**Εργασία 3 – Επισημείωση Ακολουθιών**

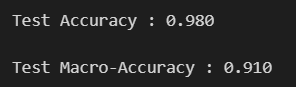
**Επεξεργασία Φυσικής Γλώσσας**

**Τζανής Νικόλαος**

**mtn2217**

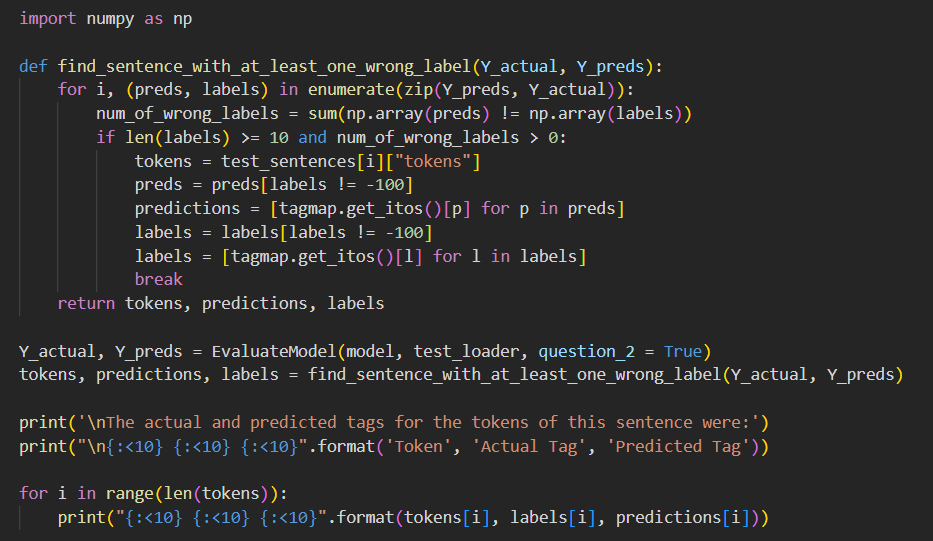
**Ερώτημα 1**

Από την εκτέλεση του κώδικα που δίνεται μάς επιστρέφονται οι εξής μετρήσεις για το accuracy και το macro-accuracy στο test dataset:

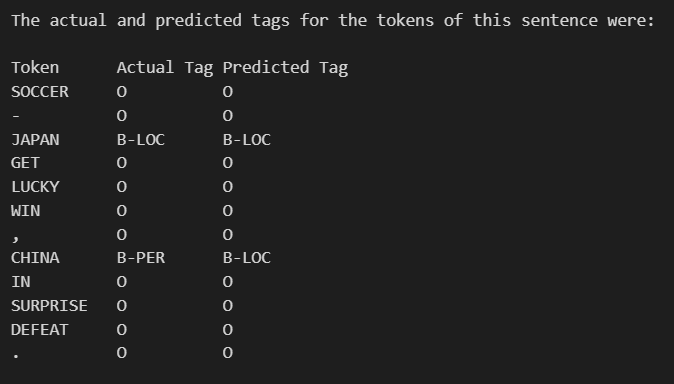


**Ερώτημα 2**

Με αυτό το κομμάτι κώδικα επιστρέφεται μία τυχαία πρόταση μεγέθους πάνω από 10 tokens για την οποία το μοντέλο μας δεν βρήκε τα σωστά tags. Ακολούθως εκτυπώνεται η πρόταση, ενώ στον πίνακα βλέπουμε το πραγματικό tag (actual tag) καθώς και το tag που προέβλεψε το μοντέλο μας:

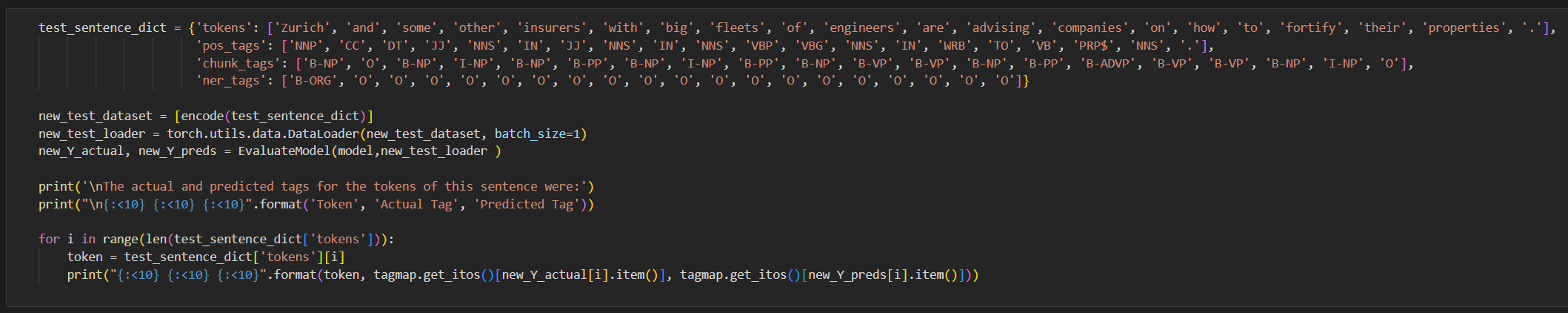


Το αποτέλεσμα της εκτέλεσης του παραπάνω κώδικα:

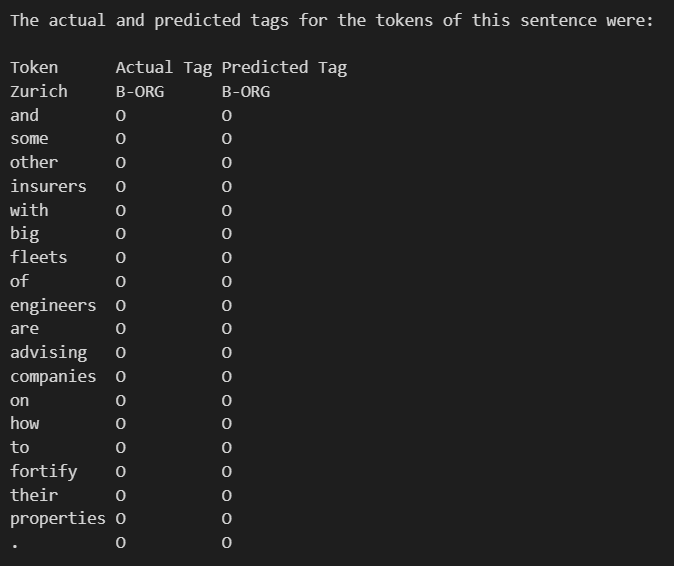


Όπως βλέπουμε, στη λέξη ‘CHINA’ θα έπρεπε να έχει δοθεί το B-PER tag, αλλά το μοντέλο έδωσε B-LOC. Για όλα τα υπόλοιπα tokens αυτής της πρότασης, το prediction του μοντέλου ήταν σωστό.

Ακολούθως επιλέγουμε μία δική μας πρόταση και επαναλαμβάνουμε την παραπάνω διαδικασία. Η πρόταση που έχω επιλέξει προέρχεται από το Wall Street Journal:



Η πρόταση βρίσκεται ήδη στη μορφή dictionary που χρειάζεται για να δώσει predictions το μοντέλο. Το αποτέλεσμα της εκτέλεσης του παραπάνω κελιού είναι:



Άρα όπως βλέπουμε, το μοντέλο δεν έχει κάνει λάθος στο prediction των tags.

**Ερώτημα 3**

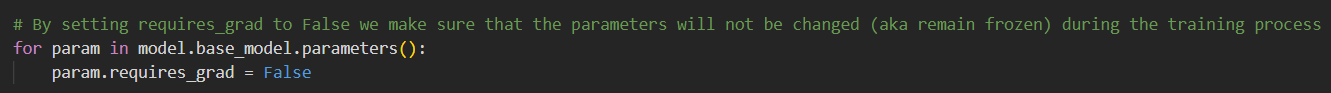
Η μέθοδος align\_label παίρνει ως είσοδο τα tokens και τα labels και εξετάζει αν για κάποιο token, το word id που του αντιστοιχεί (το ελέγχει μέσω της μεθόδου word\_ids()) είναι None. Αν ισχύει αυτό τότε αποδίδει στο token αυτό το label -100 το οποίο για την PyTorch σημαίνει ότι θα πρέπει να αγνοήσει αυτό το token στις περαιτέρω αναλύσεις.

Καθώς ο BERT tokenizer μπορεί να σπάσει ένα token σε μικρότερα subtokens, θέλουμε μόνο το 1ο subtoken να λαμβάνει κανονικό label και όλα τα subtokens του ίδιου token να παίρνουν -100. Αυτό γίνεται στο elif κομμάτι της μεθόδου.

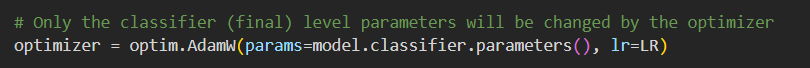
Κατά συνέπεια, τελικά η μέθοδος EvaluateModel θα αγνοήσει όλα τα tokens με label -100 και επομένως θα εκτιμήσει το accuracy του μοντέλου μόνο με τα υπόλοιπα.

**Ερώτημα 4**

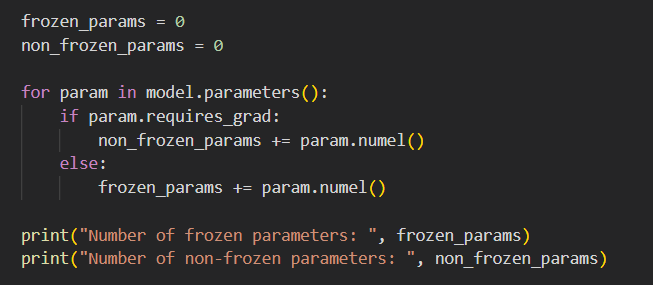
Προκειμένου να «παγώσουμε» τα βάρη του προεκπαιδευμένου μοντέλου BERT, πρέπει να προσθέσουμε τις ακόλουθες γραμμές κώδικα:



Ωστόσο, στο τελευταίο επίπεδο (classifier) θέλουμε να γίνουν αλλαγές επομένως θα προσαρμόσουμε κατάλληλα τον optimizer:



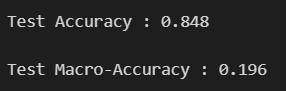
Με το ακόλουθο κομμάτι κώδικα βλέπουμε τελικά πόσα parameters έγιναν freeze και πόσα άλλαξαν:



Το αποτέλεσμα:



Η εκπαίδευση του μοντέλου με αυτή την τροποποίηση περιορίστηκε σημαντικά, ιδιαίτερα στο macro-accuracy, όπως φαίνεται από τα αποτελέσματα της εκπαίδευσης:



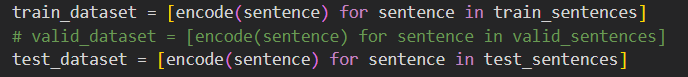
**Ερώτημα 5**

Για αυτό το ερώτημα ενώνουμε το training set και το validation set και εκπαιδεύουμε εκ νέου το μοντέλο. Οι αλλαγές που θα πρέπει να γίνουν είναι οι εξής:

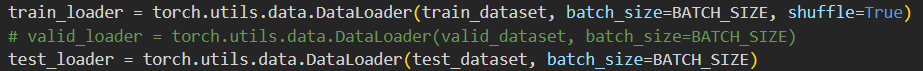
Αρχικά προσθέτουμε τα validation και τα test sentences στο κομμάτι του κώδικα που καλείται η μέθοδος load\_sentences:



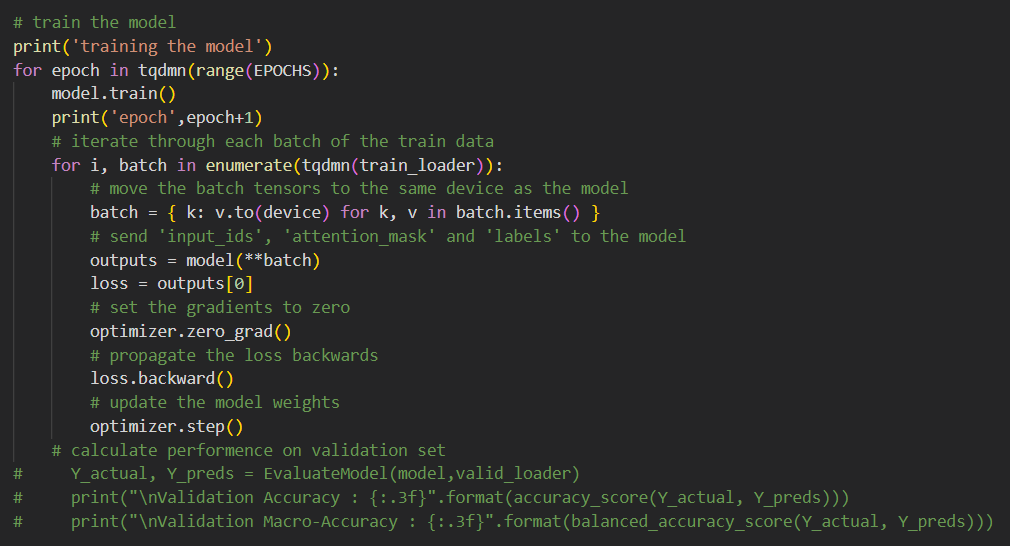
Το valid\_dataset δεν χρειάζεται πλέον οπότε κάνουμε comment τη σχετική γραμμή στον κώδικα:



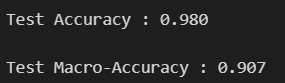
Το ίδιο ισχύει και για τον valid\_loader:



Τέλος, κατά το training του μοντέλου δεν υπάρχει πλέον validation στο τέλος κάθε εποχής, επομένως αφαιρούμε και αυτές τις γραμμές του κώδικα:



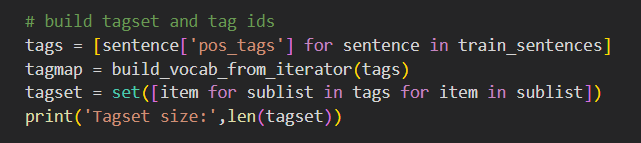
Με τις παραπάνω αλλαγές βλέπουμε ότι το accuracy και το macro-accuracy δεν διαφοροποιήθηκαν σημαντικά σε σχέση με το 1ο ερώτημα:



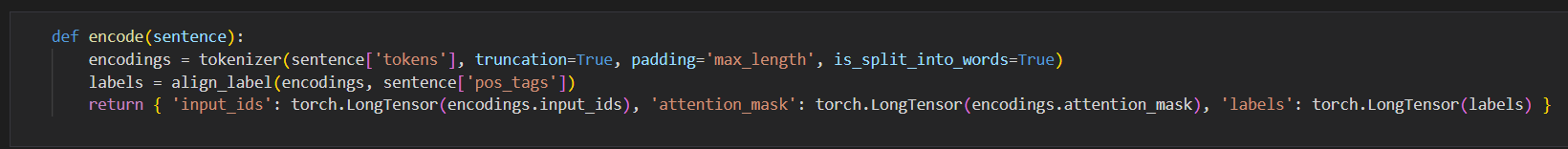
**Ερώτημα 6**

Οι αλλαγές που θα πρέπει να κάνουμε έτσι ώστε ο κώδικας να πραγματοποιεί επισημείωση μέρους-του-λόγου (POS tagging) είναι οι εξής:

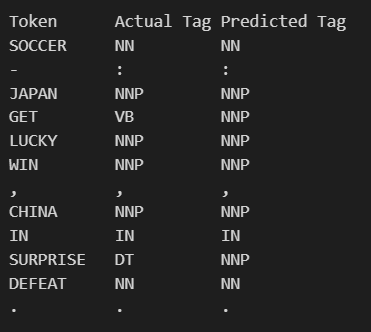
Εδώ θα πρέπει να αντικαταστήσουμε το ner\_tags με pos\_tags:



Το ίδιο και στη μέθοδο encode:

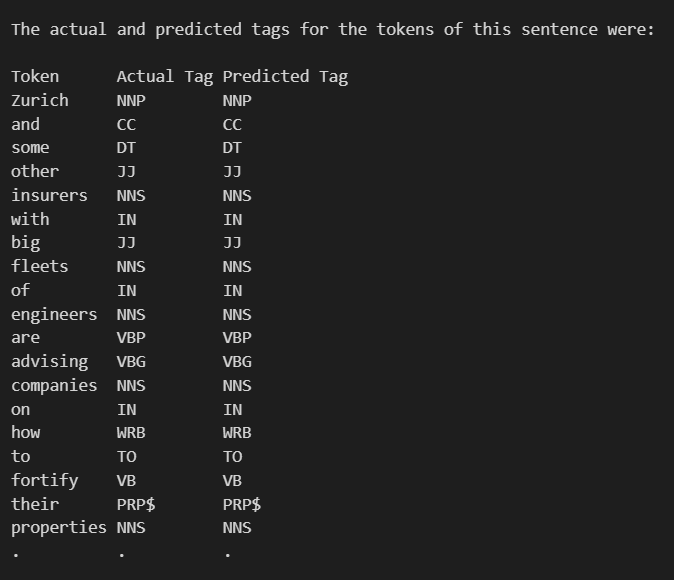


O υπόλοιπος κώδικας παραμένει ως έχει. Εδώ μία πρόταση που το μοντέλο δεν κατάφερε να προβλέψει σωστά τα pos tags ήταν η εξής:



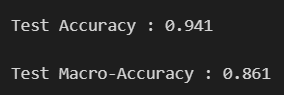
Όπως βλέπουμε στο token ‘GET’ το μοντέλο απέδωσε το tag ‘NNP’ ενώ στην πραγματικότητα θα έπρεπε να δώσει ‘VB’ και στο token ‘SURPRISE’ έδωσε ‘NNP’ενώ θα έπρεπε να έχει δώσει ‘DT’.

Στην πρόταση από το Wall Street Journal είχαμε τα εξής αποτελέσματα:



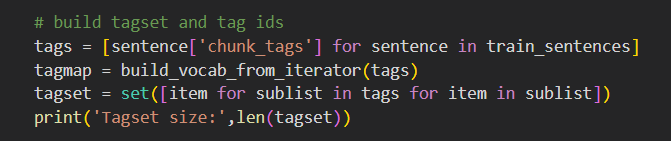
Εδώ φαίνεται ότι το model δεν έκανε κανένα λάθος και πρόβλεψε σωστά όλα τα tags.

Το accuracy και το macro-accuracy του μοντέλου στο POS tagging ήταν:

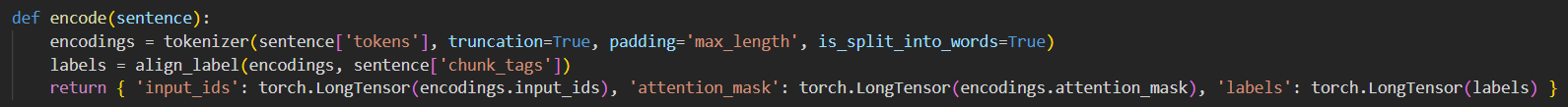


**Ερώτημα 7**

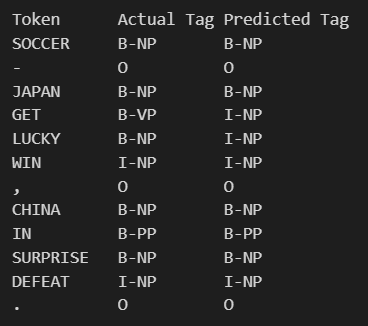
Οι αλλαγές που θα πρέπει να κάνουμε έτσι ώστε ο κώδικας να πραγματοποιεί ανάγνωση ορίων φράσεων (text chunking) είναι αντίστοιχες με αυτές του προηγούμενου ερωτήματος. Συγκεκριμένα:



και



Μία πρόταση στην οποία το μοντέλο δεν προέβλεψε σωστά τα tags ήταν η:



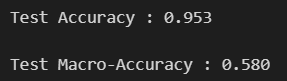
Όπως βλέπουμε στο token ‘GET’ έβαλε I-NP, ενώ θα έπρεπε να έχει βάλει B-VP.

Στην πρόταση από το Wall Street Journal υπήρχαν επίσης ορισμένα λάθη:



Συγκεκριμένα, στο token ‘insurers’ δόθηκε το tag ‘I-NP’ αντί για το ‘B-NP’, στο ‘advising’ δόθηκε ‘I-VP’ αντί για ‘B-VP’ και στο ‘fortify’ δόθηκε ‘B-VP’ αντί για ‘I-VP’.

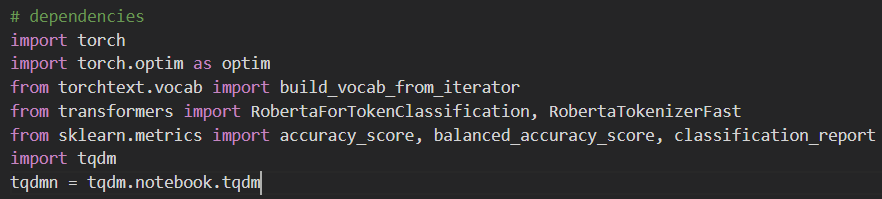
Το accuracy και το macro-accuracy του μοντέλου στο text chunking ήταν:



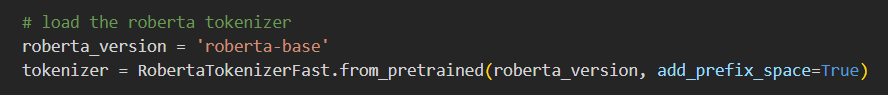
**Ερώτημα 8**

Οι αλλαγές που πρέπει να γίνουν στον κώδικα έτσι ώστε να χρησιμοποιείται το μοντέλο Roberta-base είναι οι εξής:

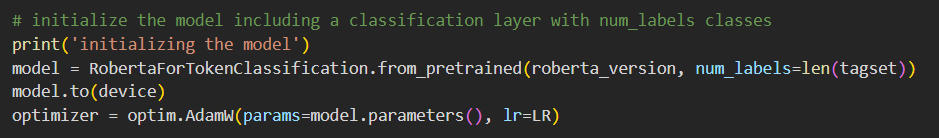
Αρχικά πρέπει να κάνουμε import το μοντέλο και τον tokenizer από τη βιβλιοθήκη transformers:



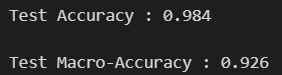
Στη συνέχεια πρέπει να φορτώσουμε τον tokenizer του Roberta:



Και τέλος να φορτώσουμε το μοντέλο:



Το μοντέλο αυτό φαίνεται ότι τελικά έχει μεγαλύτερο accuracy και macro-accuracy σε σχέση με το αρχικό μας μοντέλο με το BERT:



**Ερώτημα 9**

Με τα αποτελέσματα που λάβαμε από τα προηγούμενα ερωτήματα για τις δύο προτάσεις (την τυχαία πρόταση που είχε λάθος tags και αυτή που επιλέξαμε εμείς από το Wall Street Journal) θα εξετάσουμε ποια είναι η απάντηση του ChatGPT για αυτές τις προτάσεις και θα συγκρίνουμε τα αποτελέσματα.

Ξεκινώντας με την πρόταση «SOCCER - JAPAN GET LUCKY WIN, CHINA IN SURPRISE DEFEAT.» με zero-shot prompting έχουμε:

Prompt:

SOCCER - JAPAN GET LUCKY WIN, CHINA IN SURPRISE DEFEAT.

Perform Named Entity Recognition Tagging, POS tagging and text chunking on the tokens of the above sentence.

Απάντηση:

Sure! Let's perform Named Entity Recognition (NER) tagging, Part-of-Speech (POS) tagging, and text chunking on the tokens in the given sentence.

Tokens:

1. SOCCER

2. -

3. JAPAN

4. GET

5. LUCKY

6. WIN

7. ,

8. CHINA

9. IN

10. SURPRISE

11. DEFEAT

12. .

NER Tagging:

1. SOCCER -> O (Not a named entity)

2. - -> O (Not a named entity)

3. JAPAN -> B-LOC (Named entity - Location)

4. GET -> O (Not a named entity)

5. LUCKY -> O (Not a named entity)

6. WIN -> O (Not a named entity)

7. , -> O (Not a named entity)

8. CHINA -> B-LOC (Named entity - Location)

9. IN -> O (Not a named entity)

10. SURPRISE -> O (Not a named entity)

11. DEFEAT -> O (Not a named entity)

12. . -> O (Not a named entity)

POS Tagging:

1. SOCCER -> NN (Noun, singular or mass)

2. - -> HYPH (Punctuation mark, hyphen)

3. JAPAN -> NNP (Proper noun, singular)

4. GET -> VB (Verb, base form)

5. LUCKY -> JJ (Adjective)

6. WIN -> NN (Noun, singular or mass)

7. , -> , (Punctuation mark, comma)

8. CHINA -> NNP (Proper noun, singular)

9. IN -> IN (Preposition or subordinating conjunction)

10. SURPRISE -> NN (Noun, singular or mass)

11. DEFEAT -> NN (Noun, singular or mass)

12. . -> . (Punctuation mark, sentence closer)

Text Chunking:

1. SOCCER -> Noun Phrase

2. - -> Not a chunk

3. JAPAN -> Noun Phrase

4. GET LUCKY WIN -> Verb Phrase

5. , -> Not a chunk

6. CHINA -> Noun Phrase

7. IN -> Not a chunk

8. SURPRISE DEFEAT -> Noun Phrase

9. . -> Not a chunk

Please note that the chunking is subjective and may vary based on the specific chunking rules used.

Σχολιασμός των αποτελεσμάτων:

Για το NER tagging βλέπουμε ότι όπως και το δικό μας μοντέλο, το ChatGPT θεωρεί ότι στο token “CHINA” αντιστοιχεί το B-LOC, κάτι που είναι μάλλον σωστό καθώς η Κίνα είναι περιοχή. Άρα εδώ θα μπορούσαμε να πούμε ότι τόσο το μοντέλο μας όσο και το ChatGPT έχουν δίκιο και πιθανώς έχει λάθος το αρχικό tagging.

Για το POS tagging το ChatGPT βρήκε σωστά το tag για το token “GET”, ενώ το δικό μας μοντέλο είχε κάνει λάθος. Και τα δύο μοντέλα έκαναν λάθος για το “SURPRISE”, καθώς κανένα από τα δύο δεν έδωσε το σωστό tag “DT”.

Τέλος, στο text chunking η απάντηση του ChatGPT δεν ήταν αυτό που θα περιμέναμε και γι’ αυτό πιθανώς ενοχοποιείται το zero-shot prompting. Ωστόσο, το μοντέλο αναφέρει ότι το chunking είναι υποκειμενικό και μπορεί να διαφέρει ανάλογα με τους κανόνες που χρησιμοποιούμε κάθε φορά.

Για την πρόταση του Wall Street Journal επαναλαμβάνουμε την ίδια διαδικασία και λαμβάνουμε την εξής απάντηση από το ChatGPT:

Απάντηση:

Sure! Let's perform Named Entity Recognition (NER) tagging, Part-of-Speech (POS) tagging, and text chunking on the tokens in the given sentence.

Tokens:

1. Zurich

2. and

3. some

4. other

5. insurers

6. with

7. big

8. fleets

9. of

10. engineers

11. are

12. advising

13. companies

14. on

15. how

16. to

17. fortify

18. their

19. properties

20. .

NER Tagging:

1. Zurich -> B-ORG (Named entity - Organization)

2. and -> O (Not a named entity)

3. some -> O (Not a named entity)

4. other -> O (Not a named entity)

5. insurers -> O (Not a named entity)

6. with -> O (Not a named entity)

7. big -> O (Not a named entity)

8. fleets -> O (Not a named entity)

9. of -> O (Not a named entity)

10. engineers -> O (Not a named entity)

11. are -> O (Not a named entity)

12. advising -> O (Not a named entity)

13. companies -> O (Not a named entity)

14. on -> O (Not a named entity)

15. how -> O (Not a named entity)

16. to -> O (Not a named entity)

17. fortify -> O (Not a named entity)

18. their -> O (Not a named entity)

19. properties -> O (Not a named entity)

20. . -> O (Not a named entity)

POS Tagging:

1. Zurich -> NNP (Proper noun, singular)

2. and -> CC (Coordinating conjunction)

3. some -> DT (Determiner)

4. other -> JJ (Adjective)

5. insurers -> NNS (Noun, plural)

6. with -> IN (Preposition or subordinating conjunction)

7. big -> JJ (Adjective)

8. fleets -> NNS (Noun, plural)

9. of -> IN (Preposition or subordinating conjunction)

10. engineers -> NNS (Noun, plural)

11. are -> VBP (Verb, non-3rd person singular present)

12. advising -> VBG (Verb, gerund or present participle)

13. companies -> NNS (Noun, plural)

14. on -> IN (Preposition or subordinating conjunction)

15. how -> WRB (Wh-adverb)

16. to -> TO (To)

17. fortify -> VB (Verb, base form)

18. their -> PRP$ (Possessive pronoun)

19. properties -> NNS (Noun, plural)

20. . -> . (Punctuation mark, sentence closer)

Text Chunking:

1. Zurich -> Noun Phrase

2. and -> Not a chunk

3. some -> Not a chunk

4. other -> Not a chunk

5. insurers -> Noun Phrase

6. with big fleets of engineers -> Prepositional Phrase

7. are advising -> Verb Phrase

8. companies -> Noun Phrase

9. on -> Not a chunk

10. how to fortify their properties -> Noun Phrase

11. . -> Not a chunk

Again, please note that the chunking is subjective and may vary based on the specific chunking rules used.

Σχολιασμός των αποτελεσμάτων:

Εδώ βλέπουμε ότι η απάντηση του ChatGPT συμφωνεί με το μοντέλο μας για το NER tagging καθώς και τα δύο μοντέλα έχουν δώσει σωστά το tag “B-ORG” στο token “Zurich”.

Στο POS Tagging τόσο το ChatGPT όσο και το μοντέλο μας προέβλεψαν τέλεια όλα τα tags.

Στο text chunking και εδώ η απάντηση του ChatGPT δεν είναι αυτή που θα περιμέναμε και πιθανώς ευθύνεται και πάλι το zero-shot prompting.

**Ερώτημα 10**

Σε αυτό το ερώτημα δοκιμάζουμε το few-shot prompting για να δούμε αν βελτιώνονται οι απαντήσεις του ChatGPT σε σχέση με το zero-shot prompting. Ξεκινάμε αρχικά από την πρόταση «SOCCER - JAPAN GET LUCKY WIN, CHINA IN SURPRISE DEFEAT.» που είχε λάθος tags στο μοντέλο μας για όλα τα taggings.

Στο few-shot prompting θα δώσουμε στο ChatGPT τις πρώτες 5 προτάσεις από τα train\_sentences μαζί με τα tags του και θα ζητήσουμε να χρησιμοποιήσει την ίδια μέθοδο για να μας δώσει τα tags στην πρόταση που ζητάμε.

Prompt:

[{'tokens': ['EU',

'rejects',

'German',

'call',

'to',

'boycott',

'British',

'lamb',

'.'],

'pos\_tags': ['NNP', 'VBZ', 'JJ', 'NN', 'TO', 'VB', 'JJ', 'NN', '.'],

'chunk\_tags': ['B-NP',

'B-VP',

'B-NP',

'I-NP',

'B-VP',

'I-VP',

'B-NP',

'I-NP',

'O'],

'ner\_tags': ['B-ORG', 'O', 'B-MISC', 'O', 'O', 'O', 'B-MISC', 'O', 'O']},

{'tokens': ['Peter', 'Blackburn'],

'pos\_tags': ['NNP', 'NNP'],

'chunk\_tags': ['B-NP', 'I-NP'],

'ner\_tags': ['B-PER', 'I-PER']},

{'tokens': ['BRUSSELS', '1996-08-22'],

'pos\_tags': ['NNP', 'CD'],

'chunk\_tags': ['B-NP', 'I-NP'],

'ner\_tags': ['B-LOC', 'O']},

{'tokens': ['The',

'European',

'Commission',

'said',

'on',

'Thursday',

'it',

'disagreed',

'with',

'German',

'advice',

'to',

'consumers',

'to',

'shun',

'British',

'lamb',

'until',

'scientists',

'determine',

'whether',

'mad',

'cow',

'disease',

'can',

'be',

'transmitted',

'to',

'sheep',

'.'],

'pos\_tags': ['DT',

'NNP',

'NNP',

'VBD',

'IN',

'NNP',

'PRP',

'VBD',

'IN',

'JJ',

'NN',

'TO',

'NNS',

'TO',

'VB',

'JJ',

'NN',

'IN',

'NNS',

'VBP',

'IN',

'JJ',

'NN',

'NN',

'MD',

'VB',

'VBN',

'TO',

'NN',

'.'],

'chunk\_tags': ['B-NP',

'I-NP',

'I-NP',

'B-VP',

'B-PP',

'B-NP',

'B-NP',

'B-VP',

'B-PP',

'B-NP',

'I-NP',

'B-PP',

'B-NP',

'B-VP',

'I-VP',

'B-NP',

'I-NP',

'B-SBAR',

'B-NP',

'B-VP',

'B-SBAR',

'B-NP',

'I-NP',

'I-NP',

'B-VP',

'I-VP',

'I-VP',

'B-PP',

'B-NP',

'O'],

'ner\_tags': ['O',

'B-ORG',

'I-ORG',

'O',

'O',

'O',

'O',

'O',

'O',

'B-MISC',

'O',

'O',

'O',

'O',

'O',

'B-MISC',

'O',

'O',

'O',

'O',

'O',

'O',

'O',

'O',

'O',

'O',

'O',

'O',

'O',

'O']},

{'tokens': ['Germany',

"'s",

'representative',

'to',

'the',

'European',

'Union',

"'s",

'veterinary',

'committee',

'Werner',

'Zwingmann',

'said',

'on',

'Wednesday',

'consumers',

'should',

'buy',

'sheepmeat',

'from',

'countries',

'other',

'than',

'Britain',

'until',

'the',

'scientific',

'advice',

'was',

'clearer',

'.'],

'pos\_tags': ['NNP',

'POS',

'NN',

'TO',

'DT',

'NNP',

'NNP',

'POS',

'JJ',

'NN',

'NNP',

'NNP',

'VBD',

'IN',

'NNP',

'NNS',

'MD',

'VB',

'NN',

'IN',

'NNS',

'JJ',

'IN',

'NNP',

'IN',

'DT',

'JJ',

'NN',

'VBD',

'JJR',

'.'],

'chunk\_tags': ['B-NP',

'B-NP',

'I-NP',

'B-PP',

'B-NP',

'I-NP',

'I-NP',

'B-NP',

'I-NP',

'I-NP',

'I-NP',

'I-NP',

'B-VP',

'B-PP',

'B-NP',

'I-NP',

'B-VP',

'I-VP',

'B-NP',

'B-PP',

'B-NP',

'B-ADJP',

'B-PP',

'B-NP',

'B-SBAR',

'B-NP',

'I-NP',

'I-NP',

'B-VP',

'B-ADJP',

'O'],

'ner\_tags': ['B-LOC',

'O',

'O',

'O',

'O',

'B-ORG',

'I-ORG',

'O',

'O',

'O',

'B-PER',

'I-PER',

'O',

'O',

'O',

'O',

'O',

'O',

'O',

'O',

'O',

'O',

'O',

'B-LOC',

'O',

'O',

'O',

'O',

'O',

'O',

'O']}]

Using the provided data for the above 5 sentences as a guide, give me the pos\_tags, ner\_tags and chunk\_tags for the sentence:

SOCCER - JAPAN GET LUCKY WIN, CHINA IN SURPRISE DEFEAT.

Απάντηση:

Based on the provided data, here are the POS tags, NER tags, and chunk tags for the sentence "SOCCER - JAPAN GET LUCKY WIN, CHINA IN SURPRISE DEFEAT":

POS tags: ['NN', 'HYPH', 'NNP', 'NNP', 'VB', 'JJ', 'NN', 'NNP', 'IN', 'NNP', 'NNP', '.']

NER tags: ['O', 'O', 'B-LOC', 'O', 'O', 'O', 'O', 'B-LOC', 'O', 'B-LOC', 'O', 'O']

Chunk tags: ['B-NP', 'I-NP', 'B-NP', 'I-NP', 'B-VP', 'B-ADJP', 'I-ADJP', 'B-NP', 'B-PP', 'B-NP', 'I-NP', 'O']

Please note that the chunk tags may vary depending on the specific chunking scheme used.

Σχολιασμός των αποτελεσμάτων:

Για το NER tagging βλέπουμε ότι εδώ το ChatGPT έχει κάνει λάθος καθώς πρόσθεσε B-LOC στο token “SURPRISE” κάτι που δεν είχε κάνει με το zero-shot prompting. Τα υπόλοιπα tags είναι τα ίδια με το zero shot prompting. Το δικό μας μοντέλο είχε προβλέψει σωστά όλα τα tags εκτός από το “CHINA” στο οποίο είχε δώσει tag “B-LOC” αντί για “B-PER”.

Στο POS tagging η απάντηση του ChatGPT έχει επίσης περισσότερα λάθη αφού σε πολλαπλά tokens δεν έχει δώσει το σωστό tag. Άρα μέχρι τώρα θα λέγαμε ότι το few-shot prompting έκανε το μοντέλο του ChatGPT να μας δώσει χειρότερη απάντηση σε σχέση με το zero-shot prompting.

Τέλος, στο text chunking υπάρχει σαφής βελτίωση σε σχέση με το zero-shot prompting καθώς εκτός ελαχίστων εξαιρέσεων το ChatGPT κατάφερε να μας δώσει σωστά αποτελέσματα, τα οποία μάλιστα είναι πολύ κοντά σε αυτά που έδωσε το δικό μας μοντέλο.

Στη συνέχεια συνεχίζουμε με την πρόταση από το Wall Street Journal. Χρησιμοποιώντας το ίδιο prompt όπως και πριν (με αλλαγή μόνο την πρόταση του WSJ) παίρνουμε την εξής απάντηση από το ChatGPT.

Απάντηση:

Based on the provided data, here are the POS tags, NER tags, and chunk tags for the sentence "Zurich and some other insurers with big fleets of engineers are advising companies on how to fortify their properties":

POS tags: ['NNP', 'CC', 'DT', 'JJ', 'NNS', 'IN', 'JJ', 'NNS', 'VBP', 'VBG', 'NNS', 'IN', 'WRB', 'TO', 'VB', 'PRP$', 'NNS', '.']

NER tags: ['B-ORG', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O']

Chunk tags: ['B-NP', 'O', 'B-NP', 'I-NP', 'I-NP', 'B-PP', 'B-NP', 'I-NP', 'B-VP', 'I-VP', 'I-VP', 'B-PP', 'B-ADVP', 'B-TO', 'B-VP', 'B-NP', 'I-NP', 'O']

Please note that the NER tags indicate no named entities in this sentence, and the chunk tags may vary depending on the specific chunking scheme used.

Σχολιασμός των αποτελεσμάτων:

Εδώ βλέπουμε ότι η απάντηση του ChatGPT είναι και πάλι σωστή και συμφωνεί με το μοντέλο μας για το NER tagging, καθώς και τα δύο μοντέλα έχουν δώσει σωστά το tag “B-ORG” στο token “Zurich”.

Στο POS Tagging το few-shot prompting οδήγησε σε ορισμένα λάθη, κάτι που δεν είχε συμβεί στο zero-shot prompting. Το μοντέλο μας είχε προβλέψει σωστά όλα τα tags εδώ.

Στο text chunking υπάρχει σαφής βελτίωση σε σχέση με το zero-shot prompting καθώς το chat απέτυχε να προβλέψει σωστά ελάχιστα tokens.