

## Natural Language Processing (NLP)

- Ανήκει στον ταξιδιό της μηχανικής μάθησης & της γνωστολογίας
- Τα NLP προβλέπουν να κατανοήσουν τα λόγια που έχουν σχέση με την human language.
- Διότι: ~ κατανόηση μεμονωμένων λέξεων  
    ~ κατανόηση του context στις λέξεις

### NLP tasks and examples

#### Ταξινόμηση

#### 1. Classifying whole sentences:

Αποτελεί το εναπόθετα μες αριθμός, ανήκει στο spam email, προσδιορίζει εάν μια πρόσωπο είναι γραφικότητας εως ή αν & προτίθεται ευθετιστική λογική.

#### 2. Classifying each word in a sentence:

Προσδιορίζει μια λέξη σε ποιά κατηγορία ανήκει (ανθρώπος, όντα, επίθετο) ή υποστηρίζει την αυτότιτη της λέξης (πρόσωπο, τονοθεσία, οργανισμός)

#### 3. Generating text content:

Παρέχεται ένα κείμενο με κενά διαλογέα και το μοντέλο, γεμίζει τα κενά με λογικό και συνειδητό τρόπο.

#### 4. Extracting an answer from a text

Δίνεται μια ερώτηση και ένα πλαίσιο, το μοντέλο εξαγρίζει ανάγνωσης στην ερώτηση με βάση την πληροφορία που του παρέχεται στο πλαίσιο.

#### 5. Generating a new sentence from an input text:

Παρέχεται διαλογός ένα κείμενο, και το μοντέλο μεταφράζει το κείμενο, ή παρέχει ένα διαφορετικό κείμενο με παρόμοιο περιεχόμενο όπως το αρχικό.

Τα NLP δεν περιορίζονται στα κείμενα, αλλά σταχειρίσματα και complex tasks. Όπως αναχωρήση αριθμητικών, περιγραφή ειδών κλπ.

### Γιατί είναι πρόβλημα

- Οι υπολογιστές σεν επεφερχόσανται τις πληροφορίες με τον ίδιο τρόπο όπως οι ανθρώποι
- Για μοντέλα μηχανικής μάθησης είναι δύκοτο να ερμηνεύσουν εναυσθέντα.
- Για να μπορεί να το ερμηνεύσει, το κείμενο πρέπει να υποβάλλεται σε επεφερχασία, με τρόπο που να επιτρέπει στο μοντέλο να μάθει από αυτό. Και επειδή η γλώσσα είναι περιπλοκή, πρέπει να ενεργούν προσεταιρικά πώς πρέπει να γίνει αυτή η επεφερχασία.

## Transformers Models, what can they do?

### Transformers Models

- Τα μοντέλα αυτά χρησιμοποιούνται για την επίλυση ολών των ειδών εργασιών NLP.

- Ηερικές εταιρείες που χρησιμοποιούν μοντέλα Hugging Face και transformer, και ποιά σημαντικά τα μοντέλα τους:

- Allen Institute for AI ◦ Typeform ◦ Grammarly
- Google AI ◦ Microsoft ◦ Asteroid - team
- Facebook AI ◦ Musixmatch

Οι transformers library παρέχουν διαφόρες για την δημιουργία ναι χρήση διανομέων μοντέλων

### Pipeline

- Μια σειρά διαδικασιών βημάτων ή εργασιών που εκτελείται μαζί από διαδικασίες, με την έξοδο του ένας να αποτελεί την είσοδο του επόμενου.
- Αυτό το μοντέλο ρωτά εργασίων χρησιμοποιείται για την αυτοματοποίηση & βελτιστοποίηση των διαδικασιών.

### Pipeline function

- Μια συνάρτηση ή μέθοδος που χρησιμοποιείται για την δημιουργία, τροποποίηση ή διακείριση μιας αποτελεσματικής διαδικασίας. Είναι ένα πρόγραμμα ή ένα εύειδη.
- Επινοήθηκε ένα end to end μοντέλο που πραγματοποιεί ένα NLP task. Είναι η περιεχόμενη text's. Οπλούση χρησιμοποιεί όλα τα steps για να πάψει από ενα κείμενο ή μια πρόβλημα.
- Η pipeline function χρησιμοποιείται για να κωδεικνύει το μοντέλο με τα απαραίτητα preprocessing και postprocessing steps.

## Example Sentiment-analysis

- Κατευρητοίσι τα πρόσωπα αν είναι χακί ή αρνακί.
- Είναι ορισμένοι ή pipeline οι οποίοι είναι pretrained μοντέλο που έχει γίνει finetuned για sentiment analysis.
- Όταν υπάρχει την εντολή την δεκτεύεις θα είναι object προσωρινά ωστε αν τανατελεστεί να χρησιμοποιήσει το ίδιο μοντέλο

Όταν ο pipeline function πάρει το κείμενο ακολούθων 3 steps:

Tokenization Το κείμενο προετείρεγχεται σε ένα format που το μοντέλο μπορει να καταλάβει.

- Το προετείρεγχεμένο κείμενο θεράπευτει στο μοντέλο, και αναταραγει predictions βασισμένη σε πορά μέτρους από τις οποίες έχει εκπαιδευτή.
- Οι προβλέψεις μετατείρεγχούνται για να γίνουν

Number 1 to positive Μια ερμηνεύσιμη. Για παραδειγμα μετατρέπονται τα αριθμητικά προβλέψεις σε κατηγορίες ή ετικέτες αναγνώσιμες από τον άνθρωπο.

## EIN Pipeline

- \* feature - extraction (get the vector representation of a text)
- \* fill mask
- \* named entity recognition (ner)
- \* question - answering
- \* sentiment - analysis
- \* summarization
- \* text - generation
- \* translation
- \* zero shot classification

### Zero shot classification

Tafinopoiēti xeiμena που δεν exou katηgoronoiθei arófa. De auti tnv bouaptnan, metopeis na uathoribes monos gos tis etiuetes kai na sun xrobiuronoiθei tis nōn umārxwn etiues μe tis omoiēs exei pretrained to montēlo.

Autn o pipeline ovpaiaste zero-shot xiat i δen xpeiafetai na uaneis finetune to montēlo sta data gos. Θa etieterpue πobosca supibatorita gla kaiθe etieta.

### Text generation

- De autoc eo eisos pipeline parēxei éva prompt kai to montēlo θa to bupilopriθei autōfata
- Autō to montēlo exei to stoixeo tns tuxouiotidē kai giauto eiou polū fysio na sun paipvei bupexia to idio apotēlegma
- Metopeis na exenfai πobes πroσabees θa anaparafse (num\_return\_sequences) kai bupolikā to mēxeos tou output text (argument max\_length)

### API (Application Programming Interface)

- Éva bupodo xanōn πou epitgrēton 6e ma εfapmoxi na etiuiouinavei μe álles εfapmoxes n unipesies.
- Epitgrēton stous πrogrampatistis na alloulepiðraun μe tnv lētouρgiuotita mas εfapmoxis n evōi igiotomou.
- Babuin iðea na alloulepiðraun diaforetika bupolikata metafu tōs
- Oi πrogrampatistis xrobiuronoi API kia na evaprativouν lētouρgiuotita εfapmoxwn n unipesiun bci8 tis tōs εfapmoxes, kanontai tis tis euēlēnter dikavēi na alloulepiðraun μe álla bupolikata.

### Mask filling

- Αυτή η υπηχορία παίρνει σαν input κείμενα με κενά (mask), όπου επιβρέχει κείμενο στο οποίο έχει κάλυψε τα κενά.
- Τοπ. κ είναι η εντολή/η παραπέρασης στην οποία επιλέγεται πόσες πιθανές απαντήσεις θέλεις να δου επιβρέχει.
- Κάθε mask filling models έχουν θιάσαρτο τρόπο που αναφορίζουν το κένο (mask token) στο input. Οπότε πάντα στην περιχρήση του μοντέλου θα βραβεύει αυτό που χρειαζόμαστε.

### Named entity recognition

- Σε αυτή την υπηχορία το μοντέλο παίρνει ένα text σαν input και προβλαθεί να ενταξει κάτιες οντότητα (λέξη) σαν ~~κάτια~~ person, location, organization.
- Οταν ενταξει μια οντότητα αλλά δεν μπορεί να την ενταξει για αυτό τη παρατάνω ραμχορία, την ενταξει στην υπηχορία MISC (Miscellaneous).
- Σε περιπτώση που δεν ενταξει οντότητα θύλε κένο ([ ]).

### Question Answering!

- Αυτή η υπηχορία απαντάει σε ερωτήσεις που του υπέρβαιν, ανάλογα με το κείμενο που του παρέχεται.
- Διαμορφώνει σαν κυριώσεις εδώ δια ότι δεν αναπαράγει το βιβλίο πόνο του την απάντηση, αλλοι απαντάει, ελάχιστας πληροφορία από το κείμενο που του έχει τίθεται σαν input στο νέοιο answer.

**Summarization**

- Το αυτό το task το προτέλο μαύρει ένα μεγάλο κείμενο  
και γράφει μια σημπλήρωση κρατώντας τις ευραυλιες  
πτυχές του κειμένου.

**Translation**

Το προτέλο αυτό χρησιμοποιείται για να περιγράψει  
οι άλλες γλώσσες το κείμενο που το παρέχει ως  
input.

How do Transformers work?

(June)

To 2018 εργαζήθηκε το πρώτο pretrained Transformer model, το GPT.

Χρησιμοποιήθηκε για αυχενόμενα έργα και NLP task, αφού είχε προβεί σε διαδικασία finetuning.

To 2018 October εργαζήθηκε στην άλλη μέρα pretrained μοντέλο, το Bert.

Διεύλαστηκε να γίνει summaries καλύτερα τις προτάσεις

To 2019 February, εισήχθη εξελίξη του GPT σε GPT2

To 2019 October, εκδόθηκε μια μικρότερη version του BERT, το DistilBERT που είναι 60% χρησιμότερο, 40% ελαφρύτερο και έχει 97% ιδία απόδοση με το BERT.

Έγινε το 2019 October λύση και το BART στις οποίες είναι 2 μεράκια pretrained μοντέλα που έχουν την ίδια αρχιτεκτονική όμως τα αρχικά transformer models.

To 2020 May εργαζήθηκε μια ακόμα μεγαλύτερη εύροση αυτο το GPT2, το GPT3 που αποδίδει καλύτερα σε κάποια task χωρίς να έχει γίνει finetuning.  
(όνομά της zero-shot learning)

Ta transformer model's χωρίσιαν σε 3 κατηγορίες:

1. Auto-regressive Transformer models (GPT-like)
2. Auto-encoding Transformer models (BERT-like)
3. Sequence-to-sequence Transformer models (BART/T5)

Transformers are language models

- Τα transformers models (GPT, BERT, BART, T5 etc) είναι εκπαιδευτές για language models. Ουλαϊκή είναι εκπαιδευτές σε μεγάλο αριθμό κειμένων κατηγοριας self-supervised learning.

(Self-Supervised learning)

Ο στόχος υπολογίζεται αυτόματα από τα input που δίνονται στο μοντέλο. Αυτό ονομαίνεται δεν υαφορίζει κάποιος τις labels των data.)

- Τα μοντέλα είναι εκπαιδευτές για να ματανούν την γλώσσα στην οποία έχουν εκπαιδευτεί. Αντί δεν είναι πολύ χρήσιμα για εργασιαρικά tasks.
- Είναι από τα language pretrained models περνώντας από μια σιασικασία, που υπολογίζεται transfer learning.
- Σε αυτή την σιασικασία το μοντέλο finetuned με ένα supervised τρόπο και χρησιμοποιεί επιπλέον που έχουν δοθεί από άνθρωπο για ένα δικαιοειρημένο task.

① Example:

Προβλέψει την επόμενη λέξη σε μια πρόταση, έχοντας σιαστέες τις προηγούμενες n-λέξεις

Αυτό υπολογίζεται causal language modeling είτε στο output εξαρτάται από το παρελθόντα + παρόν input, αλλά όχι από το μελλοντικό.

② Example

Masked language model, που προβλέπει μασκαρεμένες λέξεις

## Transformers are big models

- Η ελαφρεύτηκαση περιπτώσεων (DistilBERT), για να αυξήσουμε την απόδοση των μοντέλων, πρέπει αυξήσουμε το μέγεθος του μοντέλου καθώς και τα δεδομένα στα οποία έχει pretrained, το μοντέλο.
- Το να γίνει άμεσα αυτό xρειάζεται μεγάλο αριθμό data, καθώς το οποίο είναι χρεωκόπιο δε θέρευ xρόνου και computer resources

## Transfer learning

### Notes from video

- Η pretrained model used should be as similar as possible to the task it's finetuned on
- Pretrained models transfer any bias it may contain <sup>προνοτόληγμα</sup>

## Pretraining

- Η σιασινασία που ευπαράγει ένα μοντέλο από την αρχή. Αυτό σημαίνει ότι τα weights αρχικοποιούνται τυχαία και ευπαράγεται λειτουργία χωρίς προηγούμενη γνώση.
- Η ευπαράγεται γίνεται δε μεγάλο αριθμό data, και η σιασινασία αυτή σιαρεύ μόλις πινεν

## Finetuning

- Η σιασικοσία training αφού το μοντέλο έχει non-εκπαίδευτες
- Πρώτα χρειάζεται ένα non-εκπαίδευτο μοντέλο, που να μοιάζει με το μοντέλο που θέλουμε να φτιάξουμε
- Αφού το βρούμε θα εκπαιδεύσουμε επιπρόσθετα το μοντέλο, με το dataset για το ευχειρίσμένο task.

Why not simply train directly for the final task?

- To pretrained model έχει non επαρκείται σε dataset που νοιάζει με το finetuning dataset.
- Η finetuning process ευπεισέρχεται στη γνώση που αποτίθεται από το αρχικό μοντέλο, κατά τη διάρκεια του pretrained.
- Άλλου το pretrained μοντέλο έχει non επαρκείται σε πολλά data, το finetuning χρειάζεται λιγότερα data για να γίνει αποτέλεσμα.
- Τια τους iδιαίς λόγους για τα data, γίνεται και με το χρόνο και τα resources. Επομένως θα εξουσιεύσει αποτελέσματα.
- Επομένως το finetuning ενός μοντέλου χρειάζεται λιγότερο χρόνο, data, χρήματα και περιβαντολογικό δόγμα.
- Είναι επίγειο το εύκολο και το χρήσιμο να επαναλαμβάνει σε διαφορετικά σκηνή μεταβολή finetuning.
- Τέλος η διαδικασία αυτή θα ελαγχίζεται καλύτερα αποτελέσματα.

### General Architecture

#### Video Notes:

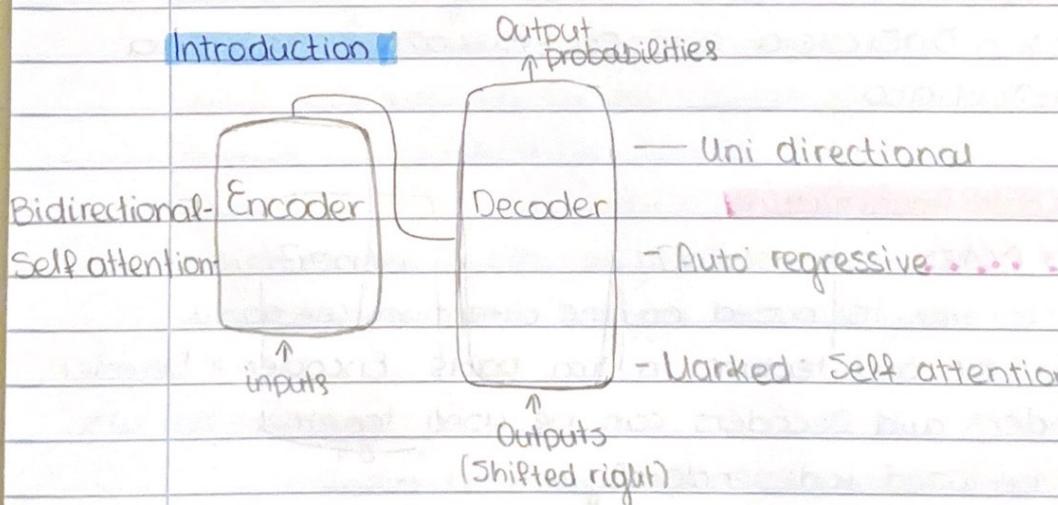
- Transformer is based on the attention mechanism
- Split the architecture in two parts Encoder & Decoder
- Encoders and Decoders can be used together but also can be used independently.
- Encoders accept inputs that represent text, and convert this text/words into numerical representations.

Video  
about  
Encoders

- This numerical representations called embeddings or features
- Self attention mechanism used as main component
- The decoder is similar to the encoder, it can also accept

**Video**: Same input as the encoder (text). It uses similar mechanism as the encoder, which is the masked Decoder's self attention as well.

- Differs from the encoder due to its uni-directional property, and is traditionally used autoregressive manner
- Combining two parts (Encoder + Decoder) results, is known as an encoder-decoder or sequence to sequence transformer
- The encoder accepts inputs and computes a high level representation of those inputs. The outputs are then passed to the decoder. The decoder uses the encoder's output alongside other inputs, in order to generate a prediction. It then predicts an output which it will re-use in future iterations, hence the term "auto regressive"



**Encoder**: Το encoder απόβαει inputs και θυάει features, διατηρώντας την κοράρωση στην παραγωγή. Η ανθεκτικότητα που είναι χρήσιμη για την ευαγγελία εργασία.

**Decoder**: Το decoder απόβαει τα features στην εργασία

Τα encoder ήταν ευνερχασία με σιναλ input, δημιουργεί μια αναλογία στόχου και output.

(π.χ πέρνει την αναπαράσταση του υφέντου (feature) και επιβιβρέφει καν output το μεταφρασμένο κείμενο)

Τα encoders ή τα decoders μπορούν να χρησιμοποιήσουν μερονομένα το ωδείο ή τη χρηστικότητα ή και τα δύο.

### Encoders-only models

Χρήσιμα για tasks που χρειάζεται να επαναποντεί το input

Οπως στην φωνή μας πρότασης για κατηγορία.

Η η κατηγοριοποίηση μας λέψης για πρότασης για κατηγορία named entity recognition

### Decoders-only models

Χρήσιμα για tasks παραχωρήσια, όπως η αναπαραχώρηση κειμένων, ή για την εγγένεια (text generation)

### Encoders-Decoders Models / Sequence to sequence models

Χρήσιμο για παραχωρήσια tasks, τα οποία χρειάζονται ένα input όπως translation or summarization

### Attention layers

◦ Τα transformer models χρειάζονται με special layers που αναδιοργανώνται attention layers.

◦ Αυτά τα layer λένε ότι προτέλο να δίνουν σημασία για συγκεκριμένες λέξεις για προσέτες που σημαίνει καν input για encoders για αναπαράσταση.

◦ Ανάλογα το task έχουμε διαφορετικές λέξεις γιας οποιες θέμειν και δίνουμε βαρύτητα.

- Μια λέξη είχε ένα υπόριθμο, αλλά το υπόριθμο επικεκλεγμένο ανάλογα το μέλισσο της λέξης. Το οποίο μέλισσο μπορεί να είναι ανοιασμένο διαν πλήρως μετα.

### Original Architecture for Transformers

- Η αρχιτεκτονική των transformer μοντέλων αρχικά δημιουργήθηκε για την μετάφραση
- Κατά την εκπαίδευση, το **encoder** λαμβάνει την ιδιαίτερη πρόταση στην επιευκότητή της χωρίς γνώση.
- Το **decoder** λαμβάνει την ίδια πρόταση στην επιευκότητή της χωρίς γνώση.
- Ωτα **encoder**, τα **attention layers** μπορεί να χρησιμοποιούν όλες τις λέξεις σε μία πρόταση.
- Το **decoder** αυτό την διάν πίνει βαρύτητα στις λέξεις που έχουν αναπαραχθεί για το target task καθώς την πρόταση που έχει μάρκερ **σαν input (feature)**
- Το να χρησιμοποιεί ο **decoder** διάν την πρόταση χίνεται για να επιταχύνει την διαδικασία. Αλλά έχει πάνω την πρόταση αυτή **σαν input** σεν μπορεί να χρησιμοποιήσει τις μελλοντικές λέξεις αυτού στην πρόταση.
- Ως ένα **decoder** το πρώτο **attention layer** σίνε προσοχή σε όλα τα inputs που υπήρχαν.
- Το δεύτερο **attention layer** χρησιμοποιεί το feature/output αυτό το **encoder**. Εάν θα έχει προσθέσει σε ανθίσματα την πρόταση ή θα μπορεί να προβλέψει καλύτερα την τρέχουσα λέξη.
- Επίσημ το **attention layer** χρησιμοποιείται από **encoder** & **decoder** για να προβέλει το μόνιμο κάτιμο των λεπτέρων λέξεις.

## Architectures vs checkpoints

**Architecture:** Ο γεγετός των ποντών, ο αριθμός κατεβαστικών layer και η διάσταση του γίνεται μέσα στο ποντέλο.

**Checkpoints:** Είναι τα weights που χρησιμοποιούνται μεταξύ δεσμών από την αρχιτεκτονική.

**Model:** Είναι όποια αρχιτεκτονική. Να είναι τόσο αριθμός ή αριθμός layer, το checkpoint. Μάλιστα μπορεί να είναι και τα δύο.

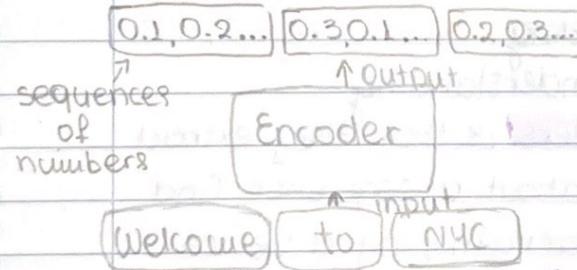
example

BERT ~ Architecture	} και τα δύο μπορεί να αναφέρεται σε αυτούς ποντέλο.
bert-base-cased ~ checkpoint	

## Encoder Models

Notes from the video

- BERT is example of **encoder**-only architecture, the most popular model of its kind.



- A text used as an input, and pass through the **encoder**.
- After the **encoder** process you receive a numerical representation of each word.
- This representation can be called a "feature vector" or "feature tensor".

- We have one vector per word, each vector is numerical representation of the word in question.

- The dimension of the vector is defined by the architecture of the model.

- The representation contain the value of the word, **disambiguated** and **contextualized**.

for  
BERT  
is 768

BERT is a standalone encoder model

• Each vector isn't only the representation of the word. But includes also the words around of the word that represented (weights). So the vector includes the representation of the word and the context around it (the weights).

• The mechanism that helps the vector to contain the "meaning" of the word in the text is self-attention mechanism.

• Self-attention mechanism allows a model to weight the importance of different positions within a sequence/text when computing a representation for each position. This mechanism helps the model to capture complex relationships and dependencies in the data.

• When should use an encoder?

1. Bi-directional : context from the left and right
2. Good extracting meaningful information
3. Sequence classification, Sentiment Analysis
4. Question - Answering,
5. Masked language modeling
6. Natural language Understanding

• The main idea of encoders is that they extract meaningful information about a sequence. And this informations are carrying by a vector

• The vectors can be handled by additional layers of neurons to make sense of them.

- Τα encoder models χρησιμοποιούν πάνω τα encoder αυτό τα transformer models
- Σε ωθεί στρώσιο , τα attention layers μεταφέρουν τα έχουν πρόβλεψε σε ωθεί γέλη στην πρόβλεψη.
- Αυτά τα ποντίκια χαρακτηρίζονται ότι έχουν

αρχισπολίν

"bi-directional" προβοκή ή ευχά ονομάζονται auto-encoding models.

- Το pretraining των μοντέλων αυτών γίνεται συνήθως με την αλοιώση μιας δεδομένης πιστασίας ή την απόβοληση μια μοντέλο να βρει ή να ανακατασκευάσει την αρχική πρόσβαση
- Τα Encoder μοντέλα είναι χρήσιμα για tasks που χρειάζονται να ματανοίσεις ή/και την πρόσβαση
- Examples of encoding transformer models

1. ALBERT

2. BERT

3. DistilBERT

4. ELECTRA

5. RoBERTa

## Decoder Models / Notes from Video

- GPT2 is an example of a popular decoder only architecture
- One can use a decoder for most of the same tasks as an encoder with a little loss of performance
  - We pass through the decoder, a numerical representation of each word
  - The decoder outputs exactly one sequence of numbers per input word.
- The part that decoder is different from the encoder is the self-attention mechanism. This mechanism it's using what is called "masked self-attention"
- In decoders inside a vector, all the words on the right of the word is masked (in other words the right context is masked.)
- Decoders have only access to the words on their left
- Masked Self-Attention differs from the self-attention mechanism by using an additional mask to hide the context on the right side.
- The words numerical representation will not be affected by the words in the hidden context.
- When should use a decoder?
  1. Unidirectional : Access to their left (or right) context
  2. The ability to generate a word or a sequence of words given a known sequence of word. (Generate Sequence or Natural Language Generation)
- Lets see an example how the decoder works.
  - We use as an input word "My" as a vector with information about the sequence.
  - We apply then a small transformation to that

decoder  
process  
of

vector so that it maps to all the words known by the model.

- We identify that the model believes the most probable following word is "name".
- Then we take this word and add it to the initial sequence. From "My", we are now at "My Name".  
(Autoregressive models re-use their past outputs as inputs in the following steps.)
- Again we follow the same process in order to find the word after "name".
- We pass the sequence through the decoder and retrieve the most probably following word. In this case we have "is".
- We repeat the operation until we're satisfied.

- Τα decoder models χρησιμοποιούν νέον το decoder auto to transformer μοντέλο.
- Σε κάθε στάδιο, για μία δωθεντική γέλη, κατεβαίνει attention layers μπορεί να έχει πρόσβαση σες γέλες που είναι τοποθετημένες πριν αυτό αυτήν γέλην πρόσβαση. Τα μοντέλα αυτά ονομάζονται auto-regressive models.
- Η σιαράγεια του pretraining για το decoder models έχει να κάνει με την πρόβλεψη της επόμενης γέλης σε μία πρόταση.
- Αυτά τα μοντέλα ταυτίζονται με tasks που έχουν να κάνουν με text generation.

### ◦ Decoder Μοντέλα

- CTRL
- GPT
- GPT2
- Transformer XL

## Encoder-decoder models / sequence to sequence models

### Notes from Video

- An example of a popular encoder-decoder model is T5
- The encoder takes word as inputs, casts them through the encoder and retrieves a numerical representation from each word cast through it.
- Numerical representation holds information about the meaning of the sequence
- Then we pass the outputs of the encoder directly to the decoder.
- With the output of the encoder we give also a sequence to the decoder.
- The decoder when prompting for an output with no initial sequence, we can give the value that indicates the start of a sequence.
- The encoder accepts a sequence as input, computes a prediction and outputs a numerical representation.
- Then sends this over to the decoder, and the decoder using this input alongside its usual sequence input, will take a stab at decoding the sequence.
- The decoder decodes the sequence, and outputs a word.
- Now that we have both the feature vector and an initial generated word, we don't need the encoder anymore.
- The decoder acts in an autoregressive manner, the word it has just output can now be used as an input.
- This and numerical representation output by the encoder, can now be used to generate a second word.
- Note that the first word is still here, as the model still outputs it.
- We can continue on and on until decoder outputs

a value that we consider a "stopping value" like a dot.

- Example with Translation Language Modeling (Transduction)
- We are gonna translate a sequence.

Translate "Welcome to NYC" in French.

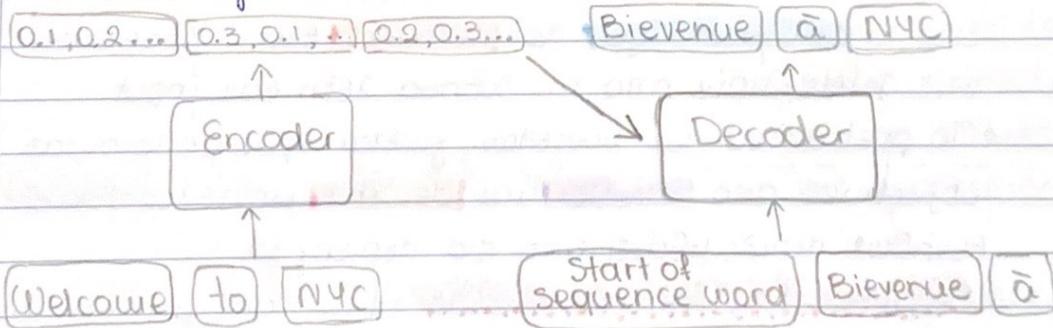
1. The **encoder** create a representation of the English sentence.

2. We cast this to the **decoder** and with the use of the start of sequence word, we ask it to output the first word.

3. The output is "Bienvenue", then we use this as an input sequence for the **decoder**. Alongside the feature vector.

4. The below info allows the **decoder** to predict the second word "à"

5. Finally the **decoder** predict the final word by repeating the process.



◦ Encoder and decoder often don't share weights.

◦ Therefore we have an entire block that can be trained to understand the sequence and extract the relevant information

◦ On the other hand, **decoder**, whose sole purpose is to **decode** the feature output by the **encoder**. Encoder can be specialized in a completely different language or modality like images or speech.

• **Encoders** - **Decoders** are special for:

1. Sequence to Sequence task (translation, summarization)
2. Weights that are not necessarily shared across the **encoder** and **decoder**
3. Input distribution different from output distribution

• You can load together an **encoder+decoder**, or to select the **encoder** and **decoder** suite perfectly for your task and then load them separately, in order to work with them.

- Τα **Encoder + Decoder** ποιεία χρησιμολογούν και τα μέρη από τη Transformer architecture.
- Σε κάθε στρεμμα, το **attention layer** του **encoder** έχει πρόσβαση σε όλες τις γέλες σε μια initial sentence ενώ τα **attention layers** των **decoders** έχουν πρόσβαση στις γέλες πριν από την δοθεία γέλην σαν input.
- Το pretrained των ποντέων, γίνεται χρησιμολογώντας objectives από **encoder** και **decoder** ποντέα. Απόδινος περιλαμβάνει και το περιπλόκο.
- **Sequence to Sequence** models είναι χρήσιμα για tasks που έχουν να γίνουν αντισταθμιστές νέων προτάσεων με βάση το input που του δοθεί. Για summarization, translation, or generative question answering.
- **Sequence to Sequence** ποιεία

- BART

- uBART

- Marian

- T5

### Bias and limitations

- Τοπίες δορές χρειάζεται να χρησιμοποιήσεις ένα pretrained μοντέλο ή ένα finetuning version, πρέπει να έχουμε στο μυαλό μας ότι όλα αυτά τα μοντέλα για να εκπαιδεύουν ηγένουν χρειάζονται αριθμό data. Τοπίες δορές αυτά τα data μπορεί να ποιοτικά μπορεί και όχι. Καθώς τα πάντα είναι διαθέσιμα στο internet.
- Ομότε πρέπει να έχεις στο μυαλό σου, ότι τα αρχικά μοντέλα που χρησιμοποιείς, μπορεί να θεριέχουν σελυγκά, πατιβετά & ομοφοβικά στοιχεία.
- Το να κάνεις finetuning ένα μοντέλο σε ένα ελαφρώς αυτά τα θεριζόντα.

### Summarization Table

Model	Examples	Tasks
Encoder	ALBERT, BERT, DistilBERT, ELECTRA, ROBERTA	Sentence classification named entity recognition, extractive question answering
Decoder	CTRL, GPT, GPT2, Transformer XL	Text generation

Encoder	BART, T5, Marian, mBART	Summarization, translation, generative question answering
Decoder		