Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Επεξεργασία Φωνής και Φυσικής Γλώσσας, 2024-2025

Αναφορά 3ου Εργαστηρίου: Fine-tune Llama

Όνομα:	Γεώργιος
Επώνυμο:	Οικονόμου
ΑΜ:	03121103

Εισαγωγή

Ο κώδικας, συνοδευόμενος από σύντομα σχόλια, παρουσιάζεται στο .ipynb αρχείο που εξήχθη από το Kaggle.

Εκτέλεση

Το kernel αντιγράφηκε σε Colab Notebook και ενεργοποιήθηκε η GPU τύπου T4 για το τρέχον session.

Βήμα 1: Εγκατάσταση Απαραίτητων Βιβλιοθηκών

Εγκαθίστανται οι απαραίτητες βιβλιοθήκες pytorch, unsloth, evaluate, xformers και εξαρτήσεις. Ο κώδικας τροποποιείται ελαφρώς ώστε να αποφεύγονται συγκρούσεις εξαρτήσεων, σε σχέση με τη βιβλιοθήκη protobuf.

Ερώτηση

Η βιβλιοθήκη Unsloth [1] αποτελεί ένα ολοκληρωμένο πλαίσιο για τη βελτιστοποίηση της εκπαίδευσης και του fine-tuning μεγάλων γλωσσικών μοντέλων, τόσο τοπικά όσο και σε πλατφόρμες όπως το Google Colab και το Kaggle. Απλοποιεί και ενοποιεί όλα τα απαραίτητα βήματα της διαδικασίας, όπως η φόρτωση του μοντέλου, το quantization, η εκπαίδευση, η αξιολόγηση, η εκτέλεση, η αποθήκευση, και η εξαγωγή.
Τα πλεονεκτήματά της σε σχέση με παρόμοιες βιβλιοθήκες είναι πως προσφέρει αποδοτική διαδικασία fine-tuning μεγάλων γλωσσικών μοντέλων, με μειωμένες απαιτήσεις σε μνήμη μέσω τεχνικών όπως το quantization και το RoPE. Είναι πλήρως συμβατή με τη βιβλιοθήκη

Hugging Face, μάλιστα είναι βασισμένη σε αυτή, και συγκεντρώνει διαδικασίες για τον χρήστη. Τέλος, υποστηρίζει σύγχρονα μοντέλα όπως το Llama 3.2 3B.

Βήμα 2: Φόρτωση του Προ-Εκπαιδευμένου Μοντέλου Llama

Το μοντέλο Llama 3.2 3B φορτώνεται χρησιμοποιώντας το εργαλείο FastLanguageModel από τη βιβλιοθήκη Unsloth.

Ερώτηση

Το 4-bit quantization μειώνει δραστικά τις απαιτήσεις μνήμης για την προσαρμογή μεγάλων γλωσσικών μοντέλων. Μειώνει την ακρίβεια των αριθμών που χρησιμοποιούνται για την αναπαράσταση των βαρών και, μερικές φορές, των ενεργοποιήσεων του μοντέλου. Το quantization μετατρέπει τους αριθμούς κινητής υποδιαστολής σε αριθμούς χαμηλότερης ακρίβειας, στην συγκεκριμένη περίπτωση σε 4-bit ακέραιους. Όσον αφορά την απόδοση, ενώ η ταχύτητα εκπαίδευσης δεν αυξάνεται σημαντικά, η ταχύτητα του inference μπορεί να βελτιωθεί αισθητά. Παρά τη μείωση της αριθμητικής ακρίβειας, η απώλεια στην ακρίβεια του μοντέλου είναι συνήθως μικρή έως αμελητέα, χάρη σε εξελιγμένες τεχνικές κβαντοποίησης που διατηρούν την απόδοση σε υψηλά επίπεδα.

Βήμα 3: Προετοιμασία Parameter-Efficient Fine-Tuning (PEFT)

Χρησιμοποιείται η μέθοδος PEFT [2] σε 28 επίπεδα του μοντέλου. Συγκεκριμένα, μέσα σε καθένα από αυτά τα 28 επίπεδα, έχει παρέμβει και τροποποιήσει για το καθένα 1 QKV, 1 O και 1 MLP υποσύστημα.

Ερώτηση

Τα βασικά πλεονεκτήματα της μεθόδου PEFT είναι η δραστική μείωση των απαιτήσεων σε μνήμη (VRAM) και υπολογιστική ισχύ, καθώς εκπαιδεύεται μόνο ένα πολύ μικρό ποσοστό των παραμέτρων του μοντέλου. Αυτό οδηγεί σε ταχύτερη προσαρμογή, μικρότερο αποθηκευτικό χώρο για τα προσαρμοσμένα μοντέλα και ευκολότερη εναλλαγή μεταξύ διαφορετικών tasks.
Η τεχνική LoRA [3] συμβάλλει εισάγοντας μικρές, εκπαιδεύσιμες μετρικές χαμηλής τάξης (Low-Rank Adaptation) σε επιλεγμένα επίπεδα του προ-εκπαιδευμένου μοντέλου. Μόνο αυτές οι λίγες νέες παράμετροι εκπαιδεύονται, ενώ το αρχικό, μεγάλο μοντέλο παραμένει παγωμένο, επιτυγχάνοντας έτσι την αποδοτικότητα της PEFT.

Βήμα 4: Προετοιμασία Δεδομένων για Supervised Fine-Tuning (SFT)

Χρησιμοποιείται το σύνολο δεδομένων FineTome-100k, σύμφωνα με το στυλ του ShareGPT. Τα δεδομένα μετατρέπονται σε μορφή συνομιλιών του HuggingFace με τη δομή ("role", "content") αντί για ("from", "value").

Άσκηση

Το σύνολο δεδομένων FineTome-100k είναι τυπικά δομημένο ως μια λίστα από συνομιλίες. Κάθε παράδειγμα αποτελείται από μια ακολουθία μηνυμάτων, όπου κάθε μήνυμα είναι ένα λεξικό με τουλάχιστον δύο βασικά πεδία: ("content", "role"). Το πεδίο "content" είναι το κείμενο του μηνύματος και το πεδίο "role" είναι ο πομπός που παρήγαγε το μήνυμα, με

συνηθισμένες τιμές το "user", που αντιστοιχεί στην είσοδο, και το "assistant", που αντιστοιχεί στην απάντηση του μοντέλου.

Βήμα 5: Προσαρμογή του Μοντέλου με Supervised Fine-Tuning (SFT)

Χρησιμοποιείται η SFTTrainer από τη βιβλιοθήκη HuggingFace TRL.

Άσκηση

Οι παράμετροι που ρυθμίζονται για το fine-tuning είναι:

- num_train_epochs: Καθορίζει πόσες φορές το μοντέλο θα περάσει από ολόκληρο το σύνολο δεδομένων εκπαίδευσης. Λίγες εποχές μπορεί να οδηγήσουν το μοντέλο να γίνει underfitted στην εκπαίδευση ενώ πολλές μπορεί να οδηγήσουν σε overfitting και το μοντέλο να μη γενικεύει σε νέα δεδομένα.
- max_steps: Περιορίζει τον αριθμό των ενημερώσεων των βαρών του μοντέλου. Αν οριστεί, η εκπαίδευση σταματά όταν φτάσει αυτό το όριο, ακόμα κι αν δεν έχουν ολοκληρωθεί όλες οι εποχές. Η σημασία του περιορισμού των βημάτων είναι πως ελέγχει τον συνολικό χρόνο εκπαίδευσης.
- batch_size [4]: Καθορίζει τον αριθμό των δειγμάτων που επεξεργάζονται ταυτόχρονα σε μία μόνο GPU κατά τη διάρκεια ενός βήματος εκπαίδευσης. Μεγαλύτερο batch_size απαιτεί περισσότερη μνήμη αλλά μπορεί να επιταχύνει την εκπαίδευση ανά εποχή.
- learning_rate: Καθορίζει το μέγεθος του βήματος με το οποίο ενημερώνονται τα βάρη του μοντέλου κατά τη διάρκεια της εκπαίδευσης, με βάση τα gradients. Αν ο ρυθμός είναι πολύ υψηλός, η εκπαίδευση μπορεί να γίνει ασταθής και τα βάρη να ταλαντώνονται. Αν είναι χαμηλός, η εκπαίδευση μπορεί να κολλήσει σε ένα τοπικό ελάχιστο του σφάλματος και να μη βρει την καλύτερη λύση.

Βήμα 5α: Παρακολούθηση Προόδου Εκπαίδευσης, Διαχείριση Checkpoints και Early Stopping

Ερωτήσεις

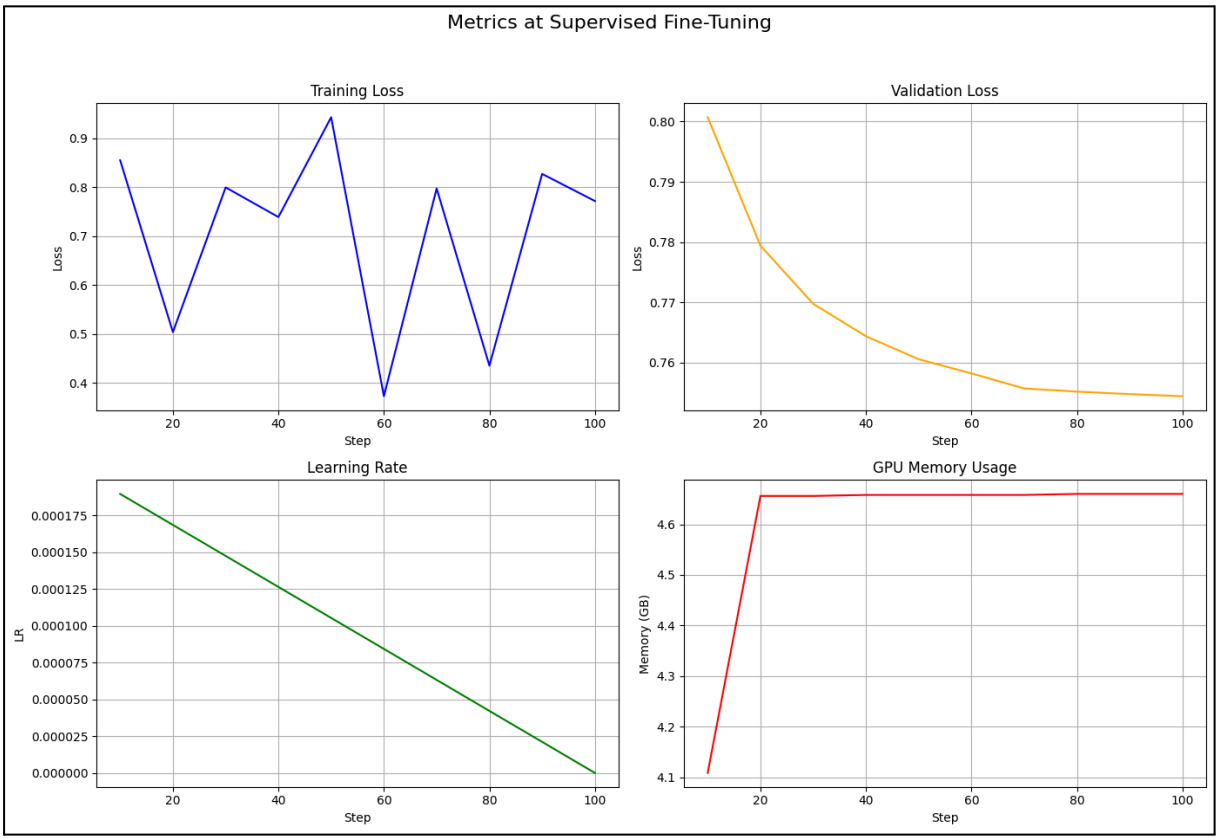
1. Κατά τη διάρκεια της εκπαίδευσης, η μέθοδος on_step_end() καλείται αυτόματα από τον Trainer στο τέλος κάθε βήματος. Σε αυτή τη μέθοδο, ελέγχεται αν το τρέχον βήμα state.global_step είναι πολλαπλάσιο του eval_log_step. Αν ισχύει, καταγράφονται βασικές μετρικές όπως το σφάλμα training_loss, ο ρυθμός learning_rate, το σφάλμα val_loss και η μνήμη της GPU, με χρήση της limited_evaluate() και της torch.cuda.memory_reserved() αντίστοιχα.
   Εάν παρατηρηθεί βελτίωση στο val_loss, αποθηκεύεται ένα νέο checkpoint του μοντέλου. Παράλληλα, γίνεται διαχείριση του πλήθους των αποθηκευμένων checkpoints. Η καταγραφή των μετρικών γίνεται μέσω των logs.get(). Όλη αυτή η διαδικασία εκτελείται αυτόματα από τον Trainer.
2. Στην υλοποίηση της μεθόδου on_step_end() το checkpoint θα αποθηκευτεί αν το val_loss βελτιωθεί από το προηγουμένως καλύτερο σφάλμα self.best_val_loss. Το early-stopping υλοποιείται μέσω της παρακολούθησης της πορείας του σφάλματος val_loss. Με άλλα λόγια, η εκπαίδευση διακόπτεται αν το σφάλμα δεν έχει βελτιωθεί επανειλημμένα και τότε το callback θέτει τη σημαία control.should_training_stop ίση με True.

3. Η χρήση μίας βιβλιοθήκης για την αποθήκευση των logs είναι εφικτή. Θα μπορούσε ενδεχομένως να χρησιμοποιηθεί η βιβλιοθήκη TensorBoard του PyTorch ώστε ο Trainer να επιστρέφει τις μετρικές έχοντας θέσει report_to ίσο με "tensorboard". Η υλοποίηση με CSV είναι απλή για βασικές ανάγκες, αλλά στερείται οπτικοποίησης, και απαιτεί χειροκίνητη συντήρηση. Αντίθετα, οι έτοιμες βιβλιοθήκες όπως το TensorBoard παρέχουν διαδραστική οπτικοποίηση και εύκολη σύγκριση.

Βήμα 5β: Οπτικοποίηση Βασικών Μετρικών κατά την Εκπαίδευση SFT

Οπτικοποιούνται οι μετρικές training loss, validation loss, learning rate, και memory usage.

Άσκηση



Οι μετρικές validation loss, learning rate, και memory usage παρουσιάζουν φυσιολογικές συμπεριφορές. Το validation loss δείχνει μία ομαλή τάση μείωσης και τείνει σε σταθερή τιμή, γεγονός που αποδεικνύει πως το μοντέλο γενικεύει. Επίσης, ο ρυθμός learning rate μειώνεται γραμμικά και η χρήση της μνήμης αυξάνεται απότομα στην αρχή έως ότου σταθεροποιηθεί, κάτι που είναι φυσιολογικό.
Η μετρική του training lost παρουσιάζει πολύ απότομες αλλαγές και ασυνέπειες. Κάποιες από τις πιθανές αιτίες μπορεί να είναι πως ο ρυθμός learning rate είναι υψηλός για το μέγεθος του batch, με αποτέλεσμα το μοντέλο να ταλαντεύεται γύρω από τοπικά ελάχιστα και να μην συγκλίνει ομαλά. Επιπρόσθετα, μπορεί τα δεδομένα εκπαίδευσης να περιέχουν πολύ διαφορετικά δεδομένα, προκαλώντας τις ασυνέπειες.

Βήμα 6: Εκτέλεση Inference με το Προσαρμοσμένο Μοντέλο

Εκτελείται το προσαρμοσμένο μοντέλο, χρησιμοποιώντας τις παραμέτρους min_p και temperature.

Ερωτήσεις

Οι παράμετροι min_p [5] και temperature [6] κατά το inference μεγάλων γλωσσικών μοντέλων ρυθμίζουν πως επιλέγεται το επόμενο token. Επηρεάζουν την έξοδο όταν η δειγματοληψία είναι ενεργοποιημένη με do_sample ίσο με True. Σχετικά με το temperature, όταν η τιμή του είναι χαμηλή η έξοδος του μοντέλου γίνεται πιο προβλέψιμη ενώ αν αυξηθεί κατά πολύ η κατανομή των πιθανοτήτων ομαλοποιείται, δίνοντας έτσι ως έξοδο λιγότερα πιθανά token. Αυτό έχει ως αποτέλεσμα η έξοδος του μοντέλου να γίνεται πιο ποικιλόμορφη και δημιουργική, ένα σημαντικό χαρακτηριστικό κατά την παραγωγή λόγου για εργασίες όπως η δημιουργική γραφή ή η αιτιολόγηση.

Βήμα 7: Δημιουργία Απαντήσεων Χρησιμοποιώντας Βοηθητική Συνάρτηση

Ορίζεται η βοηθητική συνάρτηση και επιτυχώς το μοντέλο δίνει απάντηση δοθέντος του instruction.

Βήμα 8: Αποθήκευση και Φόρτωση Προσαρμοσμένων Μοντέλων

Το τελικό μοντέλο αποθηκεύεται ως LoRA adapters και όχι ολόκληρο χρησιμοποιώντας τη μέθοδο save_pretrained().

Ερώτηση

Τα πλεονεκτήματα της αποθήκευσης μόνο των LoRA adapters έγκειται στο μειωμένο αποθηκευτικό κόστος και στην δυνατότητα του base model να επαναχρησιμοποιηθεί. Με άλλα λόγια, το μοντέλο μπορεί να επαναχρησιμοποιηθεί εφόσον οι αλλαγές από το fine-tuning αποθηκεύονται ως adapters. Από την άλλη πλευρά, το μειονέκτημα της χρήσης των LoRA adapters είναι πως κατά το inference χρειάζεται το προκαταρκτικό βήμα να φορτωθεί το base model και έπειτα να εφαρμοστούν οι adapters. Αν είναι επιθυμητό, το συνολικό μοντέλο, δηλαδή το προ-εκπαιδευμένο και οι adapters, μπορεί να αποθηκευτεί πλήρως με τη μέθοδο save_pretrained().

Βήμα 9: Αξιολόγηση Μοντέλων με το OpenAssistant Conversations Dataset

Η απόδοση του base model και του LoRA-adapted model αξιολογήθηκε χρησιμοποιώντας το OpenAssistant Conversations Dataset.

Ασκήσεις

Το σύνολο δεδομένων OpenAssistant Conversations Dataset [7] είναι τυπικά δομημένο ως μια δενδρική δομή συνομιλιών. Κάθε παράδειγμα έχει πολλές απαντήσεις, και κάθε απάντηση μπορεί να έχει πολλές περαιτέρω, σχηματίζοντας ένα δέντρο διαλόγου. Κάθε μήνυμα είναι ένα λεξικό με δύο βασικά πεδία: "prompter" και "assistant". Το πεδίο "prompter" είναι ο χρήστης που ξεκινά τη συνομιλία με το αρχικό ερώτημα και το "assistant", που αντιστοιχεί στην απάντηση του μοντέλου. Τα μηνύματα του assistant αποτελούν την έξοδο του γλωσσικού μοντέλου.

Μετρικές BLEU, ROUGE και BERTScore

- Η μετρική BLEU (Bilingual Evaluation Understudy) υπολογίζει το precision των n-grams. Δηλαδή, μετρά πόσο παρόμοιο είναι το κείμενο που το μοντέλο δίνει ως έξοδο με ένα ή περισσότερα ανθρώπινα κείμενα υψηλής ποιότητας.
- Η μετρική ROUGE (Recall-Oriented Understudy for Gisting Evaluation) υπολογίζει το recall των n-grams. Δηλαδή, μετρά σε ποιο βαθμό το κείμενο που παράγει το μοντέλο περιλαμβάνει τις σημαντικές πληροφορίες που υπάρχουν στο κείμενο αναφοράς.
- Η μετρική BERTScore υπολογίζει την σημασιολογική ομοιότητα κειμένων. Δηλαδή, χρησιμοποιώντας word embeddings από προ-εκπαιδευμένα μοντέλα όπως το BERT μετρά το cosine similarity.

BLEU	Base Model	SFT Model
	0.0606	0.0625

ROUGE	Base Model	SFT Model
rouge1	0.2755	0.2791
rouge2	0.0796	0.0790
rougeL	0.1584	0.1635
rougeLsum	0.2295	0.2330

BERTScore	Base Model	SFT Model
Precision	0.8397	0.8415
Recall	0.8384	0.8374
F1 Score	0.8388	0.8392

Σύμφωνα με τις τιμές του BERTScore, το fine-tuned μοντέλο παρουσιάζει μια μικρή αλλά σταθερή βελτίωση σε σχέση με το base model, τόσο στην precision όσο και στην recall και το F1-score.
Για την αξιολόγηση των μοντέλων σε δεδομένα συνομιλιών η μετρική BERTScore είναι συγκριτικά η πιο κατάλληλη [8] καθώς χρησιμοποιεί word embeddings για να αναγνωρίσει τη σημασιολογική ομοιότητα. Οι ανθρώπινες συνομιλίες φέρουν συχνά συναισθηματικό φορτίο και τα νοήματα αλλάζουν με βάση τις περιστάσεις. Η μετρική, λοιπόν, είναι ανθεκτική στο να αναγνωρίσει αυτόν τον ανθρώπινο παράγοντα.

Βήμα 10: Προσαρμογή του Llama Μοντέλου με Direct Preference Optimization (DPO)

Το σύνολο δεδομένων Intel/orca_dpo_pairs φορτώθηκε από τη βιβλιοθήκη HuggingFace. Ωστόσο, το callback δεν μπορούσε να εφαρμοστεί γιατί το μοντέλο δεν έλαβε την ακολουθία εισόδου που χρειάζεται για να λειτουργήσει. Κατά τη διάρκεια της προσαρμογής DPO ο

τρόπος που προωθήθηκαν τα δεδομένα προς την forward δεν είχε την αναγκαία μορφή. Επειδή τελικά δεν κατέστη δυνατό να επιλυθεί το πρόβλημα, ορισμένα από τα υποερωτήματα δεν απαντήθηκαν.

Ερωτήσεις

Η μέθοδος Direct Preference Optimization (DPO) εκπαιδεύει το εκάστοτε γλωσσικό μοντέλο χρησιμοποιώντας ένα σύνολο δεδομένων για το οποίο για κάθε είσοδος “prompt” υπάρχει μια προτιμώμενη “chosen” και μη προτιμώμενη “rejected” απάντηση. Κατά την εκπαίδευση, το μοντέλο μαθαίνει να επιλέγει τη προτιμώμενη απάντηση από το ζεύγος προτιμήσεων και να μειώνει την πιθανότητα παραγωγής της “rejected” απάντησης. Η DPO σε σύγκριση με την προσαρμογή με reinforcement learning from human feedback (RLHF) προσφέρει απλότητα εφόσον δεν χρειάζεται να εκπαιδευτεί κάποιο ενδιάμεσο μοντέλο “reward”. Τέλος, η DPO είναι μία λιγότερο απαιτητική διαδικασία.

Βήμα 11: Αξιολόγηση του Μοντέλου που Προσαρμόστηκε με DPO

Βήμα 12: Σχεδίαση Ερωτήσεων Αξιολόγησης για Ποιοτική Ανάλυση

Δημιουργήθηκαν 7 ερωτήσεις αξιολόγησης για να συγκριθούν οι απαντήσεις που παράγουν τα μοντέλα base model και SFT model με τη μέθοδο generate_response().

Ποιοτική Ανάλυση

Συνοχή (Coherence): Είναι η απάντηση λογική και καλά δομημένη;

Το base model παρουσιάζει γενικά καλή συνοχή, καθώς συχνά οργανώνει την πληροφορία με δομημένο τρόπο, όπως με τη χρήση λιστών στην απάντηση της πρώτης ερώτησης. Ωστόσο, παρατηρούνται εξαιρέσεις, όπως στην έκτη ερώτηση, όπου η απόκριση περιλαμβάνει άσχετες πληροφορίες. Αντίθετα, το SFT model εμφανίζει ενδείξεις χαμηλότερης συνοχής, καθώς σε ορισμένες περιπτώσεις (π.χ. στην έβδομη ερώτηση) παρατηρείται περιττή επανάληψη φράσεων.
Το base model υπερέχει.

Σχετικότητα (Relevance): Ανταποκρίνεται άμεσα στην ερώτηση;

Το base model ανταποκρίνεται άμεσα σε όλες τις ερωτήσεις εξαιρουμένης της τρίτης όπου απαντά με περιττές πληροφορίες για άλλες χώρες. Αντίθετα, το SFT model συχνά επεκτείνεται σε άσχετα ή μη ερωτηθέντα θέματα με επαναλήψεις που μπερδεύουν.
Το base model υπερέχει.

Ορθότητα λόγου (Fluency): Είναι γραμματικά σωστή και ευανάγνωστη;

Το base model παράγει γραμματικά σωστό λόγο και οι απαντήσεις του είναι κατανοητές. Ομοίως, και το SFT παράγει ορθό και ευανάγνωστο λόγο.
Τόσο το base όσο το SFT απαντούν με ορθότητα.

Ακρίβεια (Accuracy): Για ερωτήσεις γνώσεων, είναι οι πληροφορίες σωστές;

Το base model επιδεικνύει ακρίβεια στις ερωτήσεις γνώσεων, όπως στην τρίτη και έβδομη. Ομοίως, και το SFT έχει καλή ακρίβεια.
Εν γένει, τα μοντέλα εμφανίζουν παρόμοια ακρίβεια.

Δημιουργικότητα (Creativity): Για ερωτήσεις ανοιχτού τύπου, είναι η απάντηση ευρηματική και πρωτότυπη;

Το base model στη δεύτερη μοναδική ερώτηση που απαιτούσε δημιουργικότητα απάντησε ικανοποιητικά, αλλά με ένα αρκετά κοινότυπο σενάριο. Αντίθετα, το SFT model εμφανίζει μεγαλύτερη δημιουργικότητα από το base model, εισάγοντας μια πιο πρωτότυπη ιδέα στην ιστορία, αυτή της Αιγύπτου.
Το SFT model υπερέχει.

Συμπερασματικά, και τα δύο μοντέλα παρέχουν ικανοποιητικές απαντήσεις, με συνολικά καλή απόδοση. Το base model τείνει να είναι πιο άμεσο, με σαφή ροή λόγου, γεγονός που διευκολύνει την κατανόηση των απαντήσεών του. Αντιθέτως, το SFT model παρουσιάζει αυξημένη δημιουργικότητα και παράγει πιο εκτενές λόγο, γεγονός που το καθιστά κατάλληλο για πιο ανοιχτού τύπου ερωτήσεις.

Πηγές

[1] Fast and Efficient Model Finetuning using the Unsloth Library
[2] PEFT - Hugging Face
[3] LoRA: Low-Rank Adaptation of Large Language Models
[4] Batch Size - Stack Exchange
[5] Balancing Min-P and Temperature - Reddit
[6] What is LLM Temperature? - IBM
[7] OpenAssistant Conversations Dataset - Hugging Face
[8] BERTScore vs. BLUE