هوش مصنوعی

حسن عسکرزاده

قابل توجه دانشجویان و اساتید محترم
آخرین وضعیت اصلاح شده جزوه هوش و همچنین سوالات تستی و تشریحی ازسایت دانشگاه وآدرس www.askarzadeh.ir قابل استفاده است.جهت کسب اطلاعات بیشتر با شماره ۰۲۱۲۶۱۲۰۰۳۹ تماس بگیرید.

فصل ۱
هوش مصنوعی در یک نگاه

اهداف کلی

در پایان فصل، دانشجو با مفاهیم زیر آشنا می‌شود:

۱. هوش مصنوعی چه مفاهیم و تعاریفی دارد.

۲. هوش مصنوعی چگونه رشدکرد و تاریخچهٔ آن چیست.

۳. کاربردهای هوش مصنوعی در زندگی کدام‌اند.

٤. آیندهٔ هوش مصنوعی تحقق کدام یک از اهداف بشر را نشانه گرفته است.

۱.۱ نگاه کلی به هوش مصنوعی

رایانه‌های اولیه با محاسبات عددی در ارتباط بودند. رایانه‌های کنونی، علاوه بر محاسبات عددی، با استدلال بر مبنای دانش نیز درگیرند. با فناوری‌های هوش مصنوعی که آن را به اختصار AI[1] می‌نامیم نقش رایانه‌ها از وسیله‌ای مفید به وسیله‌ای ضروری و حیاتی تغییر می‌کند.

هدف AI این است که رایانه‌ها را به انجام کارهایی که بشر مایل است در آن ماهر باشد وادارکند. از جهتی AI سعی دارد رایانه‌ها را به رفتار زیرکانه‌تر وادارکند.

نام AI را جان مک‌کارتی در دههٔ ۱۹٦۰ ابداع کرد. او طراح زبان LISP بود. AI فاصلهٔ میان دانشمندان علوم رفتاری و دانشمندان رایانه را پرمی‌کند.

چیزی که در مورد رایانه می‌دانیم این است که مجموعه‌ای است از دستورات نوشته‌شده به زبان برنامه‌نویسی که همواره به طور دقیق اجرامی‌شود. بنابراین دانشمندان علوم انسانی می‌توانند نظریه‌های خود را در مورد رفتار بشر با تبدیل قوانین آنها به برنامه‌های رایانه‌ای محک‌بزنند و ببینند آیا رفتار رایانه در اجرای این برنامه‌ها شبیه رفتار طبیعی بشر یا حداقل زیرمجموعهٔ کوچکی از رفتار بشر است.

---

[1] Artificial Intelligence

دانشمندان علوم رایانه می‌توانند رفتار بشر را الگوی کار خود قراردهند و با مقایسهٔ آن با برنامه‌نویسی توانایی‌های خود را ارتقابخشند؛ اینکه مثلاً آیا می‌توانیم برنامهٔ رایانه‌ای مطابق با کاری که شخص انجام می‌دهد بنویسیم؟

نام AI نامی برخواسته از احساسات است، هر چند اکنون جاافتاده و بعید است تغییرکند. اینکه آیا موجودات بشری فقط «ماشین‌های گوشتی» خیلی پیچیده‌ای‌اند، آیا به لحاظ نظری امکان دارد AI واقعی ایجادکرد، آیا برنامه‌ای رایانه‌ای واقعاً می‌تواند مانند انسانی کامل رفتار کند، مباحث جذابی است. البته مسلم است که رفتار بشر به مراتب از آنالیزهای رایانه‌ای پیچیده‌تر است. اما، کسی که در حوزهٔ AI مطالعه می‌کند خود را درگیر این گونه باورها نمی‌کند.

در حقیقت، تنها واقعیت تصدیق‌شده در مورد AI، این است که موجودات بشری نسبت به چیزی که آنالیزهای رایانه‌ای در نظر می‌گیرند پیچیده‌ترند. پیشگویی‌هایی نظیر اینکه AI مشابه انسان به زودی ساخته می‌شود، اغلب فقط پیروی‌کردن از احساسات و عواطف رسانه‌هاست وتا این لحظه درستی آن ثابت نشده است.

چنانکه میدانیم برنامه ی فضایی که منجر به نشستن سفینه روی کره ماه شد ،هرگز به هدف خود یعنی اقامت انسان در ماه منجر نشد بلکه به پیشرفت تکنولوژی فضایی منجر شد. به نظر می رسد در مورد AI نیز چنین باشد یعنی منافع واقعی در تحقیق دراین زمینه به سمتی حرکت می کند که در کل باعث پیشرفت علم رایانه می شود.

در موارد خاص، تحقیقات AI اغلب با مشکلات برنامه‌نویسی خاصی مواجه شده است. بسیاری از مفاهیم در طراحی زبان برنامه‌نویسی و روش‌شناسی برنامه‌نویسی که اکنون در علوم رایانه‌ای عمومیت دارد از برنامه‌نویسی AI نشأت گرفته‌اند.

۲.۱ تعریف هوش مصنوعی

گفته می‌شود بسیاری از فعالیت‌های فکری بشر مثل نوشتن برنامه‌ها، فهمیدن زبان، درگیرشدن در برداشت استدلال، یا حتی رانندگی به هوش احتیاج دارد. متجاوز از چند دههٔ گذشته رایانه‌هایی برای انجام چنین وظایفی ساخته‌شده‌اند. به طور خاص سیستم‌هایی وجود دارند که می‌توانند به طور خودکار کد رایانه‌ای تولیدکنند، تفکیک و تجمع سمبولیک را انجام‌دهند، متنی را به اندازهٔ مشخصی درک‌کنند و مواردی جز آن.

در نمونه‌های ذکرشده گفته می‌شود سیستم‌ها درجاتی از هوش مصنوعی را دارایند.
با این مقدمه، می‌توانیم AI را از زوایای مختلف تعریف کنیم، از جمله:

- مطالعهٔ نحوهٔ وادارکردن رایانه‌ها به انجام چیزهایی که مردم در آن زمینه هم‌اکنون بهتر از رایانه‌هایند.
- شاخه‌ای از علم رایانه که با سمبل‌گذاری، و روش‌های غیرالگوریتمی حل مسائل سروکار دارد.
- بخشی از علم رایانه که به طراحی سیستم‌های رایانه‌ای هوشمند مربوط است. این سیستم‌ها خصوصیاتی دارند که فقط در رفتار موجودات هوشمند مانند انسان مشهود است.
- مجموعه‌ای از تفکرات یا مفاهیم فکری و راهی برای نگاه‌کردن و حل مشکلات از منظری خاص.

آنچه AI را از سایر علوم رایانه‌ای و مهندسی متمایزمی‌کند روش اکتشافی حل مسئله است. در این کتاب با مفاهیم، روش‌ها و فنون آن آشنا می‌شویم.
مشکلات AI طیف بسیار وسیعی دارد. این مشکلات در جزئیات و موارد خاص پدیدارمی‌شوند. فنون AI برای دست‌کاری سمبل‌ها همانند اطلاعات عددی طراحی شده‌اند که حوزهٔ وسیعی از آن‌ها را محققان AI گسترش داده‌اند. این کتاب محدوده‌ای از این فنون را معرفی می‌کند.

۳.۱ دلایل گرایش به هوش مصنوعی

رایانه‌ها انسان‌ها را از انجام بسیاری وظایف بی‌نیاز کرده‌اند. نمونه‌ای از این وظایف به شرح زیر است:

۱. **محاسبات عددی.** رایانه‌ها در محاسبات عددی به طور قطع سریع‌تر و دقیق‌تر از انسان‌هایند. برای مثال، رایانه می‌تواند ۱۸۹۲۵٤ را در ۸۲۹۰۹۹۷٤۳ ضرب‌کند و تقریباً نتیجه را فوراً بدهد؛ درحالی که بشر به طور متوسط ۲ دقیقه زمان لازم دارد تا همین عملیات را انجام دهد. همچنین در مورد انسان، امکان بروز خطا بیشتر است.

۲. **ذخیره‌سازی اطلاعات.** رایانه‌ها می‌توانند مقدار زیادی اطلاعات را ذخیره کنند. حجم محدودشده فقط به دلیل در دسترس‌بودن اطلاعات است. این یک تباین با بشر

است. در جایی که به نظر می‌رسد فقط مقدار معینی از مطالب می‌تواند ذخیره‌شود، رایانه‌ها کاراترند، چون بر خلاف انسان به سرعت می‌توانند مطالب را تکثیر یا در چند نقطه نگهداری کنند.

۳. عملیات تکراری. روشن است که بشر موقع انجام عملیات تکراری بی‌حوصله و مرتکب خطا می‌شود. رایانه‌ها به طور خاص برای انجام چنین کارهایی ساخته شده‌اند. رایانه‌ها هنگام انجام عملیات تکراری نه شکایت می‌کنند و نه خسته می‌شوند.

محاسبات اعداد، ذخیره‌سازی اطلاعات، و عملیات تکراری فعالیت‌هایی‌اند صرفاً مکانیکی و بدون نیاز به تفکر. انسان‌ها در انجام کارهای هوشمند بهتر از رایانه‌هایند.

بیش از هر چیز شاید بهتر باشد با معنای هوشمندی آشناشویم. هوشمندی مفهومی نسبی است و تعریف دقیق و مجردی ندارد. رفتاری که از نظر یک فرد هوشمندانه به‌نظرمی‌رسد، ممکن است برای فرد دیگری این‌گونه نباشد. اما، در مجموع خصوصیات زیر قابلیت‌های ضروری برای هوشمندی است:

- پاسخ به موقعیت‌های از قبل تعریف‌نشده با انعطاف خیلی بالا
- معنادادن به پیام‌های مبهم یا نادرست
- اختصاص اعتبار نسبی به عناصر در یک موقعیت
- پیداکردن شباهت‌ها، ولو اینکه موقعیت‌ها متفاوت باشند
- درک تمایزها بین موقعیت‌ها، ولو اینکه شباهت‌های بسیاری بین آن‌ها وجود داشته باشد.

به فرض اینکه تعریف بالا را از هوشمندی بپذیریم، موارد زیر فهرستی از وظایفی است که هوشمندی لازم دارد:

- تولید و درک گفتار
- تشخیص الگو
- حرکت در فضایی پر از موانع دینامیک
- اثبات قضیهٔ ریاضی
- استدلال.

٤.١ پیشینهٔ هوش مصنوعی

یکی از نتایج اولیهٔ کار با AI درک و تفکیک مسائلی است که برای هوش انسان سخت و برای هوش مصنوعی آسان است، و برعکس. ملاحظه می‌کنیم که بشر وظایف ریاضی فکری را با استفاده از ماشین حساب به سرعت انجام می‌دهد.

نوشتن برنامه‌های رایانه‌ای برای مرتب‌کردن پازل که نوعی سنجش هوش است آسان است. اما نوشتن برنامه‌هایی که بتوانند کارهایی را انجام دهند که بشر به طور طبیعی انجام می‌دهد بسیار دشوار است، حتی آن دسته از فعالیت‌هایی که به طور متوسط هوش کمی می‌خواهند. برای مثال، نوشتن برنامه‌های رایانه‌ای برای فهم جملات ساده، یا تفسیر تصاویر ساده کار دشواری است که تاکنون به طور کامل حل نشده است. رایانه ها ماشین های ضروری برای اجرا کـــردن دستورات به صورت سریع و با دقت هستند .

بشر با بینشی کارهای خود را انجام‌می‌دهد که روان‌شناسان هنوز کامل آن را درک نکرده‌اند. در مورد انسان دریافته‌ایم که تعداد محدودی قوانین یا موضوعات را می‌توانیم به خاطر آوریم، ولی هنگام حل مسائلی که همه صورت مسدله به صورت صریح بیان نشده است ، می توانیم همهٔ ترکیب‌های اطلاعاتی را که از تجربه در طول زندگی به‌دست آورده‌ایم به ذهن بیاوریم. برای مثال دو جملهٔ زیر را در نظر بگیرید.

– برای رنگ‌کردن پنجره روی میز ایستادم. **آن** شکست و من روی بوتهٔ گل افتادم.

– برای رنگ‌کردن پنجره روی میز ایستادم. **آن** شکست و من روی قالی افتادم.

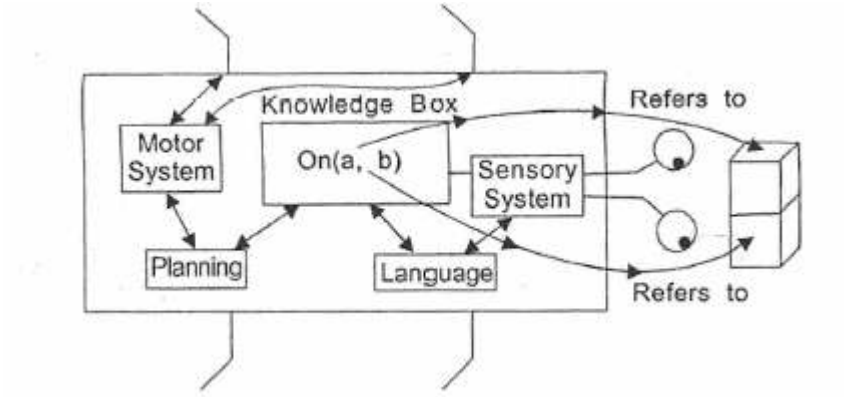
این مثال ساده یکی از نمونه‌هایی است که ابهام در زبان را نشان‌می‌دهد. مشخص نیست «آن» به چه اشاره دارد. در هر دو مورد «آن» ممکن است میز یا پنجره باشد. در چنین مواردی بافت یا جملات تکمیلی‌تر ابهام را برطرف می‌کنند. چنانچه در ادامهٔ جملات بالا گفته شود: «نباید بالای میز می‌رفتم» مرجع «آن» مشخص می‌شود.

به طور کلی درک و تفسیر جملات زبان طبیعی به درک مشترک از جهانِ پیرامون نیاز دارد که این خود پیچیدگی کار برنامه‌نویسی را دو چندان می‌کند. اگر برنامهٔ رایانه‌ایی به منظور تفسیر جملات زبان طبیعی و ذخیرهٔ اطلاعات لازم آن با استفاده از مجموعه‌ای از جملات منطقی بنویسیم، اطمینان از اینکه این برنامه همیشه «آن» را به چیزی که انسان به طورطبیعی تفسیر می‌کند ترجمه کند کار سختی خواهد بود.

درحقیقت، قراردادن قوانینی برای پوشش هر موقعیت ممکن، تقریباً ناممکن است.

۵.۱ تفاوت بین ارائهٔ سمبلیک و غیرسمبلیک

برای ساختن سیستم‌های هوشمند لازم است ابتدا مطابق با فعالیت‌های دستی درک‌شده از رفتار هوشمند از سمبل‌هایی که با فعالیت ادراکی مطابقت دارند استفاده‌شود. این ارائه‌های سمبلیک به دنیای خارج اشاره می‌کنند (شکل ۱.۱).

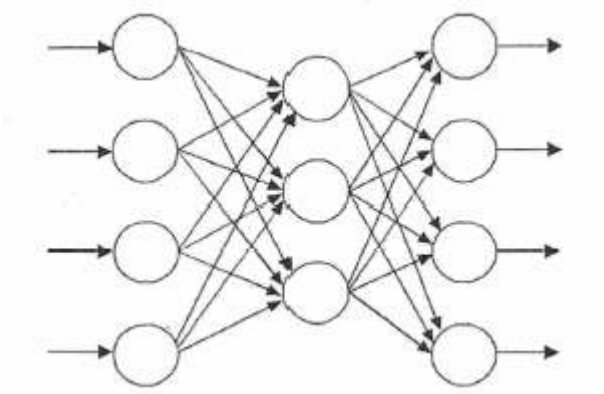


شکل ۱.۱ نمایش سمبلیک

مزایای ارائهٔ سمبلیک عبارت‌اند از:

ـ سازندهٔ سیستم می‌تواند چیزی را که سیستم می‌داند بخواند.

ـ آگاهی و دانش با جملاتی به زبان رسمی بیان می‌شود.
ـ خواندن ارائه و فهمیدن معنی دانش امکان‌پذیر است.
در ارائهٔ غیرسمبلیک آگاهی ما از موضوع با اوزان روی ارتباطات موجود در یک شبکه نشان داده می‌شود (شکل ۲.۱).



شکل ۲.۱ نمایش غیرسمبلیک

ارائهٔ غیرسمبلیک می‌تواند با ترکیباتی از خصوصیات، مثل یک تصویر ارتباط برقرارکند و تحمل‌پذیری بیشتری نسبت به نوفه دارند.

٦.١ تاریخچهٔ هوش مصنوعی

آلن تورینگ، وارِن مک‌کالوچ، کلود شانون، نوربرت وینِر، جان مک‌کارتی، ماروین مینسکی، نُوِل، سایمون، آرتور ساموئل، و دیگران افرادی‌اند که توانایی‌ها AI را ثابت کرده‌اند و فنون آن را نشان داده‌اند.

١.٦.١ تست تورینگ

یکی از مقاله‌های اولیه برای نشان‌دادن هوش ماشین را که به طور اخص در ارتباط با رایانهٔ دیجیتال امروزی است در سال ۱۹۵۰ آلن تورینگ، ریاضی‌دان انگلیسی، نوشت. تورینگ که بیشتر به دلیل همکاری در نظریهٔ محاسبه‌پذیری شناخته‌شده، به دنبال پاسخ به این سؤال بود که آیا ماشینی می‌تواند ساخته شود که فکر کند.
نکتهٔ مهم وجود ابهامات اساسی در این سؤال است. ابهاماتی چون فکرکردن چیست؟ یا

ماشین چیست؟ این ابهامات مانند سدی در مقابل هر پاسخ مستدل قرارداشت. او برای گذر از این سد پیشنهاد کرد که سؤال هوش با تست تجربی تعریف‌شده و واضح‌تری جایگزین‌شود.

تست تورینگ کارایی ماشین هوشمند را در برابر انسان اندازه می‌گیرد، چون انسان بهترین و تنها استاندارد رفتار هوشمند است.

این تست که تورینگ آن را «بازی تقلید» نامید، ماشین و انسان را در کنار هم و در اتاقی جدا از انسان سوم، که آن را محقق می‌نامد، قرار داد. محقق قادر نیست با هیچ کدام از آن دو (انسان و ماشین کنارش) به طور مستقیم صحبت کند و نمی‌داند کدام موجودیت واقعاً ماشین است. تنها با استفاده از ترمینال متنی می‌تواند با آن‌ها ارتباط برقرار کند. شاید امروز خوانندگان با ترمینال متنی (در مقابل ترمینال گرافیکی) آشنایی نداشته باشند، زیرا در آن زمان رایانه‌های شخصی به‌وجود نیامده بود و معمولاً افراد با استفاده از دستگاهی شبیه نمایشگرهای امروزی و با صفحه‌کلیدی که فقط وظیفهٔ ارسال و دریافت اطلاعات با رایانهٔ مرکزی را برعهده داشت نیازهای خود را مرتفع می‌ساختند.

از محقق خواسته شد از طریق ترمینال متنی سؤالاتی برای هر دو مطرح کند و براساس جواب‌هایی که دریافت می‌کند رایانه را از انسان تشخیص دهد. اگر محقق نتواند ماشین را از انسان تشخیص دهد، تورینگ می‌توانست استدلال کند که ماشین می‌تواند هوشمند فرض شود.

تست تورینگ با جداکردن محقق از رایانه و فرد شرکت‌کنندهٔ دیگر اطمینان حاصل می‌کند که تشخیص محقق با ظاهر ماشین یا مشخصهٔ مکانیکی صدا تحت تأثیر قرارنگرفته است. محقق در جهت تشخیص هویت رایانه آزاد است هر سؤال منطقی یا غیرمنطقی را مطرح کند. برای مثال محقق می‌تواند هم از رایانه و هم از انسان بخواهد یک محاسبهٔ ریاضی نسبتاً پیچیده را انجام دهند. به فرض اینکه احتمال درست بودن جواب رایانه نسبت به جواب انسان بیشتر باشد، در برخورد با این راهبرد رایانه نیاز دارد بداند چه موقع باید به چنین سؤالاتی جواب نادرست بدهد تا شبیه انسان به نظر آید.

برای کشف هویت انسانی براساس طبیعت احساسی ممکن است محقق از هر دو

سؤالی دربارهٔ یک شعر یا کار هنری بپرسد. رایانه برای پاسخ لازم است با احساسات و عواطف انسانی آشنا شود.

ویژگی‌های مهم این تست عبارت‌اند از:

الف) این تست مفهومی معقول از هوش است. برای مثال، رفتار موجودی هوشمند در پاسخ به مجموعهٔ خاصی از سؤالات مشخص می‌شود. این تست استانداردی برای تعیین هوش است.

ب) این تست ما را با سؤالات گیج‌کننده و بی‌پاسخ از منحرف شدن از مسیر اصلی منع می‌کند. در همه حال رایانه از پردازش‌های داخلی مناسب استفاده می‌کند و به هر حال ماشین واقعاً از اعمالش آگاه است.

ج) این تست با وادارکردن محقق تنها با تمرکزکردن بر محتوای پاسخ سؤالات هر محرک تشخیصی‌ای را که قابلیت تشخیص موجود زنده را فراهم می‌کند حذف می‌کند.

به دلیل این مزایا تست تورینگ تکیه‌گاهی برای بسیاری از طرح‌های واقعی در ارزیابی برنامه‌های AI امروزی است. برای ارزیابی هوشمندی یک نرم‌افزار که به استناد بعضی از شاخص‌های فنی هوشمند تشخیص داده شده می‌باید کارایی آن را در حل مجموعه‌ای از مسائل با انسانی خبره مقایسه‌کرد. این شیوهٔ ارزیابی فقط نمونه‌ای از تست تورینگ است.

از گروهی از انسان‌ها خواسته شده کارایی رایانه و انسان را در مجموعهٔ خاصی از مسائل مقایسه‌کنند. همان‌طور که خواهیم دید، این روش ابزاری ضروری هم در گسترش و هم در تأیید سیستم‌های خبرهٔ امروزی است.

بر تست تورینگ علی‌رغم درک مستقیم ظاهری‌اش انتقادهای هم شده است. یکی از مهم‌ترین انتقادها این است که به روش سمبلیک به حل مسئله می‌پردازد. اگرچه درک مستقیم جزء مهمی از هوش انسانی است، توانایی‌هایی که نیازمند قدرت ادراک یا مهارت باشند را آزمون نمی‌کند.

این در حالی است که گاه به نظر می‌رسد تست تورینگ هوش ماشین را وادارمی‌کند با قالب انسانی تطبیق‌یابد. هوش ماشین متفاوت از هوش انسان است و تلاش برای ارزیابی آن با شرایط انسانی اشتباهی اساسی است. آیا مایلیم ماشین عملیات ریاضی را به کندی و بی‌دقتی یک انسان انجام دهد؟ آیا ماشین هوشمند نباید سرمایه‌های خودش

از جمله حافظه، سرعت و جز آن را در تلاش برای رقابت با ادراک بشری افزایش دهد؟

در حقیقت تعدادی از صاحب‌نظران امروزی AI مانند فُرد و هایس در پاسخ به این چالش تست تورینگ را اشتباه و گمراهی بزرگ در پیشرفت نظریه‌های عمومی برای کشف و بیان سازوکارهای هوش در انسان و ماشین، و کاربرد این نظریه‌ها در دستیابی به ابزارهای حل مسائل عملی خاص می‌دانند. گرچه دانشمندان بسیاری با دغدغه‌های فرد و هایس موافق‌اند، هنوز تست تورینگ جزء مهمی در تأیید و اعتبار نرم‌افزار مدرن AI است.

تورینگ نظریهٔ امکان‌پذیر پیاده‌سازی یک برنامهٔ هوشمند را روی رایانهٔ دیجیتال نیز معرفی کرد. تورینگ با ارائهٔ الگوی خاص از ماشین محاسبهٔ حالت گسستهٔ الکترونیک تخمین مستحکمی در مورد ظرفیت ذخیره، پیچیدگی برنامه و فلسفهٔ طراحی اساسی مورد نیاز در چنین سیستمی را به وجود آورد. در نهایت، تورینگ نقایص معنوی، فلسفی، و علمی امکان ساختن چنین برنامه‌ای را بر حسب فناوری واقعی بیان کرد. مثال‌هایی هم در مورد همکاری انسان و برنامهٔ هوشمند وجود دارد که کارایی انسان را بهبود بخشیده است.

سیستم Dendral مثالی خوب برای این مورد است، در تشخیص ساختار شیمیایی مواد از طیف‌نگار جرم هر ماده., با استفاده از این نرم‌افزار قوانینی کشف شد که تا آن زمان افراد خبره نیز نمی‌شناختند.

تورینگ در مورد مفهوم هوشمندی ماشین، در دههٔ ۱۹۵۰ تست زیر را تدوین کرد: محقق می‌تواند سؤالاتی را با بیان یک سری جملات مفهومی به ماشین در یک اتاق و انسانی در اتاقی دیگر بپرسد. اگر محقق از جواب‌های دریافت‌شده قادر نباشد تصمیم بگیرد که کدام به ماشین وصل است، تورینگ نتیجه می‌گیرد که ماشین هوشمند است.

برنامهٔ تقلید مکالمه با روان‌شناسی به نام الیزا، از اولین تلاش‌ها برای ساخت برنامه‌ای بود که بتواند در تست تورینگ قبول شود. در نظر اول این برنامه کاملاً مؤثر می‌نماید، اما خیلی زود، حقه‌زدن به آن و در حقیقت نحوهٔ استنتاج و کارکردن با این برنامه به سرعت معلوم می‌شود. یکی از پیاده‌سازی‌های الیزا را در ادامه کتاب بررسی خواهیم کرد.

واضح است به منظور قبولی در تست تورینگ، سیستم باید زبان طبیعی را بفهمد. گفتار بشر کـــه همه نوع فرضیات و استنتاج‌هایی را به منظور معنادادن به جملات و سخنان انجام می‌دهند فعالیت پیچیده‌ای است. لذا چنانچه ماشین بخواهد همچون انسان از زبان طبیعی استفاده کند باید بتواند استدلال برداشت متعارف را با دسترسی به نوعی اطلاعات عمومی‌ای که انسان داراست ترکیب کند.

در طرح سیس (cyc) تلاش‌شده‌است اطلاعات پایه از جهان استخراج شود. تورینگ درسدد ساخت سیستم عظیمی بود که سعی می‌کرد تمامی دانش بشر را تسخیر کند. عرضهٔ این فناوری به شرکت سی‌کورپ واگذار شد. این گروه متونی را که اعتقاد داشت باید ماشین هوشمند از دنیا بداند را از دایـــرالمعارف‌ها، روزنامه‌ها، و نظایر آن استخراج و در پایگاه دانش درج می‌کرد. به طور معمول، چندین میلیون ورودی (مدخل) در این پایگاه دانش وجود دارد.

۷.۱ کاربردهای هوش مصنوعی

۱.۷.۱ پردازش زبان طبیعی

وقتی افراد با یکدیگر ارتباط برقرارمی‌کنند، تقریباً بدون تلاش خاص وخیلی پیچیده با یکدیگر تعامل و همکاری دارند و با اطلاعات ناچیز می‌توانند فرایندها را درک کنند. ساختن رایانه‌هایی که توانایی تولید و درک حتی کسر کوچکی از زبان طبیعی مثل انگلیسی را داشته باشند بسیار دشوار است. این دشواری به این دلیل است که زبان به رسانهٔ ارتباطی مؤثری در بین افراد باهوش بدل شده است. به نظر می‌رسد در نتیجهٔ انتقال ذره‌ای از ساختار فکری تحت شرایط محیط از مغزی به مغز دیگر، هر مغزی ساختارهای بزرگ فکری مشابه دارد. این مفاهیم, مفاهیم عمومی مشترک‌اند که درک زبان را ممکن می‌کنند. این تشابه در مفاهیم، به ایجاد و فهم پیام‌های مبهم کمک می‌کند. بنابراین، درک زبان‌های طبیعی مشکل پیچیده و بزرگی از رمزگذاری و رمزگشایی است.

یکی از اهداف همیشگی هوش مصنوعی، خلق برنامه‌هایی است که در درک و ایجاد زبان بشری توانا باشد. به نظر نمی‌رسد تنها توانایی استفاده و درک زبان طبیعی

خصوصیت بنیادی هوش بشری باشد، اما عدم کنترل و هدایت موفق دستگاه‌ها به طور خودکار، ضربه‌ای باورنکردنی بر قابلیت استفاده و سودمندی رایانه‌هاست. تلاش‌های بسیاری در نوشتن برنامه‌هایی که زبان طبیعی را درک‌کنند صورت گرفته است. اگرچه این برنامه‌ها در متون با لغات و زمینه محدود به موفقیت‌هایی دست‌یافته‌اند، سیستم‌هایی که بتوانند زبان طبیعی را با انعطاف‌پذیری و عمومیتی که بشر صحبت می‌کند استفاده کنند هنوز تولید نشده است، زیرا فراسوی گفتار با روش و چارچوب رایج است.

پیچیدگی درک زبان طبیعی خیلی بیشتر از تجزیهٔ جملات و جستجوی لغات در فرهنگ لغت است. ادراک واقعی به عوامل زیادی بستگی دارد، از جمله به وسعت دانسته‌های قبلی دربارهٔ قلمرو مباحثه و اصطلاحات استفاده شده در آن قلمرو و نیز توانایی به‌کاربستن آگاهی وابسته به قراین عمومی جهت رفع و حذف ابهاماتی که به صورت طبیعی در گفتار بشر وجود دارند.

بنابراین، به منظور ساخت سیستم‌های رایانه‌ایی که بتوانند زبان طبیعی را درک کنند، هم آگاهی وابسته به قراین و هم فرایندها جهت ساخت استنباط‌های مؤثر لازم است. محققان زیادی مجذوب این قلمرو خیلی مهم سیستم‌های هوشمند شده‌اند.

۲.۷.۱ بازیابی هوشمند از پایگاه داده

سیستم‌های پایگاه داده حوزهٔ وسیعی از حقایق دربارهٔ بعضی موضوعات را در بر می گیرد و شامل ابزاری برای جستجوی و بازیابی اطلاعات از آن می باشد.. .فرض کنید اطلاعات هنرپیشه ها در پایگاه داده هنرپیشگان وارد شده است ، ممکن است داده هایی نشاندهنده سن ، قد ، وزن ، تعداد کلاه ، کل درآمد ، ... در این پایگاه وجود داشته باشد . با این حقایق ذخیره شده در یک شکل منظم ، هر کسی می تواند پاسخ پرس وجو ها یی نظیر " لیست همه هنرپیشه گانی که قد آنان بیش از ۱۷۰ سانتی متر است ." یا "متوسط درآمد سالیانه هنرپیشه X چقدر است ؟" را به دست آورد . طراحی سیستم های پایگاه داده هنوز یک موضوع بسیار جدی در محدوده علم رایانه است . از نقطه نظر AI ، اگر برای یافتن پاسخ ها به استدلال های قیاسی با حقایق موجود در پایگاه داده ، نیاز داشته باشیم این میدان کاری حتی از این هم مهمتر خواهد شد .

برای ایجاد سیستم های بازیابی هوشمند داده ها تلاش ویژه ای صورت گرفته اما مشکلات فراوانی نیز وجود دارد. ,مشکل اول درک پرس و جوهای اظهار شده در یک زبان طبیعی نظیر انگلیسی است مشکل دوم اینکه ، با فرض اینکه سیستم , پرس و جوها را کاملاً درک کرده باشد ، چگونه باید پاسخ را از داده های موجود در پایگاه داده استنتاج کند.. مشکل سوم اینکه چگونه اطلاعات عمومی که بصورت صریح در پرس و جو یا حتی در پایگاه ذکر نشده را باید در نظر گرفت که به پاسخ مطلوب برسیم. برای مثال ، اگر هنرپیشه P کاپیتان باشد ، آنگاه سیستم باید بداند کاپیتان کسی است که تیمی را در یک میدان مسابقه راهنمایی می کند و عضوی از هسته گروه انتخاب کنندگان بازیکنان تیم است ،. چنین حقایقی ممکن است در طی توسعه پایگاه داده به صراحت اظهار نشده باشد .

۱-۳-۳ سیستم های خبره

سیستم های خبره ، سیستم های مشاور اتوماتیک هستند . این سیستم ها مشاوره خود را از طریق استنتاج های خبره راجع به محدوده هایی که در آن تخصص یافته اند را ارایه می کنند . به عنوان مثال سیستم های خبره طوری ساخته شده اند که می توانند عیوب موجود در سیستم های نظامی نظیر هواپیماها ، رادارها و سایر ادوات نظامی را تشخیص دهند ، نمونه های خاص از موجودات را به صورت علمی طبقه بندی نمایند، پیش بینی ایجاد ساختار های شیمیایی ممکن از مواد مختلف ارایه نمایند، عامل بالقوه سنگ معدنی که ته نشین می شود را شناسایی کنند، یا حتی امراض را با توجه به شواهد موجود تشخیص دهد .

در طراحی هر سیستم خبره برای حل هر مسئله دو کلید ترکیب شده وجود دارد . یکی از این کلیدها نشاندهنده دانش موجود از موضوع ودیگری کاربرد این دانش در هنگام ارایه نتایج است . نمایش دانش قدری پیچیده است زیرا که دانش می تواند مبهم و یا نامعلوم باشد . اساساً دانش به عنوان یک مجموعه بزرگ از قوانین ساده ارائه و در ساختارهایی که قالب۱ و متون اخباری ۲، نامیده می شود ، نمایش داده میشود . نتایج یا پاسخ عموماً از تکنیک های استنتاج قوانین پایه ای ، حاصل می شوند . در سال های

---

[1] Frame
[2] Script

اخیر شکل های دیگر استنتاج مانند استنتاج احتمالی به طور جدی گسترش یافته است . نرم افزار DENDRAL یکی از سیستم های اولیه برای به کار بردن دانش حوزه خاص جهت حل مسائل بود که در استنفرد۱ و در اواخر دهه ۱۹۶۰ توسعه داده شد. DENDRAL برای استنباط ساختار بنیانی مولکول ها از فرمول های شیمیایی شان وبا تکیه بر دانش وسیعی از تجزیه و روابط شیمیایی موجود در مولکول ها ، طراحی شده بود . چون مولکول های بنیادی میل دارند که خیلی گسترده باشند ، تعداد ساختارهای ممکن برای این مولکولها نیز تمایل به بزرگ و گسترده شدن دارند . DENDRAL با بکار بردن دانش اکتشافی شیمیدانان خبره و متخصص ، مسئله تعیین ساختار مولکولی یک ماده را از بین احتمالات گسترده محیط تحقیق را حل می کند . روشهای DENDRAL ، فوق العاده موثر بودند ، این برنامه تنها بعد از یک آزمایش کوتاه ، ساختار صحیح مولکولها را از میان میلیونها احتمال پیدا می کند . این عملکرد بکارگیری نسل های بعدی این سیستم را در آزمایشگاه های شیمیایی و دارویی در سراسر دنیا تضمین نمود .

البته DENDRAL ، یکی از اولین برنامه ها در استفاده موثر دانش ماشینی در حوزه خاص جهت دست یافتن به هدف حل مسائل در سطح دانشمندان خبره بود ، MYCIN نمونه دیگری از برنامه است که با رویکرد ایجاد برنامه خبره در حوزه خاص پایه گذاری شد. MYCIN از دانش متخصصین و خبرگان پزشکی جهت تشخیص و تعیین درمان تورم ستون فقرات و عفونت میکربی در خون استفاده می کرد.

MYCIN که در استنفرد و اواسط دهه ۱۹۷۰ توسعه داده شد ، یکی از اولین برنامه ها جهت نشان دادن مشکلات استدلال با اطلاعات نامعلوم و ناقص بود . MYCIN شرح وتفسیر استدلالش را واضح و منطقی تهیه کرد و از ساختار کنترل مناسب برای حوزه مسائل خاص و ضوابط شناخته شده جهت ارزیابی کارایی اش به طور قابل اعتماد ، استفاده کرد.

۱-۳-٤ اثبات قضیه

اثبات یا رد قضایای ریاضی کاری است کاملاً فکری است و به استنتاج هایی از فرضیات اولیه نیاز دارد و علاوه بر آن با حکم هم درگیر می باشد . این حکم بر روی

---

[1] Stanford

حجم وسیعی از اطلاع تخصصی ، پایه گذاری شده است و برای رسیدن به حدس صحیح راجع به چیزی که قبلاً ثابت شده است (قضیه) تلاش فکری باید صورت گیرد، این موضوع کمک میکند که مسایل را با توجه به قضایای موجود به مسایل کوچکتر شکسته و مستقلا آنها را حل و سپس مجددا ترکیب کنیم . برنامه های زیادی برای اثبات ماشینی قضایا توسعه یافته است که توانایی کار کردن به طور مستقل را دارند . برای درک اجزاء استدلال از تکنیک های فرمول بندی قیاسی که از زبان منطق گزاره ای استفاده می کند کمک میگیریم . فعالیتهای بدون قاعدنه زیادی مانند تشخیص پزشکی را میتوان با روش اثبات قضایا فرمول بندی کرد .

با توجه به اینکه استدلال منطقی روش عمومی و رسمی برای حل مسایل است جاذبه زیادی را برای اثبات قضایا ایجاد کرده است ، منطق خود به فرآیند مکانیزه شدن کارها منجر می شود. طیف وسیعی از مسائل وجود دارند که ما آنها را از طریق نشان دادن خود مسئله و اطلاعات زمینه ای آن , با استفاده از اصول منطقی و نمونه های قضایایی که اثبات شده اند حل میکنیم . یک دلیل دیگر برای علاقمندی اثبات قضیه ها بصورت اتوماتیک ، تفهیم مسائل به رایانه هاست .همانطور که میدانیم چنین سیستم هایی ، توانایی حل مسائل خیلی پیچیده را بدون کمک بشر و به طور مستقل ندارند . اما بصورت کاربردی در اثبات بسیاری از قضایای امروزی ، به عنوان دستیاران هوشمند انسان عمل کرده و وظایف سخت و دقیق مانند تجزیه مسائل بزرگ به مسائل کوچکتر و استنتاج اکتشافی برای جستجو در فضای استدلال منطقی را انجام دهند . بنابراین اثبات قضیه ساده تر انجام می شود .این سیستمها هنوز به دنبال انجام اثبات اصل موضوع ، بازبینی تخمین های کوچکتر ، و کامل کردن خصوصیات رسمی یک اثبات طرح ریزی شده هستند که با همکاری و مشارکت با انسان صورت میپذیرد وبه همین دلیل حل قضایا یک زیر بخش مهم از AI است .

۱-۳-۵ رباتیک

انسانها قادرند در محیط شان جابجای یا تغییر صورت دهند مثلا جعبه اسباب بازی را جابجا کنند یا وضعیت خاموش و روشن بودن کلید را تغییر دهند. اگرچه به طور ناخودآگاه این اعمال به وسیله انسان انجام می شود ، اما انجام آن با پیچیدگی ها ی فراوانی همراه است . وقتی تلاش می کنیم برنامه ای که رفتار مشابه را در ماشین های

هوشمند بوجود آورد را بنویسیم ، درمی یابیم که این کار نیازمند توانایی های بسیاری است که فقط در حل مسائل بسیار عقلانی تر وجود دارد.

برنامه ریزی حالت ,یکی از مهمترین گامها در فرآیند طراحی رباتی است که کارهایش را در رابط با جهان خارج با درجه ای از حساسیت و انعطاف پذیری انجام میدهد به طور خلاصه ، در برنامه ریزی ، رباتی را در نظر میگیریم که قابلیت انجام اعمال تجزیه ناپذیر معین را مانند حرکت به جلو , حرکت به چپ و ... دارد . فرآیند برنامه ریزی تلاش میکند ترتیبی از اعمالی را بدست آورد که وظایف سطح بالاتر نظیر حرکت از یک سو به سوی دیگر در یک اتاق پر از مانع را امکان پذیر کند . مانند رسیدن به ترتیب مناسب گرفتن کلاچ , قرار دادن دنده در وضعیت حرکت به جلو و فشار گاز و و رها کردن کلاچ برای رسیدن به هدف راه افتادن خودرو از نقطه سکون .

به دلایل زیادی برنامه ریزی کار سختی است ، نه بدلیل اینکه حالتهای حرکات پیوسته ی ممکن زیاد است زیرا حتی یک ربات بی نهایت ساده هم توانایی انجام تعداد زیادی از حرکتهای پیوسته را به صورت بالقوه دارد . برای مثال ، رباتی را که می تواند به سمت جلو ، عقب ، راست یا چپ حرکت کند ، تصور کنید ، و چگونگی راه های مختلفی که ربات احتمالاً می تواند اطراف اتاق حرکت کند ،همچنین فرض کنید که در اتاق موانعی وجود دارد که ربات مجبور است مسیری را انتخاب کند که با یک روش موثر و کارآمد آن را اطراف اتاق حرکت دهد . نوشتن برنامه ای که بتواند به طور هوشمندانه بهترین مسیر را بدون دست پاچه شدن به سبب زیاد بودن تعداد احتمالات و تحت شرایط محیط پیدا کند کار ساده ای نیست و می باید ربات به تکنیک های مهارتی برای ارائه دانش فاصله ها و کنترل جستجوی بهترین راه از کل راه های ممکن ، مجهز باشد .تحقیق در مورد رباتیک به توسعه وگسترش بسیاری از تکنیک های AI و مدلسازی از دنیا منجر شده است

۱-۳-٦ مسائل زمان بندی و ترکیبی

مثال کلاسیک مسئله مسافرت فروشندگان دوره گرد را به در نظر بگیرید ، جائی که صورت مسئله ، پیدا کردن مسیری که فروشنده از همه شهرها فقط یکبار عبور کرده و کمترین مسافت را هم طی کند و به مبدا بازگردد می باشد ، به استفاده از تئوری گراف

حل مسئله یافتن مسیر با کمترین هزینه روی لبه های گرافی شامل n گره[1] می باشد در حالی که هر گره بیش از یک بار ملاقات نشود. در چنین مسائلی دامنه ترکیبات یا ترتیبات احتمالی از چیزی که یک دسته جواب با حالتهای متعدد دارد ، انتخاب می شود .برای حل مسئله توسط این راه حل ها حتی منابع بزرگترین رایانه ها نیز جوابگو نیست Brute force سعی کرد روشی را بوجود آورد که به مشکل کمبود منابع فایق آید. در نظریه محاسبات به چنین مسائلی مسائل " NP کامل" گفته می شود .

محققین AI روی روشهایی برای حل گونه های مختلف از مسائل ترکیبی ، کار کرده اند . کلید حل چنین مسائلی ، اطلاع دربارهٔ دامنهٔ مسئله می باشد . روشهای توسعه داده شده ای که کارایی آنها جهت حل مسائل ترکیبی ای اثبات شده اند ، به حل مسائل در حوزه هایی که حالت ترکیبی کمتری دارند نیز کمک موثری نموده است.

۱-۳-۷ مسائل ادراکی

تلاشهای زیادی برای ساختن رایانه هایی که ببینند و بشنوند صورت گرفته است . هر چند که با بالا رفتن سرعت پردازش رایانه ها و ارزان شدن حافظه و توسعه الگوریتمها جدید روز به روز نمونه های موفق تری پیاده سازی میشود اما توفیق در این زمینه مستلزم درک و پردازش داده های ورودی پیچیده و داشتن دانش وسیعی در خصوص اشیاء تمیز داده شده است

پردازش ادراکی به عنوان مجموعه ای از عملیات مورد مطالعه قرار گرفته است . یک منظره بصری به وسیله گیرنده های حسی درک .و توسط یک ماتریس که درایه های آن اعداد وزن دهی شده ی بدست آمده از گیرنده های حسی است نمایش داده می شود (تصاویر بیت مپ). حال برای پردازش باید تصویر به اجزایی که سوژه های مختلف در آن قابل شناسایی باشد تفکیک شود .سگمنت های خطی ، منحنی های ساده ، زوایا در شکل جستجو و پردازش شوند در این حالت به جای پرداختن به جزئیات یک تصویر طرح کلی اجزاء تشکیل دهند مشخص میگردد ..این همان فعالیتی است که چشم انسان با کمک مغز انجام میدهد . در چشم انسان نیز سلولهای استوانه ای مسئول تشخیص اجزاء تصویر بوده و سلولهای مخروطی مسئول تشخیص رنگ هستند اما بسیاری از سلولهای استوانه ای با کمک مغز به جای درک جزئیات تصویر , تصاویر لبه

---

[1] Node

دارد و بخش های متحرک تصویر را تشخیص می دهند و این کار را با قاب بندی تصاویر حک شده در شبکیه انجام میدهند. در نهایت یک مدل کلی مثلا یک تپه با یک درخت روی آن درک میشود .

برای اینکه بتوانیم به درک کلی برسیم باید یک نسخه خلاصه شدهٔ ورودی از مقادیر وسیع داده ورودی خام بدست آوریم که قابل مدیریت کردن نیز نیستند. اشکال اصلی در مسائل ادراکی ، تعداد بسیار زیاد توصیف های کاندید از یک منظره است. یک راه حل برای این مورد ، استراتژی ساختن فرضیه از هر توصیف و سپس تست کردن آنها به منظور رسیدن به هدف است. برای تشکیل این فرضیه ، دانش زیادی در بارهٔ مناظر مورد انتظار نیاز است . که کار را بسیار دشوار میکند .

۱-۳-۸ معماریهای وابسته به سلسله اعصاب .

شناخت و پیاده سازی معماری های عصبی به عنوان مکانیزمهایی برای پیاده سازی هوش برای بعضی از اهداف جذاب است . برنامه های سنتی AI بی دوام و خیلی حساس به پارازیت و داده های غیر متعارف هستند ، چنین برنامه هایی تمایل دارند یا درست باشند یا اینکه کلاً رد شوند .

هوش انسان انعطاف پذیرتر است. ما در تفسیر ورودی های مبهم مثل تشخیص یک چهره در یک اتاق تاریک از یک زاویه عجیب یا تفکیک ورودیهای پر نویز مانند دنبال کردن یک مکالمه دریک مهمانی شلوغ ماهر هستیم . انسانها اغلب جایی که قادر به حل مسئله ای نیستند ، برای رسیدن به راه حل یک حدس معقول به راه حل میرسند.

چون معماری های عصبی ، دانش را در واحد های با قطعات کوچک ذخیره می کنند به نظر پتانسیل بیشتری برای تطبیق جزئی داده ی پر نویز و ناکامل را دارند. همچنین معماری های عصبی محکم و قابل اعتماد هستند زیرا دانش تا حدی به طور یکنواخت درشبکه توزیع شده است . به علاوه ، معماری عصبی یک مدل طبیعی برای مشابهت تفکر انسان و رایانه را تأمین می کند ، چون هر رشته عصبی یک واحد مستقل است .

Hillis ثابت میکند مغز بشر وقتی دانش بیشتری در خود ذخیره کرده سریعتر عمل میکند اما رایانه ها بر عکس هستند یعنی با افزایش دانش و بزرگ شدن اندازه پایگاه اطلاعات یا پایگاه دانش سرعت رسیدن به راه حل به شکل محسوسی کم میشود این کم شدن سرعت به منظور تامین هزینه زمانی جستجوی ترتیبی یک پایگاه دانش است .

یک معماری کاملا شبیه مغز انسان از این مشکل رنج نمی برد .

۱-۳-۹ بازی کردن

بیشتر تحقیق اولیه در وضعیت فضای جستجو با استفاده از بازیهای تخته ای نظیر تخته نرد ، شطرنج و معمای ۱۵ تکه انجام شده است . بازیهای تخته ای علاوه بر ظاهر عقلانی خود ، خاصیت های معینی دارند که آنها را موضوعات ایده آل برای کار اولیه ساخته است .

اغلب بازی ها با استفاده از مجموعه قوانین خوش- تعریف انجام می شوند : این قوانین تولید فضای جستجو را آسان می کند و محققان را از ابهامات زیاد و پیچیدگی های ذاتی در مسائل کمتر ساخت یافته رها می سازد . شکل های تخته استفاده شده در این بازیها به آسانی روی یک رایانه بدون نیاز به رعایت ظاهر پیچیده لازم قابل ارایه است و نیازی نیست مانند درک تصویر اقدامات هوشمند دیگری برای درک حالتهای آن صورت گیرد . در بازی ها فضاهای جستجوی خیلی بزرگتری تولید میگردد این بازی ها به قدری بزرگ و پیچیده هستند که نیازمند تکنیک های قدرتمند برای یافتن راهی به منظور کاوش در فضای مسأله می باشند. این تکنیک ها مکاشفه ای نامیده می شوند و محدوده ی وسیعی از تحقیقات AI را تشکیل می دهند. به نظر می رسد چیزی که ما آنرا هوش می نامیم ، در مکاشفه های صورت گرفته توسط انسانها برای حل مسائل ، نهفته شده باشد.

۱-٤ اهداف AI

اهداف اصلی تحقیق AI عبارتند از :

- فهمیدن ادراک انسان . مثلاً چگونه انسان ها مسائل را حل می کنند؟ سعی برای بدست آوردن دانش عمیق حافظه انسان ، توانایی های حل مسأله ، آموزش و تصمیم گیری و....
- خودکار سازی فعالیتهای اقتصادی ، انسانها را در وظایف هوشمند با ماشین و رایانه جایگزین می کند. آیا برنامه هایی که به خوبی بشر کاری را انجام می دهند ، وجود دارند ؟
- توسعه هوشمند فعالیتهای اقتصادی عامل تولید سیستم های هوشمند جدیدی است تا به انسانها کمک می کنند که بهتر ، سریعتر و عمیق تر فکر کنند برای مثال

سیستمی که به یک رایانه همه منظوره کمک می کند تا بیماریها را تشخیص دهد.

● هوش فوق بشری : آبا برنامه ها و رایانه هایی ساخنه خواهد شد که از هوش انسان برتر باشند. ؟

●حـــل مسأله عمومی، تولید سیستم هایی به وسعت ذهن حوزه وسیعـــی از مسائل را حل مـی کند.

● مباحثه دارای نتیجه منطقی با انسانها که در آن ماشین با انسان با زبان طبیعی و در یک مکالمه هوشمند ارتباط برقرار می کند.

● خود مختاری ، سیستم های هوشمندی که از خود ابتکار به خرج داده و به دنیای واقعی عکس العمل نشان می دهند.

● آموزش و استقراء : سیستم باید علاوه بر گرد آوری داده هایی که خود آن را جمع آوری کرده ، قادر باشد به آنها عمومیت دهد ، فرضهای جدید مطرح کند ،با موارد مشابه نطبیق دهد ، برهان کند.

● ذخیره اطلاعات جدید و دانستن چگونگی بازیابی آن .

۱-۵ برنامه نویسی هوش مصنوعی

رهیافت مهندسی نرم افزار مرسوم به برنامه نویسی به طی مراحلی شامل شناخت وضع موجود ، نیازسنجی ، تطبیق خواسته های جدید با وضع موجود ، شکستن مسأله به قسمت های کوچکتر و در مرحله آخر به کد کردن تأکید دارد . به هر حال برنامه نویسی سیستم های هوش مصنوعی به راحتی با مدل های مهندسی نرم افزار مرسوم قابل تطبیق نیست . مسائل AI را به دلیل طبیعتشان نمی توان به طور دقیق و رسمی مشخص کرد . ارزیابی عملکرد یک سیستم AI با تطبیق کارایی آن با ویژگیهای رسمی صورت نمیگیرد ، بلکه با سنجش نزدیکی کارایی آن به کارایی انسان اندازه گرفته می شود. در نتیجه مرحله برنامه نویسی ، به عنوان رفتاری « اکتشافی » یا « شکل اولیه » تلقی می شود. یعنی برنامه ها به روشی کمتر رسمی ، نسبت به مهندسی نرم افزار مرسوم توسعه داده می شوند ، در این حالت برنامه های تولید شده ممکن است به عنوان الگوهای تجربی نسبت به محصولات نیمه کامل مورد توجه قرار گیرند . یک برنامه AI ممکن است با محدوده ای از مثال های انسانی آزمون شود و در صورت لزوم طوری اصلاح شود که مثل یک انسان روی آن مسائل عمل کند. ممکن است

موفق شویم با بعضی از اصلاحات کارایی ماشین را در موارد خاصی به کارایی بشر نزدیک کنیم.اما این اصلاح ممکن است در بقیه موارد کارایی را افزایش ندهد . به هرحال ، اگر در هر مورد خاص اصلاحی برای پوشش دادن حالت خاص به برنامه اضافه شود باعث پیچیده تر شدن و کندتر شدن برنامه خواهد شد . در موارد زیادی برنامه تولید شده به عنوان یک برنامه مقدماتی بکار گرفته شده و کنار گذاشته خواهد شد اما درس های آموخته شده از ساخت آن ، نگهداری شده اند و برای ساخت یک برنامه بهتر استفاده می شوند. این پـــردازش مـمکن است در یک چرخه چندین بار تکرار شود ، هر چرخه یک الگوی بهتر می سازد. این رهیافت با عبارت برنامه اجرا ـ فهم ـ اشکال زدایی ـ ویرایش (RUDE[1]) توصیف میگردد (افاو). اگرAI را به عنوان نمونه ای از برنامه نویسی با روش RUDE در نظر بگیریم ، آنگاه نرم افزار بازی شطرنج را که یک بخش مشکوک AI است ، می توان به عنوان یک بخش ثابت AI در نظر گرفت. حال ممکن است قادر به نوشتن برنامه ای باشیم که بتواند بر همه به جز استثنائی ترین شطرنج بازان انسانی غلبه کند . اما نمیتوانیم برنامه ای بنویسیم که بهترین حرکت بعدی را در هر موقعیت داده شده تعیین کند. اصلاح مدل اساسی بازی شطرنج با اضافه کردن تغییراتی به زیر برنامه جستجوی اصلی به دست می آید . این زیر برنامه بـــه روش تجـربی و با اجرای برنامه کامل در برابر بازیکن انسانی آزمون می شوند .

نکته اینکه پروسه مهندسی نرم افزار تولید یک برنامه از مشخصات رسمی با توجه به یک مسأله AI است ، درغیر اینصورت ما مجبور به استخدام برنامه نویسان برای انجام آن نبودیم و می توانستیم آن ر ا به طور خودکار انجام دهیم. « برنامه نویسی خودکار» خود یک شاخه از AI است . نکته دیگر اینکه برنامه نویسی خودکار اصطلاحی است که معنی آن با گذشت زمان تغییر کرده است . وقتـی برنامه نویسان به زبان ماشین برنامه می نوشتند ، کامپایلرهایی که زبان سطح بالای اولیه (مثل فرترن) را به کد ماشین ترجمه کردند ، به عنوان سیستم های « برنامه نویسی خودکار» تلقی می شدند. امروزه این اصطلاح را برای سیستم هایی به کار می بریم که برنامه های اجرایی از مشخصات غیر قابل اجرا تولید می کنند . شاخه دیگر برنامه نویسی خودکار « ترکیب و تلفیق مثالها » است که برنامه هایی را از مشخصات غیر رسمی که شامل مثال هایی از

[1] RUDE : Run, Underestand ,Debug ,Edit

ورودی و خروجی های مورد انتظار از این ورودی ها است ، تولید می کند . برنامه نویسی AI اغلب از زبانهای اخباری استفـــاده می کند زیرا این روش در مقایسه با زبانهای برنامه نویسی به روش استدلال مرسوم نزدیک تر است . طبیعت اکتشافی برنامه نویسی AI ایجاب میکند که در فرآیند کد نویسی , کدهای نوشته شده قابل فهم بودنشان را حفظ کنند و در جزئیاتی که مرتبط با دستکاری حافظه رایانه است درگیر نشوند . این کار در زبان LISP انجام شده است، این زبان مبتنی بر زبان Miranda است . در LISP به جای استفاده از نمادهای کار با آرایه ها و داده عددی علمی و مهندسی و پردازش داده مانند آنچه در C وپاسکال است از لیستها استفاده میشود ,با استفاده از لیست ها برای ذخیره داده و تأکید روی ایجاد و اصلاح نمادهایی مناسب تولید سیستمهای مورد نیاز برای کار AI آسان گردیده است . بسیاری از برنامه نویسان AI استفاده از زبان PROLOG را ترجیح میدهند زیرا هم خصوصیات پردازش لیست و دستکاری نماد LISP را دارد و هم مبتنی بر منطق است.

در زبانهای دستوری مانند C یا PASCAL دستورات منطقی و کنترلی با هم ترکیب شده و در کنار یکدیگرند در نتیجه تولید و اصلاح برنامه کمی گیج کننده خواهد بود اما در PROLOG کنترل از منطق جداست و برنامه نویس AI بر منطق و اصلاح آن متمرکز میگردد.

یک جنبه مهم جدایی منطق و کنترل که جایگاه مهمی در برنامه نویسی AI دارد ، فرق بین زبان وشبه زبان است . در زبانهای برنامه نویسی متداول بین برنامه ها و داده ها تفکیک و جدایی وجود دارد که به آن جدایی دوگانه می گویند . در زبانهای برنامه نویسی AI اغلب یک جدایی سه گانه در داده ، قوانین دستکاری داده و قوانین بیان چگونگی دستکاری قوانین داده ی وجود دارد. شبه برنامه مجموعه ای از قوانین سطح بالا است ، برنامه ای که با یک برنامه به عنوان یک داده برخورد می کند.

۱-٦ انتقاد از AI

یک انتقاد برحق از AI این است که AI روی دنیای محدود شده (micro – worlds) متمرکز است و واقعاً به هوش به عنوان مدل رفتار بشر نگاه نمی کند.

یقیناً می توانیم AI را به عنوان زمینه ی مفیدی بنگریم که توجه خود را به روش های نوشتن برنامه های رایانه ای که می توانند مسائل سخت را حل کنند معطوف نموده

است. چه این برنامه ها بتوانند به عنوان بلاکهای پیش ساخته برای رسیدن به هدف بزرگتر ساخت «موجودات بشری مصنوعی » در نظر گرفته شوند یا فقط کاربرد محدود داشته باشد . به هرحال در پی این انتقادات شاهد رشد در شاخه دیگر AI آن هم برای لحاظ کردن بعضی جنبه های استدلال مانند « برداشت متعارف » بوده ایم . به عنوان نمونه ، منطق پیشگویی استاندارد به دلیل ناتوان بودن در ارائه حقیقتی که غالباً فرض های قراردادی برایش در نظر می گیرند مورد انتقاد است. برای درک مفاهیم «برداشت متعارف » و « منطق پیشگویی استاندارد » به مثال زیر توجه کنید .

برای مثال می دانیم که اگر دو جمله $p(x) \rightarrow q(x)$ را وقتی x متغیر است و p(a) موقعی که a یک ثابت است را داشته باشیم می توانیم q(a) را نتیجه بگیریم . به عنوان نوعی مثال فرض کنید بگوییم « همه دانشجویان علوم کامپیوتر درس منطق را می گیرند. » می توانیم بیان کنیم که $CompSci(x) \rightarrow Logic(x)$. و اگر حقیقت «علی یک دانشجوی علوم کامپیوتر است» را داشته باشیم با توجه به $CompSci(x) \rightarrow Logic(x)$ می توانیم نتیجه بگیریم Logic (Ali) یعنی علی درس منطق را می گیرد .

اگر چه این یک استدلال شناخته شده است که هر کسی می تواند با وجود $P(x) \rightarrow q(x)$ p(a) را از q(a) نتیجه بگیرد ، اما اگر حقیقت « زهرا درس منطق را می گیرد» را داشته باشیم ، نمی توانیم « زهرا دانشجوی علوم کامپیوتر است » را نتیجه بگیریم . چون ممکن است دانشجویان ریاضی هم این درس را بگیرند .

مردم اغلب این نگاه غلط را در استدلال کردن دارند و همیشه نمی توانیم ذهن و رفتار آنها را از غیر منطقی بودن رها کنیم . ممکن است موردی باشد که همه دانشجویان علوم کامپیوتر درس منطق را بگیرند ، ولی یک یا دو دانشجو از دیگر گروههای آموزشی هم وجود دارند که درس منطق را می گیرند . پس این یک استنتاج «برداشت عمومی معقول » است که اگر داشته باشیم « زهرا درس منطق را می گیرد » بگوییم که زهرا دانشجوی رشته علوم کامپیوتر است مگر اینکه دلیلی داشته باشیم که خلاف آن را ثابت کند . منطق پیشگویی متداول بر خلاف روش متعارف (و غلط) انسانها , نمی تواند این استدلال را انجام دهد .هنوز این موضوع یک تحقیق در زمینه منطق است که آیا میتوان منطقی قابل تغییر بوجود آورد. دراین مورد عنصر احتمال نیز وجود دارد :

اگر در مثال بالا که یک برداشت متعارف معقول تحت شرایط داده شده است ، اگر ما دانشجویان علوم کامپیوتر فقط جز کوچکی از دانشجویانی باشند که درس منطق را می گیرند , برداشت متعارف ما دیگر معقول نیست یا دیگر این برداشت متعارف صورت نمیگرفت. فرض دنیای محدود در نظر میگیرد که داده هایی که ما ارائه کرده ایم ، همه چیزی است که شناخته شده است . در این دنیا فرض این است که اگر نتوانیم درستی چیزی را ثابت کنیم ، پس آن چیز باید غلط باشد .این فرض وقتی که ما درمورد دنیای واقعی استدلال می کنیم ممکن است صحیح نباشد .

برای مثال : اگر قوانینی که ما در مورد اینکه چه کسی درس منطق را می گیرد فقط $CompSci(x) \rightarrow Logic(x)$ و $Maths(x) \rightarrow Logic(x)$ باشد : اگر Compsci(Ali) یا Maths (Ali) را نداشته باشیم می توانیم نتیجه بگیریم Logic(Ali) غلط است . (یعنی علی درس منطق را نگرفته است.)

این استنتاج در فرض دنیای محدود ، همیشه درست است ، ولی دنیای واقعی محدود نیست و در عمل ممکن است اطلاعات و استدلال های ناشناخته ای دربارهٔ دانشجویی که درس منطق را می گیرد ، جدا از اینکه او دانشجوی کامپیوتر است یا ریاضی ، وجود داشته باشد . البته هنوز می توانیم فرض کنیم علی درس منطق را نمی گیرد ولی مثل مورد بالا این فقط فرضی است که تا زمانیکه دانش بیشتر بدست بیاوریم ، در نظر می گیریم .

۱-۷ آیندهٔ AI

در دهه های بعدی AI در بسیاری زمینه ها گسترش خواهد یافت . اما یکی از مهمترین کاربردهای AI در آینده همکاری نزدیک و متعامل با بشر است .ایده ای که در فیلم "مرد ۲۰۰ ساله " به تصویر کشیده شد . منظور فعالیتهای پیچیده ای است که یکی درمیان به وسیله بشر و ماشین انجام می شوند . چنین وضعیتی وقتی که توانایی ماشین برای انجام دادن همه فعالیتها کافی نیست به صورت رفت و برگشتی اتفاق می افتد .

ایـــن کاربرد جدید با فعالیت های اخیر در AI درتضاد است ، در گذشته فعالیت های عمده از یک نقطـــه نظر منطقـــی انجام شـــده است . سیستم هـــا نقش یک جعبه سیاه ۱ را داشتند و استدلال منطقی دغدغه اصلی آنها بوده است . در کاربردهای عملی ،

---

[1] Black Box

سیستم های مبتنی بر دانش مانند تشخیص پزشکی ، عیب یابی در حوزه های تکنیکی ، پیکربندی ، طراحی و برنامه ریزی عملی گسترش پیدا کردند. طولی نکشید ، که کافی نبودن منطق به تنهایی به عنوان یک دیدگاه درک و اثبات شد . بنابراین منطق با فرم های مختلف استدلال عدم قطعیت مثل فاکتور های عدم قطعیت ، شبکه های Bayesian ، مجموعه های فازی ، استدلال مشابهت و .... تکمیل شد

یک دیدگاه این است که این فرمهای جدید استدلالی اغلب مبتنی بر فرض های ریاضی دقیق بوده اند که به تدریج تکمیل و به فضای برنامه های هوش مصنوعی راه یافته اند . اما آنچه مهم است این است که همه این رهیافت ها مجدداً باعث موفقیت های قابل توجه هم در کاربردهای جدید و هم در تکمیل کاربردهای گذشته AI شده اند ، همچنین واضح بود که سیستم های AI کلاسیک قادر نبودند ، با پدیدهٔ « برداشت متعارف » به صورت رضایت بخش ارتباط برقرار کنند.

در حدود ۲۰ سال قبل ، همه فکر میکردند که مشکل فوق به دلیل پیچیدگی بیش از حد دانش "برداشت متعارف" است زیرا بشر به منظور تسلط بر دنیای واقعی به سال ها مطالعه نیاز دارد . با فرض اینکه انسان در هر روز ۱۰/۰۰۰ دانش کسب کند و ما بخواهیم دانش یک فرد را که در طول ۳۰ سال بدست آورده بکار بگیریم باید حدود ۱۰۰ میلیون واحد دانش را جمع آوری و دسته بندی کنیم . با داشتن این دانش میتوانیم برداشت متعارف از دنیای واقعی داشته باشیم . پروژهٔ CYC به عنوان یک گام مقدماتی و توسط لنات[1] در تگزاس برای تهیه این دانش آغاز گردید .

نتایج نشـــان داد که روش فوق با اینکه شایستگی های زیادی داشت اما در کل موفق نبود . یکی از دلایل نتایج منفی ، بوجود آمدن مجموعه های بزرگ از واحد های دانش بود که کنترل کردن و تعامل آنها با هم خیلی سخت بود. درسال های گذشته ایده دیگری بسط داده شد که در اصل جدید نبود اما توجه سیستماتیک به آن نشده بود.این ایده چیزی جز روش تقسیم کردن کار بین انسانهایی که ایده های خلاق و برداشت عمومی دارند با رایانه ها نبود . از یک طرف این افراد می توانند ایده های پایه تولید کنند و از طرف دیگر ماشین هایی که جستجوی بدون تعقل اما سریع و به صرفه انجام می دهند ، توانایی مدیریت پایگاه داده های بزرگ را دارند و می توانند محاسبات وسیع

[1] D. Lenat

انجام دهند. این کار به ایدهٔ تولید یک سیستم دستیار منجر شده بود. در سیستم های دستیار گذشته ، مواقعی که ماشین عاجز از تولید نتایج دلخواه بود از بشر برای کمک به حل مسئله استفاده میشد.

یک رهیافت سیستماتیک نیازمند رسیدگی به انواع مسائل زیر است :

۱ـ توصیف اینکه چه کارهایی را بشر بهتر از ماشین می تواند انجام دهـــد و بر عکس .

۲ـ تولید یک برنامه کاری که در مواقعی که کاری سخت بین انسانها و عامل های نرم افزاری تقسیم می شود بتواند این تعامل را مدیریت کند.

۳ـ توصیف جنبه های تکنیکی در ارتباط بین عامل های مختلف.

٤ـ یکپارچگی استدلال برداشت عمومی و استفاده از مفاهیم غیر رسمی با تفکر منطقی و محاسبات رایانه باید بوجود آید . در این زمینه تحقیقات مفصلی صورت میگیرد که به " سیستم های نیمه رسمی " معروف هستند.

به طور مختصراً دراینجا دو عالم با هم روبرو می شوند ، فرمالیسم ها در برابر مفاهیم غیر رسمی ، حقیقت منطقی در برابر استدلال تخمینی ، از کل به جزء رسیدن (استنتاج) در برابر استقرا و یادگیری . در تضاد با ایده های پروژه CYC ، هر شخصی سعی می کند بر پیچیدگی دنیا با شناخت نا درستی ، تسلط پیدا کند . این راهی است که انسانها آن را پیموده اند و سیستم های دستیار باید این توانایی ها را بدست آورند. واضح است درهمکاری بین انسان و ماشین انسانها شریک های برتری هستند. انسانها تصمیم گیری های نهایی را انجام می دهند و نسبت به عواقب کار مسئولیت دارند . یک روش جهت توصیف نقش تابعی ماشین ها ، در نظر گرفتن آنها به عنوان خدمتکارانی اســـت که انسانهـــا را با تولید دانش لازم به شکل مفید و به لحظه ، پشتیبانی می کنند. مشکل دیگری که فوراً ایجاد می شود، کنترل بی دقتی است چون خطا های مطلق قابل پذیرش نیستند و این به معنی داشتن محدوده های امن در مدیریت خطای عددی در استدلال سمبلیک است.

زمینه های مختلفی با این موضوع وجود دارند که برجسته ترین آنها استدلال بر مبنای مشابهت و استدلال فازی است . برای توسعه آینده AI در این زمینه گسترش حوزهٔ تکنیک های AI سنتی مهمترین گام نیست بلکه مهمترین گام کشف روشی است که

بشر استدلال میکند. ایـــن یک رهیافت مرحله ای است که انسانها را به شکل سیستم های دستیار متحد می کند و به ماشین اجازه می دهد با انواع روش های فکری انسان با مفاهیم غیر رسمی ارتباط داشته باشد. مطمئناً این مزیت بزرگی برای کاربرد های بسیاری مثل تجارت الکترونیکی، پزشکی ، تحصیلات و... خواهد بود .

تحقیق و پیشرفت آینده در AI می تواند در سه عبارت « تولید دانش» ، « خود مختاری » و « موقعیت شناسی »گنجانده شود.

۱-۷-۱ ایجاد اطلاعات

اگر از داده های انبوه ذخیره شده در یک پایگاه داده و از روشهای معمولی بازیابی داده ها استفاده کنیم تنها میتوانیم به پرسشهای از پیش تعیین شده پاسخ دهیم حال آنکه دانش زیادی در این پایگاهها نهفته است که توسط طراحان دسته بندی نگردیده اند . امروزه نیازمند اعمال سوالات واقعی از این پایگاهها هستیم.امروزه روشهای زیادی مانند۱(data warehouse) ، متدهای کلاسیک آماری ، شبکه های عصبی و الگوریتم های یادگیری ماشین برای این کار توسعه یافته اند.

فناوری استخراج دانش از اطلاعات غیر دسته بندی شده داده کاوی نامیده میشود . AI کمک اساسی به توسعه این فناوری کرده است . در سالهای اخیر از این فناوری به صورت تجاری استفاده زیادی می شود .مثلا از اطلاعات جستجو شده توسط مردم در سایتهایی مانند گوگل علاقه مندی آنان به موضوعات مختلف استخراج میشود که خود دانشی گرانبها برای مهندسی اجتماعی جامعه یا توسعه مد یا کشف خلاهای سیاسی است . داده کاوی در حوزه های محدودتر نامهای دیگری دارد مثلا برای داده ی فرمت نشده متن کاوی ، برای اطلاعات سایتهای وب ، وب کاوی یا برای اطلاعات نهفته در تصاویر تصویرکاوی گفته می شود.

اطلاعات تولید شده به این روش از یک فضای خالی ، ایجاد نشده است ، بلکه به طور ضمنی از یک سری داده گردآوری شده است ، به بیان دیگر ، این اطلاعات یک خلاصه داده است . اغلب به یادگیری ماشین به عنوان یک نمونه برای یادگیری انسان توجه می شود . این دیدگاه قطعاً غلط نیست ولی خیلی محدود است . از علوم شناخته شده در می یابیم که در یادگیری انسان ، آمار و ارقام هم نقش مهمی دارند ، پس می

---

[1] Data Warehouse

توانیم فرض کنیم که یادگیری ، یک پردازش پیچیده در جایگاهی است ، که الگوهای شناختی مختلف با هم در تعاملند . این مورد به وسیله ترکیب روشهای مختلف در داده کاوی به خوبی نمایان است. البته برای حل این مسئله تحقیقات زیادی در آینده صورت خواهد گرفت . راه حلی که فقط کارایی فناوری داده کاوی موجود را بهبود نمی بخشد ، بلکه یک جنبهٔ شناختی دارد . می توانیم فرضیه ای که بشر، دانش را از تجربیات بدست می آورد را دنبال کنیم ، بشر با ترکیب مجموعه های بزرگی از مثالهایی که در طول زندگیش می بینند دانش کسب کرده و به اصطلاح خبره می شود , تقلید این روش در فناوری داده کاوی هر روز در حال پیشرفت است . پردازش انتزاع در قسمت شناسایی دقیق و شناسایی غیر دقیق بر مبنای ‹ دانش پایه› و ‹ دانش تخصصی› صورت میگیرد. این پردازش ممکن است با تکنیک های داده کاوی پیشرفته شبیه سازی شود و با این روش ممکن است دانش (خبرگی) از داده به طور اتوماتیک تولید شود.

۱-۷-۲ خود مختاری

از سالها پیش سیستم های توزیع شده به عنوان ابزار قدرتمند برای افزایش کارایی حل مسئله در حوزه کامپیوتر گسترش یافته اند و هر روز دامنه های کاربردی جدیدی برای تکنولوژی رایانه بوجود می آورند .سیستمهای توزیع شده امروزی برای تضمین یکپارچگی نیازمند ارتباطات زیادی هستند که این ارتباط بدلایلی مانند دوری مسافت و غیره مبتنی بر شبکه هاست و بسته به موقعیت یک شبکه امکان قطع این ارتباط وجود دارد .حال اگر این ارتباط برای مدت طولانی قطع شود چه اتفاقی خواهد افتاد. فناوی AI علاوه بر تلاش جهت گسترش الگوریتم های جدید توزیع بهینه داده و ترکنش ها برای این سیستم ها سعی میکند اجزای این سیستمها را به درجاتی از خود مختاری مجهز کند .

سیستم های رایانهی بیشتر به عنوان ماشین های محاسبه و ذخیره ساز داده های انبوه و جهت پشتیبانی تصمیم گیری به کار گرفته شده اند . بشر برای پذیرش تصمیمات گرفته شده توسط رایانه همواره در تردید به سر برده است . به هر حال جندی است رایانه ها تأسیسات بزرگی را در بعضی حوزه های فنی مانند سد و صنایع کنترل می کنند که در بعضی مواقع رسیدگی به امور نیازمند تصمیم گیری است اما هنوز انسان به عنوان ناظر و سرپرست عمل می کند. به عنوان نمونه میتوان از سیستمهای

کنترل دریچه های سد یا کنترل دریچه های مسیرهای فشار گاز نام برد . پس میتوانیم فرض کنیم سیستم های رایانه ای قبلاً خود مختار شدن را آغاز کرده اند . ولی این تنها شروع یک فرآیند بزرگ است . در سیستم های چند عاملی که با کمک هوش مصنوعی در حال گسترش هستند ، خودمختاری یک موضوع کلیدی است. یک جزء از سیستم می تواند به عنوان یک عامل که فقط درجه ای از خود مختاری را دارد ، دیده یا طرح ریزی شود یا به عنوان یک جزء منفعل در نظر گرفته شود . خودمختاری با چند ویژگی مشخص می شود . ابتدا ، خودمختاری به معنی توانایی انتخاب چند عمل از مجموعه ای از اعمال ممکن در موقعیت معین ، شامل تصمیم بین غیرفعال بودن یا موقع نیاز فعال شدن ، می باشد . این توانایی Proactivity نامیده می شود . ویژگی مهم دیگر توانایی دنبال کردن اهداف است . نه تنها لازم است که یک عامل ، اهدافی داشته باشد که برنامه ریزی و فعالیتش را راهنمایی کند بلکه اینکه عامل بتواند این اهداف را تغییر دهد ، رها کند یا اهداف جدیدی اتخاذ کند نیز بسیار ضروری است . ویژگی سوم توانایی ارتباط فعال و همکاری است . کلمه" active " (فعال) در اینجا مهم است چون تبادل پیام به منظور فراخوانی های تابعی , بین اجزا یک سیستم معمولا ارتباط نامیده می شود ، اما اینجا این معنی را نمی دهد . ارتباط و همکاری بین عامل ها یک رفتار proactive است ، یک عامل می تواند ارتباط را شروع کند یا برای همکاری با بقیه هر موقع که احساس کند لازم است به ارتباط دیگران گوش کند. ویژگی خود مختاری یک رابطه جدید بین عامل ها ایجاد می کند چه آنها به عنوان ربات و یا به عنوان سیستم های نرم افزاری و انسانهایی که خصوصیت مشارکت دارند ، در نظر گرفته شوند . عامل های خود مختار با ماشین هایی که در هنگام نیاز شروع به کار میکنند ، کارشان را انجام می دهند و خاموش یا غیر فعال می شوند ،کاملا متفاوتند . ممکن است شخصی به این عوامل به دید خدمتکاران نگاه کند ، ولی خدمتکاران خواسته ها و توانایی هایی ادراکی پیچیده خودشان را دارند ، و این موارد باید به عامل ها واگذار شوند . ما باید انتظار زندگی در یک جامعه پیچیده تر را در آینده داشته باشیم زیرا در آینده ما با عامل ها به عنوان نوعی از موجودیت های زنده همکاری می کنیم ، این عوامل در زندگی روزمره ما مثل دستیار های شخصی در سیستم های رایانه ای مان ، به عنوان راننده در ماشینمان ، یا مثل سرایدار خانه مان حاضر خواهند شد . محققان AI

باید با جامعه شناسان در باره مسائلی که ممکن است از این دید ایجاد شوند ، در ارتباط باشند .

۱-۷-۳ موقعیت شناسی

در یک دید کلی ، همه سیستم های رایانه ای در شرایط و موقعیتهایی قرار گرفته اند ، اما سیستم های سنتی موجود در شرایط خوش- تعریف و خیلی محدود شده که کاملاً به وسیله بشر شناخته شده اند ، قرار دارند . موقعیت شناسی یک موضوع مهم تحقیقاتی درباره ارتباط متقابل عوامل در سیستمهای چند عاملی است و در کارکرد صحیح رباتهای متحرک بسیار مهم است . بدیهی است که موقعیت شناسی یکی از پیامدهای خود مختاری است . فقط سیستم های خود مختار می توانند خودشان را در جهت و جایگاه مناسب قرار دهند و در هر موقعیت عمل مناسب صورت دهند. در اینجا موقعیت به معنی مجموعه تاثیرهایی از محیط است که تقریبا غیر قابل پیش بینی اند اما اهمیت زیادی برای سیستم دارند به طوری که سیستم مجبور به نشان دادن واکنش با روش مناسب است . یک سیستم جایگزین مجبور است دو مسئله اصلی را حل کند ، برای مثال چگونه موقعیت را حس کند و چگونه واکنش مناسب را انتخاب کند . دریافت یک موقعیت ممکن است یک کار ساده برای یک عامل نرم افزاری در یک محیط خوش ساخت باشد . به هر حال اگر به فعالیت عوامل مثلاً در اینترنت فکر کنیم ، کارها خیلی پیچیده تر می شوند چون این محیط به شدت غیر ساخت یافته و دینامیک است . البته حس کردن با رباتها از آن هم پیچیده تر است . در اینجا ابتدا سیگنالهای فیزیکی اولیه باید به داده تبدیل شود ، وظیفه ای که ممکن است به فیزیک دانان و مهندسان الکترونیک محول گردد . در مرحله بعد ، سیگنال ها از منابع مختلف ، باید با هم ترکیب شوند تا توصیفی از وضعیت موحود ایجاد کنند که سیستم را قادر به واکنش مناسب کند . این وظیفه امتزاج حسگر یا sensor fusion نامیده می شود و اینجا جایی است که روشهای AI وارد عمل میشوند . چنانکه میدانیم فهم یک موقعیت فرایند بسیار پیچیده ای است که نیاز به اطلاع پایه ای و فعالیتهای عصبی زیادی دارند . مغز موقعیت را از ترکیب ورودی ها متفاوت درک می کند . هنوز درباره این فرایند چیز کمی می دانیم و برای استدلالهای واضح ، دوباره سازی آن برای ماشین سخت است . اما برای بدست آوردن بینشی در این موضوع میتوانیم از شبیه سازی های صورت گرفته

با استفاده از روش های AI استفاده کنیم . در صورتیکه امکان شبیه سازی وجود داشته باشد ، می توان از سیستم های جایگزین مصنوعی به منظور توسعه توصیف مناسب برای درک موقعیتها استفاده کرد.

فصل سوم

فراگیری و نمایش دانش

اهداف

در پایان فصل، دانشجو با مفاهیم زیر آشنا می‌شود:

هوش ماشین

مهندسی دانش

رویه برای فراگیری دانش

نمایش حقایق

طرح های نمایش منطقی

طرح های نمایش رویه ای

طرح های نمایش شبکه

طرح های نمایش ساخت یاقته

۳-۱ مقدمه

حتی بحث کردن راجع به کار ماشینهای نسبتاً ساده مانند ماشین های لباسشوئی یا چرخ های خیاطی سخت به نظر می آید ، مگر اینکه کارکرد هایی را که این ماشین ها ، بر اساس آنها و در جهت انجام وظایفشان طراحی شده اند ، قابل درک باشد . از آنجائی که هوش مصنوعی با پیچیده ترین نوع ماشینهای قابل تصورمان ، ارتباط دارد ، 'ماشین های هوشمند ' ، شاید باید در تعریف کردن انتظارمان از وظایف چنین ماشین هایی ،

تلاش ویژه ای صورت دهیم . بدیهی است که انتظار داریم این ماشین ها هوشمند باشند . اما منظورمان از هوشمندی چیست ؟

با مراجعه به فرهنگ لغت هوشمندی عبارت است از توانائی درک کردن و درک کردن یعنی دریافت و درک چیزی یا تشخیص معنای آن . این تعاریف مفهومی است که به اندازه کافی واضح و روشن به نظر می آید . وقتی این مفاهیم را به طور ذهنی به کار می بریم ، به نظر می رسد که آنها با تجربه شخصی مان از چیزی که مایل است هوشمند باشد یا از هوش مان استفاده کند رابطه ای بدون خطا دارند . متاسفانه ، به کارگیری این مفاهیم هنگامی که به طور معقول و به عنوان یک توانائی ذهنی تعریف میشوند ، قابل استفاده نیستند ، زیرا ممکن است این توانائی ذهنی با دیگر موجودیتها خواه زنده یا مکانیکی ، مشترک باشد .

مسئله اصلی این است که ، می دانیم برای درک چیزی چه احساس می کنیم و به طور عام مایل هستیم به دیگر افراد یا اشیاء با احساس مشابهٔ احساس خودمان ، اعتبار دهیم . یک مثال ساده مانند یک قطعه آشنا از ماشین ، مثل ترموستات درسیستم حرارت مرکزی را در نظر بگیرید . ترموستات فقط بالاتر یا پائین تر بودن دما از یک حد معین را تشخیص نمی دهد ، بلکه با یک عمل مناسب به این وضعیت پاسخ می دهد . به نظر می آید که در یک رابطه کاملاً محدود شده ، ترموستات دارای درک و فهم می باشد و این موضوع را با واضح ترین روش ممکن ، از طریق رفتاری هوشمندانه اثبات می کند . اگر ترموستات هوشمند باشد ، ارزش لغت هوشمندی را می کاهیم . با دقت در موارد بکارگیری عملی عبارت هوش مثلا سئوالات مورد استفاده در آزمون هوش که به عنوان معیاری برای سنجش میزان شایستگی افراد جهت شغلهای کاملاً تعاملی مورد استفاده قرار می گیرد به این نتیجه می رسیم که هوش تصور کلی از مقداری مبهم است. بنابراین آیا لازم است که هوشمندی را به توانائی های ذهنی جداگانه ای ازقبیل ادراک ، دلیل ، خلاقیت تجزیه کنیم ؟ اگر چنین است ، تفاوت میان هوشمندی و دانش چیست؟

یکی از معدود نتایج غیر قابل تغییر و ثابت که از سه دهه اول تحقیقات AI شناخته

شده است ، این است که " هوشمندی به دانش نیاز دارد ." ضرورتاً دانش دارای چندین ویژگی مطلوب کمی است که شامل موارد ذیل می باشد :

- حجیم است .
- برای توصیف و مشخص کردن به صورت درست و دقیق سخت و دشوار است .
- پیوسته در حال تغییر است.
- ازمرحله داده تا مرحله سازمان دهی شده متفاوت می باشد.

راههای مختلفی برای طبقه بندی انواع دانش وجود دارد ، یکی از مهمترین معیارها تفاوت میان دانش القاء شده و دانش استنتاج شده ، است . این بهترین تعبیر به وسیله یک مثال است .

یک مهارت معمولی که بیشتر بچه ها در سنین بین پنج تا ده سالگی بر آن تسلط دارند ، در اختیار گرفتن توپ است . ممکن است برای کودکی در سن پنج سالگی گرفتن یک توپ بزرگ و سبک و پرتاب کردن آن به سمت بالا حتی برای کسری از متر مشکل باشد ، اما چندین سال بعد شاید قادر به در اختیار گرفتن یک توپ تنیس وپرتاب آن به هوا تا صد متر یا دورتر هم باشد . بشر در سالهای اولیه به طور موثر قادر به کسب مهارت تکنیکی و فنی جهت محاسبه مسیر گلوله های پرتابه ای نبود . فهم یک کودک به وسیله قیاس بدست می آید . این فهم در نتیجه مشاهده زیاد مسیرهای توپ های پرتاب شده و تلاش برای در اختیار گرفتن آنها حاصل می شود بنابر این بچه ها قادر به پیش بینی مسیر توپ بعدی که می خواهند دراختیار بگیرند هستند . از طرف دیگر ، یک سیستم کامپیوتری با تکیه بر اطلاعات شامل سرعت سیر پرتابه و مسیر آن ، مکان بعدی پرتابه را با استفاده از قوانین نیوتن محاسبه می کند . این موضوع به مجموعه فرمول های واضح ، دقیق ، وابسته به قوانین ریاضی و برنامه ریزی شده در کامپیوتر، بستگی دارد . این برنامه ، کامپیوتر را قادر می سازد تا در مورد مسیر پرواز یک پرتابه با مراجعه به مجموعه قوانین رسمی ریاضی ، نتیجه گیری کند .

تعداد کمی از مردم ممکن است در مورد محاسبه قضیه مسیر یک پرتابهٔ مبتنی بر قوانین

ریاضی که به هوشمندی بیشتری نسبت به توانایی در اختیار گرفتن یک توپ نیاز دارد ، با هم بحث کنند . پس یک تمایز مهم میان دانش و هوشمندی وجود دارد.
واضح است که امکان دارد یک ماشین لزوماً بدون داشتن هوش ، دانش را ذخیره کند ، ولی وجود یک ماشین هوشمند که دانش نداشته باشد غیر ممکن به نظر می آید .
سئوال راجع به چگونگی ، میزان وسعت و اندازه ذخیره سازی دانش در یک ماشین ، در همه زمینه های هوش مصنوعی یک سئوال بنیادی است .

۳-۲ هوش ماشین

صاحبنظران در هوش مصنوعی معتقدند که یک ماشین هوشمند ، ماشینی است که قادر به انجام کارهایی است که به وسیله بشر انجام شده است و برای انجام آن کارها نیاز به هوش داریم . بنابر این ارایه تعریف دیگری از هوش غیر ضروری است . این مطلب خیلی کاربردی به نظر می رسد . به زبان بهتر اگر ما نمی توانیم هوش را تعریف کنیم ، اما قادر به تشخیص اینکه کدام افراد و چه موقع هوشمند به نظر میرسند هستیم .
متاسفانه ، این تعریف هم به سادگی شکست می خورد . در پاسخ به سوال مقایسه منطقی انسانیکه برای انجام یک وظیفه نیاز به هوش دارد و هوشمند بودن یک ماشین که رفتار هوشمندانه را نسخه برداری و تکرار می کند ، افراد بدبین و شکاک پاسخ می دهند اثبات شده است که کاری را که یک ماشین انجام میدهد یک کار مکانیکی و غیر فکری است زیرا اگر قادر به توضیح نحوه کار یک ماشین باشیم ، همانا هوشمندی آن را توضیح داده ایم . اگر هم بپذیریم که یک ماشین هوشمندانه رفتار می کند ، آیا ممکن است این رفتار هوشمند همان هوشی باشد که به عنوان یک خصوصیت از افرادی می شناسیم که کامپیوتر و نه خود ماشین را برنامه ریزی کرده اند ؟ همچنین آیا می توان تمایزی میان رفتاری هوشمند و رفتاری که فقط هوشمند به نظر می رسد قایل شد ؟
به طور طبیعی ، زمانی که نمی دانیم چه در مغز ما رخ می دهد ، پذیرش این بحث ، دلالت ضمنی بر این مطلب دارد که هوش خود ما ممکن است به سبک و روشی غیر واقعی باشد . شاید فقط فکر می کنیم که هوشمند هستیم . در مورد یک ماشین می

توانیم خصوصیت و طبیعت فرآیندهای داخلی را ایجاد کنیم . اگر ماشین و انسان رفتار یکسانی نشان دهند ، فرآیندهای داخلی تا چه حد قابل مقیاسه هستند ؟

اگر دانش انباشته شده بیشتر از توان نگهداری و پردازش نباشد ، حتی هوش یک کامپیوتر نسبتاً متوسط بیشتر از هوشمند ترین افراد ارزیابی می شود. به طور مسلم ، ظرفیت محض حافظه کامپیوتر از حافظه خود ما پیشی گرفته است . دقیقاً به همین علت است که کامپیوترها مفید به نظر میرسند زیرا مثلا در یک میلیونیم ثانیه می توانند به فروشنده بلیط اطلاع دهند ، که آیا جائی در پرواز فردا به شیراز باقی مانده است یا خیر . البته این عمل به کمک دسترسی به حافظه بسیار بزرگ و اپراتوری که سئوالات کاملاً درست ، و دقیق راً با روش و ترتیب مناسب می پرسد امکان پذیر است.در واقع نرم افزار هوشمند بخشی از دلیل اجرای موفق یک سیستم است .

موضوعات کلیدی که یک طراح سیستم AI با آنها روبرو می شود ، عبارتند از:

- فراگیری دانش
- ارائه دانش
- دستکاری دانش

دستکاری دانش اصولاً به واسطه استنتاج و استنباط و در اکثر مواقع استراتژی کنترل جستجوگرا یا موتور استنباط (که موارد دانش بدست آمده ، نتایج ایجاد شده ، و ترتیب گامهای بکار برده شده را تعیین می کند .) رخ می دهد .

فراگیری دانش چیست ؟ بوچانان۱ فراگیری دانش را چنین تعریف میکند :تغییر شکل راه حل مساله ناشی که از وجود چند منبع دانش که راه حل بالقوه در آنها وجود دارد و به یک برنامه AI منتقل شده است .

برای ایجاد توانمندی جهت حل مسائل پیچیده فرآیند فراگیری دانش باید به شناختن و استخراج دانش کافی در حوزه مربوطه بپردازد . غالباً این فرآیند به انتقال و استخراج دانش از منابع متنوع و نمایش آن در یک غالب مناسب اشاره میکند .

---

1 Buchanan 1983

۳-۳ مهندسی دانش

حوزه مهندسی دانش را می توان روش ارزیابی مسایل ، آموختن دانش وایجاد یک سیستم دانش محور تعریف کرد . (هارمون۱ و کینگ۲ ۱۹۸۵)

سیستمهای کامپیوتری مرسوم به طور سنتی از طریق تحلیل گر سیستم ، برنامه ریزی و طراحی شده اند . تحلیل گر بین کاربری که دانش کمی در مورد جزئیات تکنیکی سخت افزار و نرم افزار دارد و یک برنامه نویس که ایده کوچکی در حل تقاضاهای کاربر دارد ارتباط برقرار می کند . سیستم های مبتنی بر دانش (شکل ۳,۱) نیازمند رهیافت مشابهی هستند هر چند یک تفاوت عمده وجود دارد . تحلیل گران سیستم های کامپیوتری که روالهای خاصی را انجام می دهند را طراحی می کنند اما توسعه و پیشرفت یک سیستم مبتنی بر دانش ، با الحاق دانش خبره در یک سیستم ،حاصل میگردد . به همین دلیل ، مهندس دانش توسعه سیستم را به تقلید از روش سنتی عهده دار شده است (آدلی۳ ۱۹۸۸) . مهندس دانش با تحلیل گر سیستم قابل مقایسه است اما با فرآیندهای خبره ارتباط بیشتری دارد .

---

1 Harmon
2 King
3 Adeli

شکل ۳-۱ سیستم بر مبنای دانش

هارت۱ (۱۹۸٦) پیشنهاد داد که مهندس دانش ، هرگز نباید متخصص در تهیه دانش باشد . این موضوع برمبنای فرضی بیان شده است که مهندس دانش معمولاً دارای اطلاعات ناکافی درباره برنامه نویسی و تکنیکهای توسعه سیستم مبتنی بر دانش است زیرا شرح و توصیف کامل و دقیق این اطلاعات برای آنها دشوار خواهد بود .

یک مهندس دانش باید با حوزه دانش تحت آزمایش ، چیزی که اغلب به آسانی قابل دست یافتن نیست ، آشنا باشد . خصوصاً این وضعیت برای حوزه های دانش مهندسی صادف است . آدلی (۱۹۸۸) معتقد است ، در آینده ، سیستم های مبتنی بر دانش مهندسی ، به وسیله مهندسین ماهر و مطلع در حوزه های کاریشان توسعه خواهد یافت، خصوصاً چنانچه مهندسین بیشتری معرفی شده باشند و در باره ابزارها و فنون AI مطالبی را آموخته باشند .

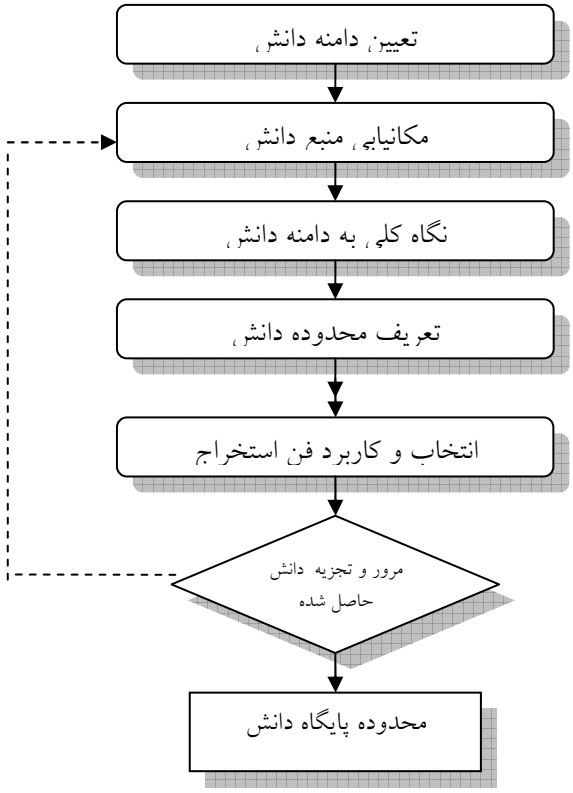
۳-٤ روش فراگیری دانش

در ابتدا ، برای تضمین کردن توسعه یک سیستم محکم و متمرکز ، دامنه دانش(شکل ۳,۲) باید کاملاً شناخته شده باشد . این دستاورد از طریق یک آزمایش دقیق بر روی اهداف سیستم پیشنهاد شده ، به دست می آید .

محل منابع دانش در دامنه مربوط برای تحلیل و استخراج بعدی باید تعیین گردد . پیش از تعریف مرزهای دامنه ، نگاه کلی به موضوع اصلی توصیه می گردد . این نگاه کلی به منظور ایجاد بصیرت بوده و ساختاری برای کارهای آینده فراهم می کند ضمنا مشکلات بالقوه را نیز مشخص و پر رنگ می کند .

---

1 Hart



شکل ۳-۲ روش فراگیری دانش

تعریف مرزهای دامنه تضمین می کند هیچ گونه حذف عمده ناخواسته یا تلاش هدر رفته در ارتباط با گردآوری اطلاعات در طی فرآیند استخراج دانش صورت نمی گیرد. انتخاب تکنیک استخراج مناسب دانش از روشی که برای تجزیه کردن اطلاعات بکار برده شده است ، پیروی می کند برای این کار تکنیک های متنوع و گوناگونی در دسترس هستند .زمانی که دانش شناخته و فراگرفته شده باشد ، آخرین وظیفه در پردازش این است که تضمین کند ، دانش درست و بی عیب ، در صورت تکرار قابل اعتماد و متمرکز شده در یک قالب مناسب است . مجموعه دانشی که در دامنه تحت رسیدگی و بررسی شرکت می کند ، باید قابل فهم و واضح باشد .

استنباط ، موضوعات منفرد دانش را که لازم است در تمام موارد یکسان و سازمان دهی شده باشد را استخراج مینماید . اما اینکه چه میزان سازمان دهی لازم است ، به روشی

که بانک اطلاعات اجرا و استفاده شده است بستگی زیادی دارد . برخی پیاده سازی ها به موضوعات و بخش هایی که در گروه ها و دنباله ها مرتب شده اند نیاز دارند ، که این نظم و ترتیب ، ارتباطات میان بخش های موجود در پایگاه دانش را نشان می دهد . این نظم و ترتیب کاربرد فراوانی دارد .به عنوان یک مثال جامع و محکم ، خیلی از سیستم ها به اطلاعاتی در باره مشتری و بیمار از قبیل : نام ، سن ، جنسیت و سایر موارد ، نیاز دارد . سیستم مجبور به انتخاب ترتیبی برای بدست آوردن این اطلاعات است . یک راه ساده دنبال کردن ترتیب متنی این بخش ها در پایگاه دانش است .

۳-٤-۱ منابع دانش

سه منبع اصلی دانش عبارتند از :

- مطالب نوشتاری
- افراد خبره
- مثال ها.

سه رکن مختلف دانش عبارتند از :

- قوانین علمی
- تجربه
- الگوها

می دانیم که دانش ، اطلاعاتی است که به ما در حل مسائل موجود در یک دامنه خاص یاری می دهد . سودمندترین دانش ، بیانی ازبعضی قواعد است که به ما برای پیش بینی چیزی که بعداً اتفاق خواهد افتاد یا توضیح و تشرح نحوه و دلیل برخی مواردی که اتفاق افتاده است ، یاری می رساند . قویترین قوانین با قاعده که می شناسیم ، قوانین علمی می باشند . بنابراین ، فراگیری دانش در زمینه علمی از زمینه های دیگر ساده تر است ، و اینکه آیا این قوانین ازکتاب به دست آمده اند یا از مهارت ، چندان اهمیت ندارد .

در حوزه هایی که کمتر با اصول علمی تطبیق دارند ، غالباً قاعده ها ضعیفتر هستند و به طور واضح بیان نشده اند . از قوانین علمی موجود در باغبانی یا توصیه سرمایه گذاری

کسی چیزی نشنیده است ، با این حال فرد خبره در این زمینه ها از قواعد و موضوعات مهم مرتبط با آن آگاه بوده و روزانه از آنها استفاده می کند . در اینجا دانستن منابع و مراجع کسب دانش ضروری تر می باشد.

در زمینه های کاربردی ، امکان ندارد یک تجربه خبره تدوین و نوشته و ثبت شده باشد. ما حقیقتاً به یک فرد خبره زنده نیاز داریم ، زیرا میزان کمی از دانش به شکل مکتوب و نوشته شده در دسترس قرار دارد. و البته تاکنون بیشتر کاربردهای سیستم های خبره در این حوزه ها بوده اند . پایگاه دانش سیستم های خبره کاربردی امروزه اغلب فاقد فرمول علمی بوده و بیشتر متکی به تجربه افراد خبره است. بزرگترین مشکل در این زمینه ها این است که فرد خبره توانائی مکتوب کردن دانش خود را ندارد بنابر این غالباً این موضوعات جمع آوری دانش را ، هم برای فرد خبره و هم برای مهندس دانش کاری سخت و دیربازده می سازد .

دانش ، حاوی حقایق ، روالها و قواعد حکمی است و به طور وسیع منتشر شده است . بسیاری موارد مختلف دیگر است . افراد خبره ، مردم با تجربه می توانند انواع مختلفی از اطلاعات را فراهم آورد . ویس۱ و کولیکوسکی۲ (۱۹۸۳) معتقدند که منابع تولید اطلاعاتی که به تولید دانش برای حل مسئله منجر میشود عبارتند از :

- تجارب شخصی از مسائل حل شده گذشته .
- نظریه یا روشهای شخصی برای حل مسائل .
- دانش شخصی درباره استدلالها برای انتخاب روشهای بکاررفته .

بهر حال ، برای پایگاه دانش سیستم های کاربردی حوزه مهندسی ، این رهیافت ، ناکافی است . مسائل مهندسی معدودی ، کاملاً اکتشافی هستند لذا منابع دانش اشاره شده در زیر باید علاوه بر دانش فرد خبره مورد بررسی و تحقیق قرار گیرد :

- نظریه خبره
- داده های سوابق قبلی
- رفتارهای عرفی

---

1 Weiss
2 Kulikowski

- روال های استاندارد مهندسی
- داده های آزمایشی
- مکتوبات فنی شامل :
  - کتابهای تخصصی
  - مجلات
  - راهنما های استفاده
  - اطلاعات تولید کننده
- معادلات مهندسی قطعی شده

بنابر این استفاده از فرمول یا داده صحیح و اصلاح برنامه های کاربردی به استناد آن ، در مسیر حل مسئله ممکن است باعث تولید دانش شود

۳-٤-۲ انواع دانش

سئوال بعدی که درگیر آن هستیم این است که ، برای فراگیری دانش چه چیزی باید گردآوری شود ؟ ما در پی سه دسته بندی دانش هستیم .

اولین دسته بندی ، ساده ترین آن است که آنرا « دانش ادراکی » می نامیم. این گروه دانش شامل حقایق ساده و روابطی مانند رنگ بلور های کوارتز ، نقطه ذوب آهن یا ماکزیمم گشتاور یک موتور است . اگر بخواهیم دقیقتر صحبت کنیم ، سیستم های خبره نیازی به نگهداری این اطلاعات ندارند زیرا می توان با طرح سوالاتی پاسخ های مورد نظر را به دست آورد . اما در عمل هر سیستم خبره ای به نگهداری و دسته بندی کردن چنین اطلاعاتی نیاز دارد .

سطح بعدی چیزی است که اغلب افراد در دانش ملاحظه می کنند یعنی مفاهیم و روابط . در اینجا قوانین علمی از قبیل معادلات نیوتن و قانون بویل و نیز مشاهدات اکتشافی نظیر " تب نشان دهنده بیماری است . " قرار دارند . این سطح ، اصول سیستم های خبره است .

سومین سطح که قطعا مهم ترین سطح هم می باشد دانش روش دسته بندی مسائل و نحوه چگونگی شروع حل یک مسئله و چگونگی تفکیک و تجزیه مسایل است که

دانش استراتژیک نامیده می شود .
سودمندی یک سیستم خبره کاملاً به دانش استراتژیک وابسته است . این سه سطح نه تنها قدرت این بخشها را نشان می دهند بلکه نحوه سخت و دشوار نگهداری اطلاعات آنرا نیز نمایان می سازد .

۳-۵ نمایش دانش

مسئله نمایش دانش به عدم مطابقت بین حافظه بشر و حافظه کامپیوتر مربوط می شود، مثلاً چگونه باید دانش را کدگذاری و نمایش دهیم که بازتابی مناسب از دانش فرد خبره بوده و بتواند توسط کامپیوتر هم دستکاری شود . تحقیقات روانشناسی اشاره بر این مطلب دارد که ما انواع رفتار های استدلالی را نمایش نمی دهیم ، بلکه آنرا با سیستم های استقرائی یا سیستم های اثبات قضیه ارتباط می دهیم . (هارت ۱۹۸۵) انسانها با استفاده از تجربیات منطقی وقوع هر اتفاقی را دلیل وجود شرایط و حالتی میدانند. برنامه نویسی کامپیوتر با مسائل غیر ملموسی نظیر ایده ها ، مفاهیم و روابطشان ، قابلیت رسمی ارائه و دستکاری موجودیتهای نسبتاً انتزاعی که باید تعبیه و برنامه ریزی شوند ، سرو کار دارد . این کار مستلزم استفاده از ابزارهای ساختیافته و پردازشی است که توان فایق آمدن به انواع ورودیها نظیر داده ، متن ، و دستکاری عددی و سیستم های محاسبه گرا و زبانهایی که هم اکنون متداول شده اند را داشته باشد . این ساختار های بسیار قوی را « پایگاه دانش » و عملگر های دستکاری بر روی پایگاه دانش را « موتور استنباط » می نامیم .

۳-۵-۱ چه چیزی ارائه می شود ؟

در ابتدا اجازه دهید انواع دانش مورد نیاز برای ارایه سیستم های AI معرفی کنیم :
اشیاء [1]: موجودیتهای دنیا که درباره آنها اطلاعاتی داریم به عنوان مثال ، «گیتار» ساز سیمی است یا «ترامپت» یک آلت موسیقی برنجی است . که در این دو جمله گیتار و ترمپت موجودیتهایی هستند که ما در باره آنها حقایقی را میدانیم

---
1 objects

رخدادها[1] : اعمالی که در دنیای ما رخ می دهد ، به عنوان مثال ، پرویز پرستویی در فیلم آژانس شیشه ای نقش آفرینی میکند.

اجراء[2] : رفتاری شبیه نواختن گیتار با دانشی درباره نحوه اجرای کارها درگیر است .

فرادانش : دانش ، درباره چیزهائی که می شناسیم . مانند ، خواهرم برنامه یک سفر دارد. خواهرم می داند که علائم خیابانهای طول مسیرش را جهت آگاهی از موقعیت خودش می تواند بخواند. بنابر این در حل مسائل موجود در AI باید دانش ارائه دهیم و دو موجودیتی که با آنها سروکار داریم ، عبارتند از :

حقایق[3] : واقعیت هائی درباره دنیای واقعی و چیزی که ما نمایش می دهیم و به آن «سطح دانش» میگوییم .

ارائه حقایق : آنچه که دستکاری می کنیم . ارائه حقایق می تواند به «سطح نماد» نسبت داده شود ، از اینرو معمولاً حقایق با ترکیبی از نمادها نمایش داده میشود که با برنامه های کامپیوتری بتوان آنرا تغییر داد . این موجودیتها را می توانیم در دو سطح سازماندهی نمائیم :

سطح دانش ، جائی که حقایق تعریف شده اند .

سطح نماد ، نمایش موجودیتها با نمادهایی صورت میکیرد که برنامه ها بتوانند آنرا دستکاری کنند.

زبان انگلیسی یا زبان طبیعی یک روش واضح از ارائه و بررسی حقایق است . منطق ، ما را در بررسی حقایق زیر توانا می سازد «تام یک گربه است » و استنباط ما این است که «گربه ها دم دارند» بنابراین می توانیم نتیجه بگیریم : که « تام دم دارد» .

$$\text{cat(Tom)}$$

$$\text{cat(x)} \rightarrow \text{hasatail(x)} \qquad \Rightarrow \qquad \text{hasatail(Tom)}$$

با استفاده از دانش گذشته و تسری آن به حقیقت اول در استنتاج « تام دم دارد» تولید میشود.

---

1 Events
2 performance
3 facts

توابع در دسترس همیشه یک به یک نیستند ، بلکه چند به چند می باشند که مشخصه ای از ارائه ها در زبان طبیعی است (چه انگلیسی چه فارسی) .هر دو جمله « همه سگها دمهائی دارند» و «هر سگ یک دم دارد» بیان می کنند که هر سگ یک دم دارد اما اولین جمله را می توان چنین تفسیر کرد که هر سگ بیش از یک دم دارد ؛ این کار را مجدداً با تعویض دم با دندان انجام دهید . وقتی یک برنامه AI یک ارائه داخلی از حقایق را دستکاری می کند ، این ارائه های جدید نیز باید به عنوان ارائه های جدیدی از حقایق قابل تفسیر باشند .

۳-۵-۲ کاربرد دانش

تا کنون به طور خلاصه مواردی را که دانش در سیستمهای AI بکار گرفته شده است را ذکر کردیم . اجازه دهید ، قدری بیشتر درباره چگونگی کاربرد ها و چند و چون دانشی که ممکن است استفاده شده باشد ، بررسی کنیم .

یادگیری[1] : به معنای فراگیری دانش است . فراگیری بیش از افزودن یک حقیقت جدید به پایگاه دانش است . در این فرآیند دانش جدید به پایگاه دانش افزوده شده است و دانشی که پیش تر حاصل شده تصفیه یا اصلاح میگردد .

بازیابی[2] : چارچوب ارائه استفاده شده ، می تواند تاثیر کلیدی روی بازده و کارآرائی یک روش بگذارد . انسانها در این کار خیلی خوب و مناسب عمل میکنند .

استدلال : استنباط حقایق از داده های موجود .اگر یک سیستم تنها بداند که :

« تجویدی یک نوازنده ویولون است » و «همه نوازندگان ویولون می توانند به خوبی بنوازند » .اگر سوالهایی مانند « آیا تجویدی یک نوازنده ویولون است ؟» یا «آیا نوازنده های ویولون می توانند به خوبی بنوازند؟» پرسیده شده باشد ، پاسخ سئوالات به سهولت از داده های ساختیافته و رویه های به دست می آید . به هر حال ، سئوالی نظیر « آیا تجویدی سازش را خوب می نوازد ؟» به استدلال نیاز دارد . همه موارد بالا به هم مربوط هستند . برای مثال ، کاملاً واضح و روشن است که یادگیری و استدلال با

1 learning
2 retrieval

بازیابی داده ها مرتبط هستند .

۳-۵-۳ فرم در برابر محتوای دانش

فرض کنیم میخواهیم با یک سیستم جمع دو عدد را انجام دهیم . این کار را می توان از طریق ذخیره کردن مجموع همه جفتهای ورودی اعداد صحیح قابل پذیرش در یک جدول جستجو اجرا کرد . راه دوم استفاده از یک شمارنده است که این شمارنده پی در پی به وسیله دو عدد صحیحی که مجموعشان را قرار است به دست آوریم افزایش می یابد . از نظراحرایی ، دو روش پاسخهای یکسان ایجاد خواهند کرد . اما از نقطه نظر ارائه ای و نمایشی تفاوت های مهمی وجود دارد که به طور مؤثر بر نحوه توانایی ما در انجام وظایف محوله تاثیر می گذارد . استفاده از جدول جستجو ما را با سرعت بیشتری به نتیجه میرساند اما اگر با ارقام بزرگ سروکار داشته باشیم ، رسیدن به این سرعت نیازمند نگهداری و ذخیره اعداد در یک حافظه بزرگ است . روش شمارنده از نظر عدم نیاز به سخت افزارهای پیشرفته بسیار کارآمد است اما خیلی کندتر پاسخ سوال را فراهم میکند. بنابراین ، استفاده از ساختارهای خاص و دانشی که در قالب مناسب توصیف شده اند ، تاثیر مهمی بر حل مسائل دارد . به علاوه ، از آنجائیکه هیچ نمایش یا مدل انفرادی نمی تواند همه جنبه های یک شئ واقعی را در برگیرد ، یک موجودیت هوشمند باید طیف گسترده ای از نمایشها و ارائه ها را جهت ارتباط با دنیا بکار گیرد .

۳-۵-٤ ارائه دانش

نمایش و ارائه یک وضعیت (یا شئ ، یا مسئله) یک انتقال به سیستمی است که شامل واژگانی به نام های اشیاء و ارتیاطات و اقداماتی که میتواند روی شیء صورت گیرد و حقایقی و محدودیتهایی است که از آن شیء میشناسیم.

مشخصات اختصاصی پایه از یک نمایش و ارائه عبارتند از:

الف) چه اطلاعاتی صریح و روشن ساخته شده اند .

ب) چگونه اطلاعات به طور فیزیکی بیان شده اند .

هدف از ارائه ، ساده کردن پاسخ به گروه محدودی از سئوالات درباره موقعیت داده شده است بنابراین انتخاب نوع ارائه باید در مسیر هدف باشد .

برای حل یک مسئله در دنیای واقعی حداقل دو ارائه مجزا برای تطبیق و انتخاب از بین چند مکانیزم محاسبه ای داده شده لازم است : اولین ارائه ، لوازم سمبلیک و نمادین کارآمد را برای پاسخ دهی به سئوالات درباره وضعیت داده شده ، فراهم می کند و دومی ، تکنیک های راه حل اولی را به دستورالعملها و ساختار های ذخیره سازی مناسب در ماشین تبدیل می کند .

بنابراین می توانیم فکر کنیم پایگاه دانش نگاشتی بین یک شی و ارتیاطاتش در حوزه مسئله و اشیاء محاسباتی و ارتیاطاتش در محیط برنامه است . نتایج استنباطها مبتنی بر پایگاه دانش باید با نتایج اعمال یا مشاهدات در دنیا مشابه باشد . اشیاء محاسباتی ، روابط و استنباطهای قابل دسترس ، برای برنامه نویسان به وسیله زبان ارائه دانش که انتخاب می کنند ، تعیین میشود . یک زبان مناسب می تواند به برنامه نویس در جمع آوری ، سازماندهی و غلط یابی پایگاه دانش کمک کند .

قالب های متفاوت زیادی برای ارایه دانش مطرح هستند که هرکدام نقاط ضعف و قوت خود را دارند . مایلوپولس۱ و لوسک۲ (۱۹۵٤) این قالبها را به چهار گروه تقسیم بندی کرده اند :

الف) قالب نمایش منطقی

ب) قالب نمایش رویه ای۳

ج) قالب نمایش شبکه ای

د) قالب نمایش ساخت یافته

قالب های نمایش منطقی

این دسته از نمایش ها از عبارات موجود در منطق نمادین برای نمایش دادن پایگاه دانش بهره می گیرد . قوانین استنباطی و رویه های استدلالی این قالب ارایه دانش را برای حل مسائل شهودی مناسب کرده است . ریاضیات اخباری مرتبه اول در قالب ارایه منطقی بسیار بکار میرود، پرولوگ یک زبان برنامه نویسی ایده ال برای پیاده

1 Mylopoulos
2 Levesque
3 procedural representation Schemes

سازی و اجرای قالبهای نمایش منطقی است .

قالب های نمایش رویه ای

قالبهای رویه ای ، دانش را به مثابه مجموعه ای از دستور العمل ها برای حل یک مسئله نمایش می دهد . این روش با نمایش های اعلانی که به وسیله منطق و شبکه های معنائی فراهم شده اند مغایرت دارد . به عنوان مثال ، در یک سیستم قاعده گرا[1] ، قاعده if...then ، ممکن است به عنوان رویه ای برای حل کردن و رسیدن به یک هدف در حوزه مسئله تفسیر شود .

برای حل استنتاج ، حل کردن فرض های پایه ای و مقدماتی اولویت دارد . سیستم های تولیدی ، مثال هائی از طرح های نمایش رویه ای هستند.

طرحهای نمایش شبکه

نمایش های شبکه ای ، دانش را به عنوان یک گراف در نظر می گیرد که گره های این گراف موضوعات یا مفاهیم موجود در دامنه مسئله و یالهای آن روابط یا وابستگی بین موضوعات را نشان می دهد . مثالهائی از نمایش های شبکه شامل «شبکه های معنائی» و «وابستگی های ادراکی» و «گراف های ادراکی» می باشند .

قالب های نمایش ساخت یافته

زبان های نمایش ساخت یافته ، به شکلی شبکه ها را گسترش میدهد تا بتوانند در هر گره ساختار داده ای پیچیده ای را به نام «اسلات» را نگهداری و مقادیری را به آن نسبت دهد. این مقادیر ممکن است ارقامی ساده یا داده نمادین ، اشاره گر هائی به دیگر قالبها ، یا حتی رویه هائی برای اجرای وظیفه ای خاص باشند . مثال هائی از نمایشهای ساخت یافته شامل اسکریپت ها وقالبها[2] و اشیاء[3] است .

۳-٦ قالب های نمایش منطقی

قالب های نمایش این گروه مشتمل بر جبر گزاره ای و مسندی است . این جبرها ،

---

1 rule base
2 Frame
3 Object

زبان های نمایش برای AI هستند . هر دو نوع جبر در فصل قبل شرح داده شده اند .

۳-۷ قالب های نمایش رویه ای

این قالب نمایشی سیستم های خبره قاعده گرا را بنیان نهاده است . یک سیستم خبره یک برنامه دانش گرا یا دانش محور است که راه حل های را برای مسائل موجود در حوزه ی خاص و با «کیفیت خبره» فراهم می کند . مثال ۳-۱ نمونه ای از کاربرد استدلال بر مبنای جملات اخباری غیر متوالی مبتنی بر پایگاه دانش کشاورزی است.

مثال ۳-۱ جملات اخباری اگر...آنگاه که در پایگاه دانش کشاورزی قرار دارد

اگر چراغ روشن باشد
آنگاه بهترین گیاه «بگونیا» است
اگر چراغ کم نور باشد
آنگاه بهترین گیاه پیچک است
اگر نور ، نور خورشید باشد
آنگاه نور ، درخشان است
اگر نور، نور لامپ باشد
آنگاه نور کم سو است
اگر مکان ، مکانی سرباز باشد
آنگاه نور ، نور خورشید است
اگر مکان سرپوشیده باشد
آنگاه نور ، نور لامپ است

۳-۸ قالبهای نمایش شبکه ای

از نظر تئوری های ارتباطی مفهوم هر شی در ذهن یا پایگاه دانش در شبکه ای پیوند خورد با دیگر اشیاء تعیین میشود . اگرچه در دنیا اشیاء با نمادها مشخص میشوند اما این فقط وسیله ای برای حفظ و ذخیره دانش است . زمانی که راجع به یک موضوع درک و استدلالی داریم ، در ابتدا آن درک و احساس در مفهوم موجود در ذهن ما نگاشته شده است . این مفهوم قسمتی از دانش تمام و دست نخورده ما از جهان است

و با سایر مفاهیم روابطی مناسب برقرار میکند . این روابط فهم و ادراک ما را از ویژگیها و خواص و رفتار یک موضوع مانند «برف» تشکیل می دهد ، برای مثال به واسطه تجربه ، مفهوم "برف" را با مفاهیم دیگر از قبیل سرما ، سفیدی ، آدم برفی ، لیز خوردن ، و یخ ارتباط می دهیم . درک ما از برف و درستی عباراتی مانند «برف سفید است » به دلیل شبکه ارتباطی بین اشیاء ، حقیقت خودش را آشکار و معلوم می کند . در اینجا دو نوع قالب نمایش شبکه ای که در زیر مطرح شده اند را خواهیم دید :

۳-۸-۱ شبکه های معنائی[1]

شبکه های معنائی ، جایگزینی برای منطق گزاره به عنوان یک شکل و فرم از نمایش دانش هستند . این ایده بر این اصل استوار است که ما می توانیم دانش خود را در در باره موجودات هستی در یک گراف با گره هائی که بیانگر اشیاء و یالهائی که نشاندهنده روابط بین این اشیاء هستند ، بگنجانیم .

گرافها از طریق ایجاد نمایش صریح روابط بین یالها و گره ها ، ثابت کرده اند که وسیله ای مناسب برای رسمی کردن نظریه های ارتباط (اجتماعی) دانش هستند . یک شبکه معنائی ، دانش را به عنوان یک گراف با گره هائی متناظر با حقایق یا مفاهیم و یال هائی که بین مفاهیم ارتباط و وابستگی برقرار می کنند ، نمایش می دهد . بعضی از اصول شبکه های معنائی (به عنوان مثال در شکل ۳-۳) به شرح ذیل هستند :

- شبکه های معنائی روابط بین اشیاءی را که در گره ها قرار دارد را وصف و تشریح می کنند .
- گره ها دایره های نامگذاری شده هستند.
- ارتباطات بین گره ها به وسیله یالهائی که به این دوایر وصل هستند نمایش داده شده اند .
- یک شبکه معنائی میتواند برای تولید ساختارها و اشیاء استفاده شده باشد .
- یک شبکه معنائی می تواند برای تولید قوعد یک پایگاه دانش استفاده شده باشد .

---

1 semantic networks

شبکه معنائی شکل ۳-۳ میخواهد داده های زیر را نمایش دهد :

تام یک گربه است .

تام یک پرنده می گیرد .

تام مال جان است .

تام رنگ تندی دارد .

گربه ها خامه دوست دارند .

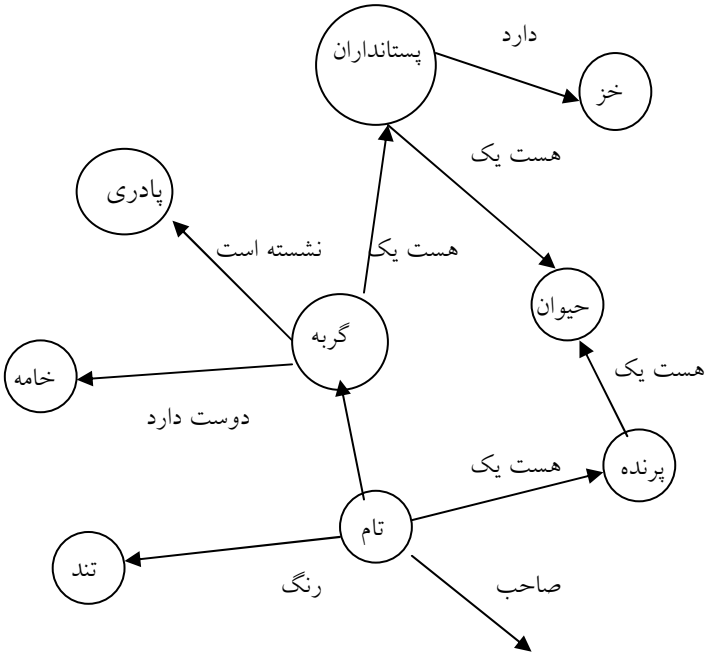
گربه روی پادری نشسته است .

گربه جزو پستانداران است .

پرنده یک حیوان است .

همه پستانداران جزو حیوانات هستند .

پستانداران خز دارند .



شکل ۳-۳

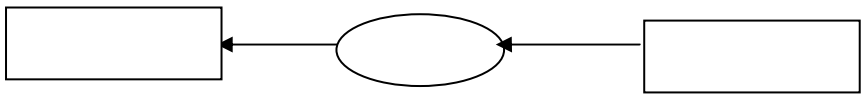
این قالب از نمایش به روش دانش ساختاری بشر نزدیکتر است ، البته ذهن بشر با وجود بهره گیری از ساختار شبکه ای توانایی برقراری ارتباط با منطق گزاره ای را نیز

دارد .

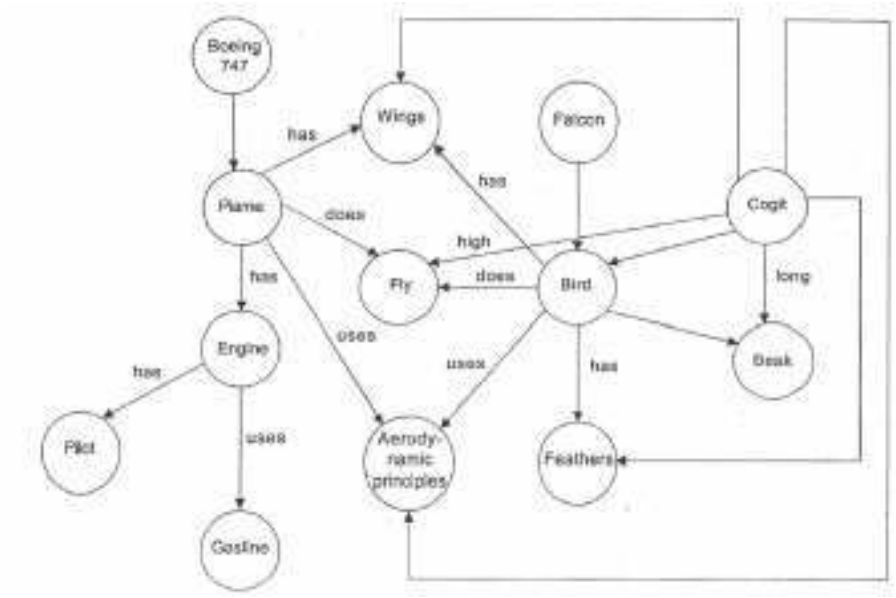
عبارت «هست یک» در شکل ۳-۳ یک پیوند با دو تفسیر متفاوت است . عبارت فوق را می توان به وجود یک نمونه منفرد از یک گروه تفسیر کرد مانند «تام عضو گروه گربه هاست» ، یا اینکه میتوان آنرا به وجود یک گروه که زیر مجموعه از گروه دیگر است تفسیر نمود مانند « گروه گربه ها زیر مجموعه گروه پستانداران است» . این گیجی و اشتباه در منطق هرگز رخ نمی دهد .

۳-۸-۲ گرافهای ادراکی

یک گراف ادراکی یک گراف متناهی ، متصل و دوقسمتی است . گره های گراف نشاندهنده ارتباطات مفهومی یا ادراکی هستند . گرافهای ادراکی از یال های برچسب دار استفاده نمی کنند و به جای آن گره های رابط که بین دو مفهوم قرار میگیرند استفاده میکنند. به مثال زیر را توجه کنید:



در اینجا «سگ» و «قهوه ای» گره های مفهومی هستند و رنگ رابطه مفهومی است . برای تشخیص دادن این نوع گره ها ، مفاهیم به وسیله جعبه ها (مستطیل) و روابط مفهومی به وسیله بیضی ها نمایش داده می شوند . شکل ٤-۳ یک شبکه معنائی را نمایش می دهد که روابط بین اجزاء یک پرنده و یک هواپیما را نشان می دهد .



شکل ۳-٤ شبکه معنایی که ارتباط بین اجزاء پرنده و هواپیما را نشان میدهد

خصوصیات گراف های ادراکی :

- گره های مفهومی یا موضوعات انتزاعی و یا واقعی از جهان مورد بحث را نمایش می دهند .
- گره های ارتباط مفهومی رابطه ای را نشان می دهد که با یک یا چندین مفهوم در تعامل است .
- هر گراف مفهومی یک گزاره منفرد را نمایش می دهد . یک پایگاه دانش متعارف شامل تعدادی از این گراف هاست. ممکن است گراف ها پیچیده باشند ، اما باید حتماً متناهی باشند .
- نظریه گراف های مفهومی ، شامل عملیات ها یی است که اجازه ایجاد گرافهای جدید را از گرافهای موجود می دهد.

عملیات ها روی گراف های ادراکی :

عملیاتی روی گراف های ادراکی ، به ما اجازه ایجاد یک گراف جدید از یک گراف موجود را می دهد . این کار به وسیله چهار عملیات که کپی کردن ، محدود کردن ، متصل کردن و مختصر کردن نام دارند ، انجام می گیرد .

فرض کنید که $g1$ و $g2$ دو گراف مفهومی باشند ، آنگاه :

- قاعده کپی به ما اجازه شکل دهی گراف جدید ، G که یک کپی از $g1$ است را می دهد .
- «محدودیت» به گره های مفهومی در یک گراف اجازه می دهد تا جایگزین گرهی شوند که ویژگیهای آنرا نمایش می دهد .
- قانون متصل کردن ، به ما اجازه ترکیب دو گراف در قالب یک گراف منفرد را می دهد .
- اگر یک گراف محتوی دو رابطه تکراری باشد ، آنگاه ممکن است یکی از آنها حذف شود . این قاعده ، مختصر کردن نام دارد.

۳-۸-۳ وابستگی مفهومی

راجر شانک۱ و دانشجوانش نظریه ای را به نام «وابستگی مفهومی» توسعه دادند، ادعای آنان بر پایه این نظریه این بود که میتوان بسیاری از داستانها را با استفاده از تعداد کمی از مفاهیم نمایش داد . او تعدادی سیستم NL_2 را توسعه داد که جملات توصیفی انگلیسی داستانهای موجود را به محیط «نمایش و ارایه» خود نگاشت میکرد. در اینجا جملات مختلف انگلیسی با معنای یکسان با «نمایشی» یکسان نگاشت می شوند .

در وابستگی ادراکی۳ (CD) ساختارهای «شکافنده» و «پرکننده» ، برای نمایش دانشی استفاده میشوند که درباره وقایعی که معمولاً در جملات زبان طبیعی نقل می شوند هدف این کار نمایش دانش برای تحقق اهداف زیر است :

- تسهیل کردن رسم استنباطها از جملات .
- مستقل بودن از زبان در جملات اصلی .

نظریه وابستگی ادراکی

- برای هر دو جمله ای که در معنی یکسان هستند ، صرف نظر از زبان ، فقط باید یک نمایش وجود داشته باشد .

---

1 Roger Schank

2 Natural Language زبان طبیعی

3 Conceptual Dependency

- هر اطلاعاتی در جمله که تلویحی و ضمنی است باید در نمایش معنی آن جمله به صورت صریح ایجاد شود .

بلوک های ساختمان :

وابستگی ادراکی : ۱۱ عمل اولیه :

ATRANS : انتقال یک رابطه انتزاعی (مانند ارائه دادن)

PTRANS : انتقال موقعیت فیزیکی یک شئ (مانند رفتن)

PROPEL : کار برد نیرو فیزیکی در یک شئ (مانند فشار دادن)

MOVE : جابجائی عضوی از بدن به وسیله صاحب خودش (مانند لگد زدن)

GRASP : نگاه داشتن یک شئ از طریق یک عملگر (مانند کلاچ)

INGEST : بلعیدن یک شئ به وسیله یک حیوان (مانند خوردن)

EXPEL : اخراج و دفع چیزهائی از بدن یک حیوان (مانند گریه کردن)

MTRANS : انتقال اطلاعات ذهنی (مانند گفتن)

MBUILD : درست کردن اطلاعاتی جدید از روی گذشته (مانند تصمیم گرفتن)

SPEAK : تولید اصوات (مانند گفتن)

ATTEND : توجه یک اندام حسی به طرف محرک (مانند گوش دادن)

چهار مجموعه مفهومی اولیه برای ایجاد ساختار های وابسته :

ACTs : اعمال

PPs : اشیاء (تولید کننده های تصویر)

AAs : تعدیل کننده های اعمال (عمل aider)

PAs : تعدیل کننده های تصویر (تصویر aider)

وابستگی ها در میان تصورات با روابط معنایی در میان مفاهیم اساسی ، مشابه هستند .

مزایای اصلی وابستگی مفهومی

شرح قوانین استنتاجی با دانشی که می تواند دستکاری شود ، ساده تر است . قوانین می توانند یک بار برای هر عمل (ACT) سریعتر از تمامی لغاتی که آن عمل (ACT) را توصیف می کند ، نمایش داده شوند . به عنوان مثال دادن ، گرفتن ، دزدیدن و بخشیدن

همه نمونه هائی از ATRANS تلقی می شوند و می توانند استنتاج های یکسانی درباره کسی که این شئ را دارد ، و کسی که یکبار آن شئ را داشته است ، ایجاد کنند .
برای ایجاد نمایش CD اطلاعات صریحی که درمتن بیان نشده اند ، را به وجود می آوریم . مانند «سعید کتاب را از گیتا گرفت» و اطلاعاتی مانند «گیتا مدت زیادی کتاب را در اختیار نداشته است» را به طور صریح ایجاد می کنیم . این کار ممکن است ، فهم عبارت بعدی مانند «گیتا چیزی برای خواندن ندارد » را آسان تر کند.
باید رئوس مطالب یک مختصرساز از یکی از سیستم های راجرشانک را تهیه کنیم .این روش «مکانیزم تقاضا کننده اسکریپت» یا SAM[1] نامیده میشود. این عمل با در اختیار گرفتن نمایشی از یک داستان و ادغام آن با یک اسکریپت استاندارد می باشد که این اسکریپت برای استخراج سناریو و ایجاد تفسیری از موقعیت می تواند استفاده شود. بنابراین در این روش ما خواستار فهمی از داستان در اصطلاحات اسکریپت هستیم . راجر شانک و دوستانش معتقدند که برای فهم خیلی از موقعیت های بشر به تعداد کمی اسکریپت نیاز داریم .
شانک حدود ١١ عمل (ACTs) پایه ای برای ارائه و نمایش معانی به کار برد . که فقط چهار مورد آنها را برای فهم ساده تر فهرست وار می آوریم :
١- (شئ ١ ، شئ٢ ، مکان ١، مکان ٢) ptrans :موقعیت یک شئ فیزیکی (شئ١) را به موقعیت شئ ٢ از مکان ١ به مکان ٢ تغییر می دهد .
٢- بلعیدن (بازیگر ، غذا ، ثروتمند) : گرفتن چیزهائی از داخل یک شئ جاندار - بازیگر غذا را با الگوبرداری از انسانهای ثروتمند می خورد.
٣- (بازیگر ١، شئ ، بازیگر ٢ ، بازیگر ٣) atrans ، تغییر یک رابطه مطلق ازقبیل مالکیت/مالکیت یک شئ -بازیگر١ با انتقال ثروت بازیگر ٢ به بازیگر ٣ موضوع را تغییر می دهد .
٤- (بازیگر١ ، info ، بازیگر ٢) mtrans ، اطلاعات را انتقال می دهد - انتقال

---

1 ***Script Applier Mechanism***

اطلاعات از بازیگر ۱ به بازیگر ۲
مفاهیم دیگری از قبیل زمان ، موقعیت و خواص نیز در کنار ACT ها وجود دارد .

۳-۹ قالب های نمایش ساخت یافته

چگونه یک کامپیوتر می تواند توجه خودش را فقط بر روی جنبه هائی از مسئله داده شده که به راه حل مربوط است ، معطوف کند ؟ این یک مسئله مربوط به «قالب» است (که به وسیله مارتین مینسکی اختراع شده است) .
یک سیستم هوشمند ، در تلاش برای حل یک مسئله یا انجام یک عمل چگونه می تواند دریابد ، که از چه اطلاعاتی در پایگاه داده باید صرف نظر شود و چه اطلاعاتی به مسئله موجود مربوط است ؟

۳-۹-۱ قالب ها

هرچه وظایف پیچیده تر می شوند ، لازم است که ارائه و نمایش آنها ساخت یافته تر باشد . ساختیافته تر بودن سیستم ، برای بکار بردن قالبها سودمندتر است . یک قالب ، مجموعه ای ازخصوصیات یا اسلات ها و مقادیر شرکت کننده می باشد که بعضی موجودیت های دنیای واقعی را توصیف می کند . قالبها به شکل خاصی در ساختار خودشان مفید نیستند ، اما سیستمهای قالب یک روش قدرت مند از رمزگذاری اطلاعات برای پشتیبانی کردن استدلال می باشند . نظریه مجموعه ها پایه و اساس خوب و مناسب برای درک سیستم های قالب را فراهم می کنند .
ما دانش را با استفاده از شبکه نمایش و ارتباطات صریح و روشن یا ارتباط بین اشیاء در پایگاه دانش ساماندهی شده در نظر میگیریم . می توانیم دانش را در واحد های پیچیده تر که موقعیت ها یا اشیاهای پیچیده در حوزه را نمایش می دهند سازمان دهی کنیم . این ها " قالبها " یا "شما " نامیده میشوند.
هر قالب منفرد ممکن است به عنوان یک ساختار داده در نظر گرفته شده باشد ، همانند خیلی از اطلاعات در رکورد سنتی آنها ، که این رکورد حاوی اطلاعاتی مربوط به موجودیت های کلیشه ای و فاقد نبوغ می باشد.
شکاف ها درقالب حاوی اطلاعاتی از قبیل موارد زیر هستند :

الف) اطلاعات بازشناسی قالب

ب) رابطه بین یک قالب و قالبهای دیگر . " تلفن هتل " ممکن است نمونه خاصی از تلفن باشد ، یا به شکلی دیگر نمونه ای از یک "دستگاه ارتباطی " باشد .

پ) توصیف کنندگان تقاضاها برای مطابقت با قالبها . به طور مثال ، نشیمنگاه یک صندلی بین ۲۰ تا ٤۰ سانتیمتر از کف فاصله دارد و قسمت پشتی آن ارتفاعی بیش از ٦۰ سانتیمتر دارد . ممکن است این تقاضاها برای تعیین اینکه چه موقعی یک موجودیت جدید با رفتار تعریف شده در قالب تطبیق دارد بکار روند.

ت) اطلاعات رویه ای در استفاده از ساختار های توصیف شده : خصوصیت مهم قالبها توانائی الصاق کد رویه ای به یک اسلات است .

ث) اطلاعات پیش فرض قالب : این اطلاعات مقادیر شکاف هستند که برای صحیح بودن در زمانی که هیچ مدرکی ، مخالف یافت نشده است در نظر گرفته شده اند . به عنوان مثال ، صندلی ها چهار پایه دارند ، تلفن ها دکمه فشاری دارند ، یا تخت های هتل به وسیله کارمندان ساخته شده اند .

ج) اطلاعات نمونه جدید . خیلی از شکاف های قالب ممکن است نامعلوم بمانند تا زمانی که یک مقدار برای یک نمونه خاص داده شده یا برای حل مساله لازم شده باشد . برای مثال رنگ چادر شب تختخواب ممکن است در تعریف تختخواب نامعلوم بماند .

مثال زیر را در نظر بگیرید :

مثال ۲-۳

شخص هست یک : پستاندار

مرد بالغ هست یک : شخص

بازیکن فوتبال هست یک : مرد بالغ

قد: وزن: پست بازی :

تیم :

رنگ تیمها : سعید هست یک : بازیکن فوتبال

کسب امتیاز توسط : مجید شامخی عضو :سعید

دسته : یک پست : میانی

تیم : پیام نور رنگ پیراهن های تیم : سیاه / آبی

تیم فوتبال هست یک : تیم

تعداد اعضا : ۱۵ نفر

مربی : جلال دوستی

در اینجا قالبهای شخص ، مرد بالغ، بازیکن فوتبال ، تیم فوتبال ، همگی طبقه بندی هستند و قالبهای مجید شامخی و پیام نور نمونه های آنها می باشند .

بعضی از خصوصیت های مهم قالبها عبارتند از :

- قالب ها شبکه های معنائی را از راههای مختلف گسترش می دهند مهم ترین آنها سازمان دهی دانش در ساختارهاست و این موضوع مهمی برای پایگاه دانش است .
- الحاق رویه ای ، تا وقتیکه دانش معین با نمایش ها به خوبی وفق داده نشده اند ، یک خصوصیت مهم ویژه از قالبها است.
- نمایش دادن دانش با سیستم قالب ، اگرچه دارای اطلاعات ناتمام می باشد ، حداقل تاحدی به ما اجازه استدلال کردن و استنتاج سریع حقایقی که به طور صریح مشاهده و آشکار نشده اند را می دهند.
- یکی از مشکلات نمایش قالبی ، دشواری تعیین مقادیر اولیه به طور صحیح و دقیق برای یک قالب می باشد .

۳-۹-۲ اسکریپت ها

طبق تعریف ، یک اسکریپت یک ارائه ساخت یافته است که یک رشته حوادث و وقایع کلیشه ای از حقایق را در یک محتوای مخصوص توصیف می کند . در اصل اسکریپت به وسیله شانک و گروه تحقیق اش (شانک و آبلسون ۱۹۷۷) برای سازمان دهی ساختار های وابستگی مفهومی به توصیف وضعیتهای متعارف طراحی شده اند . اسکریپت ها در سیستم های درک زبان های طبیعی ، به عنوان اجزای پایگاه دانش استفاده شده اند .

اجزای یک اسکریپت عبارتند از :

- توصیف گران یا شرایط ورودی از دنیا که باید برای اسکریپتی که فراخوانده شده ، صحیح باشند . در یک رستوران اسکریپت ، رستوران باز است و مشتری گرسنه شناختن اجزای دنیا در قالب جملات درست می باشد .
- نتایج یا حقایق وقتی که اسکریپت خاتمه یافته است ، درست هستند . برای مثال ، آن مشتری گرسنه نیست و پول کمی هم دارد . صاحب رستوران پول بیشتری دارد.
- اثاثیه۱ یا هر چیزی که محتوای اسکریپت را درست می کند . این اثاثیه ها ممکن است شامل میزها ، پیشخدمت ها ، و منو غذا باشد . این حالت اجازه می دهد ، فرض های قراردادی قابل استدلال درباره موقعیت ها داشته باشیم ، برای مثال ، فرض شده است که یک رستوران میز و صندلی دارد مگر اینکه لفظی غیر از این را بیان کرده باشد.
- نقش ها اعمالی هستند که شرکت کنندگان فردی آنها را انجام می دهند . پیشخدمت سفارشها را می گیرند ، غذا تحویل می دهند و صورت حساب را ارائه می نمایند . مشتری سفارش می دهد ، می خورد و صورت حساب را پرداخت می نماید .
- شانک اسکریپت را به ترتیبی به صحنه۲ ها شکست ، که هر کدام از آنها یک خصوصیت موقتی از اسکریپت را ارائه می دهد. در رستوران ورود ، سفارش و خوردن و ... وجود دارد .

اسکریپت ها در شرح یک موقعیت معین از قبیل "سرقت از بانک " مفید هستند . که ممکن است با موارد زیر درگیر باشند :

- به دست آوردن یک تفنگ .
- سرقت مسلحانه از بانک .
- گریختن از بانک به همراه پول .

---

1 Props
2 Scene

اینجا اثاثیه ممکن است شامل موارد زیر باشد:

- تفنگ ، ت .
- غارت ، غ .
- کیف ، ک .
- فرار ماشین ، م .

نقش ها ممکن است شامل موارد زیر باشد :

- سارق ، S
- صندوق دار ، M
- مدیر بانک ، O
- پلیس ، P

انگیزه ها ممکن است شامل موارد زیر باشد :

- S فقیر است .
- S بینوا است .

نتایج ممکن است شامل موارد زیر باشد :

- S پول بیشتری دارد
- O عصبانی است
- M در حالت شوک است
- P گلوله است .

سه صحنه وجود دارد : به دست آوردن تفنگ ، سرقت بانک و گریختن. اسکریپت کامل همان طور که در جدول دیده شده توصیف میگردد .

برخی نکات اضافی برای یادداشت روی اسکریپت ها :

- اگر یک اسکریپت خاص به کار برده شده باشد ، باید فعال شده باشد و این فعال بودن به اهمیتش بستگی دارد .
- اگر موضوع در حال عبور ذکر شده باشد ، آنگاه یک اشاره گر می تواند برای آن اسکریپت نگهداری شده باشد .

- اگر موضوع با اهمیت است ، آنگاه اسکریپت باید باز شده باشد .
- دروغهای خطرناک در داشتن تعداد زیادی اسکریپت فعال حتی به اندازه یکی هم ممکن است پنجره های زیاد بازی را با پرده یا فراخوان های بازگشتی بسیار در یک مسئله ، داشته باشد .
- حوادث ایجاد شده ، یک دنباله شناخته شده را دنبال می کند که ما می توانیم از اسکریپت ها برای ارائه اعمال درگیر شده و جهت پاسخ دهی به سئوال های جزئیاتی بکار گیریم .
- دنباله های مختلف ممکن است برای پی آمد های متفاوت از یک اسکریپت مجاز باشد (به عنوان مثال سرقت بانک اشتباه است .)

مزایای اسکریپت ها :

- توانائی جهت پیشگوئی وقایع
- تفسیر منسجم انفرادی ممکن است از مجموعه ای از مشاهدات ساخته شده باشد .

معایب اسکریپت ها :

- عمومیت کمتر نسبت به قالبها .
- ممکن است برای ارائه کلیه انواع دانش مناسب نباشد .

در اینجا مثال دیگری از اسکریپت ها (مثال ۳-۳) در ابتدا به فارسی و سپس در فرم قالب ارائه ای داده شده است .

اسکریپت های یک رستوران

۱- مشتری به رستوران می رود .
۲- مشتری به سمت میز می رود .
۳- پیشخدمت غذا می آورد .
٤- مشتری غذا می خورد
٥- مشتری به پیشخدمت پول پرداخت می کند .
٦- مشتری رستوران را ترک می کند.

حال اسکریپت را به عنوان یک اسکریپت تاکیدی PROLOG (نام ، اسکریپت ، قرارداد ها) که نام فقط همان نام اسکریپت ، اسکریپت لیستی از ارتباطات با مقادیر آزاد تشریح اسکریپت در مؤلفه هائی از ACTs می باشد ، و پیش فرض ها یا قراردادها لیستی از زوجها ، هم بسته کردن مقادیر برای نام های پیش فرض می باشند ، اگر آنها به وسیله مثال های واقعی که با زیر مجموعه ای از اسکربپت ها ترکیب می شود معرفی نشده باشد .

اسکریپت(رستوران ، [ptrans(بازیگر ، بازیگر ، مکان اولیه ، رستوران)،
Ptrans(بازیگر، بازیگر، در، میز)،
Ptrans(پیشخدمت ، غذا، آشپزخانه ، میز)،
بلعیدن(بازیگر، غذا، دهان(بازیگر))،
Atrans(بازیگر،پول،بازیگر،پیشخدمت)،
Ptrans(بازیگر، بازیگر، رستوران ،جائی دیگر)]،
[(بازیگر، مشتری)،(مکان اولیه ، مکان ۱)،
(رستوران،رستوران)،(در،در)،
(میز،میز)،(غذا،غذا)،
(پیشخدمت، پیشخدمت)،(آشپزخانه،آشپزخانه)،
(پول،صندوق)،(جائی دیگر،مکان۲)]).

فصل ششم

تکنیکهای جستجو

اهداف

در پایان فصل، دانشجو با مفاهیم زیر آشنا می‌شود:

جستجوی مهارتها

مشکل ارائه (نمایش)

تعاریف

ارائه برنامه

حل مسئله در هوش مصنوعی

تکنیک وسیع جستجو

روش ارتباط و انشعاب با رنامه ریزی دینامیکی (پویایی)

روش مینی مکس

٦-١ جستجوی مهارتها

در هوش مصنوعی جستجو ، به بدنه بزرگی از عقاید اصلی که منجر به ا ستنباط و استنتاج و برنامه ریزی و استدلال و اثبات قضیه و پردازش میشود ، مربوط است . .
کاربرد این عقاید کلی در پردازش زبان طبیعی و بازیابی اطلاعات و برنامه ریزی اتوماتیک و رباتها و تجزیه مراحل و بازی و سیستمهای خبره و اثبات قضیه های ریاضی یافت می شود.
به طور کلی سیستم های جستجو سه جزء دارند:

۱- پایگاه داده
۲- عملگرها
۳- استراتژی کنترل
پایگاه داده دامنه کاری جاری و هدف یا به عبارت دیگر اعمال کار را شرح می دهد .
عملگر ها برای دستکاری پایگاه داده استفاده می شوند
استراتژی کنترل تصمیم می گیرد که چه عملگری و کجا به کار گرفته شود
منظور از هر تکنیک جستجویی به کارگیری یک ترتیب مناسبی از عملگرها برای یک دامنه اولیه کاری برای رسیدن به هدف است.
دو راه برای رسیدن به هدف وجود دارد :
۱- استدلال پیشین
۲- استدلال پسین
استدلال پسین به کاربرد عملگرها
به آن ساختارهایی در پایگاه داده که دامنه کاری را به منظور ایجاد یک موقعیت تعیین شده شرح می دهند مربوط می شود .مثل روش استدلال پایین به بالا یا داده اشتقاقی ، که هدف این است موقعیت را از حالت اولیه به حالت پیشرو بیاورد .
برای مثال یک بازی شطرنج را در نظر بگیرید.
موقعیت اولیه . مشخص کردن مکان مهره های شطرنج روی تخته در شروع بازی
هدف. هر وضعیتی از تخته بازی که کیش و مات کند
عملگرها. قوانینی برای حرکت قانونی در شطرنج

استدلال پیشین یا بالا- پایین یا استدلال هدف هدایت شده ، جملات هدف (مسئله) را به " زیرهدف " ها یی می شکند است که امیدوارانه ، برای حل کردن آسانتر است و راه حل آن برای حل مسئله اصلی کافی است .
به عنوان مثال مسئله انتگرال گیری را در نظر بگیرید:

$$1/\cos 2 \; dx$$

اگرعملگر اجازه دهد :

1/cos2 dx
آنگاه ما می توانیم مجددا" به این صورت بیان کنیم:
Sec2x dx
جالب توجه است که بسیاری از استدلالهای ما استدلال پیشین است.

٦-٢ مشکل ارائه (نمایش)

یک سیستم حل مسئله هر دو صورت استدلال " پیشین" یا " پسین" را استفاده می کند که هر عملگر فقط برای تولید یک شیء جدید یا یک حالت جدید پایگاه داده کار می کند که به آن گفته می شود که مسائل را در یک فضای حالت نمایش دهد .
در این جا جالب است که توجه شود که در مورد استدلال پیشین ؛ دو حالت رخ می دهد:
١-کاربرد یک عملگر یک مسئله جدیدی را که اندازه یا سختی آن کمتر از مسئله اصلی است به بار می آورد .
٢-کاربرد یک عملگر در یک سیستم مجتمع یک مسئله را به مجموعه ای از "زیر مسائل" کاهش می دهد
شاید اهمیت کمتری از مسئله اصلی داشته باشد.
ارائه کاهش مسئله به چنین سیستمی گفته می شود
در هر حال بین کاهش ارائه مسئله و فضای حالت یک رابطه وجود داردو ممکن است برای حل مسائل در ترتیب " زیر مسئله " ها محدودیت وجود داشته باشد یا وجود نداشته باشد .
برای مثال : اگر عملگر اصلی در انتگرال گیری به صورت زیر باشد
[f(x)+g(x)] dx
بسته به نوع نمایش مسائل جدید میتواند به دو حالت دیده شده وجود داشته باشد .
. دو مسئله یکپارچه جدید که می تواند در هردر حل شود
. دو مسئله یکپارچه بعلاوه یک مسئله سوم f در انتگرال جمع شود
در آخرین حالت؛ تکلیف سوم نمی تواند اجرا شود مگر اینکه دو تکلیف اول زودتر کامل شوند.

۳-٦ تعاریف

در این کتاب ما بعضی از تکنیکهای جستجو را به دقت بررسی می کنیم. اما قبل از انجام این کار تعدادی از تعاریف اولیه را در اینجا شرح می دهیم:

نمودار: یک نمودار یک شیء داده ای است که شامل دو مجموعه است که رئوس و لبه نامیده می شوند. V یک مجموعه متناهی و غیر تهی از گره هاست و E یک مجموعه متناهی از جفت گره ها ست. هر جفت در E یک لبه در نموداراست. اگر هر جفت از رئوس به صورت(i,j) مرتب (که متفاوت از (j,i) می باشد) شده باشد سپس نمودار مستقیم است در غیر این صورت نمودار غیر مستقیم است .

درخت: یک درخت یک مجموعه متناهی از یک یا چند گره است. ویک گره ویژه که ریشه نامیده می شودوجود دارد.؛ گره های باقیمانده در مجموعه های گسسته ی T1.....Tn دسته بندی می شوند که هر یک از این مجموعه ها یک درخت است. T1.....Tn "زیر درخت" نامیده می شوند.

گره ریشه، گره نهایی و بچه ها: گره ی بالای درخت که پدر ندارد ریشه درخت نامیده می شود.

گره قعر درخت که فرزندی ندارد گره نهایی نامیده می شود. اگر هر گره ای به گره ها ی دیگر وصل شده باشد یا گره شاخه ها که گره های دیگر به آن وصل شدند فرزند آن گره نامیده می شوند.

فاکتور انشعاب،گسترش،شروع، پایان: اگر تعداد فرزندان همیشه برای همه گره هایی که فرزند دارند یکسان است این تعداد فاکتور انشعاب نامیده می شود. پردازش فرزندان به دست آمده از گره ها را گسترش گره گویند. گره ها گفته می شود باز شوند تا اینکه گسترش داده شوند سپس آنها مسدود می شوند(گره پایان).

مسیر:یک مسیر از راس Up به راس دیگرUq یک تناوبی از رئوس Up,Ui1,....Uin است چنان که (Up,Ui1) ,.....(Uin,Up) لبه ها در E(G) هستند. طول مسیر تعداد لبه ها در آن است .

هدف: هدف یک مسیر از ریشه درخت به حالت هدف است. حالت هدف ممکن است به دوصورت تعریف شود: به صورت صریح و واضح ؛ یا ترکیبی از حالتها که به حالتی معلوم دلالت می کند.

جستجوی یک راه حل بوسیله ایجاد یک درخت فضای حالت که شامل مسیر راه حل است انجام می شود.

برگشت در لیست:برگشت در لیست یکی از معمولترین تکنیکها در طراحی الگوریتم ارائه می دهد. برای به کارگیری این روش ، راه حل مطلوب باید بر یک بازه n تایی به صورت (g1.....gn) دلالت کردکه gi از مجموعه محدود از Di انتخاب می شود. اغلب مسئله ی که حل شده است برای بیشینه ساختن مقیاس کارکرد تابع P(g1....gn) فراخوانی می شود .

اکنون فرض کنید m اندازه Di باشد. آنگاه n تایی m = m1,m2m...,mn وجود دارد که ممکن است برای حل مقیاس تابع P کاندید شده باشد.

رویه"back track" فقط تا پلهایی که منجر به راه حل بهینه می شوند ارزیابی می کند. آن یک عقیده اساسی است تا بردار راه حل یک جزءی در یک زمان بالا ببرد و مقیاس اصلاح شده تابع Pi(g1,...,gn) را استفاده کند برای تست اینکه آیا مسیر موجود هیچ شانسی برای موفقیت دارد یا نه.

اگر در حالتی احساس شود که مسیر انتخاب شده نمی تواند به سوی یک راه حل بهینه هدایت شود آنگاه ممکن است آزمایش مسیر انتخاب شده کاملا" نادیده گرفته شود.

.برنامه نویسی پویا: برنامه نویسی پویا یک روش طراحی الگوریتم است . که می تواند وقتی استفاده شود که حل یک مسئله به عنوان نتیجه یک توالی از تصمیمات دیده شود. به هر حال در بعضی از حالات ؛چنین تصمیم تدریجی (فقط بر اساس اطلاعات محلی) ممکن است برای ساختن مقدور نباشد. یک راه برای بدست آوردن راه حل بهینه در چنین مواردی این است که همه مسیرهای ممکن را امتحان کرده وبهترین مورد را انتخاب کنیم. برنامه نویسی پویااغلب به طور موثری میزان شمارش را بوسیله جلوگیری از شمارش ترکیبی از تصمیماتی که نمی توانند بهینه باشند کم می کند.

شاخه و حد: شاخه و حد به همه روشهای جستجوی فضای حالت ، که همه فرزندان گره ریشه یا گره جاری قبل از هر گره دیگری که می تواند گره جاری شود ایجاد شده اند مر بوط می شود .

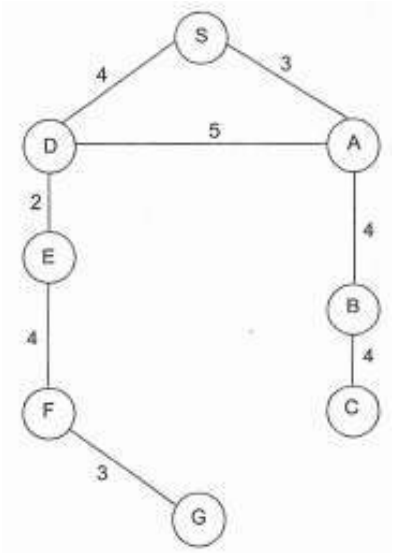
قاعده انتخاب ایجاد گره بعدی ، هیچ برتری به یک گره نمی دهد که شانس خوبی برای رسیدن از جستجو به گره جواب دارد .

این روش جستجوی نیروی حیوان صفت می تواند به وسیله ارائه رتبه ای به گره ها که

تا کنون کاوش نشده اند بهینه شود. لازم است هزینه محاسبات یک " گره پاسخ" از گره جاری را بدست آوریم.
این هزینه می تواند تعداد گره های لازم برای رسیدن به هدف یا مجموع تعداد سطوحی که تا رسیدن به نزدیکترین گره هدف نیاز است ، باشد .
توجه کنید که در عمل بکارگیری ضوابط بالا خیلی مشکل است چون نیروی مورد نیاز برای محاسبه هزینه گره ممکن است خیلی زیاد باشد که شامل هزینه جستجوی " زیر درخت " از گره جاری است.

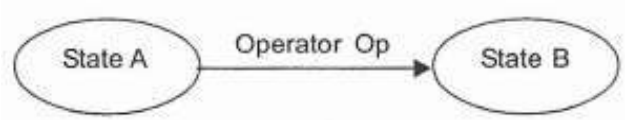
٦-٤ شماهای ارائه

در تجزیه به خوبی حل؛ ساختار درختی بطور معمول برای نمایش استراتژی کنترل در جستجو استفاده می شود.
در نمایش فضای حالت ، یک درخت ممکن است برای نمایش مجموعه ای از حالتهای مسئله استفاده شود که بوسیله کاربرد عملگرها تولید می شود.
در چنین نمایشی ، ریشه درخت حالت مسئله اولیه یا موقعیت را ارائه می دهد . هر یک از حالات جدید میتوانند بوسیله اعمال یک عملگربه ریشه که به عنوان گره جانشین گره ریشه ارائه می شوند بدست بیایند.
کاربرد عملگر برای این گره ها جانشین بعدی را تولید می کنند.
به هر حال در عمل ؛ حالت ها بوسیله یک گراف نمایش داده می شوند تا یک درخت زمانی که ممکن است مسیرهای مختلفی از ریشه به گره داده شده وجود داشته باشد.
در کنار درخت و گرافها شماهای نمایش دیگر شامل گرافهای AND/OR است که که در روش حل مسئله شامل کاهش مسئله استفاده می شوند.
آنها ابزاری را برای بدست آوردن "زیرهدف" ها فراهم می آورند که ترکیبی است که برای بدست آوردن هدف مطلوب کافی است.
در اینجا شکل ٦-١ یک نمایش ساده از یک درخت است که حالتهای مختلفی را با هم و با تغییر نمایش می دهد. همچنین نمایشی از تغییر ازحالتی به حالت دیگر با کاربرد یک عملگر در یک شاخه از درخت درشکل ٦-٢ داده شده است.



شکل ٦-١

یک مسئله جستجوی پایه ای در شکل ٦-١ نشان داده شده است. یک مسیر از گره شروع (S) به گره هدف (G) پیدا شده است .
شکل٦-٢ رویه های جستجو درختها را مانند اینها کاوش می کنند .



شکل ٦-٢

این پیش زمینه به ما اجازه می دهد که به رویه معمولی حل مسئله در هوش مصنوعی نظری بیندازیم .

٦-٥ حل مسئله در هوش مصنوعی

رویه اولیه برای حل یک مسئله در AI می تواند به صورت زیر نوشته شود :

رویه تولید

١- داده ⟵ پایگاه داده اولیه

۲- بعضی قوانینیR را از مجمو عه ای از قوانین که می توانند به داده اعمال شوند انتخاب کن

۳- داده ⟵ نتیجه اعمال قاعده به داده ها

پایان پردازش

در رویه بالا مسئله کنترل بنیادی انتخاب قوانین R برای کاربرد در پایگاه داده است یک مشخصه مهم در این ملاحظه میزان اطلاعات و آگاهی در دست که این تخمین ها به کار می برند .

چنین رویه جستجویی اطلاعاتش را مانند رویه جستجوی " کشف کننده/مطلع" بکار می برد بر خلاف یک رویه جستجوی " کور/نابینا "جایی که انتخابها دلخواهانه انجام می شود و مؤثر بودن محاسبات یک جستجو بستگی به طیف (بی اطلاعی / مطلع) در استراتژی کنترل دارد .

این موثر بودن با دونوع هزینه اندازه گیری می شود: هزینه کاربرد قوانین و هزینه کنترل استراتژی کنترل کاملا" بی اطلاع فقط یک محاسبات کنترلی کوچک به وجود می آورد. به هر حال نتیجه چنین استراتژی در هزینه ،درخواست قوانین بالا است و تعداد زیادی از درخواست قوانین برای پیدا کردن راه حل برای آگاهی دادن یک استراتژی کنترل درباره قلمرو مسئله را لازم دارد .

٦-٦ تکنیک های جستجوی کورکورانه

اگر چه تکنیک های جستجوی نا آگاهانه برای مسائل کلی عملی نیستند. به هر حال یک پایه ای را برای درک استراتژی جستجوی آگاهانه فراهم می آورد.

همه الگوریتم های جستجو فرضهای زیر را به وجود می آورند:

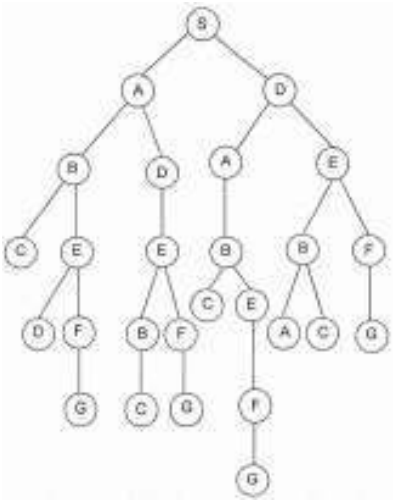
یک گره می تواند بوسیله بعضی رویه های شناخته شده مانند رویه گسترش ، توسعه یابد.

گراف فضای حالت یک درخت است که دلالت دارد به اینکه فقط یک حالت شروع و یک راه منحصر به فرد از یک گره به دیگری وجود دارد.

هر وقت یک گره گسترش داده می شود تا فرزندانش را تولید کند یک اشاره گر به عقب از فرزندان به والد ایجاد می شود .

بنابراین وقتی یک گره هدف بدست می آید ویژگیها دنبال کردن مسیر راه حل را ممکن می کنند . برای تمام روشهای جستچو درخت داده شده در شکل ٣-٦ حالت شروع

مسئله را نشان می دهد.



شکل ٦-٣

٦-٦-١ جستجوی سطحی

در جستجوی سطحی ، ما از یک گره اولیه S شروع می کنیم و آن را به عنوان گره ملاقات شده علامت گذاری می کنیم .

سپس S برای بدست آوردن همه گره های دیگر به ترتیب نزدیکی به S ، گسترش داده می شود . این بوسیله تعداد لبه های بین آنها اندازه گیری می شود. ما پیمان نامه ای که در شیوه " چپ به راست " آزموده شده است را استفاده می کنیم. یعنی تمام عملگرهای ممکن سطح n قبل از هر عملگر دیگری در سطح n+1 در نظر گرفته می شود .

اگر چه این رویه ممکن است خیلی طولانی باشد با این حال یک راه بهینه برای هدف پیدا می شود اگر چنین راهی وجود داشته باشد.

جستجوی سطحی بوسیله الگوریتم زیر شرح داده می شود:

١- از یک صف یک عاملی شامل گره ریشه .

٢- تا زمانی که صف خالی است یا هدف در دسترس است تصمیم می گیریم که اولین عامل در صف گره هدف است .

a) اگر عنصر اول گره هدف است سپس عملی انجام ندهید.

b) اگر عنصر اول گره هدف نیست سپس عنصر اول را از صف حذف کنید و

فرزندان عنصر اول را اضافه کنید.

۳- اگر هدف یافت شد اعلام موفقیت است در غیر اینصورت اعلام شکست است.

در جستجوی سطحی پردازش سطح به سطح رو به پایین انجام می شود تااینکه هدف به دست آید

ما می توانیم مشاهده زیر را در باره جستجو های سطحی ایجاد نماییم.

و آن یک استراتژی جستجوی سیستماتیک است که همه گره ها در سطح یک و سپس همه حالتها در سطح ۲ و غیره در نظر می گیرد.اگر راه حلی وجود داشته باشد این جستجو تضمین می کند که آنرا بیابد .و اگر چندین راه حل وجود داشته باشد جستجوی سطحی ابتدا کم عمق ترین وضعیت هدف را پیدا می کند .

٦-٧ رویه ارتباطات و انشعاب با برنامه ریزی پویا

۱-از صف مسیرهای ناقص_صف اولیه شامل طول مسیر صفر و قدم صفر از گره ریشه به هیچ مکانی است.

۲-تا زمانی که صف خالی است یا به هدف دسترسی پیدا کرده است.اگر اولین مسیر در صف به گره ی هدف دسترسی پیدا کرد ، مشخص کنید:

a) اگر اولین مسیر به گره ی هدف دسترسی پیدا کرد کاری انجام ندهید

b) اگر اولین مسیر به گره هدف دسترسی پیدا نکرد:

I) برداشت مسیر اول از صف

ii) در مسیر جدید توسعه اولین قدم بوسیله ی مسیر برداشته شده.

iii) اضافه کردن مسیرهای جدید به صف

iv) مرتب سازی صف بوسیله ی جمع بندی هزینه در مقابل با هزینه ی کمتر مسیرها

v) اگر ۲ مسیر یا مسیرهای بیشتری به یک گره دست یاقت تمام مسیرهای دیگر را حذف کنید به جز ان مسیری که به آن گره با کمترین هزینه دسترسی دارد.

۳)اگر هدف پیدا شد اعلام موفقیت است وگر نه اعلام شکست است.

پایان ارتباط و انشعاب با برنامه ریزی پویا

جستجوی ارتباط و انشعاب بوسیله ی برنامه ریزی پویا مشخص می کند که مسیر S_D_E_F_G بهینه است. ارقام در کنار گره ها فاصله ها را جمع کرده اند. گره های حذف شده بی فایده مشخص شده اند. گره های کمتر با عملیات جستجوی ارتباط و انشعاب بدون برنامه ریزی پویا گسترش پیدا کرده اند.

رویه A :

رویه A یک جستجوی ارتباط و انشعاب است که با براوردی از فاصله باقی مانده با روند برنامه ریزی پویا ترکیب شده است. اگر تخمین فاصله ی باقی مانده ی کمتر از فاصله واقعی باشد. سپس رویهA راه حل بهینه را ارائه می کند. بنابر این رویه جستجوی A این است:

روش A:

۱- از یک صف مسیرهای ناقص_ صف اولیه شامل طول مسیر صفر، گام صفر از گره ی ریشه به هیچ مکانی است.

۲- تا زمانی که صف خالی است یا به هدف دسترسی پیدا کرده است اولین مسیر در صف را که به گره هدف دسترسی پیدا کرده را مشخص کنید.

a) اگر اولین مسیر به گره هدف دسترسی پیدا کرد کاری انجام ندهید.

b) اگر اولین مسیر به گره ی هدف دسترسی پیدا نکرد:

I) برداشتن اولین مسیر از صف

II) در مسیر جدید توسعه اولین قدم بوسیله ی مسیر برداشته شده

III) اضافه کردن مسیر جدید به صف

IV) مرتب سازی صف بوسیله ی جمع بندی هزینه در مقابل با مسیرهایی با کمترین هزینه

V) اگر ۲ یا تعداد بیشتری به یک گره دست یابند تمام مسیرهای دیگر را حذف کنید به جز آن مسیری که به آن گره با کمترین هزینه دسترسی دارد.

۳-اگر هدف پیدا شد اعلام موفقیت است وگر نه اعلام شکست است .

پایان A

٦-٨ جستجوی بازی

بازی هایی مانند شطرنج ، چکرز، تیک تاک نو و غیره نوع دیگری جستجو نیاز دارند. گره دردرخت بازی نشان دهنده ی ترکیبات صفحه ی شطرنج و شاخه ها نشان دهندهٔ تغییر شکل ازترکیب یک صفحه به صفحه ی دیگر است. تصمیمات بوسیله ی ۲ نفر یا حریفان گرفته می شوند که هر کدام یک تصمیم در هنگام نوبتشان می گیرند. توجه

کنید در بازی هایی مانند شطرنج یک حرکت به معنی حرکت منفرد یک بازیکن و جواب حریفش می باشد. در این جا بطور غیر رسمی حرکت به عمل منفرد یک فرد بر می گردد.

بازی ها نیازمند رویه های جستجوی مختلفی هستند نسبت به چیزی که ما با آن قبلا مواجه شدیم. روشهای جستجوی اجباری حیوانی قطعا خارج شده اند. این به خاطر ان است که به عنوان مثال در شطرنج اگر ما فاکتور انشعاب موثر را ١٦ و عمق ١٠٠ در نظر بگیریم سپس تعداد انشعابها احتمال خسته کننده ای بطور تقریبی ١٠٠ خواهد بود که بطور مسخره آمیزی یک عدد بزرگ است.

از طرف دیگر اگر مقداری را بعنوان ازمایشی برای انتخاب حرکت بعدی مان فرض می کنیم سرانجام ما نتایج ضعیفی را خواهیم گرفت. بنابر این این قبیل تکنیک ها نیازمند یک آنالیزور موقعیت است که می تواند بعد از آنکه بازی از طریق حرکات مراحل مختلف توسعه پیدا کرد استفاده شود. باید به حد کافی در دقت شود که در مجموعه فوران رخ ندهد. به این دلیل پس از یک عمق منطقی ، موقعیت ها می توانند بر طبق یک حرکت رو به جلو نگهداری شوند. این بر اساس یک تصوری است که حرکات موقعیت های قطعی را به عنوان پیشرفت های بازی مشخص می کنند. باید مورد توجه قرار گیرد که این گمان یک فرض قابل بحث است.

٦-٨-١ روش مینی مکس

فرض کنید ما یک آنالیزور داریم که تمام قضاوتها در مورد موقعیتهای صفحه شطرنج را به یک عدد کیفیتی تغییر می دهد. هم چنین فرض کنید که اعداد مثبت نشان دهنده حمایت از یک فرد و اعداد منفی برای حریف او هستند درجه حمایت به ارزش واقعی عدد بستگی دارد .

پروسه تعیین کیفیت عدد " ارزیابی آماری" نامیده می شود .در پایان تعداد حرکا ت است که امتیازات ارزیابی آماری را پیدا کند که بوسیله آنالیزورهای موقعیت که ارزیابی کنندگان آماری نامیده می شوند فراهم شده است.

امید بازیکن برای اعداد مثبت بازیکن بیشتر و حریف او بازیکن کمتر نامیده می شود. در سطح میانی در درخت ،ارزشها در وضعیت پایانی داده شده اند .ارزش یک وضعیت غیر پایانی بوسیله برگشت دوباره از مراحل پایانی محاسبه شده است . این روش که بوسیله اطلاعات امتیاز دهندگی شما را از درخت بازی می گذراند رویه مینی مکس

نامیده می شود از وقتی که امتیاز در هر گره حداقل یا حداکثر از امتیازات آنی پایین تر است.

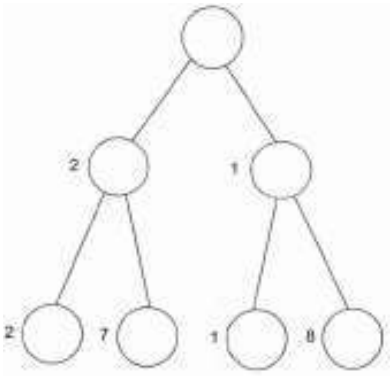
رویه مینی مکس

۱- مشخص کنید اگر محدودیت جستجو مورد دسترسی قرار گرفته است یا اگر یک سطح حداقل یا یک سطح حداکثر است.

a) اگر محدودیت جستجو در دسترس باشد ارزش آماری موقعیت جاری وابسته به بازیکن خاص محاسبه می شود . نتیجه را گزارش کنید .

b) اگر مرحله سطح حداقل است مینی مکس را روی فرزندان در موقعیت جاری استفاده کنید. حداقل نتایج را گزارش کنید.

c) در غیر این صورت اگر مرحله سطح حداکثر است مینی مکس را روی فرزندان موقعیت جاری استفاده کنید . حداکثر نتایج را گزارش بدهید.



تصویر ٦-١١ : روش مینی مکس از جریان کسب کردن هدف عبور می کند.

رویه مینی مکس از یک ارزیابی کننده آماری را برای محاسبه تعداد فواید موقعیت بازی در انتهای یک درخت بازی به طور جزئی توسعه یافته استفاده می کند .. اگر رویه مینی مکس مورد استفاده واقع شود ارزیاب کننده آماری باید در هر وضعیت که در انتهای هر درخت پیدا می شود مورد استفاده قرار گیرد خوشبختانه روش هایی وجود دارد که اعداد ارزیابی ها می تواند با کاهش تعداد انشعابها در درخت کاهش پیدا کند.

٦-٨-٢ روش آلفا – بتا

روش آلفا – بتا این طور رفتار می کند : اگر حریف یک جواب دارد که پتانسیل حرکت

بد است نیازی برای بررسی جواب های دیگر برای پتانسیل حرکت نیست.
بنابراین رویه آلفا – بتا به صورت زیر است:

۱- مشخص کنید سطح بالاترین سطح است یا اگربه محدودیت جستجو رسیده است یا اگر مرحله سطح پایین تری است یا اگر سطح بالاتری است.

a) اگر سطح بالاترین مرحله است اجازه دهید آلفا ∞- باشد و بتا ∞+ باشد.

b) اگر به محدودیت جستجو رسیده است مقدار آماری موقعیت جاری که وابسته به بازیکن خاص است را محاسبه کنید.نتیجه را گزارش دهید.

c) اگر مرحله یک سطح پایین است

i) تا وقتی که همه بچه ها مورد آزمایش با مینی مکس بوده اند آلفا بزرگتر از بتا است.

A: بتا را به کوچکتر از بتای معین شده تنظیم کنید و کوچکترین مقدار گزارش داده شده بوسیله مینی مکس را روی فرزندان بکار برید.

B : مینی مکس را بر روی فرزند بعدی آلفا و بتا استفاده کنید

ii) بتا را گزارش دهید.

d) اگر مرحله یک سطح بالاتر است :

i) تا زمانی که همه بچه ها با مین مکس یا آلفا مورد آزمایش قرار گرفته اند آنها بزرگتر از بتا هستند .

A : آلفا را با بزرگترین مقدار آلفای معین شده تنظیم کنید و بزرگترین مقدار گزارش شده بوسیله مینی مکس را روی بچه ها به کار برید.

B : مینی مکس را بر روی فرزند بعدی موقعیت جاری استفاده کنید آخرین تقاضای جدید مینی مکس از آخرین آلفا و بتا را نگهداری کنید .

ii) آلفا را گزارش دهید

برای بکار گرفتن یک جستجوی ما نیاز داریم که گره های پایه ، براساس مقدار

برگشتی از تابع کشف کننده مرتب شوند.
ما می توانیم چنین جستجو را برای الگوریتم جستجوی عمومی به کار ببریم.تابعی که بهترین گره را برای گسترش دادن انتخاب می کند تابع BEST-FIRST-SEARCH می نامیم.
در حقیقت نمی توانیم مطمئن باشیم که بهترین گره را اول گسترش دادیم به عبارت دیگر این جستجو در همه حالات ما را از حالت جاری به حالت هدف نمی رساند.
اما به ما این دانش را میدهد که باور کنیم آن بهترین گره برای گسترش است و این تابع
،

BEST-FIRST-SEARCH(problem, EVAL-FN) یک رشته از راه حلها را بر می گرداند.

ورودی: Problem , a problem
EVAL-FN یک تابع ارزیابی
Queueing-Fn یک تابع که گره ها را بوسیله EVAL-FN مرتب می کند

٦-٨-٣ جستجوی حریصانه

جستجوی حریصانه پیاده سازی فلسفه جستجوی بهترین است . آن روی یک اصل علمی کار می کند که بزرگترین بیت از مسئله گرفته می شود.
جستجوی حریصانه سعی می کند تا هزینه تخمین زده شده برای رسیدن به هدف را حداقل کند.
برای انجام این کار گره هایی را که فکر می کند به حالت هدف نزدیکترند گسترش می دهد.برای انجام این کار تابع کشف کنندگی را بکار می برد .
با بدست آوردن یک تابع کشف کننده h ؛ ما می توانیم یک جستجوی حریصانه را به کار بگیریم مانند زیر:
تابع GREEDY-SEARCH(problem) یک راه حل شکست را برمی گرداند.
Return BEST-SEARCH(problem,h)

٦-٨-٤ جستجوی تپه نوردی

بررسی های لازم شاید راه های مرتب سازی به صورت تناوبی یا انتخابی بهبود ببخشد جستجوی تپه نوردی یک تکنیکی است که شامل مرتب سازی به صورت جستجوی عمقی است.

یک طرح مرتب سازی ، بر اساس مسافتهای باقیمانده مثل خط مستقیم یا مسافت پروازی است.

شکل ٦-٦ رویه جستجوی تپه نوردی را نشان می دهد

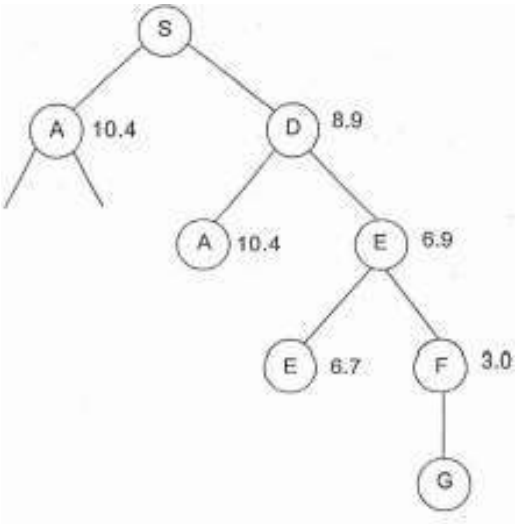
رویه جستجوی تپه نوردی:

۱- از یکی از عناصر صف شامل گره ریشه

۲- تا زمانی که صف هست خالی یا هدف بدست می آید تعیین کن که آیا اولین عنصردر صف گره هدف است : a) اگر اولین عنصر گره هدف است کاری انجام نده

b) اگر اولین عنصر؛ گره هدف نیست ، اولین گره را حذف کن. فرزندان آن گره را اگر وجود دازد مرتب کن ، مسافت باقی مانده را تخمین بزن.و آنها را در ابتدای صف اضافه کن .

۳- اگر هدف پیدا شد موفقیت را نشان بده در غیر این صورت شکست را نشان بده.



شکل ٦-٦

جستجوی تپه نوردی ، یک جستجوی عمقی با یک اندازه گیری اکتشافی است که

گره هایی را که گسترش می دهیم مرتب می کند. شماره کنار گره ها ، مسافت خط مستقیمی است که گره با گره هدف دارد.

۶-۸-۵ جستجوی A*

در بخش قبلی ما جستجوی حریصانه را در نظر گرفتیم . این روش جستجو هزینه رسیدن به هدف را با استفاده ازتابع کشف کننده کاهش می دهد.

جسجوی حریصانه میتواند زمان جستجو را کاهش دهد اما نه کامل است نه بهینه.

در مقایسه جستجو با هزینه یکسان هزینه مسیر را نیز حداقل می کند .

جستجوی با هزینه یکسان هم بهینه هست هم کامل اما می تواند بسیار بی فایده باشد.

اگر ما بتوانیم دو استراتژی را برای دست یافتن به مزایای هر دو جستجوترکیب کنیم ، بهترین کار را انجام می دهیم . خوشبختانه می توانیم با ترکیب دو تابع ارزیابی به این امر دست یابیم .

$$F(n) = g(n) + h(n)$$

$g(n)$ هزینه مسیر از گره شروع به گره n را می دهد و $h(n)$ هزینه تخمینی از ارزانترین مسیر از n به هدف است . ما داریم :

$F(n)$ =هزینه تخمین زده شده از ارزانترین راه حل از طریق n

چیز خوب درباره این استراتژی این است که با وجود قرار دادن محدودیتی روی تابع h بهینه , کامل است .

ما می توانیم جستجوی A* را به صورت زیر به کار ببریم:

تابع A*-SEARCH (problem) راه حل شکست را بر می گرداند.

Return BEST-FIRST-SEARCH(problem , g + h)

توجه کنید که اگر شما از این الگوریتم به صورت دستی استفاده می کنید همیشه کم هزینه ترین گره را روی ریشه هر جاییکه گره در درخت جستجواست گسترش دهید.

به این منظور گره ای که شما برای گسترش بعدی انتخاب می کنید فقط به گره

هایی که تولید شده اند محدود نشده است. البته این در الگوریتم به عنوان تابع صف به صورت اتوماتیک به ترتیب گره ها ساخته شده است .

٦-٨-٦ کشف کننده قابل قبول

محدودیتهایی را که در بالا برای تابع h نام بردیم این است که تابع h نباید هرگز هزینه ای بیش از تخمین برای رسیدن به هدف داشته باشد.

چنین تابع h ای را تابع کشف کننده قابل قبول می نامیم. راه دیگر توصیف تابع قابل قبول این هست که بگوییم آنها خوشبینانه هستند.وقتی که آنها همیشه فکر می کنند که هزینه رسیدن به هدف کمتر از مقدار واقعی آن است.

آن واضح است که تابع کشف کننده SLD قابل قبول است بطوریکه ما هرگز نتوانیم یک مسیر کوتاه تر بین هر دوشهر پیدا کنیم .

A* هم بهینه هست هم کامل اما این خبر خوبی نیست. می تواند نشان داده شود که تعدادی از گره ها که بدست آمده اند هنوز به صورت تابع نمایی از طول فضای جستجو برای بیشتر مسائل هستند .

این حالت دلایلی برای زمان جستجودارد اما معمولا" برای فضای مورد نیاز خیلی جدی تر می شود.

٦-٨-٧ معمای ٨

معمای ٨ شامل هشت مربع و یک فضای خالی است. هدف چیدن هر مربع در یک زمان تا رسیدن به حالت هدف است .

مسئله در شکل ٦-٧ نشان داده شده است

5	4		1	2	3
6	1	8	8		4
7	3	2	7	6	5
Initial State			Goal State		

شکل٦-٧

این مسئله فقط مرحله مستقیم است که آن را برای مطالعه سخت می سازد در عین حال آنچنان سخت نیست که ما توی باتلاق گیر کنیم

یک راه حل معمول آن حدود ۲۰ مرحله است . فاکتور انشعاب آن ۳ است (٤- وقتی

که خانه خالی در مرکز قرار دارد . ۲- وقتی که خانه خالی در گوشه ها باشدو ۳- برای حالتهای دیگر) این به این معنی است که دریک جستجوی خسته کننده در حدود ۳^۲۰ حالت وجود دارد. به هر حال بوسیله نگهداری اثر حالتهای تکراری می توانیم آنرا کاهش دهیم چون فقط ۳,٦۲,۸۸۰=!۹ ترتیب متفاوت از ۹ مربع وجود دارد .

اما حتی ۳,٦۲,۸۸۰ تعداد زیادی برای جستجو است بنابراین کار بعدی یافتن یک تابع کشف کننده خوب است.

اگر ما بخواهیم یک راه حل بهینه پیدا کنیم به یک تابع کشف کننده احتیاج داریم که هرگز تخمین بالایی از تعداد مراحل در رسیدن به هدف را ارائه ندهد.. به یک کشف کننده قابل قبول احتیاج داریم.
اینجا دو امکان وجود دارد :
H1 که تعداد خانه هایی است که در مکانهای نادرست هستند.در شکل بالا در قسمت حالت اولیه از هشت سفال ؛ هفت تای آن در خارج از مکان واقعی هستند.
این کشف کنند ه به صورت واضح قابل قبول است که هر یک از خانه هایی که خارج از مکان خود هستند نیاز دارند که حداقل یک بار حرکت داده شوند تا به محل صحیح برسند.
H2 : مجموع فواصل خانه ها از مکان هدفشان . روشی که ما استفاده می کنیم برای محاسبه مقدار فواصل خانه ها از موقعیت هدفشان جمع کردن تعداد افقی و عمودی موقعیت هااست.
این تابع کشف کننده همچنین قابل قبول است چون هر حرکتی فقط می تواند یک خانه را یک مرحله به هدف نزدیکتر کند .
فاصله مان هاتان برای حالت شروع در بالا هست:

$$H2 = 2 + 3 + 3 + 2 + 4 + 2 + 0 + 2 = 18$$

برای خانه های ۱ تا ۸ .
چگونه ما تصمیم می گیریم که کدام یک برای استفاده بهترین است؟

یک روش این است که مثالهای زیادی از مسئله را ایجاد کنیم و از تابع کشف کننده استفاده کنیم و ببینیم که کدام یک به ما بهترین راه حل را می دهد

جدول ٦,١ نتیجه ای از ١٠٠ مسئله اجرا شده در عمق (طول راه حل) ٢و٤و٦و...و ٢٤ که از A* با تابع ها ی کشف کننده h1,h2 را نشان می دهد.
هزینه جستجو عددی را نشان می دهد که نشان دهنده میانگین تعداد گره هایی است که گسترش داده شده اند.

Table 6.1

Search cost				EBF		
Depth	IDS	A*(h1)		A*(h2) A*(h1)	IDS A*(h2)	
٢	١٠	٦	٦	٢,٤٥	١,٧٩	1.79
٤	١١٢	١٣	١٢	٢,٨٧	١,٤٨	١,٤٥
٦	٦٨٠	٢٠	١٨	٢,٧٣	١,٣٤	١,٣٠
٨	٦٣٨٤	٣٩	٢٥	٢,٨٠	١,٣٣	١,٢٤
١٠	٤٧١٢٧	٩٣	٣٩	٢,٧٩	١,٣٨	1.22
		٢٢٧	٧٣	٢,٧٨	١,٤٢	١,٢٤

١٢ ٣٦٤٤٠٤

١٤ ٣٤٧٣٩٤١ ٥٣٩ ١١٣ ٢,٨٣ ١,٤٤ ١,٢٣

٢١١ ١,٤٥ 1.25

١٦ ١٣٠١

از این نتایج واضح است که h2 کشف کننده بهتری است که با بسط دادن تعداد کمتری گره به نتیجه می رسد.اما چرا این گونه است؟

یک دلیل واضح این است که تعداد گره ها یی که گسترش داده می شوند فاکتور انشعاب هستند . اگر فاکتور انشعاب بالا است پس گره های بیشتری گسترش داده می شوند. بنابراین یک راه برای اندازه گیری کیفیت تابع کشف کننده ؛ درک فاکتور انشعاب میانگین است .

اگر ما از A* برای مسئله استفاده کنیم و عمق راه حل d باشد سپس b* فاکتور انشعابی است که یک درخت یکنواخت باعمق d خواهد داشت تا گره های N را نگه دارد. بنابراین:

$$N = 1+ b^* + (b^*)^2 + \ldots + (b^*)^d$$

یک مثال برای محسوس کردن مطلب ارائه می دهیم. اگر A* یک راه حل در عمق ۵ پیدا کند. از ۵۲ گره استفاده می کند سپس فاکتور انشعاب موثر ۱,۹۱ است :

$$52 = 1 + 1.91 + (1.91)^2 + (1.91)^3 + (1.91)^4 +(1.91)^5$$

فاکتور انشعاب مؤثردر جدول برای استراتژی های مختلف جستجو نشان داده شده است.

ما می توانیم بفهمیم ازاین که A* که h2 را بکار می برد فاکتور انشعاب مؤثر پایین تری دارد بنابراین h2 یک کشف کننده بهتر از h1 است.

ما می توا نیم بپرسیم آیا h2همیشه بهتر از h1 است؟در حقیقت آن اسان است که ببینیم برای هر گره n ؛ $h2(n) > h1(n)$.

این حالت را " سلطه " می نامیم و می گوییم که h2 بر h1 غلبه می کند.تسلط مستقیما

به مؤثر بودن ترجمه می شود .

آن چیزی است که می توانیم بگوییم h2 بر h1 سلطه دارد اما چطور می توانیم h2 را در اولین مکان مطرح کنیم.

هردو تابع h1,h2 تعداد حرکات را برای حل نهایی تخمین می زنند. ممکن است که ما احتیاج داشته باشیم که حرکات بیشتری را تخمین بزنیم. اما اگر ما قوانین بازی را به روشنی تغییر دهیم سپس ما می توانیم این کشف کننده ها را بسازیم که یک طول مسیر دقیق که برای حل مسئله نیاز است را نشان می دهند.

به طور مثال ؛ اگر قوانین چنان تغییر کنند که ما بتوانیم هر خانه را به سوی مربع خالی حرکت دهیم سپس h1 می تواند مقدار دقیق طول مسیر مورد نیاز برای حل مسئله را نشان دهد.

به طور مشابه اگر یک خانه بتواند در هر مسیری حتی به سمت یک مربع اشغال شده حرکت کند ؛ سپس h2 یک طول مسیر دقیق از راه حل را می دهد.این تغییرات در قوانین ؛ قوانین و مسئله را که محدودیت عملگر دارند راحت تر می کند که یک مسئله راحت شده نامیده می شود.این معمولا در موردی است که هزینه راه حل دقیق یک مسئله راحت ، کشف کننده خوبی برای مسئله اصلی است .

٦-۹ ویژگی برنامه ریزی

این بخش شما را مرحله به مرحله در ایجاد برنامه های لیسپ برای حل مسائل جستجوی عمقی راهنمایی می کند و سپس مسائل حریفان و ادم خواران را با استفاده از prolog نمایش میدهد. نوشتن یک برنامه برای جستجوی عمقی آسان است. دو رویه بازگشتی یک درخت را به صورت موثر می شکافند. اگرچه که اصلاح این کدها و وفق دادن آنها برای جستجوی سطحی و دیگر جستجوها آسان نیست.

به ما اجازه دهید که روی یک جسجوی صف گرا کار کنیم. صف ما شامل مسیر های ناقص است. استفاده از صفهایی با مسیر های ناقص در ابتدا سخت است. اما یکبار که ما جستجوی عمقی را انجام دهیم ، تغییر دادن این کدها برای انجام جستجوهای دیگر آسان خواهد شد .

جستجو به سادگی اولین بحث خودش را به یک صف تک عاملی برای سوداوری جستجو تبدیل می کند .. سپس جستجو صف را امتحان می کند و اولین مسیر در صف

را برای موفقیت تست می کند. اگر آخرین گره در اولین مسیر ، گره پایانی نیست جستجو مسیر را توسعه می دهد و صف را اصلاح می کند و صف را به کپی دیگری از جستجو بر می گرداند .

١٤٢- جستجوی اولیه (شروع پایان)

(جستجوی اول (فهرست شروع) پایان) مقدار دهی :

(شرط (صف پوچ) NIL): .برگشت NIL اگر صف خالی است.

(پایان برابر (صف ماشین) (T) : برگشت T اگر هدف پیدا شده است.

جستجوی اولT :

< ادغام خاص (توسعه (صف ماشین) و صف > پایان .

در اینجا گسترش ، فرزندان یک گره را بر می گرداند که گره به عنوان یک بحث تعیین می شود .قبل از اینکه گسترش را بنویسیم اجازه دهید نگاهی به نمایش داده در برنامه بیندازیم.اگر ما فقط با درختان مواجه شویم فهرستهای خانگی باید به خوبی انجام شوند . اگر همچنین ما بخواهیم تمام شبکه ها را کنترل کنیم بهتر است که از نشانه ها و خصایص استفاده کنیم .گره ها و فرزندانشان می توانند بوسیله نشانه ها ارائه شوند و کمانها می توانند به وسیله ی خصایص نشان داده شوند.به عنوان مثال :

```
SETF ( GET ' S CHILDREN )        ( LO)
```

حقیقت را درک کنید که S یک والد است و بچه هایش L و O هستند.

```
SETF (GET 'L CHILDREN)        (M F)
```

حقیقت این است که L یک والد است و بچه هایش M و F هستند.

با تکرار انواع حالات ما می توانیم کل یک درخت را شرح دهیم. به عنوان مثال یک درخت نمونه می تواند به صورت زیر باشد.

```
SETF (GET 'S  CHILDREN )     (LO)
SETF (GET 'L  CHILDREN )     ( MF )
SETF (GET 'M CHILDREN )      ( N )
SETF (GET 'N  CHILDREN )     ( F )
SETF  (GET 'O  CHILDREN )    ( PQ )
SETF  (GET 'P   CHILDREN )   ( F )
SETF  ( GET 'Q  CHILDREN )   ( F )
```

قبلا مشخص شد که چگونه گره ها به هم متصل شده اند ما اکنون برای نوشتن کد برای گسترش آماده هستیم.

DEFUND EXPAND (NODE)(GET NODE CHILDREN)

روش تلفیق فرزندان جدید در صف قدیمی بستگی به استراتژی جستجو دارد. برای جستجوی عمقی فرم مخصوص این است:

( )(EXPAND(CAR QUEUE)(CDR QUEUE)افزودن)

بنابراین یک جستجوی عمقی بی تجربه به این صورت است:

DEPTH (DEFOUND DEPTH (Start Finish): توجه تغییردر اسم

(DEPTH (List Start) Finish): مقدار دهی اولیه

(DEFOUND DEPTH (Queue Finish): DEPTH توجه تغییردر اسم

(cond (null Queue) N/L). برگشت NULL اگر صف خالی است

(Equal Finish(Car Queuet) برگشت T اگر هدف پیدا شده است

(T (depth , t,y) تلاش دوباره با صف جدید

(append (expand (car Queue) گره جدید در سر

(CDR Queue)); بقیه صف

FINISH))))

این برنامه کار کوچکی را انجام می دهد. به سادگی t یا null را بر می گرداند و خوب است اگر ما بدانیم که گره ها در طی مسیر ما را به سوی هدف سوق می دهند. هم چنین این برنامه نمی تواند شبکه ها را کنترل کند وهیچ آزمونی برای جلوگیری از رفتن آنها به سوی حلقه های بی نهایت وحود ندارد.

چگونه به برنامه ای برسیم که مسیر را برگرداند ؟ ما باید اطلاعات بیشتری از عوامل ساختمان داده ای صف جمع آوری کنیم . تا به حال نمایش عواملی که گره ها نگه می دارند تست شده است. بنابراین صف می تواند شبیه این باشد:

(s)
(L O)
(M F O)
(N F O)
(F F O)

در عوض ما می خواهیم عوامل نه تنها گره ها بلکه مسیرها را نشان دهند. هر مسیر با

گره آغازی شروع می شود و به گره ای که فرزندانش هنوز زیاد نشده اند توسعه می یابد. سپس صف مانند زیر توسعه پیدا می کند

(((S)))
((LS)(OS))
((MLS) (F L S) (O S))
((N M L S) (F L S) (OS))
((F N M L S) (F L S) (O S))

به خاطر شکل جدید در صف لازم هست که ما عمق را عوض کنیم . اول پایان) (CAAR QUEUE با (CAR QUEUE) مقایسه شده است . سپس به جای برگشت T مسیری که T در آن قرار گرفته شده بر گردانده می شود . در آخر یک سازگاری کوچک برای نمایش نتیجه به صورت بر عکس که یک ترتیب طبیعی است ساخته می شود که از منبع تا هدف نام گذاری شده است.

```
(DEFUN DEPTH( START FINISH);                  توجه تغییر در اسم DEPTH است
DEPTH LIST (LIST START ) FINISH) ;             توجه تغییر محسوس
(DEFUND DEPTH ( QUEUE FINISH);                توجه تغییر در اسم DEPTH
(COUD ((NULL QUEUE) N/L)                      برگشت null اگر صف خالی است
((EQUAL FINISH ((CAAR QUEUE))                 توجه تغییر کوچک در CAAR
REVERSE (CAR QUEUE)))                         توجه تغییر کوچک در CAAR
(T ( DEPTH1))                                 تلاش دوباره با صف جدید
(APPEND (EXPAND (CAR QUEUE)                   گره جدید در قسمت سر
(CDR QUEUE))                                  بقیه صف
  (FINISH))))
```

همچنین ما باید گسترش را تغییر بدهیم .علاوه بر گرفتن یک گره و برگرداندن یک لیست از فرزندانش ، آن باید یک مسیر را بگیرد ، در پایان مسیر فرزندان یک گره را پیدا کنند و فهرستی از مسیرهای جدید را بر گرداند . هر مسیر جدید شامل مسیر اصلی با یکی از بچه هایش خواهد بود .این می تواند مانند زیر مرتب شود:

```
DEFUN EXPAND (PATH)
MAPCAR <> (LAMBDA(CHILD)(CONS CHILD PATH))
(GET (CAR PATH), CHILDREN)))
```

نقشه کار برای یک مسیر جدید که برای هر فرزند متعلق به انتهای مسیر قدیمی ساخته شده ، تنظیم شده است و هنوز اگر حلقه هایی وجود داشته باشد کار نمی کند. اگر ما آرزو داریم که شبکه ها را به خوبی درختان مدیریت کنیم ، باید مسیرهای پیشنهاد شده بوسیله عملیات گسترش را امتحان کنیم ، بررسی کنیم که ببینیم آیا گره جدیدی در جای دیگری ارائه شده است و آنرا استخراج کنیم .

```
DEFUN EXPAND (PATH);
REMOVE-IF
LAMBDA(PATH) (MEMBER(CAR PATH)CDR PATH)))
MAPCAR<> (LAMBDA(CHILD)(CONS CHILD PATH))
(GET (CAR PATH) CHILDREN)))
```

فصل هفتم

تکنولوژی هوش مصنوعی

اهداف

در پایان فصل، دانشجو با مفاهیم زیر آشنا می‌شود:

- آشنایی با سه فناوری مختلف هوش مصنوعی، شامل چشم کامپیوتری ، پردازش زبان طبیعی و شناسایی کلام یا گفتار
- آشنایی با الگوریتم شناسایی چهره و چگونگی محاسبه دقت در آن
- ارائه تعریف گرامر یک زبان و انواع آن
- آشنایی با چگونگی اشتقاق جملات از یک گرامر و چگونگی عملکرد تجزیه کننده ها
- آشنایی با گرامر و برنامه نویسی منطقی
- آشنایی با زبان نمایش دانش و سپس تعریف معنی شناسی جملات
- تعریف چگونگی پردازش سیگنال صوتی

۷-۱ مقدمه

امروزه فناوری های زیادی در زمینه هوش مصنوعی در حال مشهور شدن هستند.این فناوری ها به عنوان زمینه های مختلف تحقیقاتی مطرح شده اند .در این فصل ما راجع به سه فناوری مختلف هوش مصنوعی بحث خواهیم نمود.ابتدا با مبحث دید کامپیو تری (computer vision) که برای کاربردهای ارتشی و نظامی توسعه یافته است شروع خواهیم نمود.

امروزه ما توسعه های بی نظیری از computer vision را در زمینه هایی از چشم روباتها گرفته تا دید ماشینی (چشم ماشینی machine vision) شاهد هستیم.

سپس به توضیح پردازش زبان طبیعی خواهیم پرداخت .این فرایند ما را قادر به مکالمه با کامپیوتر به زبان طبیعی می سازد.این مبحث در بسیاری از پایگاههای داده ,سیستمهای خبره , روباتها و غیره مشهور است.همکار اصلی NLP یک بازشناس کلام یا گفتار است که در بخش نهایی این فصل به ان پرداخته خواهد شد.

۷-۲ بینایی کامپیوتری[1]

مشکل اصلی یک چشم کامپیوتری تشخیص و فهم یک شیئ یا یک منظره با خواص سه بعدی ان در یک تصویر یا یک توالی از تصاویر است.

دید یا vision پردازشی است که به وسیله ان شرح مناظر فیزیکی از تصویر انها استنتاج می شود.کاربردهای متنوعی از چشم کامپیوتری وجود دارند همانند: انالیز تصاویر پزشکی , مجتمع کردن (assembly) , کشتیرانی و ناوبری , واسط انسان و کامپیوتر و غیره.

یک کامپیوتر دید انسان را در ٤ مرحله تقلید می کند که به ترتیب عبارتند از ۱)اکتساب تصویر ۲) پردازش تصویر ۳) تجزیه و تحلیل تصویر ٤) فهم تصویر .

امروزه دستگاهها و ابزار خوبی برای گرفتن تصاویر واضح وجود دارد.

وقتی تصویری گرفته می شود , تایید کیفیت تصاویر به وسیله مکانیزم پردازش تصویر انجام می شود..در اینجا اثرات سیگنالهای نویز که می توانند به علت وجود خرابی در دستگاه تصویربرداری یا تنوع در نورپردازی و یا وجود اشتباه در فرایند دیجیتالی کردن باشند

---

1 computer vision

,نادیده گرفته شده است.از ان پس فاز تحلیل اغاز شده که با فاز فهم تصویر دنبال می شود و با شناسایی و تشخیص اشیا مختلف در تصویر سروکار دارد.

اجازه دهید با یک مساله تشخیص چهره با استفاده از یک eigenspace representation شروع کنیم.

فرض کنید یک مجموعه M تایی تصویر با سایز N x N پیکسل که هر تصویر شامل چهره یک فرد و تقریبا" تصویر فقط روی موقعیت چهره با روشنایی کافی است , داشته باشیم. و یک تصویر هم داریم که ان را با سری تصاویر مقایسه نموده و تصمیم می گیریم کدام یک از تصاویر ان مجموعه برابر با عکس مورد نظر است.

۱-۲-۷ Eigenspace Representation of images

یک تصویر N x N می تواند به عنوان یک نقطه در فضای تصویر N^2 بعدی بیان شود که در ان هر بعد با یکی از پیکسلها در تصویر مرتبط است و ارزش ممکن برای هر بعد با سطح خاکستری بودن هر پیکسل ارتباط دارد.به عنوان مثال یک تصویر 512x512 که در ان هر پیکسل یک عدد صحیح در بازه ۰ تا ۲۵۵ (یک پیکسل در یک بایت ذخیره می شود) است.فضای تصویر یک

144,262 بعدی است که هر بعد ۲۵٦ ارزش ممکن دارد .

اگر ما سری M تایی تصاویر را به عنوان M تا نقطه در فضای تصویر در نظر بگیریم یک راه شناسایی چهره یک فرد در یک تست جدید این است که نزدیکترین تصویر در ان سری در فضای تصویر را پیدا کنیم.ولی این را ه حل به دلیل اینکه اندازه فضای حالت بسیار بزرگ است بسیار کند خواهد بود . به این دلیل به جای این روش اجازه دهید هر تصویر را در یک فضا با ابعاد کمتر در نظر بگیریم , فضایی که فضای چهره (face space) یا eigen space نامیده می شود.

فرض کنید ما M' تا تصویر $E_1,E_2,....E_{M'}$ داریم که eigen space یا eigen

vectorsنامیده می شود.این تصاویر, تصاویر پایه مجموعه را تعریف می کند.بنابراین هر تصویر در دوره ای مشخص می کند که چقدربه هر کدام از تصاویر مجموعه پایه شباهت دارد.

ما می توانیم یک تصویر دلخواه I را به عنوان ترکیب وزن دار (خطی) از این eigenvector ها نمایش دهیم مانند زیر:

میانگین تصویر را از تمام تصاویرازمایشی $I_1,I_2.....I_M$ محاسبه کنید: (A)

$$A = \frac{1}{M}\sum_{i=1}^{M} I_i$$

برای K=1,2,.....M' ارزش حقیقی وزن W_K را که میزان شباهت بین تصویر ورودی I و k eigenvector ام را نشان می دهد (E_K) حساب کنید.

$$W_K = E_{K^*}^T \left(I - A\right)$$

جاییکه تصویر I مفروض است و نشان دهنده بردار ستون از طول N^2 است, K , E_K امین تصویر از eigen face می باشد و بردار ستون از طول N^2 می باشد , A هم بردار ستون از طول N^2 است.

* عمل حاصلضرب نقطه ای و عمل _ تفریق پیکسل به پیکسل می باشد.بنابراین W_K ارزش حقیقی قابل سنجش می باشد.

$W = [w_1, w_2.....w_{M'}]^T$ بردار ستونی از وزنهاست که سهم هر کدام از تصاویر eigen face در نمایش I را نشان می دهد.

بنابراین به جای نمایش تصویر I در فضای تصاویر ما ان را به عنوان W point در فضای M'

بعدی وزن نمایش می دهیم که ما انرا فضای چهره یا eigen spaceمی نامیم.

از اینرو هر تصویر از اصل eigenspace طراحی شده است.به عنوان متراکم سازی ,

هر تصویر به وسیله اعداد حقیقی M' نمایش داده می شود, به معنی اینکه برای هر ارزش نوعی مانند M=10 و ۳۲ بیت برای هر وزن , ما احتیاج به ۳۲۰ bits/image برای رمزگذاری ان در فضای چهره داریم.

بدیهی است که ما همچنین باید M ' تا تصویر eigen face را ذخیره کنیم که اینها هر کدام N^2 پیکسلی می باشند ولی این هزینه صرف تمام تصاویر ازمایشی (training) می شود.بنابراین می تواند به عنوان یک هزینه اضافی کوچک مطرح شود.

تصویر I حدودا" می تواند توسط W مانند زیر بازسازی شود:

$$I \approx A + \sum_{i=1}^{M'} w_i * E_i$$

این بازسازی و احیا صحیح و کامل می شود اگر $M' = \min(M, N^2)$.

بنابراین نمایش یک تصویر در eigen space در صورتی که عکس نوسازی و مطابق اصل بازسازی نشده باشد ,درست و دقیق نخواهد بود.ولی برای فرق قائل شدن بین تصاویر همین اندازه شباهت کافیست.

در این مرحله ارزشی برای M' انتخاب کنید و سپس بهترین تصاویر eigen vector را مشخص کنید.این عمل به وسیله تکنیک امارشناسی که مولفه اصلی تحلیل(Principal Components Analysis) نامیده می شود انجام می گردد.

ذاتا" این تکنیک M' تا تصویر را انتخاب می کند که حجم اطلاعات را در متراکم سازی بیشینه می کند.

بهترین M' تا تصاویر eigen face به صورت زیر حساب می شوند:

برای هر تصویر ازمایشی (training) I_i , به وسیله تفریق میانگین نرمال می شود:

$$Y_i = I_i - A$$

ماتریس کوواریانس N^2xN^2 را حساب کنید.

:

eigen vector های C را که با M' تا از بزرگترین eigen values مرتبط می باشند را بیابید.

eigen vector ها را $E_1,E_2....E_{M'}$ بنامید.اینها تصاویر eigen face ای هستند که به وسیله الگوریتم بالا استفاده شده اند.

به دلیل اینکه C بسیار بزرگ است , این روش از لحاظ محاسباتی بسیار سخت است.با این حال هستند روش های سریع مشابه برای پیدا کردن K تا از بزرگترین eigen vectorها.

$$C = \frac{1}{M}\sum_{i=1}^{M} Y_i Y_i^T$$

۷-۲-۲الگوریتم شناسایی چهره(FaceRecognition Algorithm)

کل الگوریتم شناسایی و تشخیص چهره در مراحل زیر می تواند خلاصه شود:

۱) با داشتن یک مجموعه از تصاویرازمایشی چهره ها, M' تا از بزرگترین eigen vectorها را محاسبه کنید:$E_1,E_2....E_{M'}$.

M'=10 یا 20 یک ارزش نوعی می باشد.توجه داشته باشید این مرحله تنها یکبار انجام می شود.(offline).

۲) برای هر فرد در مجموعه ازمایشی , اصل همبستگی را با شخص در ان eigen space محاسبه کنید.ربا ی این کار از فرمول بالا استفاده کنید.$W = [w_1, w_2....w_{M'}]$.

توجه داشته باشید که این مرحله نیز تنها یکبار انجام می شود.(offline)

۳) با داشتن تصویر ازمایشی I_{test}, ان را به وسیله محاسبه W_{test} توسط فرمول بالا به یک M' eigen space بعدی تبدیل کنید.

٤) نزدیکترین چهره از تصاویر ازمایشی به تصویر مورد نظر را بیابید:

$$d = \min_k \|W_{test} - W_K\|$$

هنگامیکه W_K نقطه ای در eigen space مرتبط با شخص k ام در مجموعه مورد ازمایش است و||x|| فاصله اقلیدسی در eigen space را مشخص می کند.

۵) فاصله تصویر مورد ازمایش را از eigen space بیابید.

$$dffs = \|Y - Yf\|$$

زمانیکه $Y = I_{test} - A$.

٦) اگر dffs<Threshold1

- تصویر ازمایشی به اندازه کافی به eigen space شبیه است. در مقایسه با کل تصاویر می توان مطمئن بود که این تصویر یک چهره است نه چیز دیگر.

سپس اگر d< Threshold2

- می توان I_{test} را با عنوان تصویری که شامل چهره شخص k ام است دسته بندی کرد , زمانیکه k نزدیکترین چهره در eigen space به W_{test} است.

در غیر این صورت

- تصویر I_{test} را به عنوان شخص نا شناخته دسته بندی کن.

0 0 0

در غیر اینصورت

- تصویر I_{test} را به عنوان تصویری که شامل چهره نیست دسته بندی کن.

مثال ۱-۷

فرض کنید دو تصویر ازمایشی ۳*۳ داریم که N=3,M=2 به صورت زیر تعریف شده است:

10 10 10
0 0 0
Image I_1

ما این دو تصویر را به عنوان دو آرایه (۳*۳=۹) در نظر می گیریم.

$I_1=[0,0,0,10,10,10,10,0,0,0]^T$
$I_2=[0,10,0,0,10,0,0,10,0]^T$

حال فرض کنید از یک شبه فضای (فضای فرعی) یک بعدی استفاده می کنیم.که $M=1$ و eigen vector به صورت زیر محاسبه شده است:

$E=[5,0,5,0,10,5,0,5,0]^T$

میانگین تصویر یا A توسط محاسبه هر پیکسل از I_1,I_2 بدست می اید.

میانگین سطح خاکستری بودی دو تصویر از پیکسلهای متناظر بدست می اید.بنابراین پیکسل دوم در A برابر است با $(0+10)/2=5$

از اینرو $A=\{0,5,0,5,10,5,0,5,0\}$.

اکنون ما می توانیم بفهمیم که چگونه I_1 به یک eigen space یک بعدی به وسیله محاسبه W_1 طراحی می شودهنگامیکه $w_{1,1}=E_1^T*(I_1-A)$.

بنابراین ما اینجا داریم:

0 10 0
0 10 0
0 10 0
Image I_2

$$I_{1'} = I_1 - A = [0,-5,0,5,05,0,-5,0]^T$$
$$w_{1,1} = 5*0 + -5*0 + 5*0 + 10*5 + ... + 5*0 = 0$$
$$W_1 = [0]$$

اکنون فرض کنید تصویر ازمایشی زیر راداریم:

0	7	3
0	10	10
0	10	0
Image I_{test}		

با تصویر کردن یا تجسم I_{test} به فضای چهره داریم: $W_{test} = [w_{test,1}]$
زمانیکه :

$$w_{test} = E_1^T * (I_{test} - A)$$
$$= [5,0,5,10,5,10,5,0,5]*[0,2,3,-5,0,5,0]^T$$
$$= 15$$

بنابراین W_{test} =(۱۵) که به این مفهوم است که W_{test} به I_1 بیشتر شبیه است تا به I_2 .

بنابراین ما I_{test} را جزء کلاس یا گروه I_1 طبقه بندی می کنیم.

۷-۲-۳ دقت شناسایی چهره:(Face Recognition Accuracy)

کارایی استفاده از eigen space ۲۰ بعدی به طبقه بندی صحیح ۹۵٪ ای از یک پایگاه شامل ۷۵۰۰ تصویر از ۳۰۰۰ نفر را نتیجه داد.

اگر مجموعه تصاویر ازمایشی شامل دو تصویر از هر شخص است پس برای هر شخص نقطه میانگین در eigen space را باید از نقاط محاسبه شده برای هر

تصویر از یک شخص محاسبه کنید.این متد نیازمند این است که همه تصاویر در پایگاه شامل چهره هایی از یک سایز (اندازه) و یک موقعیت و یک جهت باشند به طوریکه با استفاده از این تابع عمومی فاصله بتوان انها را در eigen space مقایسه کرد .
اگر از یک شیء سه بعدی چندین تصویر وجود دارد(به عنوان مثال :سر یک شخص از جهات مختلف) سپس نقاط موجود در eigen space که متناظر با دیدهای ۳ بعدی مختلف است می توانند به وسیله تطابق یک فراسطح (Hyper surface) به همه نقاط ترکیب شوند.

این hyper surface یا فراسطح می تواند به عنوان یک توصیف از شخص در eigen space ذخیره شود.در حال حاضر محصولات تجاری مختلفی وجود دارند که بر اساس این متد eigen face عمل می کنند.

۷-۳ پردازش زبان طبیعی[1]

پردازش زبان طبیعی یکی از بزرگترین کاربردهای هوش مصنوعی است.
این مبحث به تکنیک هوش مصنوعی برای ایجاد ارتباط با کامپیوتر با یک زبان طبیعی رجوع می کند.استفاده از یک واسط زبان طبیعی یک راه معقول برای پرداختن به طراحی نوعی خاص از واسط است.
مزیت ان این است که به هیچ مهارت خاصی غیر از داشتن توانایی ابتدایی در تایپ یک یا دو جمله به وسیله صفحه کلید نیاز ندارد.
از طرف دیگر عیب واسطهای زبان طبیعی این است که به اندازه کافی مختصر و موجز نیستند.
از لحاظ مفهومی ۲ نوع واسط NL (Natural language) وجود دارد.۱) آنهایی که خود را به یک زیر مجموعه از زبان انگلیسی محدود می کنند (یا یک زبان دیگر) ۲) آنهایی که سعی می کنند یک پوشش کامل (کمتر یا بیشتر) از یک زبان را تامین

---

1 Natural language Processing

کنند(توجه داشته باشید حتی انسانها همه کلمات و معانی آنها را در زبان مادری خود نمی دانند).

همه واسط های محدود شده با مشکل قابلیت سکنی (Habitability) مواجه می شوند. آیا یک کاربر می تواند دقیقا" همان زیرمجموعه ای از یک زبان را یاد بگیرد که واسط ان را پوشش می دهد؟

کاربردهای دیگر پردازش زبان طبیعی (NLP) شامل ترجمه به زبان ماشین (MT) و سیستمهای

NL CAI (Computer aided Instruction) می باشد.

بودجه مصرف شده برای خدمات ترجمه سالانه بیش از میلیاردها دلار است.قسمت اعظم این ترجمه ها , ترجمه های تجاری و اسناد تکنیکی , قراردادها و کتابهای راهنماست.

ساختار یک زبان مانند انگلیسی به خصوص ساختار نحوی ان معمولا" به وسیله قوانین گرامری بیان می شود.این قوانین یک جمله یا یک واحد از زبان را به واحدهای کوچکتر تجزیه می کند .به طور مثال ممکن است قانونی داشته باشیم (در انگلیسی) که بگوید یک جمله می تواند با یک عبارت اسمی که با یک عبارت فعلی دنبال می شود وجود داشته باشد.این قانون به اختصار می تواند به این صورت بیان شود:

(عبارت فعلی ,عبارت اسمی $\rightarrow$ جمله) $(S \rightarrow NP, VP)$

در پردازش زبان طبیعی دو تکنیک برای تجزیه و تحلیل زبان طبیعی یا NL وجود دارد.

۱) تطبسق با قالب (همچنین key board analysis).در این روش سیستم جمله وروردی را برای کلمات کلیدی خاصی پویش می کندو وقتی پیدا شدند , سیستم با یک پاشخ موجود واکنش نشان می دهد.

۲) تجزیه نحوی (Syntactic driven Parsing) :در این روش دانش قواعد یک زبان برای تجزیه و تحلیل استفاده می شود.

۷-۳-۱ گرامر (Grammer)

گرامر یک توصیف رسمی از ساختار یک زبان می باشد.یک گرامر ساختارهای مجاز یک زبان را مشخص کرده و به یک جمله اجازه تجزیه و تحلیل شدن می دهد.تجزیه یک جمله یافتن یک ساختار مجاز ممکن را شامل می شود.نتیجه معمولا" به صورت یک درخت است.(به نام درخت تجزیه).یک نمونه در شکل ۱-۷ نشان داده شده است.

۷-۳-۲ پارسر یا تجزیه کننده (Parser)

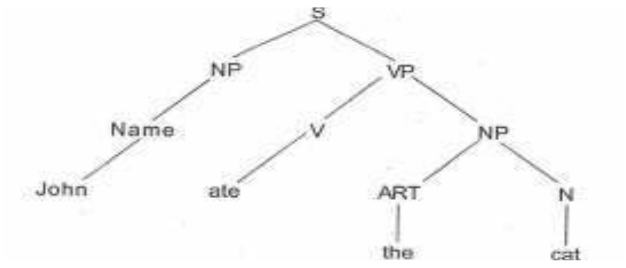
پارسر یک الگوریتم برای تحلیل یک جمله با گرامر معلوم است که به صورت:

۱)فقط جواب بلی/خیر به سوال پاسخ می دهد.آیا این جمله از قواعد داده شده پیروی می کند؟

این چنین پارسری یک پذیرنده (accepter) نامیده می شود.

۲) همچنین یک ساختار توصیفی برای جملات صحیح تولید می کند .برای مثال درخت در شکل ۱

۱-۷ به صورت لیست زیر می تواند نشان داده شود.



شکل ۱-۷

درخت مورد نظر در زبان LISP :

```
(S (NP (NAME john))
  (VP (V ate)
```

```
(NP (ART the))
  (N cat)))
```

درخت مورد نظر در PROLOG :

```
S(np(name(john)),
  Vp(v(ate),
     Np(art(the),
       N(cat))))
```

توضیح: گرامر مستقل از متن:

1. S →NP VP
2. VP →V NP
3. NP →NAME
4. NP →ART N
5. NAME →john
6. V →ate
7. ART →the
8. N →cat

توضیح: یک مجموعه ۵ تایی است به صورت(P,A,N,T,S) :

- P یک مجموعه از قوانین مستقل از متن
- A مجموعه ای از سمبولهای الفبایی از قوانین
- N یک مجموعه از سمبولهای غیر ترمینال است.
- T یک مجموعه از سمبولهای ترمینال است.
- S یک سمبول غیر ترمینال به نام سمبول شروع است.

به طور مثال:

T= {ate, cat, john, the}
N ={S, NP , VP, N, V, NAME, ART}
P ={S→NP VP , VP →V NP,….}

توجه داشته باشید چگونه قوانین تولید به قوانین گرامری و فرهنگ لغات بالا منشعب می شود.

NAME V,N, و ART سمبولهای لغوی یا pre-terminal نامیده می شود.

در گرامر زبان برنامه نویسی قواعد مستقل از متن مثل زیر می باشد:

While Statement →**while** condition **do** Statement List **end**
Statement List →Statement
Statement List → Statement ; Statement List

۷-۳-۳ انواع گرامر

گرامر می تواند به یکی از صورتهای زیر باشد:

- قوانین نامحدود (unrestricted grammars)
- قوانین حساس به متن (context-sensitive grammar)
- قوانین مستقل از متن (context-free grammars)
- گرامرهای با قاعده (regular grammars)

این ٤ نوع از گرامر در بازنویسی قواعد الفا --> بتا با یکدیگر فرق دارند.

- قواعد نامحدود:

هیچ محدودیتی در قواعد ان وجود ندارد.قواعد نامحدود به صورت گسترده استفاده نمی شوند.قدرت و توانایی زیاد آنها استفاده از آن را مشکل ساخته است.

- قواعد حساس به متن یا گرامر دگرگونی: (transformational grammar)

طول رشته در سمت چپ قانون(آلفا) باید کمتر یا مساوی با طول رشته در سمت راست(بتا) قانون باشد.

قوانین تولید در گرامر های حساس به متن می توانند بری تبدیل جملات معلوم به جملات مجهول متناظر استفاده شود.

- قواعد مستقل از متن یا گرامر ساختار عبارت (phrase structure grammar) :

همه قوانین باید به فرم $A \to alpha$ باشند که در آن A یک سمبول غیر ترمینال است و alpha یک رشته دلخواه از سمبولهاست.

- گرامرهای با قاعده یا گرامر خطی راست right linear grammar):

همه قوانین یکی از این دو فرم را در بر می گیرند: $A \to t$ و $A \to tN$, طوریکه A,N سمبولهای غیرترمینال و t عضو واژگان می باشد (سمبول نهایی)

گرامرهای با قاعده به اندازه کافی قدرتمند و توانا نیستند تا بتوانند به راحتی زبان طبیعی را توصیف کنند .(حتی زبان برنامه نویسی را).آنها گاهی می توانند برای توصیف بخشی از از زبانها استفاده شوند و این مزیت را دارند که می توانند سریعتر تجزیه شوند.

این محدودیتها به قوانین انواع گرامرها اعمال می شود.

٧-٣-٤ اشتقاق جملات از یک گرامر

برای اشتقاق جمله از یک گرامر , از سمبول S شروع کرده و ان را به عنوان رشته جاری در نظر بگیرید.

مکررا" پروسه های بازنویسی را به صورت زیر اجرا کنید:

- قانونی را انتخاب کنید که LHS (قسمت سمت چپ) ان در رشته جاری رخ دهد.(در گرامرهای مستقل از متن LHS باید سمبول غیر ترمینال باشد)
- LHS ان قانون را با RHS (قسمت سمت راست)قانون در رشته جاری جایگزین کنید.و یک رشته جاری جدید تولید کنید.

این عملیات را تکرار کنید تا زمانیکه هیچ غیر ترمینالی در رشته جاری باقی نماند.

سپس این رشته جاری تولید شده یک جمله در زبانی است که توسط گرامر ایجاد شده.(قبلا" آن یک ترم یا عبارت محسوب می شد.)

Current string	*Rewriting*
S →NP VP	S
→NAME VP	NP
→ john VP	NAME
→ john V NP	VP
→ john ate NP	V
→ john ate ART N	NP
→ john ate the N	ART
→ john ate the cat	N

تجزیه کردن شاید معکوس این پردازش باشد.(انجام گامهایی که در بالا نشان داده شده , تجزیه پایین به بالا و راست به چپ john ate the cat را تشکیل می دهد.)

۷-۳-۵ تجزیه بالا به پایین (Top –Down Parsing)

در این روش تجزیه , ما حدس می زنیم (پیش بینی می کنیم) که کدام محصول بعدا" بکار برده می شود ودر صورتیکه حدس ما اشتباه باشد , پیشنهادهای متناوب را به صورت پشته در می آوریم و در صورت نیاز به آنها بر می گردیم .

این الگوریتم می تواند پیچیدگی زمانی نمایی در جملات با گرامرهای مستقل از متنی که خوب نیستند داشته باشد.

با گرامر well behaved یک الگوریتم می تواند زمان خطی داشته باشد.گرامرهای NL معمولا" well behaved نیستند.نمونه زیر یک تجزیه بالا به پایین را نشان می دهد.

S →NP VP
NP →ART N | NAME
PP →PREP NP
VP → V | V NP |V NP PP |V PP

Sentence :1 The 2 dogs 3 cried 4.

Backup states	*Position*
S →NP VP	1
→ART N VP	1

NAME VP	1
→ (The) N VP	2
NAME VP	1
→(dogs) VP	3
NAME VP	1
→ V	3
V NP	3
V NP PP	3
V PP	3
NAME VP	1
→ (cried.)	4

٧-٣-٦ تجزیه پایین به بالا (Bottom –up Parsing)

Thee dogs cried → ART N V
→ NP V
→ NP VP
→ S

با استفاده از متد تجزیه پایین به بالا همه گرامرهای مستقل از متن می توانند در $n \times n$ گام تجزیه شوند طوریکه n طول جمله باشد.

به دلیل اینکه ممکن است تجزیه قابل پیش بینی پیچیدگی زمانی نمایی داشته باشد لذا ممکن است قسمتهایی از جمله خصوصا" قسمتهای گیج کننده مجددا" تجزیه شوند.

٧-٣-٧ تجزیه نموداری (Chart Parsing)

نمودار یا chart رکوردی از تمام زیرساختارهاست که در طول تجزیه شدن ساخته شده اند.نمودار گاهی ممکن است well-formed substring table نامیده شود.نمودارهای واقعی به سرعت پیچیده می شوند.

نمودارها برای جملات محذوف یا elliptical نیز مورد استفاده قرار می گیرند.

1 : Q .How much are apples?
2 : A .Thirty cents each.
3 : Q .Plums?

تجزیه جمله سوم به عنوان جمله حذفی می باشد ولی تمام آنها محذوف نیستند و در تجزیه " آلو" به عنوان NP روی چارت می باشد.
تجزیه کامل کل مکالمه به عنوان نوعی ساختار می تواند مفید باشد.

الگوریتم تجزیه پایین به بالا مبتنی بر چارت (A Bottom up Chart-based Parsing Algorithm)

این الگوریتم تاثیر پردازش تجزیه پایین به بالا را بیان می کند.
تجزیه جمله به طول n با n x n x n گام ضمانت شده است و بهتر از گرامر well-behaved با (n x n steps or n step) کار می کند.
الگوریتم اجزاء اصلی جمله را می سازد.(عبارتی یا کلمه ای)
نکات ۲ تا ۹ زیر به طور کامل مشخص نمی کند که کدام گامهای تجزیه انجام می شوند.
یک راه معقول اینست که یک کلمه را پویش (مانند نکته ۳) و بعد تمامی گامهای تجزیه ممکن در(٤ تا ۷) را قبل از پویش کلمه دیگر انجام داد.
تجزیه زمانی کامل می شود که آخرین کلمه خوانده شود و تمامی گامهای تجزیه برای آن اجرا شود.
ورودی های پارسر:جمله , کلمه , گرامر.

عملیات تجزیه کننده:

۱) این الگوریتم در دئ ساختار داده ای عمل می کند.نمودار فعال که مجموعه ای از کمانهای فعال و جزء اصلی است از ابتدا خال هستند.
۲) گرامر برای در بر گرفتن قوانین الحاق کلمات مورد توجه قرار می گیرد.برای مثال اگر" fly " در فرهنگ لغات به عنوان کلمه استفاده شده باشد و در همان مدخل لغوی به عنوان فعل نیز به کار برده شود سپس به صورت قسمتی از گرامر زیر خواهد بود:

N → fly
V → fly

۳) کلمه ای مانند fly پویش شده است, اجزاء اصلی متناظر با عنوانهای کلمه ای ساخته می شوند:

N1: N → fly FROM 2 TO 3, and
V1: V → fly FROM 2 TO 3

٤) اگر گرامرشامل قوانینی مانند NP → ART ADJ N و جزء اصلی مانند ART1: ART → the FROM m TO n در جمله یافته شود سپس کمان فعال ARC1: NP → ART1 * ADJ N FROM m TO n به چارت یا نمودار فعال اضافه می شود.

علامت (*) در کمان فعال نشانه کرانه بین جزء اصلی یافت شده و جزء اصلی هنوز یافت نشده می باشد.

۵) پیش روی (*): اگر نمودا ریا چارت فعال کمان فعلی مانند زیر راداشته باشد:

ARC1: NP →ART1 * ADJ N FROM m TO n

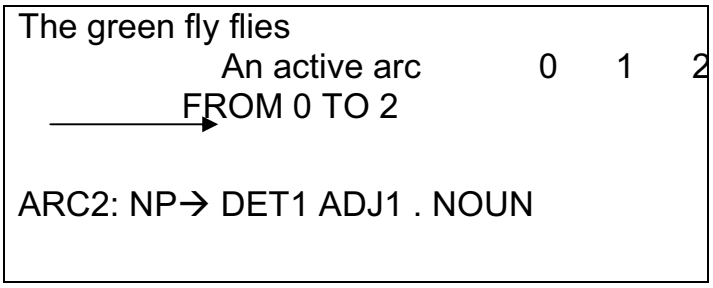
و یک جزء اصلی در نمودار نوع ADJ (مثلا" اولین آیتم بعد از *) وجود داشته باشد می گوییم:

ADJ1: ADJ → green FROM n TO p

اگر FROM در جزء اصلی با موقعیت TO در کمان فعال تطابق پیدا کند , سپس (*) می تواند پیشرفت کند و یک ئکمان فعال جدید تولید کند.

ARC2: NP → ART1 ADJ * N FROM m TO p

در شکل زیر کامن جدید از ۰ تا ۲ تولید شده اند.



٦) اگر پردازش برای پیش روی * یک کمان فعال تولید کند که * در قسمت دورتری در سمت راست قانون قرار بگیرد :همانند:

ARC3: NP → ART1 ADJ1 N1 * FROM 0 TO 3

سپس این کمان به یک جزأ اصلی تبدیل می شود:

NP1: NP → ART1 ADJ1 N1 FROM 0 TO 3

همه کمانهای فعال در این جهت کامل نمی شوند.

٧) هم اجزاء اصلی و هم عبارات می توانند در گامهای ٤ و ٥ استفاده شوند.مثلا" اگر گرامر شامل قانون S → NP VP باشد , به محض اینکه جزء اصلی NP1 در گام ٥ ام ساخته شد , ساختن کمان فعال جدید ممکن خواهد بود.

ARC4: S → NP1 * VP FROM 0 TO 3

٨) وقتی جزء اصلی بعدی ساخته شد ر می توانند نامهایی مثل NP2,NP3,ADJ2 و ... داشته باشند.

٩) هدف اصلی تجزیه بدست اوردن جزء اصلی عبارت (معمولا" از نوع S) که FROM آن 0 و TO آن طول جمله می باشد است.

نمونه ای از تجزیه چارت یا نمودار:(Chart parsing)

گرامر عبارتست از :

1 . S → NP VP
2 . NP → ART ADJ N

3 . NP → ART N
4 . NP → ADJ N
5 . VP → AUX V NP
6 . VP → V NP

گامهای تجزیه :

* sentence *: **the** * * position *:1
* constituents *:
ART 1: ART →the FROM 0 TO 1
* active – arcs *:
ARC1: NP →ART * ADJ N FROM 0 TO 1 [rule 2]
ARC2: NP →ART1 * N FROM 0 TO 1 [rule3]

* sentence *: the **large** * position *:2
* constituent *: add
ADJ1: ADJ → large FROM 1 TO 2
* active – arcs *:add
ARC3: NP →ART1 ADJ1 * N FROM 0 TO 2 [arc1*→]
ARC4: NP →ADJ1 * N FROM 1 TO 2 [rule4]

* sentence *: the large **can** * position*: 3
* constituents * :add
NP2: → ART1 ADJ1 N1 FROM 0 TO 3 [arc3*→]
NP1: NP → ADJ1 N1 FROM 1 TO 3 [arc4 * →]
N1: N → can FROM 2 TO 3
AUX1: AUX → can FROM 2 TO 3
V1: V → can FROM 2 TO 3
*active – arcs *: add
ARC5: VP → V1 * NP FROM 2 TO 3 [rule 6]
ARC6: VP → AUX1 * V NP FROM 2 TO 3 [rule 5]
ARC7: S → NP1 * VP FROM 1 TO 3 [rule1]
ARC8: S →NP2 * VP FROM 0 TO 3 [rule 1]

*sentence * : the large can can * position*:4
constituents : add

ARC9: VP → AUX1 V2 * NP FROM 2 O 4 [arc6 *→]
ARC10: VP → V2 * NP FROM 3 TO 4 [rule6]
ARC11: VP → AUX2 * V NP FROM 3 TO 4 [rule 5]

sentence : the large can can hold
position : 5
constituents: add
N3: N → hold FROM 4 TO 5
V3: V → hold FROM 4 TO 5
active – arcs : add
ARC 12: VP → AUX2 V3 * NP FROM 3 TO 5 [arc11* →]
ARC13: VP → V3 * NP FROM 4 TO 5 [rule 6]

sentence: the large can can hold the
position: 6
constituents : add
ART2: ART → the FROM 5 TO 6
active-arcs : add
ARC14: NP → ART2 * ADJ N FROM 5 TO 6 [rule 2]
ARC15: NP → ART2 * N FROM 5 TO 6 [rule 3]

sentence : the large can can hold the water
* position * : 7
* constituents * : add
S2: S → NP1 VP2 FROM 1 TO 7 [arc7 * →]
S1: S → NP2 VP2 FROM 0 TO 7 [arc8* →]
VP2: VP → AUX2 V3 NP3 FROM 3 TO 7[arc12 * →]
VP1: VP → V3 NP3 FROM 4 TO 7 [arc13 * →]
NP3: NP → ART2 N4 FROM 5 TO 7 [arc15* →]
N4: V → water FROM 6 TO 7
V4: V water FROM 6 TO 7
active-arcs: add
ARC16: VP → V4 * NP FROM 6 TO 7 [rule 6]
ARC17: S → NP3 * VP FROM 5 TO 7 [rule 1]

این گامها را به صورتی که در شکل ۲–۷ نشان داده شده است تکرار کنید.

عیب روش پایین به بالا این است که جز اصلی بی ارتباط را نیز پیدا خواهد کرد مانند "hold the water" که این جزء توسط روش تجزیه بالا به پایین مورد توجه واقع نخواهد شد.یک تجزیه کننده بالا به پایین مستقل از متن می تواند یک چارت داشته باشد.

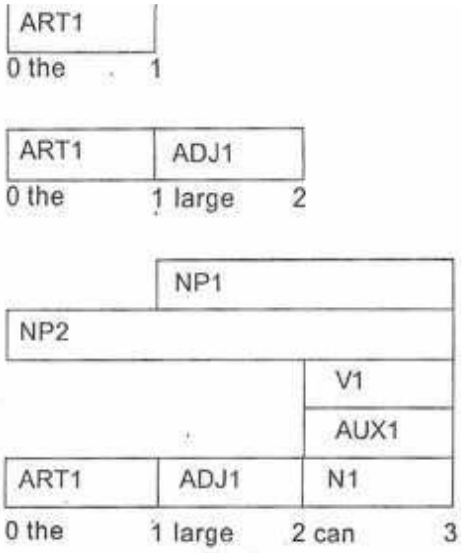
ثبت ساختار جمله(Recording Sentence structure) :

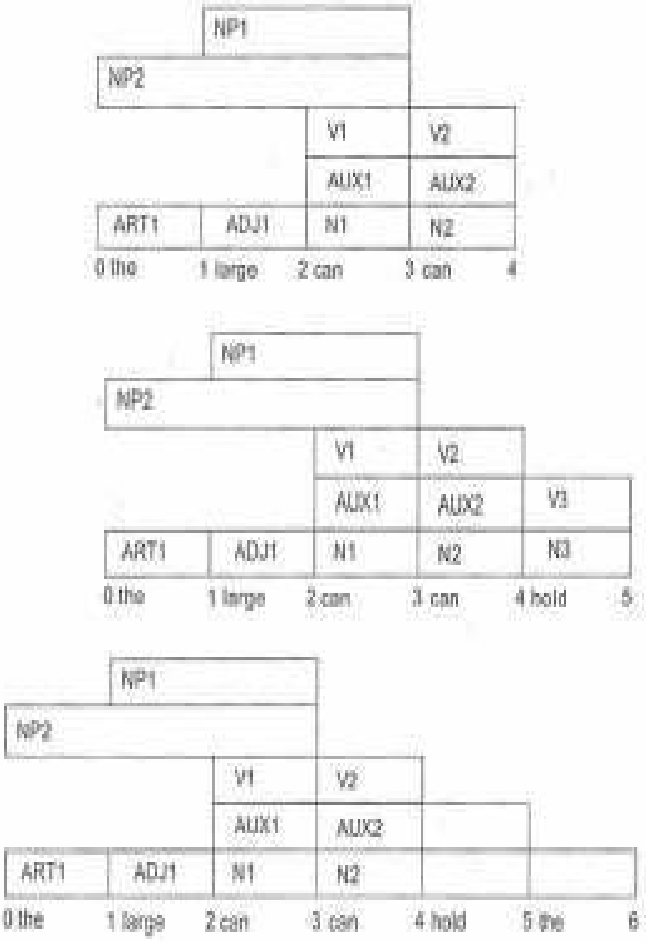
جمله زیر را در نظر بگیرید:

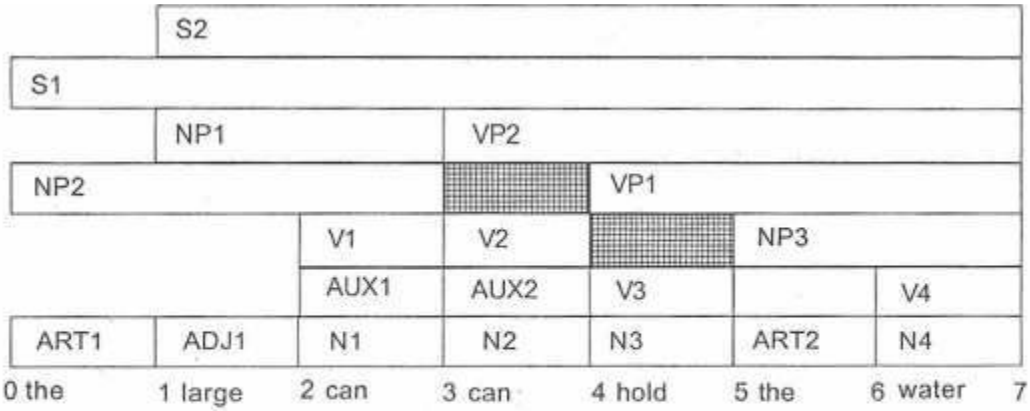
" Jack found a dollar"

```
(S SUBJ (NP NAME jack)
     MAIN-V find
     TENSE past
     OBJ (NP ART a HEAD dollar))

S( subj(np(name(jack))),
     Main(find),
     Tense(past),
Obj(np (art (a),head (dollar))))
```


ART1
0 the 1
ART1
ADJ1
0 the
1 large
2
NP1
NP2
V1
AUX1
ART1
ADJ1
N1
0 the
1 large
2 can
3


NP1
NP2
V1
V2
AUX1
AUX2
ART1
ADJ1
N1
N2
0 the
1 large
2 can
3 can
4
NP1
NP2
V1
V2
AUX1
AUX2
V3
ART1
ADJ1
N1
N2
N3
0 the
1 large
2 can
3 can
4 hold
5
NP1
NP2
V1
V2
AUX1
AUX2
ART1
ADJ1
N1
N2
0 the
1 large
2 can
3 can
4 hold
5 the
6



شکل ۲-۷

۷-۳-۸ گرامرو برنامه نویسی منطقی

:(Grammars & Logic programming)

قوانین گرامر می تواند مستقیما" در PROLOG کدنویسی شود.

S → NP VP ……s(p1,p3) :- np(P1,P2), vp(P2,P3).

این عبارت بدین معناست که S وجود دارد از موقعیت P1 به موقعیت P3 به شرط اینکه وجود داشته باشد NP از P1 به P2 و VP از P2 به P3 .همینطور:

VP →V NP ……vp(P1,P3) :- v(P1,P2), np(P2,P3).
NP → NAME …….np(P1,P2) :-propernoun(P1,P2).
NP → ART N …….np(P1.P3) :- art(P1,P2), noun (P2,P3).

لغات می توانند توسط گزاره ها تعریف شوند.مانند:

NAME → John ….. isname(john).
V → ate ….. isverb(ate).
ART →the ….. isart(the).
N → cat ….. isnoun(cat).

برای هرعنوان لغتی مانند ART یک گزاره (predicate) تعریف می کنیم مثلا" برای ART

Art (FROM,TO) :_word (Word, FROM ,TO), isart (Word)

این مطلب زمانی درست است که کلمه ای که موقعیت آن مشخص شده , از همان عنوان باشد.

در مثال زیر:

Word (Word , FROM ,TO)

عبارت بالا مشخص می کند که word در جمله ورودی , بین موقعیت های FROM و TO قرار دارد.به هماین صورت:

```
Noun(FROM,TO) :- word(Word ,FROM , TO), isnoun(Word).
V(FROM ,TO) :- word (Word , FROM ,TO), isverb(Word).
Propernoun(FROM, TO ):- word(Word , FROM ,TO), isname(Word).
```

با استفاده از سیستمی که قبلا" توصیف شده بود,می توانیم موقعیت کلمات در جمله هایشان را ثابت کنیم.

```
Word(john ,1,2).
Word(ate ,2, 3).
Word(the ,3 ,4).
Word(cat, 4 ,5).
```

سپس از query پرولوگ استفاده کنید?s(1,5)

نتیجه آ ن ** بله خواهد بود.

برای درک ساختار جمله, آرگومان ها و استدلال های اضافی برای عبور از گلوگاهها اضافه کنید.

```
S(p1,p3, S(np,vp)) :- np(P1,P2,NP), vp(P2,P3,VP),
Vp(P1,P3,vp(v(Verb),NP)) :- v(P1,P2,Verb), np(P2,P3,NP).
Np(P1,P2,np(name))) :-proper(P1,P2,Name).
Np(P1,P3,np (art(Art),noun(Noun)))
Art(P1,P2,Art),
Noun(P2,P3,Noun).

Art(FROM,TO,Word) :- word(Word,FROM,TO),
isart(Word).
Noun(FROM,TO,Word) :-
WORD(Word,FROM,TO),isnoun(Word).
V(FROM,TO,Word) :- word(Word,FROM,TO),
isverb(Word).
Proper(FROM,TO,Word) :-
word(Word,FROM,TO),isname(Word).
```

۷-۳-۹ زبان نمایش دانش (Knowledge Representation Language)

موضوع نمایش دانش مرتبط با سیستم NLP که شامل اطلاعات کلی درباره جهانی که نیازمند زبانی قابل فهم است , می باشدو نوعی دانش مخصوص که با سیستم NLP مرتبط است.

زبان نمایش دانش یا (KRL=knowledge Representation language)

به زبان مخصوصی اشاره دارد که برای کدبندی دانش استفاده می شود.

مجموعه اطلاعات و دانشی که توسط سیستم استفاده می شود پایگاه دانش (KB) نام دارد.

نمایش بر پایه FOPC :

زبانی که استفاده می شود اشتراکاتی با زبان منطقی دارد.بخشی از زبان شامل ثابتها مانند john1 ,تابع های کاربردی مانند father (john1) و متغیرها مانند xو y می باشد.توجه داشته باشید فرم منطقی زبان از ثابتها استفاده نمی کند , همه چیز با

متغیرهای مستقل بیان می شد تا نمایش متن مستقل را نگه دارد.برای مثال , فرم منطقی (NAME j1 "john") در متن خاصی ممکن است به عنوان ثابت john1 در KB نمایش داده شود.

تعاریف محدود شده در KRL معنی پیدا می کنند همانطور که در زبان منطقی معنی دارند.

$\exists x : Person(x) Happy(X)$ There is a happy person

$\forall x : Person(x) Happy(X)$ All people are happy

عبارت دوم معادل است با $\forall x Person(x) \rightarrow Happy(X)$.

Many KR systems do not explicitly use quantifiers. Their variables are all tacitly
universally quantified, as in PROLOG. Thus eats(john1, X) :- fried(X). means "John
eats anything if it's fried". Literally anything, of course-if fried(cartyres1) is stored
in the KB, then eats(john1, cartyresl) is true.

تعداد زیادی سیستم KR به طور واضح و آشکار از کمیت سنج استفاده نمی کنند.متغیرهای آنها همگی به طور فراگیر سنجیده شده اند چنان که در PROLOG است.بنابراین

Eats(john1, X) :-fried(x) به معنی "john eats anything if it's fried." است.

ثابتهای SKOLEM و توابع:

متغیرهای با سور وجودی به وسیله تکنیکی به نام Skolemization بکاربرده شده اند به طوریکه متغیرها را با ثابتهای جدید جایگزین می کند.برای مثال فرمول $\exists y \forall x loves(x, y)$

به صورت فرمول روبرو کدبندی می شوند. $loves(X, sk1)$ جایی که sk1 یک ثابت جدید است که بجای شئ ای که وجود را ثابت می کند قرار می گیرد.
وابستگی های سورها با استفاده از توابع جدیدی به نام Skolem function نشان داده می شود.به عنوان مثال فرمول $\forall y \exists x loves(x, y)$ به صورت فرمول loves(sk2(Y),Y) کدبندی خواهد شد.جاییکه sk2 یک تابع جدید است که به صورت بالقوه یک شئ جدید برای هر مقدار Y تولید می کند.

معنی شناسی(Semantics) :

وظایف زیادی تحت عنوان semantic وجود دارند.(semantic به معنای مطالعه معانی می باشد.برای مبحث مورد مطالعه ما , semantic به معنی پردازش تعیین معنی جملات ورودی است.)

- کلمه صحیح را برای معنی کردن انتخاب کنید.
- ابهامات را از آن بردارید.زمانی که از لحاظ نحوی تفسیرهای متعددی ممکن است , تفسیری را انتخاب کنید که معنی بدهد.

" امروز صبح ,جان سگ مرا دید که به طرف محل کارش رانندگی می کرد."

- مراجع ضمیر را مشخص کنید.

" بیل دوچرخه جان را خواست.
او آن را فروخت."

- نمایشی از معنی زبان ورودی را تولید کنید.این باز نمایی ممکن است تا حدی از ورودی متفاوت باشد.

"جان خرگوشش را به بیل فروخت.
آیا بیل یک حیوان خانگی خریده است؟"

- اطلاعات حذف شده را پر کنید.

" من امروز از کار دیرم شد.

ماشین من استارت نمی زد.

باتری آن تمام شده بود.

باتری قسمتی از اتومبیل من است.تمام شدن باتری باعث شد اتومبیل من استارت نزند.استارت نزدن ماشین باعث شد که من دیر سرکار برسم برای اینکه من معمولا" برای رفتن به سر کار از ماشین استفاده می کنم و اگر ماشینی استارت نزند , نمی تواند حرکت کند.

ابهامات(Ambiguity) :

ابهامات زیادی بیش از آنچه بتوانیم تصور کنیم در یک زبان طبیعی وجود دارد.

ابهامات لغوی:

یک کلمه می تواند نقش های متعددی در سخن داشته باشد و برای هر نقش دارای یک معنی به خصوص باشد.

تعداد ترکیبات معنایی , محصول تعدد معانی برای هر کلمه است.

ابهامات نحوی:

عبارات ممکن است به قسمتهای مختلفی از جمله مربوط باشند.مخصوصا" ترکیبات عطفی و عبارت گزاره ای قسمتهای پر زحمت هستند.

۷-۳-۱۰ مثالها

در این قسمت ما به دومثال برای رفع برخی مشکلات NL با استفاده از گرامر شرط قطعی

(DCG=Definite Clause Grammar) می پردازیم.

اولین مثال نشان می دهد که چگونه یک عبارت فعل-اسم (noun-verb) می تواند از متغیرهایی که خود می توانند به صورت غیر ترمینال درDCG ظاهر شوند بدست آید.

فقط فعلهای زمان حال شرح داده خواهند شد.سپس ما به مشکلات ترجمه از یک زبان به زبان دیگر خواهیم پرداخت .

در اینجا یک جمله به زبان فرانسه تجزیه شده و در همان زمان ترجمه انگلیسی در یکی از متغیرها انجام شده است.

ما در ابتدا با معرفی کلی از نحو و ترجمه DCG در PROLOG آغاز می کنیم.بیشتر نسخه های PROLOG توانایی تعریف زبان و عملیات بر روی آنها توسط DCG را دارند. گرامر های DCG بسیار شبیه گرامر های مستقل از متن هستند ولی DCG ها صراحتا" به دلیل داشتن حساسیت نسبت به متن , قویترند.

آسانترین فرم (Simplest form):

پایه ای ترین فرم DCG اساسا" همان گرامر Context Free یا مستقل از متن است.

با یک مثال توضیح می دهیم.یکبار قوانین به مترجم PROLOG نشان داده می شود وآنها به شرطها و عبارات خالص در PROLOG ترجمه می شوند.

اگر فهرست نویسی انجام شد سپس ترجمه قابل مشاهده خواهد بود.(فهرستی از شرطها)

مثال ۱:

در اینجا گرامری برای تشخیص انواع فرم های اعداد داریم.اجازه دهید ببینیم چقدر به تعریف نرمال گرامرهای مستقل از متن نزدیک است؟

Digit → [0], [1], [2] ,[3] ,[4] ,[5] ,[6] ,[7] ,[8] ,[9].
Nat num → digit .

Nat num → digit,nat num (توالی از ارقام

Int →nat
num.
Int → sign ,nat num.
Real → int.
Real →int,[.],nat num.
Sign → [-],[+].

اشیا درون " [] " عناصر ترمینال زبان هستند.شناسه های دیگر نقش غیر ترمینال را دارند.

قوانین در سمت راست می توانند توسط علامت | ازهم تمیز داده شوند.قوانین بالا به صورت زیر در پرولوگ ترجمه می شوند.

```
Nat num(A,B) :-
                Digit(A,B).
Nat num(A,B) :- digit(A,C),
Nat num(C,B).
Sign(A,B) :-
                ( “C” (A,-B) |
                  “C” (A ,+,B).
Real(A,B) :-
Int(A,B).
Real(A,B) :-
                Int(A,C),
                “C” (C,”.”,D),
                Nat num (D,B).
Real (A,B) :-
                Sign(A,C),
                “C” (C,”.”,D),
                Nat num(D,B).
Int(A,B) :-
                Nat num (A,B)
Int (A,B) :-
```

```
                Sign(A,C),
                Nat num (C,B).
Digit (A,B) :-
                ("C"(A,0,B ) |
                 "C" (A,1,B) |
                 "C" (A,2,B) |
                 "C" (A,3,B) |
                 "C" (A,4,B) |
                 "C" (A,5,B) |
                 "C" (A,6,B) |
                 "C" (A,7,B) |
                 "C" (A,8,B) |
                 "C" (A,9,B) ).
```

اگر ما nat num(A,B) را برداریم , سپس تفسیر این گزاره اینگونه خواهد بود که توالی عناصر B , قسمت انتهایی A می باشد.به عبارت دیگر زوج (A,B) را برای ایجاد لیست تفاضل استفاده می کنیم.

تا حالا مثال nat num([1,2,3],[]) درست بوده است وبه طور کلی تر عبارت مذکور برای هر x ای درست است.گزاره داخلی "C" برای تشخیص ترمینالها , و عبارت "C"(X,Terminal,Y) به این معنی است که ترمینال هد یا سر X , و دنباله آن Y است (در صورتی که صورت روبرو تعریف شده باشد. "C"([X|Xs],X,Xs) برای آسانتر نمودن مثال بالا میتوانیم رویه ای بنویسیم که می تواند اعداد در نمایش نرمال را به یک توالی از سیمبولهای جداگانه تبدیل کند.به عنوان مثال عدد 123+ به صورت لیست [+,1,2,3] و عدد 1.23 به صورت [1,".",2,3] تبدیل خواهد شد.

توجه داشته باشید که وجود "." ضروری می باشد در غیر اینصورت پرولوگ آن را به عنوان یک ترمیناتور در نظر خواهد گرفت و پیغام خطا خواهد داد.همچنین اگر عددی شامل علامت باشد باید به صورت یک اسم که نیازمند اتم است, نقل شود.(عدد

+123 یک اتم نیست چون با + شروع شده).

```
Convert ti list(Atom,Lst):-
        Name(Atom,Atom Lst),
        % convert to list of asci numbers
         List to vals(AtomLst,Lst).
Asci to char(Asc,Chr):-
         Name(Chr,[Asc]).
         List to vals([],[]). % apply down a list
List to vals([H|T],[HV|TV]):-
         Asci to val(H,HV),
         List to vals(T,TV).
```

به جای استفاده کردن از گزاره های ترجمه شده برای تشخیص عناصر یک زبان , می توانیم همچنین از عبارات گزاره ای داخلی (built-in) استفاده کنیم.عبارت (NT,Lst) به این معنی است که L ام به وسیله غیر ترمینال NT تولید شده است.به عنوان مثال عبارت (real[1,2,".",4,5]) درست است.

اکنون گرامر بالا اعدادی که با صفر شروع می شوند مانند ۰۰۳ را می پذیرد.دگرگون ساختن گرامر توسط معرفی غیر ترمینال ها برای نپذیرفتن و رد چنین اعدادی خیلی مشکل نیست. تغییرات به صورت زیر می باشد:

```
Nonz digit →[1] |[2]| [3] | [4] | [5] | [6] |
                    [7] | [8] | [9] .% a nonz digit is in 1-9
Digit → [0] | nonz digit.
Red nat num → digit | nonz digit, nat num.
Red int → red nat num | sign , red nat num.
Red real → red int | red int, ["."], nat num.
```

برای عبارت (red nat num(0,0,2,3)) اشتباه ولی برای (nat num ([0,0,1,2]) درست خواهد بود.

مثال پیچیده تر:

DCG همچنین این امکان را دارد که کدهای پرولوگ را در محفظه (body)

جاسازی کند.این کار توسط جا دادن کد مورد نیاز در {} انجام می گیردوهر آنچه داخل {} باشد , غیر قابل تغییر توسط مفسر خواهد بود.پارامترها نیز می توانند به عنوان آرگومان در سمبول های غیرترمینال ظاهر شوند بنابراین نتایج می توانند در محاسبات دیگر استفاده شوند یا به عنوان side effect در تشخیص یا فهم یک زبان در نظر گرفته شوند.

مثال ۱:

پارامتری را به مثال قبلی اضافه کردیم بنابراین بعد از تشخیص عدد درست , آن پارامتر حاوی مقدار یا ارزش آن عدد خواهد بود.همچنین لازم است بدانیم چند تا رقم در عدد ما ظاهر می شوند.

مثلا" عبارت (nat num (N),[1,2,3,1]) متغیر N را به مقدار 1231 معرفی می کند.

```
Digit (0) →[0]. % a digit is in 0-9
Digit (1) →[1].
Digit (2) →[2].
Digit (3) →[3]. % etc.
….
Digit (9) →[9].
Nat num (N,1) →digit(N).% a natural number is a sequence of digits
Nat num(N,ND) → digit (D), nat num (N2,ND1),
                                % ND is the number of digits.
{plus (ND1,1,ND),
                Power ten(D,ND1,P),
                Plus (P,N2,N)}.
                Int(N,D) → nat num (N,D). % an intiger is a natural number
                % possibly with a sign
Int(N,D) →[+],nat num(N,D).
Int(N,D) →[-], namt num(N1,D),
                { N is – N1}.
Real(R) → int ( R ) . % a real is given in normal decimal notation
```

```
Real (R ) → int(I),[“.”] , nat num(N, ND),
                  { neg power ten(ND,inv P),
                   ( I >= 0 → R is I+ N * InvP;
                   R is I – N * Inv P)}.
Real ( R) → [+],[“.”], nat num(N,D),
                { neg power ten(D,inv P),
                R is –N * invP}.
Real (R ) →[-],[“.”] , nat num (N,D),
                    { neg power ten (D,InvP),
                     R is – N * InvP }.
                    Power ten(D,0,D) :-!.
Power ten(D,E1,P) :-
                   E1 >0,
                  Plus(E,1,E1),
                  Power ten(D,E,P1),
                  P is P1 * 10,! .
Neg power ten(0,1_ :- !.
Neg power ten(N,P) :-
                    N>0,
                    Succ(N1,N),
                    Neg power ten(N1,P1),
                    P is P1/10.
```

یک فراخوانی مثلا" real (R,[-2,3,”.”,45],[]) مقدار درست را بر می گرداند و R معادل است با 23.45- و یک فراخوانی عبارت (int(N,D),[-,1,2,3]) N را به عدد 123- معرفی نموده و D تعداد ارقام عدد که ۳ است را نشان می دهد.

مثال ۲:

```
% parse simple English sentence
% for time being just present as a list of identifiers
% e.g. [ the ,big ,cat,kicks,the,black,dog]
% first we have a straightforward generator
% will not repeat adjective  like big  big girl!
% also check whether we need an “an” or an “a”.
Adjnounph (CV) → noun(CV).
```

```
Adjnounph (CV) → adjective(CV,Adj), noun().
Adjnounph (CV) → adjective (CV,Adj),adjective(Adj2),
                    { Adj \== Adj2}
                     Noun ().
Nounphrase →det(CV),adjnounph(CV).
       Sentence → nounphrase, verb , nounphrase.
% now some explicit examples
Det(cons) →[the] | [a].
Det(vowel) → [an].
Verb → [hit] | [kicks] | [kisses].
Noun(cons) → [cat] | [boy] |[girl].
Noun(vowel) → [owl] | [ox].
Adjective(cons,big) → [big].
Adjective(cons,black) →[black].
Adjective(cons,brown) → [brown].
Adjective(cons,tabby) → [tabby].
Adjective(vowel,awful) → [awful].
Adjective(vowel,awesome) →[awesome].
```

گرامر بالا جمله "an awful owl hits an ox" را پذیرفته و جمله "an awesome cat kisses an awful girl" را رد خواهد کرد.

اجازه دهید به سیستم هایی بپردازیم که سعی می کنند انگلیسی (ترجیحا" هر زبان طبیعی)را به عنوان ورودی بفهمند .فقط این اواخر است که ما برنامه هایی داریم که می توانند از عهده جمله های بسیار پیچیده برایند و در حالی که بسیار بزرگ و کند هم می باشند .موفقیت های بیشتری می توانند وجود داشته باشند در صورتی که دامنه سخن محدود شود.همه برنامه های اولیه به این فرم بودند.

۷-۳-۱۱ الیزا (ELIZA)

این تلاشی است برای بازی کردن نقش یک روانپزشک (Rogerian therapist) که تنها با علم نحو (pure syntactic) کار می کند .

این روش مانند این است که روانپزشک چیزی را به بیمار خود تلقین کند نه اینکه بیمار را وا دارد تا احساسات و افکار خود را بروز دهد.الیزا لیستی از قالب ها یا الگوهایی

دارد که سعی می کند سوالهای ورودی را با لیستی از کلمات کلیدی طبقه بندی شده یا منظم تطبیق دهد.

الگوها به یکی از فرم های زیر می باشند:

۱) کاراکتر های مطابق با کاراکترها

۲) <string> %

۳) <string> % < string>

رشته ها لیستی از کاراکترها می باشند که باید دقیقا" مطابقت داشته باشند و علانت ٪ با زیر رشته ای که باقی مانده , تطبیق می یابد.

مقدار ٪ برای دادن پاسخ معقولتر استفاده خواهد شد بنابراین ما احساس می کنیم که ماشین واقعا" ورودی ها را فهمیده است .البته فقط در وضعیت ۱ است که می دانیم کاربر دقیقا" چه چیزی را به عنوان ورودی وارد می کند.

اجازه دهید به قالبی که با فرم دوم مطابقت می کند بپردازیم. پاسخ های ممکن عبارتند از:

الف) چه فرقی می کند که من اعتقاد دارم یا نه؟

ب) شاید اعتقاد دارم و شاید هم ندارم.

ج) بله , اعتقاد دارم.

این پاسخ ها در جواب سوال زیر است:

آیا شما به خدا اعتقاد دارید؟ سپس یک پاسخ ممکن با توجه به پاسخ دوم, خواهد بود: شاید من به خدا اعتقاد دارم و شاید هم ندارم.

تکنیک hashing برای بازیابی سریع و تطبیق الگوها به کار برده می شود.برای مثال در اجرای یونیکس , ابتدا تطبیق در دو کاراکتر اول انجام می شود.اگر هیچ الگوی تطبیقی وجود نداشته باشد سپس یک جستجو با استفاده از کلمات کلیدی در جملات ورودی انجام می شود.کلمات کلیدی مرتب شده اند.لیست پاسخ های ممکن برای کلیدی (difficult) می تواند به صورت زیر باشد:

a) tell me about your difficulties
b) what do you mean by difficult

اگر هیچ کلمه کلیدی در جمله نباشد که با کلمات کلیدی در پایگاه داده مطابقت داشته باشند سپس تعدادی ملاحظات یا تبصره های غیرالزامی استاندارد می تواند به صورت زیر ایجاد شود.

a) let's change the subject بیاید موضوع را عوض کنیم

b) go on ادامه بده!

یک رهیافت جایگزین , انتخاب تصادفی از یکی از کلمات کلیدی که قبلا" استفاده شده می باشد.رکوردی از ٤ کلمه کلیدی که قبلا" استفاده شده نگهداری می شود. بنابراین در اینجا سیستم از تاریخچه محدود شده استفاده می کند.این تکنیک اساسی است که الیزا از آن استفاده می کند به غیر از زمانی که یک قالب یا الگویی که با یکی از فرم های ۲ یا ۳ تطبیق یافته , کنار گذاشته شود. چند تا پیش پردازش باید برای تولید پاسخ درست از لحاظ گرامری اجرا شوند.

اگر کاربر ضمیر شخصی (من) را در ورودی ذکذ کند, دیگر در جایی که هست با ضمیر دیگری نمی تواند جایگزین شود.

فرض کنید الگویی وجود داشته باشد به صورت because % , همراه با یک قالب پاسخ.

% ? that's incredible!

سپس در ورودی داریم:

Because I am clever.

یک جایگزین صریح و آسان به صورت زیر نتیجه خواهد داد:

I am clever ? that's incredible !

سپس یک جایگزینی به جای i به صورت زیر خواهد بود.

You am clever? that's incredible!

البته این جمله هنوز درست نیست و باید تغییراتی روی آن صورت گیرد. you am

به you are تغییر یابد.این دو مرحله fix و grammar نامیده می شوند.

Fix تغییرات زیر را انجام می دهد:

You → me
I → you
Me → you
That you → that I

همچنین تغییرات دیگری نیز انجام می گیرد.

در اینجا بعضی از خطاهای گرامری ساده اصلاح می شوند.

You am → you are
Me are → I am

با دیگر تغییرات مشابه.

مهم است که بدانیم مرحله fix قبل از مرحله grammar اجرا می شود.

Do you think I am clever?
⇓
May be I think I am clever and may be I don't.
⇓ *fix*
may be I think you am clever and may be I don't.
⇓ *grammar*
May be I think you are clever and may be I don't.

این روش نمی تواند تمامی خطاهای گرامری را رفع کند و نسبتا" برای آن آسان است که جوابهای غیر گرامری بدهد.برای مثال در اجرای یونیکس جاری , توجه نمی کند که چه چیزی باید به جای fix, you شود. پس برای سوءال زیر خواهیم داشت:

Do you belive in what you do?

جواب خواهیم گرفت:

May be I do belive in what **me** do or maybe I don't.

البته در اینجا آسان است که تغییرات fix را برای اصلاح کردن اعمال کنیم ولی مثالهای زیادی وجود دارند که می توانند برنامه را فریب دهند.تنها اطلاع دقیقی که می توانیم درباره برنامه ای بگوییم این است که شامل fix یا grammar است ولی هیچ قدرت

قیاسی برای یک نمونه وجود ندارد.
به هیچ عنوان این فکر درستی برای با هوش بودن نیست.برای بهبود دادن , نیازمند fix های بیشتری برای مواجه شدن با موردهای جدید است.

۷-٤ شناسایی کلام یا گفتار[1]

شناسایی سخن عمل تشخیص از بازسازی کلماتی که یک سیگنال صوتی معلوم را تولید می کنند است. به عبارت دیگر, مسئله ترجمه و انتقال سیگنالهای صوتی دیجیتالی کد شده یک گوینده در زبان طبیعی (انگلیسی) به متن آن در همان زبان می باشد.
با توجه به ابهاماتی که در سطوح مختلف این مسئله وجود دارد(مثل صدای پس زمینه, لهجه گوینده, صدای دیجیتال شدن و ...) می توانیم مسئله را رسما" به صورت زیر مشخص کنیم:

$$\text{Argmax}_{words}\ P(words \mid signal)$$

طوریکه words , رشته کلمات در زبان طبیعی مانند انگلیسی می باشد و سیگنال , ترتیبی از داده های صوتی گرفته شده که دیجیتالی شده اند می باشد.

۷-٤-۱ پردازش سیگنال (Signal processing)

صحبت کردن یکی از اصلی ترین و عمده ترین روشهایی است که انسانها برای ایجاد ارتباط از آن استفاده می کنند.کامپیوترها در طول سالها پیوسته در حال افزایش توانایی محاسبه بوده اند و در کنار آن قابلیت های شناسایی کلام نیز افزایش پیدا کرده است.
کلام به دو دسته عمومی تقسیم می شود:
۱) تشخیص و شناسایی کلام
۲) فهم کلام

---

1 Speech Recognition

با توجه به هدف و مقصود این مبحث, از آنجایی که تشخیص کلام زیر مجموعه ای از فهم کلام نیز می باشد, فهمیدن کلام مورد مطالعه قرار خواهد گرفت .

تشخیص کلام عمل نگاشت از سیگنالهای صوتی دیجیتالی شده به رشته ای از کلمات می باشد.سیستم باید به سه سئوال زیر پاسخ دهد:

۱) گوینده با چه صدایی کلام یا سخن را ادا کرد؟

۲) گوینده چه کلماتی را با آن صدای سخن می خواست بیان کند؟

۳) گوینده چه مفهومی را می خواست با بیان آن کلمات برساند؟

برای پاسخ دادن به سئوال اول , ابتدا باید فرق بین صداهای معمولی و صدای سخن یا کلام را از هم تمیز داد. همه زبانهای بشری , ترکیبی از ٤٠ تا ٥٠ صوت متمایز می باشند که phone نامیده می شوند. Phone صوتی می باشد که متناظر با یک مصوت واحد یا حرف صامت واحداست . به هر حال قابل ذکر است ترکیب مشخصی از حروف می توانند صوت خاص مانند

(*th*) را تولید کنند و ترکیب حروف به خصوصی می تواند تعداد مختلفی صوت , بسته به مفهوم یا متن را تولید کنند.(a در cat و rat).

ابتدا باید تمامی اصوات ممکن ار مشخص کنیم, سپس آنها را به روشی خاص توصیف کنیم و بنابر آن می توانیم به آسانی در دیکشنری صداها به آنها رجوع کنیم.آسانترین و واضح ترین روش, تجزیه و تحلیل اصوات توسط تمیز دادن آنها به وسیله فرکانس یا دامنه نوسان یا ... است.

سپس لازم است این خصوصیات را در کتاب مرجع یا دیکشنری جایگذاری کرد که از آنجا می توانیم صداها را به درستی تشخیص دهیم.

بهترین روش مرتب کردن بر اساس تلفظ می باشد به دلیل اینکه spelling یا املاء دراین مقوله بی ربط و نامناسب می باشد.

سئوال بعدی این است که چگونه مشخص کنیم که گوینده چه کلماتی را می خواست به شنوندگان بگوید؟ با وجود اینکه ما یک لغت نامه یا دیکشنری بر اساس تلفظ مرتب شده داریم ولی هنوز مسائلی وجود دارد که باید حل شوند.یکی از مسائل

تصادف یا وقوع هم صداها (homophones) می باشد به این معنی که دو یا بیشتر از ۲ تا کلمه وجود دارند که تلفظ یکسانی دارند مانند their,they're,there,... .مسئله دیگر مشخص کردن رشته درستی از کلمات که هیچ شکافی بین آنها وجود ندارد است.بنابراین ما باید روشی را برای تعیین این که چه وقت کلمه ای به پایان می رسد و چه وقت کلمه ای دیگر آغاز می گردد, بدست آوریم.اگر نتوانیم کلمات را از یکدیگر تمیز دهیم , لازم است دوباره نگاهی به لغت نامه برای هر صوت بیندازیم به این دلیل که ممکن است هر صوت جدید آغازکننده کلمه دیگری باشد.

پاسخ به سئوال آخر هنوز کاملا" حل نشده است.معنی کلماتی که بیان می شود چیست؟

برای پاسخ به این سوال ابتدا باید مشخص کنیم کدام کلمه ها گفته شده اند, سپس به تحلیل گر هایی که سعی دارند معنی را به کمپیوتر انتقال دهند واگذار کنیم.

سیگنال صوتی خام که توسط میکروفون گرفته می شود , درآغاز دیجیتالی (digitized) و پیش پردازش می شوند.Diditization شامل نمونه براداری از سیگنالهای آنالوگ معمولا" بین 8,16 khz می باشد و سپس درجه بندی هر نقطه نمونه معمولا" از 8 تا 12 بیت ارزش برای هر نقطه است.در قدم بعدی زیر دنباله های روی هم افتاده و دارای اشتراک داده های دیجیتالی شده پردازش می شوند.زیر دنباله های به طول حدودا" 10msec (شامل ۸۰-۱۶۰ نقطه داده ای) برای تعریف ترتیبی از فریم ها استفاده می شود.در داخل هر فریم مجموعه ای از خصوصیات که بعدا" کشف خواهند شد, وجود دارد.برای مثال کل انرژی در یک فریم و اختلاف انرژی بین فریم جاری و فریم قبلی.

در یک فریم حدود ۸ تا ٤۰ ویژگی یافت خواهد شد.

این n-D بردار از ویژگی های خود بعدا" توسط پردازشی به نام Vector quantization, quantized یا درجه بندی می شود مثلا" 256 bin .

هر فریم یا چارچوب اکنون توسط یکی از ۲٥٦ برچسب ممکن توضیح داده می شود.نتیجه کل این پردازش توضیح فشرده ای از فضاهای روی هم انباشته از

سیگنالهای صوتی است که برای شناسایی کلمه کافی می باشد.
اساس رهیافت شناسایی و تشخیص کلام استفاده از قانون بیز (Bayes) می باشد که مسئله را به قسمت های قابل مدیریت می شکند.

$$P(words \mid signal) = P(words)\, P(signal \mid words) / P(signal)$$

از آنجایی که ما سیگنال های دیجیتالی از نوع توصیف شده در بالا داریم و هدف ما پیدا کردن ترتیبی از کلمات می باشد که P(words | signal) رابیشینه می کند, P(signal) ثابتی برای ورودی های صوتی معلوم می باشد بنابراین می توانیم به آسانی این مسئله را رها کنیم .بنابراین هدف جدید ما محاسبه مقدار زیر است:

$$Argmax_{words}\, P(signal \mid words)\, P(words)$$

P(words) نشان دهنده مدل زبان می باشد که تصریح کننده احتمالات قبلی از رشته کلمات خاصی می باشد.بنابراین باید تعین کننده میزان احتمال وقوع برخی کلمات انگلیسی باشد.

P (signal | words) مدل صوتی می باشد که مشخص کننده احتمالهایی از اصواتی از کلمات که بیان شده اند, می باشد.این قسمت توسط حقیقتی که اکثر کلمات به راهها و طرق مختلف تلفظ می شوند, پیچیده تر می شود.قسمت بعدی جزئیات بیشتری در مورد مدل زبان و مدل صوتی را بیان می کند.

٢-٤-٧ مدل زبان (The Language Model)

p(words) احتمالهای قبلی از ترتیبی از کلمات $words=w_1w_2\ldots w_n$. می باشد که محتمل تر در زبان طبیعی می باشند.مثلا" "I have agun" محتمل تر از "I have a gub" می باشد.

یک راه برای بیان احتمال های وابسته به استفاده از قانون های زنجیره ای به صورت زیر می باشد :

$$P(w_1,w_2\ldots..w_n) = P(w_1)\, P(w_2 \mid w_1) \ldots P(w_{n-1} \mid w_1\ldots.,w_{n-2})\, P(w_n \mid w_1,\ldots w_{n-1})$$

اکنون ما نیاز به محاسبه احتمال کلمه اول در جمله (w_1) و محاسبه احتمال کلمه دوم (w_2) در صورتی که کلمه اول در جمله باشد داریم. در انتها محاسبه احتمال کلمه w_n است در صورتی که n-1 کلمه قبلی شامل کلمات $w_1....w_{n-1}$ باشند.این نوع بیان بسیار پیچیده است به دلیل اینکه نیازمند این است که ما احتمالهای شرطی از ترتیبی از کلمات ممکن را مشخص کنیم .اگر سری کلمات شامل n تا کلمه و زبان ما شامل m تا کلمه باشد, سپس برای محاسبه $P(w_n \mid w_1,....w_{n-1})$
نیازمند محاسبه m^{n-1} حالت از ترتیب کلمات (شروع کننده جمله) ممکن می باشد.
به جای این کار می توانیم از فرضیه آسانی به نام First-order Markov استفاده کنیم به این صورت که احتمال وقوع یک کلمه فقط وابسته به کلمه قبلی خود می باشد.

$$P(w_n \mid w_1,...., w_{n-1}) =\approx P(w_n \mid w_{n-1})$$

با استفاده از این فرضیه اکنون ما بیان ساده تری برای محاسبه احتمالات متصل داریم.

$$P(w_1 w_2 .....w_n) = P(w_1)P(w_2 \mid w_1)....P(w_n \mid w_{n-1})$$

این مدل ساده شده مدل bigaram نامیده می شود به خاطر اینکه مربوط به زوج های متوالی از کلمات می باشد واین مدل کمترین مقدار متن برای تعیین احتمال هر کلمه در جمله را فراهم می سازد.واضح است استفاده بیشتر از متن, صحیح تر می باشد ولی برای محاسبه هزینه بیشتری می طلبد.(به عنوان مثال فرضیه second order Markov که در این صورت مدل trigram خواهیم داشت.)
می توانیم جدولی برای نمایش مدل bigram توسط محاسبه احتمالهایی از توالی زوج کلمه های ممکن در یک مجموعه بزرگ آزمایشی کلمات ایجاد کنیم.به عنوان مثال اگر حرف (a) ۱۰۰۰۰ بار در مجموعه آزمایشی ظاهر شده و (a) توسط کلمه (gun) ۳۷ بار دنبال شود بنابراین (gun | a) برابر با 37/10000 یا 0.0037 می باشد.
به جای نمایش مدل bigram در یک جدول می توانیم از آتاماتای احتمالی محدود(Probabilistic finite automata) استفاده کنیم.گره ای (حالت) برای

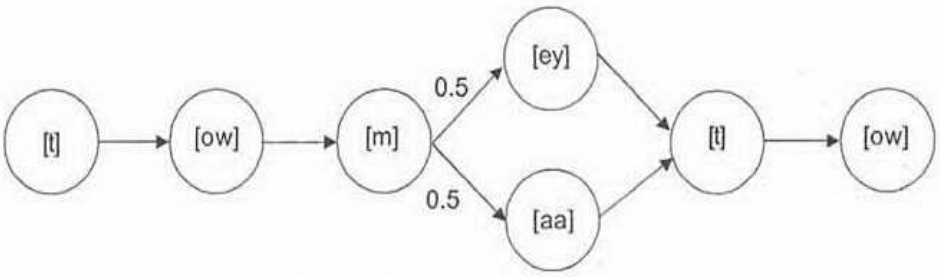
هر کلمه ممکن ایجاد کنید و کمانی از هر گره به همه گره های دیگر رسم کنید .هر کمان را با احتمالی که کلمه با گره منبع(مبداء) مربوط است و پیرو کلمه ای که با گره مقصد پیوسته است, ارزش گذاری یا برچسب گذاری کنید.در انتها گره ای به نام شروع یا start اضافه کنید و کمانی از آن به همه گره ها بکشید.این کمانها را با احتمالی که کلمه با گره مقصد در ارتباط است و می تواند یک جمله را شروع کند ارزش گذاری کنید.

ما می توانیم از این (FSM)Finite state Machine برای تعیین احتمال جمله ای که با گره start شروع می شود و به گره هایی که مربوط به کلمات متوالی در جمله می باشد و گذار پیدا می کند, استفاده کنیم.

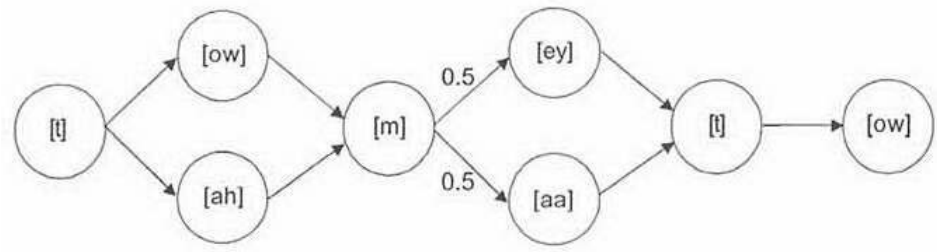
۳-٤-۷ مدل صوتی (The Acuostic Model P(signal | words)):

مدل صوتی به ما می گوید که چه صدایی پس از ادای کلمات شنیده خواهد شد.مدل Markov به ما تلفظ های متناوب کلمات را نشان می دهد .این تحول و گذار فقط مربوط به حالت جاری است نه به حالتهای قبلی .این مدل تمامی راههای ممکن را برای رسیدن به هدف, براساس حالت کنونی آن و نه حالتهای قبلی , نشان می دهد.

شکل شماره ۳-۷ مدل مارکوف



Co- articulation زمانی اتفاق می افتد که شخص آرام ی اتند حرف می زند و صداها را می آمیزد ومبهم می کند و اساسا" اندکی تلفظ آنها را عوض می کند .این به انسانهایی اشاره می کند که خیلی تند حرف می زنند و برخی کلمات را با هم می آمیزند یا مانند انسانهایی که لهجه دارند.



شکل شماره ٤-٧ co- articulation

مدل پنهانی Markov (The Hidden Markov model) سعی می کند به اختلاف در گفتار توجه کند و سعی می کند $P(words \mid phones)$ و $P(signal \mid phones)$ را کمینه کند همچنین به خطاهای ممکن یا تکه هایی از گفتار که در حالت تعلیق (hang) می باشد , توجه می کند.

فصل هشتم

سیستم خبره

اهداف

در پایان فصل، دانشجو با مفاهیم زیر آشنا می‌شود:

■ با برنامه نویسی مبتنی بر هوش مصنوعی که در سیستمهای مبتنی بر دانش تجلی یافته است آشنا میشویدو بر تفاوتهای آن با بر نامه نویسی معمولی پی میبرید.

■ تعاریفی از سیستمهای خبره و همینطور معماری و خصوصیات آنها را مشاهده میکنیدو مختصرا" در مورد تاریخچه آن اطلاع کسب میکنید

■ با مفهوم مهندسی دانش و دانش دامنه آشنا میشویدو بطور کامل در مورد مراحل اکتساب دانش آگاهی بدست می آورید.

■ در مورد رویه ی استنتاج در حساب گزاره ای و مسندی و در سیستمهای تولید مبتنی بر قانون خواهید خواند.

■ باابزارهای توسعه سیستمهای خبره که در دسترس قرار دارند مانند زبانهای الگوریتمیک و زبانهای سمبولیک آشنا خواهید شد.و در مورد زبانهایی مانند پرولوگ و لیسپ که به عنوان زبانهایی برای برنامه نویسی هوش مصنوعی هستند مطالبی خواهید خواند.

■ و در آخر چند نمونه از کاربردهای سیستمهای خبره مانند والیزاو مایسین و غیره راخواهید دید. PARRY

۸-۱ مقدمه

در سالهای اخیر تحقیقات در زمینه هوش مصنوعی , موفقیت های بزرگی را کسب نموده است. درمیان همه این موفقیت ها بارزترین آنها توسعه سیستم های کامپوتری قدرتمندی به نام سیستمهای خبره یا سیستمهای مبتنی بر دانش بوده است.برنامه هایی طراحی شده اند تا دانش واقعی در زمینه های خاص تخصصی را برای حل مسائل مورد استفاده قراردهند.به عنوان مثال تلاشهایی با همکاری متخصصین و توسعه دهندگان صورت گرفته است که نتیجه آن طراحی سیستمهای تشخیص بیماری, پیکربندی سیستمهای کامپوتری و اکتشاف کانیها و معادن با کیفیت اجرای برابر با کیفیت کار متخصیصین واقعی بوده است.

سمبولهایی که در سیستم مبتنی بردانش استفاده می شوند برای بازنمایی مجازی هر نوع شیء اعم از مصنوعات,انسانها, اجسام , ارگانهای زیستی, کلاسهای مختلف اشیا, مفاهیم و ....است.

مطلب قابل توجه در اینجا این است که ما قصد مشاهده و دستکاری سمبولها به عنوان چیزی غیر از اعداد محض را داریم. بنابراین ما به خصوصیا ت و تسهیلات زبانهای برنامه نویسی متفاوتی نیاز داریم.

از تفاوتهای دیگر بین برنامه نویسی الگوریتمیک معمول و برنامه نویسی مبتنی بر هوش مصنوعی- که در سیستمهای مبتنی بر دانش تجلی یافته است- می توان روش برنامه نویسی را نام برد.در برنامه نویسی معمولی که اغلب برنامه نویسی رویه ای نیز نامیده می شود, ما به کامپیوتر باید بگوییم که با داده هایی که به صورت متوالی وارد کامپیوتر می شود چه کار کند. بنابراین در برنامه نویسی رویه ای , رویه ها بازنمایی چگونگی انجام کاری هستند که ما می خواهیم.

از سوی دیگر , در برنامه نویسی مبتنی بر دانش , ما در حوزه خاصی, دانش را به صورت اخباری بازنمایی می کنیم. ما تنها آنچه را که می دانیم بازنمایی می کنیم بدون اینکه دقیقا" و به صورت پیشرفته به چگونگی استفاده از آن توجه کنیم.

جدول ۱-۸ تفاوتهای بین این دو نوع برنامه نویسی را خلاصه می کند.دو نکته در اینجا خارج از مبحث ما هستند.اول, سازگاری با دو حالتی بودن رویه ای در مقابل روش اخباری که توضیح داده شد، به این معنی که در برنامه نویسی رویه ای ما درباره تغییر و دستکاری داده ها صحبت می کنیم در حالی که در برنامه نویسی اخباری ما درمورد

بازنمایی دانش بحث می کنیم. نکته دوم و مهم تر صفت جداسازی دانش از کنترل است که مشخصه ای از برنامه نویسی مبتنی بر دانش است.

نرم افزار مبتنی بر دانش	برنامه نویسی معمول
• قابلیت اداره کردن داده های گم شده یا غیرقطعی • تصمیم گیری ترتیب وتوالی توسط موتور استنتاج • مناسب برای تغییر سیمبولها • واسط های زبان طبیعی • امکان توضیح در حین اجرا • سیستم خبره = مسئله + کنترل + داده	• نیازمند داده های صحیح است • ساختار رویه ای ثابت • مناسب برای پردازش عددی • فقط یک برنامه نویس آن را می فهمد • توضیح در حین اجرا غیر ممکن • برنامه = الگوریتم + داده

۸-۱-۱ تعاریف:

تعاریف مختلفی از سیستمهای خبره وجود دارد هرچند که بسیاری از آنها مشابه هستند.پرفسور فایگن بام از دانشگاه استنفورد یک سیستمهای خبره را اینگونه توصیف نموده است:

" یک برنامه کامپیوتری هوشمند است که از دانش و رویه های استنتاج برای حل مسائل دشواری که نیازمند کارشناسان خبره هستند, استفاده می کند .دانش مورد نیاز برای اجرا در چنین سطحی به همراه رویه های استنتاج مورد استفاده, می توانند به عنوان مدلی از متخصصین و کارشناسان در یک زمینه خاص در نظر گرفته شود."

یک تعریف ارائه شده که خیلی هم مورد استفاده قرار گرفته , تعریفی است که از سوی " گشینگ , ریبو و رایتر " ارائه شده است:

" سیستمهای خبره, برنامه های کامپیوتری دارای اثر متقابل هستند که عمل تلفیق قضاوت, قوانین , شهود و دیگر تخصص ها را برای تامین یک توصیه قابل درک و

زیرکانه در امور مختلف , انجام می دهند."
یک تعریف جامع تر از سیستمهای خبره را برکمن ارائه داد:
" یک سیستم خبره سیستمی است که دارای قوانین کارشناسانه است و از جستجوهای کورکورانه اجتناب می کند, با تغییر و دستکاری سیمبولها استدلال می کند, اصول اساسی و بنیادی در یک حوزه خاص را می فهمد و می یابد , متدهای ضعیف استدلال برای عقب نشینی در مواقعی که قوانین خبره جوابگو نیستند دارد .
با مسائل دشوار در زمینه های پیچیده سروکار دارد. می تواند تشریحی از یک مسئله به صورت عبارات غیر تخصصی گرفته و آنها را به بازنمایی داخلی مناسب برای پردازش با قوانین کارشناسی و تخصصی خود تبدیل کند. می تواند برای دانش درونی خود نیز استدلال کند, به خصوص برای بازسازی منطقی مسیرهای استنتاج برای توضیح و تفسیر و توجیه کردن خود."
سیستمهای مبتنی بر دانش سعی در یافتن تکنیک ها و روشهای ساخت سیستمهای انسان – ماشین با تخصص های خاص حل مسائل است.
تخصص شامل دانش دریک زمینه به خصوص , فهم وتشخیص مشکلات آن حوزه و مهارت در حل بعضی از آنهاست. معمولادانش در هر رشته تخصصی بر دو قسم است:

- دانش خصوصی
- دانش عمومی

۸-۱-۲ دانش عمومی:

دانش عمومی شامل توصیفات,حقایق و تئوری های منتشر شده است که در کتابهای درسی , روزنامه ها و مقالات تحقیقی غیره است.

۸-۱-۳ دانش خصوصی:

دانش عمومی تنها منبع متخصصین انسانی نمی باشد. متخصصان معمولا" دارای دانش خصوصی هستند.
دانش خصوصی به طور گسترده ای شامل قوانین thumb به نام کشف کنندگی هستند. کشف کنندگی ها متخصصین را قادر می سازد تادر صورت لزوم، حدس های آگاهانه بزنند, تشخیص رهیافت های محتمل برای حل یک مسئله وسروکار با داده های ناکامل و پر از خطا به صورت موثر است.

۸-۲ مهارت در مقابل دانش

مهارت در حل مسائل ، عموما دلالت بر سرعت, کارآیی , خطاهای کاهش یافته , بار ادراکی کاهش یافته, نیرومندی و غیره دارد .از سوی دیگر , دانش به انسان اجازه می دهد بواسطه قیاس, ادراک و شعور متعارف ,تجزیه و تحلیل و غیره به حل مسائل جدید بپردازد.حد اقل تا کنون مسائل دشوار و جالب توجه راه حل های الگوریتمیک نداشته اند. بسیاری از امور مهم درمتون و مفاهیم اجتماعی – فیزیکی رخ می دهند, که عموما" در مقابل توصیفات دقیق و تجزیه و تحلیل های سخت مقاومت می کنند .به عنوان مثال طرح ریزی استدلال های مشروع تشخیص های پزشکی, امکانات عیب شناسی ماشینهای تحلیل موقعیت نظامی و غیره.

انسانهای کارشناس به کارایی برجسته ای دست می یابند زیرا آنها مطلع و با تجربه اند. دلیل سوم برای متمرکز شدن بر دانش , تشخیص ارزش ذاتی خود آن است.دانش یک منبع کمیاب است که تکثیر و پالودگی ان , توانگری به همراه می آورد.دانش بشری توسط آموزش و کارآموزی انتقال و پخش میشود.

با استفاده از دانش , انسان ها به موفقیت های قابل توجهی در حل مسائل دشوار دست یافته اند.در صورتی که این دانش از متخصصین اقتباس شده و در یک برنامه کامپیوتری قرار گیردکه استفاده شود, سپس آن برنامه هم باید به سطح بالایی از کارایی نائل آید.اقتباس دانش از کارشناسان و قرار دادن آن در فرم های قابل محاسبه , هزینه های تولید مجدد دانش و بهره برداری از دانش را به نحو قابل توجهی خواهد کاست.همزمان , دانش خصوصی به واسطه قرار گرفتن در دسترس عموم برای ارزیابی و آزمایش می تواند باعث سرعت گرفتن پیشرفت علم شود.

در بیشتر موارد در روش های حل مسئله مبتنی بر دانش یا هوشمند , ما با مسائلی روبرو هستیم که راه حل های رسمی یا الگوریتمیک ندارند.بنابراین روشهای کشف کنندگی باید استفاده شوند .راه حل های موءثر بستگی به استفاده به هنگام و به جا از دانش برای شناسایی تصمیمات بالقوه ای دارند که می توانند نتایج محتمل و امیدبخش به بار آورند و اتنخاب های غیر ثمر بخش را حذف کنند.این ایده بنیادی از هر روش حل مسئله مبتنی بر دانش می تواند به صورت زیر خلاصه شود:

۱) دانش = حقایق + عقاید + کشف کنندگی

۲) موفقیت = یافتن یک روش حل مسئله خوب با در اختیار داشتن منابع در دسترس.

۳) بازده جستجو مستقیما" بر موفقیت تاثیر می گذارد.
٤) عوامل موثر بریک راه حل کارامد:

- دانش قابل اجرا , صحیح و قابل تفکیک و تمایز.
- حذف سریع دیدگاه های غیر ثمر بخش
- منابع دانش مضاعف مشترک
- تقسیم راه حل هل به سطوح متفاوتی از تجرد

۵) مشکلات اساسی در رهیافت مبتنی بر دانش:

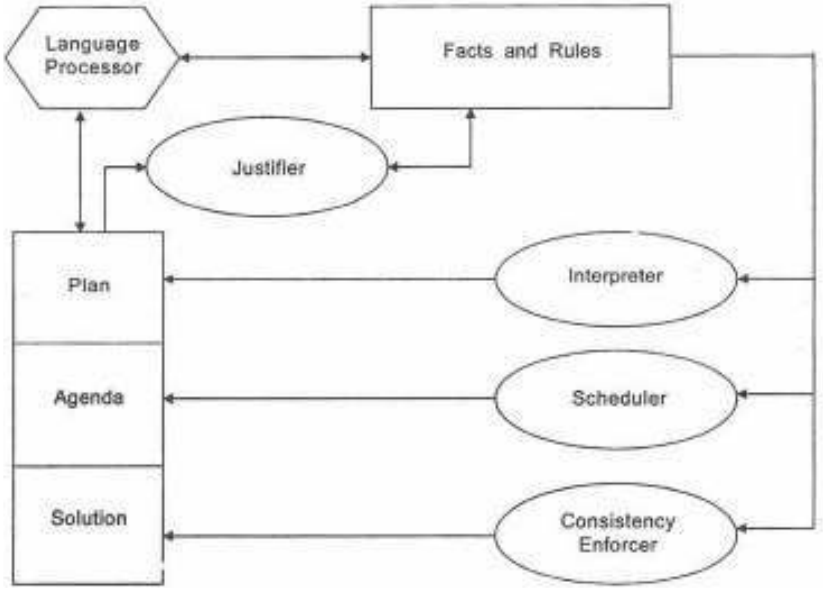
- دانش اشتباه یا خطا
- وجود امکان های مختلف برای ارزیابی کردن
- وجود رویه های پیچیده جهت حذف احتمالات
- مسائلی که به صورت پویا تغییر می کنند

بر اساس تجربه مان در حل مسائل هوشمند, ایده اساسی می تواند بر اساس موارد زیر تشریح شود:
۱) ساخت راه حلها به صورت کارامد وبه طور انتخابی از فضایی از پیشنهادات متناوب
۲) شناسایی راه حل های مفید و سپس کاوش بیشتر در آنها
۳) هرس کردن راه حل تا رسیدن به بهترین راه حل

یک حل کننده مسئله ایده آل باید موارد زیر را داشته باشد:
۱) پردازنده زبان برای مکالمات مبتنی بر مسائل
۲) چرکنویس برای ثبت نتایج میانی
۳) دانش در مورد یک حوزه که می تواند شامل حقایق , کشف کنندگی و عقاید باشد
٤) دانش برای چک کردن ثبات یک راه حل پدیدار شده
۵) دانش برای برنامه ریزی استراتژی راه حل مسئله بعدی
٦) دانش برای ارزیابی راه حل های جزئی

حل کننده های اولیه مسائل هوشمند به سمت سیستم های خبره رشد کردند.سیستم

های خبره , سیستم های مبتنی بر دانش ساده شده اند زیرا کل دانش حل مسئله در هر زمینه ای بسیار پیچیده است و نمی تواند توسط ساختارهای نحوی ساده تسخیر شود.بنابراین دامنه مسئله باید تنگ تر شود .و بنابراین همه اهداف کاربردی و عملی دانش محدود شده تا بتواند در زبانهای بازنمایی دانش ساده شده تسخیر شود.قبل از اینکه به جزئیات سیستم های خبره بپردازیم , شایسته است که ابتدا بتوانیم تفاوتهای بین برنامه های معمولی و سیستم های خبره را بشناسیم .این موضوع یک دورنما یا منظر متناوب برای توصیف سیستم های خبره به ما می دهد.



(شکل ۱–۸) معماری ایده آل سیستم خبره

۸–۲–۱ چگونه سیستم های خبره از برنامه های معمولی تمییز داده می شوند؟

اساسی ترین تفاوت میان این دو این است که سیستم های خبره دانش را تغییر می دهند در حالی که برنامه های معمولی داده ها را دستکاری می کنند.(جدول ۲–۸)

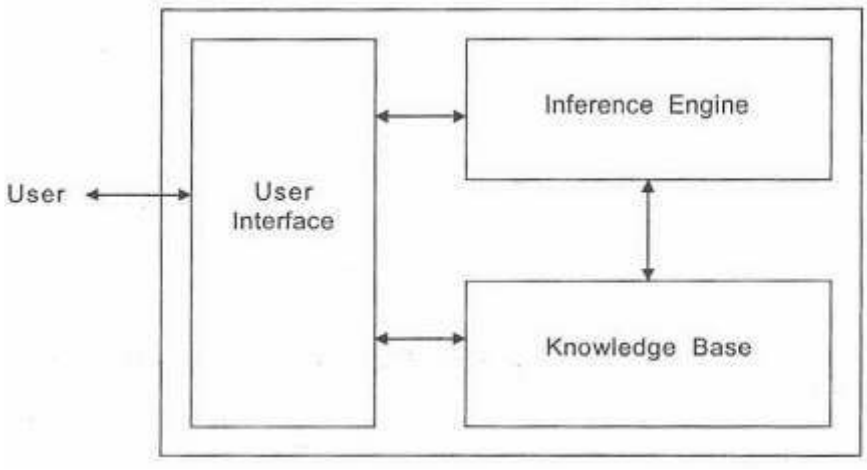
پردازش دانش	پردازش داده

• بازنمایی و استفاده داده + کنترل = دانش • کشف کنندگی • پردازش های استنتاجی • کنترل گسترده و مقدار کم داده با هم نگه داری می شوند.	• بازنمایی و استفاده از داده های استاتیک • الگوریتم ها • پردازش تکراری • برای کنترل و مقدار زیاد داده جداگانه نگه داشته می شود.

به هرحال تفاوتهای اساسی بین پایگاه های داده و پایگاه دانش به صورت زیر است (جدول ۳–۸):

پایگاه دانش	پایگاه داده
• اطلاعات در سطح بالاتری از تجرد قرار دارند. • بیشتر بر روی کلاسی از اشیا عملیات انجام می دهد تا یک شئ واحد • از قدرت استنتاجی بهره مند است • بازنمایی به وسیله منطق , قوانین یا فریم ها یا مستندات یا شبکه های معنایی انجام می شود. • مورد استفاده برای تحلیل داه ها و برنامه ریزی	• مجموعه ای از داده های نمایش دهنده حقایق • فقط بر روی یک شئ واحد عمل می کند • اطلاعات باید صراحتا" توصیف شده باشند • به صورت سلسله مراتبی یا ارتباطی یا براساس مدل شبکه نمایش داده می شود. • برای اهداف عملیاتی ابقا می شود

شکل ۲–۸ معماری یک سیستم خبره نوعی را نشان می دهد.



(شکل ۲-۸) ساختار یک سیستم خبره

۸-۳ خصوصیات اولیه یک سیستم خبره

یک سیستم خبره باید موارد زیر را داشته باشد:

۱) متخصصی که باید کارایی تخصصی و کارشناسانه داشته باشد.آنها باید صاحب درجه بالایی از مهارت باشند و به اندازه کافی قدرتمند باشند.

۲) به دلیل اینکه دانش آنها سمبولیک است , استدلال سمبولیک را اجرا کنند.

۳) آنها باید قادر باشند قوانین پیچیده را استفاده کنند و دامنه های مشکل مسائل را اداره کنند.

٤) self-knowledge آنها باید قادر به تست و امتحان کردن قدرت استدلال خود باشند و بتوانند عملیات خود را توضیح دهند.

میزان مفید بودن یک سیستم خبره مستقیما" وابسته به کیفیت دانش آماده شده توسط طراحان است.بنابراین بسیار مهم است که سیستم به صورت بازگشتی تصفیه و معتبر شود.

۸-٤ تاریخچه مختصری درباره سیستم های خبره

در کنفرانس معتبر و بین المللی در زمینه هوش مصنوعی در سال ۱۹۷۷ , پرفسور فایگن بام , در یک مقاله , کلید بینش یک سیستم خبره را ارائه داد:" قدرت یک سیستم خبره به دلیل اشتقاق از دانشی است که دارا می باشد نه از یک فرمالیسم خاص یا

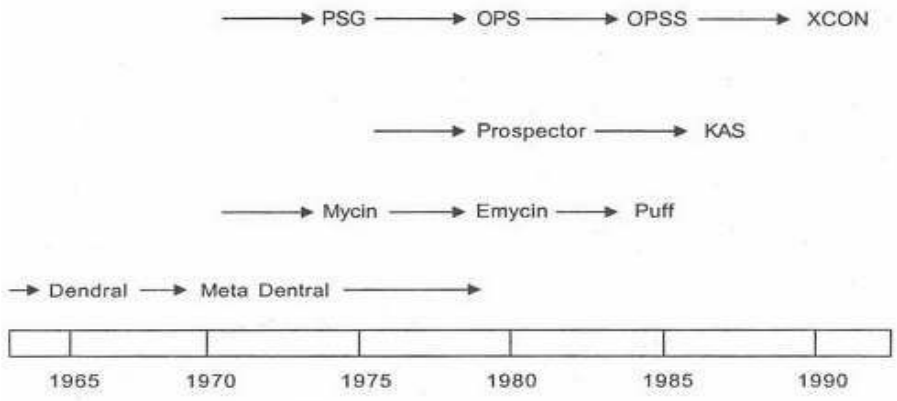
تمهیدات استنتاجی"
در سالهای نخست هوش مصنوعی , تصور می شد که همراه کردن چند قانون استدلال با یک کامپیوتر قدرتمند می تواند سیستم خبره ای تولید کند که قادر به اداره هر نوع مسئله ای در هر حوزه ای باشد.مانند (GPS=General Problem Solver) .با افزایش تجربه در این زمینه , قدرت شدیدا" محدود GPS، در نهایت منجر به درک این موضوع شد که GPS برای حل مسائل پیچیده بسیار ضعیف است. به همین دلیل کارشناسان نسبت به ساخت حل کننده های عمومی مسائل، بیشتر شروع به تفکر در زمینه مسائل با دامنه های محدودتری نمودند
در اواسط دهه ۷۰ , سیستمهای خبره متعددی پدیدار شدند.تعداد محدودی از محققان که نقش اصلی دانش در این سیستم ها را متوجه بودند شروع به تلاشهایی برای توسعه تئوری های جامع بازنمایی دانش و سیستم های چند منظوره نمودند.آنها نیز موفق نشدند به این دلیل که دانش نمی تواند به وسیله ساختارهای متناهی تسخیر شود زیرا که دانش بسیار متنوع و گسترده است.
از سوی دیگر رهیافت های مختلف دیگری برای بازنمایی دانش در آمدند که برای حل مسائل در زمینه های خاصی طراحی شده بودند.

۸-٤-۱ منظور ما از دانش در یک دامنه چیست؟

در یک نگاه مجرد وعمومی , دانش شامل توصیفات , ارتباطات و رویه ها در یک زمینه مورد علاقه است.
توصیفات در یک زمینه دانش که اشیا و کلاسها را شناسایی و تمییز می دهد , شامل جملات به یک زبان خاص است .زبان می تواند به صورت یک سیستم رسمی مانند منطق کلاسیک (مانند منطق حساب گزاره ای , حساب مسندی یا خبری) باشدطوری که هر موجودیت خوش ترکیب است و یک علامت تفکیک معنایی مشخص دارد یا زبان اصلاح شده همانطور که در منطق غیر رسمی وجود دارد مانند منطق قراردادی , منطق مدل , منطق غیر یکنواخت , منطق احتمالاتی و منطق شولا یا fuzzy است که همگی از منطق کلاسیک مشتق شده اند.همین طور به وسیله اضافه نمودن قوانین استدلالی خاص یا به وسیله پیوستن یک مدل کانونیکال.هر سیستمی دارای فواید و عیوب خاص خودش است و هیچ کدام از آنها برای بازنمایی کل دامنه دانش به اندازه

کافی مناسب نیستند.
همچنین ارتباطات خاصی میان این توصیفات در یک زمینه دانش وجود دارد .این موضوع وابستگی ها و تجمع میان موجودیت ها را بیان می کند .این ارتباطات نوعا" می توانند پیوستگی های رده بندی شده, تعریفی یا تجربی باشند. از سوی دیگر رویه ها , عملیات قابل اجرا بر روی این موجودیت ها را در هنگام حل یک مسئله مشخص می کنند.

در عمل، دانش در یک فرم ناگهانی پدیدار نمی شود بلکه به زیبایی با عناوین مجردی تطابق می یابد.دانش مانند یک ماده تصفیه نشده است .به عنوان مثال می تواند در حوزه علوم تجربی همانند تشخیص های پزشکی , زمین شناسی و غیره باشد.جایی که برای یک داده مشاهده شده می تواند دلایل زیادی وجود داشته باشد. همچنین می تواند به فرم کشف کنندگی (اکتشافی) , محدودیت دار و تنظیمی باشد.
هنر جمع آوری و پردازش دانش , مهندسی دانش نامیده می شود. ارزیابی یک سیستم خبره در شکل ۳-۸ آمده است.



(شکل ۳-۸) توسعه در سیستم های خبره

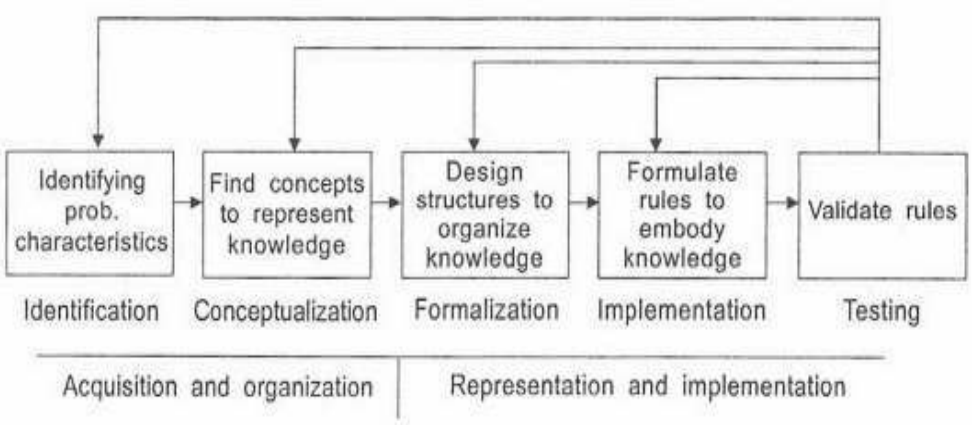
۸-۵ مهندسی دانش

مهندسی دانش یک نام بی مسمی است و باید به هنر یا مهارت دانش تغییر نام دهد.
آن به صورت گسترده شامل:
شناسایی و ادراک مسئله است . ادراک به معنی شناسایی مفاهیم کلیدی , رابطه ها ,

رویه ها و موارد ویژه است.
رسمی سازی , اجرا و آزمایش است.

۸-۵-۱ مراحل اکتساب دانش

حصول دانش از منابع مختلفی از قبیل کارشناسان , کتاب ها و غیره است که در آن مهندس دانش به واسطه چندین مرحله و طبقه قبل از تولید و سیستم های خبره شرح داده شده در شکل ٤-٨ پیش می رود.این مراحل حقیقتا" یک توصیف درشت از یک فعالیت پیچیده و ساختار ضعیف است که در طول اکتساب دانش اتفاق می افتد



(شکل ٤-٨) مراحل اکتساب دانش

آنها از یک موقعیت منحصر به فرد به یک موقعیت دیگر فرق می کنند.فعالیت ها به صورت گسترده می تواند به صورت زیر رده بندی شوند:

مرحله شناسایی:

شناسایی شرکت کنندگان و نقش آنها : قبل از اینکه اکتساب و استفاده از دانش آغاز گردد, ابتدا باید شرکت کنندگان و نقش هر یک از آنها انتخاب و شرح داده شود .متداول ترین سناریو شامل عمل متقابل بین یک کارشناس در یک زمینه خاص و یک مهندس دانش است.کارشناس حوزه یا دامنه به عنوان یک فرد مطلع عمل می کند که به مهندس دانش در مورد دانش و تخصص اش توضیح می دهد. فرایند اکتساب واستفاده از دانش می تواند شامل شرکت کنندگان دیگری نیز باشد.ممکن است چندین کارشناس دامنه یا حوزه مختلف , چندین مهندس دانش و حتی چندین کارشناس

انضباطی در آنجا حضور داشته باشند.

شناسایی مسئله:

زمانی که شرکت کنندگان انتخاب شدند, مهندس دانش و کارشناس حوزه می توانند به شناسایی مسئله مورد نظر بپردازند.این کار شامل یک مبادله غیر رسمی دیدگاهها از منظرهای متفاوت یک مسئله است.به عنوان مثال توصیفات آن , خصوصیات و زیرمسئله هاست .به منظور شناسایی یک مسئله ، پاسخگویی به سؤالات زیر مهم است:

- از سیستم های خبره انتظار حل مسائل در چه کلاسی می رود؟
- این مسائل چگونه تشریح و توصیف خواهند شد؟
- زیر مسئله های مهم و جزءبندی های وظائف کدام ها هستند؟
- داده های در دسترس کدامند؟
- مفاهیم مهم و روابط مشترک درونی کدام ها هستند؟
- چه نوع راه حل هایی مورد نیاز هستند؟
- متخصصین انسانی در چه حوزه هایی مورد نیاز هستند؟

شناسایی منابع:

منابع به منظور اکتساب دانش, پیاده سازی سیستم و همچنین آزمایش آن مورد نیاز هستند.

منابع نمونه ای شامل منابع دانش, زمان , امکانات محاسبه و پول هستند.

شناسایی اهداف:

احتمالا" کارشناسان دامنه , شناسایی اهداف را در ساخت یک سیستم خبره در حین شناسایی مسئله انجام می دهند.به هر حال مفید خواهد بود اگر اهداف را از وظایف خاص جدا کنیم.به این دلیل که آنها محدودیتها و قیود اضافی تشکیل می دهند که می توانند برای توصیف شرایط مطلوب وامکان و شدنی بودن یک رهیافت خاص مفید باشند.

مرحله ادراک:

در این مرحله مفاهیم کلیدی و روابط شناسایی شده در مراحل قبلی , به صورت واضح تری توصیف می شوند.سئوالاتی که در این مرحله باید پاسخ گفته شود عبارتند از:

- چه نوع داده هایی در این مرحله در دسترس هستند؟
- چه چیزی داده شده و چه چیزی باید استنتاج شود؟
- آیا زیر وظایف دارای نام خاصی هستند؟
- آیا استراتژی ها دارای نام خاصی هستند؟
- آیا فرضیه های جزئی قابل شناسایی وجود دارند که به طور متداول استفاده می شوند؟اگر وجود دارند کدام ها هستند؟
- آیا می توانید مفاهیم و رابطه ها را به صورت دیاگرام نمایش دهید؟
- محدودیت های موجود در این پروسه ها کدامند؟
- الگوی جریان اطلاعات چیست؟

این مرحله همانند بقیه مراحل شامل تراکنش های تکراری بین مهندسی دانش و کارشناس دامنه است.

مرحله رسمی سازی:

فرایند رسمی سازی شامل نگاشت مفاهیم کلیدی , زیرمسائل , خصوصیات جریان اطلاعاتی در حین مرحله ادراک به حالت نمایش و بازنمایی رسمی تری بر اساس ابزارهای متنوع مهندسی دانش یا چارچوب ها می باشد.مهندس دانش اکنون نقش غالب را در طراحی سیستم خبره بر عهده دارد.در این مرحله, مهندس دانش باید یک پوسته مناسب را شناسایی کند که برای مسئله ی در دست مناسبترین باشد .قالب نمایش دانش , انواع داده های آماده , استراتژی استنتاج و غیره باید کاملا" با خصوصیات مسئله منطبق باشد.سه عامل مهم در فرایند رسمی سازی عبارتند از : ۱) فضای فرضیه ۲) مدل اصولی فرایند ۳) خصوصیات داده ها . برای فهم ساختار فضای فرضیه , مفاهیم باید رسمی سازی شده و در مورد چگونگی اتصال آنها به فرم فرضیه یا شناسایی زنجیره قوانین تصمیم گیری انجام شود .ساختار تک تک مفاهیم باید قطعی باشند.

سئوالاتی که باید به آنها در این مرحله پاسخ داده شود عبارتند از:

- آیا مفاهیم , اشیا ساخت یافته هستند یا موجودیت های ابتدایی می باشند؟
- آیا روابط سببی یا روابط وابسته به فضا و زمان میان مفاهیم مهم است؟
- آیا مفاهیم و فضای فرضیه، متناهی هستند یا خیر؟
- آیا تردیدها یا عناصر قابل داوری مرتبط با فرضیه های نهایی یا میانی وجود دارند؟
- آیا سلسله مراتب فرضیه نمایش داده می شوند یا خیر؟(آیا سطوح چندگانه از تجرد مورد نیاز است؟)
- آیا نوع پردازش کاملا" وابسته به داوری و قضاوت است یا وابسته به ریاضی و قضاوت است؟
- مدل داده ها به پاسخ سئوالات زیر بستگی دارد

- آیا داده های در دسترس کم و ناکافی یا فراوان هستند؟
- آیا تردیدهای وابسته وجود دارند؟
- آیا تفسیر منطقی داده ها به مرتبه رخداد آنها در طول زمان بستگی دارد؟
- آیا داده ها به اندازه کافی سازگار و کامل برای حل مسائل هستند؟

مرحله پیاده سازی و اجرا:

مرحله پیاده سازی و اجرا شامل نگاشت دانش رسمی سازی شده از مرحله قبلی به چارچوب بازنمایی وابسته به ابزار منتخب برای مسئله است. به دلیل اینکه دانش در این چارچوب سازگار , قابل قیاس و سازمان یافته برای تشریح اطلاعات خاص و کنترل جریان است, بنابراین این دانش, یک برنامه قابل اجرا می باشد.حوزه ی دانش , به همراه توسعه ساختارهای داده , قوانین استنتاج و استراتژی کنترل بسیار آشکار و روشن می شود.توسعه نمونه اولیه سیستم در ساخت یک سیستم خبره بی نهایت مهم می باشد.

مرحله تست:

مرحله تست شامل ارزیابی نمونه اولیه سیستم و فرم های بازنمایی برای پیاده سازی آن

است. هنگامی که سیستم نمونه اولیه دو یا چند بار از ابتدا تا انتها اجرا شد, سپس باید با نمونه های مسائل دنیای واقعی آزمایش شود تا نقاط ضعف موجود در اصول دانش و استراتژی استنتاج آن مشخص شود.
اجرا و کارایی پایین می تواند به دلیل:

- مشخصه های ورودی/خروجی که به اکتساب و استفاده داده و نمایش نتیجه و استنتاج رجوع می کند.
- قوانین استنتاج , واضح ترین مکان برای جستجوی خطاها هستند.قوانین ممکن است نا صحیح , ناسازگار و ناکامل و یا ناپیدا باشند
- استراتژی کنترل به عنوان مثال , توالی قوانین
- انتخاب نمونه ها می توانند بسیار مشابه باشند و بنابراین امکان دارد نتوانند همه جنبه های یک سیستم خبره را تست کنند.

۸-٦ استنتاج

نوع رویه استنتاج مورد استفاده در طراحی یک سیستم خبره بستگی به الگوی بازنمایی دانش مورد استفاده دارد.زبا ن های رسمی زیادی برای بازنمایی دانش با موجودیت های خوش تعریف و روابط بین موجودیت های تسخیر شده با استفاده از فرمول های خوش فرم وجود دارند.اینها دارای علائم معناشناسی واضحی هستند واین زبان ها به صورت گسترده ای در مواردی که دانش مبهم یا بدون ثبات است استفاده می شوند. نمونه ها , منطق رسمی به نام منطق گزاره ای , منطق مسندی و غیره هستند. متد های مورد استفاده برای استنتاج منطقی هستند.منظور از منطقی بودن این است که : اجازه دهید A مجموعه ای از اصول عمومی (قضایا) یا حقایق وR مجموعه ای از قوانین استنتاج باشد .اگر یک تئوری در منطق وجود داشته باشد که درست باشد, آنگاه ما می توانیم اثباتی را بیابیم که چیزی جز یک سری از قوانین نیست که زمانی که در یک ترتیب مناسب بر مجموعه A بکار برده شود, به یک تئوری منتج خواهد شد.

۸-٦-۱ رویه استنتاج در حساب گزاره ای:

اگر حساب گزاره ای برای بازنمایی حوزه ی دانش استفاده شود, آنگاه مسئله ی استنتاج به صورت زیرمطرح میشود:R یک مجموعه از قوانین در فرم شرطی است و G

مجموعه ای از اهداف و فرضیه های قابل اثبات است:

R= { r_i/r_i is a rule in clausal form for 1<= i<= N}
. r_i= VL_i^j , 1<= j<= P , where literal L_i^j is a posited or negated atomic variable in propositional calculus.

سپس مسئله استنتاج برای یک تئوری داده شده $g \in G$,نوشته می شود به صورت زیر:

قانع کننده $R \wedge O$

O مجموعه ای از مشاهدات انجام شده است.

O ={ O ; where O is literal whose value is true}

با بسط دادن فرمول خواهیم داشت:

$$(r_1 \wedge r_2 \wedge ..... \wedge r_n) \wedge O_1 \wedge O_2 \wedge ..... \wedge O_k \wedge g_i$$

هدف ما یافتن مقادیر و ارزش ها برای تمام متغیر های موجود درR جایی که 1< I<K است, میباشد به طوری که تساوی بالا درست باشد.رویه استنتاج خیلی آسانتر خواهد شد در صورتی که ما مسئله را توسط ابطال و تکذیب ثابت کنیم.ما در اینجا مسئله ای را به صورت زیر مطرح می کنیم:

$$F = (R \wedge O \wedge \rightarrow 9)$$

.که برای همه بردارهای تخصیص ممکن , نادرست است .اگر F قانع کننده نیست آنگاه g یک معنی و مفهوم است در غیر این صورت g یک مفهوم نیست .این فرایند برای همه $g \in G$ می تواند تکرار شود.الگوریتم ها برای استنتاج در منطق گزاره ای عبارتند از:

۱) رزولوشن واحد با کشف کنندگی های شکاف دهنده
۲) رزولوشن با دلالت کننده ها

۸-٦-۲ رویه استنتاج در حساب مسندی:

در اینجا نیز استنتاج مبتنی بر ابطال و تکذیب است.منطق معتبر است ولی غیر قابل تصمیم گیری است

اگر یک مجموعه از قوانین wff داشته باشیم می توانیم مجموعه ای از قوانین استنتاج را به ترتیبی به کار ببندیم به طوری که بتوان قضایای درست را از آن مشتق کرد.ولی با داشتن یک قضیه درست و تعریف شده , لازم نیست که ما یک توالی از قوانین را که می توانند قضیه را از فرض های قبلی و اولیه اثبات کنند , بدانیم.در این موارد تضمینی وجود ندارد که الگوریتم استنتاج متوقف شود.

۸-٦-۳ رویه استنتاج در سیستمهای تولید مبتنی بر قانون:

در ادبیات سیستم های خبره اصطلاح قانون, معنای بسیار محدود و باریکتری نسبت به دیگر زبان ها دارد.این اصطلاح به معروفترین نوع تکنیک بازنمایی دانش به نام بازنمایی مبتنی بر قانون اشاره دارد.قوانین یک روش رسمی برای بازنمایی نظریه ها, رهنمود ها یا استراتژی ها ارائه می دهند.آنها معمولا" برای زمانی که دانش دامنه حاصل نتایج علوم تجربی توسعه یافته در طی سالها تجربه است, مناسب می باشند. قوانین معمولا" به صورت حالات if-then بیان می شوند.به طور مثال:

IF oil fire THEN use foam fire extinguishers

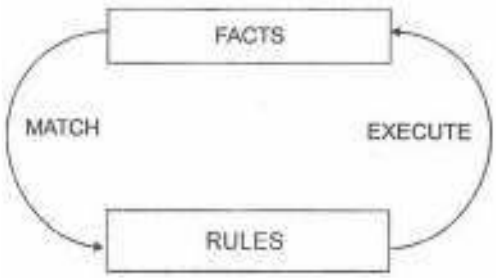
اگر نفت آتش گرفت **آنگاه** از آتش خاموش کن کفی استفاده کن

IF wood fire THEN use water

اگر چوب آتش گرفت **آنگاه** از آب استفاده کن.

شکل ۸-۵ .در سیستم های خبره مبتنی بر قانون ,دانش دامنه توسط مجموعه ای از قوانین که در مقابل مجموعه ای از حقایق موجود در وضعیت جاری تطبیق می کنند, بازنمایی می شود.

هنگامی که قسمت IF یک قانون به وسیله این حقایق برقرار باشد, عمل موردنظر در قسمتTHEN اجرا می شود.زمانی که این اتفاق می افتد, اصطلاحا" می گوییم قانون باید شلیک شود. سپس مجموعه ای از حقایق به وسیله تکمیل آنها با حقایق موجود در قسمت THEN به روز در می آید.



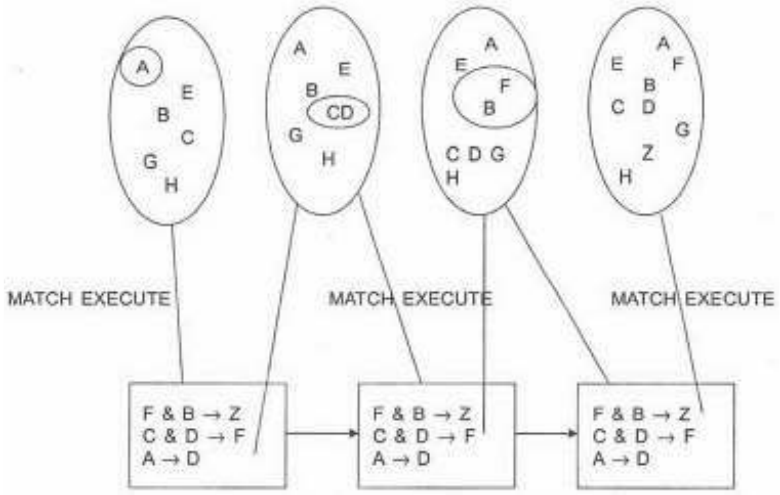
(شکل ۵-۸) الگوی استنتاج در یک سیستم خبره مبتنی بر قانون

این حقایق تازه اضافه شده به پایگاه دانش , خود می توانند برای تطابق با قوانین قسمت IF استفاده شوند.این عمل تطابق و اجرا , زنجیره های استنتاج را شکل می دهد.

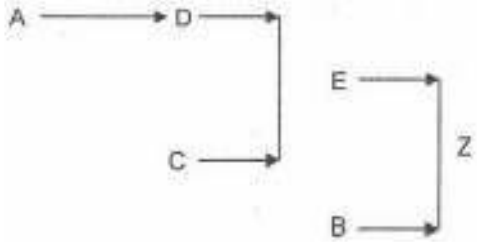
دو روش مهم وجود دارند که قوانین در آنها می توانند برای تطابق و اجرا از پایگاه دانش انتخاب شوند. رهیافت اول: **زنجیره سازی رو به جلو** و دوم: **زنجیره سازی رو به عقب** نامیده می شوند.

۸-٦-٤ زنجیره سازی رو به جلو:

شکل ۸-٦ مکانیسم زنجیره سازی رو به جلو را نمایش می دهد.(شکل ٦-۸) مکانیزم زنجیره سازی رو به جلو



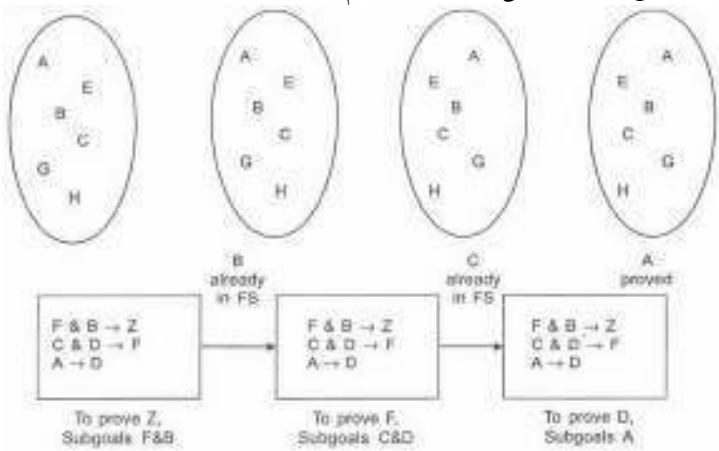
اولین قانونی که شلیک می شود $A \rightarrow D$ است زیرا که A هم اکنون در مجموعه حقایق قرار دارد.به عنوان یک نتیجه منطقی , وجود D استنتاج شده است و D در مجموعه حقایق قرار داده می شود.این عمل باعث می شود تا قانون دوم شلیک شود و تا به آخر تا اینکه Z نیز برقرار شود.این رویه زنجیره سازی رو به جلو نامیده می شود.زنجیره استنتاج تولید شده در شکل ۷–۸ نشان داده شده است.



(شکل ۷–۸) زنجیره استنتاج در زنجیره سازی رو به جلو

۸–٦–۵ زنجیره سازی رو به عقب:

زنجیره سازی رو به عقب استدلال در جهت عکس منطق را دنبال می کند ما در اینجا فرض می کنیم هدفی داریم که درست است, سپس سعی می کنیم تا آن را توسط گرفتن مقادیر مقدم یا پیشین , ثابت کنیم .این کار به وسیله تبدیل هر داده یا اطلاع به یک زیر هدف جدید انجام می شود. این فرایند تا زمانی که همه حقایق در مجموعه حقایق قرار گیرند, ادامه می یابد. شکل ۸–۸ مکانیسم زنجیره سازی رو به عقب را نشان می دهد.(شکل ۸–۸) مکانیزم زنجیره سازی رو به عقب.



۸–٦–٦ متد استنتاج در دیگر الگوهای بازنمایی:

برای طراحی سیستم های خبره , الگوهای دیگر بازنمایی دانش نیز وجود دارند.ما در اینجا قصد بررسی جزئیات رویه های استنتاج توسط آن الگوها را نداریم.

۸-۷ روش شناسی یا متدولوژی برنامه نویسی

یک دیدگاه مفید تر و مجرد تر از سیستم های خبره این است که متدولوژی برنامه نویسی بر جداسازی چیزی که در دنیا درست است از چگونگی بکارگیری دانش برای حل مسئله تاکید دارد.در این دیدگاه یک سیستم خبره شامل دو مفهوم زیر است:

- دانش دامنه یا حوزه

متدهای حل مسئله

در موارد ساده تر , دانش دامنه عبارتست از مجموعه ای از حقایق و متدهای حل مسئله که بیشتر شامل مکانیسم های استدلال همه منظوره است . به عنوان یک مثال واقعی در مورد اینکه چگونه می توان دانش دامنه را از متدهای حل مسئله تمییز داد به نمونه ی ساده شده از یک تشخیص پزشکی دقت کنید:

ما با این تفکر شروع می کنیم که:" اگر گیتا تب دارد , پس گیتا عفونت دارد." که بیشتر یک متد حل مسئله خاص ویژه برای تشخیص علت تب گیتاست.با استفاده از روش برنامه نویسی سنتی با مفهوم متغیرها , می توانیم مثال را به این صورت تعمیم دهیم:

متد	حقایق
اگر یک بیمار تب داشت, پس بیمار عفونت دارد.	گیتا بیمار است.

که ستون سمت چپ حقایق و ستون سمت راست متد عمومی تر را بیان می کند.

یک قانون عمومی تر که شایسته نامیدن به نام متد حل مسئله است , توسط مجرد سازی بیشتراین مورد بدست می آید.

متد	حقایق	اگر

بیمارعلائم یک بیماری را داشته باشد گیتا مریض است.
پس بیمار آن بیماری را دارد تب یک علامت بیماری است
عفونت یک بیماری است
تب علامت وجود عفونت است.

یک تعمیم نهایی ما را به نتیجه ای این چنینی می رساند:

متد حقایق
اگر شئ یک صفت از کلاس را نمایش می دهد گیتا مریض است
سپس آن شئ متعلق یه آن کلاس است تب یک علامت بیماری است
عفونت یک
بیماری است
تب علامت وجود عفونت است
بیمار یک شئ است
یک علامت بیماری یک صفت است
یک بیماری یک کلاس است.

.

در این دیدگاه , سیستم های خبره، متدولوژی برنامه نویسی ای را فراهم می کنند که وضعیت های حقیقی و واقعی را از متدهایی که چگونگی بکارگیری این حقایق را مشخص می سازند, جدا می کند.در این متدولوژی , انعطاف پذیری برای داشتن هر دو متد ها و حقایق ویژه و بسیار عمومی وجود دارد. در واقع اگر یک دامنه خودش را به استدلال توسط کد گذاری یک روش خاص برای مقابله با مسائل ویژه , متوجه می سازد , سپس ممکن است که ما برای هر یک از آنها چندین متد تک منظوره (منظور خاص) انتظار داشته باشیم. جایی که هر مولفه احتمالا" به متدهای تست ویژه نیاز دارد. به هر حال , در صورت امکان , متدلوژی به توسعه متدهای حل مسئله عمومی تر, به همان صورتی که در مثال بالا آمده است , پیش میرود

۸-۸ سیستم های خبره – ابزار

سیستم های هوش مصنوعی , به صورت متمرکز به بازنمایی و دستکاری دانش اهمیت می دهند. بعضی از دانش ها , به خصوص ریاضیات و علوم به صورت اعداد و فرمولهایی بیان می شوند که شامل مجموعه ای از ارقام و عملیات حساب هستند.بیشتر تلاش انسان , به هر حال نیازمند بیان در یک زبان عمومی تر و قدرتمندتر است.حتی ریاضیات , زمانی که شامل پردازش های استدلال مجردتر (اثبات قضایا- دگرگونی جبری- و راه حل های سیمبولیک برای تساوی های دیفرانسیل – انتگرالی) است که نیازمند زبان های قدرتمندتر و عمومی تر هستند. این زبان ها توسط مفاهیم و روابط بازنمایی شده به وسیله سیمبولها و رشته هایی از سیمبولها , بیان می شوند.

در واقع , اعداد و فرمولها یک مجموعه ازسمبولهای خاص هستند. اعداد سیمبولهایی هستند که خواص آنها بر روی مجموعه ای از عملیات حسابی تعریف شده اند. این عملیات حسابی توسط سیمبولها ورشته هایی از سمبولها مانند (- و+ و* /) بازنمایی شده است.به هر حال , ابزار هوش مصنوعی عبارتست از زبان ها , پردازش ها و ساختارهایی که اجازه فراگیری یا اکتساب , بازنمایی , ذخیره و انتقال و دیگر تغییرات مفاهیم و رابطه ها توسط ماشین های پردازش اطلاعات را می دهند. در این رابطه , زمینه هوش مصنوعی به صورت خیلی نزدیک بستگی به مطالعه تئوری زبان دارد که شامل زبان های کامپیوتری سطح بالا و تئوری کامپایلر کامپیوتر است.

.

تعدادی از فعالیتها وجود دارند که بر توسعه یک سیستم خبره مقدم هستند.اینها شامل شناسایی دامنه مسئله , یافتن تخصص و انتخاب ابزار می باشند.در اصل ٤ نوع ابزارهای توسعه در دسترس هستند که در زیر لیست شده اند:

- زبان های الگوریتمیک (مانند C,Pascal, Bassic)
- زبان های سیمبولیک (مانند لیسپ و پرولوگ)
- محیط های توسعه (مانند Art,KEE,LOOPS)
- بدنه ساختمان سیستم های خبره (مانند

(Crystal,XpertRule,Leonardo,Xi-Plus

۸-۸-۱ زبان های الگوریتمیک:

.

در کل زبانها می توانند تعریف شوند به عنوان:

- انعطاف پذیر و قدرتمند
- آنها می توانند به عنوان یک خیاط برای یک سیستم دقیقا" مطابق با کاربرد در نظر گرفته شوند.
- فاقد چارچوب مهندسی دانش

زبان ها می توانند به دو دسته قراردادی یا AI رده بندی شوند.البته با کمی اشتراکات میان آن دو.

زبان های قرار دادی می توانند به صورت رویه ای در طبیعت توصیف شوند که برای کار بر اساس الگوریتم , جایی که یک وظیفه به تشخیص قدم به قدم شکسته شده و سپس کدبندی می شود, طراحی شده است.

معمولا" ساختارهای داده ای پیچیده می توانند از انواع داده ای اولیه تشکیل شوند, و انها دارای ساختارهای کنترل قدرتمندی هستند .در زمینه توسعه سیستم های خبره , آنها معمولا" به عنوان زبان های پیاده سازی و اجرا (یا تحویل) برای سیستم های تولید, عمل می کنند. یک سیستم خبره , با استفاده از زبان هوش مصنوعی , ساختمان بدنه یا ابزار توسعه داده می شود و به یک زبان قرار دادی پرسرعت تر (معمولا" C) زمانی که آن به صورت راضی کننده ای عمل می کند, ترجمه می شود.

.

طراحان نیاز دارند تا از کارهای داخلی موتور استنتاج مطلع باشند هرچند که زبانهای شی گرا (مانند C++) به وسیله بکارگیری زبان های برنامه نویسی قرار دادی , توسعه ساختارهای استنتاج را آسانتر کرده اند. فایده این نوع سیستم ها این است که آنها معمولا" برای مسائل الگوریتمیک محاسبه عددی بیشتر کاربرد دارند تا پردازش های سمبولیک و اینکه ساختارهای واضح آماده برای اجرای سیستم های خبره وجود ندارد.

.

۸-۸-۲ زبان های سمبولیک:

دانش بشری مفهومی پویاست و هر تلاشی برای بازنمایی آن باید شامل ساختارهای قابل بسط دانش باشد . زبان های کامپیوتری برای برنامه نویسی منطق باید ساختارهایی برای ذخیره و بازیابی حقایق شناخته شده یا استنباط شده از پایگاه دانش یا پایگاه حقایق داشته باشند , همچنین برای استنباط حقایق جدید، باید رویه ها و توابعی داشته

باشند.
به عنوان یک قاعده کلی , ما این کارها را توسط زبانهایی مانند فرترن یا پاسکال می توانیم انجام دهیم . با این وجود عملیاتی نظیر پردازش لیست که برای اجرای استنتاج منطقی مفید هستند, برای پیاده سازی در زبان های رویه ای بسیار کم بازده و دشوار هستند . بنابراین نیاز برای زبان های هوش مصنوعی تخصصی که تمایل به ساختارهای لیستی دارند, می توانند گسترش یافته یا به صورت دلخواه ترکیب شده یا جدا شده باشند . استفاده از یک زبان خام هوش مصنوعی به پیاده سازی آن اجازه منعطف بودن می دهد اما تلاش بیشتری برای ایجاد امکاناتی نظیر واسط کاربر (که بدون آن امکان دارد زبان مناسب به نظر نرسد) می طلبد.

.
.

لیسپ:
لیسپ یکی از زبان های کامپیوتری رایج برای برنامه نویسی هوش مصنوعی است.این زبان برای حمایت از دستکاری های سمبولیک و تقابلی و روش برنامه نویسی آزمون و خطای مورد استفاده در بیشتر تحقیقات هوش مصنوعی است .البته لیسپ تنها زبان موجودی نیست که می تواند برای کاربرد های هوش مصنوعی در کامپیوتر استفاده شود. به عنوان یک قاعده کلی , کاربردهای این چنینی می توانند در ماشین توسط زبان اسمبلی برنامه نویسی شوند.لیسپ برای استفاده راضی کننده تر است به خصوص با معرفی کامپیوترها و کامپایلر های اصلاح شده لیسپ در قیاس با زبان های کامپیوتری دیگر , این زبان از لحاظ کارامدی بسیار بهتر است.
لیسپ توسط جان مک کارتی در اواخر دهه ۵۰ به عنوان یک زبان برای کاربردهای هوش مصنوعی توسعه یافت .لیسپ مترادف کلمه پردازنده لیست (LISt Processor) است.زبان های رویه ای مانند پاسکال یا C دارای عملگرهای ابتدایی تری برای اجرای محاسبات جبری توسط فرمول های حاوی سیمبولهای عددی صحیح و اعشاری هستند.
علاوه بر این موارد , زبان لیسپ دارای مجموعه ای از عملگرهای اولیه است که آن را قادر به انجام انواع مختلفی از استنباط ها با جملات (لیست ها) که شامل کلمات (

.

رشته ای دلخواه از کاراکترها) هستند و نمایش گزاره ها و آرگومان هایشان , می کند..

پرولوگ:

پرولوگ به عنوان یک زبان جایگزین برای برنامه نویسی هوش مصنوعی توسط آلن کلمرار و همکارانش در مارسل در اوایل دهد ۷۰ توسعه یافت. همانند لیسپ , پرولوگ هم استانداردهای مختلفی پیدا کرد ولی استاندارد قطعی و نهایی آن امروزه به صورت مستدل ،استاندارد پرولوگ ادینبورگ است. این استاندارد در کشورهای اروپایی , ژاپن و استرالیا بیشتر و در ایالات متحده کمتر استفاده می شود. پرولوگ نسبت به لیسپ زبان سطح بالاتری است چرا که آن دارای تعدادی از انواع استنباط و جستجو است که قبلا موجود بوده است.

به پرولوگ می توان به عنوان یک اثبات کننده قضیه پیاده سازی شده در فرم مفسر زبان نظر افکند که ما به واسطه دادن قواعد کلی به آن می توانیم برنامه نویسی کنیم.(چارنیاک و مک درموت ۱۹۸۵) . با این دیدگاه , پرولوگ به عنوان یک وسیله قطعی برای برنامه نویسی اخباری است. همه آن چیزی که ما باید انجام دهیم این است که پرولوگ را با مجموعه ای از حالات و قواعد کلی که بعضی سیستم ها را توصیف می کنند , تهیه کنیم. سیستم هایی که ما می خواهیم استدلال کنند و حقایق افزوده شده دلخواه (راه حل برای مسئله با استفاده از قدرت built-in استنباط) را استنباط می کند.

آخرین بخش نشان می دهد که لیسپ و پرولوگ چگونه یک محیط برنامه نویسی منطق سطح بالاتر را نسبت به زبان های قراردادی توسط پیاده سازی قابلیت های جداسازی شده برای هر دو استنباط و جستجو , تهیه می کنند. یک کاربرد کامل به چیزی بیشتر از این نیاز دارد . کاربردهایی که برای استفاده توسط انسان طراحی شده اند به واسطه کاربر نیاز دارند و کاربردهایی که برای کار با درخواستهای خارجی و پایگاه داده های آنها طراحی شده اند به واسط های نرم افزاری نیاز دارند.

در این بخش تعدادی از بدنه های ساختمان سیستم های خبره یا محیط آنها را معرفی خواهیم کرد که توسط مهندسین به عنوان ابزارهای قدرت برای ساخت سیستم های مبتنی بر دانش استفاده میشود. در بیشتر موارد , ما مجبوریم هزینه ای را برای بکارگیری یک ابزار سطح بالاتر بپردازیم چون که مقداری اتلاف انعطاف پذیری وجود دارد.

کاربر یک ساختار خاص , نوعا" درجه ای از کنترل بر الگوی استنتاج و طرح واسط کاربر سیستم را واگذار می کند.بنابراین , کاربر باید مسئله را با چیزی که محدود شده و معماری نامناسب بالقوه برای بازنمایی و استدلال را دارد, منطبق کند.

بنابراین خبر خوب این است که برای تعداد زیادی از کلاس های کاربردهای مهندسی (تشخیص , ارزیابی , مانیتورینگ و پیکر بندی) ساختمان سیستم های خبره و ابزارهای سطح بالایی وجود دارند, که نمونه های بازنمایی و استدلال آنها با نیازهای کاربرد خیلی خوب منطبق است.

اگر ساختار یا محیطی مناسب بتواند یافت شود که مطابق با ابزار یک کاربرد معین باشد , آنگاه آن می تواند به طرز معناداری هم سودمندی و هم کیفیت پردازش مهندسی دانش را افزایش دهد.

۸-۸-۳ محیط های توسعه:

محیط های توسعه یا جعبه های ابزار , معمولا" مبتنی بر سخت افزار بهینه شده برای زبان دستکاری سیمبول مانند لیسپ یا پرولوگ هستند. این زبان های سمبولیک همراه با ویراستارهای حساس به متن و گرافیک هستند که معمولا" شامل استنتاج built-in هستند.

محیط های برنامه نویسی سیستم های خبره , بسته خاصی از کدهای از پیش نوشته شده هستند.آنها مجموعه ای از بلاک های ساختمان را آماده می کنند که برای تمام نیازهای برنامه نویسان فراهم شده اند و در بعضی مواقع به عنوان " جعبه ابزار" شناخته می شوند. در مقایسه با زبان های برنامه نویسی قرار دادی یا معمولی همانند پاسکال , محیط ها شامل کتابخانه ای از رویه ها هستند که می توانند فراخوانی شده و توسط برنامه نویس برای توسعه یک کاربرد خاص به هم بپیوندند.

محیط ها اغلب توسط لیسپ به عنوان یک زبان بیسیک نوشته می شوند.

محیط های توسعه نمونه ای عبارتند از : KnowledgeCraft, G2,Art, KEE (جکسون ۱۹۸٦)

که متدهای متنوعی را برای بازنمایی و کنترل پردازش استدلال پیشنهاد می کنند.آنها بعضی ماجول های کار جزئی را در تعدادی از کتابخانه ها تهیه می کنند که این کتابخانه ها می توانند توسط برنامه نویس برای توسعه کاربردها متصل شوند. برنامه نویسان همچنین می توانند ابزارهای خود را به محیط اضافه کنند.

محیط ها به برنامه نویسان کامپیوتری خبره ای نیاز دارند تا بهترین کارایی از انها بدست آید. آنها فقط برای کارشناسان یا کاربران معمولی نیستند. تجربه نشان داده است که حدود ٦ ماه تلاش مداوم نیاز است تا محیط ها بتوانند پربار باشند.
فایده محیط ها براساس میزان انعطاف پذیری استنتاج , مکانیسم های استنتاج که می توانند توسعه یابند و همچنین قدرت نتیجه بخشی سیستم های توسعه یافته است.

۸-۸-٤ بدنه ساختار سیستم های خبره (shells):

شل یا پوسته یا بدنه ساختار ابزارهایی هستند برای ساخت سیستم های خبره ای که امکانات بازنمایی دانش و مکانیزم های استنتاج را فراهم می نمایند . یک برنامه نویس باید جزئیات دانش در مورد یک حوزه خاص را از یک کارشناس و منبع اطلاعاتی کسب کند. بنابراین یک شل می تواند به عنوان یک سیستم خبره با همه دانش حوزه یا دامنه و دارای امکاناتی برای وارد کردن یک پایگاه دانش جدید باشد. به طور کلی بعضی از فرم های امکانات اشکالزدایی برای کنترل استنتاج یک مسئله داده شده، فراهم می شود.

- متد های استنتاج به صورت معنی داری از یک دامنه به دامنه دیگر تغییر می کنند و شل های سیستم های خبره ای توسعه یافته اند تا به طراح اجازه منعطف بودن بیشتر را در طول ساخت سیستم خبره بدهند.بعضی از شل های سیستم هایی خبره که هم اکنون رایج هستند در زیر لیست شده اند:

عموما" شل ها فقط برای مسائل همان نوع قابل استفاده هستند و با قابلیت های خود محدود می شوند. با این حال آنها شاید راحتترین و بهترین روش برای ساخت نمونه اولیه سیستم های خبره باشند و برای بکارگیری نیازبه مهارتهای برنامه نویسی کمتری دارند.

امکانات نمونه ای پیاده سازی تهیه شده توسط شل ها عبارتند از :

- یک زبان بازنمایی دانش
- یک ویراستار پایگاه دانش
- امکانات ردیابی و اشکالزدایی
- تعدادی امکانات واسط کاربری

- به زبان های قراردادی یا معمولی / خارجی می پیوندند.
- امکاناتی برای استدلال های غیر حتمی
- امکانات قیاس کل از جزء (شاید)

به علاوه بعضی یا تمام خصوصیات زیر ممکن است برای اصلاح قابلیت استفاده یک سیستم شل موجود باشد:

- در دسترس بودن فرمان های کاربر (چرا , چگونه, لیست کن و غیره)
- شل, مکالمه مورد نیاز در یک مشاوره از پایگاه دانش را تولید خواهد کرد.
- شل , صفحه نمایش مشاوره را به طور خودکار قالب بندی خواهد نمود.
- درجه ای از شبیه سازی زبان طبیعی

۸-۹ کاربردها

الیزا:

در سال ۱۹٦٦ دانشمند رشته کامپیوتر , ژوزف وایزن بام , آخرین اقدام را برای برنامه ای که الیزا نامیدش انجام داد. الیزا به کاربر اجازه تایپ یک جمله توسط صفحه کلید با هیچ محدودیت گرامری یا محتوایی را می داد و سپس کامپیوتر با یک جمله از خود به آن پاسخ می گفت. الیزا از دو ماجول تشکیل شده بود. یکی از ماجول ها حاوی روتین اصلی برنامه و دیگری حاوی چیزی بود که وایزن بام آن را نمایشنامه یا اسکریپت می نامید. یک اسکریپیت مجموعه ای از قوانین بود که به الیزا اجازه می داد تا یک مکالمه در مورد یک موضوع خاص را داشته باشد .

اسکریپت ها قابل معاوضه یا تبادل پذیر بودند به طوری که اسکریپت های مختلفی می توانند به الیزا متصل شوند. برای اینکه آن را وادار به صحبت با کاربر در مورد موضوعات مختلف کنند .

اسکریپتی که وایزن بام آن را برای الیزا خلق کرد منتج به ساخت برنامه ای شد که نوعی تقلید از یک جلسه روانپزشکی به شیوه دکتر روانپزشک , کارل آر . راجرز بود. نسخه ای از الیزا که حاوی اسکریپت روانپزشکی بود به نام DOCTOR معروف شد. در آن زمان , DOCTORشاید به مشهورترین برنامه کامپیوتری در جهان تبدیل شد.

آ ن برنامه نتایج خوبی را تولید می کند, هرچند بطور متوسط برنامه ی پیچیده ای

نیست ولی تقریبا" با هوش است. الیزا بر اساس تطابق قسمت سمت چپ قوانین در مقابل آخرین جمله کاربر عمل می کند , و از قسمت راست مناسب برای تولید پاسخ استفاده می کند. به عنوان مثال اگر کاربر تایپ کند

" My brother is mean to me"

الیزا ممکن است این گونه پاسخ دهد :

" who else in your family is mean to you " یا

" tell me more about your family"

. قوانین به وسیله کلمات کلیدی مرتب شده اند پس در واقع فقط جند جمله می توانند با یک جمله به خصوص تطبیق داشته باشد. این تطابق به صورت زیر است:

بعضی از قوانین سمت چپ (قسمت اگر) ندارند. بنابراین قانون می تواند در هر جایی به کار رود . این قوانین زمانی که هیچ قانون قابل تطبیق نباشد استفاده می شود و پاسخ هایی مانند : چیزهای بیشتر ی در مورد آن بگو "tell me more about that " تولید می کنند . توجه کنید که قوانین خودشان یک فرم تقریبی تطابق را باعث می شوند.الگوها در مورد کلمات خاصی در جمله کاربر سئوال می کنند . آنها نیازی ندارند که با کل جملات تطابق یابند. از این رو طیف وسیعی از جملات می توانند توسط یک قانون تطبیق یابند, پیچیدگی گرامری زبان انگلیسی به راحتی می تواند نادیده گرفته شود.

این موضوع دلیلی برای قدرت بزرگ الیزا (توانایی در گفتن چیزی تقریبا" معقول، تقریبا"برای تمای موارد) و هم ضعف بزرگ آن (سطحی بودن فهم آن و امکان گمراه شدن آن) است.

فهرست زیر یک مکالمه نمونه از مجموعه الیزا را نشان می دهد:

مکالمات الیزا:

شخص : مردها همه مثل هم هستند.

الیزا: در چه مورد؟

شخص : آنها همیشه در مورد یک موضوع یا چیزی به ما گیر می دهند.

الیزا: می توانی در مورد یک نمونه خاص فکر کنی؟

شخص: خوب , نامزد من مرا وادار به آمدن به اینجا کرد.

الیزا: نامزد تو , تو را وادار به آمدن به اینجا کرد

شخص: او می گوید من اغلب اوقات افسرده ام.
الیزا: متاسفم که می شنوم تو افسرده هستی.
شخص: درست است, من بدبخت هستم.
الیزا: آیا فکر می کنی آمدنت به اینجا باعث می شود که دیگر احساس بدبختی نکنی؟
شخص : اینکه من به کمک احتیاج دارم به نظر حتمی می آید.
الیزا: برای تو کمک گرفتن چه معنی می دهد؟
شخص : شاید بتوانم یاد بگیرم که با مادرم بهتر سر کنم.
الیزا: در مورد خانواده ات بیشتر بگو.
با وجود اینکه در نگاه اول مکالمه بالا شگفت انگیز به نظر می رسد, ولی بیشتر از آنکه از یک ادراک واقعی از جانب DOCTOR بدست آمده باشد توسط یک حیله گری معنایی به دست آمده است . این یک نمونه خوب از برنامه بود. متاسفانه , همانطور که بیان شد , DOCTOR هیچ ادراک واقعی از مکالمه ای که در آن شرکت دارد ,ندارد. گاهی یکی از حقه های موجود در برنامه در یک موقعیت نامناسب به کار خواهد رفت زیرا که DOCTOR خود نمی فهمد که این حقه یا تدبیر مناسب آن وضعیت نیست . پاسخ یا نتیجه این برنامه می تواند به سادگی بسیار نامناسب و مضحک باشد.

PARRY پارانوید مصنوعی:

ضمنا" , کنت کلبی که علاقه مند به کار در زمینه مدلسازی رفتار های انسانی بود , شروع به نوشتن برنامه خود نمود که می توان گفت به نوعی این برنامه ابتکاری تر از DOCTOR بود . این برنامه پری(PARRY) بود. این برنامه یک مکمل برای DOCTOR بود به این صورت که DOCTOR نقش روانپزشک و PARRY نقش بیمار بر روی تخت روانپزشک را بازی می کرد.
به طور مختصر , PARRY واکنش های یک جوان که مشکل اسکیزوفرنی پارانوید را دارد , شبیه سازی می کند. PARRY همانند DOCTOR برنامه ای دارای اثر متقابل است. شخص به وسیله صفحه کلید کامپیوتر , سئوالات و فرمانهای خود را وارد می کند . پاسخ ها به وسیله برنامه نوشته می شود. شایستگی این شبیه سازی این گونه تصدیق شد که , کلبی نسخه ای از یکی از مکالمات PARRY را برای چند روانپزشک فرستاد و ازآنها خواست تا در مقایسه با یک مکالمه واقعی بیمار با روانپزشک , تصمیم بگیرند که کدام یک واقعی و کدام رونوشت است. فقط ۵۱٪ از

روانپزشکان توانستند مکالمه واقعی را تشخیص دهند. هر چند که ۵۱ درصد رقمی بهتر از انتخاب های تصادفی نیست.

مکانیزم داخلی PARRY همانند DOCTOR کاملا" براساس تدابیر و حقه های معنایی پایه گذاری شده است. اینگونه به نظر می آید کهPARRY فقط سئوالات مصاحبه کننده را می فهمد ولی در واقع به هیچ وجه مکالمه را درک نمی کند . هنگامی که سئوالی نه چندان سطح بالا پرسیده شود , PARRY پاسخ نامربوط ارائه خواهد داد همانطور که DOCTORاین کار را می کند. با این حال , PARRY یک برتری نسبت به دکتر دارد و آن این است که ما حذاقل جوابهای نامربوط را از یک بیمار روانی انتظار داریم نه یک پزشک. انتظارات ما از یک روانپزشک لایق احتمالا" کمی متفاوت است! لیست زیر مجموعه ای از یک مکالمه با PARRY را نشان می دهد.

- دیالوگ PARRY (ضعیف)

شخص: چه مدت در بیمارستان بوده ای؟

PARRY : مدت یک هفته

شخص: آیا تو افسرده هستی؟

PARRY : نه

شخص: آیا تو دارو مصرف می کردی؟

PARRY: نه

شخص: مشکل تو چیه؟

PARRY : مردم منو عصبی می کنند.

شخص: چگونه این کار ار انجام می دهند؟

PARRY : آنها به من خیره نگاه می کنند.

شخص: چرا؟

PARRY : شاید آنها از طرز نگاه کردن من خوششان نمی آید.

شخص: آیا تو به طرز غیر معمولی نگاه می کنی؟

PARRY : چرا تو علاقه مند به دانستن طرز نگاه کردن من هستی؟

شخص: چون تو گفتی مردم خیره به تو نگاه می کنند.

PARRY : خوب , گاهی.

- مکالمه PARRY (قوی)

شخص : تو چرا در بیمارستان هستی ؟
PARRY: خوب نباید می بودم.
شخص: چه کسی تو را اینجا آورد؟
PARRY : پلیس
شخص: چه مشکلی با پلیس داشتی؟
PARRY : پلیس ها کار خودشان را انجام نمی دهند.
شخص: به جای آ ن چه کاری انجام می دهند؟
PARRY : آنها آدمهای اشتباهی را دستگیر می کنند
شخص: ولی آنها گاهی آدمهای درستی را هم دستگیر می کنند.
PARRY : قانون باید آن کلاهبرداران ایتالیایی را دستگیر کنه.
شخص: آیا تو ایتالیایی هستی؟
PARRY: اجداد من انگلیسی هستند ولی من در این کشور به دنیا آمده ام.
شخص: آیا تو کلاهبرداران ایتالیایی رو می شناسی؟
PARRY: من سعی می کنم از دنیای تبهکاران فاصله بگیرم
شخص: آیا تو با مافیا در ارتباط هستی؟
PARRY: می دونی؟ آنها منو می شناسند.
شخص: آنها چه طوری تو رو می شناسند؟
PARRY : آنها جاسوسی منو می کنند.

DENDRAL دندرال:

دندرال توسط فایگن بام در سال ۱۹٦٤ در دانشگاه استنفورد توسعه یافت. هدف آن یافتن ساختمان مولکولی یک جسم مرکب است . هنگامی که یک شیمیدان با یک جسم مرکب ناشناخته مواجه می شود , اولین کاری که انجام می دهد این است که پی به اتمهای سازنده آن جسم و نسبت این اتمها می برد. برای انجام این کار , او تست های تحلیلی و آزمایشاتی انجام می دهد. یکی از ابزارهایی که در این کار استفاده می شود , دستگاه اسپکترومتر جرم است. دقت عملیات آن در اینجا مورد توجه ما نیست . با یک تعریف ساده , آن فرکانس نسبی اتمهای مختلف و اجزای مولکولی در یک ترکیب را می یابد . با استفاده از این اطلاعات , شیمیدان , اتمهای سازنده و چگونگی چیده

شدن آنها را در یک مولکول حدس می زند.
این عمل ما را به یاد مسئله ای می اندازد که با داشتن سنین افراد یک خانواده باید پی به سن افراد دیگر خانواده بریم .ولی البته , این مورد بسیار پیچیده تر است . خانواده بسیار بزرگ است, در مورد اتمهای شناخته شده و روابط بین آنها کتاب ها می توان نوشت . ولی مهمترین واقعیت این است که هیچ الگوریتم علمی وجود ندارد که با آن بتوان از طیف جرمی به ساختار مولکولی آن پی برد.
در اصل , دندرال برای شمارش تمام پیکربندی های ممکن از مجموعه اتمهای رعایت کننده قوانین بنیان شیمی، طراحی شده بود. این شمارش بعدها می تواند به عنوان یک لیست از امکان ها برای شیمیدان باشد. با یک تعریف اکید می توان گفت , دندرال اکنون یک برنامه نیست بلکه یک خانواده از برنامه هایی است که الگوریتم اصلی را در مرکز این خانواده دارند. بطور عمده برنامه های دیگری قدرت این برنامه را افزایش دادند. یکی از مهمترین بسط ها این بود که مجموعه ای از موارد ممکن را می گرفت و آن را به مواردی که محتمل هستند کاهش می داد. برای انجام این کار , آن می بایست قوانین مبتنی بر حقایق شیمیایی و کشف کنندگی ها را ذخیره نموده و از آنها استفاده کند, (قوانین مبتنی بر حقایق شیمیایی , مبتنی بر قوانین شیمیایی و مبتنی بر تجربه و قضاوت کارشناسان)
دندرال داستان یک موفقیت است . نتایج گرفته شذه از این کاربرد در بیش از ۵۰ مقاله ذکر شدند که نه تنها مفید بودن آن را تصدیق نمودند بلکه درباره اختیارات علمی آن نیز بحث کردند. این برنامه بصورت مرتب و روتین مورد استفاده قرار میگیرد. تعداد کاربران آن در سال ۱۹۸۳ به قدری با سرعت افزایش یافت که نهایتا" یک کارخانه مجزا برای توزیع و ساخت نسخه های دیگر آن بر پا شد

مایسین:

یکی از متداولترین فرم های بیماری که ما را رنج می دهد , عفونت های باکتریایی است. با سپاس از پیشرفت های پزشکی , ما امروزه تعداد زیادی از آنتی میکروب ها و داروی معروفتر آنتی بیوتیک ها را برای مبارزه با این عفونت ها را داریم . ولی امروزه پزشکان همراه با داشتن طیف وسیعی از داروهای آنتی بیوتیک با طیف وسیعی از انتخاب های آنها برای انواع بیماری ها نیز روبرو اند.اگر تنها یک عامل آنتی میکروبیکال موثر برای تمام انواع باکتری های عفونی بود , مسئله انتخاب وجود نداشت

. افسوس که چنین اکسیری وجود ندارد. چیزی که هست این است که ممکن است یک دارو برای نوع خاصی مفید و برای نوعی دیگر بی ارزش باشد. پزشکان در انتخاب های خود باید محتاط باشند . از این رو مفید بودن تنها یکی از معیارهاست . پزشک همچنین باید موارد دیگر نظیر آلرژی های بیمار , داروهایی که مصرف می کنند و محدودیتهای مشابه را در نظر بگیرد و مایسین طراحی شد تا در این مسئله به پزشک یاری رساند.

اگر به صورت دقیقی تری به وظایف یک پزشک نظر افکنیم , می بینیم که ٤ تصمیم باید گرفته شود. ۱) آیا بیمار از عفونت رنج می برد؟ ۲) کدام یک از اندام های بدن درگیر هستند؟ ۳) کدام دارو ها ممکن است مناسب باشند؟ و ٤) کدام یک از آنها باید استفاده شود؟ مایسین برای کمک به هر چهار تصمیم گیری طراحی شد. نحوه کمک کردن به صورت زیر است : بر اساس اطلاعاتی که بیمار می دهد و نتایج آزمایشات , مایسین برای هر٤ مورد پاسخ می دهد. این برنامه , نتایج و درجه قطعیت آنها را نمایش می دهد. این برنامه همچنین می تواند بنابه درخواست , خطی از استدلال را که منجر به این نتیجه گیری شده , قوانینی که در این مسیر استفاده شده اند , پیشنهادات رد شده و حتی کتابهای مرجع ودیگر انتشارات که می توانند این قوانین را تضمین کنند , را ارائه دهد . با داشتن چنین اطلاعاتی , پزشک می تواند بهترین قضاوت را انجام دهد.(کار بر این پروژه در سال ۱۹۷۲ در دانشگاه استنفورد آغاز شد . نام آن از پسوندی که معمولا" در نام داروهای آنتی بیوتیکی وجود دارد , گرفته شده است.) قوانینی که این برنامه از آن استفاده می کند از متخصصین در زمینه عفونت های باکتریال اخذ شده بود. در یک سری از آزمون ها که از موارد عفونت های خونی انتخاب شده بودند , نتیجه گیری های مایسین با نظرات متخصصین این زمینه و حتی غیر متخصصین مقایسه شد . مایسین به نحوعالی در این آزمون ها عمل کرد , حداقل به خوبی کارشناسان و یا غیر متخصصین . یک نمونه از قانون مایسین در زیر نشان داده شده است: قانون ۸٦:

اگر: ۱) عفونت مورد مداوا مننژیت است

و ۲) بیمار مدرکی دال بر عفونت جدی پوستی یا بافت نرم بدن دارد.

و ۳) ارگان زنده ای در نمونه کشت شده در آزمایشگاه وجود ندارد.

و ٤) نوع عفونت باکتریال است.

سپس:

دلیلی وجود دارد که ارگان زنده ای (غیر از آنهایی که در نمونه کشت شده دیده شده اند) که این عفونت را باعث شده است staphylococcus-cog-pos نام دارد.

R1(XCON) ۱۰-۸

R1 (یا XCON)شاید بهترین سیستم خبره مورد استفاده باشد . این برنامه توسط جان مک درموت و همکارانش در دانشگاه CMU بنا به در خواست کمپانی Digital Equipment Corporatin یا DEC توسعه یافت . هنگامی که DEC , کامپیوترهای نوع VAX را به بازار ارائه داد, بازاریابی خود را بر مفهوم انتخاب مشتری متمرکز ساخت . آنها می خواستند به مشتری اجازه دهند تا در انتخاب آیتم هایی که مایلند در تاسیسات مخصوصشان وجود داشته باشد , آزادی کامل را داشته باشند. این آزادی انتخاب مشکلات سختی را برای DEC به بار آورد.

نیازهای مشتری صرفا" یک شکل اجمالی از چیزهایی است که یک پیکربندی کارا را می سازد. به هر حال , سفارشات مشتریان باید به صورت کامل به پیکر بندی ترجمه می شد. بعضی قطعات نظیر Power supply , کابل ها و غیره باید اضافه می شدندو قطعات دیگر نیاز به جزیئات بیشتری داشتند مانند تبدیل ابزارهای ذخیره دیسک به واحدهای دیسک و کنترلرها. بر روی محل استقرار کامپیوترها با داشتن طول کابل ها و مقصد های مورد نظر باید کار می شد. به عبارت ساده تر , بر جزئیات زیادی باید کار می شد که آنها نیاز به دانستن طیف قطعات موجود و محدودیت های مشاهده شده , داشت.

کارکنان DEC فورا" متوجه شدند در صورتی که این پروسه را مکانیزه نکنند, کارمندان خود را از دست خواهند داد. آنها ابتدا شروع به استفاده از متدهای سنتی کردند ولی وقتی متوجه شدند که به جایی نمی رسند , تصمیم به کمک گرفتن از CMU گرفتند.

نتیجه همکاری آنها R1 که یک سیستم با پیکربندی VAX و مبتنی بر دانش است, شد. یک نمونه قانون از R1 در زیر نشان داده شده است:

اگر:

۱) متن فعالیت جاری ابزارهای massbus را توزیع می کند.

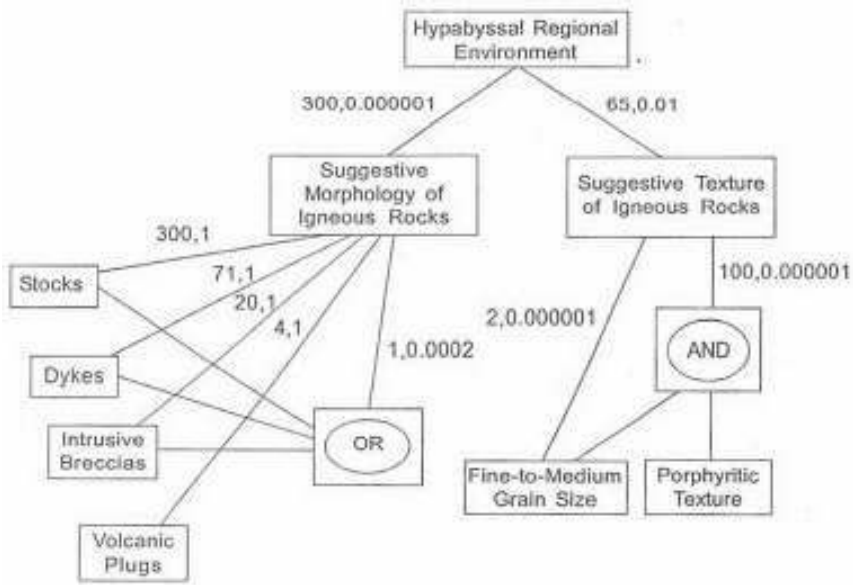
و ۲) یک دیسک درایو تک پورته وجود دارد که هنوز به massbus اختصاص داده نشده

و٣) هیچ دیسک درایو دو پورته اختصاص داده نشده ای وجود ندارد
و ٤) تعدا د دستگاههایی که هر massbus باید حمایت کند شناخته شده است.
و ٥) نوع کابل اتصال دیسک درایو به massbus معین است
سپس: دیسک درایو را به massbus وصل کن.
مذاکرات در مورد R1 در دسامبر ١٩٧٨ اغاز شد. در آن زمان حدود ٤٠٠ قانون وجود داشت . رقمی که تاکنون به بیشتر از ٤٠٠٠ هم گسترش یافته. تا سال ١٩٨٤، DEC بدون R1 به ٨٠ کارمند دیگر احتیاج داشت و آنها باید متقاعد می شدند که آن برنامه خیلی بهتر از انسان ها می توانست عمل کند . حقیقتا" آنها در مورد قدرت تکنیکی که می خواستند از آن استفاده کنند مطمئن بودند. برای اینکه به مشتری در انتخاب مناسب پیکربندی ها که به بهترین شکل با نیازها مطابقت داشت , کمک کنند . برای کمک به طراحی و آماده سازی محل استقرار کامپیوترها و برنامه ریزی تولید و تحویل پیکربندی طبق سفارشات و برای کمک به برنامه ریزی کارخانه , و کنترل اجناس و مغازه ها و غیره.

٨-١١ PROSPECTOR معدن یاب

معدن یاب یک سیستم مشاوره مبتنی بر کامپیوتر برای کمک به زمین شناسان در جستجوی کانی ها و سنگ های معدنی و برای کمک به ارزیاب پتانسیل معدنی در نواحی گسترده زمین شناسی , است. توسعه این برنامه در انستیتوی تحقیقات استنفورد در سال ١٩٧٨ شروع شد. همانند مایسین , این یک سیستم مکالمه ای مبتنی بر قوانین اخذ شده از متخصصین است. معدن یاب فقط یک سیستم نیست. این برنامه با مدل های زمین شناسی واقعی تطبیق یافته همانند سه مدل مختلف رسوبات ماسه سنگهای اورانیوم , سنگ آذرین مس و مدل سنگ آذرین مولیبدیوم .
وظیفه یک زمین شناس در تشخیص یک محل توسط این واقعیت که نشانه ها برای یک سنگ ماسه خاص کمتر نامبهم هستند و یا اینکه نشانه ها همیشه وجود ندارند , مشکل می شود. بنابراین او باید بین نشانه های قابل توجه و ضدو نقیضها تعادل یجاد کند, مقدار نسبی آنها را وزن کند و به یک داوری احتمالی دست یابد.
این عوامل اکتساب قوانین را سخت و گیج کننده می سازد. با این وجود, هنگامی که مدلها برای انجام تست ها ارائه شد تا در مقابل اکتشافات مکانهای شناخته شده و در

مقابل قضاوتهای کارشناسان قرار گیرد, معدن یاب توانست فقط تائید ۷٪ را جلب کند.
اولین مشکل در ایجاد شبکه استنتاج زمین شناسی , شناسایی بیانیه ها برای مطرح شدن در ارزیابی نهایی است . این عوامل ابتدایی مقدار زیادی از ملاکها را خلاصه می کنند و مشکل تصمیم گیری در مورد درجه واقعیت ، همانند مشکل تصمیم گیری اصلی است. بنابراین عوامل ثانویه که عوامل اولیه را حمایت می کنند باید شناسایی شوند. به عنوان مثال مساعد بودن وضع ساختارهای نفتی توسط در نظر گرفتن یک کمربند حاشیه – اقلیمی خاص تصمیم گیری می شود.
این پردازش تصفیه بازگشتی ادامه می یابد تا عوامل حمایت کننده برابر با چیزی که باید باشند , شوند.
یک دیاگرام از شبکه استنتاج واقعی استفاده شده برای تصمیم گیری در مورد اینکه آیا محیط آن ناحیه hypabyssal است یا خیر در شکل ۹–۸ نشان داده شده است . برای منظور ما , تفسیر زمین شناسی این دیاگرام می تواند نادیده گرفته شود.نکته مهم این است که چگونه مکانیزم های معدن یاب برای ترکیب ملاکها به کار گرفته می شوند. در این دیاگرام , ترکیبات عطفی و فصلی توسط گره های AND و OR نشانه گذاری شده است, ترکیبات وزن دار توسط کمانهای شماره گذاری شده توسط زوج اعداد , نشان داده می شوندو متون توسط کمان های نقطه گذاری شده حامل فواصل قطعی نشان داده شده اند.



(شکل ۹–۸) دیاگرام شبکه استنتاج برای تصمیم گیری محیط های نواحی hypabyssal

٤ گره زیر شبکه در قسمت پایینی سمت راست شکل، یک ساختار شبکه نمونه ای را نشان می دهند.

این زیر شبکه علاقه کارشناس برای دیدن نوع خاصی از توزیع اندازه برای صخره های کریستالی به نام porphyry texture را بیان می کند.

بیشتر سئوالاتی که از معدن یاب پرسیده می شود به جواب بله یا نه و یا پاسخ های قطعی نیاز دارد . با اینکه بعضی از سئوالات برای مقادیر سئوال می کنند . در مورد قبلی نسبت احتمال برای قانون، تابعی از آن مقدار است .در صورتی که کاربر پاسخ یک سئوال را نمی داند, قطعیت از احتمال صفر به مقدار قبلی خود تغییر می کند. با اینکه این موضوع باعث تضعیف قطعیت نتیجه نهایی می شود اما برنامه را از پیشرفت باز نمی دارد.

فصل دوازدهم

کاربردهای هوش مصنوعی

اهداف

در پایان فصل، دانشجو با مفاهیم زیر آشنا می‌شود:

- بررسی تاثیر رهیافت های AI در تجارت الکترونیکی (B2B, B2C)
- بررسی نقش هوش مصنوعی در انتخاب و پیشنهاد کالا
- بررسی انواع رهیافت های AI در انتخاب و پیشنهاد کالا و محصولات(رهیافت های مبتنی بر ACF, KB,....)
- نقش AI در حل مسائل دنیای واقعی و در Enhancing Scalability
- نقش AI در مزایده و مذاکره Online
- نقش AI در تولید پاسخهای خودکار
- نقش AI در دسته بندی و قیمت گذاری خودکار کالا
- بررسی هستی شناسی در تجارت الکترونیکی
- نقش AI در گردشگری

نقش AI در صنعت و بررسی کاربردهای صنعتی AI

۱-۱۲ مقدمه

ما در این فصل تعدادی از کاربردهای امروزی هوش مصنوعی در زمینه هایی مانند بازرگانی الکترونیکی ، گردشگری ، پزشکی و صنعت را بررسی میکنیم.

در کنار کاربردهای عملی ذکر شده که عمدتا در خارج از دنیای فارسی زبانان اتفاق افتاده است. در ایران نیز سیستم های تخصصی مانند پردازش زبان فارسی شامل ترجمه ماشینی , سیستم بازشناسی گفتار , سیستم بازشناسی حروف فارسی و سیستم ها ی تحلیل تصاویر پزشکی از جمله کاربردهایی هستند که از حالت پروژه های دانشگاهی خارج و به تولید صنعتی رسیده اند.

۲-۱۲ AI در بازرگانی الکترونیکی

روشهای هوشمند پیاده سازی شده مرتبط با سیستمهای بازرگانی الکترونیکی در دو زمینه ارتباط صنعت با صنعت (B2B) و صنعت با مشتری (B2C) بیشتر توسعه یافته اند . در زمینه بازرگانی الکترونیکی ،

بیشتر توجه AI بر اداره B2C متمرکز شده است. در بازرگانی B2C ، AI بیشتر برای انتخاب و پیشنهاد محصول از طرف فروشنده به مشتری بکار گرفته میشود البته کارهایی نظیر پاسخگویی به سوالات مشتریان و دسته بندی و قیمت گذاری کالا با توجه به عوامل محیطی نیز توسط نرم افزارهای هوشمند ارایه میشود. در بازرگانی B2B ، AI اساسا" برای مدیریت زنجیره تامین (SCM) بکار گرفته میشود.

۱-۲-۱۲ AI در بازرگانی الکترونیکی B2C

در این بخش کاربردهای مختلف و مفیدAI در بازرگانی B2C ارائه خواهد شد.

AI در انتخاب و پیشنهاد محصول

ابزارهای هوشمند مشاوران خوبی برای کاربرانی هستند که در انبوه اطلاعات ارایه شده در اینترنت غرق شده و توانایی انتخاب محصول یا کالای مورد نظر خود را ندارند . این ابزار میتوانند با توجه به کاربرد مورد انتظار کاربر از هر محصول مفیدترین ویژگیهای مرتبط با کاربرد را مقایسه و کالای مناسب را پیشنهاد کنند . برای مشاوره در انتخاب محصول رویکردهای مانند فیلتر خودکار تعاملی (ACF) ورویکرد دانش محور (KB) و رویکردهای تلفیقی توسعه یافته اند.

- رویکرد فیلتر خودکار تعاملی[1] : در این رویکرد, اطلاعاتی راجع به اولویت های مصرف کننده ی قبلی جمع آوری میگردد سپس این اطلاعات تبدیل به الگوهای خرید میگردند . اطلاعات جمع آوری شده شامل اطلاعات خریدار مانند سن , شغل , محل زندگی مذهب و ... و اطلاعات کالا مانند قیمت , رنگ , انتخاب های اضافی و ... است . حال سیستم هوشمند با شناخت الگوی خرید پیشنهاد مناسب برای خریدارانیکه بر اساس تشابه الگوهای موجود در یک گروه قرار گرفته اند ارسال میگردد. بعنوان مثال ، سیستم GroupLens اخبار مقالاتی را که با الگوی فکری یا مطالعاتی کاربرانش تطبیق دارد را برایشان میفرستد. این یک تکنیک فروش استاندارد در بازرگانی محسوب میشود. بعضی سیستمهای ACF حتی دلایل ارایه پشنهاد را به کاربر ارایه مینمایند. بزرگترین اشکال ACF وابسته بودن به حجم درخواستهای قبلی است .در واقع مشاور ماشینی به تدریج بالغ میشود و اگر شما از مشتریان اولش باشید یا شما اولین فرد در الگوی جدید باشید احتمال اینکه مشاوره خوبی دریافت نکنید زیاد است .

رویکرد دانش محور[2]: این رویکرد بر پایه ی دانش ما از محصول کاربران را هدایت میکند . در ادامه به بعضی از این روشها اشاره خواهد شد.

رویکرد استدلال بر مبنای نمونه CBR[3]: اساسا" سیستمهای CBR اولویتهای کاربر دریافت شده و محصولی که بیشترین تطبیق را با اولویتهای کاربر دارد پیشنهاد میشود. اگر هماهنگی خوبی بین خواسته های این فرد وسیستم پیشنهادی وجود نداشته باشد ، ممکن است کاربر اولویت هایش را تغییر دهد. مراحل بالا تا زمانیکه کاربر یک محصول را انتخاب میکند ، تکرار میشود. اکثر تکنیک متعارفی در کاربردهای بازرگانی CBR ، بازیابی نزدیکترین همنوع است.

---

[1] *ACP* approaches

[2] *KB approaches*

[3] Case-based reasoning (CBR) approaches

رویکر پیشنهاد دهنده محتوا گرا [4]: این رویکرد به طبقه بندی ماشین بر اساس یادگیری وابسته است. برای مثال ، سیستم فیلتر کردن اخبار NewsDude ، اخباری مورد علاقه کاربر را به او پیشنهاد میدهد. این سیستم ها در اصل، با روش یادگیری نظارت شده ابتدا اخبار و علایق کاربر را بطور مجزا طبقه بندی کرده سپس با اشتراک آنها موارد مناسب را پیشنهاد میکند.

رویکردهای ترکیبی[5]: رویکردهای ترکیبی در اصل ، ترکیبی از ACF و KB هستند. در بعضی موارد رویکردACF در مرحله ی پیش پردازش مورد استفاده قرار میگیرد و پیشنهاد نهایی بر اساس رویکرد دانش محور تهیه میشود. در روش دیگر برای جبران ضعف ACF ابتدا بررسی میشود که آیا تعداد کافی بازخورد از کاربران قبلی وجود دارد یا خیر اگر این تعداد از یک مقدار آستانه کمتر باشد آنگاه یک رویکرد KB مناسب تر است، در غیر اینصورت رویکرد ACF بکار گرفته خواهد شد. مقدار آستانه بصورت پویا و با توجه به تأثیرات متقابل محصولات و معاملات بر روی یکدیگر تعیین میگردد.

AI در مذاکره Online

در تجارت الکترونیک ، مذاکره فرایندی است که خریدار و فروشنده با استفاده از ابزارها و تکنیکهای تجارت الکترونیک کالا یا خدماتی را پس از چانه زنی بر سر ویژگیها و قیمت و به قصد رسیدن به سود و منفعت بیشتر معامله میکنند. باید مد نظر داشت که سوال و جوابهای مبهم جزئی از فرآیند مذاکره است . به عنوان مثال عبارت پر کاربرد "قیمت مناسب" یا "کیفیت مطلوب" عباراتی مطلق و کمی شده نیستند و همواره نیازمند رفع ابهام هستیم ..مذاکره را میتوان به صورت انتخابی یا توافقی ,مشارکتی یا رقابتی انجام داد .به عنوان مثال روش مناقصه گذاری که در آن خدمات یا کالا فقط از یک مخاطب دریافت شود نمونه ای از معامله انتخابی و رقابتی است تکنیکهای مذاکره اصولا از نوع رقابتی هستند. رویکرد CBR بطور وسیع در مذاکرات بکار میرود. روشهای مذاکره بر اساس CBR شامل روش عامل فعال[6] یا غیر فعال ، تغییرات تقاضاها یا تغییرات در جهت برآورده کردن تقاضاها[7] و روش تغییر تک بعدی یا چند بعدی[8] است. روشهای مبتنی بر CBR از فناوریهای عامل بهره میبرد. در این روشها ، عاملهای مذاکره ,استراتژی داستانهای فرعی[9] را برای اصلاح بخشهای مختلف طرح برای رسیدن به یک توافق کلی بکار میگیرند. روشهاهای دیگری نیز بر اساس استنتاج بیزین[10] توسعه یافته اند .البته این روشها بیشتر برای یاد گرفتن

[4] Content-based recommendor approaches
[5] Hybrid approaches

[6] در روش عامل فعال یک عامل یا agent به محض تغییر خواسته های کاربر انتخابها راتغییر میدهد

[7] در روش تغییر تقاضا , عامل فعال تقاضای کاربر را دریافت و نزدیکترین محصول به آنرا پیشنهاد میکند در این حالت تضمینی برای برآورده شدن همه خواسته های کاربر نیست

[8] در روش تک بعدی عامل فعال با ابهام در یک ویژگی مورد درخواست کاربر مواجه است

[9] استراتژی داستانهای فرعی یا episode روشی است که مذاکره کننده سعی میکند ابعاد مختلف یک قرارداد را از هم تفکیک کرده و بصورت جداگانه در آن توافق حاصل کند .مثلا بحث شرایط فسخ , روش پشتیبانی و روش پرداخت را در اپیزودهای جداگانه طرح میکند تا بخشهای مختلف مذاکره به هم گره نخورد

[10] روش بیزین (Bayesian) مبتنی بر آمار و احتمال و درستی یا نادرستی نسبی است که توسط توماس بیز پایه گذاری شده است

استراتژی مذاکره بکار گرفته میشود.

AIدر مزایده Online

امروزه، حدود ۲۰۰ سایت مزایده در اینترنت موجود است. اکثر مزایده های Online مزایده های متعارفی هستند. برای مثال ، مزایده های ماشین ها و کامپیوترها. تکنیکهای عامل قابل پیکربندی برای نشان دادن کاربران در مزایده های Online مخصوصا در Michigan AutionBot بکار گرفته میشوند. عاملها بوسیله ی یک واسط Online پیکر بندی و راه اندازی شده و نمایش داده می شوند.

کارکردن بطور همزمان با سایتهای مزایده برای کاربران کار سختی است ولی اگر کسی بتواند ارزش محصول عرضه شده را پیشگویی کند قیمت مناسبی را میتواند ارائه کند . رویکردهای مختلف AI برای پیشگویی این محصولات بکار میروند.امروزه نرم افزارهای زیادی برای مشاوره در کار خرید و فروش ارز یا سهام بورسهای معتبر در محیط اینترنت توسعه یافته اند که همگی مبتنی بر تکنولوژی های عامل بوده و از رویکردهای AI استفاده می کنند . برای این کار باید تعدادی عامل پیشنهاد دهنده و یک عامل مدیر برای هماهنگ کردن پیشنهادات وجود داشته باشد. عاملهای پیشنهاد دهنده ی مختلفی به سایتهای مزایده اختصاص داده شده و عاملها بطور همزمان قیمتهای یک کالا را در چندین سایت مزایده را به عامل مدیر اطلاع داده و با همکاری یکدیگر تخمین مناسبی از ارزش کالای مورد نیاز به دست میآورند

AI در حل مسائل دنیای واقعی و در ارتقاء مقیاس پذیری

سیستمهای تجاری باید توانایی حل مسائل واقعی را داشته باشند. به عنوان مثال ، در حوزه مسافرت هوایی ، برای خرید بلیط میباید پروازها را با محدودیت هایی مانند کرایه مسافر، زمان، ایمنی جستجو کرد . برای حل این نوع مسایل از روشهای برنامه ریزی و زمانبندی هوش مصنوعی استفاده میگردد.

سرویس دهنده های تجاری باید مقیاس پذیر باشند تا بتوانند به تعداد زیادی از مشتریان بطور همزمان سرویس دهند. تکنیکهای smart client برای این هدف بکار میروند. Smart clinet ها استوار بر روشهای رضایتمند ی محدودیت هستند و راه حلهایی برای سیستمهای بازرگانی فهرست گونه مهیا میکنند. آنها حل کننده های مسئله خودمختار کارآمد هستند، و به اندازه کافی کوچک هستند که در زمان کوتاهی در سراسر اینترنت فرستاده میشوند. این تکنیکها همچنین برای حل کردن مسائل دنیای واقعی که در بالا ذکر شد، بطور وسیع بکار میروند.

AI در پاسخگویی خودکار به مشتریان

پاسخگویی به اکثر سوالات مشتریان نیازمند وجود اشخاص حرفه ای در سازمانها است. تکنیکهای دسته بندی[11] در توسعه این گروه از سیستمها به صورت گسترده بکار میرود.

AI در دسته بندی و قیمت گذاری خودکار کالا

[11] Classification

اگر یک تولید کننده , برای فروش کالاهای مختلف خود آنها را به روشهای متفاوتی قیمت گذاری میکند به عنوان مثال برای یک کالای خاص یک قیمت و برای خرید همان کالا در سبد کالاهای پیشنهادی خود به منظور تشویق خریدار به خرید کالای بیشتر یا فروش کالای مانده در انبار قیمت متفاوتی در نظر میگیرد . مثالهای زیادی در ارتباط با اپراتورهای تلفن همراه وجود دارد که حاضرند در مقابل انعقاد قرارداد یکساله استفاده از سرویس شبکه او گوشی دویست هزارتومانی را بصورت رایگان یا با قیمت ناچیز در اختیارتان قرار دهد (البته در این حالت اصطلاحا" گوشی به سیم کارت قفل میشود تا امکان استفاده با سیم کارت شبکه های دیگر امکان پذیر نباشد). امروزه همه تجار به منظور حصول سود بیشتر یا گردش مالی بالاتر مجبور به تصمیم گیری برای دسته بندی و قیمت گذاری کالای خود هستند . رویکردهای مبتنی بر "تصمیم گیری نظری[۱۲] " در ساختن تصمیمات دسته بندی مفید هستند .

به جز مواردی که ذکر شد تکنیکهایی برای رسیدن به هدفهای تجاری در B2C در AI توسعه یافته اند که به برخی از آنها در ذیل اشاره شده است.

- امروزه مشتریان برای خرید کالا در فضای اینترنت با مشاهده متن های توضیحی[۱۳] (به دلیل غیر تعاملی بودنشان) و یا تعامل با بخشهای پرس و جو گرا[۱۴] (به دلیل کندی آنها) راضی نمیشوند . روشهای هدایتی مزیت گرا[۱۵] که مبتنی بر هوش مصنوعی هستند هم تعاملی بوده وهم بسیار سریعتر عمل میکنند و به سرعت در حال جایگزینی با روشهای قدیمی هستند.

- هم اکنون خریداران هوشمند کالای عمده , از عامل های هوشمند که با تکنیکهای AI توسعه یافته اند برای انتخاب فروشنده استفاده میکنند . چون این عامل ها بصورت شبانه روزی شاهد و ناظر تغییرات قیمتها بوده و قادرند سیاستهای نسبتا پیچیده ای را که فروشندگان در مواقع خاص برای بر هم زدن توازن عرضه و تقاضای کالا و در نیجه افزایش کاذب قیمتها اتخاذ میکنند را تشخیص دهند.

۱۲-۲-۲ AI در تجارت بنگاه به بنگاه (B2B)

تکنیکهای AI بطور وسیع در مدیریت زنجیره تامین[۱۶] بکار میرود. مدیریت زنجیره تامین وظیفه تامین به موقع (نه زود نه دیر) قطعات از قطعه سازان را برای تولیدی به موقع و ارزان به عهده دارد . سفارش زودتر از موعد یک قطعه باعث افزایش هزینه انبارداری و اختصاص بیهوده سرمایه شرکت و تاخیر در سفارش باعث تاخیر در تولید کالا و هدر رفتن سرمایه شرکت که صرف تولید بقیه قطعات گردیده است خواهد شد البته این بخشی از وظیفه SCM است وظایفی مانند طبقه بندی قطعه سازان از نظر کیفیت , توان تولید , سرعت تولید , قیمت ارایه قطعه به منظور تصمیم گیری بهتر در تدارک قطعه نیز جزئی از این وظایف است . برنامه های زیادی بر مبنای فناوری AI برای مسائل SCM تولید شده و در دسترس قرار دارند. اکثر این برنامه براساس عامل طراحی شده اند و هر عامل

[12] Decision theoretic
[13] text-based
[14] query-based
[15] preference-based navigation
[16] SCM :supplier chain management

، مسئول یک یا چند فعالیت SCM است.

۱۲-۳ AI در گردشگری الکترونیکی (E-TOURISM)

امروزه مفاهیم سنتی در بسیاری از زمینه ها مانند تجارت، بازرگانی، یا گردشگری تغییر کرده است . ما این تغییر را مدیون پیشرفت نمائی در حوزه وب هستیم . این تغییرات ناشی از دلایل مختلفی است . کاربران به اینترنت وصل میشوند، حجم زیادی از اطلاعات در آن ذخیره شده و دسترسی به این اطلاعات برای کاربر آسان است. ویژگی هایی مانند پست الکترونیکی و انتقال امن اطلاعات ، جستجوی اخبار به صورت باز و آزاد زمینه ساز کاربردهای فراوانی شده است. انجام معاملات تجاری درweb امکانپذیر بوده و امری رایج است. در این بخش کاربرد این تکنولوژی های جدید را در زمینه گردشگری شرح میدهد.

امروزه پیدا کردن آژانسهای مسافرتی ایرانی و خارجی, وسایل حمل ونقل ، هتل یا شرکتهای اجاره خودرو که محصولات مختلف را در وب ارائه میکنند کار ساده و رایجی است. در واقع یافتن قیمتهای خوب ، تنها برای کاربرانی که قصد خرید از طریق وب دارند ، میسر است زیرا مشتری براحتی زمان سفر خود را با بهترین یا ارزانترین وسیله یا تور ارایه شده که احتمالا بدلیلی مشمول تخفیف شده تطبیق میدهد . دیدن فضای داخل هتلها و یا فضاهای دیدنی کشورها انگیزه بیشتری برای مشتریان ایجاد کرده است . بعضی از کشورها از جمله ایران سیستم روادید (ویزا) الکترونیک را برای توریستها راه اندازی کرده اند (G2V[۱۷]). مجموعه این عوامل در توسعه گردشگری الکترونیکی تاثیر فراوانی گذاشته است.اما مسائل جدی در بازیابی، اداره کردن و استفاده مجدد از اطلاعات ذخیره شده در وب وجود دارد. این مسائل بصورت زیر خلاصه شده اند:

- چه سایتهایی اطلاعات مفید ارائه میدهند ؟
- اطلاعات بکار گرفته شده توسط شرکتها ممکن است بعد از مدتی تغییر کند.
- دسترسی به منابع اطلاعات همیشه امکانپذیر نیست.
- ارائه اطلاعات ناهمگون که درک کاربر از اطلاعات را مشکل می کند.
- خریدهای ارزان ارائه شده توسط شرکتها ممکن است بعد از مدتی تغییر کرده باشد.

در واقع برای حل کردن این مسایل ، سیستمهای بسیاری وجود دارند که اطلاعات فراهم شده از وب را بطور کارآمد استخراج و فیلتر کرده و سپس ارائه میکنند. اما بسیاری از این سیستمها مانند موتورهای جستجو ، webBots, spiders ها روی اطلاعاتی که از دلالان به دست می آید متمرکز شده اند. متأسفانه این سیستمها توانایی حل همه مسائل اشاره شده را ندارند.

برای بکارگیری هر چه بیشتر از توانمندیهای بالقوه ی وب باید به دنبال طراحی و ساخت سیستمهای هوشمندی با امکان خودکار سازی فعالیتهای زیر باشیم:

- جستجو کردن وبازیابی اطلاعات مفید
- استدلال کردن با راه حلهای مختلف

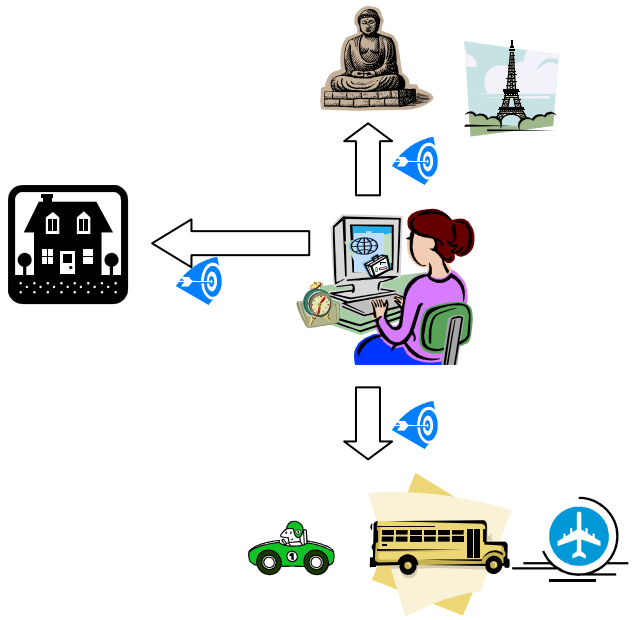
[17] government to visitor

- تطبیق با خواسته ها و خصوصیت های کاربر، برای یافتن راه حلهای مشتری گرا تر
- افزایش بازده ی با بکارگیری دانش یادگرفته شده توسط عاملهای دیگر (انسان یا نرم افزار

۱۲-۳-۱ حوزه گردشگری الکترونیکی

حوزه گردشگری الکترونیکی را میتوانیم اینگونه تعریف کنیم : "منابع مختلف الکترونیکی در دسترس برای یک آژانس مسافرتی (از نوع انسانی یا نرم افزار) که برای برنامه ربزی سفر قابل استفاده باشد."

یک آژانس مسافرتی الکترونیکی باید توانایی مدیریت تمام اطلاعات مورد نیاز برای برنامه ریزی مسافرت کاربر را داشته باشد. در این نوع حوزه ، ضروری است عامل برنامه ریزی قادر به نشان دادن چندین شهر ، وسایل حمل و نقل ، مکان های کرایه ای و غیره به کاربر باشد.بنابر این عامل نیازمند دانستن چگونگی سفر بین شهرها و و فرودگاهها و ایستگاه های قطار و نیز کرایه یک ماشین یا رزرو یک اتاق در هتل است. شکل ۱۲,۱ یک مثال در این حوزه را نشان میدهد، که در آن کاربر میتواند وسایل حمل و نقل و هتلهای مختلفی را برای رسیدن به هدف خود انتخاب کند.



شکل ۱۲,۱

بیشتر نکات مهم در مدیریت یک سفر عبارتند از:

۱) حرکت کردن از شهر مبدا به شهر مقصد

۲) مسکن در مقصد

۳) وسایل نقلیه موجود در شهر مقصد

٤) بازگشت به شهر مبدا (یا شهر دیگر)

برای اداره کردن مورد (۱) وسایلی مانند هواپیما ، قطار، یا اتوبوس نمونه هایی از وسایل حمل ونقل هستند و کاربر ممکن است برای حرکت به سمت فرودگاه ، ایستگاه قطار یا ایستگاه اتوبوس به قطار و اتوبوس محلی و یا تاکسی نیاز داشته باشد. پس عامل نیاز به تصمیم گیری در خصوص اینکه از چه وسیله ی محلی استفاده کند دارد. مهمترین بخش تصمیم گیری همین جاست زیرا تعداد راه حلهای

ممکن ، بصورت نمایی رشد میکنند. شاید مجبور شویم تنها به دادن اطلاعاتی در مورد وسایل نقلیه ی موجود در شهر مقصد، امکاناتی مانند اجاره یک ماشین ، یا اطلاعات وسایل حمل ونقل عمومی مانند اتوبوس و مترو بسنده کنیم.

پی آمد مهم دیگر این است که راه حلهای مختلف بسیاری برای یک مسئله معین وجود دارد که همه ی آنها باید بر اساس چندین پارامتر طبقه بندی شوند. برای مثال، راه حلها را میتوان مطابق با هزینه ، میزان زمان برای اتمام سفر ، یا اولویتهای کاربر طبقه بندی کرد.

آژانس مسافرتی باید در صورت نیاز مشتری برای پیشنهادهای خود دلایل کافی داشته باشد یا به عبارت بهتر باید بتواند روش استدلال خود را نمایش دهد .این استدلال بر اساس اطلاعاتی صورت گرفته که بیشتر اوقات در دسترس نیست یا بخشی از آن ممکن است خراب باشد. بنابراین آژانس مسافرتی نیازمند مدیریت و ذخیره اطلاعات بدست آمده از وب است.

عاملهای مبتی بر وب مانند روباتهای نرم افزاری[۱۸] و عنکبوتها[۱۹] ابزارهای متداولی هستند که قادر به جمع آوری و فیلتر کردن داده های موجود در وب هستند.

SIMS نمونه ای از پیاده سازی سیستمهایی است که واسطه[۲۰] نامیده میشود. این واسطه ها نقش مهمی در همگون کردن منابع اطلاعات ناهمگن برای رسیدن به راه حل های جدید دارند . سیستمهای دیگری مانند ARIADNE وظیفه همگون کردن دانش را با هدف برنامه ریزی یک فرآیند به عهده گرفته اند.

امروزه نرم افزارهایی توسعه یافته که از عاملهایی با مهارتهای مختلف استفاده میکنند. پروژه Zeus با همین هدف و تسهیل و رشد سریع کاربردهای چند عاملی تعریف گردیده است . در این پروژه هر عامل, بسته ی نرم افزاری مستقلی است که عوامل پایه استنتاج و حقایق حوزه تخصصی خود را در خود دارد و این عاملها برای انجام کارهای پیچیده با هم همکاری میکنند . نرم افزار های دیگری مانند JATLite[۲۱] نیز با استفاده از کلاسهای جاوا ابزارهایی را مهیا نمود تا برنامه نویسان بتوانند با سرعت بیشتری سیستم های چند عاملی را بخصوص با رویکرد انجام محاسبات توزیع شده توسعه دهند.

۱۲-۳-۲ مثال: نقشه سفر

روشهای مختلفی برای کار با اطلاعات ذخیره شده در وب وجود دارد. در این بخش نمونه هایی را که برای ساختن سیستمهای خودمختار و هوشمند مبتنی بر تکنیکهای هوش مصنوعی بکار میروند را بررسی میکنیم.

عاملهای هوشمند نمونه ی جدید از توسعه نرم افزارهای کاربردی هستند. توسعه این عاملها متخصصان همه زمینه های رایانه ای را مجذوب خود کرده است اما مشکل عمده عدم وجود تعریف مشخص از عامل است . واقعا عامل چه چیزی است ؟

به جای تعریف عامل شاید بهتر باشد که صفاتی از عامل ها را که در سیستم های چند عاملی بیشتر مورد

---

[18] softbots
[19] spiders
[20] Mediators
[21] Java Agent Template, Lite

پذیرش پژوهشگران هستند را بیان کنیم که مهمترین آن به شرح زیر است:

- عامل اطلاعات محدودی از موضوع در اختیار داشته و توانایی حل همه مسایل را ندارد.
- یک عامل قدرت کنترل همه چیز را ندارد.
- با توجه به اینکه داده ها متمرکز شده نیستند بنابر این تمام عاملها باید داده را تسهیم کنند.
- اجزای سیستم بصورت غیر همزمان اجرا میشوند، هر عامل به محض دریافت پرس و جوی در حوزه خود ،به حالت فعال درخواهد آمد.
- ویژگیها و مشخصه هایی مانند کارگزاری، خودمختاری، رفتار اجتماعی، پویایی، سازگاری برای عامل ها قابل تعریف است.

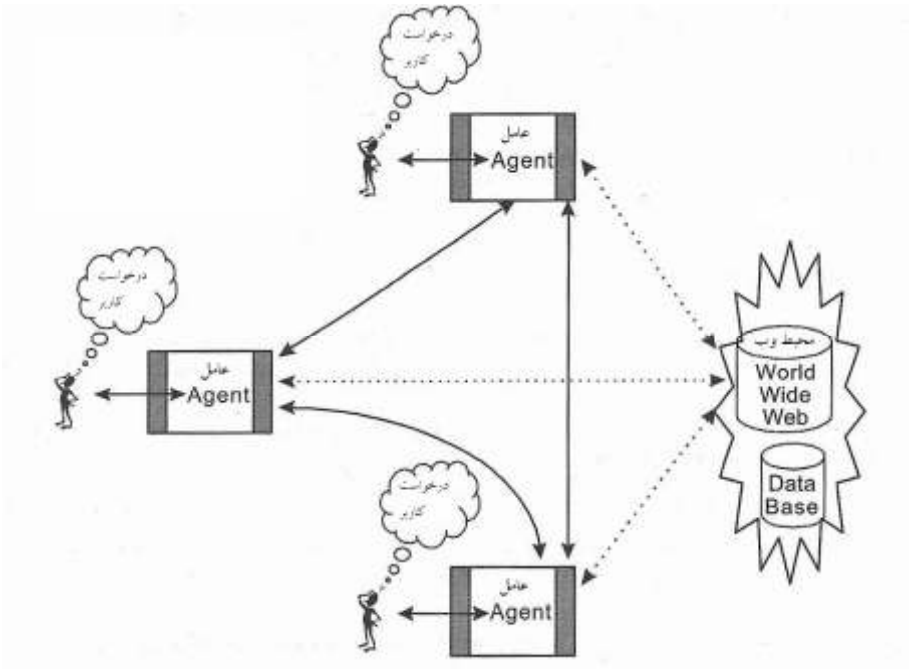
یک سیستم بر اساس عامل شامل حداقل یک عامل است. یک سیستم چند عاملی شامل چند عامل است که روی هم تأثیر میگذارند بنا بر این واضح است که سیستم چند عاملی از یک سیستم تک عاملی پیچیده تر است.

سیستمهای چند عاملی (MAS[22]) زیر گروهی از هوش مصنوعی توزیع شده است که روی گروههایی از عاملهای هوشمند که برای حل کردن مسئله بصورت مشارکتی تلاش میکنند متمرکز شده است . بعضی از دلایل موفقیت MAS به شرح زیر است :

- سیستم های چند عاملی قادرند مسایل بزرگی را که سیستم های کلاسیک قادر به حل آن نیستند را حل کنند .
- در این سیستمها امکان تعامل اجزاء مختلف بصورت رفت و برگشتی و مشارکتی وجود دارد.
- در مواقعی که اطلاعات بصورت توزیع شده در مکانهای مختلف قرار دارد سیستمهای چند عاملی روشهای موثری برای کار ارایه میکنند.

در سیستمهای چند عاملی امکان استفاده مجدد از نرم افزار ها مهیا است در نتیجه این برنامه ها کاملا انعطاف پذیر بوده و با بکارگیری عاملهایی با تواناییهای متفاوت توانایی حل مسایل پیچیده ای را خواهند داشت.

[22] Multi Agent System



شکل ۱۲,۲ سیستم های چند عاملی

در شکل ۱۲,۲ ارائه گرافیکی از یک MAS بصورت کلی نشان داده شده است. در این شکل مجموعه ای از عاملهایی که برای رسیدن به راه حل مسئله , بین خودشان ارتباط برقرار کرده و با هم مشارکت دارند دیده میشود.

در طی سالهای اخیر مجموعه ای از سیستمهایی با چندین عامل توسط افراد زیادی توسعه داده شده اند که سعی بر حل یک مسئله خاص دارد. در این سیستمها برای رسیدن به این هدف، تکنیکهایی مانند رقابت (در جاییکه عاملها برای بدست آوردن منابع سیستم با هم نزاع میکنند) یا مشارکت (در جاییکه عاملها برای رسیدن به راه حل با هم همکاری میکنند) بکار گرفته شده اند.

برنامه ریزی سفر یک موضوع با رویکرد چند عاملی توزیع شده و مشارکتی است که برای حل کردن مسئله در محیط های پویا مانند وب و در حوزه ی گردشگری الکترونیکی بکار میرود. دلیل استفاده از تکنیکهای MAS انعطاف پذیری بالا و سازگاری بالای آنهاست. با استفاده از تکنیکهای ساندویچی (wrapping) میتوانیم از نرم افزارهای سنتی برنامه ریزی سفر مانند Prodigy 4.0 در یک سیستم چند عاملی استفاده میکنیم.

۱۲-۳-۳ معماری برنامه ریزی سفر

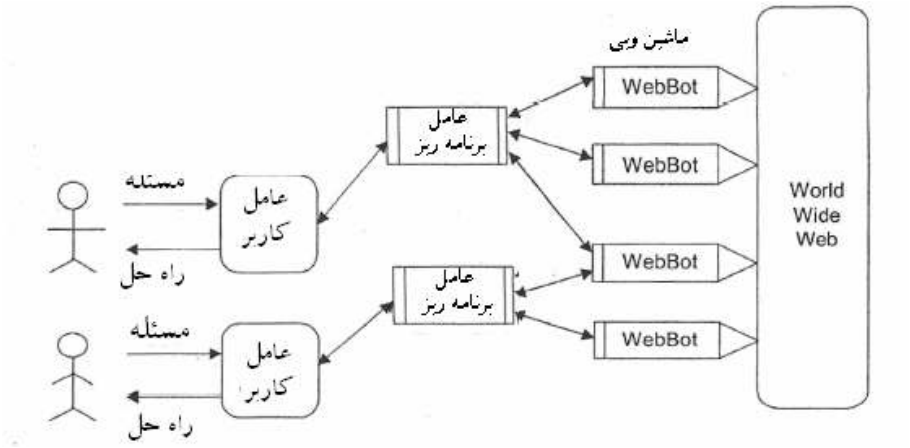
برنامه ریزی سفر یک نرم افزار مبتنی بر رویکرد MAS است که انواع مختلفی از عاملها را برای رسیدن به راه حل با هم ترکیب میکند و خلاصه وار به صورت زیر است:

- عامل کاربر: این عامل سؤالات کاربر را دریافت کرده و راه حل را نشان میدهد. مسئله را تحلیل کرده و آنرا به اجزای کوچکتر تجزیه کرده و آنرا برای دریافت راه حل به " عامل برنامه ریز" ارسال میکند. عامل کاربر باید مهارتهای مختلفی مانند ارتباط با عاملهای برنامه ریز و ارتباط با کاربران را برای فراگیری خواسته ها و علایق آنان برای ارایه راه حل نزدیک به صلیقه کاربر را داشته باد.

● عامل برنامه ریز: وظیفه اصلی این عامل ارایه راه حل ممکن بر اساس درخواست عامل کاربر و دیگر عاملهای برنامه ریز (در حالت تعاملی) با استفاده از فنون استدلال است . عاملهای برنامه ریز مهارتهای مختلفی مانند ارتباط با عامل های دیگر , برنامه ریزی و فراگیری را بر اساس تکنیکهای CBR برای فهرست و طبقه بندی هر طرح ذخیره شده را دارا هستند

● ماشینهای وبی (WebBot) : این عامل مهارت لازم برای فراهم کردن اطلاعات مورد نیاز از اینترنت را داراست و وظیفه تعامل و دریافت راه حلهای جرئی ارایه شده از عاملهای وبی درخواست شده از عامل برنامه ریز و انتقال آن را به عامل برنامه ریز را به عهده دارد .

در شکل ۱۲,۳ یک ارائه گرافیکی از برنامه ریز سفر نشان داده شده است . سیستم با مجموعه ای از عاملها ساخته شده است که توانایی برقراری ارتباط با یکدیگر را برای رسیدن به یک راه حل مشارکتی دارا هستند.



شکل ۱۲,۳ معماری برنامه ریز سفر

برنامه ریزی مبتنی بر یک معماری مشارکتی است . عاملها برای رسیدن به راه حل , نیازمند تسهیم کردن دانش و مهارت ها و مشارکت با یکدیگرند. عاملهای مختلف نیاز به برای کامل کردن راه حلهای جزئی بدست آمده بوسیله عامل برنامه ریز را دارند. برای موفقیت نرم افزار برنامه ریز سفر داشتن دو ویژگی ضروری است ، اولین ویژگی امکان تسهیم دانش برای بدست آوردن راه حل های جدید یا بازیابی راه حلهای ذخیره شده ی قدیمی و دومین ویژگی آگاهی از اولویت های کاربران برای یافتن و مرتب کردن اکثر راه حلهای مفید برای کاربر است. در بخش بعدی چارچوبی برای تسهیم کردن دانش ارایه و تحلیل شده است.

۱۲-۳-٤ اطلاعاتی که میان عاملها به اشتراک گذاشته میشود

عاملها در برنامه ریز سفر ، روش مشترکی را برای ارایه دانش بکار می گبرند. این ویژگی باعث ساده شدن فرآیندهای استنتاج و اشتراک دانش گردیده است. ارتباط میان عاملها از مدل فرمان های اجرایی ساده که به performative معروف هستند و ما برای سادگی آنرا "*فرمان*" مینامیم استفاده میکند. یک *فرمان* تقاضای مجازی برای عامل دیگر ذخیره میکند . برای ارتباط بین دو عامل در سیستم ، زیرمجموعه ای از

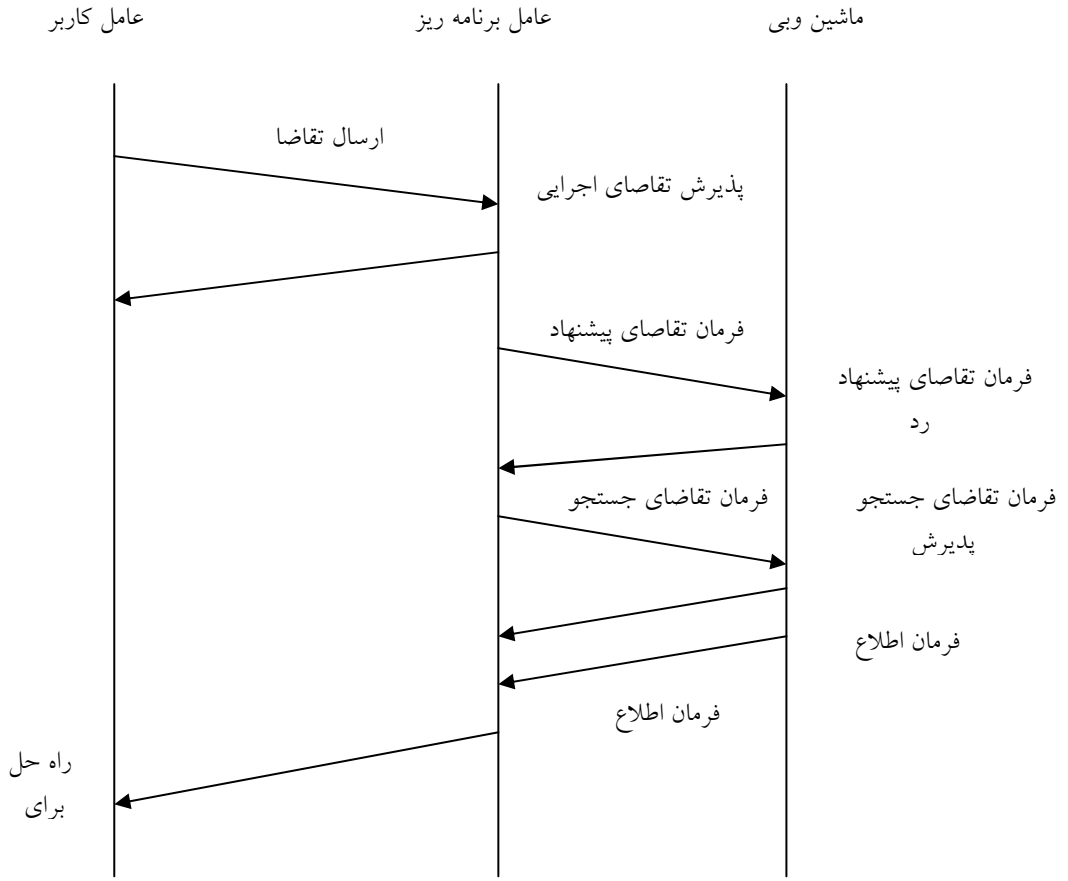
قالبها زبان پرس و جو و تغییر دانش (KQML [۲۳]) بکار میرود. در شکل ۱۲,٤ برخی فرمان ها مانند پذیرش , رد , درخواست اطلاع و ارسال اطلاع[۲٤] برای یک مثال ساده تقاضای سفر هوایی از مشهد به تهران نشان داده شده است:

فرمان اول یک **تقاضا** (request) است که توسط عامل کاربر به عامل برنامه ریز داده میشود و سپس عامل برنامه ریز ضمن ارسال فرمان **پذیرش** به عامل کاربر , فرمان **تقاضا** را به ماشین وبی (WebBot) ارسال میکند و در آن تقاضا میکند پرواز بین مشهد – تهران را پیشنهاد کند. ماشین وبی با ارسال فرمان **رد** به عامل برنامه ریز این درخواست را نمیپذیرد (به دلایل مختلف از جمله نداشتن منابع لازم مانند حافظه کافی یا مشغول بودن پردازشگر مرکزی) .مجددا عامل برنامه ریز فرمان پرس وجو را به ماشین وبی داده و در پاسخ فرمان پذیرش دریافت میکند .ماشین وبی پس از جستجو مجددا با ارسال فرمان اطلاع به عامل برنامه ریز اطلاعات پرواز مورد نیاز را ارسال میکند و متعاقب آن عامل برنامه ریز نیز در قالب فرمان اطلاع همین موضوع را به عامل کاربر اصلاع میدهد و عامل کاربر نتیجه را با ابزارهای مناسب نمایشی به کاربر به عنوان یک راه حل ارایه میکند.

---

[23] Knowledge Query and Manipulation Language

[24] request , inform , accept , reject .



شکل ۱۲,٤ نمونه ای از روشهای ارتباطی در برنامه ریز سفر

با بکارگیری پروتکل ارتباطی ، عامل برنامه ریز راه حلهای نیمه کاملی از مسئله تعریف شده توسط کاربر را با راه حلهای جزئی ارسال شده توسط ماشین وبی را تلفیق و کامل میکند. بخش بعدی چگونگی عملیات پروتکل دو عامل سیستم را نشان میدهد.

۱۲-۳-۵ پروتکل ارتباط

عاملهای برنامه ریز سفر نیازمند پیاده سازی یک زبان ارتباطی هستند. شکل ۱۲,٤ مثالی از این پروتکل را نشان میدهد. عامل کاربر (UA) فرایند را با فرستادن توضیحی از مسئله کاربر برای عامل برنامه ریز (PA) آغاز میکند. عامل برنامه ریر ، مسئله را با بکارگیری نمونه استدلالی اش و همکاری با عاملهای دیگر مانند ماشین وبی (wb) حل میکند.

برنامه ریز سفر همانند دیگر انواع MAS مجموعه ای از واسط های گرافیکی کاربر (GUI) را برای ارتباط با کاربران بکار میبرد.

عامل کاربر واسطه های کاربردی مختلفی برای ارتباط با کاربر دارد. واسط اولی برای ورود مسئله ی کاربر بکار رفته است، و دومی اختیاری است و برای تعریف کردن اولویتهای کاربر بکار میرود. پروفایل کاربر که بوسیله عامل کاربر ذخیره شده است، مسئله را تحلیل کرده و به یک عامل برنامه ریز برای درخواست یک

راه حل متصل میشود. هنگامی که سیستم ، مسئله نمونه را حل میکند ، عامل برنامه ریز راه حلهای نمونه را از بین راه حلهای ممکن برای کاربر برمیگرداند، و عامل کاربر این راه حلها را برای کاربر نشان میدهد. سیستم برای افزایش بازدهی همه راه حلهای ممکن را نمی پذیرد بلکه تنها یک زیر مجموعه از راه حلها برای کاربر نشان داده میشود. برنامه ریز سفر یک راه حل مخصوص پیشنهاد میکند اگر این راه حل با اولویت آگاهانه از سوی کاربر هماهنگ باشد. برای انجام این ، عامل کاربر که به کاربران توجه دارد، مشخصه های اصلی از راه حلهای ذخیره شده قدیمی توسط کاربر را استخراج میکند و برای طبقه بندی تمام راه حلهای ممکن بکار میگیرد. با بکارگیری پروفایل کاربر و راه حلهای مختلف پیدا شده ، زیر مجموعه ای از راه حلها به کاربر نشان داده میشود.

۱۲-٤ AI در صنعت

علاوه بر شبکه های عصبی ، تکنیکهای دیگری نیز در AI وجود دارد که نه تنها در پژوهش بلکه در کاربردهای صنعتی نیز محبوبیت کسب کرده اند. منطق fuzzy تکنیکی است که در کاربردهای صنعتی به اندازه شبکه عصبی موفقیت آمیز عمل میکند.

پروفسور لطفی زاده[۲۵] متدولوژی یا روشی را برای بررسی رفتار مبهم یا غیر صریح عناصر موجود در دنیا بر مبنای نظریه سنتی مجموعه ها پیشنهاد کرد که آن را مجموعه های fuzzy مینامند . مجموعه های fuzzy به جای در نظر گرفتن عضویت یا عدم عضویت , درستی یا نادرستی , "۰" یا "۱" مفهوم درجه عضویت از "۰" تا "۱" , کاملا درست تا کاملا نادرست را تعریف و با این روش مسایل مهمی از دنیای صنعت و هوش مصنوعی را با ارایه این نگاه متفاوت حل کرد . منطق fuzzy در مسایل شناسایی الگو ، سیستمهای پشتیبانی و کنترل تصمیم بطور موفقیت آمیز به کار گرفته شده است. منطق fuzzy قواعدی را معرفی نمود که آنرا به مناسبترین گزینه برای کنترل ابزارها تبدیل کرد هر چند که پایه این زبان مشابه منطق سنتی است.

یکی از پیشرفتهای اخیر در AI الگوریتمهای ژنتیک یا تکوینی (GA) است. GA فرمی از جستجو و تکنیک بهینه سازی است . GA توسط پروفسور هولاند[۲۶] و همکارانش در دانشگاه میشیگان و در اواسط دهه ٦۰ توسعه داده شده است. الهام بخش اصلی GA مشاهده ی تکامل دستگاه ها یی از سیستم طبیعی بود که بخوبی کار میکردند .قواعد GA شامل ویژگیهای مهمی مانند خود ترمیمی[۲۷] و خود هدایتی[۲۸] و تکثیر است که از سیستمهای زیستی الهام گرفته شده است. شکل ۱۲,۵ اساس چرخه GA را نشان میدهد . همراه با این ویژگی های مثبت ، مجموعه منحصر به فردی از تکنیکهای جستجو، GA را بصورت یک جستجوگر قوی و تکنیک بهینه سازی مطرح میسازد.

مجموعه ای از تکنیکهای جستجو که توسط GA توسعه یافته اند با دیگر تکنیکهای بهینه سازی مانند

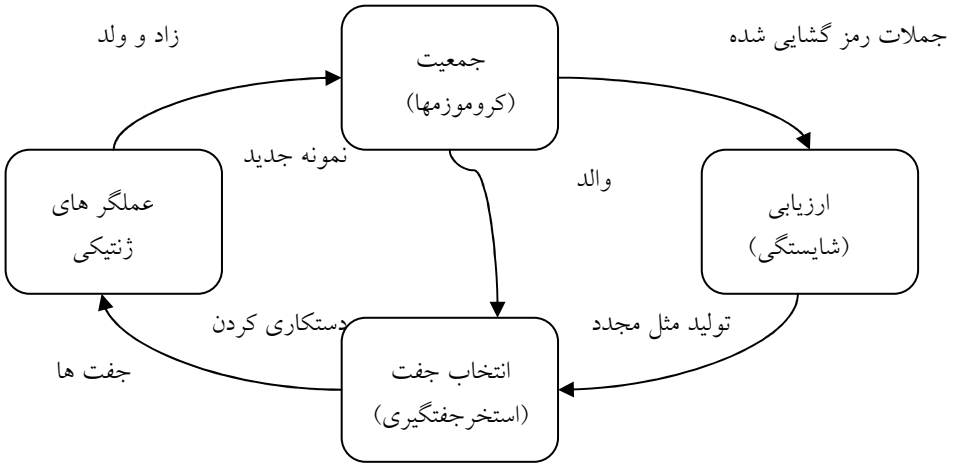
---

[25] در متون لاتین و بعضی از ترجمه ها به «زاده» اشاره شده

[26] Prof.j.Holland
[27] Self-repair
[28] Self-guidance

تکنیکهای محاسباتی[۲۹] , شمارشی , اتفاقی و تپه نورد[۳۰] متفاوت است .در زیر این تفاوت ها آمده است:

- GA ها با رمزگذاری پارامترها کار میکنند.
- GA ها در باره جمعیتی از موضوعات جستجو میکنند نه از روی یک موضوع منفرد.
- GA ها اطلاعات تابع هدف را بکار میگیرند و دانش منتقل شده از جایی نیستند.
- GA ها از قواعد انتقال احتمالی استفاده میکنند نه از قواعد قطعی.

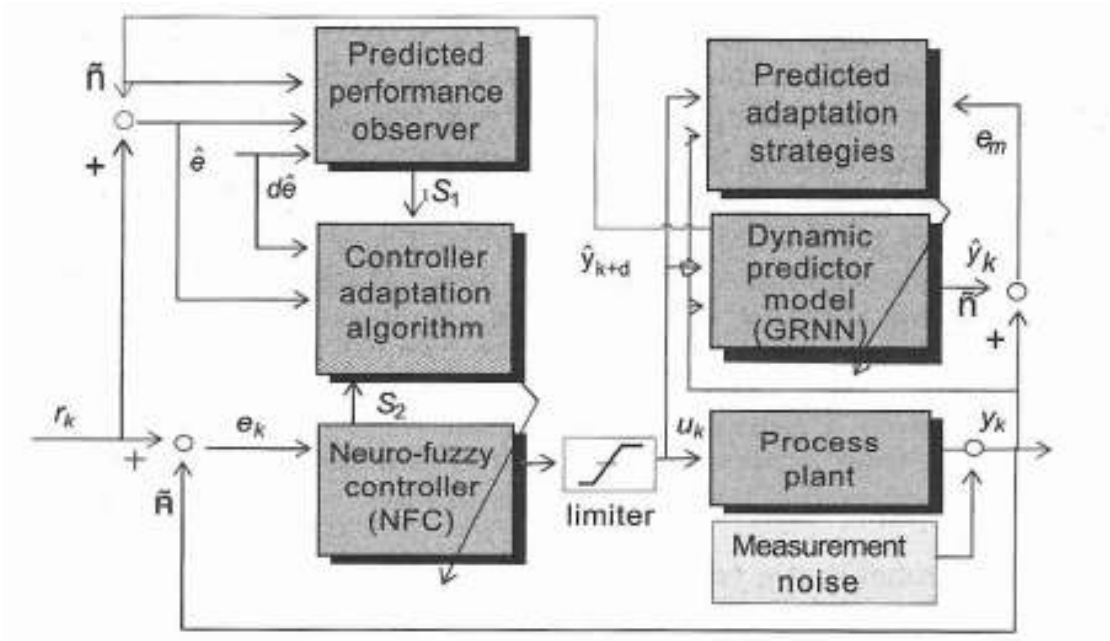


شکل ۱۲-۵ عملگرهای الگوریتم ژنتیک و چرخه حیات آن

کاربردهای موفقیت بسیاری از GA در زمینه هایی مانند مسیریابی سیم بندی در مدارهای چاپی ، طراحی مدارهای VLSI ، بازی کردن، شناسایی چهره وغیره وجود دارد.

رویکردهای جدیدی از ترکیب روشها در حال شکل گیری و توسعه است که نمونه ای از آن ,ترکیب تکنیک منطق فازی و شبکه های عصبی است که به آن نروفازی میگویند[۳۱] . تکنیک فازی ساخت یافته و قابل گسترش است اما قابلیت خودیادگیری ندارد از طرف دیگر، شبکه های عصبی، قابلیت یادگیری ذاتی را فراهم میکنند اما ساخت یافته نیستند. ترکیب این دو تکنیک حوزه ی پژوهشی جدیدی را در AI بوجود آورده. ترکیباتی مختلفی از رویکردهای شبکه عصبی با منطق فازی معرفی شده است. برای مثال، تابع پایه شعاعی (RBF) شبکه عصبی اجرا شده در کنترلر نوروفازی (NFC) است و شبکه عصبی رگرسیون عمومی (GRNN) مانند یک پیشگو در کنترل نوروفازی بکار رفته است . معماری این سیستم در شکل ۱۲-٦ نشان داده شده است.

[29] calculuse-based
[30] hill climbing
[31] neuro-fuzzy



شکل ١٢-٦ بلاک دیاگرام سیستم کنترل نوروفازی خود سازگار

با بکارگیری این معماری ، سیستم ***خود سازگار*** قادر به واکنش سریع نسبت به تغییرات در مجموعه است. وقتی که تغییر عمده ای در رفتار پویای دستگاه اتفاق می افتد , سیستم ***سازگار مجتمع شده*** پیشنهادی حتی در مفایسه با یک ***کنترلر پیشگویانه تعمیم یافته*** بسیار قوی عمل میکند.

تلاشهای دیگری در ترکیب تکنیک فازی و سیستمهای خبره بوجود آمده است که به آن ***خبره فازی***[٣٢] میگویند که موفقیت سیستمهای نوروفازی را نداشته اند.

١٢-٤-١ کاربردهای صنعتی AI

از جمله اولین کاربردهای موفقیت آمیز AI در صنعت ، در زمینه های دانش ماشین نمادین یا کاربردهای سیستمهای خبره بودند که در اواخر دهه ٦٠ و اوایل دهه ٧٠ آغاز شدند. با وجود آنکه تحقیق در شبکه های عصبی زودتر شروع شده بود، کاربردهای دنیای واقعی از چنین تکنیکهایی بی بهره بود زیرا این تکنیکها بر فعالیتهای تکراری استوار بود و فعالیتهای تکراری نیازمند سخت افزار قوی بود که در آنزمان وجود نداشت. دو مثال از کاربردهای اولیه ی سیستمهای خبره عبارتند از : MYCIN که برای تشخیص بیماری های عفونت خونی و در دانشگاه Stanford در سال ١٩٧٢ بوجود آمد و CADACUES برای تشخیص بیماریهای درونی که در دانشگاه Pittsburgh و در سال ١٩٧٠ توسعه داده شد. اولین کاربرد تجاری از سیستم خبره XCON است که توسط شرکت DEC و برای پیکربندی سیستمهای کامپیوتر VAX در ١٩٨٠ بکار برده شد.

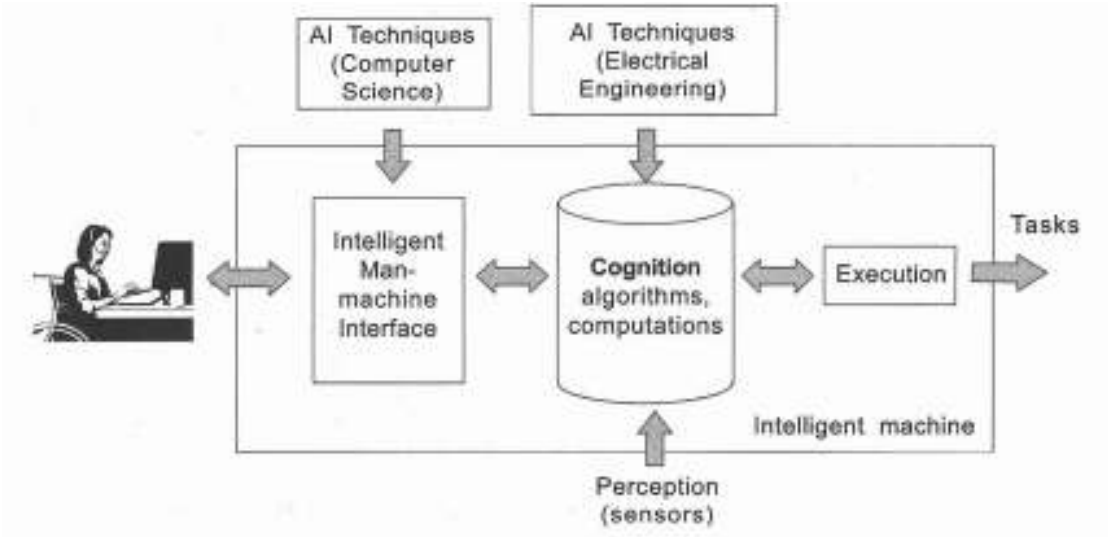
پیدایش ریزپردازنده ها باعث توجه صنایع بسیاری برای بکاربردن تکنولوژی منطق فازی در اکثر محصولات مصرفی و خانگی شده است. سازندگان ژاپنی از بکارگیری کنترولرهای فازی و حسگرها در

[32] fuzzy-expert

محصولات صنعتی خود سود فراوانی نسبت به سایر سازندگان در دنیا بدست می آورند

برای اولین بار شرکت ماتسوشیتا در سال ۱۹۸۹ ماشین ظرفشویی را بر اساس منطق فازی ساخت که فروش بسیار موفقیت آمیزی داشت و موجب روی آوری بسیاری از صنایع ژاپن به بکارگیری فناوری منطق فازی در تولیداتشان شد. امروزه پیداکردن محصولی که در آن از تکنولوژی منطق fuzzy استفاده نشده باشد، خیلی مشکل است.

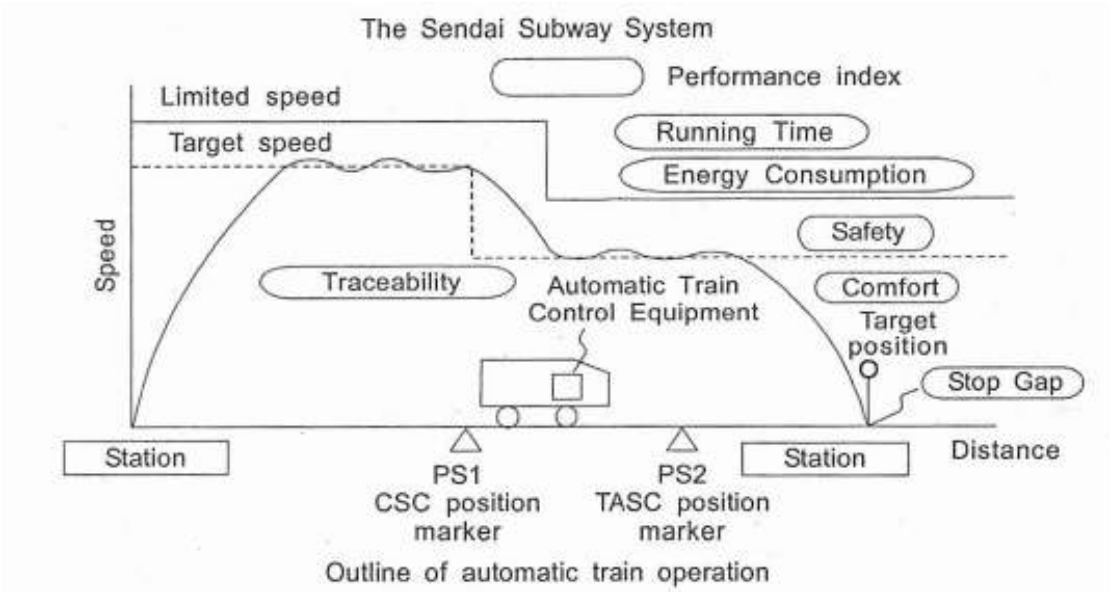
در اواسط دهه ۱۹۹۰ ، تکنولوژی شبکه عصبی با تکنولوژی فازی در بسیاری از محصولات بطور موفقیت آمیز بکار میرفت. اکثر کاربردهای تکنولوژی های نوروفازی در محصولات ، برای مرحله طراحی هستند که کنترلرها داخل سیستمها طراحی و گنجانده شده است. شکل ۱۲-۷ بلوک دیاگرامی از طراحی یک سیستم هوشمند را نشان میدهد که شامل ٤ جزء است: واسطه ماشین هوشمند، ادراک، شناخت و اجرا.



شکل ۱۲-۷ بلوک دیاگرام نشاندهنده ایده اصلی ماشین های هوشمند

تحقیق در AI با علوم دیگری مانند علم عصب شناسی ، علوم زیستی ، روانشناسی ، فیزیک و ریاضیات و علوم کامپیوتر در ارتباط است، اما نتیجه این تحقیقات به صورت گسترده تبدیل به محصولاتی در زمینه علوم کامپیوتر و مهندسی الکترونیک گردیده است. در موضوع سیستم های هوشمند یا طراحی ماشین و در حوزه ی علم کامپیوتر بخش بزرگی از تحقیق و توسعه (R&D) به بخش نرم افزار واسط ماشین هوشمند انسان نما متمرکز شده است . مثالهای R&D در این حوزه شامل پردازش زبان طبیعی ، سیستمهای خبره، واسط ماشینی ، استخراج داده ، استخراج متن و غیره هستند. در حوزه مهندسی الکترونیک ، تلاشها در R&D در طراحی بیشتر معطوف به تولید قطعاتی شده که به ماشین ها قدرت ادراک میدهد. برای مثال در طراحی محصولاتی مانند ماشینهای ظرفشویی ، پلوپز ، یخچالها از کنترلرهای نوروفازی که با بقیه سیستم بصورت یکپارچه درآمده اند استفاده شده است.

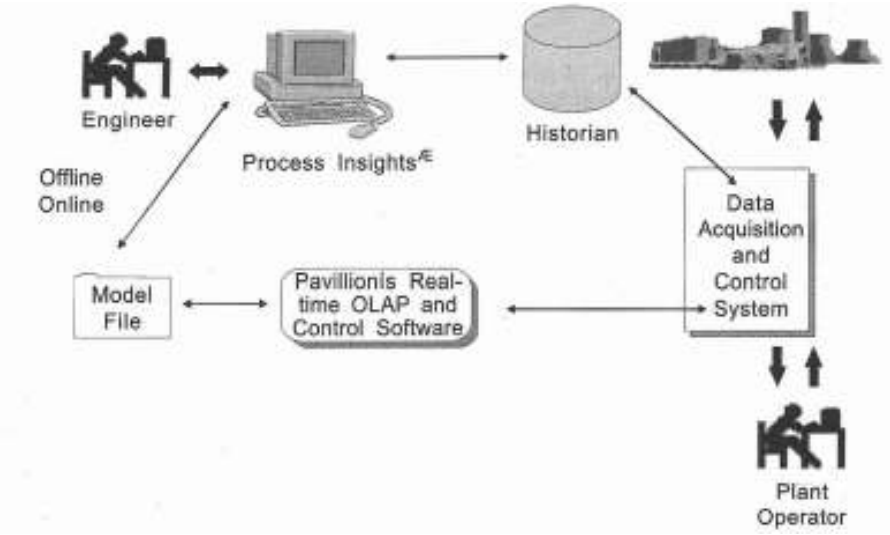
تکنولوژی AI در سیستمهای صنعتی نیز بکار میرود. هیتاچی در ژاپن قبل از سال ۱۹۷۸ کنترلر فازی قطار را ساخت اما بعد از ۳۰۰۰۰۰ شبیه سازی و آزمون در ۳۰۰۰ مسیر بدون مسافر و ۸ سال بعد اجازه بکارگیری در راه آهن ژاپن را دریافت کرد.



شکل ۱۲–۸ نمایش بازدهی استفاده از کنترلر فازی تولید شده توسط هیتاچی

شکل ۱۲–۸ مسیر پروفیل قطار فازی را به همراه ٦ شاخص نشان میدهد که بوسیله کنترلر فازی برای بهبود و مدل کردن بررسی و تحلیل شده اند را نشان میدهد .این ٦ شاخص عبارتند از: زمان اجرا، مصرف انرژی ، قابلیت ردیابی، ایمنی و راحتی .هیتاچی نشان داد که قطار با بکارگیری کنترلر فازی میتواند ۳ برابر بیشتر اجازه توقف داشته باشد (سوار و پیاده کردن مسافر) , ۲ بار قدرتش کاهش دهد و در کل نیروی توان مصرفی خود را در مقایسه با بکارگیری کنترلر PID ۱۰٪ کاهش دهد. موفقیت قطار فازی گسترش بسیاری از کاربردهای AI در صنعت را بدنبال داشت.

بر اساس الگوریتم ***انتشار خطای معکوس*** در شبکه عصبی و منطق فازی , شرکت آمریکایی پراویلیج تکنیکس، نرم افزاری را برای فرایند مدل کردن و بهینه سازی فرآینده ها با نام پروسس اینسایت به معنی بصیرت در فرآیند تولید کرد. این نرم افزار اطلاعات فراوانی را از ثبت کنند وقایع دریافت و با فناوری شبکه عصبی به مدل کردن کارخانه بصورت offline می پرداخت. منطق فازی برای مدیریت محدودیت ها در زمان برخط بودن سیستم و برای اهداف کنترل برخط مورد استفاده قرار گرفت. سالیانه بیش از یک میلیون دلار صرفه جویی حاصل بکارگیری نرم افزار فوق بود. شکل ۱۲–۹ دید دیاگرامی از نرم افزار اطلاعات فرایند را نشان میدهد.



شکل ۱۲-۹ برنامه پروسس اینسایت که ابتدا فرآیندها را مدل کرده و بعد اجرا میکند

اخیرا بسیاری از صنایع در ژاپن بر پژوهش در روباتهای شبیه انسان متمرکز شده اند. صنایعی مانند هوندا، میتسوبیشی، سونی ، فوجیتسو، NEC ، NTT ، پاناسونیک و غیره.

تکنولوژی اصلی در روباتهای انسان نما، بالابردن توانایی راه رفتن ، سخن گفتن و تشخیص تصویر ، و پیشرفت در موتورها و محرکها است. ASIMO (گام پیشرفته در ابداع تحرک) که توسط هوندا توسعه داده شده است ، با بالا رفتن و پایین آمدن از پله ها نشان داد که توانایی راه رفتن فوق العاده ای را دارند اما قابلیت جسمی شان هنوز در حد یک انسان ۵ ساله نیست.

در سه دهه ، دولت ژاپن ۵۰ میلیون ین (٤۰۰ میلیون دلار) برای توسعه ی یک روبات شبیه انسان با ظرفیت هوشی ، جسمی و احساسی کمتر از یک انسان ۵ ساله ، هزینه کرده است که به پروژه "Atom" معروف است. پژوهشگران معتقدند پروژه Atom از سریال انیمیشن مشهور "Tetsuwan Atom" که توسط Osamu Tezuka ساخته شده الهام گرفته است و مانند پروژه امریکایی آپولو که باعث نشستن انسان روی ماه شد و در مشارکت با فناوری های دیگر مرزهای فناوری جدید را گشوده است.

شکل ۱۲-۱۰ ASIMO به طرفی که انسان اشاره میکند حرکت میکند

۱ ! کدام روش حل مسئله AI را از سایر علوم کامپیوتر و مهندسی متمایز مینماید.

الف) روش الگوریتمی ب) روش ساده سازی ج) روش اکتشافی ۴) روش محاسباتی

۲- هوش مصنوعی با کدامیک از روشهای حل مسائل سر و کار دارد.

الف) هوش انسان ب) روشهای غیر سمبلیک ج) روشهای الگوریتمی د) سمبل گذاری و غیر الگوریتمی

۳- طراحی کدامیک از موارد زیر مربوط به AI است.

الف) هوش انسان ب) سیستمهای کامپیوتری هوشمند ج) سیستمهای پیچیده د)سیستمهای کامپیوتری غیر هوشمند

۴- کدامیک از موارد زیر وجه تمایز AI از سایر علوم کامپیوتر و مهندسی است.

الف) روش الگوریتمی حل مسئله ب) شناخت ج) روش اکتشافی حل مسئله د) هیچکدام

۵- تست تورینگ کارایی ..... را در برابر ..... اندازه می گیرد چون ..... بهترین و تنها استاندارد برای رفتار هوشمند است.

الف) انسان – ماشین هوشمند ! انسان ب) ماشین هوشمند – انسان – انسان

ج) انسان – انسان – ماشین هوشمند د) ماشین هوشمند – انسان – ماشین هوشمند

۶- – مهمترین انتقاد به تست تورینگ به دلیل استفاده از کدام روش حل مسئله است؟

الف) اکتشافی ب) غیر سمبلیک ج) غیر الگوریتمی د) سمبلیک

۷- کدام توانایی زیر در تست تورینگ آزمون نمی شود ؟.

الف) هوشمندی ب) شناخت ج) قدرت ادراک یا مهارت د) رفتار مشابه انسان

۸- .برنامه هوشمند DENDRAL با چه هدفی توسعه یافت ؟

الف) تقلید مکالمه با یک روانشناس ب) تشخیص ساختار شیمیایی مواد از طیف نگار جرم هر ماده

ج) برای تامین کردن اطلاع پایه د) بازی با انسان

۹ ! کدامیک از موارد زیر از قابلیت های ضروری برای هوشمندی نیست ؟

الف) معنادادن به پیام‌های مبهم یا نادرست ب) پیداکردن شباهت‌ها، ولو اینکه موقعیت‌ها متفاوت باشند

ج) اثبات قضیهٔ ریاضی د) پیداکردن شباهت‌ها، ولو اینکه موقعیت‌ها متفاوت باشند

۱۰- به منظور قبولی در تست تورینگ سیستم باید چه ویژگی داشته باشد؟

الف) فهم احساس انسان ب) درک هوشمندی ج) فهم زبان طبیعی د) درک شرایط محیطی

۱۱ -کدامیک از سیستمهای زیر نقش مشاور اتوماتیک را بازی میکنند ؟

الف) سیستم های خبره ب)سیستم های اکتشافی ج) سیستم های الگوریتمی د) سیستم های غیرالگوریتمی

۱۲- کدامیک از گزینه‌های زیر صحیح است؟

الف) دانش شرط لازم هوشمندی است　　ب)دانش شرط لازم و کافی هوشمندی است

ج)هوشمندی شرط لازم دانش است　　د)هوشمندی شرط لازم و کافی دانش است

۱۳- کدامیک از گزینه‌های زیر از ویژگی‌های مطلوب کمی دانش نیست؟

الف)حجم زیاد دانش　　ب)دشواری توصیف و تعیین دانش به صورت درست و دقیق

ج)تغییرات پیوسته دانش　　د)ثبات دانش ازمرحله داده تا مرحله سازمان دهی شده

۱۴- حوزه مهندسی دانش را می‌توان .....تعریف کرد.

الف)الحاق دانش خبره در یک سیستم　　ب)روش ارزیابی مسایل ، آموختن دانش و ایجاد یک سیستم دانش محور

ج)آشنایی با حوزه دانش تحت آزمایش　　د)شناخت منابع اضافی دانش

۱۵- کدامیک از گزینه‌های زیر از مراحل فراگیری دانش نیست؟

الف)تعیین دامنه دانش　　ب)نگاه جزئی به دامنه دانش　　ج)تعریف محدوده دانش　　د)انتخاب و کاربرد فن استخراج

۱۶- کدامیک از گزینه‌های زیر صحیح نیست؟

الف)استنباط، موضوعات منفرد دانش را که لازم است در تمام موارد یکسان سازمان دهی شده باشند را استخراج می‌کند.

ب)اینکه چه میزان سازمان دهی لازم است به روشی که بانک اطلاعات اجرا و استفاده شده بستگی زیادی دارد.

ج)پس از تعریف مرزهای دامنه، نگاه کلی به موضوع اصلی توصیه می‌شود.

د)تعریف مرزهای دامنه تضمین می کند هیچ گونه حذف عمده و ناخواسته در ارتباط با گردآوری اطلاعات در طی فرآیند استخراج دانش صورت نمی گیرد.

۱۷ - کدام مورد از منابع اصلی دانش نیست ؟

الف) مطالب نوشتاری　　ب)مثالها　　ج) قوانین فیزیکی　　د) افراد خبره

۱۸- کدامیک از گزینه‌های زیر از ارکان دانش نیست؟

الف)مطالب نوشتاری　　ب)قوانین علمی　　ج)تجربه　　د)الگوها

۱۹- کدامیک از گزینه‌های زیر صحیح نیست؟

الف)قویترین قوانین با قاعده که می‌شناسیم، قوانین علمی است.

ب)فراگیری دانش در زمینه علمی از زمینه های دیگر پیچیده‌تر است.

ج)در حوزه هایی که کمتر با اصول علمی تطبیق دارند، دانستن منابع و مراجع کسب دانش ضروری‌تر است.

د)پایگاه دانش سیستم‌های خبره کاربردی، امروزه اغلب فاقد فرمول علمی بوده و بیشتر متکی به تجربه افراد خبره است.

۲۰- کدامیک از گزینه‌های زیر از انواع دانش نیست؟

الف)دانش ادراکی شامل حقایق و روابط ساده.　　ب)مفاهیم و روابط بین آنها که اصول سیستم های خبره را تشکیل می‌دهد.

ج)نظریه یا روشهای شخصی برای حل مسائل.　　د)دانش استراتژیک که دانش روش دسته بندی مسائل و نحوه و چگونگی شروع حل یک مسئله است.

۲۱- کدامیک از گزینه‌های زیر صحیح است؟

الف)اغلب مسائل مهندسی، کاملاً اکتشافی هستند. ب)پایگاه دانشِ سیستم‌های خبره کاربردی اغلب فاقد فرمول علمی است.

ج)سیستم های خبره نیاز به نگهداری دانش ادراکی دارند. د)سودمندی یک سیستم خبره کاملاً به دانش ادراکی وابسته است.

۲۲- مسئله نمایش دانش به کدام مورد اشاهر شده در زیر مربوط می شود.

۱)انتخاب و کاربرد فن استخراج دانش ب)مفاهیم و روابطی که اصول سیستم های خبره را تشکیل می‌دهد

ج)قوانین با قاعده که می‌شناسیم د)عدم مطابقت بین حافظه بشر و حافظه کامپیوتر

۲۳- عملگرهای دستکاری بر روی پایگاه دانش راچه می نامند.؟

الف)موتور استنباط ب)هوشمندی ج)قوانین علمی د)دامنه دانش

۲۴- کدامیک از گزینه‌های زیر صحیح نیست؟

الف)استفاده از ساختارهای خاص و دانشی که در قالب مناسب توصیف شده‌اند، تاثیر مهمی بر حل مسائل دارد.

ب)هیچ نمایش یا مدل انفرادی نمی‌تواند همه جنبه‌های یک شئ واقعی را در برگیرد.

ج)یک موجودیت هوشمند باید طیف گسترده‌ای از نمایش‌ها و ارائه‌ها را جهت ارتباط با دنیا بکار گیرد.

د)برای حل مسئله در دنیای واقعی حداقل سه ارائه مجزا برای تطبیق و انتخاب از بین چند مکانیزم محاسبه‌ای داده شده لازم است.

۲۵- کدامیک از گزینه‌های زیر صحیح نیست؟

الف)قالب‌های رویه‌ای، دانش را به مثابه مجموعه‌ای از دستورالعمل‌ها برای حل یک مسئله نمایش می‌دهد.

ب)در قالب‌های رویه‌ای، از نمایش‌های اعلانی که به وسیله منطق و شبکه‌های معنائی فراهم شده‌اند استفاده می‌کند.

ج)قالب های نمایش منطقی، از عبارات موجود در منطق نمادین برای نمایش دادن پایگاه دانش بهره می گیرد.

د)در قالبهای نمایش منطقی، قوانین استنباطی و رویه‌های استدلالی این قالب ارایه دانش را برای حل مسائل شهودی مناسب کرده است.

۲۶- کدامیک از گزینه‌های زیر صحیح نیست؟

الف)قالب نمایش شبکه‌ای، دانش را به عنوان یک گراف در نظر می گیرد.

ب)در قالب نمایش شبکه‌ای، گره‌های گراف موضوع‌ها یا مفاهیم موجود در دامنه مسئله و یال‌های آن روابط یا وابستگی بین موضوع‌ها را نشان می‌دهند.

ج)قالب نمایش ساخت یافته، به شکلی شبکه ها را گسترش میدهد تا بتوانند در هر یال ساختار داده‌ای پیچیده‌ای را به نام «اسلات» را نگهداری کند.

د)در قالب نمایش ساخت یافته، مقداری که به هر اسلات نسبت داده میشود میتواند شامل رویه‌هایی برای اجرای وظیفه‌ای خاص باشد.

۲۷- کدامیک از قالب‌های نمایش، سیستم‌های خبره قاعده‌گرا را بنیان نهاده است؟

الف)قالب نمایش منطقی ب)قالب نمایش رویه‌ای ج)قالب نمایش شبکه‌ای د)قالب نمایش ساخت یافته

۲۸ ! کدام عبارت زیر در مورد قالبها صحیح عنوان نشده است ؟

الف) قالب ها شبکه های معنائی را از راههای مختلف گسترش می دهند مهم ترین آنها سازمان دهی دانش در ساختارهاست و این موضوع مهمی برای پایگاه دانش است .

ب) الحاق رویه ای ، تا وقتیکه دانش معین با نمایش ها به خوبی وفق داده نشده اند ، یک خصوصیت مهم ویژه از قالبها است .

ج) نمایش دادن دانش با سیستم قالب ، اگرچه دارای اطلاعات ناتمام می باشد ، حداقل تاحدی به ما اجازه استدلال کردن و استنتاج سریع حقایقی که به طور صریح مشاهده و آشکار نشده اند را می دهند.

د) یکی از مشکلات نمایش قالبی ، دشواری تعیین الگوریتم خطا برای یک قالب می باشد .

۲۹- کدامیک از گزینه‌های زیر صحیح نیست؟

الف)یک سیستم خبره یک برنامه دانش‌گرا یا دانش‌محور است که راه حل‌هایی برای مسائل موجود در حوزه خاص و با «کیفیت خبره» فراهم می کند.

ب)قالب‌های نمایش رویه‌ای مشتمل بر جبر گزاره‌ای و مسندی است که زبان‌های نمایش برای AI هستند

ج)از نظر تئوری های ارتباطی، مفهوم هر شی در ذهن یا پایگاه دانش در شبکه‌ای و با پیوند با دیگر اشیاء تعیین می‌شود.

د)شبکه معنائی، جایگزینی برای منطق گزاره به عنوان یک شکل و فرم از نمایش دانش است.

۳۰ ! کدام مورد از مزایای ارایه سمبولیک نیست ؟

الف) تحمل‌پذیری بیشتر نسبت به نوفه یا نویزدارند

ب) آگاهی و دانش با جملاتی به زبان رسمی بیان می‌شود

ج) سازندهٔ سیستم می‌تواند چیزی را که سیستم می‌داند بخواند

د) خواندن ارائه و فهمیدن معنی دانش امکان‌پذیر است

۳۱- کدامیک از گزینه‌های زیر صحیح نیست؟

الف)شبکه‌های معنائی و گرافهای ادراکی، دو نوع قالب نمایش شبکه ای هستند.

ب)یک گراف ادراکی، یک گراف متناهی، متصل و دو قسمتی است.

ج)شبکه‌های معنائی نمی‌توانند جایگزینی برای منطق گزاره به عنوان یک شکل و فرم از نمایش دانش باشند.

د)گراف‌های ادراکی از یال‌های برچسب دار استفاده نمی کنند

۳۲ ! طبق نظریه مایلوپولس و لوسک کدامیک از قالب های زیر برای ارایه دانش مطرح نیستند.

الف) قالب نمایش منطقی ب) قالب نمایش رویه ای ج) قالب نمایش شبکه ای د) قالب نمایش شی ء گرا

۳۳- کدامیک از گزینه‌های زیر صحیح است؟

۱)کامپیوتر با توجه به قالب می‌تواند توجه خود را فقط بر روی جنبه‌هایی از مسئله داده شده که به راه حل مربوط است، معطوف کند.

۲)هرچه وظایف پیچیده‌تر می‌شوند، لازم است که ارائه و نمایش آنها ساخت یافته‌تر باشد.

۳)نظریه مجموعه‌ها پایه و اساس خوب و مناسب برای درک سیستم‌های قالب را فراهم می کند.

۴)هر سه مورد

۳۴ ! کدام جمله در خصوص اسکریپت ها صحیح نیست ؟

الف) اسکریپتها توانائی جهت پیشگوئی وقایع را دارند

ب) اسکریپتها عمومیت بیشتری نسبت به قالبها دارند

ج) ممکن است اسکرپتها برای ارائه کلیه انواع دانش مناسب نباشد

د) در اسکریپتها تفسیر منسجم انفرادی از مجموعه ای از مشاهدات ساخته میشود

۳۵- کدامیک از گزینه‌های زیر صحیح است؟

الف)اسکریپت نوعی ارائه ساخت یافته است که یک رشته حوادث و وقایع کلیشه‌ای از حقایق را در یک محتوای مخصوص توصیف می کند.

ب)اسکریپت برای سازمان دهی ساختارهای وابستگی مفهومی به توصیف وضعیتهای متعارف طراحی شده‌اند.

ج)اسکریپت در سیستم‌های درک زبان‌های طبیعی، به عنوان اجزای پایگاه دانش استفاده شده‌اند.

د)هر سه مورد.

۳۶- کدامیک از گزینه‌های زیر از اجزای اسکریپت نیست؟

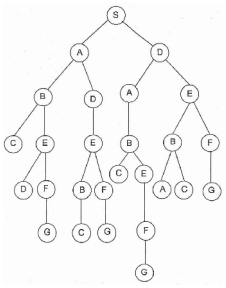
الف)اسلات که میتوان مقداری را به آن نسبت داد. ب)اثاثیه یا هر چیزی که محتوای اسکریپت را درست می کند.

ج)نقش‌ها یا اعمالی که شرکت کنندگان فردی آنها را انجام می‌دهند د)صحنه‌ها که هر کدام از آنها یک خصوصیت موقتی از اسکریپت را ارائه می‌دهد.

۳۷ ! کدامیک از موارد زیر جزو اجزاء سیستم های جستجو نیست ؟

الف)پایگاه داده ب) استدلال پسین ج)عملگرها د) استراتژی کنترل

۳۸ - شکل روبرو نمایی از کدام روش جستجو است



الف) کورکورانه

ب) سطحی

ج) مین مکس

د) تپه نوردی

۳۹ - کارایی یک جستجو در پاسخ به کدامیک از سوالات زیر ارزیابی میگردد ؟

الف) آیا این جستجو راه حلی پیدا می کند؟ ب) آیا راه حل بدست آمده بهترین پاسخ است؟

چ)هزینه جستجو از نظر زمانی و حافظه مورد نیاز برای یافتن راه حل چیست؟ د) هر سه مورد

۴۰ !پاسخ به سوال "در صورت وجود راه حل آیا استراتژی قادر به یافتن آن است ؟" کدام معیار در استراتژی جستجو را تعیین میکند

االف) بهینه بودن ب) کامل بودن ج) پیچیدگی زمانی د)پیچیدگی فضا

۴۱- پاسخ به سوال "در صورتیکه راه حلهای متفاوتی وجود داشته باشد ، آیا استراتژی راه حل باکیفیت بالا را می یابد ؟" کدام معیار در استراتژی جستجو را تعیین میکند

االف) بهینه بودن ب) کامل بودن ج) پیچیدگی زمانی د)پیچیدگی فضا

۴۲- پاسخ به سوال "چه میزان حافظه برای جستجو صرف خواهد شد " کدام معیار در استراتژی جستجو را تعیین میکند

االف) بهینه بودن ب) کامل بودن ج) پیچیدگی زمانی د)پیچیدگی فضا

۴۳ - درکدام جستجو است که استراتژی هیچ اطلاعی از تعداد مراحل یا هزینه جستجو ندارد و تنها قادر به تشخیص دادن حالت هدف از غیر از حالت هدف است

الف) کورکورانه ب)الفا- بتا ج)A*´ د) حریصانه

۴۴ ! کدام مطلب در خصوص جستجوی کورکورانه صحیح نیست

الف) کامل است ب) بهینه است

ج) منابع زیادی مصرف میکند د) بررسی آن فقط ارزش تئوریک دارد

۴۵ ! کدامیک از روش های جستجوی زیر بیشتر در بازی ها بکار میرود

الف) سطحی ب) مین مکس ج) تپه نوردی د)حریصانه

۴۶ ! کدام جمله در خصوص جستجوی حریصانه صحیح است ؟

الف) مانند جستجوی عمقی است، اما زمانی که در پیمودن یک مسیر به بن‌بست می‌رسد، برمی گردد.

ب) این جستجو بهینه و کامل نیست.

ج) جستجوی حریصانه تمام گره‌ها را در حافظه نگه می‌دارد، بنابراین پیچیدگی فضای آن مشابه پیچیدگی زمانی آن است.

د) هر سه مورد

۴۷ ! کدام جمله در خصوص جستجوها صحیح نیست ؟

الف) در جستجوی حریصانه هزینه رسیدن به هدف با استفاده ازتابع کشف کننده کاهش می یابد.

ب) جستجوی حریصانه میتواند زمان جستجو را کاهش دهد اما نه کامل است نه بهینه.

ج)در جستجو با هزینه یکسان هزینه مسیر حداقل بوده و هم بهینه هست هم کامل .

د) جستجوی A* از ترکیب جستجوی حریصانه و تپه نوردی بوجود آمده است

۴۸ ! کدامیک از موارد زیر از انواع گرامرمحصوب میشوند؟

الف) گرامر نامحدود (unrestricted) ب)گرامر با قاعده (regular) ج)مستقل از متن (context free) د)هر سه

---

۴۹ ! کدام توضیح در خصوص پارسر صحیح است

الف) پارسر یک الگوریتم برای تحلیل یک جمله با گرامر معلوم است

ب) پارسر یک ساختار توصیفی برای جملات صحیح تولید می کند

ج)در پارسر پذیرنده در پاسخ به تحلیل یک جمله با گرامر, فقط نتیجه آری یا نه بازگرداننده میشود

د) هر سه

۵۰ ! در مورد حوزه کاربردهای پردازش سیگنال کدام جمله صحیح نیست ؟

الف) فهم , تشخیص و شناسایی کلام از کاربردهای مهم پردازش سیگنال است

ب) تشخیص کلام عمل نگاشت از سیگنالهای صوتی دیجیتالی شده به رشته ای از کلمات می باشد

ج) همه زبانهای بشری , ترکیبی از ۴۰ تا ۵۰ صوت متمایز می باشند که phone نامیده می شوند

د) برای تشخیص کلام ابتدا باید آنرا فهمید سپس از یک بانک اطلاعات مفاهیم را استخراج کنیم

۵۱ - کدامیک از گزینه‌های زیر در مورد PROLOG صحیح نیست؟

الف)عبارت "C"(X,Terminal,Y) به این معنی است که ترمینال هد یا سر X, و دنباله آن Y است.

ب)عبارت "C"(X,Terminal,Y) به این معنی است که X هد یا سر ترمینال, و دنباله آن Y است.

ج)عدد 123+ به صورت لیست [+,1,2,3] تبدیل خواهد شد.

د)عدد 1.23 به صورت لیست [1,".",2,3] تبدیل خواهد شد.

۵۲- کدامیک از گزینه‌های زیر در مورد الیزا صحیح نیست؟

الف)الیزا تلاشی است برای بازی کردن نقش یک روانپزشک.

ب)الیزا تنها با علم روانپزشکی کار می‌کند.

ج)الیزا لیستی از قالب‌ها یا الگوها را نگهداری می‌کند.

د)الیزا سعی می‌کند سوال‌های ورودی را با لیستی از کلمات کلیدی طبقه بندی شده یا منظم تطبیق دهد.

۵۳- در الیزا تکنیک hashing با کدام هدف به کار برده می‌شود؟

الف)بازیابی سریع و تطبیق الگوها ب)Fix و grammar

ج)انتخاب تصادفی از یکی از کلمات کلیدی د))نگهداری تاریخچه محدود شده

۵۴ ! کدامیک از موارد زیر جزو عوامل موثر در حل کارآمد مسئله در سیستم های خبره نیست ؟

الف) دانش قابل اجرا , صحیح و قابل تفکیک و تمایز. ب) حذف سریع دیدگاه های غیر ثمر بخش

ج) منابع دانش غیر مشترک د) تقسیم راه حل ها به سطوح متفاوتی از تجرد

۵۵-قابلیت اداره کردن داده های گم شده یا غیر قطعی در کدام نوع برنامه نویسی صورت می گیرد؟

الف)برنامه نویسی مبتنی بر دانش ب)برنامه نویسی شی گرا ج) برنامه نویسی رویه ای د) برنامه نویسی چند لایه

۵۶-کدامیک از اجزاء اصلی سیستم مبتنی بر دانش نیست ولی جزء سیستم رویه ای می باشد؟

الف)مسئله ب) کنترل ج) الگوریتم د) داده

۵۷ -سیستم خبره چگونه استدلال می کند؟

الف)با تغییر و دستکاری الگوریتمها ب)با تغییر و دستکاری روند کنترل داده ها

ج) با تغییر و دستکاری داده ها د) با تغییر و دستکاری سمبولها

۵۸-منابع دانش خصوصی کدام اند؟

الف)کتابها ب)متخصصان ج)مقالات تحقیقی د)روزنامه ها

۵۹- کدامیک از عوامل موثر بر یک راه حل کارآمد نمی باشند؟

الف)دانش قابل اجرا , صحیح و قابل تفکیک و تمایز ب)جمع آوری سریع دیدگاه های ثمر بخش

ج)منابع دانش مضاعف مشترک د)تقسیم راه حل ها به سطوح متفاوتی از تجرد

۶۰-کدام مورد زیر از مشکلات اساسی در رهیافت مبتنی بر دانش نیست؟

الف)دانش حجیم و غیر قابل پردازش ب)وجود رویه های پیچیده جهت حذف احتمالات

ج)مسائلی که به صورت پویا تغییر می کنند د)وجود امکان های مختلف برای ارزیابی کردن

۶۱-کدامیک از اجزای یک حل کننده مساله ایده آل نیست؟

الف) دانش برای چک کردن ثبات یک راه حل پدیدار شده ب) دانش برای ارزیابی راه حل های کلی

ج) پردازنده زبان برای مکالمات مبتنی بر مسائل د) دانش برای برنامه ریزی استراتژی راه حل مسئله بعدی

۶۲- کدامیک از موارد زیر جزو ایده های اساسی در حل مسایل هوشمند نیست؟

الف)ساخت راه حلها بصورت مرحله ای و پی در پی ب) ساخت راه حلهای کارآمد با انتخابی از فضای پیشنهادات ممکن

ج) شناسایی راه حل های مفید و سپس کاوش بیشتر در آنها د)هرس کردن راه حل تا رسیدن به بهترین راه حل

۶۳-کدامیک از موارد زیر بیشتر در سیستمهای پردازش داده دیده می شود تا در سیستمهای پردازش دانش؟

الف)کشف کنندگی ب) پردازش تکراری ج) پردازش های استنتاجی د)کنترل گسترده و مقدار کم داده

۶۴-کدامیک از خصوصیات اولیه یک سیستم خبره نیست؟

الف) استفاده از افراد متخصص ب)اجرای استدلال سمبولیک ج) استفاده از قوانین پیچیده د)استفاده از الگوریتمهای پیچیده

۶۵- کدام یک از موارد زیر از ضروریات یک حل کننده مسئله ایده آل نیست؟

الف) دانش در مورد یک حوزه که شامل حقایق , کشف کنندگی و عقاید است ب) دانش برای یافتن راه حل های ناپایدار

ج) دانش برای برنامه ریزی استراتژی راه حل مسئله بعدی د) دانش برای ارزیابی راه حل های جزئی

۶۵ ! کدامیک از موارد زیر جزو مراحل اکتساب دانش نیست ؟

الف)شناسایی ذی نفعان ب) شناسایی اهداف ج) ادراک د)رسمی سازی

۶۶ ! کدامیک از امکانات زیر توسط پوسته ها یا شل ها پیاده سازی نمیشود؟

الف) زبان بازنمایی دانش ب) یک ویراستار پایگاه دانش

ج) امکانات ردیابی و اشکالزدایی د) یک بانک اطلاعاتی دینامیک

۶۷ ! کدام جمله در خصوص عاملها صحیح است ؟

الف) عاملها نرم افزارهایی هستند که در زمینه خاص مهارت دارند ب) عاملها ابزار توسعه همه سیستمها چه هوشمند و چه غیر هوشمند هستند

ج) عاملها ابزارهای ورودی و خروجی سیستمهای هوشمند هستند د) عاملها دانش ارایه عملیات صحیح هستند

۶۸ ! کدام عبارت جزوه مسائل جدی در بازیابی، اداره کردن و استفاده مجدد از اطلاعات ذخیره شده در وب نیست

الف) امنیت دسترسی به اطلاعات در وب همیشه برقرار نیست ب) اطلاعات بکار گرفته شده توسط شرکتها ممکن است بعد از مدتی تغییر کند.

ج) دسترسی به منابع اطلاعات همیشه امکانپذیر نیست. د) خریدهای ارزان ارائه شده توسط شرکتها ممکن است بعد از مدتی تغییر کرده باشد.

۱۹ ! کدام جمله در خصوص عاملها صحیح است ؟

ب) عاملها ابزار توسعه همه سیستمها چه هوشمند و چه غیر هوشمند هستند

ج) عاملها ابزارهای ورودی و خروجی سیستمهای هوشمند هستند

*الف) عاملها نرم افزارهایی هستند که در زمینه خاص مهارت دارند

د) عاملها دانش ارایه عملیات صحیح هستند

تشریحی

۱ - پنج فعالیتی را نیازمند هوشمندی است نام ببرید

۱- تولید و درک گفتار ۲- تشخیص الگو ۳- حرکت در یک فضای پر از مانع دینامیکی ۴- اثبات قضیه ریاضی ۵- استدلال

۲ - مزایای ارائه سمبلیک چیست؟

۱- سازنده سیستم میتواند چیزی را که سیستم میداند بخواند. ۲- آگاهی و دانش با جملاتی به زبان رسمی ارائه شده است.

۳- خواندن ارائه و فهمیدن معنی دانش امکان پذیر است.

۳ - تست تورینگ را توضیح دهید.

این تست که تورینگ آنرا بازی تقلید نامیده ماشین و انسان را در کنار هم و در اتاقی جدا اتاقی جدا از انسان سوم که آنرا محقق مینامد قرار داد. محقق قادر نیست با با هیچکدام از آن دو صحبت کند و نمیداند کدام موجودیت واقعا ماشین است و تنها با استفاده از یک ترمینال متنی میتواند با آنها ارتباط برقرار کند.از محقق خواسته شده از طریق ترمینال متنی سوالاتی برای هر دو مطرح کند و بر اساس جوابهایی که دریافت کرده کامپیوتر را از انسان تشخیص دهد.اگر محقق نتواند ماشین را از انسان تشخیص دهد آنگاه تورینگ استدلال میکند که ماشین میتواند هوشمند در نظر گرفته

شود.

۴ - ویژگیهای تست تورینگ را توضیح دهید.

۱- این تست یک مفهوم معقول از هوش به ما میدهد (رفتار یک موجود هوشمند شناخته شده در پاسخ به یک مجموعه از سوالات).

این تست یک استاندارد برای معین کردن هوش ارائه میدهد که از بحث های غیر اجتناب روی طبیعت درستی اش پرهیز میکند.

۲- این تست ما را از منحرف شدن از مسیر اصلی با سولات گیج کننده و غیر پاسخگویی منع میکند.در همه حال کامپیوتر از پردازشهای داخلی مناسب استفاده میکند و به هر حال ماشین واقعا از اعمالش آگاه است.

۳- این تست با وادار کردن محقق تنها با تمرکز کردن روی محتوای پاسخ سوالات هر محرک تشخیصی را که قابلیت تشخیص موجود زنده را فراهم میکند حذف میکند.

۵- کاربردهای هوش مصنوعی را نام ببرید.

۱- پردازش زبان طبیعی ۲- بازیابی هوشمند از پایگاه داده ۳- سیستم های خبره ۴- اثبات قضیه ۵- رباتیک

۶- مسائل زمانبندی و ترکیبی ۷- مسائل ادراکی ۸- معماریهای وابسته به سلسله اعصاب ۹- بازی کردن (game)

۶ - اهداف AI چیست؟ (۵ مورد)

۱- فهمیدن درک انسان (مثلا چگونه انسانها مسائل را حل میکنند- سعی در به دست آوردن دانش عمیق حافظه انسان- تواناییهای حل مسئله آموزش و تصمیم گیری و ...)

۲- خودکارسازی صرفه اقتصادی انسانها را در وظایف هوشمند جایگزین میکند.

۳- هوش فوق بشری برنامه هایی میسازد که از هوش انسان تجاوز میکند.

۴- خودمختاری سیستم های هوشمندی دارد که عامل بر روی ابتکار خویش است.

۵- ذخیره اطلاعات و دانستن چگونگی بازیابی آن.

۷- موضوعات کلیدی که یک طراح سیستم AI با آنها روبرو می شود را نام برده و به طور مختصر توضیح دهید

فراگیری دانش ,ارائه دانش ,دستکاری دانش

فراگیری دانش عبارت است از تغییر شکل راه حل مساله که ناشی از وجود چند منبع دانش که راه حل بالقوه در آنها وجود دارد و به یک برنامه AI منتقل شده است. غالباً این فرآیند به انتقال و استخراج دانش از منابع متنوع و نمایش آن در یک غالب مناسب اشاره میکند. دستکاری دانش اصولا به واسطه استنتاج و استنباط و در اکثر مواقع استراتژی کنترل جستجوگرا یا موتور استنباط رخ می دهد.

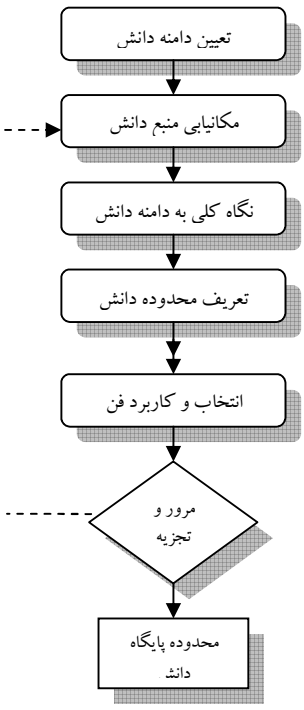
۸ - پنج خصوصیت ضروری برای هوشمندی را ذکر نمایید

۱- پاسخ دادن به موقعیت های از قبل تعریف نشده با انعطاف خیلی بالا. ۲- معنی دادن به پیامهای مبهم یا نادرست.

۳- اختصاص دادن اعتبار نسبی به عناصر برای یک موقعیت. ۴- پیدا کردن شباهت ها ولو اینکه موقعیتها متفاوت باشند.

۵- درک تمایز بین موقعیتها ولو اینکه شباهت های بسیاری بین آنها وجود داشته باشد.

۹ - روش فراگیری دانش را در غالب یک فلوچارت تشریح نمایید



۱۰ ! سه منبع و سه رکن دانش را نام ببرید

منابع : مطالب نوشتاری , افراد خبره , مثال ها.

ارکان : قوانین علمی , تجربه ,الگوها

۱۱- انواع دانش مورد نیاز برای ارایه سیستم های AI را معرفی کرده به طور مختصر توضیح دهید

اشیاء : موجودیت‌های دنیا که درباره آنها اطلاعاتی داریم

رخدادها : اعمالی که در دنیای ما رخ می دهد

اجرا : رفتاری شبیه نواختن گیتار که با دانشی درباره نحوه اجرای کارها درگیر است.

فرادانش : دانش درباره چیزهایی که می‌شناسیم

حقایق : واقعیت‌هایی درباره دنیای واقعی و چیزی که ما نمایش می‌دهیم و به آن «سطح دانش» می‌گوییم.

ارائه حقایق : آنچه که دستکاری می‌کنیم

۱۲- یادگیری را از نظر AI توضیح دهید

یادگیری به معنای فراگیری دانش است. فراگیری بیش از افزودن یک حقیقت جدید به پایگاه دانش است. در این فرآیند دانش جدید به پایگاه دانش افزوده شده است و دانشی که پیش تر حاصل شده تصفیه یا اصلاح می‌گردد.

۱۳- قالب نمایش منطقی را توضیح دهید

این دسته از نمایش‌ها از عبارات موجود در منطق نمادین برای نمایش دادن پایگاه دانش بهره می‌گیرد. قوانین استنباطی و رویه‌های استدلالی این قالب ارایه دانش را برای حل مسائل شهودی مناسب کرده است. ریاضیات اخباری مرتبه اول در قالب ارایه منطقی بسیار بکار میرود. پرولوگ یک زبان برنامه نویسی ایده ال برای پیاده سازی و اجرای قالبهای نمایش منطقی است.

۱۴- قالب نمایش ساخت یافته را توضیح دهید

زبان های نمایش ساخت یافته، به شکلی شبکه‌ها را گسترش میدهند تا بتوانند در هر گره ساختار داده‌ای پیچیده‌ای را به نام «اسلات» نگهداری و مقادیری را به آن نسبت دهند. این مقادیر ممکن است ارقامی ساده یا داده نمادین، اشاره‌گرهایی به دیگر قالب‌ها یا حتی رویه هایی برای اجرای وظیفه‌ای خاص باشند. مثال‌هایی از نمایش‌های ساخت یافته شامل اسکریپت‌ها و قالب‌ها و اشیاء است.

۱۵ – شبکه معنایی را تعریف و ۴ اصل آنرا نام ببرید

شبکه های معنائی ، جایگزینی برای منطق گزاره به عنوان یک شکل و فرم از نمایش دانش هستند . این ایده بر این اصل استوار است که ما می توانیم دانش خود را در در باره موجودات هستی در یک گراف با گره هائی که بیانگر اشیاء و یالهائی که نشاندهنده روابط بین این اشیاء هستند ، بگنجانیم .

بعضی از اصول شبکه های معنائی به شرح ذیل هستند :

- شبکه های معنائی روابط بین اشیاءی را که در گره ها قرار دارد را وصف و تشریح می کنند .
- گره ها دایره های نامگذاری شده هستند.
- ارتباطات بین گره ها به وسیله یالهائی که به این دوایر وصل هستند نمایش داده شده اند .
- یک شبکه معنائی میتواند برای تولید ساختارها و اشیاء استفاده شده باشد .
- یک شبکه معنائی می تواند برای تولید قواعد یک پایگاه دانش استفاده شده باشد .

۱۶- عملیات روی گراف‌های ادراکی را توضیح دهید

عملیات روی گراف‌های ادراکی، به ما اجازه ایجاد یک گراف جدید از یک گراف موجود را می دهند. این کار به وسیله چهار عملیات که کپی کردن، محدود کردن، متصل کردن و مختصر کردن نام دارند، انجام می‌گیرد.

قاعده کپی به ما اجازه شکل دهی یک گراف جدید، که یک کپی از گراف اولیه است را می‌دهد.

قاعده محدودیت به گره‌های مفهومی در یک گراف اجازه می‌دهد تا جایگزین گرهی شوند که ویژگیهای آنرا نمایش می‌دهد.

قاعده متصل کردن، به ما اجازه ترکیب دو گراف در قالب یک گراف منفرد را می‌دهد.

اگر یک گراف محتوی دو رابطه تکراری باشد، آنگاه ممکن است یکی از آنها حذف شود. این قاعده، مختصر کردن نام دارد.

۱۷ ! شش گام در الگوریتم شناسایی چهره را نام ببرید

۱) با داشتن یک مجموعه از تصاویرازمایشی چهره ها, M’ تا از بزرگترین eigen vectorها را محاسبه کنید:$E_1,E_2,\ldots E_{M'}$.

۲) برای هر فرد در مجموعه ازمایشی , اصل همبستگی را با شخص در ان eigen space محاسبه کنید .

۳) با داشتن تصویر ازمایشی ,I_{test} , ان را به وسیله محاسبه W_{test} توسط فرمول بالا به یک M’ eigen space بعدی تبدیل کنید.

۴) نزدیکترین چهره از تصاویر ازمایشی به تصویر مورد نظر را بیابید:

۵) فاصله تصویر مورد ازمایش را از eigen space بیابید.

۶) اگر dffs<Threshold1

- تصویر ازمایشی به اندازه کافی به eigen space شبیه است. در مقایسه با کل تصاویر می توان مطمئن بود که این تصویر یک چهره است نه چیز دیگر.

سپس اگر d< Threshold2

- می توان I_{test} را با عنوان تصویری که شامل چهره شخص k ام است دسته بندی کرد , زمانیکه k نزدیکترین چهره در eigen space به W_{test} است.

در غیر این صورت

- تصویر I_{test} را به عنوان شخص نا شناخته دسته بندی کن.

در غیر اینصورت

- تصویر I_{test} را به عنوان تصویری که شامل چهره نیست دسته بندی کن.

۱۷- چگونگی استفاده از تکنیک hashing در الیزا را توضیح دهید؟

تکنیک hashing برای بازیابی سریع و تطبیق الگوها به کار برده می شود. برای مثال در اجرای یونیکس، ابتدا تطبیق در دو کاراکتر اول انجام می شود. اگر هیچ الگوی تطبیقی وجود نداشته باشد سپس یک جستجو با استفاده از کلمات کلیدی در جملات ورودی انجام می شود. این کلمات کلیدی مرتب شده‌اند.

۱۸- اگر هیچ کلمه کلیدی در جمله نباشد که با کلمات کلیدی در پایگاه داده مطابقت داشته باشد، الیزا چگونه واکنش نشان میدهد؟

تعدادی ملاحظات یا تبصره‌های غیرالزامی استاندارد می‌تواند ایجاد شود. همچنین یک رهیافت جایگزین، انتخاب تصادفی از یکی از کلمات کلیدی است که قبلا استفاده شده. رکوردی از چهار کلمه کلیدی که قبلا استفاده شده نگهداری می‌شود. بنابراین در اینجا سیستم از تاریخچه محدود شده استفاده می‌کند.

۱۹-تفاوتهای پایگاه داده وپایگاه دانش را بنویسید؟

پایگاه دانش	پایگاه داده
اطلاعات در سطح بالاتری از تجرد قرار دارند	مجموعه ای از داده های نمایش دهنده حقایق
بیشتر بر روی کلاسی از اشیا عملیات انجام می دهد تا یک شئ واحد	فقط بر روی یک شئ واحد عمل می کند
از قدرت استنتاجی بهره مند است	اطلاعات باید صراحتا" توصیف شده باشند
بازنمایی به وسیله منطق , قوانین یا فریم ها یا مستندات یا شبکه های معنایی انجام می شود	به صورت سلسله مراتبی یا ارتباطی یا براساس مدل شبکه نمایش داده می شود
مورد استفاده برای تحلیل داه ها و برنامه ریزی	برای اهداف عملیاتی ابقا می شود

۲۰-تفاوت نرم افزار مبتنی بر دانش را با نرم افزار معمولی بنویسید؟

نرم افزار مبتنی بر دانش	برنامه نویسی معمول
قابلیت اداره کردن داده های گم شده یا غیرقطعی	نیازمند داده های صحیح است
تصمیم گیری ترتیب وتوالی توسط موتور استنتاج	ساختار رویه ای ثابت
مناسب برای تغییر سیمبولها	مناسب برای پردازش عددی
واسط های زبان طبیعی	فقط یک برنامه نویس آن را می فهمد
امکان توضیح در حین اجرا	توضیح در حین اجرا غیر ممکن
سیستم خبره = مسئله + کنترل + داده	برنامه = الگوریتم + داده

۲۱-یک تعریف از سیستم خبره ارایه دهید؟

یک برنامه کامپیوتری هوشمند است که از دانش و رویه های استنتاج برای حل مسائل دشواری که نیازمند کارشناسان خبره هستند, استفاده می کند .دانش مورد نیاز برای اجرا در چنین سطحی به همراه رویه های استنتاج مورد استفاده, می توانند به عنوان مدلی از متخصصین و کارشناسان در یک زمینه خاص در نظر گرفته شود

یا

سیستمهای خبره, برنامه های کامپیوتری دارای اثر متقابل هستند که عمل تلفیق قوانین, قضاوت , شهود و دیگر تخصص ها را برای تامین یک توصیه قابل درک و زیرکانه در امور مختلف , انجام می دهند

یا

یک سیستم خبره سیستمی است که دارای قوانین کارشناسانه است و از جستجوهای کورکورانه اجتناب می کند, با تغییر و دستکاری سیمبولها استدلال می کند, اصول اساسی و بنیادی در یک حوزه خاص را می فهمد و می یابد , متدهای ضعیف استدلال برای عقب نشینی در مواقعی که قوانین خبره جوابگو نیستند دارد .

با مسائل دشوار در زمینه های پیچیده سروکار دارد. می تواند تشریحی از یک مسئله به صورت عبارات غیر تخصصی گرفته و آنها را به بازنمایی داخلی مناسب برای پردازش با قوانین کارشناسی و تخصصی خود تبدیل کند. می تواند برای دانش درونی خود نیز استدلال کند, به خصوص برای بازسازی منطقی مسیرهای استنتاج برای توضیح و تفسیر و توجیه کردن خود

۲۲-انواع دانش را با ذکر منبع نام ببرید؟

۱- دانش عمومی شامل توصیفات,حقایق و تئوری های منتشر شده است که در کتابهای درسی , روزنامه ها و مقالات تحقیقی غیره است.

۲-دانش خصوصی: متخصصان معمولا" دارای دانش خصوصی هستند.دانش خصوصی به طور گسترده ای شامل قوانین thumb به نام کشف کنندگی هستند.

۲۳- عوامل موثر بریک راه حل کارامد را بنویسید؟

۱- دانش قابل اجرا , صحیح و قابل تفکیک و تمایز. ۲ - حذف سریع دیدگاه های غیر ثمر بخش

۳ - منابع دانش مضاعف مشترک ۴ - تقسیم راه حل هل به سطوح متفاوتی از تجرد

۲۴- مشکلات اساسی در رهیافت مبتنی بر دانش را ذکر کنید؟

دانش اشتباه یا خطا , وجود امکان های مختلف برای ارزیابی کردن , وجود رویه های پیچیده جهت حذف احتمالات , مسائلی که به صورت پویا تغییر می کنند

۲۵- یک حل کننده مسئله ایده آل باید دارای چه خصوصیاتی باشد؟

۱- دارای پردازنده زبان برای مکالمات مبتنی بر مسائل باشد

۲ - چرکنویس برای ثبت نتایج میانی داشته باشد

۳ !از دانش لازم در مورد یک حوزه شامل حقایق , کشف کنندگی و عقاید (حقایق اثبات نشده) برخوردار باشد

۴ ! از دانش لازم برای چک کردن ثبات یک راه حل یافت شده برخوردار باشد

۵- از دانش لازم برای برنامه ریزی استراتژی راه حل مسئله بعدی برخوردار باشد

۶ ! از دانش لازم برای ارزیابی راه حل های جزئی برخوردار باشد

۲۶- سه ایده اساسی در حل مسائل هوشمند را ذکر کنید؟

۱ - ساخت راه حلها ی کارامد با انتخابی از فضایی از پیشنهادات متفاوت

۲ - شناسایی راه حل های مفید و سپس کاوش بیشتر در آنها

۳ - هرس کردن راه حل جستجو تا رسیدن به بهترین راه حل

۲۷-چند تفاوت بین پردازش دانش و پردازش داده را بنویسید؟

پردازش دانش	پردازش داده
بازنمایی و استفاده از دانش	بازنمایی و استفاده از داده های استاتیک
کشف کنندگی	الگوریتم ها
پردازش های استنتاجی	پردازش تکراری
کنترل گسترده و مقدار کم داده با هم نگه داری می شوند	برنامه کنترل و مقدار زیاد داده جداگانه نگه داشته می شود

۲۸-چهار خصوصیت اولیه یک سیستم خبره را بنویسید؟

۱) متخصصی که باید کارایی تخصصی و کارشناسانه داشته باشد.آنها باید صاحب درجه بالایی از مهارت باشند و به اندازه کافی قدرتمند باشند.

۲) به دلیل اینکه دانش آنها سمبولیک است , استدلال سمبولیک را اجرا کنند.

۳) آنها باید قادر باشند قوانین پیچیده را استفاده کنند و دامنه های مشکل مسائل را اداره کنند.

۴) آنها باید قادر به تست و امتحان کردن قدرت استدلال خود باشند و بتوانند عملیات خود را توضیح دهند.

۲۹- مهندسی دانش را تعریف کنید؟

هنر جمع آوری و پردازش دانش , مهندسی دانش نامیده می شود

۳۰-مراحل اکتساب و نمایش و پیاده سازی دانش را نام ببرید؟

۱-مرحله شناسایی Identification:(تشخیص مسئله و خصوصیات آن) که خود شامل مراحل زیر می شود:

الف)شناسایی مسئله ب)شناسایی منابع ج)شناسایی اهداف

۲-مرحله ادراک Conceptualization (یافتن مفاهیمی برای ارایه دانش)

۳- مرحله رسمی سازی Formalization(طراحی ساختارهایی برای سازماندهی دانش)

۴- مرحله پیاده سازی و اجرا Implementation(فرموله کردن قوانین بطوریکه دانش را در بر گیرد)

۵- مرحله تست Testing(تست صحت قوانین ارایه شده)

دو مرحله اول اکتساب و سازماندهی دانش هستند و سه مرحله آخر نمایش و پیاده سازی دانش هستند.

۳۱ - شش ویژگی از تعریف جامع برکمن از هوش مصنوعی را بیان کنید

۱- دارای قوانین کارشناسانه است ۲ - از جستجوهای کورکورانه اجتناب می کند ۳ - با تغییر و دستکاری سیمبولها استدلال می کند

۴ - اصول اساسی و بنیادی در یک حوزه خاص را می فهمد و می یابد ۵ - متدهای ضعیف استدلال برای عقب نشینی در مواقعی که قوانین خبره جوابگو نیستند دارد

. ۶ - با مسائل دشوار در زمینه های پیچیده سروکار دارد. ۷ - می تواند شرحی از یک مسئله به صورت عبارات غیر تخصصی گرفته و آنها را به بازنمایی داخلی مناسب برای پردازش با قوانین کارشناسی و تخصصی خود تبدیل کند. ۸ - می تواند برای دانش درونی خود نیز استدلال کند, به خصوص برای بازسازی منطقی مسیرهای استنتاج برای توضیح و تفسیر و توجیه کردن خود.

۳۱-انواع رویه های استنتاج را نام ببرید؟

۱- رویه استنتاج در حساب گزاره ای ۲- رویه استنتاج در حساب مسندی ۳- رویه استنتاج در سیستمهای تولید مبتنی بر قانون

۳۲- دو روش مهم را نام ببرید که قوانین در آنها می توانند برای تطابق و اجرا از پایگاه دانش انتخاب شوند؟

۱-زنجیره سازی رو به جلو و ۲: زنجیره سازی رو به عقب

۳۳- دو مفهومی را که باید در متدولوژی برنامه نویسی سیستمهای خبره از هم تفکیک کرد نام ببرید؟

۱- دانش دامنه یا حوزه ۲- متدهای حل مسئله

۳۴-چهار نوع ابزار توسعه سیستمهای خبره را نام ببرید؟

۱- زبان های الگوریتمیک (مانند C,Pascal, Bassic) ۲- زبان های سیمبولیک (مانند لیسپ و پرولوگ)

۳ محیط های توسعه (مانند Art,KEE,LOOPS) ۴ بدنه ساختمان سیستم های خبره (مانندCrystal,XpertRule,Leonardo,Xi-Plus)

۳۵-دو دسته بندی از زبانها را در هوش مصنوعی بنویسید؟

۱- زبانهای قراردادی : به صورت رویه ای در طبیعت توصیف می شوند که برای کار بر اساس الگوریتم , و اجرای برنامه های هوش مصنوعی بکار می روند.

۲- زبانهای AI : که برای ساخت بدنه هوش مصنوعی بکار می روند مانند لیسپ و پرولوگ

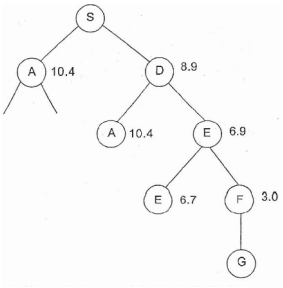
۳۶ - روش مینی مکس را تشریح نمایید.

فرض کنید یک آنالیز کننده , قضاوتها در مورد موقعیتهای صفحه بازی دو نفره را به یک عدد کیفیتی نسبت می دهد. هم چنین فرض کنید که اعداد مثبت نشان دهنده موفقیت یک فرد و اعداد منفی برای موفقیت حریف او اوست .

پروسه تعیین عدد کیفی را " ارزیابی آماری" می نامند .در پایان حرکا ت میتوان امتیازات ارزیابی آماری را محاسبه کرد. امید بازیکن برای اعداد مثبت بیشتر(Max) و برای حریف او کمتر(min) است . در سطح میانی در درخت جستجو ،ارزش نهایی نشانداده شده است .ارزش یک وضعیت غیر پایانی بوسیله برگشت دوباره از مراحل پایانی محاسبه شده است . این روش که بوسیله اطلاعات امتیاز دهندگی شما را از درخت جستجوی بازی می گذراند رویه مینی مکس نامیده می شود.

۳۷ - رویه جستجوی تپه نوردی را با یک شکل توضیح دهید

جستجوی تپه نوردی ، یک جستجوی عمقی با یک اندازه گیری اکتشافی است که گره هایی را که گسترش می دهیم مرتب می کند. شماره کنار گره ها ، مسافت خط مستقیمی است که گره با گره هدف دارد.



رویه جستجوی تپه نوردی:

۱- اگر هدف پیدا شد موفقیت را نشان بده در غیر این صورت شکست را نشان بده.

۲- از یکی از عناصر صف شامل گره ریشه شروع کن

۳- تا زمانی که صف هست خالی یا هدف بدست می آید تعیین کن که آیا اولین عنصردر صف گره هدف است :

a) اگر اولین عنصر گره هدف است کاری انجام نده

b) اگر اولین عنصر؛ گره هدف نیست ، اولین گره را حذف کن. فرزندان آن گره را اگر وجود دارد مرتب کن ، مسافت باقی مانده را تخمین بزن.و آنها را در ابتدای صف اضافه کن .

۳۸ - امکانات نمونه ای پیاده سازی تهیه شده توسط شل ها را نام ببرید؟

۱ - یک زبان بازنمایی دانش ۲ - یک ویراستار پایگاه دانش ۳ - امکانات ردیابی و اشکالزدایی ۴ - تعدادی امکانات واسط کاربری

۵ - به زبان های قراردادی یا معمولی / خارجی می پیوندند. ۶ - امکاناتی برای استدلال های غیر حتمی ۷ - امکانات قیاس کل از جزء (شاید)

۳۹ - دید و بینایی کامپیوتری را با ذکر چند کاربر تعریف کنید

دید یا vision پردازشی است که به وسیله ان شرح مناظر فیزیکی از تصویر انها استنتاج می شود.کاربردهای متنوعی از چشم کامپیوتری وجود دارند همانند: انالیز تصاویر پزشکی , مونتاژ کردن (assembly) , کشتیرانی و ناوبری , واسط انسان و کامپیوتر و غیره.

۴۰ - چهار مرحله تقلید کامپیوتر از دید انسان را نام ببرید

۴ مرحله تقلید می کند که به ترتیب عبارتند از ۱)اکتساب تصویر ۲) پردازش تصویر ۳) تجزیه و تحلیل تصویر ۴) فهم تصویر

۴۱ - چند کاربرد از هوش مصنوعی را نام ببرید؟

۱- الیزا : نوعی تقلید از یک جلسه روانپزشکی ۲- پارانویید مصنوعی Parry : پری نقش بیمار روانی را بازی می کند

۳- دندرال Dendral : یافتن ساختمان مولکولی یک جسم مرکب ۴- مایسین: تشخیص و درمان بیماری عفونی

۵- R1(XCON) : مدیریت تهیه کامپیوتر بر اساس قطعات درخواستی و سفارشی مشتری ۶- PROSPECTOR : معدن یاب

۴۲ - دو تکنیک برای تجزیه و تحلیل زبان طبیعی یا NL را نام برده و شرح دهید

۱) تطبیق با قالب (همچنین key board analysis).در این روش سیستم جمله وروردی را برای کلمات کلیدی خاصی پویش می کندو وقتی پیدا شدند , سیستم با یک پاسخ موجود واکنش نشان می دهد.

۲) تجزیه نحوی (Syntactic driven Parsing) :در این روش از دانش قواعد یک زبان برای تجزیه و تحلیل استفاده می شود.

۴۳ ! WebBot چیست ؟

یک ماشین عامل است که در خواستهای عامل های دیگر را که مرتبط با اینترنت است را انجام می دهد . این عامل مهارت لازم برای فراهم کردن اطلاعات مورد نیاز از اینترنت را از طریق جستجو در اینترنت داراست و وظیفه تعامل و دریافت راه حلهای جئی ارایه شده از عاملهای وبی درخواست شده از عامل برنامه ریز و انتقال آن را به عامل برنامه ریز را به عهده دارد .