



دانشکده مهندسی کامپیوتر

ارائه راهکاری برای یکپارچه سازی دستگاه‌های اینترنت اشیاء با سکوی کوبرنیتز

پروژه کارشناسی مهندسی کامپیوتر گرایش هوش مصنوعی

سینا شعبانی کومله

استاد راهنما

دکتر محسن شریفی

تابستان ۱۴۰۲

تأییدیه‌ی هیأت داوران جلسه‌ی دفاع از پروژه

نام دانشکده: دانشکده مهندسی کامپیوتر

نام دانشجو: سینا شعبانی کومله

عنوان پروژه: ارائه راهکاری برای یکپارچه سازی دستگاه‌های اینترنت اشیا با سکوی کوبرنیتز

تاریخ دفاع: تابستان ۱۴۰۲

رشته: مهندسی کامپیوتر

گرایش: هوش مصنوعی

ردیف	سمت	نام و نام خانوادگی	مرتبه دانشگاهی	دانشگاه یا مؤسسه	امضاء
۱	استاد راهنما	دکتر محسن شریفی	استاد تمام	دانشگاه علم و صنعت ایران	
۲	داور نهایی	دکتر TODO	دانشیار	دانشگاه علم و صنعت ایران	

تأییدیه‌ی صحت و اصالت نتایج

باسمه تعالی

اینجانب سینا شعبانی کومله به شماره دانشجویی 97521351 دانشجوی رشته مهندسی کامپیوتر مقطع تحصیلی کارشناسی تأیید می‌نمایم که کلیه‌ی نتایج این پروژه حاصل کار اینجانب و بدون هرگونه دخل و تصرف است و موارد نسخه‌برداری شده از آثار دیگران را با ذکر کامل مشخصات منبع ذکر کرده‌ام. در صورت اثبات خلاف مندرجات فوق، به تشخیص دانشگاه مطابق با ضوابط و مقررات حاکم (قانون حمایت از حقوق مؤلفان و مصنفان و قانون ترجمه و تکثیر کتب و نشریات و آثار صوتی، ضوابط و مقررات آموزشی، پژوهشی و انضباطی ...) با اینجانب رفتار خواهد شد و حق هرگونه اعتراض در خصوص احقاق حقوق مکتسب و تشخیص و تعیین تخلف و مجازات را از خویش سلب می‌نمایم. در ضمن، مسئولیت هرگونه پاسخگویی به اشخاص اعم از حقیقی و حقوقی و مراجع ذیصلاح (اعم از اداری و قضایی) به عهده‌ی اینجانب خواهد بود و دانشگاه هیچ‌گونه مسئولیتی در این خصوص نخواهد داشت.

نام و نام خانوادگی: سینا شعبانی کومله

تاریخ و امضا:

مجوز بهره‌برداری از پایان‌نامه

بهره‌برداری از این پایان‌نامه در چهارچوب مقررات کتابخانه و با توجه به محدودیتی که توسط استاد راهنما به شرح زیر

تعیین می‌شود، بلامانع است:

☐ بهره‌برداری از این پایان‌نامه برای همگان بلامانع است.

☐ بهره‌برداری از این پایان‌نامه با اخذ مجوز از استاد راهنما، بلامانع است.

☐ بهره‌برداری از این پایان‌نامه تا تاریخ ممنوع است.

استاد راهنما: دکتر محسن شریفی

تاریخ:

امضا:

چکیده

در حال حاضر، مدیریت و نظارت بر دستگاه‌های اینترنت اشیا¹ به یک چالش عمده تبدیل شده است. راه حل‌های موجود برای کنترل و نظارت بر این دستگاه‌ها اغلب ناسازگاری‌ها و محدودیت‌هایی دارند که موجب کاهش کارایی و پیچیدگی مدیریت در مقیاس بالا می‌شوند. به منظور حل این مسئله، این پروژه تلاش می‌کند تا با استفاده از کوبرنیتز² و پروژه کوبلت مجازی³ یک سازوکار جامع برای نظارت و مدیریت دستگاه‌های اینترنت اشیا ارائه دهد. انگیزه اصلی پروژه متمرکز کردن کنترل و نظارت بر دستگاه‌های اینترنت اشیا به صورت یکپارچه و موثر است. راه حل‌های کنونی اغلب ناسازگاری‌هایی با استانداردها و فناوری‌های مختلف دستگاه‌های اینترنت اشیا دارند و به تنهایی قادر به ارائه یک محیط یکپارچه برای مدیریت و نظارت نیستند. این پروژه شامل سه بخش اصلی، یعنی تامین‌کننده⁴، کنترل‌کننده⁵ و دستگاه‌ها⁶ این بخش‌ها با یکدیگر ارتباط برقرار میکنند تا اطلاعات مفیدی درباره دستگاه‌های اینترنت اشیا مورد کنترل ارائه دهند و این اطلاعات را در دسترس خوشه کوبرنیتز قرار دهند.

واژگان کلیدی: اینترنت اشیا، کوبرنیتز، کوبلت مجازی، نظارت یکپارچه، پایش مقیاس پذیر

Internt Of Things (IoT)¹

Kubernetes²

Virtual Kubelet³

Provider⁴

Controller⁵

Device⁶

فهرست مطالب

ح فهرست تصاویر

خ فهرست جداول

1	مقدمه	فصل 1:
1-1	شرح مسأله	
2-1	اهداف پروژه	
3-1	ساختار گزارش	
3	مفاهیم پایه	فصل 2:
1-2	مقدمه	
2-2	بستر ابری	
3-2	سکوی کوبرنیتز	
1-3-2	کانتینر	
2-3-2	پاد	
3-3-2	گره	
4-3-2	کوبلت	
5-3-2	خوشه کوبرنیتز	
4-2	HTTP	
1-4-2	توابع HTTP	
2-4-2	درخواست HTTP	

11	پاسخ HTTP	3-4-2
11	REST	4-4-2
12	رابط کاربردی قابل برنامه‌ریزی	5-4-2
13	معماری فن‌آوت	5-2
14	کوبلت مجازی	6-2
15	معماری کوبلت مجازی	1-6-2
16	اینترنت اشیاء	7-2
17	جمع‌بندی	8-2
18	کارهای مرتبط	فصل 3:
18	مقدمه	1-3
19	روش پیشنهادی	فصل 4:
19	مقدمه	1-4
19	معماری سامانه	2-4
20	نحوه استفاده	3-4
20	رابط کاربردی گرافیکی	4-4
21	مقدمه	5-4
21	معماری سیستم	6-4
22	ورودی اولیه و خروجی نهایی	7-4
23	مدل Encoder	8-4
23	مدل LXMERT [?]	1-8-4
24	مدل VisualBERT [?]	2-8-4
24	مدل Decoder	9-4
25	جمع‌بندی	10-4
26	ارزیابی روش پیشنهادی	فصل 5:

26	مقدمه	1-5
--------------	-------	-----

27	نتیجه گیری و کارهای آینده	فصل 6:
----	---------------------------	--------

27	نتیجه گیری	1-6
--------------	------------	-----

28	دستاوردها	2-6
--------------	-----------	-----

28	کارهای آینده	3-6
--------------	--------------	-----

29	واژه نامه فارسی به انگلیسی
----	----------------------------

31	واژه نامه انگلیسی به فارسی
----	----------------------------

فهرست تصاویر

1-2	معماری کلی کوبرنیتز	5
2-2	نمای کلی درخواست و پاسخ HTTP	9
3-2	نمای کلی از REST	12
4-2	نمای کلی از رابط کاربردی قابل برنامه‌ریزی	13
5-2	کوبلت مجازی در یک خوشه کوبرنیتز	15
6-2	نمای کلی از اینترنت اشیاء	17
1-4	نمای کلی معماری	20
2-4	شمای کلی سیستم پرسش و پاسخ تصویری	21
3-4	نمونه‌ای از کارکرد Tokenizer	22
4-4	معماری مدل LXMERT همراه با ورودی و خروجی	23
5-4	میزان توجه بخش متن به تصاویر در VisualBERT	24
6-4	ساختار مدل Autoregressive Decoder	25

فهرست جداول

فصل 1

مقدمه

1-1 شرح مسأله

یکی از مسائلی که امروزه در زمینه کنترل و پایش دستگاه‌های اینترنت اشیا وجود دارد، عدم یکپارچگی و هماهنگی میان دستگاه‌های مختلف است. این دستگاه‌ها از فناوری‌ها، پروتکل‌ها و استانداردهای متنوعی برای ارتباط و عملکرد استفاده میکنند، که این تنوع باعث پیچیدگی و مشکلاتی در کنترل و پایش مرکزی آنها میشود. به عنوان مثال، در یک بستر اینترنت اشیا ممکن است دستگاه‌هایی با پروتکل‌های ارتباطی مختلف، مانند HTTP¹، MQTT² و CoAP³ داشته باشند که هر کدام نیازمند روش‌ها و فناوری‌های جداگانه برای کنترل و پایش خود هستند. همچنین، دستگاه‌های اینترنت اشیا ممکن است از نظر تکنولوژی و نوع عملکرد با هم تفاوت داشته باشند. برای مثال، یک سنسور دما و یک قفل هوشمند دارای نیازهای کنترل و پایش متفاوتی هستند. این تنوع در دستگاه‌ها باعث پیچیدگی در توسعه و اجرای یک سیستم کنترل یکپارچه می‌شود. محدودیت منابع نیز یک چالش اساسی در محیط‌های اینترنت اشیا است. این دستگاه‌ها منابع محدودی نظیر پردازشگر، حافظه و پهنای باند شبکه دارند که توان محاسباتی آنها را به شدت کاهش می‌دهد. بنابراین، ضرورت بهره‌برداری بهینه از این منابع و مدیریت آنها به منظور افزایش کارایی و بهره‌وری دستگاه‌ها مطرح میشود. همچنین، امنیت و حفاظت از اطلاعات حساس در محیط‌های اینترنت اشیا نیز از اهمیت بالایی برخوردار است، زیرا این دستگاه‌ها اطلاعاتی حساس را در محیط شبکه منتقل میکنند که به تهدیدات امنیتی از جمله نفوذ، جاسوسی و دسترسی

¹Hypertext Transfer Protocol

²Message Queuing Telemetry Transport

³Constrained Application Protocol

غیرمجاز معرض هستند.

2-1 اهداف پروژه

هدف اصلی پروژه امکان‌سنجی، طراحی، پیاده‌سازی و ارزیابی سیستمی برای کنترل و پایش دستگاه‌های اینترنت اشیا بر سکوی کوبنیتز است.

3-1 ساختار گزارش

در این پروژه هدف ارائه روشی نو برای حل مسئله کنترل و پایش دستگاه‌های اینترنت اشیا بر سکوی کوبنیتز است. در ابتدا به معرفی مفاهیم پایه استفاده شده در این پروژه و سپس به معرفی روش‌ها و کارهای مرتبط پرداخته خواهد شد. پس از آن به معرفی روش و ارزیابی آن پرداخته شده است. در انتها نتیجه‌گیری و کارهای آینده معرفی می‌شوند.

فصل 2

مفاهیم پایه

1-2 مقدمه

در این بخش به معرفی مفاهیم پایه درباره سکوی کوبرنیتز، پروژه کوبلت مجازی، اینترنت اشیاء و برخی معماری‌هایی که در این پروژه استفاده شده‌اند به مانند استخر کارگران¹ پرداخته شده است.

2-2 بستر ابری

بستر ابری² به محیطی اشاره دارد که منابع محاسباتی، شبکه و ذخیره سازی را برای ارائه خدمات به صورت ابری و توسط یک ارائه دهنده ابری فراهم می کند. در این محیط، خدمات و برنامه‌ها بر روی سرورهای فیزیکی مجازی سازی شده قرار می گیرند و کاربران می توانند به آنها از طریق اینترنت وصل شوند و از آنها استفاده کنند. با استفاده از محیط ابری، امکاناتی مانند انعطاف پذیری بالا، قابلیت مقیاس پذیری، اشتراک گذاری منابع و مدیریت آسانتر برای خدمات فراهم می شود. همچنین یکی از مزیت‌های بستر ابری ساده سازی ساخت، مدیریت و انتشار یک خدمت می‌باشد.

Worker Pool¹
Cloud Environment²

3-2 سکوی کوبرنیتز

کوبرنیتز³ یک سامانه مدیریت کانتینرها⁴ است که توسط گوگل توسعه داده شده است و در حال حاضر تحت نظارت و پشتیبانی مؤسسه CNCF⁵ قرار دارد. این ابزار به توسعه‌دهندگان و مدیران سامانه امکان می‌دهد برنامه‌ها و خدمات را در بسترهای ابری⁶ مدیریت کنند و کانتینرها را به طور موثر و مقیاس‌پذیر در محیط‌های توزیع شده⁷ مدیریت کنند. از طریق کوبرنیتز، می‌توان کانتینرها را بر روی یک سرور مجازی⁸ یا فیزیکی⁹ اجرا کرده و مدیریت آنها را ساده‌تر و مؤثرتر نمود. این سامانه با بهره‌گیری از روش‌هایی مانند اتوماسیون¹⁰، توازن بار¹¹ و تشخیص خودکار اشکال¹²، مدیریت و کنترل بهبود یافته‌ای در محیط‌های مبتنی بر کانتینر فراهم می‌کند. این ابزار برای حل مشکلات زیر موثر است:

1. مقیاس‌پذیری: کوبرنیتز می‌تواند تعداد کانتینرها و پیش‌نمونه‌های برنامه را بر اساس نیازهای ترافیک و خدمت تنظیم کند. با استفاده از مدیریت منابع مبتنی بر درخواست، میزان منابع مورد استفاده توسط برنامه را به طور خودکار تنظیم می‌کند و مقیاس‌پذیری عمودی و افقی را به راحتی فراهم می‌کند.

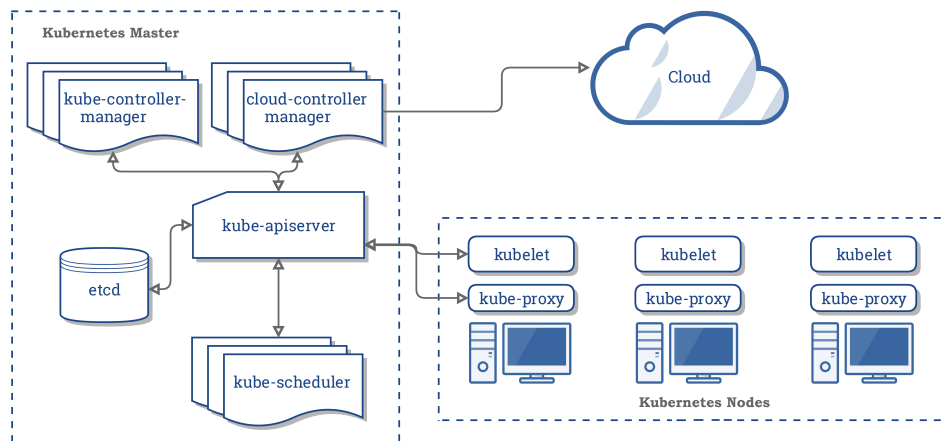
2. توازن بار: با استفاده از کوبرنیتز، می‌توان بار کار را به طور متوازن بین نودها و سرورهای مختلف تقسیم کرد. این باعث بهبود عملکرد و عدم وقوع اختلال در سامانه می‌شود. همچنین، در صورتی که یک نود یا سرور دچار مشکل شود، کوبرنیتز به طور خودکار کار را به سایر نودها منتقل می‌کند.

3. مدیریت پیچیدگی: کوبرنیتز امکاناتی را برای مدیریت پیچیدگی سامانه‌های کانتینری فراهم می‌کند. این ابزار اجرا، مدیریت، نظارت و زندگی دوباره سازی کانتینرها را ساده می‌کند. همچنین امکاناتی برای مدیریت تنظیمات، آپدیت‌ها، و تغییرات در حال اجرا نیز در اختیار کاربران قرار می‌دهد.

4. قابلیت انتقال: کوبرنیتز امکان انتقال برنامه‌ها و خدمات بین بسترهای مختلف را فراهم می‌کند. با استفاده از این امکان، می‌توان برنامه‌ها را بین محیط‌های توسعه، آزمون و تولید به راحتی منتقل کرد. کوبرنیتز با استفاده از

Kubernetes³Containers⁴Cloud Native Computing Foundation⁵Cloud Environment⁶Distributed Environment⁷Virtual Machine⁸Physical Server⁹automation¹⁰Load Balancing¹¹Automatic Diagnostics¹²

این قابلیت‌ها و خصوصیات، به توسعه‌دهندگان و مدیران سامانه امکان می‌دهد برنامه‌ها را به صورت مؤثر، قابلیت مقیاس‌پذیری و قابل اطمینان در محیط‌های کانتینری مدیریت کنند و عملکرد سامانه را بهبود دهند.



شکل 2-1: معماری کلی کوبرنیتز

1-3-2 کانتینر

کانتینرها یک فناوری پیشرفته در زمینه مدیریت و اجرای نرم‌افزار هستند. یک کانتینر، یک واحد نرم‌افزاری است که تمام نیازمندی‌های لازم برای اجرای یک نرم‌افزار را شامل می‌شود. در واقع، کانتینرها مجموعه‌ای از عملیات سامانه‌ای، کدها و تنظیماتی هستند که با یکدیگر در یک بستر محصور می‌شوند. از ویژگی‌های برجسته کانتینرها، می‌توان به استقلال و حمل‌پذیری آن‌ها اشاره کرد. به عبارتی دیگر، یک کانتینر می‌تواند بدون تغییر و با حفظ کارایی خود، بین بسترها و سامانه‌های عامل منتقل شود. این ویژگی باعث شده است که کانتینرها در صنعت فناوری اطلاعات بسیار محبوب شوند. برای مدیریت کانتینرها، ابزارهای مختلفی وجود دارند. یکی از محبوب‌ترین ابزارها برای مدیریت کانتینرها، داکر¹³ است. داکر یک بستر توسعه نرم‌افزار مبتنی بر کانتینر است که به توسعه‌دهندگان امکان می‌دهد تا برنامه‌های خود را در یک کانتینر قرار داده و آن را در هر سامانه‌ای اجرا کنند. با استفاده از کانتینرها، عملیات توسعه، آزمون و استقرار نرم‌افزارها سریع‌تر و ساده‌تر می‌شود. با توجه به این که هر کانتینر دارای محیط مستقلی است، احتمال بروز تداخل بین برنامه‌ها به

¹³Docker

حداقل می‌رسد و تغییرات در یک کانتینر بر روی سایر کانتینرها تأثیری نمی‌گذارد. همچنین، سبک بودن کانتینرها امکان مقیاس‌پذیری بالایی را فراهم می‌آورد.

2-3-2 پاد

پادها¹⁴ در کوبرنیتز واحد اصلی اجرا و مدیریت برنامه‌ها و خدمات هستند. یک پاد شامل یک یا چند کانتینر مرتبط است که به صورت مشترک منابع شبکه و ذخیره‌سازی را به اشتراک می‌گذارند. همچنین، هر پاد دارای یک آدرس یکتا درون کلاستر است. پادها به صورت لایه‌ای مجازی شبیه سازی می‌شوند و انتزاعی از یک ماشین مجازی یا سامانه عامل فیزیکی هستند. این انتزاع به برنامه‌ها امکان می‌دهد تا بدون احتیاج به اطلاعات جزئیات بستری که برای اون اجرا می‌شوند، در محیط کنترلی کوبرنیتز اجرا شوند. بنابراین، پادها برای توسعه‌دهندگان و مدیران سامانه، یک واسطه سطح بالا و یک فضای کاری است.

3-3-2 گره

در بستر کوبرنیتز، گره‌ها¹⁵ از اجزای کلیدی هستند که برنامه‌ها و خدمات در آن‌ها اجرا می‌شوند. یک گره معمولاً یک سرور فیزیکی یا ماشین مجازی است که بر روی آن کانتینرها اجرا می‌شوند. هر گره شامل عناصر زیر است:

1. پلتفرم سخت‌افزاری: این شامل سرورها، سامانه‌های فیزیکی، یا ماشین‌های مجازی است که منابع سخت‌افزاری مانند پردازنده، حافظه، و دیسک را فراهم می‌کنند. گره‌ها بسته به نیازهای برنامه‌ها و خدمات، می‌توانند از طریق شبکه به یکدیگر متصل شوند.

2. کوبلت¹⁶: مسئول مدیریت و اجرای کانتینرها در گره است. آن بر روی هر گره نصب شده و با کنترل‌کننده‌های کوبرنیتز برای دریافت توصیف کانتینرها و مدیریت آن‌ها در ارتباط است.

3. پروکسی¹⁷: یک کنترل‌کننده شبکه است که مسئول مدیریت ترافیک شبکه بین کانتینرها در گره است. این عملکرد به ارتباط و مسیریابی درخواست‌ها بین کانتینرها و اجزای دیگر کوبرنیتز مرتبط است.

¹⁴ پاد

¹⁵ Node

¹⁶ Kublet

¹⁷ Kube-proxy

4. حافظه مشترک¹⁸: گره‌ها از یک حافظه مشترک برای ذخیره و به اشتراک گذاری اطلاعاتی مانند پیکربندی‌ها و وضعیت گره‌ها استفاده می‌کنند. این حافظه مشترک معمولاً از طریق ابزارهای ذخیره‌سازی مانند ETCD پیاده‌سازی می‌شود.

با استفاده از گره‌ها، کوبرنیتز قادر است برنامه‌ها و خدمات را بر روی یک سری از سرورها یا ماشین‌های مجازی توزیع کند و به طور همزمان و مقیاس‌پذیر اجرا کند. این باعث افزایش انعطاف‌پذیری، بهره‌وری و پایداری در محیط‌های ابری و مجازی می‌شود.

4-3-2 کوبلت

کوبلت یکی از اجزای اصلی سامانه مدیریت کانتینرها کوبرنیتز است. کوبلت مسئول اجرا و مدیریت کانتینرها در یک گره می‌باشد. کوبلت در هر گره از خوشه¹⁹ کوبرنیتز نصب شده و وظیفه‌ای اساسی را بر عهده دارد که شامل موارد زیر است:

1. مدیریت کانتینرها: کوبلت مسئول ساخت و اجرای کانتینرها بر اساس توصیف‌هایی که از طرف کنترل‌کننده‌های کوبرنیتز به آن ارسال می‌شود، می‌باشد. این توصیف‌ها شامل اطلاعاتی مانند نرم‌افزار مورد نظر، تنظیمات شبکه و منابع مصرفی کانتینر می‌شوند.

2. پایش²⁰ منابع: کوبلت مسئول نظارت بر منابع مصرفی کانتینرها است و اطلاعات مربوط به استفاده از پردازنده، حافظه، شبکه و دیگر منابع سامانه را جمع‌آوری کرده و گزارش می‌دهد. این اطلاعات به کنترل‌کننده‌های کوبرنیتز ارسال می‌شوند تا بتوانند به طور هوشمند منابع را تخصیص دهند و بهینه‌سازی منابع را انجام دهند.

3. بروزرسانی و نگهداری کانتینرها: کوبلت مسئول بروزرسانی و نگهداری کانتینرها است. اگر نسخه جدیدی از نرم‌افزار موجود باشد، کوبلت قادر است آن را دریافت و کانتینرها را بروزرسانی کند. همچنین، در صورت خطا در اجرای کانتینر یا توقف آن، کوبلت تلاش می‌کند کانتینر را به طور خودکار مجدداً راه‌اندازی کند.

4. ارتباط با سایر اجزا: کوبلت وظیفه برقراری ارتباط با اجزای دیگر کوبرنیتز را نیز دارد. به عنوان مثال، با کنترل‌کننده²¹ برای دریافت دستورات مدیریتی، با برنامه‌ریز²² برای دریافت جدول‌بندی پیشنهادی و با کنترل‌کننده

¹⁸ Shared Memory

¹⁹ Cluster

²⁰ Monitoring

²¹ kube-controller-manager

²² kube-scheduler

شبکه²³ برای تنظیمات شبکه در ارتباط است.

به طور خلاصه، کوبلت یکی از اجزای کلیدی کوبرنیتز است که وظیفه مدیریت و اجرای کانتینرها را در گره‌های سامانه بر عهده دارد. این کامپوننت از طریق ارتباط با سایر اجزا و دریافت توصیف‌های مربوطه، به ایجاد و مدیریت یک محیط توزیع شده و مقیاس‌پذیر برای اجرای برنامه‌ها و خدمات در کوبرنیتز کمک می‌کند.

5-3-2 خوشه کوبرنیتز

یک خوشه کوبرنیتز²⁴ یک بستر توزیع شده است که شامل مجموعه‌ای از گره‌ها است که برای مدیریت و اجرای برنامه‌ها و ارائه خدمات از طریق کوبرنیتز استفاده می‌شود. خوشه کوبرنیتز شامل اجزا و خدماتی متعددی است که با همکاری میان گره‌ها، برنامه‌ها را مدیریت می‌کنند.

HTTP 4-2

پروتکل HTTP²⁵ یک پروتکل ارتباطی است که در اینترنت استفاده می‌شود و برای انتقال اطلاعات بین سرور²⁶ و کلاینت²⁷ استفاده می‌شود. به طور کلی، HTTP به عنوان روشی برای انتقال اطلاعات و محتوا در وب مورد استفاده قرار می‌گیرد. در یک ارتباط، HTTP کلاینت درخواستی به سرور می‌فرستد و سپس سرور با پاسخ مناسب به درخواست کلاینت پاسخ می‌دهد. این درخواست و پاسخ در قالب پیام‌های متنی انجام می‌شود، که ممکن است شامل سرآیندها²⁸ و محتوای پیام²⁹ باشند. سرآیندها شامل اطلاعاتی مانند نوع محتوا، تاریخ، طول پیام و سایر جزئیات مربوط به ارتباط است. HTTP برای انتقال انواع مختلف منابع و اطلاعات در وب استفاده می‌شود. مثلاً می‌توان از HTTP برای دریافت صفحات وب، تصاویر، ویدیوها و سایر منابع در سرور استفاده کرد. همچنین، HTTP از مدل درخواست-پاسخ پیروی می‌کند، به این معنی که کلاینت درخواستی ارسال می‌کند و سرور با یک پاسخ مناسب به آن پاسخ می‌دهد. پروتکل HTTP اساسی است در عملکرد وب و تعامل بین سرویس‌دهنده و کلاینت. این پروتکل به صورت گسترده در نرم‌افزارها و سرویس‌های وب مورد استفاده قرار می‌گیرد و امکان انتقال اطلاعات و ارتباط بین کامپیوترها و دستگاه‌ها را فراهم می‌کند.

²³kube-proxy

²⁴Kubernetes Cluster

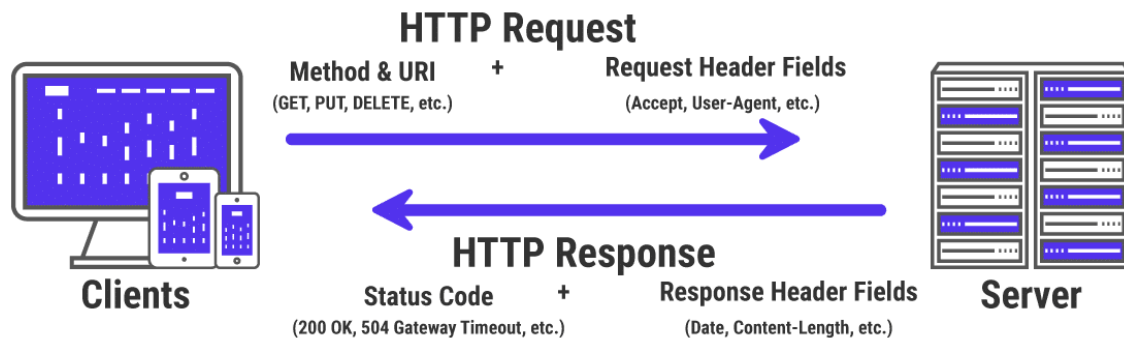
²⁵Hypertext Transfer Protocol

²⁶Server

²⁷Client

²⁸Header

²⁹Body



شکل 2-2: نمای کلی درخواست و پاسخ HTTP

1-4-2 توابع HTTP

پروتکل HTTP توابع³⁰ مختلفی را برای تعیین نوع عملیاتی که باید انجام شود تعریف کرده است که عبارت است از:

1. **GET**: این تابع برای دریافت (خواندن) اطلاعات از یک منبع مشخص درخواست می‌شود. مثلاً با استفاده از تابع GET می‌توانید صفحات وب، تصاویر یا سایر منابع را از سرور دریافت کنید. درخواست GET بدون تغییر و اثری روی منبع مورد درخواست، انجام می‌شود.

2. **POST**: این تابع برای ارسال داده‌ها به سرور استفاده می‌شود. از تابع POST برای ایجاد یا ارسال داده جدید به سرور استفاده می‌شود، مانند ارسال فرم‌ها در وب، ارسال نظرات یا انجام عملیات‌های پردازشی. درخواست POST می‌تواند تغییراتی روی منبع مورد درخواست ایجاد کند.

3. **PUT**: این تابع برای به‌روزرسانی (بازنویسی) یک منبع مشخص استفاده می‌شود. با استفاده از تابع PUT می‌توانید اطلاعات موجود در سرور را با داده‌های جدید جایگزین کنید. درخواست PUT می‌تواند تغییراتی روی منبع مورد درخواست اعمال کند یا در صورت عدم وجود، یک منبع جدید ایجاد کند.

4. **DELETE**: این تابع برای حذف یک منبع مشخص استفاده می‌شود. با استفاده از تابع DELETE می‌توانید یک منبع را از سرور حذف کنید. درخواست DELETE منبع مورد نظر را از سرور حذف می‌کند.

³⁰HTTP Methods

5. PATCH: این تابع برای اعمال تغییرات جزئی روی یک منبع مشخص استفاده می‌شود. با استفاده از تابع PATCH، می‌توانید تغییرات کوچکی را روی یک منبع اعمال کنید بدون ایجاد تغییرات بزرگتری در داده‌های موجود. این توابع به عنوان پایه‌های پروتکل HTTP عمل می‌کنند و به کلاینت و سرور امکان ارتباط و انجام عملیات‌های مختلف را می‌دهند. درخواست‌ها و پاسخ‌های HTTP بر اساس این توابع تعریف می‌شوند و برای تعامل موثر بین سرویس‌دهنده و کلاینت استفاده می‌شوند.

2-4-2 درخواست HTTP

در پروتکل HTTP، وقتی که کلاینت (مانند مرورگر وب یا برنامه‌ای که درخواست ارسال می‌کند) برای دستیابی به منبع مورد نظر خود، درخواستی ایجاد می‌کند، یک درخواست HTTP ایجاد می‌شود. درخواست‌های HTTP شامل اطلاعاتی درباره نوع و منبع درخواست شده هستند. در زیر توضیحی از مهمترین اجزای یک درخواست HTTP را می‌یابید:

1. توابع: HTTP تابع مشخص می‌کند کلاینت درخواست انجام چه عملیاتی را بر روی منبع مورد نظر دارد.
 2. درس³¹: مشخص می‌کند که منبع مورد نظر کجا قرار دارد و آدرس دقیق آن چیست. آدرس شامل پروتکل، نام دامنه سرور³² و مسیر³³ منبع مورد نظر است.
 3. سرآیندها: درخواست HTTP می‌تواند شامل سرآیندها باشد که اطلاعات اضافی درباره درخواست و مشخصات کلاینت را ارائه می‌دهند. به عنوان مثال، سرآیندها می‌توانند شامل اطلاعات مربوط به نوع محتوا، زبان مورد نظر، تاریخ و غیره باشند.
 4. بدنه درخواست³⁴: بدنه درخواست، در صورتی که درخواست ارسالی دارای داده‌های ارسالی است، این داده‌ها را در خود نگه می‌دارد. بدنه درخواست به صورت متنی یا در قالب داده‌های باینری می‌تواند باشد و معمولاً در درخواست‌هایی مانند POST و PUT استفاده می‌شود.
- هنگامی که کلاینت درخواست HTTP را به سرور ارسال می‌کند، سرور با استفاده از اطلاعات درخواست، منبع مورد نظر را پیدا کرده و به مناسبت درخواست، پاسخ مناسب را به کلاینت ارسال می‌کند. درخواست‌های HTTP بسته به نوع و عملیات مورد نظر، تعیین می‌کنند که چه عملیاتی باید در سمت سرور انجام شود و اطلاعات مربوطه را برگردانند.

³¹ Uniform Resource Locator (URL)

³² Domain Name

³³ Path

³⁴ Request Body

3-4-2 پاسخ HTTP

هنگامی که یک درخواست از سوی کلاینت به سرور ارسال می‌شود، سرور با یک پاسخ³⁵ مناسب به آن درخواست پاسخ می‌دهد. پاسخ‌های HTTP حاوی اطلاعات مربوط به نتیجه درخواست و وضعیت آن است.

1. کد وضعیت³⁶: هر پاسخ HTTP دارای یک کد وضعیت است که نشان می‌دهد درخواست با موفقیت انجام شده است یا با مشکل مواجه شده است. برخی از کدهای وضعیت رایج عبارتند از:

(آ) کد ۲۰۰: درخواست با موفقیت انجام شده است و پاسخ داده مورد نظر در دسترس است.

(ب) کد ۴۰۴: منبع مورد نظر پیدا نشد.

(ج) کد ۵۰۰: مشکلی در سمت سرور رخ داده است که موجب عدم توانایی در ارسال پاسخ مورد نظر شده است.

2. سرایندها: پاسخ HTTP شامل سرایندها است که اطلاعات اضافی در مورد پاسخ و منبع مورد نظر را ارائه می‌دهند. به عنوان مثال، سرایندها می‌توانند شامل اطلاعات مربوط به نوع محتوا، طول پاسخ، تاریخ ارسال، و غیره باشند.

3. محتوای پاسخ³⁷: پاسخ HTTP ممکن است حاوی محتوای مورد نظر باشد که توسط سرور ارسال می‌شود. محتوای پاسخ می‌تواند اطلاعاتی مانند متن، تصویر، ویدیو یا سایر منابع مورد نیاز را شامل شود.

پاسخ‌های HTTP ارسال شده توسط سرور معمولاً به درخواست‌های ارسالی از سوی مشتری پاسخ می‌دهند و اطلاعات مورد نیاز را در اختیار مشتری قرار می‌دهند. این پاسخ‌ها با کدهای وضعیت، سرایندها و محتوای پاسخ کامل شده و تعیین می‌کنند که درخواست با موفقیت انجام شده است یا خطا رخ داده است.

4-4-2 REST

REST³⁸ یک معماری نرم‌افزاری است که برای طراحی سامنه‌های توزیع‌شده و مبتنی بر وب استفاده می‌شود. REST بر اساس مجموعه‌ای از اصول و محدودیت‌ها ساختاردهی شده است که در تبادل اطلاعات بین سرور و کلاینت نقش

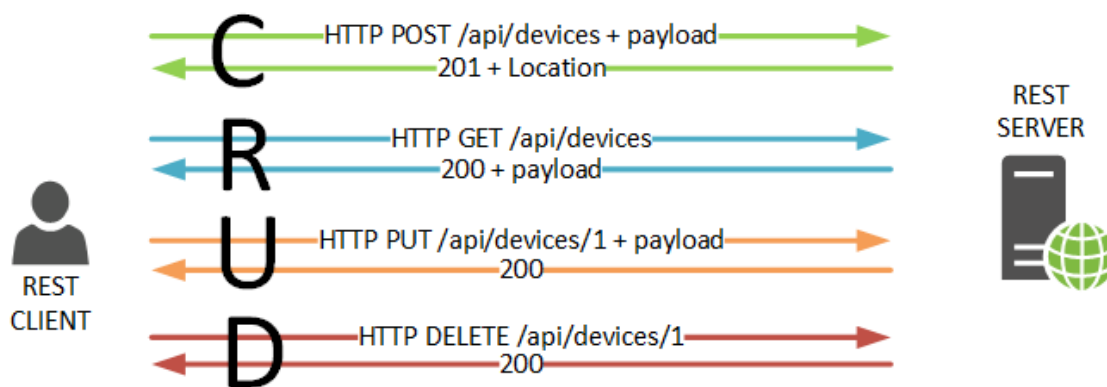
³⁵ HTTP Response

³⁶ Status Code

³⁷ Response Body

³⁸ Representational State Transfer

مهمی دارد. یکی از اصول اساسی، REST تعریف یکپارچگی³⁹ برای سامنه است. این به این معنی است که سامنه باید یک مجموعه‌ی مشخص از روش‌های استاندارد را برای تعاملات در نظر بگیرد. معمولاً در سامنه‌های REST، از متدهای HTTP مانند GET، POST، PUT و DELETE برای تعیین عملیات مورد نیاز استفاده می‌شود. همچنین، منابع⁴⁰ در سامنه REST به صورت یکتا شناسایی می‌شوند و آدرس‌های مشخصی برای آنها تعیین می‌شود. با توجه به این اصول و محدودیت‌ها، سامنه‌های REST قابلیت‌هایی مانند قابلیت توزیع، قابلیت مقیاس‌پذیری، انعطاف‌پذیری و قابلیت استفاده مجدد را فراهم می‌کنند. از طریق روش‌های استاندارد و اصول REST، سامنه‌ها قادر به تعامل با یکدیگر بدون توجه به جزئیات داخلی و پیچیدگی‌های نحوه پیاده‌سازی هستند.



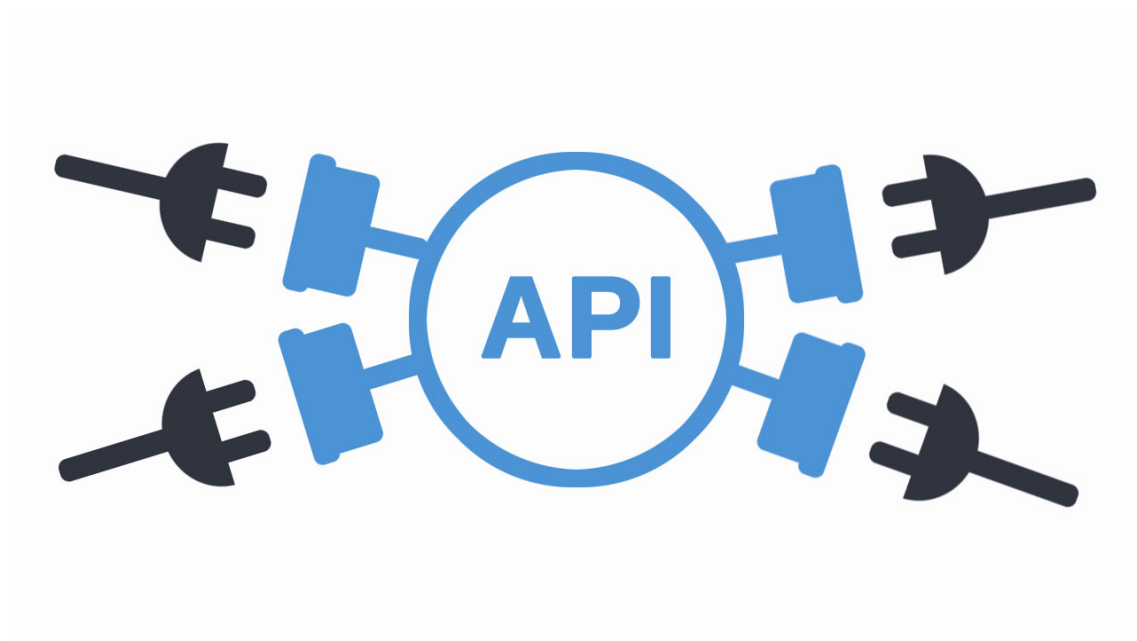
شکل 2-3: نمای کلی از REST

5-4-2 رابط کاربردی قابل برنامه‌ریزی

رابط کاربردی قابل برنامه‌ریزی⁴¹ به قوانین، پروتکل‌ها و دستوراتی گفته می‌شود که برای ارتباط و تعامل بین نرم‌افزارها، خدمات و برنامه‌های مختلف به کار می‌رود. به طور کلی، رابط کاربردی قابل برنامه‌ریزی نقش یک میانجی بین سامانه‌ها را بازی می‌کند و به برنامه‌نویسان اجازه می‌دهد با استفاده از آن، به منابع و امکانات موجود در سامانه دیگر دسترسی

Uniform Interface³⁹
Resources⁴⁰
Application Programming Interface⁴¹

پیدا کنند. رابط کاربردی قابل برنامه‌ریزی‌ها می‌توانند در دو شکل مختلف عمل کنند: به صورت وب خدمت یا به صورت کتابخانه برنامه‌نویسی. در حالت وب خدمت، رابط کاربردی قابل برنامه‌ریزی بر روی یک سرور میزبان شده است و از طریق پروتکل‌های اینترنتی مانند HTTP قابل دسترسی است. در حالت کتابخانه برنامه‌نویسی، دستورات و توابع مشخصی به برنامه اضافه می‌شوند که برنامه‌نویسان می‌توانند از آنها به عنوان قسمتی از برنامه خود استفاده کنند. استفاده از رابط کاربردی قابل برنامه‌ریزی‌ها به برنامه‌نویسان امکان می‌دهد که بخشی از دستورات را استفاده کنند و از خدمات و قابلیت‌های ارائه شده توسط یک سامانه دیگر بهره ببرند. این راهکار می‌تواند زمان و هزینه توسعه برنامه را کاهش داده و امکان ادغام بین نرم‌افزارها را فراهم کند. به طور خلاصه، رابط کاربردی قابل برنامه‌ریزی، مانند یک پل ارتباطی است که برنامه‌نویسان می‌توانند از آن استفاده کنند تا بین نرم‌افزارها و خدمات اطلاعات را به اشتراک بگذارند و تعامل کنند.



شکل 2-4: نمای کلی از رابط کاربردی قابل برنامه‌ریزی

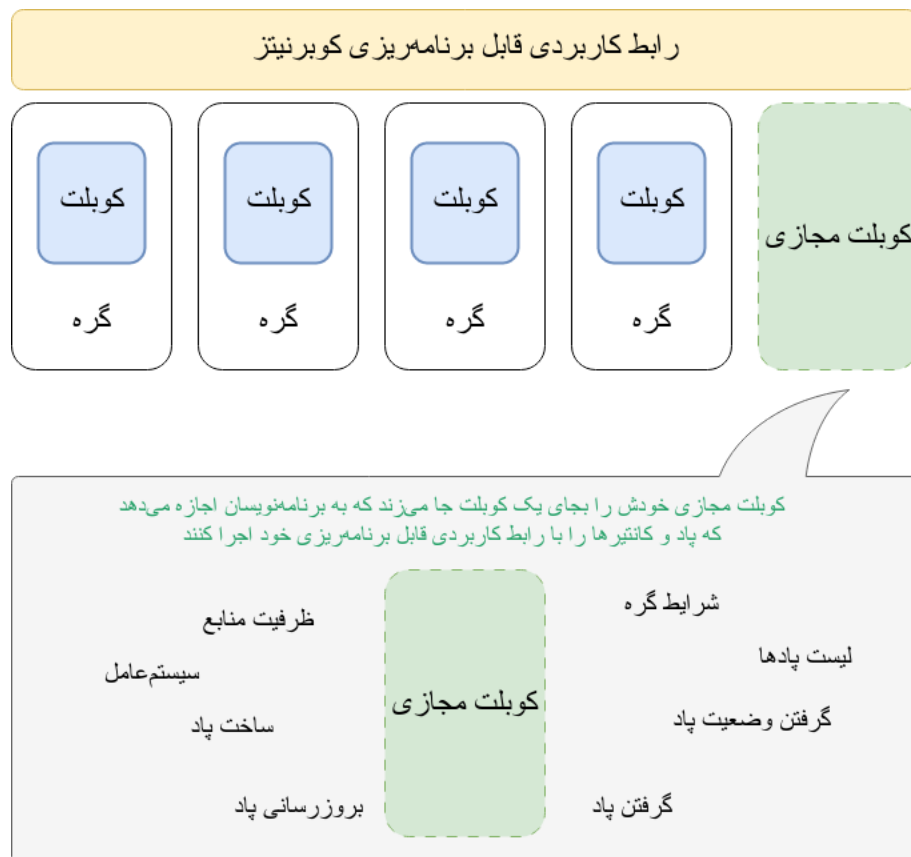
5-2 معماری فن آوت

معماری فن آوت⁴²: الگوی طراحی است که به طور معمول در سامانه‌های توزیع‌شده برای مدیریت همزمان یا موازی

درخواست‌ها استفاده می‌شود. این معماری به توزیع درخواست‌های ورودی به چند واحد پردازش، که به عنوان کارگرها شناخته می‌شوند، برای انجام عملیات مورد نیاز می‌پردازد. معماری فن‌آوت با بهره‌گیری از پردازش موازی و توازن بار، امکان مقیاس‌پذیری و بهبود عملکرد سیستم را فراهم می‌کند. در معماری فن‌آوت، هنگامی که یک درخواست دریافت می‌شود، آن را به چندین کارگر مستقل تکرار یا ارسال می‌کنند تا هر کدام از آن‌ها قادر به پردازش درخواست به صورت مستقل باشند. این رویکرد به چندین کارگر اجازه می‌دهد تا به صورت همزمان روی بخش‌های مختلف درخواست کار کنند و بهبود ظرفیت تولید و کاهش زمان پاسخ را به همراه داشته باشد.

6-2 کوبلت مجازی

کوبلت مجازی⁴³ یک پروژه متن باز است که با گسترش رابط کاربردی قابل برنامه‌ریزی کوبرنیتز، امکان ادغام⁴⁴ خدمات و سامانه‌های خارجی را در قالب گره‌های کوبرنیتز فراهم می‌کند. این پروژه این امکان را می‌دهد که یک گره "مجازی" بسازیم که توسط کوبرنیتز قابل مدیریت باشد و امکان ادغام سامانه‌های متنوع را در داخل خوشه کوبرنیتز به صورت یک پارچه فراهم کند.



شکل 2-5: کوبلت مجازی در یک خوشه کوبرنیتز

1-6-2 معماری کوبلت مجازی

کوبلت مجازی از چندین بخش کلیدی تشکیل شده است:

1. تامین‌کننده: تامین‌کننده مسئول پیاده‌سازی رابط کوبلت مجازی است و به عنوان پل ارتباطی بین سامانه یا خدمت خارجی و کوبرنیتز عمل می‌کند. این بخش، فراخوانی‌های رابط کاربردی قابل برنامه‌ریزی کوبرنیتز را به عملیات مناسب در سامانه خارجی ترجمه می‌کند.
2. گره: گره مجازی نمایانگر یک سامانه یا خدمت خارجی در خوشه کوبرنیتز است. عملکرد آن شبیه یک گره کوبرنیتز عادی است، اما به جای اجرا در زیرساخت فیزیکی، از طریق تامین‌کننده با سامانه خارجی ارتباط برقرار می‌کند.

3. پاد: پاد در کوبرنیتز، شامل یک یا چند کانتینر است. در کوبلت مجازی، پادها نماینده کارها⁴⁵ هستند که در گره‌های مجازی ایجاد شده توسط سامانه خارجی اجرا می‌شوند.

4. برنامه‌ریز⁴⁶: برنامه‌ریز کوبرنیتز مسئول تخصیص پادها به گره‌های موجود بر اساس نیازهای منابع و محدودیت‌ها است. در کوبلت مجازی، برنامه‌ریز مسئول زمانبندی پادها به گره‌های مجازی ایجاد شده توسط تامین‌کننده است. این بخش تضمین می‌کند که منابع بصورت بهینه استفاده شده و کارها به طور مناسب در سامانه خارجی توزیع شوند.

7-2 اینترنت اشیاء

اینترنت اشیاء⁴⁷ به مجموعه‌ای از دستگاه‌ها، سنسورها، دستگاه‌های هوشمند و شبکه‌های مرتبط که قادر به تبادل اطلاعات با یکدیگر از طریق اتصال به اینترنت هستند، اشاره دارد. این اشیاء می‌توانند شامل تلفن همراه و ساعت هوشمند تا لوازم خانگی هوشمند، خودروهای متصل و تجهیزات صنعتی باشند. اینترنت اشیاء با اتصال اشیاء و جمع‌آوری اطلاعات، امکان برقراری ارتباط و کنترل بیشتری را بین دنیای فیزیکی و دنیای دیجیتال فراهم می‌کند. مزیت اصلی اینترنت اشیاء در جمع‌آوری و تبادل داده‌ها است. سنسورها و دستگاه‌ها در اینترنت اشیاء می‌توانند اطلاعات مربوط به بستر، شرایط، موقعیت جغرافیایی، وضعیت و داده‌های دیگر را جمع‌آوری کرده و به سرورها یا سامانه‌های مرکزی ارسال کنند. این اطلاعات در سرورها تحلیل می‌شوند و می‌توانند به عنوان منبعی برای ارائه داده‌های مفید، تجزیه و تحلیل ترافیک، پیش‌بینی و اتخاذ تصمیم‌های هوشمند استفاده شوند. از جمله کاربردهای اینترنت اشیاء می‌توان به موارد زیر اشاره کرد:

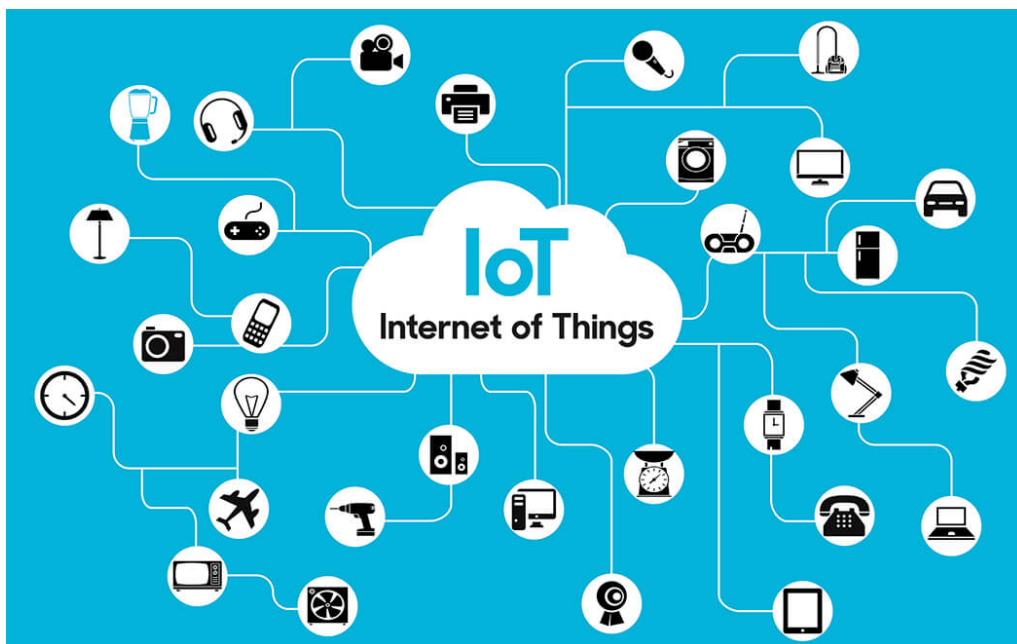
1. خانه هوشمند: اتصال لوازم خانگی مانند تلویزیون، سامانه‌های روشنایی، دستگاه‌های گرمایشی و سرمایشی، سامانه‌های امنیتی و سایر دستگاه‌ها به اینترنت به کاربران امکان می‌دهد تا این دستگاه‌ها را از راه دور کنترل و مدیریت کنند.

2. صنعت هوشمند: در صنعت، اینترنت اشیاء می‌تواند در جمع‌آوری داده‌ها از تجهیزات و سنسورها به منظور نظارت بر فرآیندها، پیشگیری از خرابی‌ها، بهینه‌سازی استفاده از منابع و افزایش بهره‌وری مورد استفاده قرار گیرد.

Workload⁴⁵Scheduler⁴⁶Internet of Things (IoT)⁴⁷

4. سلامتی هوشمند: استفاده از دستگاه‌های پوشیدنی، سنسورها و دستگاه‌های پزشکی هوشمند، امکان نظارت بر سلامتی فرد، جمع‌آوری داده‌های پزشکی، پیشگیری از بیماری‌ها و ارائه مراقبت بهتر را فراهم می‌کند.

به طور کلی، اینترنت اشیاء با اتصال اشیاء به اینترنت و جمع‌آوری داده‌ها، امکانات جدیدی را در دسترس قرار می‌دهد و قابلیت‌های جدیدی را برای کنترل، مدیریت و بهبود عملکرد اشیاء فراهم می‌کند.



شکل 2-6: نمای کلی از اینترنت اشیاء

8-2 جمع بندی

در این فصل به شرح مفاهیم پایه‌ای پرداخته شد که در مراحل مختلف انجام این پژوهش مورد استفاده قرار گرفته و برای درک کامل خواننده‌ی این نوشتار مورد نیاز است. در این فصل معماری‌ها و مدل‌های استفاده شده در این پژوهش به صورت جزیی مورد بررسی قرار گرفت. سپس به معرفی انواع مدل‌های موجود برای حل مسائل تصویر و زبانی پرداخته شد.

TODO

فصل 3

کارهای مرتبط

1-3 مقدمه

در این فصل به معرفی کارهای مشابه و شرح رویکردهای مورد استفاده برای حل مسأله‌ی پرسش‌وپاسخ تصویری می‌پردازیم. در ابتدا به بررسی روش‌های متفاوتی که برای حل مسئله پرسش‌وپاسخ تصویری ارائه شد و سپس به کارهای مشابه با تولید پاسخ در پرسش‌وپاسخ تصویری و مجموعه داده‌های موجود در این زمینه پرداخته شده است. در نهایت به مدل‌های از پیش آموزش داده شده در این زمینه می‌پردازیم.

فصل 4

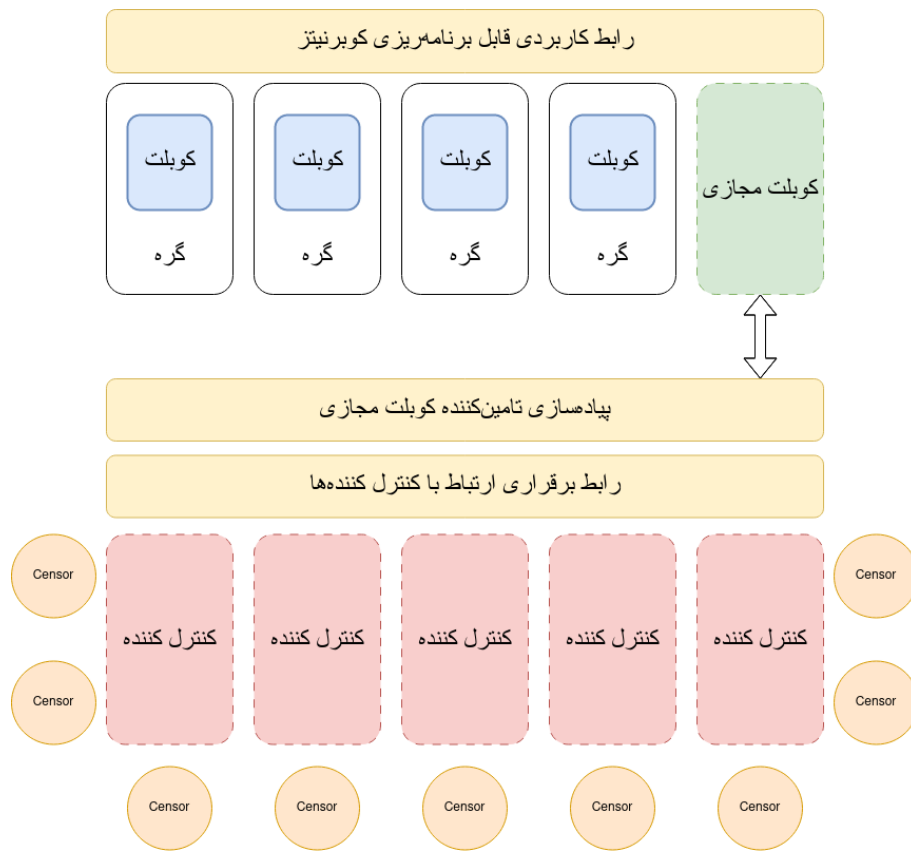
روش پیشنهادی

1-4 مقدمه

هدف اصلی پروژه امکان‌سنجی، طراحی، پیاده‌سازی و ارزیابی سامانه‌ای برای کنترل و پایش دستگاه‌های اینترنت اشیا با استفاده از امکانات کوبلت مجازی بر سکوی کوبنیتز است.

2-4 معماری سامانه

اجزای اصلی سامانه متشکل از پیاده‌سازی تامین‌کننده کوبلت مجازی، رابط بین تامین‌کننده و کنترل‌کننده‌ها، کنترل‌کننده‌ها و دستگاه‌ها می‌باشد که در ادامه به تعریف هرکدام پرداخته



شکل 4-1: نمای کلی معماری

3-4 نحوه استفاده

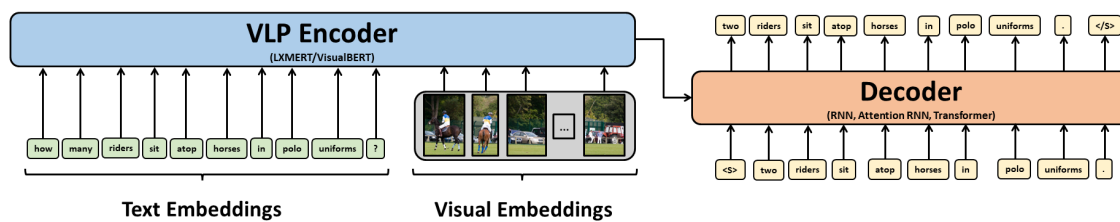
4-4 رابط کاربردی گرافیکی

5-4 مقدمه

هدف از این پژوهش انتشار روشی برای تولید پاسخ‌ها به صورت جمله است. برای این مورد، از شبکه‌های از پیش آموزش داده شده استفاده شده است. از LXMERT [?] به عنوان یک معماری دوجریان و از VisualBERT [?] به عنوان یک معماری تک جریان بهره‌برداری شده است.

6-4 معماری سیستم

برای حل این مسئله از معماری کدگذار-کدگشا استفاده شده است به گونه‌ای که از یک شبکه از پیش آموزش داده شده به عنوان کدگذار و از معماری‌های متفاوتی به عنوان کدگشا استفاده شده است.

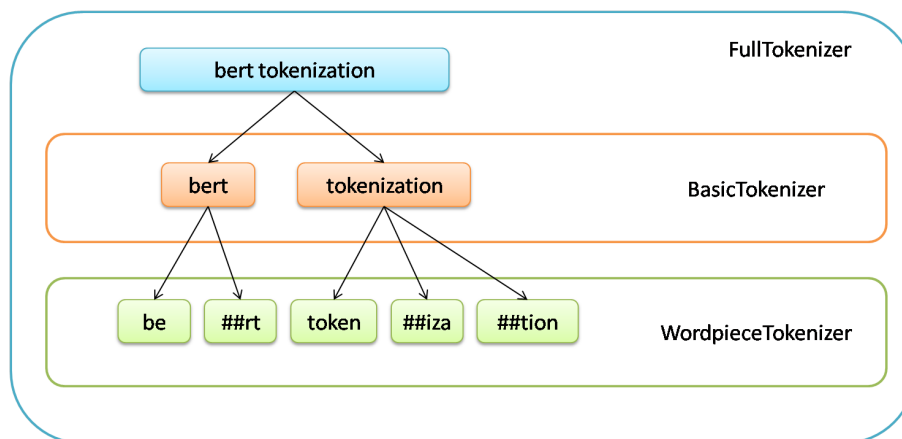


شکل 4-2: شمای کلی سیستم پرسش و پاسخ تصویری

7-4 ورودی اولیه و خروجی نهایی

در ورودی سیستم باید بتوانیم تصویر و متن را به سیستم ورودی بدهیم. برای انجام این عمل باید هر دو بخش متن و تصاویر را به بردارهای ویژگی تبدیل کنیم.

1. **بردارهای متن:** برای پردازش دقیق پرسش‌ها، لازم است که پرسش‌ها را به بخش‌های کوچک‌تری تقسیم کنیم که شبکه‌های عصبی عمیق قادر به اعمال محاسبات لازم باشند. به قسمت‌های کوچک‌تر اصطلاحاً توکن گفته می‌شود. به عمل جداسازی قسمت‌های یک جمله Tokenization گفته می‌شود. عکس این عمل که همان اتصال توکن‌ها و تشکیل جمله است را De-Tokenization نامیده می‌شود. در شکل 3-4 کارکرد این جداسازی مشاهده می‌شود. با توجه به اینکه هر دو شبکه LXMERT و VisualBERT بر پایه شبکه BERT هستند، برای بدست آوردن بردارهای ویژگی متن ورودی از جداساز شبکه BERT استفاده شده است.



شکل 3-4: نمونه‌ای از کارکرد Tokenizer

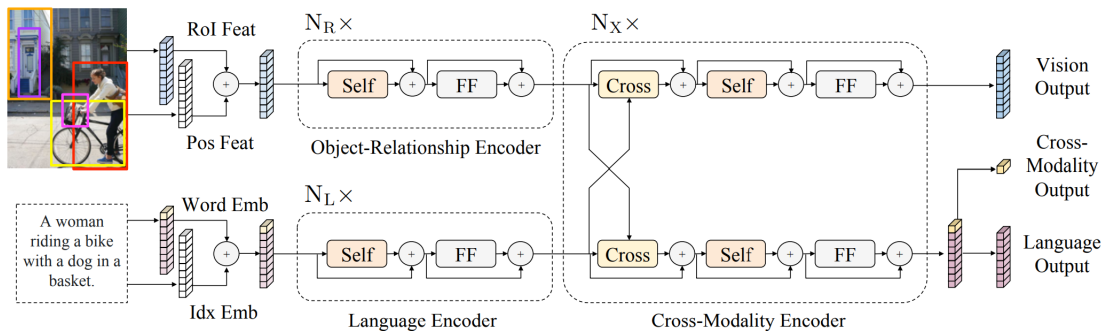
2. **بردارهای تصویر:** برای تبدیل تصاویر به بردارهای ویژگی مطابق با مقاله‌ای برای حل مسائل زبان-تصویر [?] می‌توانیم هر تصویر را مجموعاً به صورت 36 شیء در نظر بگیریم که از شبکه عصبی Faster-RCNN [?] تشخیص داده شده و هر شیء برداری با اندازه 2048 دارد.

8-4 مدل Encoder

از آنجایی که در بخش‌های قبل گفته شد، از مدل‌های از پیش آموزش داده شده استفاده می‌کنیم. به منظور مقایسه هر دو حالت تک‌جریان و دو‌جریان از دو مدل تبدیل شونده با معماری‌های متفاوت استفاده شده است تا بتوان مقایسه جامع و کاملی حول انواع مدل‌ها و عملکرد آن‌ها داشت. هدف از به کارگیری بخش کدگذار محاسبه یک بازنمایی از پرسش و تصویر همراه با یک‌دیگر و عبور دادن آن به بخش کدگشا است.

1-8-4 مدل LXMERT [?]

این مدل به عنوان یک مدل پایه برای حل مسئله پرسش و پاسخ تصویری ارائه شد. این مدل شامل سه بخش کدگذار تصویری، زبانی و میان‌ماژولی است. این مدل دو ورودی می‌گیرد، یک تصویر و یک متن که همان پرسش مربوط به تصویر است. با ترتیب لایه‌های توجه به خود و توجه میانی باعث می‌شود که مدل بتواند یک بازنمایی از تصویر، یک بازنمایی از متن و یک بازنمایی میان‌ماژولی از ورودی بدست بیاورد. محققان در این مدل از روش‌های متفاوتی نظیر MLM¹، MOP² و تناظر میان‌ماژولی³ برای آموزش استفاده کرده‌اند که باعث شده است روابط درون‌ماژولی و میان‌ماژولی به خوبی تشخیص داده شود. همانطوری که از تصویر 4-4 مشخص است این مدل یک مدل دو‌جریان است.

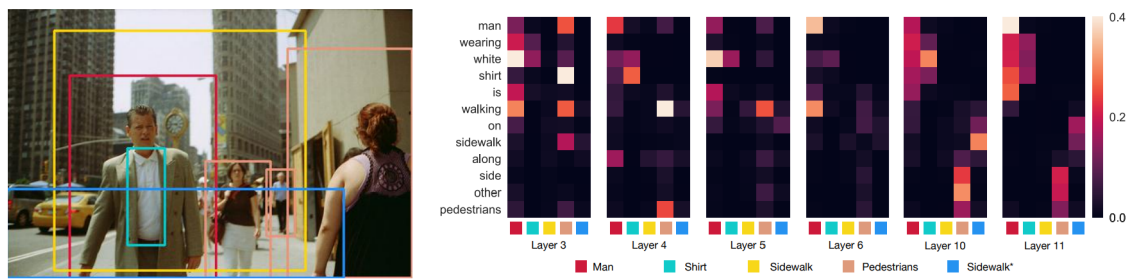


شکل 4-4: معماری مدل LXMERT همراه با ورودی و خروجی

¹Masked Language Modeling
²Masked Object Prediction
³Cross-modality Matching

2-8-4 مدل VisualBERT [?]

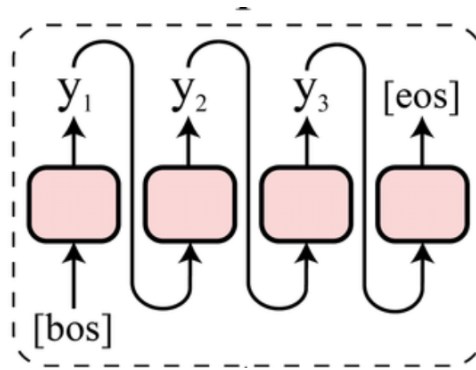
پس از انتشار LXMERT و پیشرفت چشمگیرش در حل مسئله پرسش و پاسخ تصویری پژوهش‌های بسیاری برای بهبود بیشتر انجام شد. از جمله این پژوهش‌ها بررسی مدل BERT و عملکرد آن در مسائل زبان-تصویر را می‌توان نام برد که منجر به انتشار مدل VisualBERT [?] شد. این مدل به صورت تک‌جریان است و دنباله‌ای از بردارهای ویژگی متن همراه با دنباله‌ای از بردارهای ویژگی تصویر را ورودی می‌گیرد و خروجی را به صورت بردارهای بازنمایی می‌دهد. به علاوه، میزان توجه بردارهای ویژگی توکن‌ها و بردارهای ویژگی اشیاء موجود در تصویر را با یکدیگر اندازه‌گیری کردند و اشیاء با توکن‌های مربوطه به خوبی توجه خورده‌اند. برای مثال با توجه به شکل 4-5 در لایه 11 کلمه man به خوبی با محدوده مربوط به مرد موجود در تصویر مرتبط شده است!



شکل 4-5: میزان توجه بخش متن به تصاویر در VisualBERT

9-4 مدل Decoder

پس از آنکه بردارهای بازنمایی تصویر و پرسش توسط کدگذار محاسبه شد، لازم است که آن‌ها را برای تولید پاسخ به بخش کدگشا بدهیم. بخش کدگشای معماری‌های ارائه شده از دو معماری شبکه‌های عصبی بازگشتی و شبکه‌های تبدیل‌شونده استفاده شده است. برای تولید پاسخ از مکانیزم Autoregressive استفاده شده است. در این مکانیزم توکن‌ها بر اساس توکن‌های قبلی پیش‌بینی می‌شوند و سپس همان توکن جدید همراه با دنباله قبلی برای توکن جدید دیگری مطابق با شکل 4-6 به کدگشا ورودی داده می‌شوند.



شکل 4-6: ساختار مدل Autoregressive Decoder

10-4 جمع بندی

در این بخش از نوشتار به بررسی دقیق اجزای تشکیل دهنده ی سیستم پیشنهادی برای حل مساله ی پرسش و پاسخ تصویری پرداختیم. روشی نو برای حل این مسئله که باعث رفع ابهامات بسیاری می شود. اشاره شد که از مدل های از پیش آموزش داده شده استفاده کردیم و همچنین برای مقایسه نتایج و بدست آوردن بهترین معماری، چندین حالت بررسی شد که در بخش 5 به مقایسه نتایج پرداخته خواهد شد.

فصل 5

ارزیابی روش پیشنهادی

1-5 مقدمه

در این فصل برای بررسی عملکرد روش پیشنهادی در فصل قبل، سیستم مورد نظر به طور کامل پیاده‌سازی شده و بر روی یک مجموعه داده‌ی شناخته‌شده اجرا شده است. برای ارزیابی روش پیشنهادی روش‌های متفاوتی ارائه شده است. برای انتخاب رویکرد مناسب ارزیابی از میان خیل عظیمی از روش‌های شناخته‌شده عوامل متعددی مورد بررسی قرار می‌گیرد و با توجه به آن‌ها یک یا چند روش انتخاب می‌شود. مهم‌ترین عامل ساختار و جنس داده‌های خروجی مدل و همچنین جنس داده‌های ارزیابی است. با توجه به مدل و مجموعه داده مورد استفاده، در این پژوهش نیاز به معیارهای ارزیابی برای مقایسه جملات داریم. از آنجایی که این روش، روشی نو است برای اثبات درستی آن از معیارهای ارزیابی متفاوتی استفاده کردیم که به طور کلی در دو بخش دسته‌بندی می‌شود.

فصل 6

نتیجه‌گیری و کارهای آینده

1-6 نتیجه‌گیری

در این پروژه ابتدا روش‌های مختلف حل پرسش و پاسخ تصویری مورد تحلیل واقع شد. سپس روشی نو برای حل آن ارائه شد که به درک انسان از سوالات و پاسخ دادن به آن‌ها نزدیک‌تر است. تولید پاسخ‌ها بدون هیچ دانش خارجی و به صورت جمله در پاسخ به سوالات مربوط به یک تصویر نسبت به پاسخ به صورت تک‌کلمه روشی معمول‌تر و رایج‌تر است و حل این مسئله به این روش می‌تواند سیستم را به روش‌های انسانی نزدیک‌تر کند.

در هسته در هسته‌ی اصلی سیستم پیشنهادی مدلی با معماری کدگذار-کدگشا قرار دارد. قسمت کدگذار که عمده وظیفه تحلیل داده ورودی را دارد از مدل‌های از پیش آموزش دیده در مسائل زبان-تصویر است که در این پژوهش از دو مدل LXMERT و VisualBERT استفاده شده است. مدل کدگشا نیز معماری‌های متفاوتی در نظر گرفته شد که مدل‌های تبدیل شونده بهترین عملکرد را داشتند و به دقت مناسبی بر مجموعه داده رسیدند.

برای ارزیابی هر معماری، چندین معیار در نظر گرفته شد که از ابعاد معنایی و نحوی مقادیر خروجی را ارزیابی می‌کنند. پس از ارزیابی برترین معماری‌ها، این نتیجه حاصل شد که استفاده از این معماری‌ها در مجموعه داده مدنظر مفید واقع شده و توانسته‌اند مسئله را با دقت خوبی حل کنند. سپس با روش‌های ارزیابی انسانی ثابت شد که این روش به رفع ابهامات در پرسش کمک بسیاری می‌کند.

در نظر گرفتن این نکته حائز اهمیت است که این دقت بالا در معیارها واقعی نیست زیرا مجموعه داده از داده‌های موجود در دنیای بیرون بسیار متفاوت‌تر است و به صورت خودکار تولید شده‌اند و حالت طبیعی خود را ندارند. لیکن این

به معنای حل کامل مسئله نیست و لازم به اعمال بسیاری برای ادعای حل کامل مسئله پرسش‌وپاسخ تصویری به صورت تولید متن است.

2-6 دستاوردها

در تمامی قسمت‌های این پژوهش اعم از قسمت‌های طراحی، پیاده‌سازی و ارزیابی نوآوری‌هایی مطرح شد که به شرح زیر است:

1. ارائه روشی نو برای حل مسئله پرسش و پاسخ تصویری که در صورت حل، روشی کامل‌تر از روش‌های فعلی است.
2. معرفی و استفاده از معماری‌های نو متناسب با روش پیشنهادی
3. اثبات درستی و فواید استفاده از این روش برای حل مسئله

3-6 کارهای آینده

همانطور که در قسمت ارزیابی مشاهده شد، با وجود اعداد و ارقام به نسبت بالا، همچنان نمی‌توان ادعا کرد که این مسئله حل شده است. هم‌چنین معماری‌های اراسه شده به نسبت قدیمی هستند و روش‌های بسیار نوآورانه‌تری معرفی شده‌اند که می‌توانند این مسئله را با سادگی بیشتری حل کنند. چندی از این ایده‌ها که بهبود این مسئله کمک بسیاری می‌کنند در زیر آورده شده است.

1. ارائه مجموعه داده‌ای جامع و کامل، به طوری که پاسخ‌ها به صورت جمله و طبیعی باشند. اولین قدم برای بهبود این سیستم ارائه مجموعه داده‌ی کامل‌تری است که بتوان معماری‌های پیچیده‌تر را بر آن آموزش داد و همچنین به نتایج بدست آمده و نزدیک بودن آن به دنیای واقعی اطمینان بیشتری داشت.
2. استفاده از کدگشاهای از پیش آموزش داده‌شده، یکی دیگر از راه‌های بهبود حل، ارائه معماری‌هایی غنی‌تر با پیچیدگی بالاتر است که قادر به حل مسئله به صورت جامع‌تر باشند. از این معماری‌ها می‌توان استفاده از مدل‌هایی نظیر BART و یا GPT را نام برد. استفاده از این قبیل معماری‌ها باعث عمومی‌سازی بیشتر سیستم می‌شود.

واژه‌نامه فارسی به انگلیسی

Visual Question Answering	پرسش و پاسخ تصویری
Deep Learning	یادگیری عمیق
Attention	توجه
Transfer Learning	یادگیری انتقالی
Error	خطا
Specification	مشخصات
Representation Learning	یادگیری بازنمایی
Recurrent Neural Network	شبکه عصبی بازگشتی
Encoder	کدگذار
Decoder	کدگشا
Sequence-to-Sequence	دنباله به دنباله
End-to-End	انتها به انتها
Transformer	تبدیل شونده
Tokenization	نشانه گذاری
Embedding	تعییه
Key	کلید
Query	پرسش
Value	مقدار
Multi-Head Attention	توجه چندسر
Masked Language Model	مدل زبانی ماسک دار
Forward Propagation	گسترش رو به جلو
Gradient Decent	گرادیان کاهشی
Self-attention	توجه به خود
Single-stream model	مدل تک جریان
Dual-stream model	مدل دو جریان
Word-similarity	شباهت نحوی

Human Evaluation ارزیابی انسانی

واژه‌نامه انگلیسی به فارسی

پرسش و پاسخ تصویری	Visual Question Answering
یادگیری عمیق	Deep Learning
توجه	Attention
یادگیری انتقالی	Transfer Learning
خطا	Error
مشخصات	Specification
یادگیری بازنمایی	Representation Learning
شبکه عصبی بازگشتی	Recurrent Neural Network
کدگذار	Encoder
کدگشا	Decoder
دنباله به دنباله	Sequence-to-Sequence
انتها به انتها	End-to-End
تبدیل شونده	Transformer
نشانه گذاری	Tokenization
تعبیه	Embedding
کلید	Key
پرسش	Query
مقدار	Value
توجه چندسر	Multi-Head Attention
مدل زبانی ماسک دار	Masked Language Model
گسترش رو به جلو	Forward Propagation
گرادیان کاهشی	Gradient Decent
توجه به خود	Self-attention
مدل تک جریان	Single-stream model
مدل دو جریان	Dual-stream model
شباهت نحوی	Word-similarity

..... Human Evaluation ارزیابی انسانی

Abstract

Visual Question Answering is a multi-modal task under the consideration of both the Vision and Language communities. Present VQA models are limited to classification answers and cannot provide answers for reasoning questions. In this work, we introduce an encoder-decoder model using vision-and-language pre-trained embedding, which delivers multi-word generated sentences as answers. We utilise LXMERT and VisualBERT embedding space with three different generative decoder heads, including RNNs, Attention RNNs and Transformers. Extensive experiments show competitive performance on the FSVQA dataset through qualitative and quantitative evaluation and a Human Error Analysis.

Keywords : Visual Question Answering, Natural Language Generation, NLG, VQA



Iran University of Science and Technology
Computer Engineering Department

Generate Answer to Visual Questions with Pre-trained Vision-and-Language Embeddings

Bachelor of Science Thesis in Computer Engineering – Artificial Intelligence

By:

Hadi Sheikhi

Supervisor:

Dr. Sauleh Eetemadi

Feb 2023