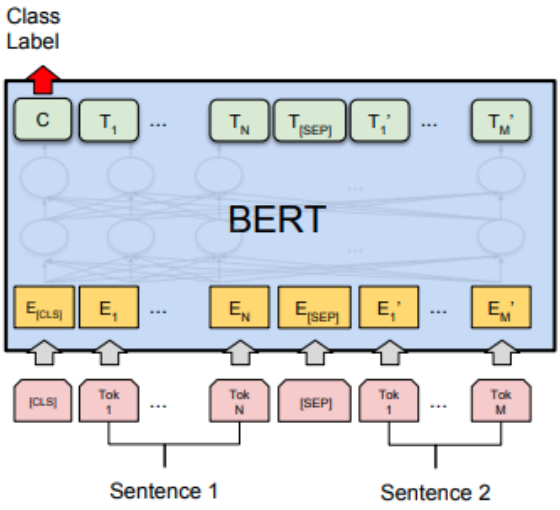
Είναι η βάση για εργασίες όπου το μοντέλο βγάζει συμπεράσματα ή απαντάει σε ερωτήσεις. Αυτού του είδους το task θα είναι ιδιαίτερα χρήσιμο στην παρούσα εργασία.
Αυτό γίνεται στο pre-training με τον εξής τρόπο: Στο μοντέλο δίνονται 2 προτάσεις χωρισμένες με [SEP]. Αυτές οι προτάσεις μπορεί να έπονται η μία της άλλης στο υλικό εκπαίδευσης ή η $2^{η}$ πρόταση να είναι τυχαία σε σχέση με την $1^{η}$. Το μοντέλο εκπαιδεύεται να βρίσκει πότε η $2^{η}$ πρόταση έπεται της $1^{ης}$.

Fine-tuning

Οι εργασίες fine-tuning υποθέτουν την ύπαρξη ενός μικρού σχετικά classifier στο τέλος του μοντέλου. Συνήθως επαρκεί η χρήση του διανύσματος εξόδου για το token [CLS] ως είσοδος σε αυτόν τον classifier.

"The optimal hyper-parameter values are task-specific, but we found the following range of possible values to work well across all tasks:

- *Batch size: 16, 32*
- *Learning rate (Adam): 5e-5, 3e-5, 2e-5*
- *Number of epochs: 2, 3, 4"*



(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

Feature Based Approach

Μια άλλη δυνατότητα θα ήταν να χρησιμοποιηθεί το μοντέλο για feature extraction, δηλαδή να χρησιμοποιηθούν τα embeddings εξόδου κατευθείαν και όχι το token [CLS].

Αυτή η περίπτωση έχει το πλεονέκτημα ότι χρειάζεται να υπολογισθούν μόνο μια φορά τα διανύσματα των εισόδων που μας ενδιαφέρουν και έπειτα μπορούν να χρησιμοποιηθούν σε έναν οποιονδήποτε classifier.

Συγκεκριμένα φαίνεται ότι η συνένωση των 4 τελευταίων hidden layers ως features, δίνει τα καλύτερα αποτελέσματα.