دانشگاه شهید بهشتی
دانشکده مهندسی و علوم کامپیوتر

پروژه‌های نهایی درس پردازش زبان طبیعی

مقایسه عملکرد جاسازی‌های مبتنی بر بافتار و ایستا در آزمون قیاس و بررسی اثر ریزتنظیم مدل‌های مبتنی بر بافتار بر روی آن (پروژه اول و دوم)

کارشناسی ارشد مهندسی کامپیوتر
گرایش هوش مصنوعی، رباتیکز و رایانش شناختی

دانشجو:
سیدسروش مجد

استاد درس:
خانم دکتر مهرنوش شمس‌فرد

بهمن ۱۴۰۱

فهرست مطالب

شماره صفحه

۱- مقدمه

استفاده از تکنیک‌های پردازش زبان طبیعی (NLP) در سال‌های اخیر اهمیت فزاینده‌ای پیدا کرده است، زیرا ماشین‌ها را قادر می‌سازد تا انسان‌ها را به روشی طبیعی‌تر درک کنند و با آن‌ها تعامل داشته باشند. یکی از پرکاربردترین روش‌هایی که پردازش زبان طبیعی و کارهای مختلف استفاده می‌شود، جاسازی است که فرآیند نمایش کلمات، عبارات و جملات به عنوان بردار عددی در فضایی با ابعاد بالا می‌باشد. در این پروژه، دو نوع اصلی جاسازی ایستا و مبتنی بر بافتار را بررسی خواهیم کرد و عملکرد آنها را در آزمون قیاس تجزیه و تحلیل می‌کنیم. همچنین اثرات ریزتنظیم یک مدل تعبیه متنی را بر روی مجموعه‌داده استنتاج زبان طبیعی (NLI) به زبان فارسی بررسی خواهیم کرد. در نهایت، یافته‌ها و نتایج به دست آمده را مورد بحث قرار داده و در مورد اثربخشی و دقت این روش‌ها برای آزمون قیاس نتیجه‌گیری خواهیم کرد.

۲- جاسازی‌های ایستا

برای آزمون قیاس با استفاده از جاسازی‌های ایستا، مدل‌هایی مانند GloVe، Fasttext و word2vec سه کلمه را به عنوان ورودی می‌گیرد و نزدیک‌ترین کلمه چهارم را با استفاده از فاصله بین دو کلمه اول و دوم پیدا می‌کند. این کار با یافتن بردار جاسازی برای کلمه چهارم انجام می‌شود به شکلی که بردار فاصله جاسازی بردار چهارم و بردار جاسازی سوم نزدیک ترین حالت ممکن به بردار فاصله بردار جاسازی کلمه اول و کلمه دوم باشد. نزدیکی فاصله‌ها نیز با شباهت کسینوسی اندازه‌گیری می‌شود. در انگیسی نشان داده شده است که این روش در گرفتن روابط معنایی بین کلمات موثر است و برای زبان فارسی در ادامه آزمایش می‌شود. ا از ابزار dadmatools برای دسترسی به این جاسازی‌ها استفاده کردیم.

۲-۱ جاسازی GloVe

جاسازی GloVe (Global Vectors for Word Representation) یک الگوریتم یادگیری بدون نظارت برای به دست آوردن جاسازی کلمات است. GloVe توسط محققان دانشگاه استنفورد در سال ۲۰۱۴ ارائه شده است و تعدادی مزیت نسبت به روش‌های ساده‌تر و مبتنی بر شمارش مانند word2vec دارد. بردارهای GloVe توسط یک پیکره بزرگ و هم‌رخدادی کلمات آموخته می‌شوند و می‌توانند قاعده‌های معنایی و نحوی و را در هر زبانی در بردارهای کلمات ارائه کنند. این بردارها بر روی مجموعه بزرگی از متون، مانند ویکیپدیا آموزش داده شده‌اند و می‌توانند برای نمایش کلمات در هر زبانی مانند فارسی استفاده شوند. این بردارها در ابعاد مختلف (۵۰، ۱۰۰، ۲۰۰، ۳۰۰) موجود هستند.

۲-۱-۱ نتایج دقت GloVe (Accuracy top-1)

city	capital	currency	family	gram_thirdperson	gram_past	gram_firstperson
0.2	0.2	0.2	0.2	0.3	0.1	0.0

gram_adj2adv	gram_noun2adv	gram_antonym	gram_comparative	gram_nationality	gram_plural
0.0	0.0	0.2	0.2	0.4	0.0

برای هر کدام از کتگوری ها تعداد نزدیک‌ترین بردارها به بردار جاسازی کلمه سوم متفاوت است و بر اساس آن دقت هر کتگوری زیاد یا کم میشد.

۲-۲ جاسازی word2vec

Word2Vec یک الگوریتم یادگیری بدون نظارت است که توسط محققان گوگل در سال ۲۰۱۳ توسعه یافته که جاسازی برای کلمات را از پیکره‌های بزرگ می‌آموزد. برخلاف GloVe که از آمار هم‌رخدادی برای یادگیری بازنمایی‌های برداری کلمات استفاده می‌کند، Word2Vec از یک معماری شبکه عصبی کم عمق برای یادگیری جاسازی کلمات از پنجره‌های متون اطراف هر کلمه در یک پیکره استفاده می‌کند. جاسازی‌های Word2Vec می‌توانند برای نمایش کلمات در هر زبانی استفاده شوند. این بردارها نیز در ابعاد مختلف (۵۰، ۱۰۰، ۲۰۰، ۳۰۰) وجود دارند.

۲-۲-۱ نتایج word2vec (Accuracy top-1)

city	capital	currency	family	gram_thirdperson	gram_past	gram_firstperson
0.1	0.1	0.0	0.1	0.8	0.1	0.3

gram_adj2adv	gram_noun2adv	gram_antonym	gram_comparative	gram_nationality	gram_plural
0.0	0.0	0.3	0.5	0.2	0.0

برای هر کدام از کتگوری ها تعداد نزدیک‌ترین بردارها به بردار جاسازی کلمه سوم متفاوت است و بر اساس آن دقت هر کتگوری زیاد یا کم میشد.

۲-۳ جاسازی Fasttext

FastText نیز اگوریتمی با یادگیری بدون نظارت است که توسط محققان فیس‌بوک در سال ۲۰۱۶ ایجاد شده است. بر خلاف Word2Vec که نمایش های برداری را فقط برای کل کلمات می آموزد، FastText می تواند نمایش هایی را برای

زیرکلمه ها یا ngram های کاراکتر بیاموزد که به آن اجازه می‌دهد اطلاعات بیشتری در مورد کلمات نادر یا دیده نشده نسبت به روش‌های سنتی مانند GloVe یا Word2Vec که فقط بر آمار وقوع همزمان متکی هستند، بگیرد. تعبیه‌های FastText در مجموعه‌های بزرگی مانند ویکی‌پدیا از قبل آموزش داده شده‌اند و می‌توانند برای نمایش زیرکلمه‌ها یا ngram کاراکترها در هر زبانی استفاده شوند. وکتورهای از پیش آموزش دیده در ابعاد مختلف (۵۰، ۱۰۰، ۲۰۰) موجود می باشند.

۲-۳-۱ نتایج Fasttext (Accuracy top-1)

city	capital	currency	family	gram_thirdperson	gram_past	gram_firstperson
0.2	0.4	0.0	0.3	0.6	0.3	0.6

gram_adj2adv	gram_noun2adv	gram_antonym	gram_comparative	gram_nationality	gram_plural
0.0	0.1	0.3	0.2	0.6	0.0

برای اجرای کد Fasttext از high ram در کلب استفاده شده است و بدون آن احتمالا اجرا کردن ممکن نباشد. مشاهده شد که اگر تعداد نزدیک‌ترین همسایه‌ها به کلمه سوم خیلی زیاد باشد دقت این مدل کاهش می‌یابد. علت دقت بیشتر برای nationality یا firstperson و thirdperson به دلیل این است که این مدل بر اساس کاراکتر بردارها را بازنمایی می-کند.

۳- جاسازی‌های مبتنی بر بافتار

پیش‌بینی کلمات ماسکه شده روشی است که برای ارزیابی عملکرد جاسازی‌های متنی بافتاری مانند BERT و مبدل‌هایی مانند آن استفاده می‌شود. در این روش کلماتی خاص در یک جمله ماسکه شده و با استفاده از مبدل‌ها آن‌ها را با توجه به متن و بافتاری که در آن قرار دارند پیش‌بینی می‌کنند و سپس مدل بر اساس دقت آن در پیش‌بینی کلمه صحیح ارزیابی می‌شود. در این آزمایش، از پیش‌بینی کلمات شده شده برای ارزیابی عملکرد مبدل‌ها در آزمون قیاس استفاده کردیم. آزمون قیاس شامل مجموعه ای از سوالات است که کاربر را ملزم می‌کند تا روابط بین کلمات یا عبارات را شناسایی کند. به عنوان مثال، با توجه به عبارت "نسبت شاه به ملکه مانند نسبت آقا به ____ است"، پاسخ صحیح "خانم" خواهد بود. برای ارزیابی مبدل‌ها در این کار، از دو روش استفاده کردیم. در روش اول عبارات خاصی برای انواع خاصی از کلمات تولید کردیم و کلمه‌های چهارم را در هر عبارت ماسکه کردیم. سپس با مبدل‌ها آن کلمات را پیش‌بینی کردیم و در نهایت دقت هر مدل با

مقایسه پیش‌بینی‌های آن با پاسخ‌های صحیح اندازه‌گیری شد. در این روش عبارات متعددی آزمایش شد تا بیشترین دقت به دست آید. در روش دوم ابتدا رابطه بین دو کلمه اول و دوم را با استفاده از مبدل‌ها تعیین کردیم و با استفاده از آن برای تخمین کلمه چهارم استفاده کردیم. به بیان دیگر کلمه‌ای را به عنوان کلمه چهارم که ماسکه شده انتخاب کردیم که نسبت آن به کلمه سوم برابر با رابطه پیشبینی شده بین کلمات اول و دوم باشد. برای مثال اگر رابطه کلمه اول و دوم توسط عبارتی مانند "تهران ___ ایران است"، پایتخت به دست آمد، کلمه چهارم را از طریق پیشبینی کلمه ماسکه شده در عبارت واشنگتن پایتخت ___ است"، که جواب صحیح آمریکا است. تابع predict_masked_word جمله را به عنوان ورودی می‌گیرد و MASK موجود در آن را با مدل زبانی تخمین می‌زند. این تابع سه تا از نزدیک‌ترین پیشنهادات برای MASK را نیز چاپ می‌کند.

تعدادی از عبارات تولید شده در روش اول: (برای هر ۱۳ کتگوری عبارت تولید شده است).

City: شهر word1 در استان word2 در نتیجه شهر word3 در استان [MASK] می باشد.
Capital: شهر word1 پایتخت کشور word2 است پس شهر [MASK] پایتخت کشور word4 است.
Family: نسبت خانوادگی word1 به word2 مانند نسبت خانوادگی word3 به [MASK] است.
gram_thirdperson: فعل word1 سوم شخص مفرد و فعل word2 سوم شخص جمع است در نتیجه فعل word3 سوم شخص مفرد و فعل [MASK] سوم شخص جمع آن است.

gram_firstperson: من word3 پس ما [MASK] من word1 و ما word2

چالش اصلی در روش دوم پیدا کردن رابطه درست بین کلمه اول و دوم است.

۳-۱- مدل ParsBERT (HooshvareLab/bert-fa-base-uncased)

۳-۱-۱ دقت در روش اول: (Accuracy top-1)

city	capital	currency	family	gram_thirdperson	gram_past	gram_firstperson
0.8	0.8	0.1	0.4	0.6	0.0	0.0

gram_adj2adv	gram_noun2adv	gram_antonym	gram_comparative	gram_nationality	gram_plural
0.1	0.0	0.0	0.0	0.3	0.1

۳-۱-۲- دقت در روش دوم (Accuracy top-1)

city	capital	currency	family	gram_thirdperson	gram_past	gram_firstperson
0.6	0.7	0.1	0.0	0.0	0.0	0.0

gram_adj2adv	gram_noun2adv	gram_antonym	gram_comparative	gram_nationality	gram_plural
0.0	0.0	0.0	0.0	0.2	0.00

در روش دوم پیدا کردن رابطه بین کلمه اول و دوم چالش جدیدی است. برای مثال در اسلام‌شهر در [MASK] تهران است. به جای آن که سیستم تشخیص دهد کلمه ماسکه شده استان است، لمه ماسکه شده را جنوب تشخیص داده است. ولی در کل دقت برای city و capital مناسب است ولی برای افعال بسیار پایین می‌باشد و مدل توانایی تشخیص آن‌ها را ندارد. در روش اول نیز توانایی تشخیص افعال وجود نداشت.

۳-۲- مدل m-BERT (bert-base-multilingual-uncased)

۳-۲-۱- دقت در روش اول (Accuracy top-1)

city	capital	currency	family	gram_thirdperson	gram_past	gram_firstperson
0.2	0.0	0.0	0.1	0.0	0.0	0.0

gram_adj2adv	gram_noun2adv	gram_antonym	gram_comparative	gram_nationality	gram_plural
0.0	0.0	0.0	0.0	0.2	0.0

مدل m-bert در روش اول دقت در همه کتگوری‌ها خیلی پایینی داشت.

۳-۲-۲ دقت در روش دوم (Accuracy top-1)

city	capital	currency	family	gram_thirdperson	gram_past	gram_firstperson
0.1	0.0	0.1	0.0	0.0	0.0	0.0

gram_adj2adv	gram_noun2adv	gram_antonym	gram_comparative	gram_nationality	gram_plural
0.0	0.0	0.0	0.0	0.0	0.0

مدل m-bert توانایی تشخیص درست روابط و در نتیجه کلمات چهارم را نداشت ولی جواب‌های خیلی پرت نمی‌داد. به این معنی که جواب‌هایی که می‌داد در حوزه‌های کتگوری بودند ولی درست نبودند.

۴- عملکرد جاسازی‌های مبتنی بر بافتار در آزمون قیاس بعد از ریزتنظیم آن‌ها با مجموعه‌داده NLI فارسی (FarsTail)

به نظر می‌رسد که ریزتنظیم مبدل‌ها بر روی مجموعه داده‌های استنتاج زبان طبیعی (NLI) می تواند عملکرد آن را در آزمون قیاس بهبود بخشد. این به این دلیل است که مجموعه‌داده NLI شامل تعداد زیادی جفت جملات است که مدل برای آن‌که بتواند به طور دقیق آن‌ها را دسته‌بندی بندی کند نیاز دارد تا رابطه بین دو جمله را درک کند. مدل با ریزتنظیم یک مبدل مانند BERT بر روی NLI مدل می‌تواند یاد بگیرد که روابط بین کلمات را بهتر درک کند و آنها را در آزمون‌های قیاس به کار گیرد. آزمون های قیاس به مدلی نیاز دارند تا روابط بین کلمات یا عبارات را شناسایی کند تا مشخص شود کدام کلمه یا عبارت به بهترین شکل یک قیاس معین را کامل می‌کند. به عنوان مثال، اگر این آزمون قیاس را در نظر بگیریم: "گربه میو می‌کند همانطور که سگ ____ می‌کند"، مدل باید بتواند تشخیص دهد که گربه و سگ هر دو حیوان هستند و میو کردن یک عمل مرتبط با گربه است و در نتیجه پارس کردن مرتبط با سگ است. با ریزتنظیم مدل BERT بر روی داده‌های NLI، مدل می‌تواند این نوع روابط را یاد بگیرد و آن‌ها را هنگام آزمون قیاس اعمال کند. علاوه بر این، مجموعه داده‌های NLI شامل انواع ساختار جمله و الگوهای زبانی است که می‌تواند به مدل کمک کند تا زبان را بهتر درک کند. این درک بهتر از زبان می‌تواند به تشخیص بهتر روابط بین کلمات یا عبارات در آزمون قیاس کمک کند. به طور کلی، ریزتنظیم یک مدل روی مجموعه داده NLI می‌تواند عملکرد آن را در آزمون‌های قیاس به کمک درک بهتر زبان طبیعی و شناسایی روابط بین کلمات یا عبارات زبانی بهبود بخشد. در استنتاج زبان طبیعی هدف تعیین رابطه استنتاج بین یک فرض p و یک فرضیه h است. این یک تسک سه کلاسه است که در آن هر جفت (p, h) به یکی از این کلاس‌ها اختصاص داده می‌شود: "ENTAILMENT" اگر بتوان فرضیه را از فرض استنباط کرد، "CONTRADICTION" اگر فرضیه با فرض مخالف است و "NEUTRAL" اگر هیچ یک از موارد فوق صدق نکند. مجموعه داده های بزرگی مانند SNLI، MNLI و SciTail برای NLI در زبان انگلیسی وجود دارد، اما مجموعه داده های کمی برای زبان های با داده های ضعیف مانند فارسی وجود دارد. ما از اولین مجموعه داده فارسی نسبتاً بزرگ را برای کار NLI به نام FarsTail استفاده می‌کنیم که در مجموع ۱۰۳۶۷ نمونه از مجموعه ۳۵۳۹ سوال چند گزینه‌ای تولید شده است. بخش‌های آموزش، ولیدیت و آزمایش به ترتیب شامل

۷۲۶۶، ۱۵۳۷ و ۱۵۶۴ نمونه می‌شوند. NLI همان مسئله Classification سه کلاسه است. ابتدا به مدل‌ها یک لایه سه نورونه اضافه کرده و عملیات ریزتنظیم NLI را انجام دادیم. سپس از وزن‌های Encoderهای به دست آمده در آزمون قیاس استفاده کردیم. (با دستور trained_bert.bert.encoder.layer=model.bert.encoder.layer وزن‌های انکودر model که مدل آموزش دیده است را برابر با وزن‌های انکودر برت‌ها قرار دادیم). برای ساختن مجموعه‌داده برای آموزش مدل نیز از کلاسی که از torch.utils.data.Dataset ارث‌بری کرده است استفاده شده است.

جزئیات ریزتنظیم:

Sentences Max-Length = 32

Batch Size = 16

Epochs = 5

With Early Stopping

Choosed the best model in all epochs (`load_best_model_at_end=True`)

۴-۱- نتایج آزمون قیاس بعد از ریزتنظیم با NLI

۴-۱-۱- نتایج ParsBERT

دقت با روش اول:

city	capital	currency	family	gram_thirdperson	gram_past	gram_firstperson
0.0	0.0	0.0	0.0	0.0	0.0	0.0

gram_adj2adv	gram_noun2adv	gram_antonym	gram_comparative	gram_nationality	gram_plural
0.0	0.0	0.0	0.0	0.0	0.0

دقت با روش دوم:

city	capital	currency	family	gram_thirdperson	gram_past	gram_firstperson
0.3	0.1	0.0	0.0	0.0	0.0	0.0

gram_adj2adv	gram_noun2adv	gram_antonym	gram_comparative	gram_nationality	gram_plural
0.0	0.0	0.0	0.0	0.0	0.0

۴-۱-۲- نتایج m-BERT

دقت برای هر دو روش صفر است. بعد از ریزتنظیم این مدل با NLI فارسی حتی جواب‌ها به زبان فارسی نبودند و کاملا پرت بودند.

۵- نتیجه‌گیری

به طور کلی، این پژوهش نشان داد مبدل‌ها که جاسازی مبتنی بر بافتار تولید می‌کنند در آزمون قیاس لزوما بهتر از جاسازی-های ایستا عمل نکردند و این نشان می‌دهد که جاسازی‌های مبتنی بر بافتار حتی با اینکه قادر به پیش‌بینی دقیق روابط بین کلمات یا عبارات با توجه به بافتار جملات هستند، برای کارهای مختلف پردازش زبان طبیعی مانند آزمون قیاس در همه موارد و موضوعات مختلف لزوما نمی‌توانند از جاسازی‌های ایستا مفیدتر باشند. همچنین نتایج بعد از ریزتنظیم مبدل‌ها نشان می‌دهد مجموعه‌داده ParsTail برای آزمون قیاس فارسی مناسب نیست. تمامی کدها و جواب‌ها در نوت‌بوک‌های ضمیمه شده موجود هستند و برای اجرا باید از اول تا آخر هر cell به ترتیب اجرا شوند.