به نام خدا

دانشگاه تهران

دانشکدگان فنی

دانشکده مهندسی برق و کامپیوتر

درس پردازش زبان طبیعی

تمرین سوم

فروردین ماه ۱۴۰۳

فهرست

سوال اول

در پردازش زبان طبیعی برچسب گذاری نقش معنایی(Semantic Role Labeling) که به اختصار SRL هم گفته می‌شود به تخصیص برچسب[1] به کلمات یا عبارت‌های یک جمله با توجه به نقش معنایی[2] آن‌ها در جمله گفته می‌شود. در این مسئله جمله به همراه گزاره[3] یا فعل جمله داده می‌شود و باید با استفاده از آن‌ها شناسه‌های[4] مرتبط به آن فعل را تشخیص بدهیم و برچسب گذاری کنیم. این شناسه‌ها به طور معمول شامل کنشگر[5]، کنش‌پذیر[6]، ابزار[7] و همچنین سایر شناسه‌ها مانند زمان و مکان انجام عمل فعل هستند. شناسایی و برچسب گذاری این شناسه‌ها یک وظیفه کلیدی برای فهم عمیق‌تر معنا و ساختار جمله‌ها است و در مسئله‌هایی مثل بازیابی اطلاعات، پرسش و پاسخ، خلاصه سازی استفاده می‌شود.

برای درک بهتر جمله زیر که فعل آن accept است داده شده است:

He wouldn’t accept anything of value from those he was writing about.

حال می‌خواهیم آن را برچسب گذاری معنایی کنیم:

[Arg0 He] wouldn’t accept [Arg1 anything of value] from [Arg2 those he was writing about].

این نشانه گذاری با توجه به ProBank که منبعی برای برچسب گذاری نقش معنایی است انجام شده است. که Arg0 همان شناسه کنشگر، Arg1 شناسه کنش‌پذیر و Arg2 شناسه ابزار یا ذینفع[8] هستند.

در این تمرین شناسه‌های مورد استفاده سه شناسه ذکر شده به همراه شناسه ArgM-Loc و ArgM-TMP هستند. که ArgM-LOC برای مکان فعل و ArgM-TMP برای زمان انجام فعل استفاده می‌شود. در این سوال به شما جملات به همراه فعل یا گزاره آن‌ها داده می‌شود و شما باید با کمک شبکه‌های عصبی بازگشتی این پنج شناسه را شناسایی کنید.

مجموعه داده

مجموعه داده شامل سه فایل train، valid و test با فرمت json است. که هرکدام چهار فیلد دارند:

۱.text: شامل جملات است که برای هر جمله لیستی از توکن‌های آن را دارد.

۲.verb_index: مکان قرارگیری فعل در جمله است. توجه کنید که اولین کلمه در مکان صفرم در نظر گرفته می‌شود.

۳.srl_label: نشان دهنده برچسب هر توکن در مسئله SRL است.

۴. word_indices: ایندکس توکن‌ها را نشان می‌دهد.

[1] Label
[2] Semantic Role
[3] Predicate
[4] Arguments
[5] Agent
[6] Patient
[7] Instrument
[8] Beneficiary

بخش اول: آماده کردن مجموعه داده

ابتدا مجموعه داده را با کمک کتابخانه json بارگذاری کنید. این مجموعه داده شامل چهار فیلد می‌باشد که برای آشنا شدن با آن ابتدا برای دومین جمله Training این چهار فیلد را به عنوان نمونه نمایش دهید سپس باید برچسب‌ها را به فرم عددی تبدیل کنید. به ترتیب زیر:

{'O':0, 'B-ARG0':1, 'I-ARG0':2, 'B-ARG1':3, 'I-ARG1':4, 'B-ARG2':5, 'I-ARG2':6, 'B-ARGM-LOC':7, 'I-ARGM':8, 'B-ARGM-TMP':9, 'I-ARGM-TMP':10}

یک تابع برای این کار بنویسید.

همچنین نیاز داریم که یک تابع داشته باشیم که طول همه جملات را یکسان کند. برای این کار از اضافه کردن توکن Pad_token به تعداد کافی به جملات استفاده کنید.

در نهایت یک کلاس Vocab پیاده سازی کنید که مجموعه داده را به عنوان یک آبجکت Vocab همراه توابع کمکی ارائه می‌دهد. این کلاس شامل چندین تابع باید باشد که در ادامه کار هر کدام را شرح دادیم.

وظیفه هر تابع:

__init__(self, word2id=None): این متد سازنده[9] کلاس است. اگر که مقدار word2id خالی نبود همین مقدار را در متغیر word2id کلاس بریزید در غیر اینصورت یک دیکشنری می‌سازد و توکن‌های Pad, Start, End, Unknown را به آن اضافه می‌کند. در آخر هم متغیر id2word را با برعکس کردن کلید و مقدار word2id مقداردهی کنید.

__getitem(self, word): این تابع اجازه می‌دهد که ایندکس کلمه داده شده در وکب را بازیابی کنید. اگر کلمه وجود نداشت ایندکس توکن unknown را برگردانید.

__len__(self): باید تعداد توکن‌های موجود در وکب را بشمارد.

add(self, word): اضافه کردن کلمه به وکب اگر کلمه جدید باشد. برای این کار طول وکب را به عنوان ایندکس کلمه در نظر بگیرید. ایندکس آن کلمه را هم برگردانید.

word2indices(self, sents): تبدیل لیستی از جملات به لیستی از ایندکس‌ها.

indices2words(self, word_ids): تبدیل لیستی از ایندکس‌ها به کلمات متناظر.

to_input_tensor(self, sents: list[list[str]]): تبدیل لیستی از جملات یا کلمات به تنسور با اضافه کردن Padding متناسب با طول جمله. این تابع تنسوری که جمله‌ها را با ایندکس متناظر با کلمات وکب نشان می‌دهد برمی‌گرداند.

from_corpust(corpus, size, remove_frac, freq_cutoff): متن پیکره[10] را می‌گیرد و وکب می‌سازد. متن پیکره همان فیلد text مجموعه داده است. سایز نشان دهنده حداکثر تعداد کلمات وکب است، freq_cutoff مشخص می‌کند حداقل تعداد تکرار کلمه در متن چقدر باشد تا آن کلمه را به وکب اضافه کنیم، remove_frac هم درصدی از کلمات که به نسبت بقیه کلمات کم کاربردتر هستند را فیلتر می‌کند. مقادیر این دو را به صورت دلخواه انتخاب کنید.(راهنمایی: از تابع add(word) برای ساخت وکب از روی پیکره کمک بگیرید.)

[9] Constructor
[10] Corpus

در نهایت این تابع وکب ساخته شده از پیکره را برمی‌گرداند. جنس آن هم باید از کلاس vocab باشد.

در بخش‌های بعدی از این کلاس و توابع آن استفاده کنید.

بخش دوم: LSTM Encoder Model

قسمت ۲-۱: پیاده سازی مدل LSTM

از یک لایه LSTM برای پیش بینی برچسب نقش معنایی کلمات استفاده کنید.

ورودی مدل: توکن‌ها و فعل جمله.

معماری شبکه عصبی:

۱. بردار embedding هر کلمه را از LSTM رد کنید و لایه مخفی LSTM متناظر را دریافت کنید.

۲. hidden state فعل را هم دریافت کنید.

۳. hidden state فعل را به hidden state هر توکن بچسبانید.

۴. سپس خروجی‌های بدست آمده از مرحله قبل را از یک لایه خطی برای تولید خروجی عبور دهید.

در این بخش، وظیفه شما این است که یک مدل LSTM Encoder مطابق معماری ذکر شده به وسیله کتابخانه‌ی پایتورچ پیاده سازی کنید. سپس مدل را برای چند ایپاک آموزش دهید. بعد از آموزش، نمودار خطا[11] و دقت[12] برای مجموعه داده valid و train رسم شوند. در نهایت، نتایج به دست آمده را به طور خلاصه تحلیل کنید.

مقادیر پیشنهادی برخی پارامترها و هایپرپارامترها:

سایز امبدینگ: ۶۴، hidden_dim: ۶۴، نرخ یادگیری: 0.1، تعداد ایپاک: ۱۰، سایز بچ: ۶۴

حداکثر سایز وکب: ۲۰۰۰۰، میزان remove_frac: 0.3

قسمت ۲-۲: برای مجموعه داده validation مقدار F1 score محاسبه و گزارش شوند.

بخش سوم: GRU Encoder Model

قسمت ۳-۱: مراحل بخش قبل را یکبار دیگر با جایگزین کردن GRU بجای LSTM انجام دهید و نتایج آن‌ها را با هم مقایسه کنید.

قسمت ۳-۲: به سوالات زیر پاسخ داده شود.

سوال یک. مزیت LSTM به RNN چیست؟

سوال دو. تفاوت LSTM و GRU را توضیح دهید.

[11] Loss
[12] Accuracy

سوال دو. چرا نیاز داریم که hidden state فعل را با hidden layer همه توکن‌ها در این مدل concatenate کنیم؟

سوال چهار. اگر در شبکه‌های بازگشتی مشکل محو شدگی گرادیان[۱۳] رخ بدهد چه راه حلی را پیشنهاد می‌دهید.(بدون اینکه خود مدل را عوض کنید.)

بخش چهارم: Encoder-Decoder Model

می‌توانیم مسئله SRL را به فرمت مسئله پرسش و پاسخ(QA) تبدیل کنیم.

برای مثال :He wouldn’t accept anything of value from those he was writing about

این جمله ورودی با فعل accept را داریم و میخواهیم ARG0, ARG1, ARG2, ARGM_TMP, ARGM_LOC را تشخیص دهیم. برای این کار می‌توان هر نمونه را به ۵ جفت پرسش و پاسخ تبدیل کنیم.

فرمت هر جفت پرسش و پاسخ: Predicate [SEPT] sentence label

پاسخ هر پرسش دنباله متناظر به آن برچسب است. اگر هم دنباله خالی باشد، آن لیبل در جمله وجود ندارد.

Input 1: accept [SEPT] He wouldn’t accept anything of value from those he was writing about.ARG0

Output1: <s> He </s>

Input 2: accept [SEPT] He wouldn't accept anything of value from those he was writing about . ARG1

Output 2: <s> anything of value </s>

Input 3: accept [SEPT] He wouldn't accept anything of value from those he was writing about . ARG2

Output3: <s> </s>

Input 4: accept [SEPT] He wouldn't accept anything of value from those he was writing about . ARGM-TMP

Output 4: <s> </s>

Input 5: accept [SEPT] He wouldn't accept anything of value from those he was writing about . ARGM-LOC

Output 5: <s> </s>

در این بخش ورودی به شما داده می‌شود و باید یک مدل seq2seq برای تولید خروجی استفاده کنید.

قسمت ۴-۱: پیش پردازش مجموعه داده

ابتدا لازم است که مجموعه داده خام را به شکل جفت‌های پرسش و پاسخ مانند بالا تبدیل کنید. بطوری که ورودی و خروجی مناسب شبکه عصبی بازگشتی را داشته باشید. این کار را برای هر سه فایل مجموعه داده انجام دهید. سپس Glove embedding را بارگذاری کنید و برای هر کلمه در وکب، بردار جانمایی[۱۴] را ذخیره کنید که بعدا از این بردارهای جانمایی در مدل به عنوان مقداردهی اولیه استفاده شوند.

[13] Vanishing gradient
[14] Embedding

قسمت ۴-۲: پیاده سازی مدل

در این بخش از معماری Encoder-Decoder استفاده خواهید کرد. برای Encoder از مدل Bidirectional LSTM استفاده کنید و دربخش Decoder از مدل LSTM با مکانیسم توجه(Attention) استفاده کنید. همچنین برای تولید خروجی از beam_search با سایز ۱۶ استفاده شود. سپس مدل را آموزش دهید. و نمودار خطا و دقت را رسم کنید.

قسمت ۴-۳: برای مجموعه داده valid مقدار F1 score محاسبه و گزارش شود.

قسمت ۴-۴: به سوالات زیر پاسخ دهید.

سوال اول. محدودیت‌های روش تبدیل مسئله SRL به مسئله QA با استفاده از مدل encoder-decoder چیست؟

سوال دوم. چرا هنگام آموزش مدل از توکن‌های <s/>و <s> در ابتدا و انتهای خروجی استفاده می‌کنیم؟

بخش پنجم: تحلیل

قسمت ۵-۱: دو مدل بخش دو و بخش چهار را با هم از نظر کمی مقایسه کنید. دقت کنید که اگر مثلا مدل اول از مدل دوم بهتر بود در کدام نوع نقش معنایی بهتر عمل کرده است؟

قسمت ۵-۲: دو مدل بخش دو و بخش چهار را از نظر کیفی با هم مقایسه کنید. باید یکسری نمونه بیاورید و توضیح دهید چرا یکی درست پیش بینی می‌کند و دیگری نه. آیا مثالی وجود دارد که هر دو درست پاسخ دهند یا هر دو اشتباه کنند؟ اگر جواب مثبت است آیا می‌توانید دلیلی برای اینکه چرا این اتفاق می‌افتد بیاورید؟

ملاحظات (حتما مطالعه شود)

تمامی نتایج شما باید در یک فایل فشرده با عنوان NLP_CA3_StudentID تحویل داده شود.

- خوانایی و دقت بررسی‌ها در گزارش نهایی از اهمیت ویژه‌ای برخوردار است. به تمرین‌هایی که به صورت کاغذی تحویل داده شوند یا به صورت عکس در سایت بارگذاری شوند، ترتیب اثری داده نخواهد شد.
- کدهای نوشته شده برای هر بخش را با نام مناسب مشخص کرده و به همراه گزارش تکلیف ارسال کنید. همه‌ی کدهای پیوست گزارش بایستی قابلیت اجرای مجدد داشته باشند. در صورتی که برای اجرا مجدد آنها نیاز به تنظیمات خاصی می‌باشد بایستی تنظیمات مورد نیاز را نیز در گزارش خود ذکر کنید.
- تمرین تا یک هفته بعد از مهلت تعیین شده با تاخیر تحویل گرفته می شود. دقت کنید که شما جمعا برای تمام تکالیف، ۱۴ روز زمان تحویل بدون جریمه دارید که تنها از ۷ روز آن برای هر تمرین می‌توانید استفاده کنید، در صورتی که این ۱۴ روز به اتمام رسیده باشد، به ازای هر روز تاخیر در ارسال تمرین، ده درصد جریمه میشوید.
- **توجه کنید این تمرین باید به صورت تک نفره انجام شود و پاسخ‌های ارئه شده باید نتیجه فعالیت فرد نویسنده باشد (همفکری و به اتفاق هم نوشتن تمرین نیز ممنوع است). در صورت مشاهده تشابه به همه افراد مشارکت کننده، نمره تمرین صفر و به استاد نیز گزارش می‌گردد.**
- در صورت بروز هرگونه مشکل با ایمیل زیر در ارتباط باشید:

parhambicharanlu1378@gmail.com

مهلت تحویل بدون جریمه: ۷ اردیبهشت ۱۴۰۳

مهلت تحویل با تأخیر، با جریمه ۱۰ درصد : ۱۴ اردیبهشت ۱۴۰۳