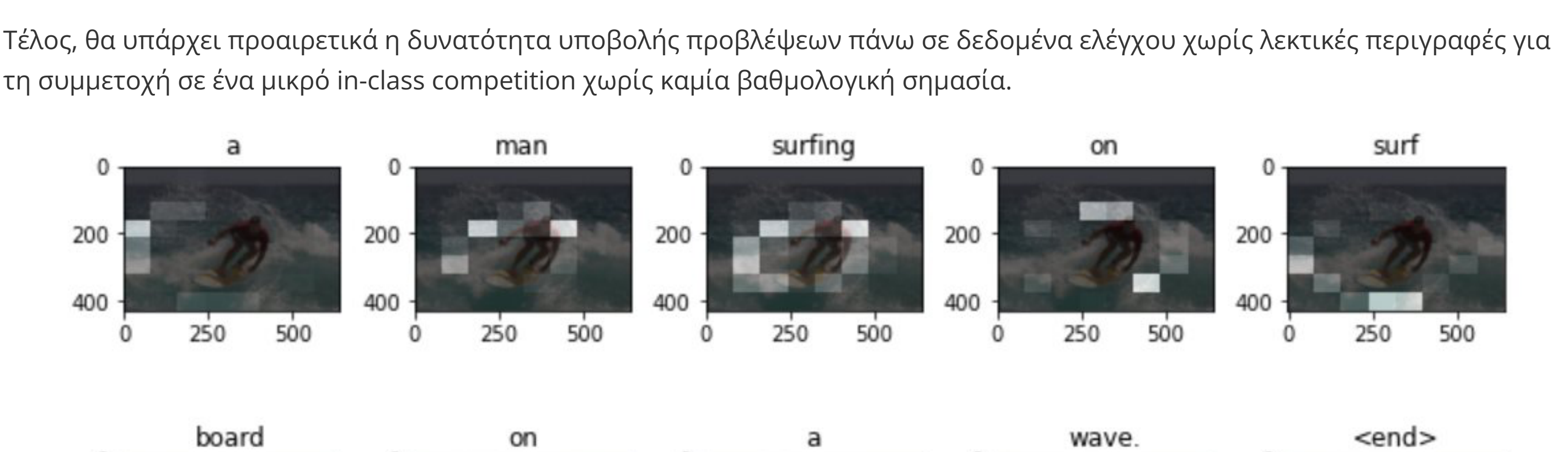
3η Εργασία: Βαθιά Μάθηση

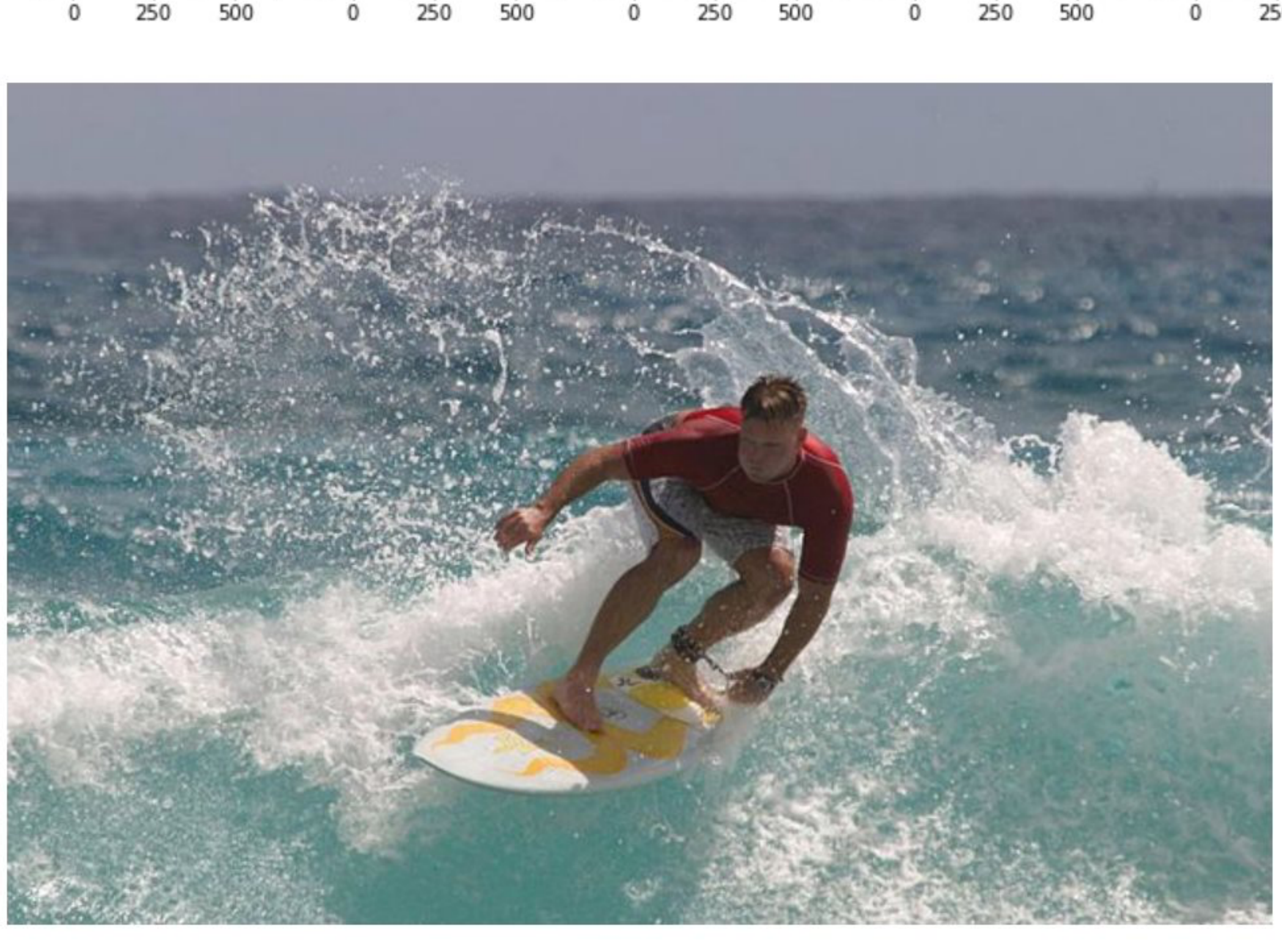
Image Captioning

Θέμα της 3ης εργασίας του μαθήματος είναι η Βαθιά Μάθηση. Θα μελετήσουμε ένα πρόβλημα που συνδυάζει Όραση Υπολογιστών και Επεξεργασία Φυσικής Γλώσσας. Συγκεκριμένα, θα φτιάξουμε ένα νευρωνικό δίκτυο παραγωγής λεκτικών περιγραφών από εικόνες (Image Captioning).

Σημείο εκκίνησης θα είναι το επίσημο tutorial του TensorFlow "Image captioning with visual attention". Θα δουλέψουμε ωστόσο σε άλλο dataset και θα προσπαθήσουμε να βελτιώσουμε το tutorial σε διάφορα σημεία.

Τέλος, θα υπάρχει προαιρετικά η δυνατότητα υποβολής προβλέψεων πάνω σε δεδομένα ελέγχου χωρίς λεκτικές περιγραφές για τη συμμετοχή σε ένα μικρό in-class competition χωρίς καμία βαθμολογική σημασία.





Σύνολο δεδομένων

Τα ευρύτερα χρησιμοποιούμενα datasets στο Image Captioning είναι τα Flickr8k, Flickr30k, και το COCO. Το παράδειγμα του TensorFlow χρησιμοποιεί το COCO. Εμείς θα χρησιμοποιήσουμε το "flickr30k-images-ecemod", μια παραλλαγή του flick30k για το μάθημά μας.

Τα δεδομένα του flickr30k-images-ecemod είναι τα εξής:

- ένας φάκελος "image_dir" με 31.783 εικόνες από το Flickr
- ένα αρχείο "train_captions.csv" με 148.915 captions για τις εικόνες του "image_dir"
- ένα αρχείο "test_images.csv" με 2.000 ονόματα εικόνων από το "imag_dir" που δεν έχουν captions εκπαίδευσης

Στο επόμενο κελί κατεβάζουμε τις εικόνες του dataset:

```
# Download image files
image_zip = tf.keras.utils.get_file('flickr30k-images-ecemod.zip',
                                     cache_subdir=os.path.abspath('.'),
                                     origin='https://spartacus.1337.cx/flickr-mod/flickr30k-images-ecemod.zip',
                                     extract=True)
os.remove(image_zip)
```

και με το επόμενο τα "train_captions.csv" και "test_images.csv":

```
# Download train captions file
train_captions_file = tf.keras.utils.get_file('train_captions.csv',
                                          cache_subdir=os.path.abspath('.'),
                                          origin='https://spartacus.1337.cx/flickr-mod/train_captions.csv',
                                          extract=False)

# Download test files list
test_images_file = tf.keras.utils.get_file('test_images.csv',
                                          cache_subdir=os.path.abspath('.'),
                                          origin='https://spartacus.1337.cx/flickr-mod/test_images.csv',
                                          extract=False)
```

Το flickr30k-images-ecemod έχει παρόμοια οργάνωση με το COCO. Κάθε εικόνα έχει 5 captions που έχουν γίνει από διαφορετικούς ανθρώπους μέσω της υπηρεσίας Mechanical Turk της Amazon. Ένα παράδειγμα:

`_100007487.jpg| 0| A young child walks down a gravel path lined with a row of red outdoor chairs .`

`_100007487.jpg| 1| A racetrack with red chairs stacked beside fence with a child walking .`

`_100007487.jpg| 2| A child in a striped shirt walks by some red chairs .`

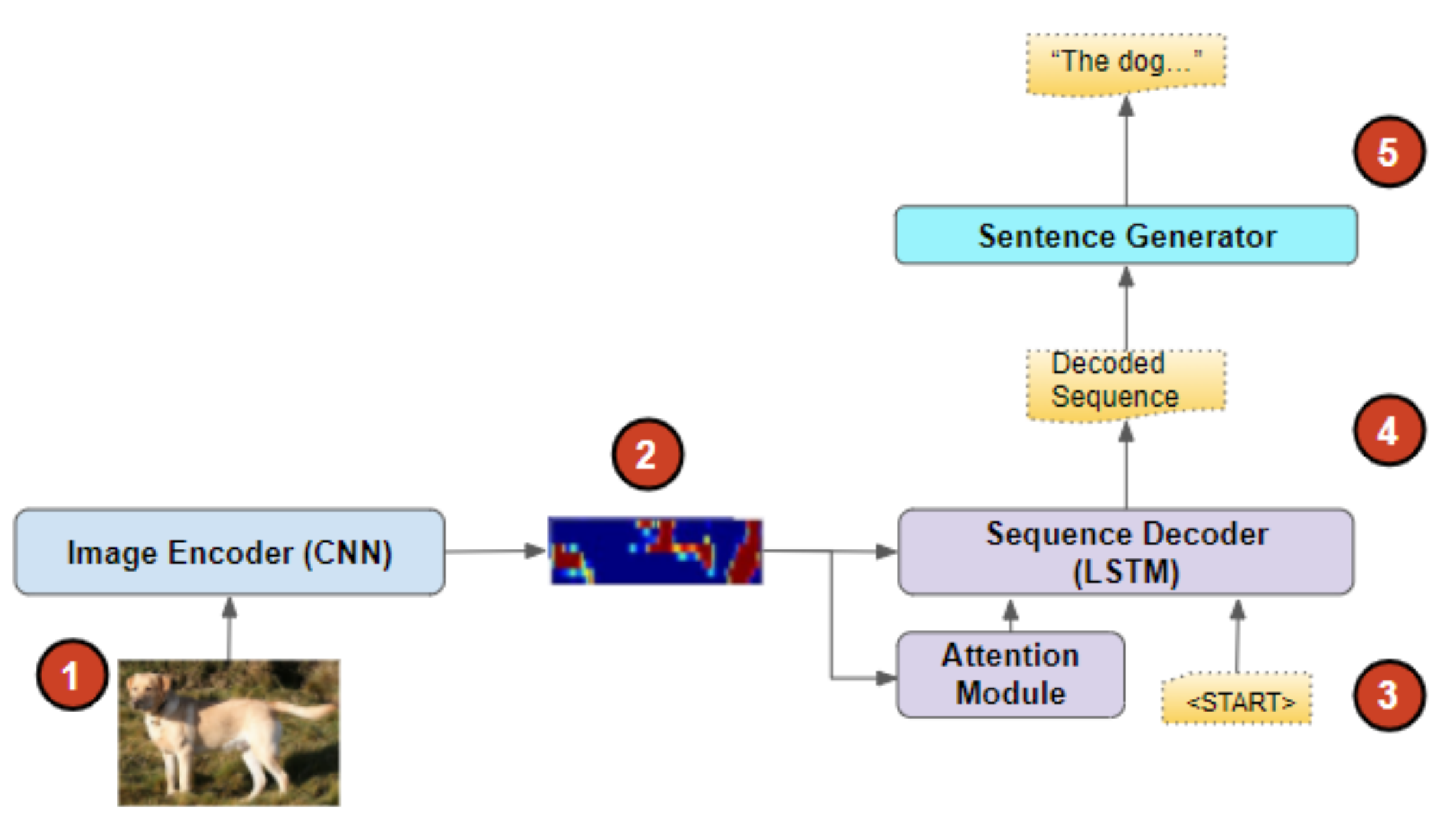
`_100007487.jpg| 3| A child walking and leaving a trail behind them .`

`_100007487.jpg| 4| A little kid is walking next to red banners .`

Κάθε caption έχει τρία πεδία, το όνομα του αρχείου της εικόνας, τον αύξοντα αριθμό του caption και τέλος το ίδιο το caption.

Το μοντέλο

Το μοντέλο του tutorial ακολουθεί την συνήθη αρχιτεκτονική στη Βαθιά Μάθηση του Encoder - Decoder.



Στον Encoder αντιστοιχούν τα βήματα 1 και 2 και στον Decoder τα βήματα 3-5.

Encoder: ένα pretrained συνελικτικό λαμβάνει ως είσοδο την εικόνα (1). Η μεταφορά μάθησης γίνεται χωρίς το classification head. Μας ενδιαφέρει μόνο η έξοδος που δίνουν για την εικόνα τα ανώτερα συνελικτικά επίπεδα του δικτύου. Εφόσον έχει εκπαιδευτεί επιτυχώς σε κάποιο classification task, τα συνελικτικά επίπεδα θα πρέπει να εξάγουν αποτελεσματικά τα σημαντικά χαρακτηριστικά της εικόνας (2). Το CNN λειτουργεί λοιπόν ως feature extractor δημιουργώντας μια "πλούσια" κωδικοποίηση της εικόνας.

Decoder: ο decoder λαμβάνει την είσοδο του encoder μέσα από ένα fully conected επίπεδο. Λαμβάνει επίσης μια-μία τις λέξεις της περιγραφής με τη μορφή embeddings (3). Το επίπεδο της προσοχής (Attention Module[1]) λαμβάνει ως είσοδο το hidden state του προηγούμενου βήματος και την είσοδο από τον encoder. Υπολογίζει την ευθυγράμμιση (alignement) των περιοχών της εικόνας με την κρυφή κατάσταση και περνάει τις τιμές από softmax. Οι τιμές αυτές πολλαπλασιάζονται με την είσοδο του encoder δημιουργώντας το context vector: κάποιες περιοχές της εικόνας θα είναι πιο ενεργές για τη δημιουργία της επόμενης κατάστασης. Το επαναληπτικό δίκτυο λαμβάνει ως είσοδο τα embeddings και το context vector και παράγει την επόμενη κρυφή κατάσταση (4). Τέλος, το δίκτυο εξάγει πιθανότητες για κάθε δυνατή επόμενη λέξη και ο Sentence Generator την επιλέγει με κάποια μεθοδολογία όπως με τυχαία επιλογή που ακολουθεί την κατανομή των πιθανοτήτων των λέξεων (5).

Ο μηχανισμός της προσοχής (attention) είναι βασικός στην αρχιτεκτονική των Transformers που είναι το state-of-the-art σε παρόμοια tasks (βλέπε paper "Attention Is All You Need").

Αξιολόγηση της ποιότητας του captioning

Το tutorial δεν περιλαμβάνει κάποια αναφορά στην ποιότητα του παραγόμενου captioning. Αν θεωρήσουμε ότι κάθε εικόνα έχει κάποια αληθινά captions (references) και το νευρωνικό παράγει ένα δικό του caption (hypothesis) θα χρησιμοποιήσουμε το BLEU (Bilingual Evaluation Understudy) score, μεταξύ hypothesis και references. Συνοπτικά, το BLEU είναι ένας σταθμισμένος μέσος όρος του πλήθους των κοινών unigrams, bigrams, trigrams, και fourgrams μεταξύ hypothesis και references. Το χειρότερο captioning λαμβάνει 0 και το καλύτερο 1. Δείτε ένα αναλυτικό παράδειγμα υπολογισμού του BLEU εδώ.

Η NLTK στο nltk.translate.bleu_score παρέχει τις απαραίτητες συναρτήσεις για τον υπολογισμό των BLEU scores:

- Για να μπορείτε να αξιολογείτε το captioning ενός μεμονωμένου παραδείγματος θα χρησιμοποιήσετε την sentence_bleu.
- Για να αξιολογείτε το captioning περισσότερων εικόνων πχ ολόκληρου του validation set θα χρησιμοποιήσετε την corpus_bleu. Σημειώστε ότι το corpus_bleu δεν είναι μέσος όρος των sentence_bleu.

Σε όλες τις περιπτώσεις χρησιμοποιήστε τις εξής παραμέτρους: weights=(0.4, 0.3, 0.2, 0.1) και smoothing_function=SmoothingFunction().method1.

Βελτιώσεις (και παραδοτέα)

Εφόσον αποκτήσετε μια εικόνα της επίδοσης του έτοιμου δικτύου (loss και χρόνος εκπαίδευσης στο train, BLEU στο validation) θα δοκιμάσουμε κάποιες ιδέες για βελτιώσεις στο δίκτυο.

Προεπεξεργασία κειμένου

1. Τα captions έχουν διαφορετικά μήκη. Τα πολύ σύντομα και τα πολύ μεγάλα δεν είναι χρήσιμα στην εκπαίδευση. Στο tutorial χρησιμοποιείται ad-hoc ένα μέγιστο μήκος 50 λέξεων. Φιλτράρετε το dataset έτσι ώστε να έχετε ένα range από 25 έως 35 διαφορετικά μήκη, χωρίς ούτε τα πολύ μικρά και χωρίς ούτε τα πολύ μεγάλα (δοκιμάστε ένα ιστόγραμμα).
2. Η συνάρτηση standardize θα μπορούσε να περιλαμβάνει και μερικά φίλτρα ακόμα κανονικοποίησης. Προσοχή εδώ δεν κάνουμε stemming.
3. Το tutorial αποφασίζει ad hoc για ένα vocabulary 5000 λέξεων. Δοκιμάστε και κάποια διαφορετικά (μεγαλύτερα) μεγέθη vocabulary.

Encoder

Στα έτοιμα μοντέλα CNN του Keras υπάρχουν 4 μοντέλα που εμφανίζονται να έχουν καλύτερες επιδόσεις από το InceptionV3 του tutorial. Κάντε mod 4 τον αριθμό της ομάδας σας στο εργαστήριο και δοκιμάστε να χρησιμοποιήσετε ως encoder, εκτός του InceptionV3, το δίκτυο που σας αντιστοιχεί ως ακολούθως:

mod 4 αριθμού ομάδας	δίκτυο
0	NASNetLarge
1	InceptionResNetV2
2	Xception
3	ResNet152V2

Σημειώστε ότι κάθε δίκτυο απαιτεί διαφορετική προεπεξεργασία των εικόνων ώστε να είναι η ίδια με αυτήν με την οποία εκπαιδεύτηκε. Όπως αναφέρεται στο documentation του κάθε δικτύου, μπορείτε να χρησιμοποιήσετε τις εκάστοτε συναρτήσεις "preprocess_input".

Embeddings

Στο παράδειγμα του tutorial τα embeddings μαθαίνονται κατά την εκπαίδευση του μοντέλου. Αντί αυτού μπορούμε να χρησιμοποιήσουμε έτοιμα embeddings με μεταφορά μάθησης κάτι που θα μειώσει και το πλήθος των βαρών προς εκπαίδευση. Δείτε πώς αποδίδει το δίκτυο με τα embeddings glove-wiki του Gensim διαφόρων διαστάσεων (50, 100, 200, 300).

Regularization

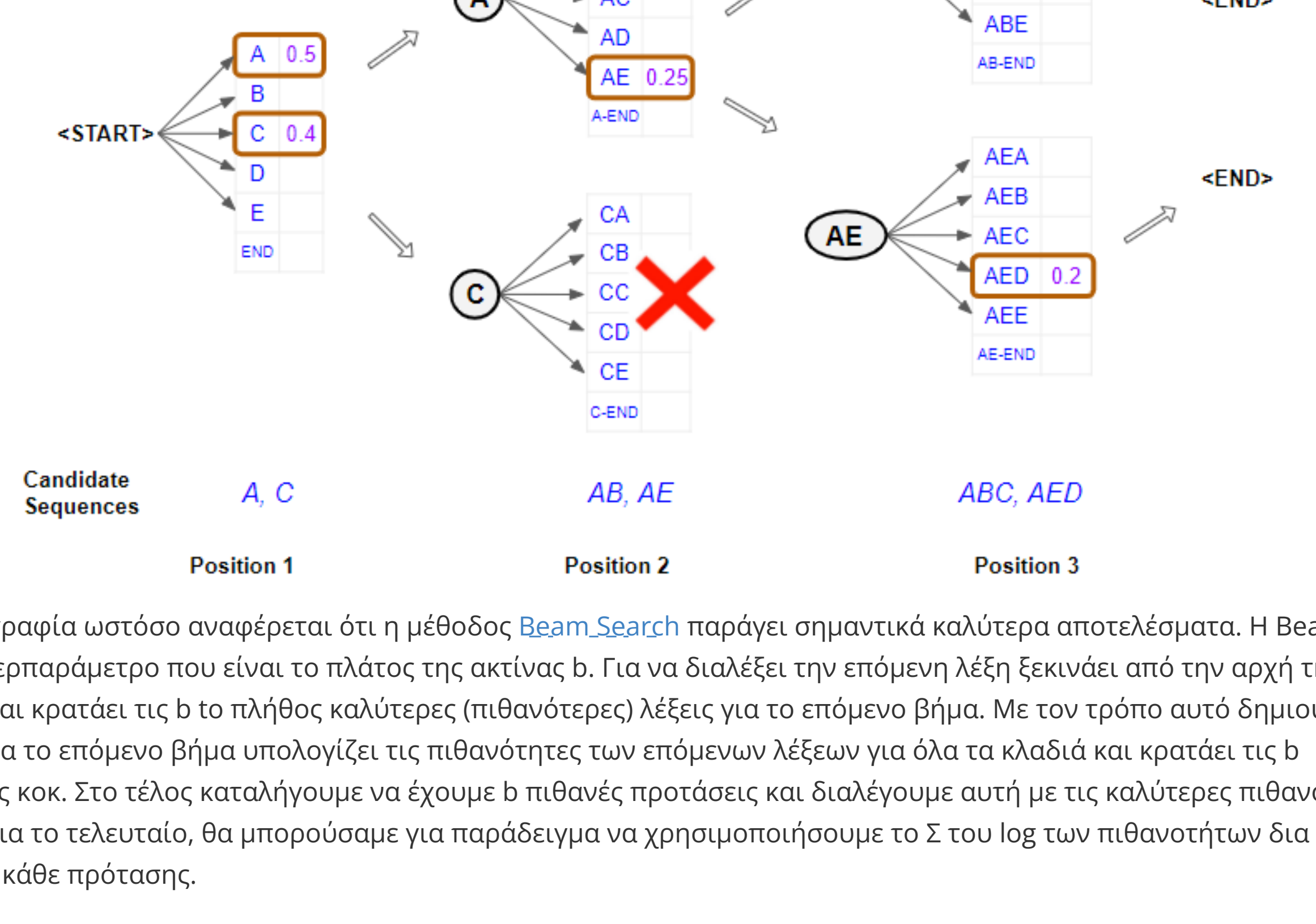
Το δίκτυο του tutorial δεν έχει κανένα μηχανισμό ομαλοποίησης. Δοκιμάστε να προσθέσετε επίπεδα Dropout κοντά στα πυκνά επίπεδα.

Decoder

Το δίκτυο του tutorial χρησιμοποιεί 512 μονάδες GRU. Δείτε πως επιδρά το πλήθος των μονάδων στην επίδοση (όχι στη λογική gridsearch) καθώς και το είδος τους, αν έχουμε δηλαδή LSTM αντί για GRU.

Sentence Generator

Το παράδειγμα χρησιμοποιεί την tf.random.categorical για να επιλέξει την επόμενη λέξη. Μια άλλη δυνατότητα είναι να διαλέγει κανείς την επόμενη πιο πιθανή λέξη (Greedy Search).



Στη βιβλιογραφία ωστόσο αναφέρεται ότι η μέθοδος Beam Search παράγει σημαντικά καλύτερα αποτελέσματα. Η Beam Search έχει μια υπερπαράμετρο που είναι το πλάτος της ακτίνας b. Για να διαλέξει την επόμενη λέξη ξεκινάει από την αρχή της πρότασης και κρατάει τις b το πλήθος καλύτερες (πιθανότερες) λέξεις για το επόμενο βήμα. Με τον τρόπο αυτό δημιουργούντα b κλαδιά. Για το επόμενο βήμα υπολογίζει τις πιθανότητες των επόμενων λέξεων για όλα τα κλαδιά και κρατάει τις b πιθανότερες κοκ. Στο τέλος καταλήγουμε να έχουμε b πιθανές προτάσεις και διαλέγουμε αυτή με τις καλύτερες πιθανότητες συνολικά. Για το τελευταίο, θα μπορούσαμε για παράδειγμα να χρησιμοποιήσουμε το Σ του log των πιθανοτήτων δια του μήκους της κάθε πρότασης.

Θα βρείτε έτοιμες υλοποιήσεις για το Beam Search που μπορείτε να προσαρμόσετε, όπως επίσης και συνήθεις τιμές για το b.

Παρατηρήσεις και συμβουλές

- Οι χρόνοι εκπαίδευσης είναι σημαντικοί (substantial). Χρησιμοποιήστε οπωσδήποτε κάποια μέθοδο αποθήκευσης του μοντέλου ώστε να μπορείτε να συνεχίζετε την εκπαίδευση.
- Στο Colab συνίσταται να κάνετε mount ένα Google Drive για να αποθηκεύονται τα μοντέλα. Στο Kaggle για να αποθηκευτεί κάτι θα πρέπει να γίνει commit το notebook. Θα βρείτε την έξοδο προηγούμενου notebook αν κάνετε "Add data", tab "Notebook Output Files" και φιλτράρετε με "Your Work". Εναλλακτικά, μπορείτε να κατεβάζετε ένα αρχείο που βρίσκεται στο directory "working" με την μέθοδο IPython και να το ανεβάζετε ως dataset (χωρίς απαραίτητα να είναι). Δείτε επίσης και τη μέθοδο gdown για το Kaggle για γρήγορο φόρτωμα δεδομένων από Drive.
- Στις δοκιμές μας το Kaggle ήταν 4-5 φορές ταχύτερο από το Colab σε ό,τι ήταν GPU intensive. Οι 36 ώρες ενεργής GPU του Kaggle ανά εβδομάδα μπορούν να είναι 3x36 για όλα τα μέλη της ομάδας.
- Μπορείτε να χρησιμοποιήσετε seed, np.random.seed, και tf.random.set_seed για να έχετε σταθερά split στο dataset. Επίσης, αν δεν κάνετε shuffle για να πάρετε το training set, δεν θα χρειάζετε να φτιάχνετε κάθε φορά τα αρχεία npy.
- Προφανώς δεν μπορούμε να κάνουμε cross-validation των παραμέτρων και αρχιτεκτονικών. Χρησιμοποιείτε σταδιακό training (σχετικά λίγα epochs) και εκτίμηση της απόδοσης του δικτύου με βάση το loss και τον απαιτούμενο χρόνο, ενώ παράλληλα εξετάζετε την ποιότητα του captioning με την sentence_bleu για επιλεγμένες εικόνες και την corpus_bleu για ένα μικρότερο κομμάτι του validation set.
- Συμβουλευτείτε το FAQ των ασκήσεων για λύσεις σε συγκεκριμένα προβλήματα και συμβουλές για την εκπαίδευση.

Παραδοτέο

Το notebook με το καλύτερο μοντέλο σας ως προς το corpus_bleu σε ολόκληρο το validation set. Χρησιμοποιήστε markdown για να εξηγήσετε τις επιλογές σας. Μπορείτε να σημειώσετε τιμές του bleu και για ενδιάμεσες επιλογές και να τις παρουσιάσετε σε πίνακα. Δώστε παραδείγματα εικόνων με επιτυχημένα και αποτυχημένα captioning.

Διαγωνισμός

Η συμμετοχή στο διαγωνισμό είναι προαιρετική και δεν έχει καμία βαθμολογική σημασία. Επειδή ωστόσο ο διαγωνισμός υπολογίζει το corpus_bleu για τα captions των εικόνων του test_images.csv μπορεί να σας δίνει μέσω του leaderboard μια ένδειξη για το πόσο έχετε προχωρήσει στη βελτίωση των captions.

Δημιουργία αρχείου υποβολής

Έστω ότι έχετε μια συνάρτηση που δημιουργεί captions για μια λίστα εικόνων hypotheses = get_caption(files). Για να μπορείτε να υπολογίζετε corpus_bleu στο validation θα πρέπει η συνάρτηση να επιστρέφει λίστα η οποία αν περιείχε δύο hypotheses θα είχε την μορφή:

`[['a', 'woman', 'floats', 'with', 'her', 'face', 'out', 'of', 'water', 'in', 'a', 'pool', 'with', 'another', 'woman', 'nearby', 'posing', 'for', 'the', 'camera'], ['a', 'black', 'and', 'white', 'dog', 'walks', 'along', 'a', 'sandy', 'beach']]`

Υπολογίζετε τα captions για τις εικόνες του test_images.csv σε μια μεταβλητή test_hypotheses και την αποθηκεύετε ως JSON ως εξής:

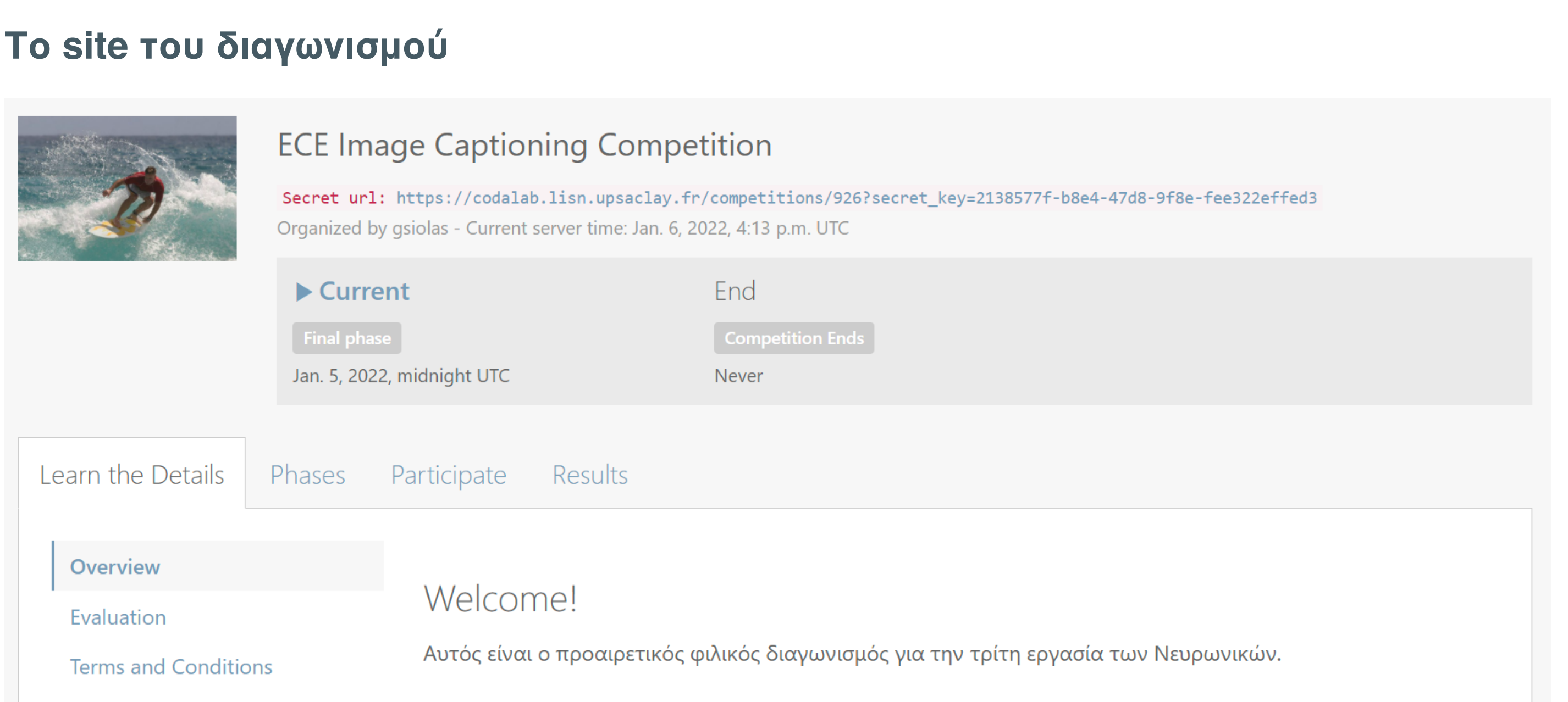
```
import json

jsonString = json.dumps(test_hypotheses)
jsonFile = open("test_hypotheses.json", "w")
jsonFile.write(jsonString)
jsonFile.close()
```

Κατεβάζετε τοπικά το αρχείο test_ypotheses.json, το μετονομάζετε υποχρεωτικά σε answer.txt και το zipάρετε σε zip file το όνομα του οποίου δεν έχει σημασία.

Συνημμένο στην εκφώνηση θα βρείτε ένα παράδειγμα λειτουργικού answer.txt.

Το site του διαγωνισμού



ECE Image Captioning Competition

Secret url: https://codalab.lisn.upsaclay.fr/competitions/926?secret_key=2138577f-b8e4-47d8-9f8e-fee322effed3

Organized by gsiolas - Current server time: Jan. 6, 2022, 4:13 p.m. UTC

▶ Current	End
Final phase	Competition Ends
Jan. 5, 2022, midnight UTC	Never

Welcome!

Αυτός είναι ο προαιρετικός φιλικός διαγωνισμός για την τρίτη εργασία των Νευρωνικών.

Ο διαγωνισμός βρίσκεται στο Codalab σε αυτό το σύνδεσμο. Χρησιμοποιήθηκε το Codalab αντί του Kaggle γιατί το Kaggle στο community tier δεν επιτρέπει custom μετρικές όπως η BLEU. Το Codalab είναι επίσης το επίσημο αποθετήριο του COCO.

Θα πρέπει να κάνετε register. Σημειώστε ότι αν πάτε στα settings στο profile σας μπορείτε να ορίσετε όνομα ομάδας και μέλη της με τα username σας στο Codalab. Προφανώς το όνομα της ομάδας μπορεί να είναι ό,τι θέλετε, άσχετο με τον αριθμό στο εργαστήριο.

Υποβολή απάντησης και leaderboard

Για να ανεβάσετε μια απάντηση, θα πρέπει να πάτε στο tab "Participate", στο "Submit/View Results" και να πατήσετε Submit για να ανεβάσετε το αρχείο υποβολής.

Στο tab "Results" θα βρείτε το leaderboard των submissions με βάση το corpus_bleu. Στο διαγωνισμό θα λαμβάνετε λίγο μεγαλύτερες τιμές corpus_bleu σε σχέση με αυτό που παρατηρείτε στο validation set λόγω ασυμβατότητας της version της Python του Codalab με τους τελευταίους ορισμούς στην nltk.translate.bleu_score (κάναμε κάποιες μικρές τροποποιήσεις στο source για να τρέξει).

Παρακαλούμε: όχι υποβολές με δίκτυα που έχουν εκπαιδευτεί σε φυσικές GPU, όχι προσπάθεια reverse engineering των captions του test set. Και τα δύο γίνονται αλλά δεν είναι στο πνεύμα του διαγωνισμού.

---

1. https://d2l.ai/chapter_attention-mechanisms/ ↩