דודי פריזדה 300592870
אלחנן משרקי 200086049

פרויקט בהבנת טקסטים
קורס 89565

תיאור מערכת

רכיבים בהם השתמשנו:

- 3.0WORDNET עם 1.6extJWNL.
  - עבור כל מילה לקחנו 6 רמות של hypernyms (מילים גוררות), אך בגלל בעיות של זמן ריצה וזיכרון נאלצנו להשתמש רק ב 3 רמות.

- DIRECT
  - עם 200 מילים נגררות לכל היותר לכל מילה. אך בגלל בעיות של זמן ריצה וזיכרון נאלצנו להשתמש רק ב20 מילים נגררות לכל היותר.

- Naïve
  - היה משאב נאיבי על מנת לאפיין את הדוגמאות והוא רק ציין שמילה גוררת את עצמה עם חלק דיבר זהה עם ציון גרירה 1.

- SVMLIGHT
  - עבור ההחלטה הסופית בדבר גרירה או אי גרירה של משפט נתון, אימנו מסווג SVM ונתנו לו להחליט עם poly kernel כאשר d=1.

קינפוג המערכת:

יצרנו לSVM וקטור מאפיינים.
באופן כללי ספרנו עבור כל משפט מועמד סכמנו לכל מילה בהיפוטזה את המילים שגררו את אותה המילה עם אותו חלק דיבר מתוך המשפט המועמד.
שהורכב מהמאפיינים הבאים: (כאשר עבור בדיקות הכריתה נכרתו גם המאפיינים המתאימים)

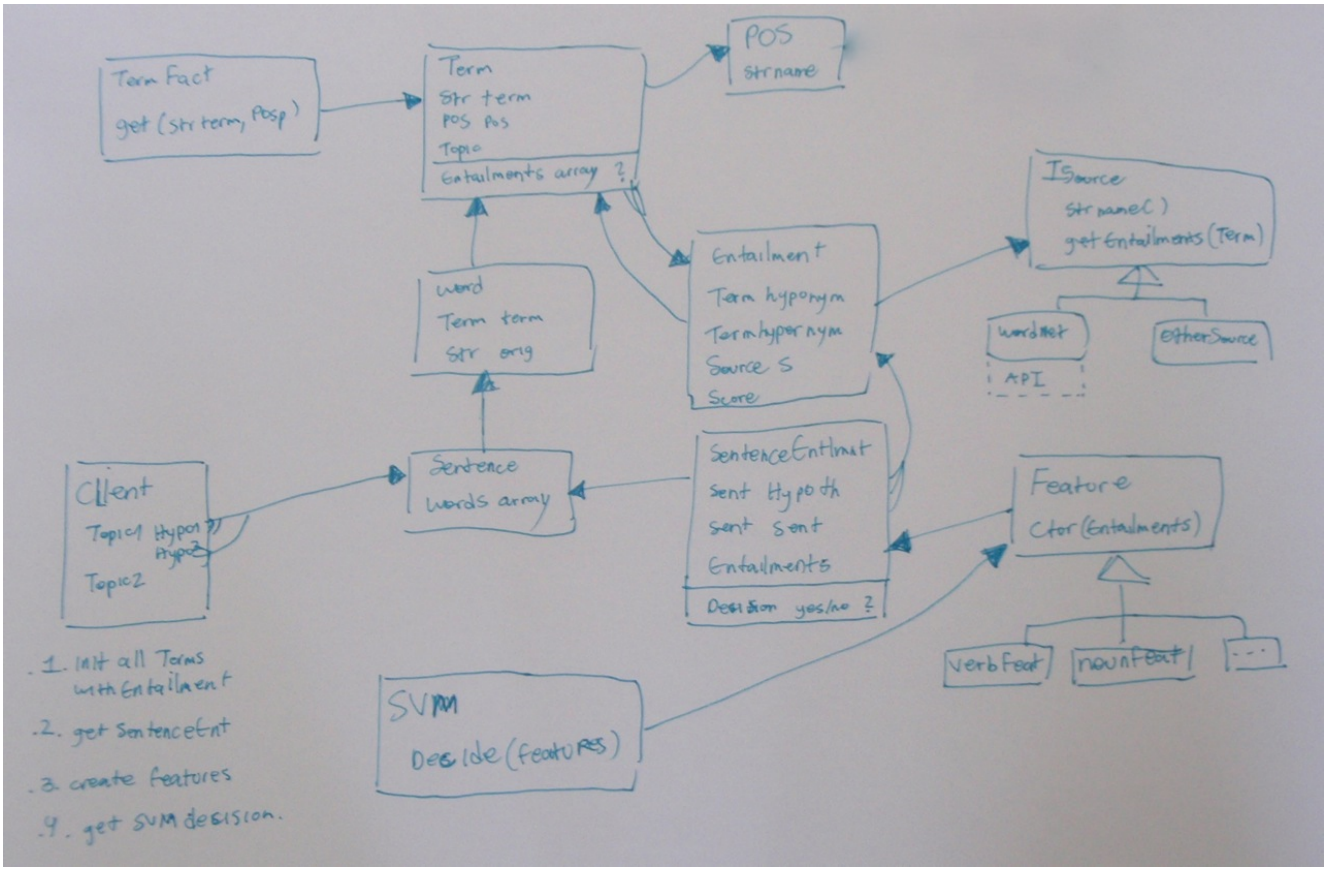
- NaiveNouneFeature סכימת סך הנקודות שנצברו עבור naïve כאשר חלק הדיבר הוא NOUN
- NaiveVerbFeature סכימת סך הנקודות שנצברו עבור naïve כאשר חלק הדיבר הוא VERB
- NaiveAdjectiveFeature סכימת סך הנקודות שנצברו עבור naïve כאשר חלק הדיבר הוא ADJECTIVE
- NaiveAdverbFeature סכימת סך הנקודות שנצברו עבור naïve כאשר חלק הדיבר הוא ADJVERB
- NaiveCarFeature סכימת סך הנקודות שנצברו עבור naïve כאשר חלק הדיבר הוא CAR
- NaiveOFeature סכימת סך הנקודות שנצברו עבור naïve כאשר חלק הדיבר הוא O
- NaivePFeature סכימת סך הנקודות שנצברו עבור naïve כאשר חלק הדיבר הוא P
- NaiveDFeature סכימת סך הנקודות שנצברו עבור naïve כאשר חלק הדיבר הוא D
- NaiveCFeature סכימת סך הנקודות שנצברו עבור naïve כאשר חלק הדיבר הוא C
- WordNetNounFeature סכימת סך הנקודות שנצברו עבור WORDNET כאשר חלק הדיבר הוא NOUN
- WordNetAdjectiveFeature סכימת סך הנקודות שנצברו עבור WORDNET כאשר חלק הדיבר הוא

ADJECTIVE

- WordNetVerbFeature סכימת סך הנקודות שנצברו עבור WORDNET כאשר חלק הדיבר הוא VERB
- WordNetAdverbFeature סכימת סך הנקודות שנצברו עבור WORDNET כאשר חלק הדיבר הוא ADJVERB
- DirectNounFeature סכימת סך הנקודות שנצברו עבור DIRECT כאשר חלק הדיבר הוא NOUN
- DirectVerbFeature סכימת סך הנקודות שנצברו עבור DIRECT כאשר חלק הדיבר הוא VERB
- TotalFeature סכימת סך הנקודות שנצברו עבור DIRECT כל האפשרויות
- TotalScorelessFeature סכימת סך ההתאמות שנצברו עבור DIRECT כל האפשרויות

כל ערכי המאפיינים נורמלו לפי ההקשר.

תרשים UML של המערכת



בדיקות כריתה

להלן ציון 1F ללא כריתה ועבור כריתת הרכיבים השונים:

ללא NAIVE	ללא WORDNET	ללא DIRECT	כולל הכל
0.0	10.810810810810812	11.267605633802816	14.56953642384106

אנליזה של קבצי הפלט:

בריצות הראשונות של המערכת לא הגענו לשום תוצאות, כלומר 1F = 0.
הסיבה ששערנו היא שהמאפיינים שהעברנו למסווג לא היו ייצוגיים מספיק כדי להבדיל בין דוגמאות חיוביות לשליליות.
מכאן שעשינו כמה ניסויים עם ספירה של גרירות ממרכיבים שונים של המשפט, לפי מקור הגרירה (/wordnet direct/ naive), ולפי ציון שנתנו מכל מערכת.
בסופו של דבר גילינו שהמאפיין החזק ביותר הוא דווקא הפשוט ביותר, שסופר מספר מילים נגררות בהיפותזה ללא התחשבות במקור הגרירה ובתיוג המילה.
בקבצי הפלט ניתן לראות בבירור שהnaive הוא המקור של רוב הגרירות, אבל גם משני המקורות הלקסיקליים, ניתן לראות שיש גרירות. בגלל מגבלות של זמן ריצה וזיכרון, היינו חייבים לחתוך את כמות הגרירות שכל מערכת החזירה, ובכל זאת, בwordnet ככל שעולים בעץ הגרירות שלו, המילה הגוררת מאבדת כמעט כל הקשר למילה הנגררת, למשל המילה dog גוררת לפי wordnet את המילה object (כמעט כל noun גורר object), ואין סיכויי כמעט שגרירה כאת תועיל, ולכן חתכנו את הגרירות ב3 רמות מהעלה.
הגילוי שרוב הגרירות הנכונות נובעות מהגרירה הנאיבית, הוא די ברור, כי ברוב המקרים בהיפותיזה ובמשפט הגורר יש די הרבה מילים/ למות זהות.
ההפרדה שיצרנו בין הגרירות הנאיביות לבין בגרירות משני המקורות האחרים, הייתה רעיון טוב, שכן זכינו לראות עד כמה אנחנו מצליחים לשפר מערכת נאיבית עם מקורות מתוחכמים.

שיפורים אפשריים:

המערכת בנוייה בצורה של פלאגינים, כלומר ניתן להוסיף ולהוריד חלקים יחסית בקלות. על מנת לשפר את תוצאות המערכת, היינו פותחים במחקר שמטרתו למצוא אילו מאפיינים מייצגים דוגמאות טובות של משפטים שגוררים היפותיזה, ובמה הם שונים ממשפטים שלא גוררים את ההיפותיזה.
בנוסף אפשר להוסיף עוד מקורות לקסיקליים יחסית בקלות.

בתכנון המקורי היינו שומרים עבור כל מילה בהיפותיזה, את כל המילים שעשויות לגרור אותה. אפשרות נוספת לשיפור המערכת שביצענו הייתה למצוא רק את הגרירה הטובה ביותר עבור כל מילה בהיפותיזה, ולשכוח את כל יתר הגרירות. באמצעות השיפור על ידי סינון הגרירות הפחות טובות, חשבנו שנוכל ליצור ווקטור מאפיינים איכותי יותר אך בפועל, מצאנו שאין הבדל בתוצאות. הסיבה לכך הייתה שלרוב המילים בהיפותיזה, הייתה בלאו הכי רק גרירה אחת ולכן לא הייתה השפעה לאלגוריתם הנ"ל.

במערכת כרגע אנחנו מחלקים ציונים לגרירות באופן שרירותי. ניתן היה לממש, מנגנון שנותן עדיפות לגרירות מסוימות לפי פרמטרים ובכך לשפר את הביצועים - יצירת ווקטור מאפיינים איכותי יותר.