Final Project - Almog Ben Simon (206571135) and Roee Barak (207375675)

המודל:

במהלך הפרויקט, עבור משימת התרגום עצמה, עבדנו עם מודל T5-Base שהינו pre-trained (והtokenizer המתאים). בחרנו לעבוד עם מודל זה, שכן על אף שאינו אומן על תרגום מגרמנית לאנגלית, הוא אומן על הכיוון ההפוך, ולכן מכיל מיפוי הגיוני של hidden space עבור 2 השפות.

שמנו לב כי מודל זה מקבל חסם לאורך הפלט, ומכיוון שלא חווינו חסרונות מהותיים בהצבת ערך גדול מאוד בתכונה זו, הצבנו 1000 במהלך העבודה.

במהלך הניסויים שקלנו להכניס לencoder של המודל את הגרמנית וה'רמזים' (הroots והmodifiers) בנפרד, ולשלב את הhidden state שלהם (תחילה ניסינו לבצע שכבת attention בין שני פלטי הencoder, וזה לא צלח מבחינת תוצאות ואלידציה. שניית ניסינו לקנקט את הוקטורים וגם זה לא צלח אמפירית את הולאדיציה), ואז להכניס לdecoder. אך הגענו למסקנה (אמפירית וסמי-תיאורטית) ששילוב ה'רמזים' בinput של המודל, במקום הפרדת תהליכי הencoding, היא יותר אפקטיבית, ומאפשרת למודל ללמוד נכון כיצד להסתמך על 'הרמזים' בתהליך התרגום. כלומר כאשר הencoder מקבל קלט הוא מצליח להבין את הקשרים בין כל המילים בקלט וגם להפריד אותם, וכאשר מכניסים אותם לencoders נפרדים הקשר בין הרמזים לקלט הגרמנית נאבד.

לכן לבסוף החלטנו להכניס קלט יחיד המשלב גרמנית ואת הroot and modifiers יחדיו, אך משימתנו הייתה להכניס למודל את הקלט כל שהוא ידע להפריד בין הרמזים למשפט בגרמנית בצורה היעילה ביותר. ניסינו מגוון הפרדות שאמצעות מילים מיוחדות, תווים מיוחדים, סדר של הרמזים, הצגת הרמזים כ " the hints: {hints} " למשל. והפרדות בין כל חלק והצגת הקלט בצורה היעילה ביותר. לבסוף החלטנו על קלט מסוג:
"Translate German to English ||| roots: { roots } ||| modifiers: { modifiers } ||| german: { the german sentence }".
ובאמת אמפירית קיבלנו תוצאות טובות יותר.

הערכת ערך מדד הBelu על קבצי התחרות – 38:

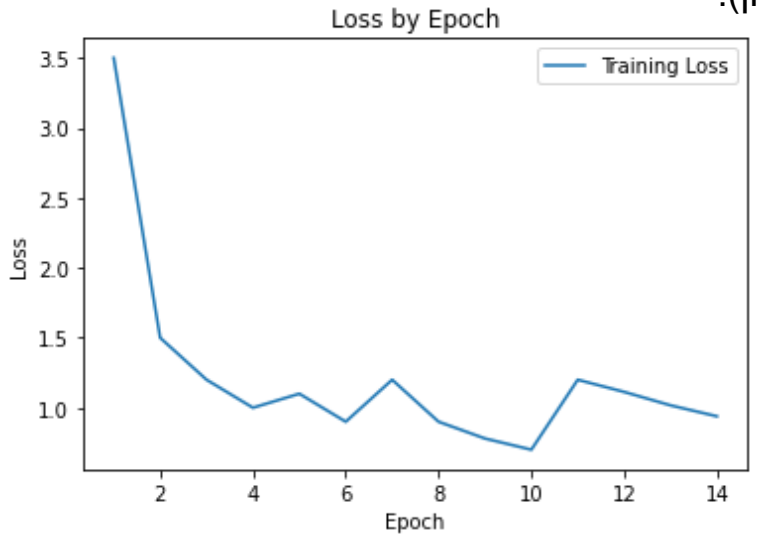
מניחים ערך זה מכיוון שאנו מניחים כי עשינו התאמות למודל על פי הval, כלומר שיפרנו את המודל רק בהתאם לקובץ זה. ואנו מניחים שקובץ הcomp שונה, והתאמות אלה לא בהכרח יהיו מתאימות באותה מידה לקובץ המבחן.

אימון (ומבחן):

עבור חישובי הloss, לשם אימון המודל, השתמשנו בcross entropy loss. מהסיבה הפשוטה שזה שגם loss אפקטיבי למשימה, וגם מובנה במודל T5-Base.
ביצענו בסהכ 14 אפוקים כדי להגיע לתוצאה הרצויה.
תחילה ניסינו לבצע fine-tuning על המידע שבtrain, ללא התחשבות בrootsוהmodifiers. רצינו לקבל Baseline לציון הBleu אליו נוכל להשוות מודלים מתקדמים יותר שנבצע בהמשך.
להפתעתנו קיבלנו תוצאת Bleu מאוד גבוה – 37, שגבוה בהרבה מסף ההגשה (30).
בהמשך, תכננו לשלב את הrootsוהmodifiers במודל, ולכן גם בתהליך האימון. לשם כך הבנו כי עלינו ליצור מיפוי דומה של הrootsוהmodifiers לקובץ הtrain, והחלטנו שנצטרך מודל נפרד בשביל משימה זו.
ראשית, החלטנו לנסות להגריל מילים אקראיות בתור הrootsוהmodifiers עבור קובץ האימון, מתוך ראייה שגם ללא המשמעות התחבירית, מדובר ב'רפרנס' למילים שאמורות להופיע בפרדיקציה – מידע מאוד חזק בפני עצמו.
לאכזבתנו, על אף שהצלחנו לעבור את התוצאה הקודמת של ה37, לא עברנו את ה38.

לאחר מכן, החלטנו להשתמש במודל (*'en_core_web_sm'*) מספריית spacy לשם חילוץ נכון של הrootsוהmodifiers עבור קובץ האימון (במקום ההגרלה הרנדומית שביצענו קודם).
כאשר קיבלנו יותר מ-2 modifiers עבור הroot, הגרלנו 2 מהם בלבד, כפי שבנוי הפורמט בקבצי הvalוהcomp.
יש לציין, שב-2 המקרים (גם כאשר הגרלנו מילים רנדומליות, וגם כאשר ביצענו ניתוח תחבירי), יצרנו מספר גרסאות שונות לא-מתויגות של קובץ הtrain, כאשר השוני נובע מאילו modifiers נבחרו עבור (במקרה של הניסיון הראשון, של ההגרלה הרנדומלית, גם הroot היה שונה בין הגרסאות).
במהלך האימון, בכל epoch, התאמנו על גרסה לא-מתויגת שונה על קובץ האימון, על מנת להכליל יותר את הלמידה ולאפשר למודל ללמוד יותר נכון את משמעות הmodifiers שאנו מספקים (כמו שבתחום של computer vision מבצעים 'אוגמנטציות' לתמונות על מנת להעשיר את מאגר האימון).

גרף לוס האימון שקיבלנו:



הסקה:

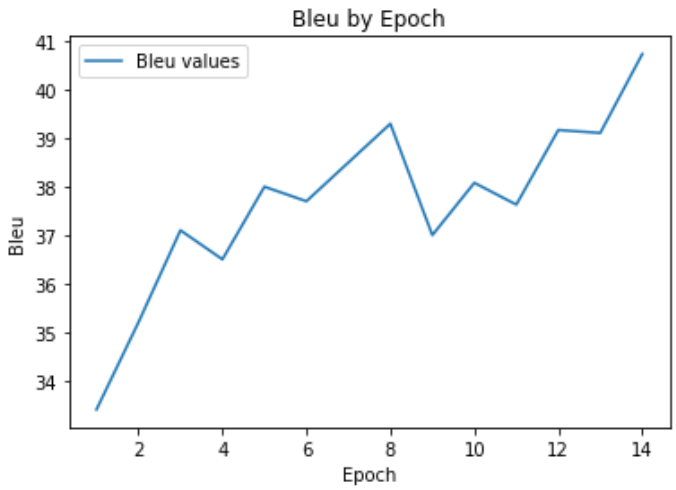
לשם ההסקה השתמשנו בdecoding המובנה של T5-Base שמממש Beam Search.

כדי לשפר את ביצועי המודל חקרנו את פונקציית הגי׳נרוט ואת התכונות השונות והייפר פרמטרים שלה (כגון num_beams, top_p, top_k, tempeture וכו׳) , אך לאחר שינויים רבים הגענו לתוצאה בה רק ה num_beams הינו הפרמטר שבאמת משפר את המודל ולקחנו את ערכו להיות 10 (אידיאלי מבחינת תוצאות ומבחינת זמן ריצה).

מבחן:

ציון bleu על קובץ הוולידציה – **40.74**

להלן ציוני bleu עבור סט המבחן:



תחרות:

החלק התחרותי התבטא במודל שלנו כאשר תחילה ניסינו להפריד את הקלט לשני encoders, אחד עבור הגרמנית ואחד עבור הroot and modifiers, ולבצע מניפולציות של attention בניהם, אך לצערנו מודל זה עבד הרבה פחות טוב מהמודל עם הקלט היחיד.

בנוסף עבור הכללה גדולה יותר יצרנו 9 קבצים שונים עבור קבצי האימון כך שיצרנו root and modifiers, עבור כל משפט באנגלית לפי התרגום של הקלט, וככה יצרנו קלט יותר אינפורמטיבי ומתאים לקובץ המבחן. משום שלכל משפט ישנם לרב יותר משני modifiers, החלטנו להוסיף רנדומיות לייצורם בקובץ כך שבכל קובץ יבחרו modifiers באופן רנדומלי וככה המודל ידע להכליל טוב יותר, וידע להתמודד עם מגוון רחב יותר של modifiers.

בנוסף עבור החלק התחרותי עשינו המון ניסויים של היפר פרמטרים כמוזכר בהסקה.

חלוקת העבודה:

חלוקת העבודה שהתבצעה הינה כך ששנינו תמיד עבדנו יחד, אך על משימות שונות במקביל. כלומר כל אחד היה על תת משימה משלו, אך בקשר רציף על התת משימה של השותף השני. כאשר נתקלנו בקשיים מיוחדים, בעיקר סביב הsyntax של ספריית Transformers, עבדנו במקביל בניסיונות שונים (אך מתואמים) להתגבר על הבעיה