Amirkabir University of Technology
(Tehran Polytechnic)

Department of
Computer Engineering

Natural Language Processing

Homework 3

Najmeh Mohammadbagheri

99131009

گزارش تمرین

بخش اول

بارگیری داده‌ها: در این قسمت دیکشنری که یک فایل xml بود با کتابخانه‌ی etree پارس شد و به فرمت دیکشنری در پایتون ذخیره شد. همچنین فایل‌ها آموزش و آزمون با فرمت مناسب در پایتون ذخیره شدند. قبل از ذخیره‌ی این دو فایل، یک سری پیش‌پردازش‌ها برروی بافت انجام شد.

پیش‌پردازش داده‌ها:

در این قسمت ابتدا تمام حروف به شکل کوچک تبدیل شدند، سپس علائم نگاریشی و اعداد و کلمات پرتکرار از آن حذف شدند.

برای انجام این قسمت از کتابخانه‌ی nltk استفاده شد.

همچنین لازم به ذکر است که این پیش پردازش بر روی تمام بافت انجام نشد و تنها بر روی ۹ کلمه بعد و قبل کلمه‌ی مبهم انجام شد. زیرا با توجه به چالش‌هایی که برای زمان اجرای برنامه داشتیم، گفته شد حداکثر با پنجره‌ی ۹ بافت را بررسی کنیم.

پس از انکه پیش‌پردازش بر روی داده‌ها انجام شد این فایل‌ها به شکل دیکشنری و به ترتیب کلمه و مفهوم ذخیره‌شد. (زیرا هربار اجرای از اول آن زمان‌بر بود.)

بخش دوم

در این بخش همانطور که در صورت تمرین گفته شده بود بردارها استخراج شدند که در ادامه به صورت مختصر هر کدام را توضیح می‌دهیم.

۱) در این قسمت بافت کلمه‌ی مبهم به برت داده شد و بردار خروجی تنها برای کلمه‌ی مبهم گرفته شد. این بردار ابعادش ۷۶۸ بود و لازم بود که به ۳۰۰ کاهش یابد. بدین منظور تمام بردارها استخراج شده در داده‌های آموزش و آزمون به PCA داده شد و خروجی کاهش یافته گرفته شد.

۲) این حالت نیز مانند حالت قبل تمام بافت(۱۹کلمه) را به برت می‌دهیم ولی این بار بردار کلمه‌ی مبهم و سه کلمه‌ی اطراف آن را میگیریم. بعد از آن از یک مدل tf_idf استفاده می‌کنیم و به ازای هر کلمه وزن آنرا گرفته و با یک میانگین وزن‌دار بر روی این ۷ بردار، یک بازنمایی برای کلمه‌ی مبهم می‌سازیم.

در این حالت نیز مانند حالت قبل از PCA استفاده می‌کنیم. همچنین برای ساخت مدل tf_idf از کتابخانه‌ی Gensim کمک می‌گیریم.(همانند کاری که در تمرین قبل انجام دادیم انجام می‌دهیم)

۳) این حالت نیز کاملا شبیه به حالت قبل است با این تفاوت که بجای ۳ کلمه‌ی همسایه، ۹ کلمه‌ی همسایه یا به عبارتی کل بافت، از برت گرفته می‌شود و بازنمایی بر اساس آن ۱۹ کلمه ساخته می‌شود.

در این سه حالت به دلیل زیاد بودن زمان اجرا، در هر سه مرحله بردارهای خروجی در فایل ذخیره می‌شوند.

ساخت مدل tf_idf با یک دیکشنری بر روی تمام کلمات متون و مدل tf_idf جداگانه برای هر مفهوم انجام می‌شود.

۴) در این حالت بردارها از مدل ورد۲وک گرفته می‌شوند که نیاز به پردازش خاصی ندارد. یعنی بردار کلمه‌ی مبهم با ۳ همسایه‌هایش به صورت وزن‌دار با وزن tf_idf میانگین گرفته می‌شود و بردار بازنمایی حاصل می‌شود. در این حالت چون زمان اجرا زیاد نیست، خروجی‌ها در فایل ذخیره نمی‌شوند و هر بار از اول اجرا می‌شود.

۵) این حالت نیز کاملا شبیه به حالت قبل است با این تفاوت که همسایگی ۹ می‌باشد.

در ادامه از این بازنمایی‌ها استفاده می‌کنیم تا دسته‌ی هر مفهوم مبهم را پیدا کنیم.

بخش سوم

الف)

تنظیمات مدل لاجیستیک: max_iter=20000,C=0.08

تنظیمات مدل جنگل تصادفی: n_estimators=150,max_depth=100

آزمایش	دقت	f-score
برت بدون بافت + لاجیستیک	فعل : ۰.۳۸ اسم: ۰.۳۱ صفت: ۰.۳۲	فعل: ۰.۳۸ اسم: ۰.۳۱ صفت: ۰.۳۲
برت بدون بافت + جنگل تصادفی	فعل : ۰.۵۲ اسم: ۰.۶۰ صفت: ۰.۳۹	فعل: ۰.۵۲ اسم: ۰.۶۰ صفت: ۰.۳۹
برت با پنجره ۳ + لاجیستیک	فعل : ۰.۳۶ اسم: ۰.۳۰ صفت: ۰.۳۲	فعل: ۰.۳۶ اسم: ۰.۳۰ صفت: ۰.۳۲
برت با پنجره ۳ + جنگل تصادفی	فعل : ۰.۴۵ اسم: ۰.۵۲ صفت: ۰.۴۴	فعل: ۰.۴۵ اسم: ۰.۵۲ صفت: ۰.۴۴
برت با پنجره ۹ + لاجیستیک	فعل : ۰.۳۷ اسم: ۰.۲۹ صفت: ۰.۳۲	فعل:۰.۳۷ اسم:۰.۲۹ صفت: ۰.۳۲
برت با پنجره ۹ + جنگل تصادفی	فعل : ۰.۴۸ اسم: ۰.۵۲ صفت:۰.۳۹	فعل: ۰.۴۸ اسم: ۰.۵۲ صفت: ۰.۳۹
ورد۲وک با پنجره ۳ + لاجیستیک	فعل : ۰.۴۵	فعل: ۰.۴۵

	اسم: ۰.۵۰ صفت: ۰.۳۹	اسم: ۰.۵۰ صفت: ۰.۳۹
ورد۲وک با پنجره ۳ + جنگل تصادفی	فعل : ۰.۵۱ اسم: ۰.۵۴ صفت: ۰.۴۳	فعل: ۰.۵۱ اسم: ۰.۵۴ صفت: ۰.۴۳
ورد۲وک با پنجره ۹ + لاجیستیک	فعل : ۰.۴۵ اسم: ۰.۵۰ صفت: ۰.۳۹	فعل: ۰.۴۵ اسم: ۰.۵۰ صفت: ۰.۳۹
ورد۲وک با پنجره ۹ + جنگل تصادفی	فعل : ۰.۵۳ اسم: ۰.۵۵ صفت: ۰.۴۳	فعل: ۰.۵۳ اسم: ۰.۵۵ صفت: ۰.۴۳

سه بهترین حالت:

۱. برت بدون بافت + جنگل تصادفی

۲. ورد۲وک با پنجره ۳ + جنگل تصادفی

۳. ورد۲وک با پنجره ۹ + جنگل تصادفی

برای قسمت بعد از این سه حالت و دسته‌بند گروهی استفاده می‌شود.

ب)

سه دسته‌بند جنگل تصادفی با داده‌های مختلف(قسمت قبل ذکر شد) آموزش می‌دهیم و سپس بین نظرات آنها رای‌گیری می‌کنیم. دقت‌ها در جدول زیر ذکر شده است.

نقش کلمه	دقت	f-score
فعل	۰.۵۲	۰.۵۲
اسم	۰.۵۶	۰.۵۶
صفت	۰.۳۵	۰.۳۵

ج)

در این قسمت از خروجی سه دسته‌بند به عنوان یک بازنمایی برای هر داده استفاده می‌کنیم. یعنی هر داده با یک بردار ۳ بعدی نمایش داده می‌شود. سپس این داده‌های سه بعدی را به یک دسته‌بند می‌دهیم تا نتیجه را بررسی کنیم.

به دلیل اینکه بهترین دسته‌بند در قسمت اول، جنگل تصادفی بود، در این قسمت نیز از جنگل تصادفی استفاده می‌کنیم. دقت‌ها در جدول زیر ذکر شده است.

نقش کلمه	دقت	f-score
فعل	۰.۵۲	۰.۵۲
اسم	۰.۵۶	۰.۵۶
صفت	۰.۳۵	۰.۳۵

بخش چهارم

تحلیل:

در بخش قبل دیدیم که برت بدون کلمات اطراف بهتر از دو حالتی بود که کلمات اطراف را میگرفتیم. از این مشاهده می‌توان نتیجه گرفت کلمات اطراف کلمه‌ی مبهم در دادگان تست، برای بازنمایی با برت گمراه‌کننده بوده‌اند و این امر باعث پایین آمدن دقت در حالت‌های پنجره‌ای شده است.

همچنین مشاهده شد که هر دو حالت مدل ورد۲وک بهتر از همین حالت‌ها در مدل برت بود. از این مشاهده نیز نتیجه می‌گیریم مجموعه‌ی داده‌های آموزشی استفاده شده در مدل ورد۲وک به داده‌های تست ما نزدیکتر بوده‌اند (حدس). همچنین یک دلیل بسیار مهم دیگر این است که در مدل برت بردارها را کاهش بعد دادیم؛ یعنی حدودا نصف کردیم و این به معنای حذف میزان زیادی از اطلاعات است. به همین دلیل عملکرد مدل برت پایین‌تر از مدل ورد۲وک بود (فکت).

یک مشاهده‌ی دیگر این بود که در دسته‌بندها دقت دسته‌بند جنگل تصادفی بهتر از دسته‌بند لاجیستیک بود. همانطور که میدانیم جنگل تصادفی یک روش دسته‌بندی ensemble است و این دسته‌بندها دقت بسیار بالایی دارند.

البته این نکته نیز قابل ذکر است که هر دو دسته‌بند باید پارامترهای بهینه‌شان پیدا شود و ممکن است در این آزمایشات، بنده به پارامترهای بهینه نرسیده باشم و با توجه به بهترین پارامترهایی که بدست آوردم جنگل تصادفی بهتر بوده باشد.