Automatic Speech Recognition
تشخیص اتوماتیک گفتار

مهدی صفاری – فارغ التحصیل مهندسی نرم افزار کامپیوتر

دانشگاه آزاد لاهیجان – اردیبهشت ۱۳۸۶

Email: msaffari2005@gmail.com
WebSite: Www.PersianArticles.com
Weblog: Www.e-commerc.blogfa.com

فصل اول

مقدمه

فهم زبان محاوره ای کار سختی است و این قابل ملاحظه است که انسان این کار را بخوبی انجام می دهد. مسئله تشخیص گفتار اتوماتیک یا ASR(Automatic Speech Recognition) ساخت سیستم هایی است که یک سیگنال صوتی را جزء به جزء به یک رشته لغات تبدیل کند.

مسئله اصلی، رونویسی از گفتار بصورت اتوماتیک است که بوسیله هر سخنگویی و در هر محیطی انجام شود که هنوز فاصله زیادی تا حل شدن دارد. اما در سالهای اخیر تکنولوژی ASR رشد کرده تا نقطه ای که در حوزه های محدود و خاص استفاده می شود. یکی از کاربردهای اصلی آن بعنوان واسط بین انسان و رایانه است. از آنجایی که خیلی از کارها بوسیله ارتباطات بصری بهتر حل می شود، گفتار هم دارای این توانایی است که یک نوع رابط بهتر از صفحه کلید باشد. مخصوصا برای کارهایی که بصورت ارتباط زبان طبیعی هستند و یا بجای آندسته از صفحه کلیدهایی که مناسب یک کار خاص نیستند بسیار مفید است. کاربرد ASR شامل آندسته از اعمال که در آنها چشم و دست مشغول هستند نیز می شود مانند مکانهایی که کاربر مجبور است با دست عملی را انجام دهد یا تجهیزاتی را کنترل نماید.(مانند کنترل پرواز برای خلبان یک هواپیما یا هنگام عمل جراهی توسط یک پزشک متخصص). ناحیه کاربردی دیگر که هم اکنون از ASR استفاده می شود در تلفن است. بعنوان مثال برای وارد کردن اعداد، تشخیص کلمه "الو" برای پذیرش تماس و ... . سرانجام

ASR در دیکته عملی شد که عبارتست از رونویسی سخنرانی طولانی یک شخص ویژه. این کاربرد هم در مکانهایی مثل دادگاهها و مکانهایی که نیاز به ثبت گفتگوهای طولانی دارند عملی شد.

تکنولوژی ASR یکی از پیچیده ترین تکنولوژیهاست که با وجود تلاش بسیاری از دانشمندان در سالهای گذشته هنوز کم و کاستی هایی دارد. در کل هر فرآیندی که بوسیله مخلوقات خداوند انجام می شود بسیار پیچیده است ولی در اولین نگاه سهل و آسان پنداشته می شود. هوش مصنوعی که یکی از گرایشات رشته مهندسی کامپیوتر است بر روی اعمال مهم و خارق العاده و در عین حال ساده موجودات زنده متمرکز شده است و با الهام گرفتن از آنها سعی در حل مسائل بشر می کند. با توجه به کاربردهای سیستم های تشخیص گفتار که در بالا ذکر شد حتما متوجه اهمیت موضوع شده اید ولی ایجاد همچنین سیستمی احتیاج به سطح علمی و فکری بسیار بالایی دارد. تشخیص گفتار در مقطع فوق لیسانس هوش مصنوعی و مخابرات تدریس می شود ولی برای اینکه فارغ التحصیل این رشته ها بتوانند سیستم ASR قابل قبولی ایجاد کنند باید خیلی تلاش کنند. اینها صحبتهای فارغ التحصیلان دانشگاه صنعتی شریف است که با آنها مکاتبه داشته ام. بنابراین قابل توجه دوستانی که به این مبحث علاقه مند شده اند که به این زودیها نباید به ساخت یک سیستم ASR مناسب امید داشته باشند.

مباحث مهم در ASR عبارتند از:

پردازش سیگنال دیجیتال(Digital Signal Processing)

پردازش صوت

آشنایی با قواعد زبانی و شکل شناسی

شبکه های عصبی

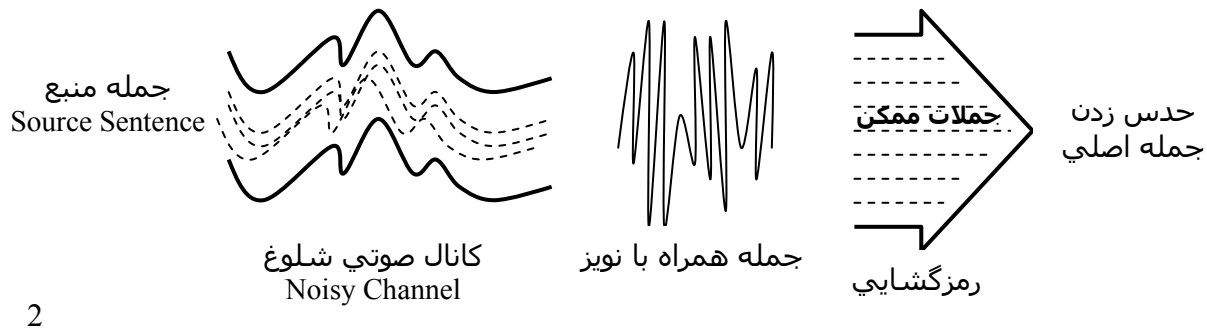
قواعد احتمالاتی(قاعده بیز، گوس)

پردازش زبان طبیعی و ....

معماری تشخیص گفتار

مدل Noisy Channel

سیستم های تشخیص صوت با وروردی صوتی شامل جمله همراه با نویز برخورد می کنند بنابراین برای کدگشایی این جملات ما همه جملات ممکن را در نظر می گیریم و برای هر کدام احتمال اینکه بتواند جمله اصلی را تولید کند محاسبه می کنیم، سپس جمله با بیشترین احتمال را انتخاب می کنیم.



مدل Noisy Channel برای تمامی جملات عملی است. تشخیص دهنده گفتار مدرن بوسیله جستجو از میان فضای بسیار بزرگی از جملات منبع کار می کند و یکی را که بیترین احتمال تولید جمله نویزدار را دارد انتخاب می کند. برای انجام این کار آنها باید قالبهایی که احتمال جملات را بیان می کنند را بعنوان رشته ای از لغات مجسم سازند(N-grams) و قالبهایی که احتمال مجسم کردن لغات را بعنوان یک رشته مشخص از صداها بیان می کنند(HMMs) و قالبهایی که احتمال مجسم کردن صداها را بعنوان صدا یا کیفیت طیفی بیان می کنند.(گاوس/MLPS).

عملی کردن مدل Noisy Channel که در شکل قبل آنرا بیان کردیم نیاز به حل دو مسئله دارد. اول، انتخاب جمله ای که بیشترین شباهت را با ورودی همراه با نویزی داشته باشد که ما نیاز خواهیم داشت. بدلیل اینکه گفتار بسیار بی ثبات و تغییر پذیر است، ممکن است یک جمله ورودی صوتی واقعا با هیچ مدلی که ما برای آن داریم مطابقت نداشته باشد. همانگونه که در بخش قبل پیشنهاد کردیم ما از احتمال بعنوان میزان اندازه گیری استفاده خواهیم کرد و نشان خواهیم داد که چگونه با ترکیب احتمات مختلف برآورد کاملی برای احتمال یک توالی از جملات کاندید می توان بدست آورد.

دوم ، از آنجایی که مجموعه کل جملات انگلیسی(یا فارسی) بسیار بزرگ است، ما به یک الگوریتم کارا احتیاج داریم که نمی خواهد از میان همه جملات ممکن جستجو انجام دهد. اما فقط یکی که شناس بیشتری برای شباهت با ورودی دارد را بدهد. نام این مسئله رمز گشایی یا جستجو است. دو رهیافت در این مورد عبارتند از رمزگشایی ویتربی(Viterbi) یا برنامه نویسی پویا(Dynamic Programming) و رمز گایی A* یا پشته.

ابتدا می خواهیم مدل بیز در احتمال را برای تشخیص گفتار معرفی کنیم.

هدف معماری Noisy Channel وابسته به احتمال برای تشخیص گفتار می تواند اینگونه خلاصه شود:

شبیه ترین جمله از میان همه جملات زبان L که از ورودی صوتی O گرفته شده چه چیزی است؟

ما می توانیم با ورودی صوتی O بعنوان توالی منحصر بفردی از "علامتها" یا "مشاهدات" رفتار کنیم.(بعنوان مثال بوسیله قطعه قطعه کردن ورودی در هر ده میلی ثانیه و نمایش(بیان) هر قطعه بوسیله مقداری از انرژی یا فرکانس آن قطعه). بعضی اوقات هر شاخص فواصل را نشان می دهد و Oi متوالی موقتا بر تکههایی متوالی از ورودی اشاره دارد :

$$O=o1,o2,o3,\ldots,on$$

بطور مشابه ما با یک جمله ای که بطور ساده از رشته ا از لغات تشکیل شده، رفتار خواهیم کرد.

$$W=w1,w2,w3,\ldots,wn$$

هر دوی اینها ساده می شوند. بعنوان مثال تقسیم جمله به لغات بعضی اوقات یک تقسیم خوب است و گاهی اوقات یک تقسیم نامناسب.

$$\hat{W}=\underset{W\ \varepsilon\ L}{\operatorname{argmax}}\ P(W/O)$$

با توجه به موارد بالا می توانیم نشان دهیم که :

فراخوانی تابع $\operatorname{argmax} f(x)$ به این معنی است : "به ازای هر x که f(x) بزرگترین است"

تساوی بالا به ما جمله W را میدهد. ما هم اکنون احتیاج داریم تا یک معادله عمل شدنی بسازیم. که آن هست : برای جمله W داده شده و توالی صوتی O ما احتیاج به کامل کردن P(W/O) – که احتمال w به شرط رخ دادن O است – داریم. برای فراخوانی هر احتمال داده شده بصورت P(x/y) می توانیم از قاعده بیز استفاده کنیم تا آن را به جمله زیر ساده کنیم :

$$P(x/y) = \frac{P(y/x) \cdot P(x)}{P(y)}$$

$$\hat{W} = \underset{W \varepsilon L}{\operatorname{argmax}} \frac{P(O/W) \cdot P(W)}{P(O)}$$

می توانیم تساوی قبلی را بصورت

احتمالات سمت راست تساوی بالا برای ساده تر کردن محاسبه $P(W/O)$ است. $P(O/W)$ قابل محاسبه است. اما $P(O)$ که احتمال توالی مشاهده صدایی است که ثابت می شود بسختی قابل محاسبه است. خوشبختانه می توانیم از مقدار $P(O)$ صرفنظر کنیم،چرا؟ از آنجایی که ما برای همه جملات ممکن ماکزیمم احتمال را لازم داریم و می خواهیم مقدار $\frac{P(O/W) \cdot P(W)}{P(O)}$ را برای هر

جمله در زبان محاسبه کنیم. اما $P(O)$ برای هر جمله تغییر نمی کند. برای هر جمله بالقوه ما مشاهدات مشابه O را امتحان می کنیم که باید احتمال مشابه $P(O)$ را داشته باشد. بنابراین :

$$\hat{W} = \underset{W \varepsilon L}{\operatorname{argmax}} \frac{P(O/W) \cdot P(W)}{P(O)} = \underset{W \varepsilon L}{\operatorname{argmax}} P(O/W) \cdot P(W)$$

بطور خلاصه بیشترین احتمال جمله W به ازای توالی مشاهده شده O می تواند بوسیله گرفتن حاصلضرب دو احتمال برای هر جمله محاسبه شود و جمله ای که بیشترین حاصلضرب را داشته باشد انتخاب می کند. این دو اصطلاح بیه این شکل نامگذاری شده اند: $P(W)$ احتمال پیشین که مدل زبانی نامیده می شود و $P(O/W)$ که احتمال مشاهده است، مدل صدایی نامیده می شود.

مقدار $P(W)$ را می توان بوسیله مدلهای زبانی N-gram بدست آورد. اکنون نشان می دهیم که چگونه $P(O/W)$ مدل صدایی را را محاسبه کنیم. در دو گام، اول فرض ساده ای می کنیم که توالی ورودی از صداهای ساده در مقابل مشاهدات صوتی است. احتمال این مشاهدات صوتی یک لغت را تولید می کند. نشان خواهم داد که HMMs چگونه برای بدست آوردن احتمال یک توالی از جمله داده شده صوتی توسعه داد شده می شوند.

شکل بعد طرح مختصری از اجای سیستم تشخیص گفتار را نشان می دهد. شکل، یک سیستم تشخیص گفتار را نشان می دهد که به سه مرحله تقسیم شده است. در مرحله پردازش سیگنال یا استخراج کیفیت (Feature Extraction) شکل موج صوتی به فریم هایی تقسیم شده است(معمولا در هر ۱۰، ۱۵ یا ۲۰ میلی ثانیه) که به استخراج طیفی (Spectral Feature) تبدیل شده که اطلاعاتی درباره میزان انرژی موجود در فرکانس های مختلف است. در مرحله تشخیص صدا یا تشخیص کلمات کوچک، ما از تکنیک های احتمالاتی مانند "شبکه های عصبی" و "مدلهای گاوس" بمنظور تشخیص صداهایی مثل P یا b از یکدیگر استفاده خواهیم کرد. برای یک شبکه عصبی، خروجی این مرحله یک بردار(Vector)

از احتمالات صداهای هر فریم است.(در این طرح احتمال P برابر ۰،۸ و احتمال برای b ۰،۱ و F ۰،۰۲ و ... است). برای مدلهای گاوس، احتمالات کمی متفاوت است. سرانجام در مرحله رمزگشایی ما یک فرهنگ لغت از مدلهای زبانی و تلفظات داریم و از روشهای رمزگشایی A* یا Viterbi برای پیدا کردن توالی لغاتی که بیشترین احتمال ورود صوتی را دارد پیدا می کنیم.

آواشناسی

مطالعه و بررسی تلفظ لغات قسمتی از رشته آواشناسی(Phonetics) است که همانا مطالعه روی صداهای گفتار در همه زبانهای جهان است. ما تلفظ یک لغت را که رشته ای سمبل هاست ، مدل خواهیم کرد. ما صداهای ساده را بوسیله علامتهای نشان خوایم داد که بعضی شباهتها با یک حرف در الفبای زبانی مثل انگلیسی دارند. بنابراین بعنوان مثال صدایی بوسیله L نمایش داده می شود که معمولا با حرف L مطابقت دارد و صدایی بوسیله P نشان داده می شود که معمولا با حرف P مطابقت دارد. در حقیقت همانطور که خواهیم دید صداها با حروف خیلی تفاوت دارند. در این بخش فقط بطور مختصر دستی بر جنبه هایی از علم ترکیب صداها(آواشناسی) مانند وزن شناسی(Prosody) خواهیم داشت که شامل مواردی مثل تغییرات در زیر و بمی صدا(Pitch) و مدت زمان(Duration) می شود.

این بخش شامل برآوردی از صداهای مختلف انگلیسی مخصوصا انگلیسی-آمریکایی است که نشان می دهد صداها چگونه تولید می شوند وچگونه با علامت نشان داده می شوند. ما از دو الفبای متفاوت برای تشریح صداها استفاده خواهیم کرد. اولین آن الفبای صدایی بین المللی (International Phonetic Alphabet –IPA) است. IPA استانداردی است که بوسیله انجمن آواشناسی بین المللی در سال ۱۸۸۸ گسترش یافت، با هدف اینکه صداهای همه زبانهای انسانها آوانویسی شوند.IPA فقط الفبا نیست بله مجموعه ای از قواعد کلی برای نسخه برداری است. بنابراین سخن یکسان می تواند به روشهای مختلفی مطابق با قواعد کلی IPA آوانویسی شوند. بدلیل اختصار می خواهیم روی علامتهایی که به انگلیسی مربوط می شوند متمرکز شویم. شکل ۱-۱ زیر مجموعه ای از علامتهای IPA را برای حروف بی صدا (Consonant) نشان می دهد در حالی که شکل ۲-۱ زیر مجموعه ای از علامتهای IPA برای مصوتهاست (Vowels) این جداول همچنین علامتهای ARPAbet را می دهند. ARPAbet الفبای فونتیک دیگری است. اما برای انگلیسی-آمریکایی طراحی شده و از علامتهای ASCII استفاده می کند. می توان اینطور تصور کرد که ARPAbet بعنوان یک نمایش اسکی مناسب برای انگلیسی-آمریکایی، زیر مجموعه ای از IPA است. علامتهای ARPAbet اغلب در کاربردهایی استفاده می وشند که فونتهای غیر اسکی نامناسب باشند، مانند فرهنگ لغتهای تلفظی آنلاین.

بسیاری از علامتهای IPA و ARPAbet معادل حروف رومی هستند که در انگلیسی و بسیاری از زبانهای دیگر استفاده می وند. بعنوان مثال علامت [p] در IPA و ARPAbet نشاندهنده حرف بی صدا در آغاز کلمه Platypus (تلفظ آن : لاتیپوس) و Pachyderm (تلفظ آن : آکی درم) است. ارتباط بین املا درست حروف انگلیسی و علامتهای IPA به ندرت به این سادگی است. به ره حال به این دلیل ارتباط بین املا درست انگلیسی و تلفظات مبهم است که یک لغت می

تواند در متون مختلف صدهای مختلفی داشته باشد. شکل ۳-۱ حرف انگلیسی c را نشان می دهد که برای کلمه cougar در IPA بوسیله [k] نشان داده می شود، اما برای کلمه civet در IPA بوسیله [s] نشان داده می شود. بعلاوه صداهایی که بعنوان C و K ظاهر می شوند، صداهایی که بعنوان [k] در IPA علامتگذاری شده می تواند بعنوان قسمتی از x در fox، بعنوان ck در jackel و بعنوان cc در raccoon باشد. بشیاری از زبانهای دیگر مانند اسپانیایی شفافیت بیشتری در ارتباط بین املا درست و صداها نسبت به انگلیسی دارند.

اندامهای صوتی

اکنون درباره اینکه چطور صداها تولید می وند توضیح خواهیم داد. بعنوان مثال اندامهای مختلف درون دهان،گلو و بینی، جریانهای هوای از شش ها را تغییر می دهند.

صدا بوسیله حرکات سریع مولکولهای هوا تولید می شود. اغلب صداها در زبانهای گفتاری انسانی بوسیله خروج هوا از داخل ششها و گذر آن از میان نای، حنجره تولید می شوند. حنجره شامل دو چین کوچک ماهیچه ای است که می توانند همراه با یکدیگر یا جداگانه حرکت کنند. فضای بین این دو چین دهانه حنجره(Glottis) نام دارد. اگر این دو قسمت با هم بسته شوند(اما نه کاملا بسته) هنگام عبور هوا ببه ارتعاش در می آیند.اگر بین آنها فاصله زیاد باشد نوسان نخاهند کرد. در زبان انگلیسی صداهایی که بوسیله این لرزش بوجود می آیند شامل [z],[v],[g],[d],[b] و همه مصوت های دیگر، Voiced نامیده می شوند و صداهایی که بدون لرزش این اندامها ایجاد می شوند صداهای گنگ نام دارند مثل [z],[f],[k],[t],[p] و ... .

ناحیه بالای نای قسمت صوتی نامیده می شود که شامل بخش زبانی و بخش تودماغی(Nasal) می شود. بعد از اینکه صدا از نای خارج شد می تواند از دهان یا بینی خارج شود. اغلب صداها بوسیله گذر از میان دهان ایجاد می شوند. صداهایی که بوسیله عبور هوا از طریق بینی ایجاد می شوند صداها تودماغی نام دارند. این صداها از قسمتهای زبانی و دماغی بعنوان اسباب ارتعاش استفاده می کنند. صداهای تودماغی انگلیسی مثل [ng],[n],[m].

صداها به دو کلاس اصلی تقسیم می شوند. حروف بی صدا و مصوت ها. هر دو نوع بوسیله حرکت صدا از میان دهان، نای یا بینی شکل می گیرند. حروف بی صدا مانند [z],[s],[v],[f],[q], [k],[d],[t],[b],[p] و حروف صدادار مانند [aa],[ae],[aw],[ao],[ih],[ow], ...

حروف نیمه صدادار(SemiVowel) مانند [w] , [y] مشخصاتی از هر دو دارند. آنها شبیه حروف صدادار هستند اما کوتاه و دارای هجاهای شمرده کمتر هستند(مانند حروف بی صدا).

فصل دوم

صدا در کامپیوترهای شخصی

۱. مقدمه :

مدت زیادی از ورود صدا به دنیای کامپیوتر می گذرد و در این مدت تکنولوژی های مربوط به پردازش صوت پیشرفتهای چشمگیری داشته اند. از زمان ورود صدای واقعی به دنیای کامپیوتر تکنولوژی های پردازش صوت تغییرات عمده ای را پشت سر گذاشته اند، این تغییرات شامل قوی تر شدن پردازنده های صوتی، بالا رفتن کیفیت ضبط و نمونه برداری صدا و ... می باشند.

صدا در کامپیوتر در قالبهای مختلفی ایجاد و نگهداری می شود. به عنوان مثال در فرمت Mid اطلاعات صوتی بصورت نت(Note) ذخیره می شوند، یعنی اطلاعات مربوط به هر ساز، به همراه پرده ها، نت ها و سایر اطلاعات، جداگانه ذخیره می شوند ولی در فرمت Wav اطلاعات صوتی بصورت طول موج های صدا ذخیره می شود و صدا ها قابل تفکیک نیستند. تعدادی از مهمترین فرمتهای صوتی به این شرح است : Mid, Wav, MP3, Wma, Ra .

۲. اصول کارکرد دیجیتالی کردن صدا :

زمانی که کاربری در یک میکروفن صحبت می کند امواج صوتی توسط میکروفن به امواج الکتریکی تبدیل می شوند. اما هنوز این امواج برای کامپیوترها قابل فهم نیستند. برای این کار سیگنالهای الکتریکی ساخته شده باید به روشی تبدیل به داده های دیجیتالی شوند. اینکار شامل سه مرحله است :

- کوانتیزه کردن یا تقریب زدن(Quantization) : برای آنکه هر سیگنال آنالوگی را بتوان به سیگنال دیجیتال تبدیل کرد قبل از هر کاری باید تعداد سطوح ممکن را مشخص کرد. همانطور که می دانیم در یک سیگنال دیجیتال سطوح ممکن ۲ سطح است (صفر یا یک مثلا معادل با برق صفر ولت و ۲۰ ولت). اما در سیگنال آنالوگ یک سیگنال می تواند بین دو سطح مثلا صفر تا ۲۰ ولت هر مقدار ممکنی داشته باشد، مثلا ۱۵/۷ولت. برای اینکه بتوان چنین سیگنالی را به سیگنال دیجیتال تبدیل کرد درگام اول باید فرض کرد سیگنال مورد نظر بین دو مقدار حداقل و حداکثر خود چه تعداد سطح دارد. به عنوان مثال برای کاربردهای تلفنی فرض می شود که یک سیگنال آنالوگ می تواند معادل یکی از ۲۵۶ سطح بینابینی باشد و در غیر اینصورت سطح سیگنال در یک لحظه خاص به نزدیکترین مقدار سطح ولتاژ گرد می شود.

- نمونه برداری(Sampling) : در مرحله بعد تعیین می شود که سیگنال آنالوگ باید در چه فاصله زمانی نمونه برداری شود. هر چه تعداد نقاط نمونه برداری بیشتر باشد دقت عملیات تبدیل و کیفیت سیگنال دیجیتال بدست

آمده بالاتر خواهد بود. به عنوان مثال در سیستم های آنالوگ در هر ثانیه ۴۰۰۰ بار از سیگنال آنالوگ نمونه برداری می شود(تا بعدا تعیین شود که در هر یک از این لحظه هاسطح سیگنال معادل کدام یک از آن ۲۵۶ سطح ممکن است). در هر ثانیه چهار هزار بار نمونه برداری یعنی سرعت نمونه برداری معادل 4KHz است. این عدد در استانداردهای صوتی امروزی بسیار پائین است. بد نیست بدانید که سرعت نمونه برداری برای یک سیگنال صوتی مربوط به موسیقی در حد کیفیت FM معادل 22KHz و برای سیستمهای صوتی با کیفیت CD معادل با 44KHz است.

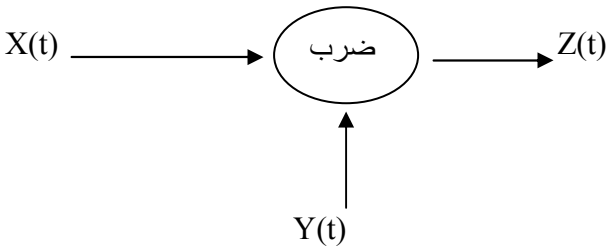
- کدگذاری(Coding) : در آخرین مرحله باید اندازه یا سطح سیگنال را در هر یک از لحظات نمونه برداری کرده و آنرا به زبان دیجیتال تبدیل کنیم.

به منظور فهم بیشتر مراحل دیجیتالی کردن صدا، به مثال زیر توجه کنید :

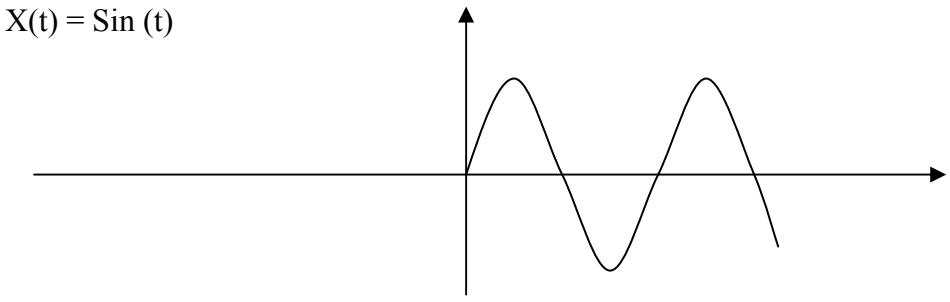
با توجه به اینکه فرکانس موج مکالمه انسان به طور کلی از صفر تا چهار کیلوهرتز در نظر گرفته می شود، مراحل به این گونه انجام می شوند :

۱- کوانتیزه کردن : اگر تنها ۸ سطح اندازه گیری در نظ بگیریم و مقدار آستانه را مقدار میانی هر سطح ولتاژ داشته باشیم.

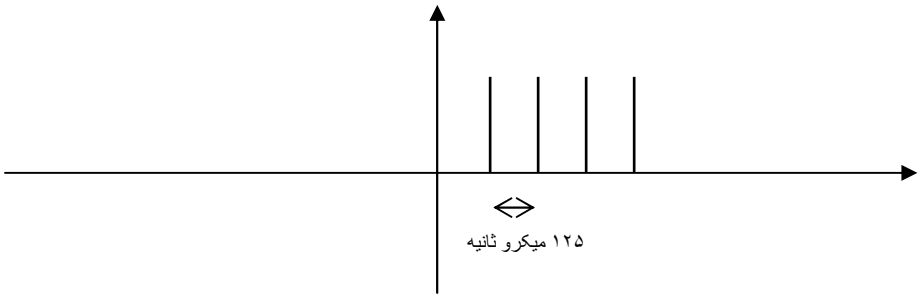
۲- نمونه برداری : عمل نمونه برداری بوسیله ضرب موج اصلی در یک تابع مخصوص انجام می شود:



شکل موج X(t) :



شکل موج تابع نمونه بردار Y(t) :



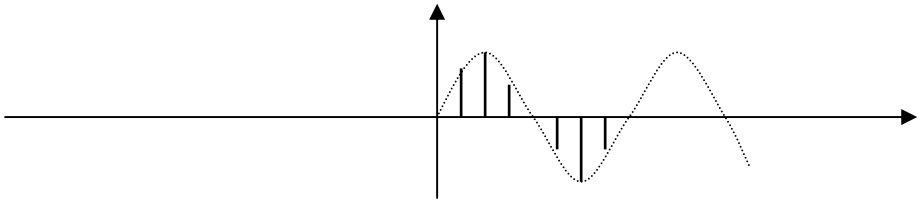
فاصله نمونه برداری اینطور تعیین می شود :

ماکسیمم فرکانس موج ورودی(موج مکالمه) = ۴ KHz = F

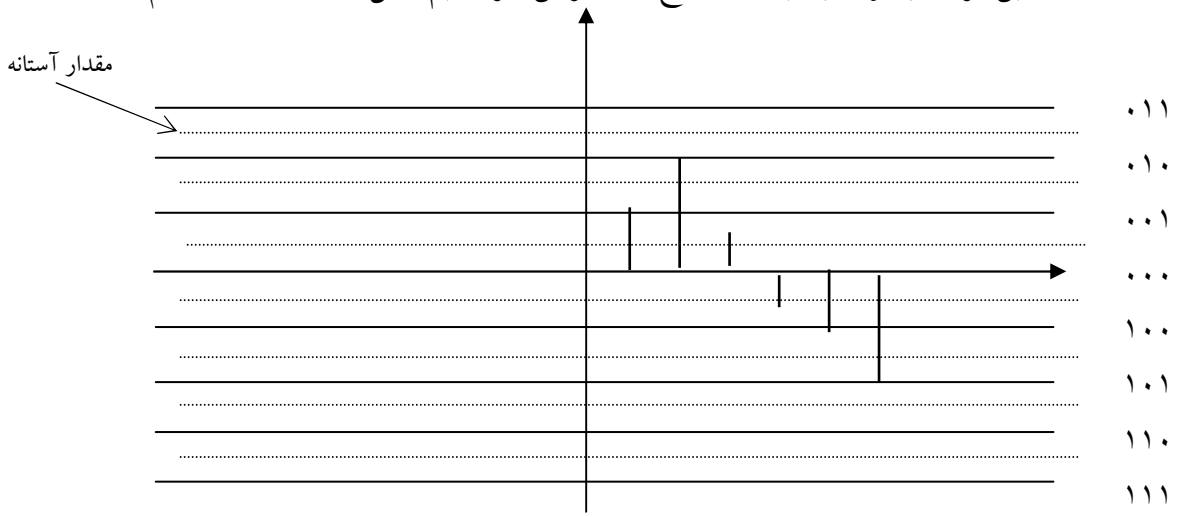
۲ × F = ۸ KHz

دوره تناوب = T = ۱۲۵ میکرو ثانیه

حاصلضرب دو تابع :



۳- کدگذاری : در این مرحله با توجه به اینکه ۸ سطح ولتاژ فرض کرده ایم عمل کد گذاری را انجام می دهیم.



اگر مقدار آستانه در نظر گرفته نشود برای بعضی از نمونه ها نمی توان کدی بدست آورد چون روی سطح مشخصی قرار ندارند. بنابراین مقدار آستانه را بین دو سطح در نظر گرفته، اگر مقدار نمونه روی سطح یا بزرگتر بود به مقدار بالایی و اگر کوچکتر بود به مقدار پائینی گرد می کنیم و در نهایت برای هر نمونه کد مربوط به آنرا ذخیره می کنیم.

به این صورت است که صدا در کامپیوتر بصورت صفر و یک ذخیره می شود.

فصل سوم

نرم افزارهای تشخیص گفتار

۱. مقدمه :

حتما تابحال با نرم افزار Microsoft Word کار کرده اید و احتمالا برای تایپ مستنداتی که دارای حجم زیادی از لغات است با مشکل مواجه شده اید. اما آیا می دانستید که به وسیله قابلیت تشخیص گفتار این نرم افزار و داشتن یک میکروفن می توانید متن خود را از روی کاغذ برای کامپیوتر خوانده و متن را در نرم افزار Word قرار دهید؟

در این بخش قصد پرداختن به نحوه کارکرد و عملکرد این قابلیت نرم افزار Microsoft Word را نداریم بلکه می خواهیم شما را بیشتر با نرم افزارهای تشخیص گفتار آشنا سازیم.

۲. ترجمه اصوات به کلمات :

هسته اصلی نرم افزارهای تشخیص گفتار بخشی از برنامه است که قابلیت ترکیب و گروهبندی مجموعه ای از triphoneهای آشکار شده از فرایند شنیدن را داراست.(triphone چیست؟؟؟؟؟؟؟؟؟؟)

به عنوان مثال دو عبارت زیر را در نظر بگیرید:

Atax accountant, Attacks accountant

این دو عبارت دارای تعداد زیادی از اصوات هم صدا هستند که از توالی یکسانی برخوردارند. وظیفه موتورهای تشخیص گفتار نرم افزاری تفکیک صحیح چنین عبارات و واژگانی است. درچنین مواردی روشهای متفاوتی برای تجزیه و تحلیل صحیح گفتار به کار می رود. این روش ها عبارتند از:

- گرامر: یکی از روش های مورد استفاده آن است که نرم افزار به قوانین حاکم بر زبان و همچنین نقش واژگان در ساختار جمله آگاه باشد
- مدل های استاتیکی زبان: در این روش از بانک های اطلاعاتی ویژه ای که دارای آمار و احتمال بکارگیری کلمات در موقعیت های مختلف در جمله هستند استفاده می شود. برای مثال از بین دو عبارت " Black and White" و "Blackend White" عبارت اول از سوی نرم افزار به ازای یک ورودی صوتی انتخاب می شود، زیرا در بانک اطلاعاتی نرم افزار ثبت شده است و احتمال آنکه واژه میانی بین دو اسم یا صفت کلمه "and" باشد بیشتر از حالات دیگر است.

۳. رقابت تکنولوژیک :

تحقیقات و رقابت در این زمینه بین شرکتهای رقیب بسیار فشرده و میلیمتری است. محققان فعال در این زمینه با دشواری کارآیی محصولات خود را افزایش می دهند. برخی از اصول شناخته شده کلی که برای بهبود کارآیی این گونه نرم افزارها بکار برده می شوند به قرار زیر است :

- تصحیح نویز: یکی از بزرگترین موانع سیستم های تشخیص گفتار جداسازی و تفکیک کلام انسان از نویز محیط است. یعنی ممکن است محصول در محیط آزمایشگاهی که نویز کمتری دارد خوب عمل کند ولی در محیط عملیاتی کارآیی آن پائین بیاید. یکی از ترفندهایی که برای کاهش اثرات نامطلوب نویز محیط بکار گرفته شده است استفاده از بانک اطلاعاتی از اشکال مختلف نویز است. بدین معنی که با نویز هم به همان ترتیبی که برای تشخیص کلام بکار گرفته شده است، برخورد شود. در این روش به کمک بانک های اطلاعاتی، سیگنال های نویز شناسایی شده و سپس حذف می شوند. این روش در زمانی بیشترین تاثیر را دارد که کاربر بتواند شکل نویز محیط را به نرم افزار معرفی کند. اگر شما با نرم افزارهای حرفه ای ضبط صدا کار کرده باشید با این تکنیک آشنا هستید. برای این کار نرم افزار به شما امکان می دهد تا چند ثانیه "سکوت" ضبط کنید تا شکل نویز محیط برای نرم افزار معلوم شود. در مرحله بعدی شکل موج نویز محیط ضبط شده از شکل موج مکالمه حذف می شود. برای مثالی دیگر که شاید عامیانه تر باشد : در تلفن های همراه قابلیتی وجود دارد که شما می توانید محیطی که در آن قرار دارید مثل هواپیما، جلسه و ... را برای دستگاه مشخص کنید، به این ترتیب دستگاه می تواند صدای نویز محیط را از صدای شما حذف کرده و صدای شما را واضح تر به گیرنده ارسال کند.
- کاهش تعداد واژگان قابل قبول : می توان دایره واژگان مورد استفاده را کاهش داد. یعنی اگر از ابتدا مشخص باشد که سیستم تشخیص گفتار قرار است در چه محیطی و برای چه کاربردی مورد استفاده قرار گیرد می توان عملکرد سیستم را با شناسایی نویز محیط و در نظر گرفتن فقط واژگان مورد استفاده آن محیط یا کاربرد بسیار بهتر کرد. زیرا به دلیل کوچکتر بودن بانک اطلاعاتی واژگان قابل تشخیص، می توان جستجوی دقیقتر و سریعتری انجام داد. کاربرد این روش ها بیشتر در سیستمهای پاسخگوی تلفنی(مثل تلفن بانک) است که در آن ها کاربر با ادای فرامین گفتاری از سیستم می خواهد تا خدمات مربوطه را انجام دهد.

۴. اصول کارکرد :

فرآیند تشخیص و ترجمه صداها یا گفتار به صورت واژه های متنی، شامل ۲ بخش است:

- تشخیص صدای آشکار سازی شده و تولید واژه هایی که احتمالا معادل با صدای ورودی است.
- استفاده از روش های گوناگون برای انتخاب واژه نهایی از بین کاندید های فاز قبلی.

در فاز اول نرم افزار وظیفه دارد تا triphoneها را تشخیص دهد و بر اساس مقابله آنها با فهرستی از واژه ها یک یا چند واژه نهایی را مشخص کند.

در فاز دوم که به فرایند جستجو(Search Process) معروف است در واقع هسته اصلی نرم افزارهای تشخیص گفتار محسوب می شود. وظیفه این بخش آن است که رشته ای از واژگان خروجی بخش قبل را دریافت کرده و با تجزیه و تحلیل آنها و استفاده از قواعد زبان یک جمله خروجی بسازد. دقت و سرعت که در تضاد با یکدیگرند دو فاکتوری هستند که اساس کارآیی و قابلیت های چنین نرم افزار هایی را تعیین می کند. به عنوان مثال سرعت چنین نرم افزارهایی باید بطور حداقل برابر با سرعتی باشد که کاربر با آن سرعت مکالمه می کند. همانطور که گفته شد این بخش اساس کار نرم افزارهای تشخیص گفتار محسوب می شود و به همین دلیل روش های بکار گرفته شده جزء اسرار تکنولوژیک تولید کنندگان بوده و نمی توان براحتی درباره آن صحبت کرد. اما در مراجع علمی تصویر تقریبی از روش های پشت پرده ترسیم می شود. در یکی از این تکنیک ها، نرم افزار یک حدس اولیه انتخاب می کند و در مراحل بعدی احتمال درست بودن این حدس اولیه با اضافه شدن کلمات بعدی محاسبه شده و بر اساس آن تصمیم گیری می کند. استفاده از قوانین زبان یکی دیگر از این تکنیک هاست. محاسبات آماری معروف به زنجیره های مارکوف در این مرحله استفاده می شود.

فصل چهارم

لغات (Words)

لغات قسمت سازنده و بنیادی زبان هستند. هر زبانی که صحبت می شود یا نوشته می شود از لغاتی تشکیل شده است. هر حوزه ای از پردازش گفتار و زبان، از تشخیص گفتار گرفته تا ترجمه ماشینی و بازیابی اطلاعات در وب، احتیاج به دانش وسیعی درباره لغات دارند.

این بخش مدلهای محاسباتی املایی، تلفظی و ریخت شناسی(Morphology) لغات را معرفی می کند و سه موضوع اصلی را با تکیه بر دانش وابسته به فرهنگ لغات پوشش می دهد که عبارتند از :

تشخیص گفتار خودکار(Automatic Speech Recognition)، تصحیح اشتباهات املایی و تولید گفتار از روی متن (Text-To-Speech Synthesis : TTS)

سرانجام مهمترین مدلهای محاسباتی برای پردازش زبان و گفتار شرح داده خواهد شد، یعنی ماشین های اتومات. چهار مدل ماشینهای اتومات عبارتند از:

- اتوماتهای متناهی(محدود) و عبارتهای منظم(Finite-State Automata & Regular Expresion)
- مبدلات محدود (Finite-State Transducer) و مبدلات وزن دار (Weighted Transducers)
- مدل مخفی مارکوف (Hidden Markov Model)
- مدل زبانی Nگرم (N-gram Model)

عبارات منظم و اتوماتها

در این بخش عبارت منظم را معرفی می کنیم که استانداردی برای توصیف توالی های متن است.

اتوماتهای محدود نه فقط ابزاری ریاضی برای استفاده در عبارات منظم هستند همچنین یکی از ابزارهای مهم در زبانشناسی محاسباتی(Computational Linguistic) است. انواع مختلف اتوماتها مانند اتومات حالت محدود و عبارات منظم، مبدلات محدود، مدلهای مارکوف پنهان و گرامرهای Nگرم از اجزای اصلی تشخیص و تولید گفتار مصنوعی محسوب می شوند.

۱- عبارات منظم

عبارت منظم ابزار تئوری مهمی در تمامی علوم کامپیوتر و زبانشناسی است. یک عبارت منظم(که اولین بار توسط کلین- Kleen- در سال ۱۹۵۶ گسترش یافت) قاعده ای است در یک زبان مشخص که در تکنیکهای جستجو بر اساس متن غالبا استفاده می شود. یک رشته، هر توالی از کاراکترهای الفبا عددی (شامل لغات، اعداد، فاصله ها، نشانه گذاری ها) است. در اینجا یک فاصله، درست مثل کاراکترهای دیگر است و ما آنرا بوسیله یک علامت نشان می دهیم.

یک عبارت منظم نمادهای جبری برای مشخص کردن مجموعه ای از رشته هاست. بنابراین آنها می توانند برای تعیین رشته های جستجو بخوبی استفاده شوند. می خواهیم بحث را با صحبت درباره عبارات منظم بعنوان روشی برای انجام جستجو در متن آغاز کنیم و پیش برویم.

جستجو در عبارت منظم به یک الگو که می خواهیم جستجو کنیم و به مجموعه ای از متون برای جستجو در آن احتیاج دارد. تابع جستجو که از میان مجموعه ای از متون عمل خواهد کرد، همه متونی که شامل الگو باشند را بر می گرداند. در یک سیستم بازیابی اطلاعات مانند یک موتور جستجوی وب، نتایج ممکن است شامل همه مستندات یا صفحات وب شود. در یک پردازشگر لغت نتایج ممکن است لغات منحصر بفرد یا شامل چند خط از سند باشد. بنابر این زمانی که ما الگوی جستجویی داریم، فرض خواهیم کرد که موتور جستجو خطی از سند را بر می گرداند. این چیزی است که خط فرمان grep در یونیکس انجام می دهد.

۱-۱ الگوهای عبارت منظم

ساده ترین نوع عبارت منظم، توالی از کاراکترهای ساده است. بعنوان مثال برای جستجوی لغت "Buttercup" ما عبارت /Buttercup/ را تایپ می کنیم. بنابراین عبارت منظم /Buttercup/ با هر رشته ای شامل زیر رشته Buttercup مطابقت می یابد.(بعنوان مثال خط "I am called Buttercup") در اینجا ما علامت اسلش را اطراف هر عبارت منظمی قرار خواهیم داد تا از الگو تشخیص داده شود.اسلش بخشی از عبارت منظم نیست.

رشته جستجو می تواند شامل یک حرف(مثل /!/)یا توالی از حروف(مثل /urgl/) باشد. می توان برای جستجو تعیین کرد که تنها اولین مورد یافته شود یا بیشتر از یک مورد را بدهد.

عبارت منظم به بزرگ یا کوچک بودن حروف حساس است، یعنی شکل کوچک /s/ با حرف بزرگ آن یعنی /S/ متفاوت است. این یعنی اگر الگو /woodchucks/ باشد، با رشته Woodchucks مطابقت ندارد. ما این مسئله را با استفاده از براکت، به معنی یا، حل می کنیم. یعنی الگوی /[Ww]/ با w یا W جور می شود و بهمین ترتیب داریم :

/[A-Z]/ و /[a-z]/ و /[0-9]/

موارد مطابقت یافته	عبارت منظم
Woodchuck یا woodchuck ‘a’ یا ‘b’ یا ‘c’ هر عددی	/[Ww]oodchuck/ /[abc]/ /[1234567890]/

با اسفاده از براکت می توان عبارت منظم را طوری تعریف کرد که شامل یک کاراکتر نباشد. اینکار بوسیله علامت ^ انجام می شود که باید بعد از [و قبل از حرفی که نباید در الگو باشد قرار گیرد. این تنها زمانی درست است که بعنوان اولین سمبل بعد از [باشد وگرنه جزء خود عبارت در نظر گرفته می شود. مثال :

موارد مطابقت یافته	عبارت منظم
هیچ یک از حروف بزرگ انگلیسی نه s یا نه S e یا ^ با الگوی a^b جور می شود	[^A-Z] [^Ss] [e^] a^b

اما برای اینکه بتوانیم woodchuck را از woodchucks تشخیص دهیم باید چگونه عمل کنیم؟ چون می خواهیم s باشد یا نباشد نمی توان از [] استفاده کرد. بنابراین از /؟/ استفاده می کنیم به این معنی که یا حرف قبل از ؟ یا هیچ. بنابراین :

موارد مطابقت یافته	عبارت منظم
woodchuck یا woodchucks color یا colour	woodchuck?s colou?r

روشی برای اینکه بتوان تعداد یک چیز را مشخص کرد وجود دارد. مثلا برای اینکه یک عبارت منظم با این عبارات جور شود : ba , baaa , baaaaaa , ...

می توان از * Kleen (کلین استار) استفاده کرد که به معنی تعداد تکرار است. می تواند صفر یا بیشتر باشد. بنابراین /a*/ یعنی رشته ای از صفر یا تعداد بیشتری a. عبارت منظم /a*/ با این الگوها می تواند جور شود: a, aaa, aaaaa, ... بنابراین عبارت منظمی که شامل حتما یک a و بیشتر است به اینصورت است : /aa*/

/[ab]*/ یعنی عبارت منظمی که شامل صفر یا بیشتر از aها و یا bها است. یعنی الگوهایی مثل aaa, ababab , bbb.

یادآوری می کنیم که RE برای یک عدد صحیح یک رقمی /[0-9]/ بود.RE برای یک عدد صحیح چند رقمی هم بصورت /[0-9][0-9]*/ است.

برای نمایش عدد صحیح چند رقمی بهتر است از Kleen + استفاده کنیم که به معنی تعداد یک یا بیشتر است.(برخلاف * که به معنی تعداد صفر یا بیشتر است)

یکی از مهمترین کاراکترهای مخصوص نقطه است. /./ یک جایگزین شونده است که با هر کاراکتری مطابقت دارد. مثال :

مثال	موارد مطابقت یافته	عبارت منظم
Begin , beg'n , begun	هر کاراکتری که بین beg و n قرار می گیرد	/beg.n/

جایگزین شونده می تواند همراه با * برای هر رشته ای از کاراکترها استفاده شود. مثل /abbas.*ali/ برای اینکه خود علامت شامل عبارت منظم شود باید بعد از آن علامت بک اسلش قرار داد. مثال /the dog\./ در این حالت علامت نقطه دیگر به معنای کاراکتر چایگزینی نیست.

علامت(Anchor) \b هم مشخص کننده حدود کلمه است. مثلا برای عبارت منظم /\bThe\b/ الگوی The مطابقت دارد ولی Other نه. همچنین در مورد /\b99/ که با 99 جور می شود ولی با 299 نه.

۲-۱ تفکیک، گروهبندی و اولویت(Disjunction , Grouping and Precedence)

جداکننده : فرض کنیم عبارت جستجویی به صورت "Dog or Cat" داشته باشیم. یعنی حاصل جستجو شامل عباراتی که کلمات Dog یا Cat دارند است. در عبارت منظم این کار را با براکت [] نمی توان انجام داد پس نیاز به عملگر جدیدی داریم بنام جداکننده یا Pipe (|). عبارت منظم /Dog|Cat/ با رشته هایی که Dog یا Cat دارند مطابقت دارد.

اما چگونه می توانیم عبارت منظم را هم برای الگوی Guppy و هم Guppies مشخص کنیم؟ مطمئنا RE اینطور نمی شود: /Guppy|ies/. بنابراین بوسیله پرانتز اولویت تعیین می کنیم که ابتدا عملگر جداکننده در نظر گرفته شود. یعنی RE صحیح اینطور می شود : /Gupp(y|ies)/

اگر * را همراه با () استفاده کنیم یعنی می توان رشته ای را که درون () است را صفر بار یا بیشتر تکرار کرد. مثال : /(dog)*/

اولویت عملگرهای RE به این ترتیب است :

()

+ ؟ []

Anchor و جملات

(Pipe) |

* از جملات اولویت بالاتری دارد یعنی /*The/ با Theee جور می شود نه با TheThe. جملات از | اولویت بالاتری دارند یعنی /Dog|Cat/ با Cat یا Dog جور می شود.

/*[a-z]/ می تواند با o یا on یا onc یا once جور شود. بعبارت دیگر REها همیشه با بزرگترین رشته که می توانند جور می شوند.

۲- اتوماتها

با توجه به اهمیت اتوماتها در بحث تشخیص گفتار توضیحات مقدماتی در اینباره شرح داده خواهد شد ولی می توانید برای بدست آوردن اطلاعات بیشتر به منابع نظریه زبانها و ماشینها مراجعه نمائید.

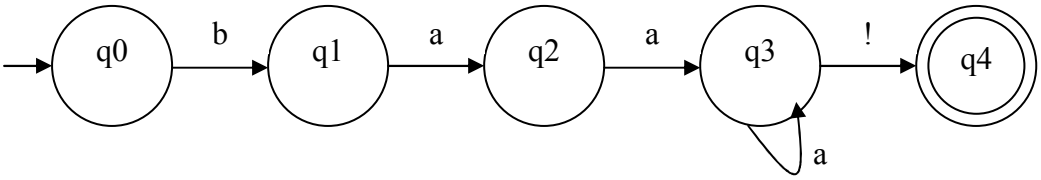
۲-۱ اتومات حالت محدود(Finite-State Automata)

عبارت منظم بیشتر از یک فرازبان مناسب برای جستجوی متن است. اولا اینکه، عبارت منظم روشی برای وصف کردن ماشین حالت محدود(FSA) است. هر عبارت منظم می تواند بوسیله یک FSA اجرا شود و بطور بالعکس هر ماشین حالت محدود را می توان با یک عبارت منظم وصف کرد. دوم اینکه، RE روشی برای توصیف کردن نوعی ویژه از زبانهای رسمی بنام زبانهای منظم است. هر دوی عبارت منظم و FSA برای توصیف زبان منظم(زبان منظم مجموعه ای از رشته هاست که هر رشته از علامتهای تعریف شده آن زبان(آلفابت) تشکیل شده است). استفاده می شوند. اگرچه ما بحث را با کاربرد FSA در اجرای عبارات منظم شروع می کنیم ولی FSAها زمینه کاربرد وسیعی دارند.

۲-۲ استفاده از FSA در تشخیص صدای گوسفندی : ....baaa

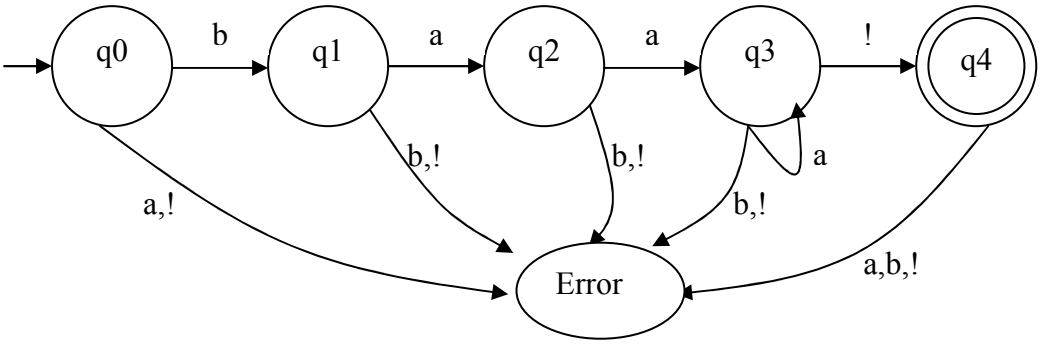
ما زبان گوسفندی را بعنوان رشته ای مطابق زیر تعریف کرده ایم: ...., baa! , baaa! , baaaaa!

RE برای آن بصورت /Baa+!/ است و FSA برای آن به شکل زیر است :



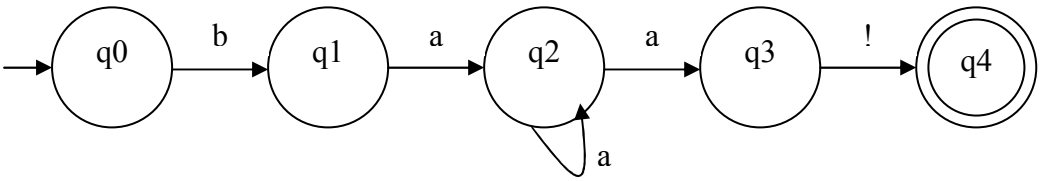
اتومات یک گراف جهت دار است شامل تعداد محدودی راس(نود)، بهمراه تعدادی لینک بین هر جفت نود(Node) نودها بوسیله دایره و لینکها که نشانه انتقال از یک نود به نود دیگر است توسط پیکانهایی نمایش داده می شوند.FSA می تواند برای تشخیص(پذیرش) رشته یک زبان بکار برده شود.

اضافه کردن حالت Error به ماشین :



۳-۲ اتومتهای محدود غیر قطعی (non-Deterministic FSAs : NFSA)

اجازه دهید بحثمان را در مورد کلاس دیگری از FSA ادامه دهیم، یعنی NFSA. شکل زیر را ملاحظه کنید که چه شباهتی با شکل قبل دارد.



در این ماشین حالت Error نداریم.

تفاوت بین این دو ماشین در این هست که حلقه تکرار ایجاد a در حالت q2 است بجای q3. این ماشین هم برای پذیرش صدای گوسفندی است. در این ماشین زمانی که به حالت q2 می رسیم اگر یک a ببینیم نمی دانیم که آیا باید در همین حالت بمانیم یا به حالت بعدی برویم. اتومات با این حالت تصمیم گیری را ماشین غیر قطعی می نامند و اتوماتی که برای هر ورودی مشخص است تا به چه حالتی برود ماشین قطعی گفته می شود.

۳- مبدلات حالت محدود و شکل شناسی (Morphology And Finite-State Transducers)

در قسمتهای قبل به معرفی عبارت منظم پرداختیم، بعنوان مثالی نشان دادیم که چگونه یک رشته مفرد توانست در پیدا کردن هر دو کلمه "woodchuck" و "woodchucks" در موتور جستجو متنی استفاده شود. فرآیند جستجو برای یافتن حالت جمع یا مفرد این کلمه آسان است چون در حالت جمع به آخر آن تنها یک s اضافه می شود. اما فرض کنید که ما در جستجوی کلمات دیگری هستیم، مثلا fox, fish, goose . فرآیند جستجو برای حالت جمع این کلمات

چیزی بیشتر از اضافه شدن یک s است. حالت جمع fox، foxes است، برای goose ، حالت جمع geese است، همچنین برای کلمه fish که حالت جمع آن تغییر نمی کند و همان fish است.

قواعد املائی به ما می گویند که لغات انگلیسی که به y ختم می شوند هنگام جمع y به i تبدیل شده و به آخر آن es اضافه می شود. دستورات شکل شناسی به ما می گویند که کلمه fish حالت جمع ندارد و حالت جمع کلمه goose با تغییر حروف صدادار آن تشکیل می شود یعنی geese.

مسئله تشخیص اینکه لغت foxes به دو قسمت fox و es شکسته می شود را تجزیه شکلی(Morphological Parsing) نامند. تجزیه به معنای دریافت ورودی و تولید ساختارهایی برای آن است. می خواهیم از اصطلاح "تجزیه" خیلی وسیعتر استفاده کنیم، شامل انواع زیادی از ساختارهایی که ممکن است ایجاد شوند: ریخت شناسی، قواعد صرف و نحوی، قواعد معنایی و عملگرایانه.

در عمل تجزیه فقط بحث شکل حالت جمع و مفرد مطرح نیست. مثلا برای لغاتی مثل talking , going و ... . می خواهیم آنها را به دو دسته شامل ریشه لغت بعلاوه ing تجزیه کنیم. بنابراین برای شکل ورودی going ممکن است بخواهیم فرم تجزیه شده ای بصورت VERB-go + GERUND-ing تولید کنیم. این بخش می خواهد آگاهی هایی مربوط به ریخت شناسی را که برای نمایش زبانهای مختلفی احتیاج می شود بررسی کند و اجزاء اصلی یک الگوریتم مهم برای تجزیه ریختی را معرفی کند، یعنی مبدل حالت محدود.

مسئله ای دیگر این است که چرا ما شکل جمع همه اسامی یا شکل ingدار همه افعال را در فرهنگ لغت لیست نمی کنیم؟ بدلیل اینکه مثلا ing یک پسوند است که به هر فعلی اضافه می شود یا بطور مشابه s به اغلب اسامی اضافه می شود. بنابراین ایده لیست کردن همه اسامی و افعال می تواند ناکارآمد باشد. بنابراین مطمئنا نمی توانیم همه اشکال مختلف هر لغتی را لیست کنیم.

۳-۱ تجزیه ریختی حالت محدود(Finite-State Morphological Parsing)

اکنون اجازه دهید به بحث تجزیه ریختی زبان انگلیسی بپردازیم. به یک مثال ساده توجه کنید :

تجزیه حالت جمع اسم و ingدار شدن فعل. هدف ما این خواهد بود که وروردی هایی مثل شکل بعد بدهیم و خروجی های لازم را تولید کنیم:

ورودی	خروجی
Cats	Cat + N + PL
Cat	Cat + N + SG
Cities	City + N + PL
Geese	Goose + N + PL
Goose	Goose + N + SG یا Goose + V
Gooses	Goose + V + 3SG
Merging	Merge + V + Pres-Part

Caught	Catch + V + Past یا Catch + V +Past-Part

ستون دوم شامل ریشه ای از هر لغت همراه با ویژگی های ریختی مناسب آن است. این ویژگی ها اطلاعات اضافه ای درباره ریشه لغت است. بعنوان مثال ویژگی N+ به این معنی است که لغت اسم است، SG+ یعنی مفرد، PL+ یعنی جمع است.

به منظور ساخت یک تجزیه کننده ریختی ما به موارد زیر نیاز داریم:

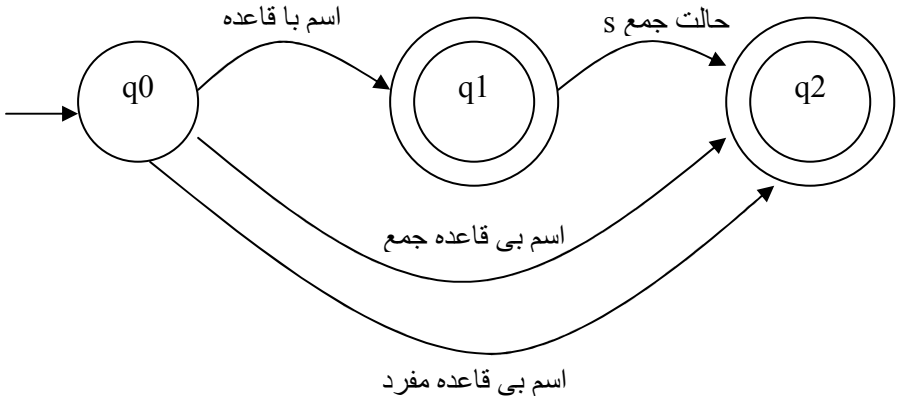
- لغت نامه(Lexicon) : لیستی از اصل لغات و پسوندها همراه با اطلاعاتی اساسی در مورد آنها (مثلا اینکه لغت اسم است یا فعل؟)
- Morphotactics : مدلی از ترتیب واژکها(کوچکترین واحد معنی دار) که کلاسهایی از واژکهایی که می توانند از کلاسهای دیگر پیروی کنند را شرح می دهد.
- قواعد املائی(Orthography Rules) : این قواعد برای مدلسازی تغییراتی که در یک لغت اتفاق می افتد استفاده می شود. مثلا زمانی که دو واژک با هم ادغام می شوند(مثل City+s که به Cities تبدیل می شود نه به Citys).

۲-۳ لغت نامه و مدل واژکها(The Lexicon And Morphotactics)

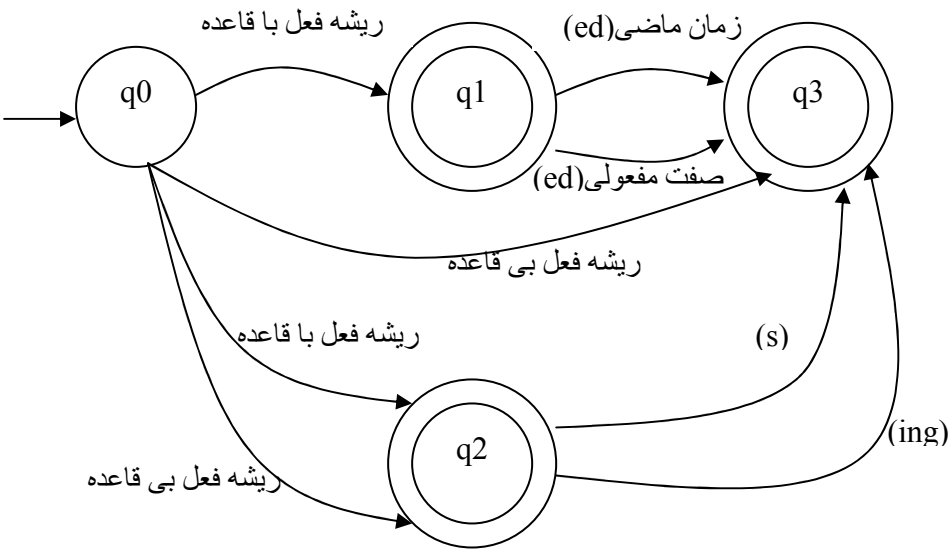
لغت نامه انباری از لغات است. ساده ترین لغت نامه می تواند شامل لیستی از هر لغتی از یک زبان به شکل زیر باشد.(هر لغتی، یعنی شامل اختصارات، نامها و اسامی اشخاص و ...)

a
AAA
AA
Aachen
Aardwolf
Aba
Aback و ...

از آنجایی که خیلی اوقات محتوای لغت نامه اینگونه نیست، لغت نامه های محاسباتی معمولا بوسیله لیستی از مدلهای واژک که به ما می گویند چگونه به هم می چسبند، ساختار بندی شده اند. روشهای زیادی برای مدلسازی واژکها وجود دارد. یکی از رایج ترین آنها ماشینهای حالت محدود FSA است. یک مدل بسیار ساده FSA برای اسمها در انگلیسی به شکل زیر است:



و یک مدل مشابه برای افعال در انگلیسی :



این لغت نامه سه کلاس ریشه دارد : ریشه فعل با قاعده، بی قاعده و حالت گذشته بعلاوه ۴ کلاس پسوند شامل : ed (فعل ماضی)، ed (صفت مفعولی)، ing و s برای فعل سوم شخص.

۳-۳ تجزیه ریختی بوسیله مبدلات حالت محدود (Morphological Parsing With Finite-State Transducers)

اکنون که دیدیم چگونه از FSA برای نمایش لغت نامه و ضمنا انجام تشخیص شکل شناسی استفاده می شود، اجازه دهید به بحث تجزیه ریختی بپردزایم. بعنوان مثال، برای ورودی Cats دوست داریم تا خروجی به شکل Cat +N +PL شود تا به ما بگوید که Cats یک اسم جمع است. می خواهیم اینکار را بوسیله ریخت شناسی دو سطحی انجام دهیم که ابتدا "کاسکونینی" آنرا مطرح کرد.

ریخت شناسی دو سطحی یک لغت را بعنوان تناظر بین سطح واژه ای(Lexical Level) – که یک الحاق ساده از واژکهایی است که یک لغت را می سازند نشان می دهد- و سطح ظاهری (Surface Level) – که املای درست لغت نهایی است – را بیان می کند.

تجزیه ریختی بوسیله ساخت قواعد نگاشتی - که توالی از حروف(مانند Cats) را در سطح ظاهری به واژکها و توالی از ویژگی ها می نگارد – انجام می شود.(مانند Cat +N +PL در سطح واژه ای)

شکل زیر دو سطح برای لغت Cats را نشان می دهد :

واژه ای (Lexical) :

C	a	t	+N	+PL

ظاهری (Surface) :

C	a	t	s

نکته اینکه سطح واژه ای اصل و ریشه کلمه را دارد. مطابق اطلاعات ریخت شناسی، +PL +N به ما می گویند که Cats یک اسم جمع است.

اتوماتی که ما برای انجام تناظر بین این دو سطح استفاده می کنیم FST یا Finite-State Transducer یا مبدلات حالت محدود نام دارد. یک FST این کار را توسط اتومات محدود انجام می دهد. بنابراین معمولا ما یک FST را بعنوان اتوماتی دو نواره تصور می کنیم که زوج رشته هایی را تشخیص می دهد یا تولید می کند. بنابراین FST کارکرد اصلی زیادی نسبت به FSA دارد. در جایی که FSA یک زبان رسمی را بوسیله تعریف یک مجموعه رشته ها تعریف می کند، FST ارتباط بین مجموعه رشته ها را مشخص می کند. این، چشم انداز دیگری از FST را بازگو می کند. بعنوان ماشینی که یک رشته را می خواند و نتایجی تولید می کند.

در اینجا خلاصه ای از چهار دسته موارد کاربرد مبدلات حالت محدود آورده شده است:

- FST بعنوان شناسنده(Recognizer): مبدل یک زوج رشته بعنوان ورودی می گیرد و Accept (پذیرش) بعنوان خروجی می دهد اگر زوج رشته در زبان موجود باشد و Reject (عدم پذیرش) می دهد اگر موجود نباشد.
- FST بعنوان مولد(Generator): ماشینی که زوج رشته های زبان را در خروجی می دهد. خروجی Yes یا No است بهمراه زوج رشته .
- FST بعنوان مترجم(Translator): ماشینی که یک رشته را می خواند و رشته ای دیگر را ترجمه شده می دهد.
- FST بعنوان مجموعه ای از بازگوگرها(Setrelator)

FST می تواند رسما به چند روش تعریف شود. ما روی تعریف زیر تاکید می کنیم که بر پایه چیزی بنام Mealy Machine است و تعمیمی از یک FSA ساده است.

Q : مجموعه ای محدود از n حالت q0,q1,q2,q3,….,qn

$\sum$: الفبای محدودی از علائم پیچیده است. هر علامت از یک جفت ورودی خروجی i:o تشکیل شده است.

q0: حالت آغازین است.

F : مجموعه ای از حالت های پایانی که در آنجا رشته توسط ماشین پذیرش می شود.

$\partial(q , i:o)$: معرف انتقال بین حالتها به ازای ورودی-خروجی مشخص است.

جائیکه FSA زبانی که دارای الفبای محدودی است را پذیرش کند- مانند زبان گوسفندی

FST – $\sum=\{ b,a,! \}$ زبانی را پذیرش می کند که روی جفت علامتهاست :

$\sum=\{ a:a, b:b, !:!, a:!, a,\varepsilon , \varepsilon:! \}$

۴- مدلسازی آماری زبان

مدلهای زبانی که به منظور بازشناسی گفتار و دیگر فناوری های زبانی بکار برده می شوند، برای اولین بار در سال ۱۹۸۰ مطرح شدند. از آن زمان تاکنون تلاشهای فراوانی برای اصلاح و توسعه این مدلها به جهت کاربرد در سیستمهای پیشرفته امروزی صورت گرفته است. مدلهای آماری زبان توزیع احتمال واحدهای زبانی مختلفی مانند آواها، کلمات و جملات یک متن را محاسبه می نمایند.

مدلسازی زبان تلاشی در جهت تسخیر قواعد زبان طبیعی به منظور بهبود کارآئی کاربردهای مختلف زبان طبیعی است. مدلهای زبانی برای کاربردهای مختلفی از فن آوری زبان از جمله بازشناسی گفتار، ترجمه ماشینی، طبقه بندی متون، بازشناخت نوری کاراکترها، بازشناسی دست نوشته و تصحیح هجاها و ... بکار گرفته شده اند.

مدلهای آماری زبان از روی دادگان متنی، پارامترهای بسیار زیادی را تخمین می زنند و بنابراین به حجم بالائی از دادگان تعلیم نیاز دارند. موفق ترین فن آوری SLM دانش بسیار محدودی را از آنچه که یک زبان براستی است، در نظر می گیرد. مشهورترین مدلهای زبانی(Nگرم ها) واقعیتی را مدل می کنند که زبان نیست، بلکه دنباله ای از نمادها است و هیچ ساختار عمیقی ندارد.

در ادامه برخی از فن آوری های بروز SLM مرور می شود:

تقریبا تمامی مدلهای آماری زبان احتمال یک جمله را به حاصل ضرب احتمال های شرطی تجزیه می نمایند.

استفاده از مدلهای زبانی نه تنها در سطح کلمه، بلکه در سطح آوا نیز کاملا رایج است. هاوس و نئوبرگ نشان دادند که محدودیت های موجود روی زنجیره آواها به عنوان روش موثری در شناسائی می تواند بکار گرفته شود. در کار انجام شده، نشان داده شد که این محدودیت ها به عنوان مشخصه قدرتمند در بازشناسی گفتار، حتی در مواردی که گفتار به بخش های متنوعی تعلق دارد، می تواند مورد استفاده قرار گیرد.

در هر مساله مدلسازی در ابتدا دو عامل اصلی می باید ابداع و تدوین شوند- ساختار مناسب و جامع برای مدل و سپس چگونگی تنظیم و محاسبه پارامترهای آن. بر اساس مطالعات و بررسی های انجام شده و با توجه به سوابق و کاربردهای متعدد، مدل پنهان مارکوف برای مدلسازی کلمات و مدلهای N-gram برای جملات در نظر گرفته شده اند. از آنجا که هر روش یادگیری نیاز به مجموعه ای از داده های آموزشی داد لذا روشی کارآمد و جامع برای نمایش متن باید در نظر گرفته شود.

در ادامه مدل مارکوف پنهان و مدل زبانی N-gram بطور مفصل شرح داده خواهند شد.