بسمه تعالی

استخراج واحدهای اندازه‌گیری و کمیت‌ها از متن

گزارش تمرین 1

مرضیه نوری - محمد دل‌خواه

رویکرد کلی

در این پروژه، گروه ما به ترک استخراج مقادیر و کمیت‌ها پرداخت. از آنجا که انواع کلمات و واحدهای این مقوله بسیار زیاد است، تیم ما ابتدا به شناسایی الگوهای بکارگیری متفاوت در جملات برای استخراج کمیت‌ها پرداخت. این الگوها محور اصلی پیاده‌سازی و معماری کد را تشکیل دادند. با این حال با پخته شدن شناخت از این الگوها و پیشرفت پیاده‌سازی، محتوای آن‌ها نیز دستخوش تغییر شد. برخی از الگوها پیش از پیاده سازی یا در حین آن با یکدیگر ادغام شدند و برخی دیگر اضافه شدند. همچنین برخی به علت دشواری و دور بودن از بکارگیری مستقیم یک کمیت پیاده‌سازی نشدند.

رویکرد اصلی در یافتن این الگوها چنان بود که به علت تعدد کلمات درگیر در این پروژه، آن‌ها را بر اساس نوع کلی کلمات و عبارات در الگوها تفکیک کنیم و جزییات امکان ترکیب آن‌ها با یکدیگر را به صورت داده‌محور پیاده سازی کنیم. این موضوع در ادامه بیشتر توضیح داده خواهد شد.

بر اساس نوع کلی کلمات و عبارات بکار رفته شده در الگوها، دسته‌های عدد، واحد، صفت، کلیدواژه، ادات سوال و کلمات معین‌گر تعداد شناسایی شد. در بخش توضیح داده‌ها به شرح هر یک خواهیم پرداخت.

	A	B	C	D
1	ID	الگو	مثال	Done
2	0	عدد + واحد [+ آیتم]	3 کیلوگرم برنج خریدم| یک ماشین به وزن 3 تن محمد را زیر گرفت|باتری خود را 85صدم وات شارژ کرد	Yes
3	1	کلیدواژه + عدد	خانه اتاقی به ضلع 4 دارد	Yes
4	2	کمیت + صفت	ماشین با سرعت زیاد تصادف کرد	Merged
5	3	کلیدواژه + صفت	رودخانه‌ای با عرض کم دیدم	Yes
6	4	صفت	سریع از کنارم گذشت	Yes
7	5		یک کیلو شیرینی خرید و نصف آن را خورد	--
8	6	کلیدواژه	سرعت حرف اول را می‌زند | سرعت گرفت	Yes
9	8	کمیت + نوع کمیت	نیروی گرانشی|اصطکاک|عمودبرسطح|کشسانی|الکترومغناطیسی|مقاومت هوا|	Merged
10	9	کمیت +نوع کمیت + صفت	نیروی اصطکاک زیاد	Merged
11	10	تشبیه بدون وجه شبه صریح	مثل برق از کنارم گذشت	--
12	11	به + کلیدواژه + ی + اسم	وزنه‌ای به سنگینی یک فیل را بلند کرد.	--
13	12	مثل + اسم + کلیدواژه	مثل فیل سنگین بود	--
14	13	ادات سوال + واحد	چند کیلو داریم؟	Merged
15	14	ادات سوال + دهدهی + واحد + [آیتم]	چند صد ژول انرژی	Yes

معماری و ساختار کد

همان طور که در فایل ipynb پروژه قابل مشاهده است، کد از 2 نوع قسمت تشکیل شده:

- کلاس MeasurementExtractor
- توابع تشخیص الگو

البته این تفکیک تنها به علت تست و کارکردن راحت‌تر با کد و استفاده از امکانات سلول‌های jupyter است و به راحتی می‌توان تمام توابع تشخیص الگوها را به داخل کلاس MeasurementExtractor منتقل نمود.

مطابق خواست دستورالعمل تمرین، تابع اصلی کلاس run بوده که با گرفتن یک رشته در خروجی قطعه‌هایی که به کمیت‌ها اشاره دارند را به همراه اطلاعات قابل استخراج آن‌ها به صورت آرایه‌ای از jsonها باز می‌گرداند. کار اصلی تابع run اجرا و ادغام توابع تشخیص الگو و در نهایت بازگرداندن خروجی نهایی است.

کار اصلی دیگر کلاس MeasurementExtractor خواندن دادگان مورد نیاز از فایل‌ها و آماده‌سازی آن‌ها برای استفاده است. همچنین توابع کمکی مورد استفاده در برخی الگوها در آن قرار دارد.

داده‌ها

همان طور که ذکر شد، در این پروژه تلاش شده رویکرد تا جای ممکن data-driven باشد و داده‌ها از کد تفکیک شوند. تشخیص عدد و گروه اسمی بر عهده کتابخانه‌های خارجی parsi.io و هضم قرار داده شده و دیگر المان‌های محوری موجود در الگوها، یعنی واحد، صفت، کلیدواژه، ادات سوال و کلمات معین‌گر تعداد، همه به صورت فایل‌های csv مجزا در کنار پروژه قرار داده می‌شوند. تا بدون نیاز به شناخت کد بتوان آن‌ها را تکمیل کرد و ارتقا داد.

واحدها

واحدها از سایت باحساب استخراج شده و در یک فایل csv قرار داده شده که هر خط آن متعلق به یک نوع کمیت، و عناصر هر خط به صورت جفت‌های «کلمه واحد» و «نماد واحد» هستند.

	A	B	C	D	E	F	G	H	
1	length	متر	m	سانتیمتر	cm	اینچ	inch	میلیمتر	mm
2	mass	تن	Ton	کیلوگرم	kg	پوند	lb	گرم	g
3	pressure	پاسکال	pa	بار	bar	پی اس آی	psi	کیلو پاسکال	kpa
4	volume	لیتر	L	میلی لیتر	mL	متر مکعب		سانتیمتر مکعب	
5	temperature	سانتیگراد	C	فارنهایت	F	کلوین	K	رانکین	R
6	area	متر مربع		سانتیمتر مربع		میلی متر مربع		کیلومتر مربع	
7	speed	متر بر ثانیه	m/s	فوت بر ثانیه	fps	کیلومتر بر ساعت	km/h	مایل بر ساعت	mph
8	force	نیوتن	N	گرم-نیرو	gf	کیلوگرم-نیرو	kgf	پوند-نیرو	lbf
9	energy	ژول	j	میلی ژول	mj	میکرو ژول	µj	کیلو ژول	kj
10	power	وات	W	میلی وات	mW	کیلو وات	kW	مگا وات	MW
11	torque	نیوتن-متر	N.m	فوت-پوند نیرو	ft.lbf	متر-کیلوگرم نیرو	m.Kgf	متر-کیلو پوند	m.Kp
12	time	ثانیه	s	کیلو ثانیه	ks	میلی ثانیه	ms	میکرو ثانیه	µs
13	density	تن بر متر مکعب	t/m³	کیلوگرم بر متر مکعب	kg/m³	کیلوگرم بر لیتر	kg/L	کیلوگرم بر دسی متر م	kg/dr
14	frequency	هرتز	Hz	دور بر ثانیه	rps	دور بر دقیقه	rpm	دور بر ساعت	rph
15	degree	رادیان	Rad	درجه	Deg	گراد	Grad	ثانیه قوسی	arc s
16	acceleration	متر بر مجذور ثانیه		کیلو متر بر مجذور ثانیه		سانتی متر بر مجذور ثانیه		فوت بر مجذور ثانیه	
17	debi	تن بر ثانیه		تن بر دقیقه		تن بر ساعت		تن بر روز	
18	debi-v	متر مکعب بر ثانیه		متر مکعب بر دقیقه		متر مکعب بر ساعت		متر مکعب بر روز	
19	data-storage	بیت	Bit	کیلو بیت	kb	کیبی بیت	Kib	مگا بیت	Mb
20	data-transfer	بیت برثانیه	Bit/s	کیلو بیت برثانیه	kb/s	کیبی بیت برثانیه	Kib/s	مگا بیت برثانیه	Mb/s
21	ratio	درصد	٪						
22	wildcard	واحد							

صفات

صفات از رده «صفات فارسی» سایت wiktionary استخراج شده‌اند و در میان آن‌ها صفاتی که مرتبط با کمیت‌ها و قابل استفاده در الگوها هستند در ستون Useful با عدد 1 به عنوان پرچم مشخص گشته‌اند. همچنین در میان صفات مرتبط، همه در تمام الگوها قابل استفاده نیستند برای همین این که در چه الگویی امکان استفاده از آن‌ها وجود دارد با یک پرچم در ستون آن الگو مشخص شده است. در نهایت برای الگوی 4، از آنجا که نوع کمیت باید از صفت استنتاج شود، در یک ستون نوع کمیت ارجح برای آن صفت مشخص شده است.

شایان ذکر است همه صفات در wiktionary موجود نبودند و برخی مانند خنک، گرم و ... به صورت دستی توسط تیم به آن اضافه گشته‌اند.

	A	B	C	D	E
1	adjective	useful	valid for pattern 4	pattern 4 type	valid for pattern 3
2	آبی	1			
3	آرام	1	1	speed	1
4	آستان	1	1	length	
5	آستین دراز	1			
6	آشکار	1			
7	آکبند	1	1	time	
8	آهسته	1	1	speed	1
9	ابتدایی	1	1	time	1
10	ارزان	1			
11	افسارگسیخته	1			1
12	انبوه	1			
13	اندک	1			1
14	انگیزش	1	1	energy	
15	اونجا	1			
16	ایستاده	1	1	speed	

کلیدواژه‌ها

کلیدواژه‌ها نیز مشابه صفات بوده با این تفاوت که همه آن‌ها به صورت دستی توسط اعضای تیم نوشته شده‌اند. در کلیدواژه‌ها نیز مشابه صفات الگویی نیازمند تفکیک کلمات موجود به قابل استفاده و غیر قابل استفاده بود که با روش پرچم گذاری انجام شد.

acceleration	افزایش سرعت	1	
acceleration	کاهش سرعت	1	
acceleration	تغییر سرعت	1	
acceleration	میرا	0	
acceleration	ترمز	1	
data-storage	حجم	1	
data-storage	اندازه	1	
data-storage	بزرگ	0	
data-storage	کوچک	0	
data-storage	حافظه	1	
mass	وزن	1	
mass	سنگینی	1	وزنه‌ای به سنگینی یک فیل را بلند کرد.
mass	سنگین	0	بسته سنگین بود.
mass	سبکی	1	چیزی به سبکی پر در هوا معلق بود.
mass	سبک	0	بسته سبک بود.
mass	ترازو	1	ترازو روی عدد 2 ایستاد.
mass	جرم	1	جسمی با جرم بالا سقوط کرد. | جسمی به جرم دو کیلوگرم سقوط کرد.
temperature	دما	1	
temperature	دمای	1	
temperature	داغ	0	.آب داغ بود
temperature	سرد	0	آب سرد بود.

کلمات معین‌گر تعداد

در این فایل علاوه بر پیشوندهای یونانی، کلمات خاص مانند جین و دوجین و کلمات اعداد دودویی (کیبی، گیبی و ...) قرار داده شده است. بنا بود این کلمات از ابتدای واحدها جدا شده و به صورت مجزا تشخیص داده شوند تا با توجه به عدد متناظر آن‌ها بتوان در تبدیل واحد از آن‌ها استفاده نمود. متاسفانه با توجه به اتمام وقت این کار پیاده‌سازی نشد.

میلیاردیوم		0.000000001
تریلیونیوم		0.000000000001
کوادریلیونیوم		0.000000000000001
کوئینتلیوم		0.000000000000000001
سکسیتلیوم		0.000000000000000000001
سپتیلیونیوم		0.000000000000000000000001
جفت	Pair	2
کیبی	Ki	1024
مبی	Mi	1048576
گیبی	Gi	1073741824
تبی	Ti	1099511627776
پبی	Pi	1125899906842624
اگزبی	Ei	1152921504606846976
زبی	Zi	1180591620717411303424
یوبی	Yi	1208925819614629174706176
جین		6
دوجین	dozen	12

فایل‌های کمکی

در این فایل‌ها بعضی کلمات بجای نوشته شدن در کد قرار گرفته اند مانند ترجمه نوع کمیت‌ها و ادات سوال.

	A	B
1	english	persian
2	length	طول
3	mass	وزن
4	pressure	فشار
5	volume	حجم
6	temperature	دما
7	area	مساحت
8	speed	سرعت
9	force	نیرو
10	energy	انرژی
11	power	توان
12	torque	گشتاور
13	time	زمان
14	density	چگالی
15	frequency	فرکانس
16	degree	زاویه
17	acceleration	شتاب
18	debi	شارش جرمی
19	debi-v	شارش حجمی
20	data-storage	ذخیره دیجیتال
21	data-transfer	انتقال داده
22	wildcard	علم

	A
1	چند
2	چقدر
3	چقد
4	اینقدر
5	اینقد
6	انقد
7	انقدر
8	همینقد
9	همینقدر
10	چه مقدار
11	چندین

شناسایی آیتم

با توجه به آن که نقش کلمات در جملات بسیار متفاوت بوده و گاهی میان آن‌ها حرف اضافه یا علایم نگارشی خاصی ظاهر نمی‌گردد، تشخیص مسقیم آیتم با استفاده از عبارات منظم همیشه ممکن نیست. برای این کار

گروه ما از قابلیت chunker کتابخانه هضم استفاده نمود تا گروه‌های اسمی را تفکیک کند و هنگامی که در ادامه یک کمیت اسمی آمده باشد، تنها در صورتی به عنوان آیتم شناسایی شود که جزو همان گروه اسمی باشد. با این حال متاسفانه مدل chunker کتابخانه هضم همیشه درست کار نمی‌کند و باعث بروز خطا می‌شود. استفاده درست از علایم نگارشی مانند نقطه در انتهای جمله، خطای این کتابخانه را کاهش می‌دهد اما به صفر نمی‌رساند.

عدم وابستگی به کد و پیاده‌سازی

از مزیت‌های نحوه پیاده‌سازی این پروژه توسط گروه ما، تفکیک مناسب داده و رفتار است به طوری که در صورت نیاز به افزودن کمیت‌های کاملا جدید، هیچ نیازی به تغییر کد نبوده و کافی است اطلاعات لازم در فایل‌های csv اضافه شود. با این اتفاق تمام الگوهای موجود کمیت جدید را شناسایی کرده اطلاعات لازم را از نمونه‌های آن در متن داده شده استخراج می‌کنند. تنها در صورتی تغییر کد نیاز است که الگوی جدیدی مد نظر باشد که برای این کار کافی است تابعی برای آن نوشته شده و در run کلاس MeasurementExtractor صدا زده شود.