
بسم الله الرحمن الرحيم

دانشکده مهندسی برق و رباتیک

گروه الکترونیک

جستجوی کلمات کلیدی در رشته‌ی گفتار

دانشجو: هادی نادری

استاد راهنما

دکتر حسین مروی

استاد مشاور

دکتر امیدرضا معروضی

پایان نامه ارشد جهت اخذ درجه کارشناسی ارشد

بهمن ۱۳۹۱

تقدیم به روح پدرم...

و

تقدیم به خانواده ی عزیزم

ن از وضعف تو حاجت زارت وحا ام از عظمت وجودت سرگشته و حیران

من بیزجان ناقابل چه دارم که به تو تقدیم کنم...

مادر

تقدیم به مادرم...

ای کریمی که زمام اختیارم به دست اوست، تو دستم را بگیر

که اوقات مرا در خدمتت وقف و پیوسته به خدمت و بندگیت بگذرانی و اعمالم را مقبول حضرتت فرمایی

تا کردار و گفتارم، همه یک جهت و خالص برای تو باشد و احوالم تا ابد به خدمت و طاعتت مصروف گردد

ای کریمی که تمام اعتماد و توکلم بر اوست، و شکایت از احوال پریشانم به حضرت اوست

و جوارحم را به خدمتت مقام بندگیت قوت بخشی و دلم را عزم ثابت ده

و ارکان وجودم را به خوفت، سخت بنیان ساز و پیوسته به خدمت در حضرتت بدار

تا آنکه من در میدان طاعتت، بر همه پیشینیان سبقت گیرم و از همه شتابندگان به درگاهت زودتر آیم

و عاشقانه با مشتاقانت به مقام قربت خاص بشتابم و مانند اهل خلوص به تو نزدیک گردم

تو خود به بندگانت دستور عبادت دادی و امر به دعا فرمودی و اجابت ضمانت کردی

اینک من به دعا رو به سوی تو آوردم و دست حاجت به درگاه تو دراز کردم

پس به عزت و جلالت قسم که دعایم مستجاب گردان و مرا (که نومید از وصال تو نیست) برسان و امیدم را به فضل و کرمت ناامید مگردان.

تقدیر و تشکر:

تو سپاس خداوندم که به انسان نعمت عقل و سلامتی و آرامش و امنیت را عطا فرمودی و همواره حامی و پشتیبانم بودی تا از ناملایمات و سختی های زندگی عبور کرده و راه عبادتت را مشتاقانه تر بپیمایم....

در سته پدر عزیزم، به خواهرشتت تو را از ما جدا کرد، اما هنوز هم وجودت را احساس می کنم... از تو ممنونم که همواره باد لسوزی و کار و تلاش، حامی و پشتیبانم بودی...

خواهر عزیزم... اگر این دوره از تحصیلاتم و فقط این پایان نامه را به پایان رساندم، بی گمان مدیون زحمات و حمایت تو هستم... از خانواده عزیزم هم که همیشه حامی و پشتیبان من بودند نهایت کمال تقدیر و تشکر را دارم...

از استاد عزیزم جناب آقای دکتر هروی که با راهنمایی ها و صبر و حوصله ی فراوان نقش مهمی را در به پایان رسیدن این پروژه داشتند کمال تقدیر و تشکر را دارم و از خداوند متعال بهترینها را برایشان مسالت می نمایم.

و اما باز نا توانم که چه بگویم و چگونه از تو قدردانی کنم ... مانند ققط می توانم بگویم.... دوستت دارم...

تعهد نامه

اینجانب **هادی نادری** دانشجوی دوره کارشناسی ارشد رشته **الکترونیک سیستم** دانشکده **مهندسی برق و رباتیک** دانشگاه صنعتی شاهرود نویسنده پایان نامه با عنوان :

جستجوی کلمات کلیدی در رشته ی گفتار

تحت راهنمائی آقای دکتر **حسین مروی** متعهد می شوم .

- تحقیقات در این پایان نامه توسط اینجانب انجام شده است و از صحت و اصالت برخوردار است .
- در استفاده از نتایج پژوهشهای محققان دیگر به مرجع مورد استفاده استناد شده است .
- مطالب مندرج در پایان نامه تاکنون توسط خود یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جا ارائه نشده است .
- کلیه حقوق معنوی این اثر متعلق به دانشگاه صنعتی شاهرود می باشد و مقالات مستخرج با نام « دانشگاه صنعتی شاهرود » و یا « Shahrood University of Technology » به چاپ خواهد رسید .
- حقوق معنوی تمام افرادی که در به دست آمدن نتایج اصلی پایان نامه تأثیرگذار بوده اند در مقالات مستخرج از پایان نامه رعایت می گردد.
- در کلیه مراحل انجام این پایان نامه ، در مواردی که از موجود زنده (یا بافتهای آنها) استفاده شده است ضوابط و اصول اخلاقی رعایت شده است .
- در کلیه مراحل انجام این پایان نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته یا استفاده شده است اصل رازداری، ضوابط و اصول اخلاق انسانی رعایت شده است.

تاریخ

امضای دانشجو

مالکیت نتایج و حق نشر

- کلیه حقوق معنوی این اثر و محصولات آن (مقالات مستخرج ، کتاب ، برنامه های رایانه ای ، نرم افزار ها و تجهیزات ساخته شده است) متعلق به دانشگاه صنعتی شاهرود می باشد . این مطلب باید به نحو مقتضی در تولیدات علمی مربوطه ذکر شود .
- استفاده از اطلاعات و نتایج موجود در پایان نامه بدون ذکر مرجع مجاز نمی باشد.

*** متن این صفحه نیز باید در ابتدای نسخه های تکثیر شده پایان نامه وجود داشته باشد .**

چکیده:

تشخیص کلمات کلیدی یا keyword-spotting در حالت کلی به معنای یافتن یک کلمه‌ی کلیدی در یک پرونده‌ی نوشتاری و یا گفتاری است. در این تحقیق، یک روش جدید تشخیص یا بازشناسی کلمات کلیدی در زبان فارسی، در دوحالت پیوسته و گسسته معرفی شده است. در هر دوحالت تشخیص کلمات کلیدی در گفتار پیوسته و گسسته، از روش Dynamic Time warping(DTW) استفاده شده است که با سیستم‌هایی که بر اساس مدل مخفی مارکوف طراحی شده و امروزه به طور گسترده از آنها استفاده می‌شود، متفاوت هستند.

روش ارائه شده برای تشخیص کلمات کلیدی در حالت پیوسته، بر پایه‌ی حالت اصلاح شده‌ای از روش قدیمی Dynamic Time Warping است، که یک روش ابتدایی برای محاسبه‌ی میزان شباهت دو دنباله‌ی متغیر با زمان است. در مرحله‌ی پردازش، سیگنال گفتار به فریم‌هایی با طول کم تقسیم می‌شود. هر فریم به صورت یک بردار کوانتیزه شده از ویژگی‌ها نمایش داده می‌شود. هم کلمه‌ی کلیدی و هم گفتار اصلی که جستجو در آن انجام می‌شود، به صورت دنباله‌ای یک بعدی از اندیس‌های کتاب کد تبدیل می‌شوند. سپس دنباله‌ی اندیسهای گفتار اصلی به چندین جزء تقسیم شده و فاصله‌ی هرکدام از این بخشها طبق معیار DTW با اندیس‌های کلمه‌ی کلیدی بدست می‌آید. برای هر بخش یک امتیاز اعوجاج محاسبه می‌شود و بخشی که کمترین امتیاز اعوجاج را داشته باشد، به عنوان محل احتمالی حضور کلمه‌ی کلیدی معرفی می‌شود.

در روش ارائه شده برای تشخیص کلمات کلیدی در حالت گسسته نیز از روش DTW استفاده شده است. در این حالت، ابتدا از نمونه‌های مختلف از یک کلمه‌ی کلیدی خاص که توسط یک یا چند گوینده بیان شده‌اند و دارای طول متفاوت هستند، بردارهای ویژگی را استخراج کرده و سپس کلمه‌ای که دارای کوچکترین طول است را به عنوان نمونه‌ی مرجع انتخاب می‌کنیم. سپس با استفاده از روش DTW، مسیر همترازی نمونه‌ی مرجع و سایر

نمونه‌ها را بدست آورده و از روی مسیر همترازی، ابعاد ماتریس ویژگی سایر نمونه‌ها را هم بعد با نمونه‌ی مرجع می‌سازیم. در نهایت از ماتریس ویژگی تمام نمونه‌ها، میانگین می‌گیریم تا یک نمونه‌ی مرجع عام برای هر کلمه‌ی کلیدی بدست آید. در ادامه از این نمونه‌ی مرجع عام در فرایند تشخیص استفاده می‌کنیم.

کلمات کلیدی: تشخیص کلمات کلیدی - جستجوی صوتی– پردازش گفتار – تشخیص کلمات جدا – مدل مخفی مارکوف - الگوریتم DTW

فهرست مطالب

فصل پنجم: روش پیشنهادی ب: جستجوی کلمات کلیدی در گفتار پیوسته

فهرست تصاویر

فصل اول

فصل دوم

فصل سوم

فصل چهارم

فصل پنجم

فصل ششم

فهرست جداول

فصل چهارم

فصل پنجم

فصل ششم

فصل اول

مقدمه‌ای بر تشخیص کلمات کلیدی

۱-۱ مقدمه

تشخیص کلمات کلیدی شاخه‌ای از علم پردازش گفتار است که به شناسایی تعداد محدودی از کلمات در یک گفتار بلند می‌پردازد. امروزه افراد زیادی در زمینه‌ی ساده‌تر کردن و روان‌تر کردن ارتباط بین انسان و ماشین کار می‌کنند. چالش‌های فراوانی در این زمینه وجود دارد؛ که از جمله می‌توان به وجود کلمات غیر مربوط در مکالمات انسانها و صداهای غیر ارادی از جمله سرفه، فریاد و انواع نویزها اشاره کرد. اگر بتوان واژه‌هایی که در سیگنال گفتار وجود دارد را به نحو دقیقی استخراج کرد، آنگاه محاسبات ما دقیق‌تر و مقاوم‌تر خواهد بود. اما مردم به صورت واژه‌های جدا[۱] صحبت نمی‌کنند و هیچ مرز مشخصی بین واژه‌ها در سیگنال گفتار وجود ندارد؛ که این مساله باعث سخت‌تر شدن فرایند بازشناسی می‌شود. علاوه بر این، حتی اگر این کلمه‌ی کلیدی توسط یک گوینده‌ی واحد بیان شود، همیشه مقداری تغییرات در بیان‌های مختلف یک کلمه‌ی کلیدی وجود دارد. اینها همه مسائلی هستند که یک سیستم تشخیص کلمات کلیدی باید بتواند بر آنها فائق آید. از برخی نمونه‌های این سیستم‌ها می‌توان به سیستم‌های تماس صوتی و فرمان صوتی، موتورهای جستجوی صوتی و غیره اشاره کرد. تشخیص کلمات کلیدی کاربردهای مستقیم دیگری از نظیر دسترسی به اطلاعات درونی پرونده‌های صوتی و سیستم‌های نظارت پنهان بر مکالمات و غیره دارد.

سیستم‌های تشخیص کلمات کلیدی را می‌توان به دو گروه تقسیم کرد. سیستم‌های وابسته به گوینده و سیستم‌های مستقل از گوینده. درسیستم‌های وابسته به گوینده، مدل‌ها فقط برای یک گوینده‌ی خاص ایجاد می‌شوند و از آنها انتظار نمی‌رود که عملکرد مطلوبی برای سایر گوینده‌ها داشته باشند. در سیستم‌های مستقل از گوینده نیازمند ایجاد مدل‌هایی با عمومیت بیشتر هستیم و از این رو پیچیدگی مدل نیز بیشتر خواهد شد. بر خلاف سیستم‌های تشخیص دست نوشته،

[۱] - isolated word

سیستم‌های تشخیص کلمات کلیدی گفتاری دارای پیچیدگی بسیار زیادی هستند و این پیچیدگی به دلیل تغییرات فراوان در تلفظ‌ها، حتی برای یک گوینده‌ی واحد می‌باشد. به این دلیل که تلفظ یک کلمه توسط یک گوینده، وابستگی زیادی به متن و حالت گوینده دارد.

یک سیستم تشخیص کلمات کلیدی را می‌توان به عنوان یک کلاسه بند[۱] باینری در نظر گرفت که خروجی‌های آن مثبت و منفی هستند. برای خروجی یک سیستم تشخیص گفتار، چهار حالت متصور است: اولین حالت این است که یک کلمه‌ی کلیدی که در جمله وجود دارد، به درستی تشخیص داده شود؛ که به این حالت، مثبت صحیح[۲] می‌گوییم. دومین حالت این است که کلمه کلیدی در جایی که حضور ندارد، به اشتباه تشخیص داده شود. به این حالت مثبت اشتباه[۳] و یا هشدار اشتباه گفته می‌شود. در حالت سوم، کلمه‌ی کلیدی در جایی که حضور ندارد، تشخیص داده نیز نمی‌شود. به این حالت نیز صحیح منفی[۴] گفته می‌شود. و در نهایت در آخرین حالت، یک کلمه‌ی کلیدی در جایی که حضور دارد، تشخیص داده نمی‌شود، که به این حالت اشتباه منفی[۵] گفته می‌شود. چهار خروجی ممکن را می‌توان به صورت یک ماتریس به شکل زیر نمایش داد:

اشتباه مثبت	**صحیح مثبت**
صحیح منفی	**اشتباه منفی**

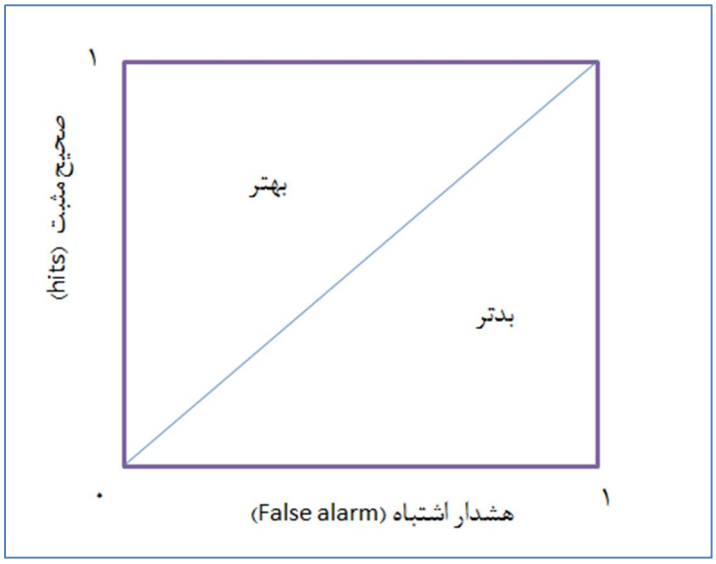
شکل ۱-۱ ماتریس خروجی‌های ممکن سیستم تشخیص کلمات کلیدی

۱ - classifier
۲ - True Positive(TP)
۳ - False Positive(FP) or False Alarm(FA)
۴ - True Negative(TN)
۵ - False Negative(FN) or miss

دقت کلاسه بند باینری توسط این رابطه محاسبه می‌شود:

$$\text{دقت} = \frac{\text{تعداد صحیح مثبت + تعداد صحیح منفی}}{\text{تعداد کل تشخیص‌های ممکن}}$$

از ماتریس خروجی می‌توان پارامترهای ارزیابی سیستم را استخراج کرد. یکی از معمول‌ترین پارامترهای ارزیابی سیستم، منحنی خصوصیات عملکردی دریافتی[1] است که با نام ROC شناخته می‌شود. منحنی ROC، نرخ صحیح مثبت (Hit) و نرخ اشتباه مثبت (FA) را در خود جای داده است. این منحنی اساسا نمودار تشخیص‌های صحیح حضور کلمه‌ی کلیدی (در زمانی که کلمه‌ی کلیدی واقعا حضور دارد) در مقابل تشخیص‌های اشتباه حضور کلمه‌ی کلیدی (در زمانی که کلمه‌ی کلیدی حضور ندارد) را نشان می‌دهد.



شکل ۱-۲ منحنی ROC

[1] - Receiver Operating Characteristic(ROC)

در شکل ۱-۲ خط قطری، منحنی ROC را به دو بخش تقسیم می‌کند. نقاط کار بالای خط قطری به عنوان عملکرد خوب و نقاط پایین خط قطری به عنوان عملکرد ضعیف شناخته می‌شوند. طراحی بهینه منجر به افزایش نرخ تشخیص صحیح و کم کردن نرخ هشدار اشتباه تا حد ممکن می‌شود. در یک سیستم ایده‌آل، نقطه‌ی کار در گوشه‌ی بالا سمت چپ، مختصات (۰,۱)، واقع می‌شود که در این نقطه دارای دقت تشخیص ٪۱۰۰ خواهیم بود. روش‌های متنوعی برای تشخیص کلمات کلیدی وجود دارد. معمول‌ترین این روش‌ها، مدل مخفی مارکوف، تطبیق نمونه[۱] و شبکه‌های عصبی[۲] هستند. امروزه از سیستم‌هایی که بر اساس مدل مخفی مارکوف طراحی می‌شوند، به دلیل کارایی بسیار بالا و وجود داده‌های آموزشی فراوان، بیشتر از سایر روش‌ها استفاده می‌شود. اشکال این روش این است که دارای پیچیدگی بسیار زیاد است و نیازمند وجود داده‌های آموزشی فراوان به همراه فایل‌های رونوشت[۳] می‌باشد. با این وجود، روش‌هایی که بر پایه‌ی تطبیق نمونه‌ها هستند، امروزه هنوز در سیستم‌هایی با مقیاس کوچک کاربرد دارند.[۷]

۱-۲ اهداف تحقیق:

هدف از این تحقیق، پیاده سازی یک سیستم ساده و کارامد تشخیص کلمات کلیدی مستقل از گوینده در گفتار پیوسته و گسسته، با استفاده از روش تطبیق نمونه‌ها می‌باشد. گرچه روش‌های آماری مانند مدل مخفی مارکوف دارای دقت بالایی هستند، در مقابل روشهای بر پایه تطبیق نمونه به علت اینکه از لحاظ پیاده سازی سخت افزاری آسان‌تر هستند، هنوز هم در کاربردهایی با مقیاس کوچک نظیر تلفن همراه و ... استفاده می‌شوند.[۸]

۱ - Template Matching
۲ - Neural Networks
۳ - Label

مهمترین نقص روش‌هایی که بر اساس تطبیق نمونه‌ها هستند این است که این روش‌ها قادر به بهره گیری از حجم زیادی از داده‌های آموزشی نمی‌باشند. با این وجود محققان روی روش‌هایی کار کرده‌اند که الگوریتم‌های تطبق نمونه‌ها نیز بتوانند از وجود داده‌های آموزشی بهره ببرند.

به منظور بکارگیری الگوریتم Dynamic Time Warping در گفتار پیوسته، گفتار باید به کلمات جدا شکسته شود که نیازمند این است که محل دقیق شروع و پایان هر کلمه مشخص باشد. این مشکل در زمانی که گوینده به صورت غیر رسمی مشغول به صحبت است و از قواعد گرامری پیروی نمی‌کند، بسیار برجسته می‌شود. یک راه برای حل این مشکل این است که گفتار را بخش بندی کنیم و با محتوی هر بخش مانند یک کلمه‌ی جدا رفتار کرده و فرایند تشخیص را انجام دهیم. این روش بسیار ناکارامد است و کارایی کلی سیستم تا حدود زیادی به کارایی الگوریتم بخش بندی وابسته است. یک راه دیگر برای حل این مشکل، اصلاح محدودیت‌های مرزی قدیمی در روش DTW است تا مرزها دارای انعطاف بیشتری باشند. در این روش نیازمند دانستن مرز دقیق کلمات برای بکارگیری الگوریتم DTW نخواهیم بود. روش‌های متعددی برای اصلاح مرز در الگوریتم DTW بررسی شده است[۱۲] و[۱۱]. در روشی که ما در حالت پیوسته بکار برده‌ایم، از الگوریتم DTW با مرزهای غیر محدود استفاده شده است تا بتوانیم کلمات کلیدی در گفتار پیوسته را تشخیص دهیم.

در بخش تشخیص گفتار گسسته نیز از الگوریتم DTW به منظور میانگین گیری از کلمات کلیدی برای ایجاد یک مدل عام استفاده شده است. نحوه انجام این کار بدین صورت است که از نمونه‌های مختلف از یک کلمه‌ی کلیدی که دارای طول متفاوت هستند میانگین‌گیری می‌کنیم تا برای هرکلمه‌ی کلیدی یک مدل ایجاد شود. در ادامه، این مدل را در شبکه قرار می‌دهیم و گفتار تست با آن مقایسه می‌شود.

تمامی آزمایشات انجام شده در این کتاب، بر روی دادگان گفتاری پیوسته‌ی فارس دات متعلق به لابراتوار پردازش گفتار دانشگاه تهران، پیاده سازی شده است.

۱-۳ ساختار پایان نامه

کتابی که پیش روی شماست، در هفت فصل تنظیم شده است که بدین ترتیب می‌باشند:

فصل اول، شامل کلیات مساله‌ی تشخیص کلمات کلیدی و اهداف طرح می‌باشد.

در **فصل دوم**، روش‌های رایج در تشخیص کلمات کلیدی مورد بحث و بررسی قرار می‌گیرند.

در **فصل سوم**، تعدادی از کارهایی که سایر محققان در زمینه‌ی تشخیص کلمات کلیدی در گذشته انجام داده‌اند، به صورت مختصر بررسی می‌شود.

در **فصل چهارم**، روش پیشنهادی این پایان نامه در بخش جستجوی کلمات کلیدی در رشته‌ی گفتار گسسته مورد بررسی، آزمایش و ارزیابی قرار خواهد گرفت.

در فصل پنجم، روش پیشنهادی این پایان نامه در بخش جستجوی کلمات کلیدی در گفتار پیوسته مورد بحث و بررسی قرار خواهد گرفت.

در **فصل ششم**، به صورت تفصیلی، نتایج مربوط به سیستم طراحی شده در فصل ۵، مورد آزمایش و ارزیابی قرار خواهد گرفت.

در **فصل هفتم**، نتیجه گیری کلی از روش‌های پیشنهادی انجام، و در مورد اقداماتی که در آینده انجام خواهد شد، بحث می‌شود.

فصل دوم

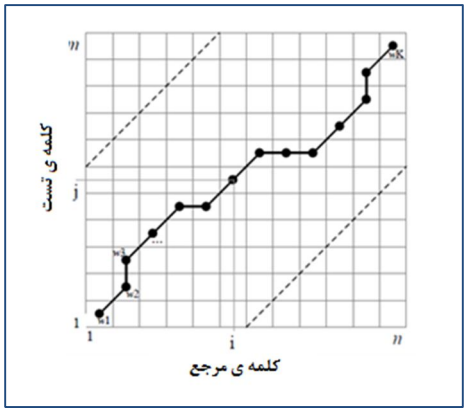
روش‌های رایج در تشخیص کلمات کلیدی

۲-۱ روش‌های رایج در تشخیص کلمات کلیدی:

گوینده‌های مختلف، یک واژه را به طرق مختلف تلفظ می‌کنند. حتی اگر گوینده یکسان باشد بازهم تغییراتی در کلمه‌ی بیان شده، وجود خواهد داشت. سرعت تلفظ، نوع متن، حالت گوینده و جایگاه کلمه در جمله، همگی باعث ایجاد این تغییرات می‌شوند. زمانی که در مورد گفتار محاوره‌ای سخن می‌گوییم، این مشکل برجسته‌تر نیز می‌شود. بنابراین برای مقایسه‌ی هر کلمه‌ی آزمون با کلمه‌ی مرجع، نیازمند الگوریتمی هستیم که تا حدودی نسبت به این تغییرات منعطف و مقاوم باشد. اولین روشی که بدین منظور مطرح شد، روش برنامه نویسی پویا بود که با نام نرمالیزه کردن زمان یا Dynamic Time Warping (DTW) [۱۰] شناخته می‌شد.

۲-۱-۱ Dynamic Time Warping (DTW)

Dynamic Time Warping، یک الگوریتم کارآمد به منظور یافتن مسیر همترازی غیر خطی بین دو دنباله‌ی زمانی است که از نظر طولی با همدیگر برابر نیستند. این الگوریتم در محاسبه‌ی شباهت دنباله‌های زمانی بسیار کارآمد است و اثر جابجایی و اعوجاج سیگنال را به حداقل می‌رساند. این روش یکی از اولین روشهایی است که برای تشخیص کلمات جدا بکار رفته است. در سیستم تشخیص کلمات کلیدی نیز یک کلمه‌ی کلیدی نمونه در سیستم ذخیره شده و با سایر کلمات در جمله مقایسه می‌شود.



شکل ۲-۱ شبکه DTW همراه با مسیر همترازی

همانطور که شکل ۲-۱ نشان می‌دهد، بردار ویژگی‌های کلمه کلیدی مرجع و کلمه‌ی ورودی در امتداد دو محور افقی و عمودی شبکه قرار داده می‌شوند. هرکدام از بلوک‌های شبکه، محتوی فاصله‌ی بین بردارهای ویژگی متناظر با آن بلوک می‌باشند. بهترین تطابق بین دو دنباله را می‌توان از مسیری در امتداد شبکه که فاصله‌ی تجمعی کل را مینیمم می‌کند، بدست آورد. این مسیر با یک خط پررنگ در شکل ۲-۱ نشان داده شده است. فاصله‌ی کلی بین داده‌ی مرجع و داده‌ی تست، همان فاصله‌ی تجمعی در طول مسیر است.

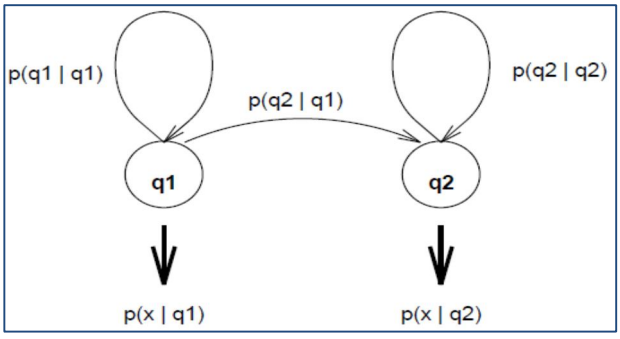
با توجه به شکل ۲-۱، واضح است که تعداد مسیرهای ممکن در طول شبکه به طور نمایی با افزایش طول کلمه، افزایش می‌یابد. با اعمال برخی از محدودیت‌ها [۱۰]، مسیرهای ممکن می‌تواند به تعداد محدودی کاهش یابد و محاسبات ساده‌تر گردد. این محدودیت‌ها اساسا مانع از به وجود آمدن مسیرهای بینهایت می‌گردند. محدویت‌های اعمال شده به این صورت هستند:

- **شرط یکنوایی**: مسیر باید به صورت یک تابع یکنوا افزایش یابد و امکان برگشت وجود ندارد. اگر **i** و **j** اندیس‌های مشخص کننده‌ی کلمه‌ی مرجع و کلمه‌ی آزمون باشند، مقدار آنها یا باید ثابت بماند و یا افزایش یابد.
- **شرط پنجره‌ی تنظیم**: مسیر بهینه نباید خیلی از مسیر قطری فاصله بگیرد. این شرط از این حقیقت ناشی می‌شود که مسیر ایده‌آل در امتداد خط قطری قرار دارد.
- **شرایط مرزی**: مسیر از گوشه‌ی پایین سمت چپ شروع می‌شود و به گوشه‌ی بالا سمت راست ختم می‌شود. از لحاظ منطقی، این شرط برای اطمینان از مقایسه‌ی کامل دو کلمه گذاشته می‌شود.
- **شرط محدودیت شیب**: مسیر نباید دارای شیب خیلی زیاد و یا خیلی کم باشد. شیب خیلی زیاد و یا خیلی کم، باعث مقایسه‌ی غیر واقعی یک الگوی کوچک با یک الگوی بزرگ و یا بر عکس خواهد شد.

با توجه به نوع کاربرد الگوریتم، این شرط‌ها ممکن است تغییراتی داشته باشند. در بازشناسی کلمات پیوسته، تعیین مرز دقیق کلمات مشکل است. از این رو ممکن است شرط نقطه‌ی شروع و پایان به نحوی تغییر یابد که بتوان نقاط شروع و پایان متفاوتی را در نظر گرفت[۱۱]. به منظور افزایش دقت تشخیص، می‌توان از تعداد نمونه‌های بیشتری به عنوان نمونه‌ی مرجع استفاده کرد[۸]. این سیستم‌ها به حافظه‌ی بیشتری برای ذخیره‌سازی نمونه‌ها احتیاج دارند. یک راه بهتر برای حل این مشکل این است که یک نمونه‌ی مرجع را ایجاد کنیم که محتوی اطلاعات سایر نمونه‌ها باشد. نمونه‌ی ایجاد شده باید مستقل از گوینده باشد و انعطاف بیشتری روی تغییرات سایر نمونه‌ها داشته باشد. محققان سعی کرده‌اند که از طریق میانگین‌گیری بردار ویژگی‌ها، تمام نمونه‌های یک کلمه‌ی کلیدی را در یک نمونه ترکیب کنند[۸].

۲-۱-۲ مدل مخفی مارکوف:

مدل مخفی مارکوف یک مدل آماری است که در آن فرض می‌شود که سیستم، یک فرایند مارکوف با حالتهای غیر قابل مشاهده باشد. خروجی هرکدام از این حالت‌ها قابل مشاهده است و هر حالت یک توزیع احتمالاتی روی خروجی‌های ممکن دارد. یک شکل ساده از یک مدل مخفی مارکوف دو حالته در شکل ۲-۲ آورده شده است.



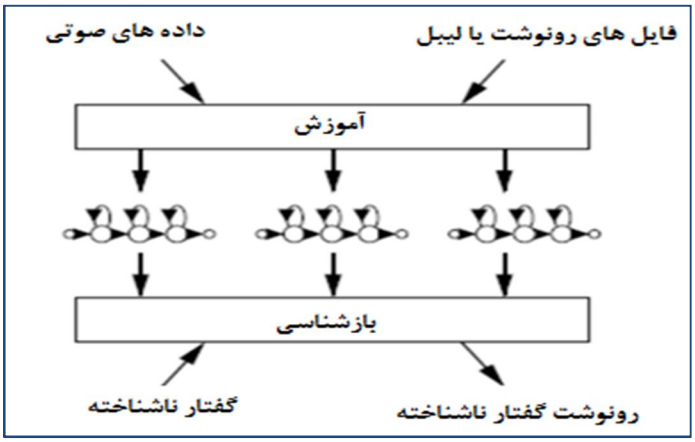
شکل ۲-۲ مدل مخفی مارکوف با دو حالت مخفی

HMMها به این دلیل مخفی هستند که حالت مدل یا q، مشاهده نمی‌شود. اما خروجی آن یا x قابل مشاهده است. دسته‌ی دیگر از احتمالات قابل مشاهده، احتمالات انتقال بین حالت‌ها ($P(q_i|q_j)$) هستند. با فرض اینکه سیستم، یک فرایند مارکوف مرتبه‌ی اول است، احتمال حالت بعدی یا **j** فقط به حالت فعلی یا **i** وابسته است. از دنباله‌ی قابل مشاهده‌ی خروجی‌ها، محتمل ترین جواب را می‌توان استنتاج کرد. در نتیجه با مشاهده‌ی یک دسته از خروجی‌ها، می‌توان محتمل‌ترین دنباله‌ی حالت‌ها را بدست آورد.

HMM، پر استفاده‌ترین مدل در پردازش گفتار و همچنین تشخیص کلمات کلیدی می‌باشد. هر کلمه به عنوان دنباله‌ای از واحدها، توسط دسته‌ی مشخصی از ویژگی‌های صوتی مدلسازی می‌شود. این مدل توانایی بکارگیری مدل‌های زبانی و گرامری را نیز در فرایند تشخیص دارد. اگر چه می‌توان از مدل‌های مارکوف چند سیلابی نیز استفاده کرد، اما مدل سازی تک واج‌ها بیشترین استفاده را در مدل مخفی مارکوف دارد. بازشناسی کلمات جدا توسط HMM را می‌توان در دو مرحله بیان کرد:

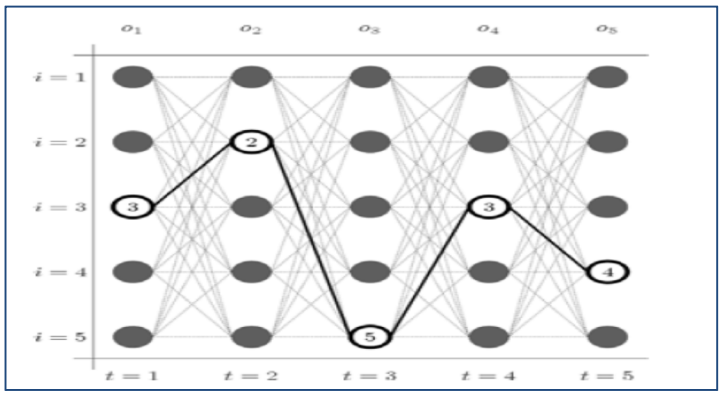
i. برای هر کلمه‌ی کلیدی در واژه نامه یک مدل کلمه‌ی کلیدی را می‌سازیم و با استفاده از داده‌های آموزشی، پارامترهای مدل را بدست می‌آوریم.
ii. دنباله‌ی مشاهدات که در واقع ویژگی‌های کلمه‌ی تست هستند را به سیستم نشان داده و محتمل‌ترین دنباله‌ی حالتهای مخفی را توسط الگوریتم ویتربی بدست می‌آوریم.

اگر بخواهیم به طور خلاصه روش کار مدل مخفی مارکوف را بیان کنیم، با توجه به شکل ۲-۳ مراحل را بیان می‌کنیم.



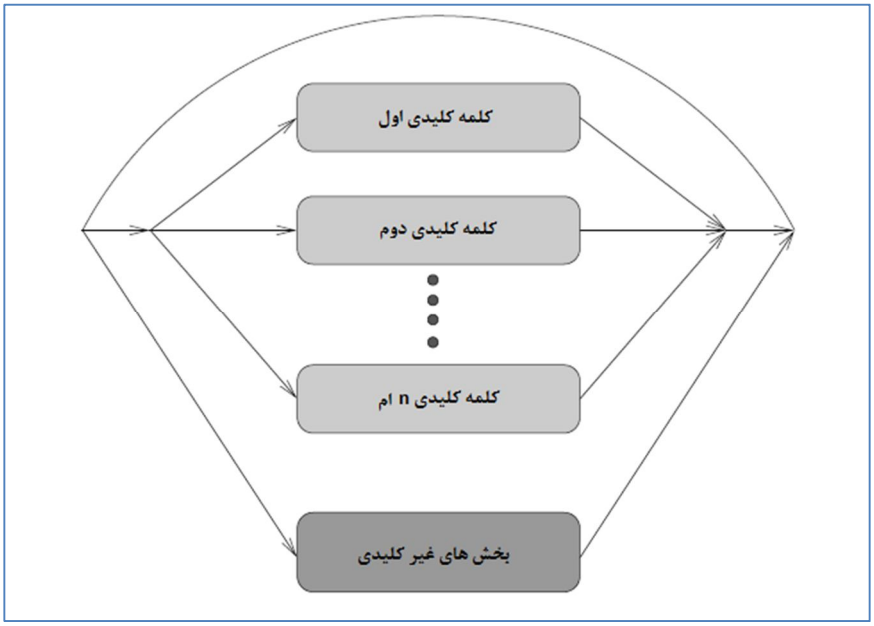
شکل ۲-۳ روند آموزش و بازشناسی در مدل مخفی مارکوف

اساس کار به این صورت است که در ابتدا داده‌های صوتی به همراه داده‌های متنی آنها، به سیستم معرفی شده و برای هر کلمه یا واج یا سیلابس، یک مدل مارکوف شکل می‌گیرد. بعد از اتمام آموزش، حالتهای مخفی سیستم، همان متن کلمات یا آواهای مختلف خواهند بود. زمانی که گفتار تست یا ناشناخته به عنوان بردار مشاهدات وارد سیستم می‌شود، آنگاه از طریق روش کد گشایی ویتربی، محتمل ترین دنباله‌ی مخفی که همان محتمل‌ترین رشته‌ی نوشتاری است، به عنوان خروجی سیستم ظاهر می‌شود.



شکل ۲-۴ یافتن مسیر بهینه برای دنباله‌ای از مشاهدات

مدل مخفی مارکوف علاوه بر این که بیشترین استفاده را در تشخیص گفتار دارد، بیشترین استفاده را در تشخیص کلمات کلیدی نیز دارد. یک شبکه‌ی تشخیص کلمات کلیدی بر مبنای مدل مخفی مارکوف به صورت شکل ۲-۵ می‌باشد:



شکل ۲-۵ شبکه‌ی تشخیص کلمات کلیدی بر مبنای مدل HMM

در این نوع سیستم تشخیص کلمات کلیدی، ابتدا مدل‌های مارکوف مربوط به هر کدام از کلمات کلیدی، توسط فایل صوتی که محتوی هر کدام از کلمات کلیدی است آموزش می‌بینند. در ادامه، مدل کلمات غیر کلیدی نیز توسط بخش‌هایی از گفتار که شامل کلمات کلیدی نیستند، آموزش می‌بینند. در این نوع شبکه‌ها، خطای سیستم به علت وجود کلمات خارج از داده‌های آموزشی مانند اسامی خاص، و در نتیجه احتمال تشخیص اشتباه آنها به عنوان کلمه‌ی کلیدی، بالا می‌رود. شبکه‌های مختلف تشخیص کلمات کلیدی بر مبنای مدل HMM در [۲۱] بررسی شده است.

۲-۱-۳ سایر روشها:

روش HMM به مقدار زیادی از داده‌های آموزشی در کنار آموزش با نظارت احتیاج دارد. منظور از آموزش با نظارت این است که این داده‌ها باید به صورت دستی بخش‌بندی و برچسب گذاری شوند. علاوه بر پیچیدگی و سختی فرایند آموزش و بازشناسی، مدل‌های HMM فقط برای یک زبان خاص کار می‌کنند و برای اعمال روی یک زبان دیگر باید داده‌های آموزشی فراوان از آن زبان موجود باشد[۳]. از روشهایی بر پایه‌ی شبکه‌های عصبی نیز برای تشخیص گفتار استفاده می‌شود. در کنار این روش‌ها، روشهای دیگری نیز برپایه‌ی ماشین بردار پشتیبان[۱] وجود دارد[٤]. خوشه‌بندی تکراری، فضاهای خود تنظیم شونده[۲]، و آموزش کوانتیزاسیون برداری[٥]، روش‌های هیبردی و ترکیب دو یا چند روش مختلف نیز در تشخیص کلمات کلیدی استفاده شده است[۲۶]. تمامی روش‌های فوق با آنکه اندکی بهبود در کارایی را حاصل کرده‌اند، اما هیچکدام موفق نشده‌اند برتری برجسته‌ای نسبت به مدل مخفی مارکوف بدست آورند.

۲-۲ ا نواع روش‌ها در نحوه‌ی پیاده سازی سیستم جستجوی کلمات کلیدی:

روش‌های مختلفی برای پیاده‌سازی موتور جستجوی کلمات کلیدی وجود دارد. این روش‌ها به طور کلی به شاخه‌های زیر تقسیم می‌شوند:

روش‌های بر پایه‌ی دیکشنری گسترده: این روش از یک فرایند دو مرحله‌ای استفاده می‌کند. در اولین مرحله تبدیل گفتار به متن توسط یک موتور تشخیص گفتار پیوسته دیکشنری گسترده (LVCSR) انجام می‌شود. سپس با استفاده از یک الگوریتم جستجوی متنی، کلمه کلیدی در متن پیدا می‌شود.

[۱] - Support Vector Machine (SVM)
[۲] - Self-Organizing Maps(SOM)

روش‌های بر پایه تشخیص واجی: این روش از یک فرایند دو مرحله ای استفاده می‌کند. در مرحله‌ی اول، گفتار به دنباله‌ای از واج‌های تبدیل می‌شود. در مرحله دوم، برنامه در میان دنباله واجی که در قدم اول بدست آمده، دنباله‌ی واجی کلمه کلیدی را جستجو می‌کند.

روش‌های بر پایه تشخیص لغت: در این روش سیستم در یک فرایند یک مرحله‌ای به دنبال کلمه کلیدی می‌گردد. تشخیص برپایه واج است و موتور جستجوی کلمات کلیدی برپایه دنباله واجی که کلمه کلیدی را نمایش می‌دهند، دنبال کلمه کلیدی درون جمله می‌گردد.

مقایسه سه روش مختلف بیان شده:

در این بخش سه روش بیان شده با هم مقایسه می‌شوند و برتری‌ها و ضعف‌های هر کدام بیان می‌شود:

روش فرهنگ لغات گسترده: این روش مناسب برای جستجو روی پایگاه داده‌هایی است که نمی‌توان موتور جستجوی کلمات کلیدی را روی آنها پیاده کرد. در این حالت یک پایگاه داده بزرگ به متن تبدیل شده تا بعدا به صورت نوشتاری یک جستجوی سریع روی آن انجام شود.

برتری‌های اصلی این روش به این ترتیب است که:

۱- ذخیره متن مکالمات ساده و اقتصادی است.

۲- امکان جستجوی دوباره و سریع وجود دارد.

۳- لازم نیست کلمات کلیدی از قبل تعریف شده باشند.

۴- وقتی یکبار عمل تبدیل به متن با دقت انجام شد، جستجو برای کلمه کلیدی جدید، ساده و سرراست است.

ضعف‌های عمده این روش بدین ترتیب هستند:

۱- استفاده از روش LVCSR نمی‌تواند تمام دسته کلماتی که ممکن است به عنوان کلمه کلیدی باشند را پوشش دهد؛ مخصوصا نام‌ها و اصطلاحات خارجی. اضافه کردن کلمات جدید به واژه نامه هم کار پیچیده‌ای است.

۲- LVCSR میزان زیادی از حافظه را اشغال می‌کند و بار زیادی را به CPU تحمیل می‌کند. بنابراین زمانی که ما به زمان پردازش کم احتیاج داریم در استفاده از این روش با محدودیت‌هایی روبه‌رو هستیم.

روش‌های برپایه تشخیص واجی :

در مقابل روش LVCSR، این روش در یک سطح زبانی پایین تر کار می‌کند. یعنی سطح واجی. واج‌ها واحدهای پایه‌ی گفتار هستند که زمانی که ترکیب می‌شوند می‌توانند هر کلمه‌ای را در زبان نمایش دهند. در قدم اول گفتار به متن تبدیل می‌شود. اما این بار متن، دنباله‌ای از واج‌ها است و نه دنباله‌ای از کلمات. در قدم دوم جستجو برای یافتن دنباله‌ای از واج‌ها که کلمه کلیدی ما را تشکیل می‌دهند، انجام می‌شود.

برتری‌های اصلی این روش نیز بدین ترتیب هستند:

۱- جستجوی مرحله دوم سریع تر از جستجوی مستقیم روی رشته‌ی گفتار است.

۲- لازم نیست که لیست کلمات کلیدی از قبل تعریف شده باشند.

ضعف‌های عمده‌ای روش عبارت‌اند از:

۱- اولین مرحله‌ی تشخیص واجی، نتایج نویزی تولید می‌کند که در آنها نرخ خطا کم نیست و می‌توانند روی نتایج مرحله‌ی دوم اثر بگذارند.

۲- جستجوی واجی برای نمایش کلمه‌ی کلیدی به منابع بیشتری نسبت به روش LVCSR نیاز دارد.

۳- این روش دو مرحله‌ای است که قدرت پردازشی بالایی را طلب می‌کند و در پشتیبانی از پردازش real-time، محدودیت دارد.

روش‌های بر پایه تشخیص لغت :

در این روش، یک موتور جستجوی کلمات کلیدی بر پایه واج، پیاده سازی می‌شود. جستجوی کامل برای لیست کلمات کلیدی در یک مرحله انجام می‌شود بدون آنکه پیش پردازشی روش رشته‌ی صوتی درون کلمات یا دنباله‌ی واجی انجام شود. موتور جستجو، لیست تایپ شده‌ی کلمات را که به صورت واجی نمایش داده شده‌اند را به عنوان ورودی دریافت می‌کند و در خروجی خود کلمه کلیدی مورد نظر را آشکار می‌کند.

برتری‌های عمده این روش:

۱- این عملکرد یک مرحله‌ای می‌تواند به صورت real-time انجام شود. البته می‌توانیم به صورت off-line هم این کار را انجام دهیم.

۲- از آنجایی که لیست کلماتی که در جستجوها استفاده می‌شود، محدود است؛ این روش از سایر روش‌ها عملکرد بهتری دارد.

۳- نام‌ها و کلمات غیر معمول نظیر اسامی و اصطلاحات خارجی، به راحتی توسط این روش مدیریت می‌شوند.

ضعف‌های این روش نیز به این صورت است:

۱- برای هر جستجوی مجدد یا تغییر در لیست کلمات کلیدی نیازمند اجرای دوباره‌ی الگوریتم روی رشته‌ی گفتار هستیم.

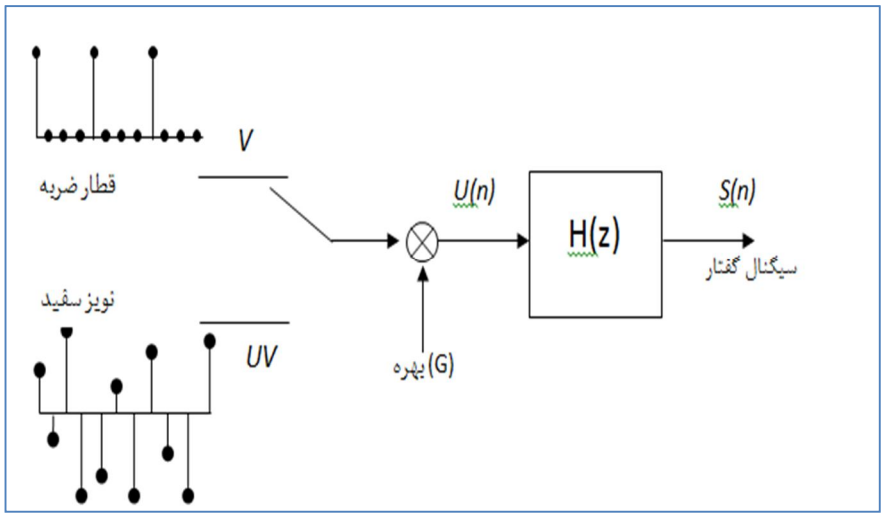
۲-۳ روش‌های رایج در استخراج ویژگی از سیگنال گفتار

۲-۳-۱ استخراج ویژگی:

سیگنال گفتار را در شکل اصلی خود پردازش نمی‌کنند، بلکه برخی از خصوصیات مهم را از سیگنال اصلی استخراج می‌کنند که نه تنها ابعاد سیگنال را کاهش می‌دهد، بلکه باعث افزایش کارایی در پردازش‌های بعدی نیز می‌شود. کارایی ویژگی‌های مختلف استخراج شده از سیگنال گفتار برای چندین دهه مورد تحقیق و بررسی بوده است. برخی آزمایشات نیز با استفاده از ترکیب ویژگی‌ها انجام شده است[۲۵]. برخی از ویژگی‌های مختلف که در پردازش گفتار از آنها استفاده می‌شود در اینجا آورده شده است.

۲-۳-۲ ضرایب Linear Prediction Coding(LPC) :

LPC سعی می‌کند که سیستم تولید گفتار انسان را با استفاده از یک فیلتر تمام قطب مدلسازی کند. بسته به اینکه صدای تولید شده، صوت[۱] و یا غیر صوت[۲] است، ورودی فیلتر می‌تواند یک قطار ضربه‌ی متناوب و یا یک نویز سفید باشد. دوره‌ی تناوب سیگنال گفتار[۳] نیز توسط دوره‌ی تناوب قطار ضربه مشخص می‌شود. دامنه سیگنال گفتار نیز با بهره‌ی کنترلی G مشخص می‌شود.



شکل ۲-۶ تخمین فرایند تولید گفتار توسط LPC

$$H(z) = \frac{۱}{۱ + a_۱ z^{-۱} + a_۲ z^{-۲} + \ldots + a_n z^{-n}}$$

ضرایب فیلتر LPC، vocal tract را مدل می‌کنند. هرچقدر که مرتبه‌ی فیلتر بالاتر باشد، مدل سازی دقیق‌تر خواهد بود. با استفاده از مدل LPC، تمام سیگنال گفتار مشخص شده و در صورتی که ضرایب به خوبی

[۱] - voiced
[۲] - unvoiced
[۳] - pitch

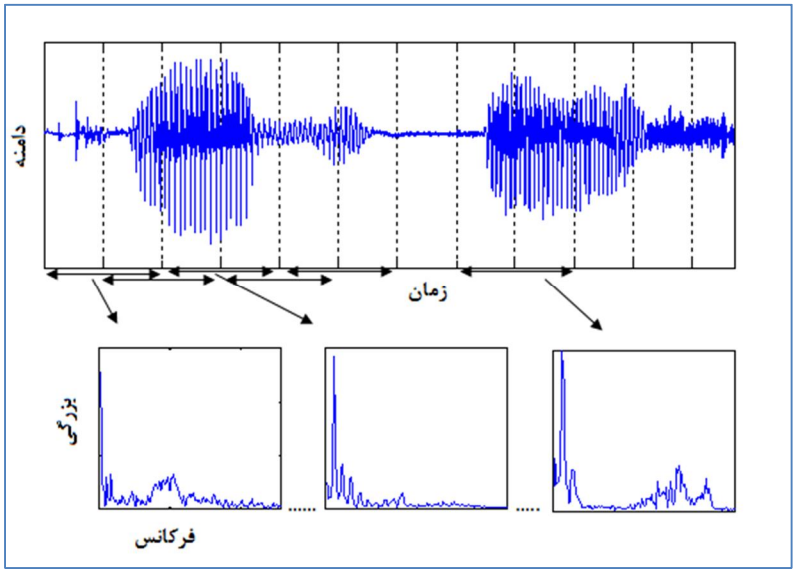
محاسبه شده باشند، می‌توان سیگنال گفتار را مجددا تولید کرد. ضرایب LPC به طور عمده در سنتز گفتار بکار می‌روند؛ اما به دلیل عدم توانایی آنها درمدل سازی نحوه‌ی درک انسان، در بازشناسی گفتار کارآمد نیستند.

۲-۳-۳ ضرایب Mel-Frequency Cepstrum Coefficients (MFCC)

ضرایب MFCC به صورت گسترده‌ای در پردازش گفتار بکار می‌روند. اگر بخواهیم به صورت خلاصه بیان کنیم، ضرایب MFCC طبق مراحل زیر محاسبه می‌شوند.

۱- سیگنال گفتار در ابتدا به فریم‌های کوچکی تقسیم می‌شود.

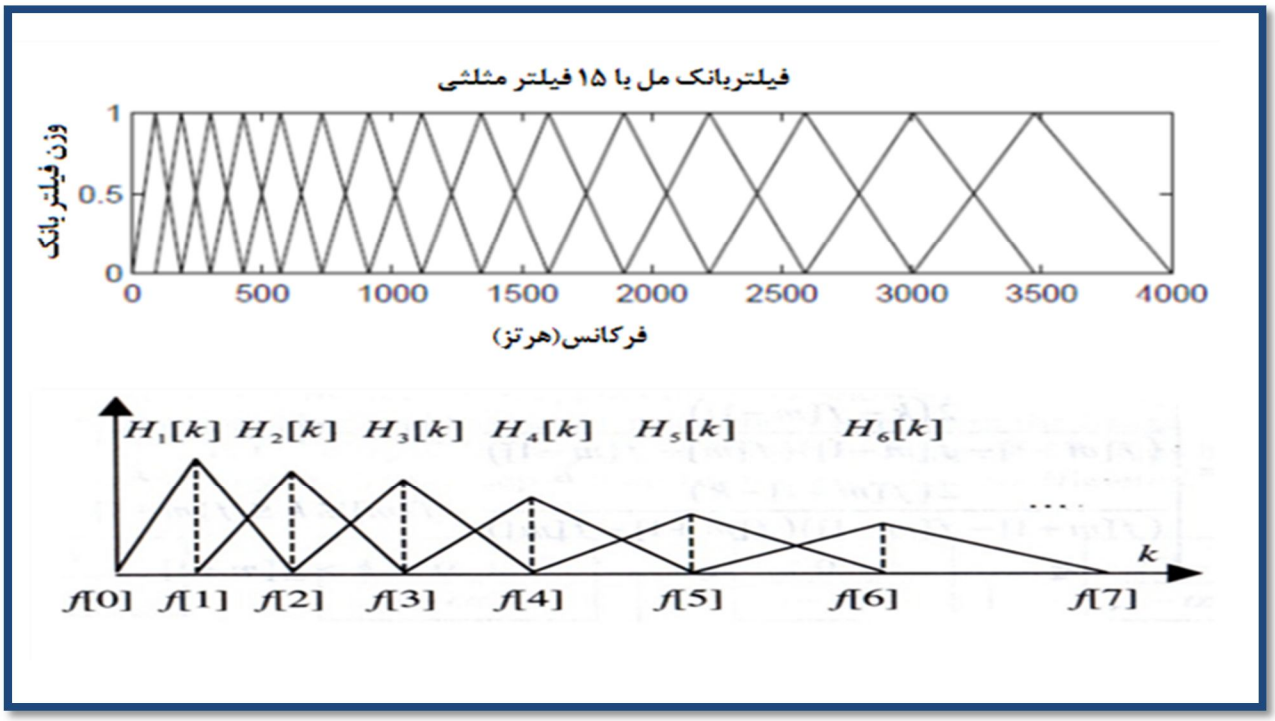
۲- تبدیل فوریه‌ی هر کدام از فریم‌ها بدست می‌آید.



شکل ۲-۷ تبدیل فوریه سیگنال گفتار. تصویر بالا سیگنال گفتار و تصویر پایین تبدیل فوریه برای هر پنجره را نشان می‌دهد.

در شکل ۲-۷، سیگنال گفتار به فریم‌هایی با طول ۲۰ میلی ثانیه شکسته شده که به میزان ۱۰ میلی ثانیه با یکدیگر همپوشانی دارند. در شکل پایین، تبدیل فوریه برای هر پنجره نمایش داده شده است. از این رو یک بردار یک بعدی گفتار به یک بردار دو بعدی کوچکتر از ویژگی‌ها (در اینجا تبدیل فوریه) شکسته شده است.

۳- مقیاس مل[۱] به صورت پنجره‌های مثلثی که با یکدیگر دارای همپوشانی هستند، اعمال می‌شود. در نوع و تعداد فیلتر بانک‌های بکار رفته، تنوع و تغییرات زیادی وجود دارد.



شکل ۲-۸ دو نوع پیاده سازی فیلتربانک مل. در حالت اول، وزن برای تمام فیلترها یکسان است اما در حالت دوم مقدار وزن از ۱ تا ۰.۵ متغیر است

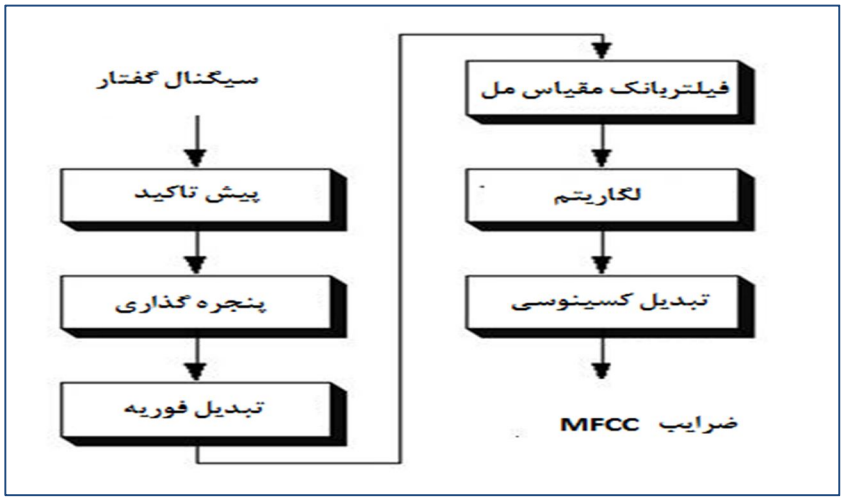
۴- لگاریتم توان را در هر کدام از فرکانس‌های مل بدست می‌آوریم.

[۱] - Mel-Scale

۵- از توان لگاریتمی مل، تبدیل کسینوسی گسسته(DCT) می‌گیریم.

۶- دامنه‌های طیف حاصل، همان ضرایب MFCC هستند.

مراحل محاسبه‌ی ضرایب MFCC در شکل ۲-۹ خلاصه شده است.



شکل ۲-۹ بلوک دیاگرام نحوه‌ی محاسبه‌ی ضرایب MFCC

آزمایش‌هایی برای تعیین تعداد بهینه‌ی فیلترها مورد نیاز است. امروزه از ۱۳ فیلتر مل به همراه ضرایب Δ و Δ^2 که مشتقات اول و دوم استخراج شده از کپستروم هستند، استفاده می‌شود. با احتساب ضرایب Δ در مجموع ۳۹ ضریب MFCC برای هر فریم خواهیم داشت.

۲-۳-۴ سایر ویژگی‌ها :

از ضرایب PLP[۱] و تبدیل موجک می‌توان عنوان سایر ضرایب بکار رفته در پردازش گفتار نام برد. PLP در ابتدا طیف زمان کوتاه گفتار را بهینه کرده و سپس از گفتار حاصل ضرایب LPC را استخراج می‌کند. به طور کلی PLP ترکیبی از پیش‌بینی خطی و تبدیل فوریه‌ی گسسته می‌باشد.

[۱] - Perceptual Linear Predictive Coefficients

فصل سوم

مطالعات پیشین

۳-۱ تحقیقات پیشین در زمینه تشخیص کلمات کلیدی در زبان‌های غیر فارسی

تشخیص کلمات کلیدی دارای یک پیشینه‌ی غنی در علم پردازش گفتار است. اولین آزمایشات در این زمینه روی تشخیص کلمات جدا انجام شد(کلمات با سکوت از هم جدا شده بودند.) [۱۰] . این تحقیقات با تشخیص اعداد متصل بهم و شماره‌گیری صوتی ادامه یافت[۱۱]. این روش‌ها عموما بر پایه‌ی تطبیق نمونه‌ها بودند. تطبیق نمونه‌ها بدان معنی است که به ازای هر کدام از کلمات کلیدی، یک یا چند نمونه از هر کدام را در سیستم ذخیره می‌کنیم و واژه‌ی ناشناس ورودی را با آنها مقایسه می‌کنیم. بعدها، این روش با روشهای آماری نظیر مدل مخفی مارکوف و شبکه‌های عصبی جایگزین شد. برتری این روش‌ها به علت توانایی آنها در یادگیری و استفاده از داده‌های آموزشی فراوان می‌باشد. نقص‌های عمده‌ی روش‌های برپایه تطبیق نمونه‌ها بدین صورت خلاصه می‌شوند:

- هر کلمه‌ی کلیدی به یک و یا چند نمونه نیاز دارد که این باعث محدود شدن اندازه‌ی واژه نامه و به نیاز به وجود حافظه‌ی بزرگتری می‌شود.
- برای تغییرات موقتی در گفتار باید چاره اندیشی شود. این کار نیازمند ایجاد همترازی‌های غیرخطی برای مقایسه‌ی کلمات کلیدی است.
- مرز کلمات باید به دقت مشخص شود. نحوه‌ی مشخص کردن مرز تاثیر زیادی روی کارایی کلی سیستم دارد.

مقاومت بیشتر به تغییرات گوینده و کانال از جمله برتری‌های HMM به سیستم‌های تطبیق نمونه می‌باشد. مهمترین تکامل و بهبودی که در روش‌های برپایه‌ی HMM صورت گرفته است، توسعه مدل‌های واجی پایه است که در آن یک مدل کلمه از چند مدل پایه تشکیل شده است. این مدل‌ها در کلمات مختلف مشترک هستند. این بدان معنی است که مدل یک کلمه کلیدی داده شده نه تنها از گفتارهای آموزشی که شامل این کلمه هستند سود می‌برد، بلکه از تمام گفتارهایی که شامل زیر واحدهایش هستند نیز سود می‌برد. یک برتری

دیگر مدل‌های واجی، توانایی شناسایی آن دسته از کلمات کلیدی است که در گفتار آموزشی وجود ندارند. با استفاده از این برتری می‌توانیم کلمات جدیدی را که از زیر واحدهای آموزش یافته تشکیل شده‌اند، را نیز بسازیم.

در این بخش از کتاب به بررسی برخی از تحقیقات گذشته در زمینه‌ی تشخیص کلمات کلیدی در گفتار فارسی و غیر فارسی خواهیم پرداخت. محققان زیادی در زمینه‌ی تشخیص کلمات کلیدی فعالیت کرده‌اند که در اینجا به صورت مختصر به بررسی تحقیقات برخی از آنها می‌پردازیم. برخی از این محققان از معیارهای متفاوتی برای بیان نتایج خود استفاده کرده‌اند، همچنین هر کدام از تحقیقات انجام شده، روی یک زبان متفاوت پیاده سازی شده است. علاوه بر این، در بحث جستجوی کلمات کلیدی، دقت تشخیص برای کلمات مختلف متفاوت است و بیان یک درصد تشخیص کلی برای هر کدام از روش‌های این محققان امکان پذیر نمی‌باشد.

در [۲۷] محقق به بررسی تفصیلی روش‌های رایج در تشخیص کلمات کلیدی پرداخته و به بیان مزایا و ضعف‌های هر کدام از سیستم‌ها پرداخته است. همچنین راهکارهایی که سایر محققان برای حل این ضعف‌ها بیان کرده‌اند را نیز معرفی است. محقق روش جدیدی را بر مبنای استفاده از سطح بالایی از دانش زبانی برای تشخیص کلمات کلیدی و همچنین یک روش جدید برای بهبود بخشیدن به جستجو در فضای واجی را معرفی کرده است.

در [۲۸] یک سیستم تشخیص گفتار کلیدی برای داده‌های تلفنی طراحی و پیاده سازی شده است. طبق ادعای این محققان، علاوه بر دستیابی به دقت بیشتر، زمان پردازش نیز کاهش یافته است. این سیستم که در دو حالت offline و online کار می‌کند و بر روی دادگان گفتار تلفنی محاوره‌ای در زبان کشور جمهوری چک مورد ارزیابی قرار گرفته است. همچنین در این روش از مدل‌های واجی استفاده شده و عملکرد سیستم برای ضرایب MFCC و PLP نیز مقایسه شده است. این محققان در تحقیق خود از تبدیلی به نام HLDA استفاده کرده‌اند که به گفته‌ی آنها علاوه بر بهبود نتایج، زمان محاسبات را نیز تا میزان ۲۵ درصد کاهش می‌دهد.

محقق دیگری در[۲۹] مخلوطهای گوسی و مدل مخفی مارکوف را برای تشخیص کلمات کلیدی بکار برده است. در این روش از مدل‌های واجی وابسته به متن و مستقل از متن استفاده شده است. محقق در این تحقیق از شبکه‌های متنوع بازشناسی کلمات کلیدی و همچنین از مدل‌های واجی مختلف استفاده و نتایج آنها را با هم مقایسه کرده است. در این روش نیز از ضرایب PLP و MFCC استفاده شده و سیستم بر روی دادگان گفتاری ICSI که یک دادگان گسترده‌ی محاوره‌ای است، مورد ارزیابی قرار گرفته است.

در روش دیگری که در [۳۰] ذکر شده است، نویسنده در ابتدا از طریق نمونه‌های آوایی که توسط کاربر مشخص می‌شوند، کلمات کلیدی را تعریف می‌کند. فرایند تشخیص کلمات کلیدی در این کار در دو فاز جستجو و بازبینی صورت می‌پذیرد. به منظور جلوگیری از افزایش نرخ هشدار اشتباه، نویسنده پیشنهاد کرده است که آن دسته از مدل‌های واجی که میان کلمات کلیدی و کلمات زائد مشترک هستند، از مدل زوائد حذف گردند. بعد از انجام فرایند بازشناسی، گوینده از دو الگوریتم DTW و GMM به منظور تصدیق و بازبینی کلمات تشخیص داده شده، استفاده کرده است. محقق در روش خود موفق شده است که دقتی در حدود ۸۵ درصد را بر روی دادگان تلفنی و با استفاده از ضرایب MFCC، بدست آورد.

در [۴] محققی دیگر روش کاملا جدیدی را برای تشخیص کلمات کلیدی معرفی کرده است. این روش با تمام متدهای قبلی مانند مدل مخفی مارکوف و DTW متفاوت است و بر پایه روش‌های کرنلی می‌باشد. برخلاف روش‌های قبلی، روش معرفی شده از یک فرایند یادگیری جداساز[۱] بهره می‌برد. این سیستم تشخیص کلمات کلیدی، نمایش صوتی جمله‌ی مورد جستجو و کلمات کلیدی ورودی را به فضای برداری می‌نگارد و با استفاده از روش‌های کرنلی، یک دسته بندی را در این فضای برداری انجام می‌دهد و باعث می‌شود که بخش‌های کلیدی و غیر کلیدی گفتار از یکدیگر جدا شوند. محقق آزمایشات خود را بر روی دادگان Timit و HTimit و با استفاده از

[۱] - discriminative

یک الگوریتم تکرار پیاده‌سازی کرده است و طبق ادعای او، نتایج بهتری نسبت به مدل‌های HMM مستقل از متن گرفته است. او نتایج مدل HMM را حدود ۵ درصد بهبود بخشیده است.

محقق دیگری در[۳۱] سیستمی را پیاده سازی کرده است در آن بر خلاف سایر روش‌ها، فرایند تشخیص کلمات کلیدی، بعد از مرحله‌ی بازشناسی آغاز می‌شود. در روش ارائه شده توسط این شخص، ابتدا کل متن به دنباله‌ای از واج‌ها تبدیل می‌شود. شناسایی محل حضور کلمه‌ی کلیدی از روی دنباله‌ی واجی همواره خطاهای زیادی را به دنبال دارد. از جمله‌ی این خطاها را می‌توان به خطای حذف، جایگزینی و یا اضافه شدن یک یا چند واج در محل حضور کلمه‌ی کلیدی نام برد. ولی محقق الگوریتم جستجوی مقاومی را ارائه کرده است که به گفته‌ی او، تا حدود زیادی قادر است اثر این خطاها کم‌رنگ کند.

در[۳۲] نویسنده از مدل مخفی مارکوف برای فرایند تشخیص استفاده کرده است، با این تفاوت که از مدل کلمات غیر کلیدی استفاده نکرده است و از یک معیار امتیاز دهی برای تشخیص این که کلمه، کلیدی و یا غیر کلیدی است استفاده کرده است. نویسنده در بخش نتایج ادعا کرده است که تا ۸۰ درصد در HMMهای وابسته به متن و ۷۰ درصد در HMMهای مستقل از متن، درصد تشخیص صحیح را بدست آورده است.

در [۳۳] محققان در کنار استفاده از مدل مخفی مارکوف، از مدل‌های مخلوط گوسی برای جذب بخش‌های غیر کلیدی گفتار استفاده کرده‌اند. این تحقیق نیز روی دادگان گفتار تلفنی در زبان فرانسوی پیاده سازی شده و موفق شده است که نرخ خطا را تا ۱۰ درصد نسبت به روش HMM کاهش دهد.

در [۳۴] نویسنده به بررسی مفصل انواع روش‌ها در پیاده سازی سیستم تشخیص کلمات کلیدی پرداخته و معایب و برتری‌های هر روش را نیز بیان کرده است.

در[۳۵] نیز نویسنده به بررسی تفصیلی نحوه‌ی پیاده سازی سیستم تشخیص کلمات کلیدی در گفتار پیوسته با استفاده از مدل مخفی مارکوف پرداخته است. آزمایش‌های وی نیز روی دادگان گفتاری SwitchBoard پیاده

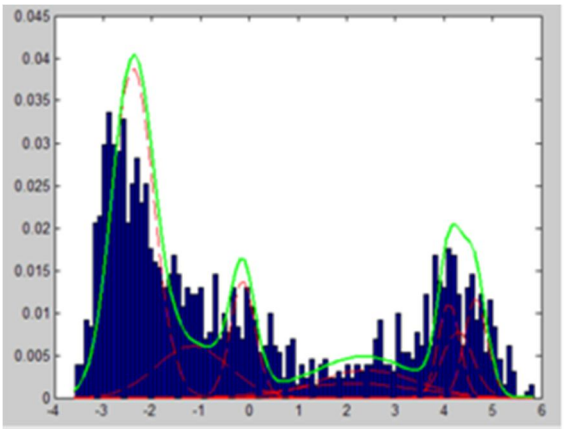
شده و از مدلهای HMM وابسته به متن و همچنین شبکه‌های سه واجی استفاده کرده است. درصد تشخیص حاصل شده در کلمات کلیدی مختلف نیز بررسی شده که بسته به نوع کلمه، از ۶۰ تا ۱۰۰ درصد تشخیص صحیح، متغیر می‌باشد.

۳-۲ تحقیقات پیشین در زمینه تشخیص کلمات کلیدی در زبان فارسی:

با توجه به بررسی‌های انجام شده، منابع محدودی در زمینه‌ی تشخیص کلمات کلیدی در زبان فارسی وجود دارد. یکی از این محدود منابع موجود در [۳۶] آمده است. محققان در این مقاله، یک روش تشخیص کلمات کلیدی را بر مبنای SVM معرفی کرده‌اند. در روش پیشنهادی، صحت هر کلمه‌ی کلیدی شناسایی شده، بر اساس میزان شباهت آن با یک سری کلمات کلیدی و غیر کلیدی نزدیک به آن و با کمک یک دسته بندی کننده‌ی SVM محاسبه می‌گردد. در این روش از مدل مخفی مارکوف برای آموزش مدل‌های بازشناسی استفاده می‌گردد. مدل‌های بازشناسی عبارتند از مدل‌های واجی زبان فارسی و نیز مدل‌های کلمات کلیدی. برای هر کلمه‌ی کلیدی یک SVM مجزا آموزش داده می‌شود. آزمایش‌های انجام شده برای بازشناسی و تشخیص ۱۰۰ کلمه‌ی کلیدی فارسی طبق ادعای محققان به میزان ۹.۴٪ باعث افزایش کارایی سیستم نسبت به استفاده از مدل زوائد شده است. همچنین شرکت عصر گویش نیز موفق به طراحی نرم افزار واژه یاب در گفتار فارسی شده است که متاسفانه در مورد تکنولوژی ساخت آن و روش استفاده شده، اطلاعات بیشتری در دست نمی‌باشد.

۳-۳ مطالعات قبلی تشخیص کلمات کلیدی بر پایه روش S- DTW (استفاده شده در پایان نامه):

ما در تحقیق خود از گونه‌ی بهبود یافته‌ای از روش DTW استفاده کرده‌ایم. در گذشته نیز محققانی از این روش برای تشخیص کلمات کلیدی استفاده کرده‌اند. مهمترین این تحقیقات مربوط به [۲] است که در آن، محقق از مدل مخلوط گوسی و حالت بهبود یافته‌ی الگوریتم DTW، استفاده کرده است. در این روش از مقایسه‌ی بردارهای احتمالاتی گوسی کلمه‌ی کلیدی و جمله، برای یافتن محل حضور کلمه‌ی کلیدی استفاده شده است. در این روش گفتار ابتدا به فریم‌هایی تقسیم شده و ویژگی‌های MFCC برای هر فریم محاسبه می‌شود. یک چگالی مخلوط گوسی کامل به صورت مخلوطی از وزن‌ها، بردارهای میانگین و ماتریس‌های کواریانس می‌باشد. در این روش، مدل‌های مخلوط گوسی برای تمام فریم‌های داده‌های آموزشی تخمین زده می‌شوند.



شکل ۳-۱ تخمین بردارهای ویژگی با استفاده از مجموع وزن یافته‌ی گوسی‌ها. منحنی پررنگ بهترین تخمین را نشان می‌دهد

اگر مدل‌های GMM و بردار ویژگی‌های تست داده شده باشد، آنگاه میزان احتمالی که یک دسته از ویژگی‌ها متعلق به یک GMM خاص باشند، قابل محاسبه است. اگر یک گفتار با تعداد n فریم را به صورت $S=(s_1,s_2,\ldots,s_n)$ نشان دهیم، آنگاه بردار احتمال گوسی (GP) به این صورت تعریف می‌شود:

$$GP(s)=(q_1,q_2,\ldots,q_n) \quad (۱-۳)$$

که در آن هر q_i می‌تواند توسط این رابطه‌ی زیر محاسبه شود:

$$qi = (P(C1|Si), P(C2|si), \ldots , P(Cm|si)) \quad (۲-۳)$$

که در آن C_i نشان دهنده‌ی i امین جزء گوسی از یک GMM می‌باشد و M تعداد مخلوط‌های گوسی را نشان می‌دهد. همچنین اختلاف میان دو بردار احتمال گوسی به این صورت تعریف می‌شود:

$$D(p,q)=-\log(p.q) \quad (۳-۳)$$

اگر بخواهیم به صورت خلاصه بیان کنیم، برای جمله‌ی اصلی و کلمه‌ی کلیدی یک دسته از بردارهای احتمال به صورت $GPi = (p1; p2; \ldots; pm)$ و $GPj = (q1; q2;\ldots ; qn)$ موجود خواهد بود و با استفاده از الگوریتم بهینه شده‌ی DTW، و مقایسه‌ی بردار احتمالاتی جمله اصلی و کلمه‌ی کلیدی، قادر خواهیم بود محل حضور کلمه‌ی کلیدی را تعیین کنیم. در مورد الگوریتم بهینه شده‌ی DTW و نحوه‌ی انجام این کار در ادامه صحبت خواهیم کرد.

فصل چهارم

روش پیشنهادی الف

جستجوی کلمات کلیدی در گفتار گسسته

۴-۱ نگاه کلی:

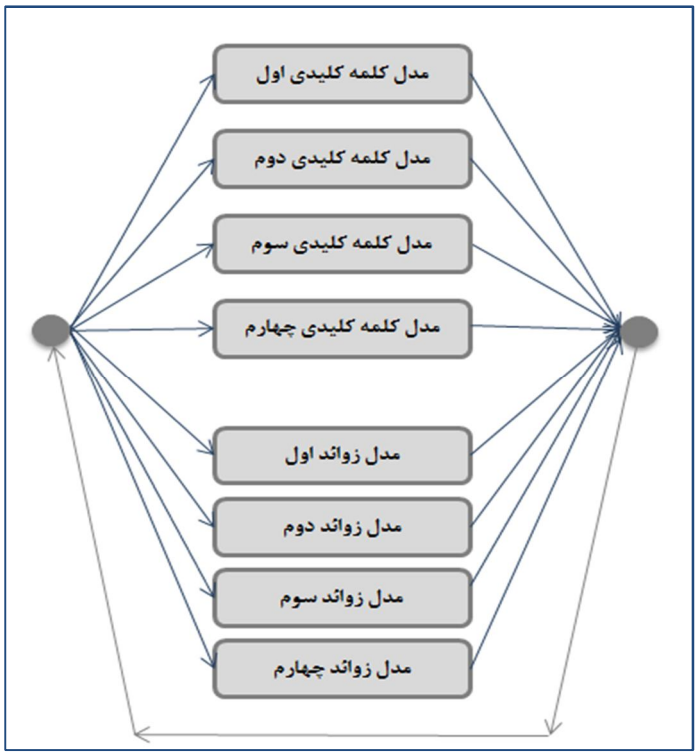
جستجوی کلمات کلیدی در گفتار گسسته یکی از کاربردی‌ترین مباحث زمینه پردازش گفتار است. بسیاری از مباحثی که در زمینه‌ی پردازش گفتار مطرح است، مانند استخراج ویژگی و حذف نویز، در واقع مقدمه‌ای برای مبحث تشخیص کلمات کلیدی در گفتار گسسته است. اهمیت این مبحث به دلیل کاربردهای فراوان آن در سیستم‌های مختلف می‌باشد؛ که از جمله‌ی آنها می‌توان به اجرای فرامین صوتی و شماره گیرهای صوتی اشاره کرد. تشخیص کلمات کلیدی در گفتار گسسته با عنوان تشخیص کلمات جدا[۱] نیز شناخته می‌شود، که البته تفاوت این دو موضوع به نحوه‌ی کاربرد آنها بستگی دارد. در تشخیص کلمات گسسته که حالت کلی‌تر تشخیص کلمات کلیدی گسسته است، ما لیستی از کلمات داریم که در سیستم تعریف می‌شوند. بعد از تعریف این کلمات، در زمان تشخیص کلمه‌ای ناشناس با تمام کلمات موجود در لیست مقایسه می‌شود و کلمه‌ای که بیشترین شباهت را با کلمه‌ی ورودی دارد به عنوان کاندیدای تشخیص معرفی می‌شود. در این نوع سیستم ما این اطمینان را داریم که واژه‌ی ناشناسی که به سیستم وارد می‌شود، حتما در میان لیست کلمات ذخیره شده در سیستم وجود دارد و شبیه ترین واژه به واژه‌ی ناشناس به عنوان کاندید تشخیص معرفی می‌شود. در این زمان اگر کلمه‌ای به سیستم وارد شود که در میان لیستی که از قبل تعریف کرده‌ایم، نباشد، آنگاه این مساله باعث بروز خطا می‌شود. علت خطا این است که اساس کار این سیستم‌ها بگونه‌ای است که حتما یکی از کلمات موجود در لیست باید کاندیدای تشخیص معرفی شود و زمانی که کلمه‌ای خارج از لیست وارد سیستم شود، با وجود اینکه تمام کلمات لیست فاصله‌ی زیادی با کلمه‌ی ورودی دارند، باز هم یکی از کلمات خواهد بود که فاصله ای کمتر از سایرین داشته باشد و این مساله باعث به وجود آمدن خطا و تشخیص اشتباه می‌شود. برای حل این مشکل، مساله‌ی تشخیص کلمات گسسته را در قالب تشخیص کلمات کلیدی گسسته بیان می‌کنیم. تفاوت این دو

[۱] - Isolated Word Recognition

روش در این است که در تشخیص کلمات کلیدی، علاوه برای ایجاد لیستی از کلماتی که قصد تشخیص آنها را داریم، لیستی دیگر را نیز ایجاد می‌کنیم که محتوی کلماتی است که هدف تشخیص نیستند اما احتمال اینکه بیان شوند زیاد است. هدف در اینجا این است که این کلمات زائد به گونه‌ای حذف شوند تا باعث ایجاد خطا در سیستم نشوند. بنابراین ما کلمات را به دو بخش کلیدی و غیر کلیدی تقسیم می‌کنیم. کلمات غیر کلیدی را ((زائد[1])) می‌نامیم. برای تفهیم بهتر موضوع فرض کنید که فردی با سیستم تلفن بانک تماس گرفته و قصد دارد قبوض خود را پرداخت کند، اما در ابتدا بیان می‌کند که ((سلام...مجیدی هستم....)). آنگاه ممکن است سیستم تلفن بانک کلمه‌ی مجیدی را به عنوان کلمه‌ی موجودی تفسیر کند و درگیر عملیاتی ناخواسته شود. برای حل این مشکل، ما لیستی از کلماتی را هم به سیستم آموزش می‌دهیم که از نظر آوایی با کلمه‌ی موجودی تشابه دارند و از سیستم می‌خواهیم که در صورت شناسایی این کلمات به آنها واکنشی نشان ندهد. به عنوان مثال واژه‌هایی نظیر((مجید، مجیدی، مجد ...)) با کلمه‌ی موجودی تشابه آوایی دارند. نمونه‌های تشابه دیگر از کلمات ((مژده و مژگان – سعید و حمید – آزادی و آبادی – آتش و آرش و غیره)) اشاره کرد. در این جفت کلمات مشابه و یا خانواده کلمات مشابه بزرگتر، می‌توان بعد از تعریف یک کلمه‌ی کلیدی، سایر کلمات مشابه را به عنوان مدل‌های غیر کلیدی به سیستم معرفی کرد که در صورت بیان هرکدام از آنها توسط گوینده، برنامه آن کلمه را به اشتباه به عنوان کلمه‌ی کلیدی تشخیص ندهد. البته کلمات زائد فقط محدود به کلمات مشابه با کلمه کلیدی نمی‌شوند و می‌توان کلماتی نظیر ((سلام، خدانگهدار، بسمه تعالی و ...)) و سایر واژه‌هایی که معمولا برخی گویندگان بیان می‌کنند را نیز به عنوان مدل غیر کلیدی معرفی کرد. در شکل ۴-۱ نحوه‌ی مدل سازی کلمات کلیدی و غیر کلیدی به صورت تصویری بیان شده است.

[1] - garbage

پیچیدگی الگوریتم تشخیص کلمات کلیدی در گفتار گسسته، بسیار کمتر از حالت پیوسته است. در اینجا فقط کافی است که داده‌های آموزشی کافی شامل نمونه‌هایی از کلمات کلیدی و کلمات زائد، موجود باشد. با موجود بودن نمونه‌های کافی، سیستم را می‌توان توسط یک الگوریتم مانند مدل مخفی مارکوف یا شبکه‌های عصبی آموزش داد. پس از آموزش سیستم، شبکه ای مانند آنچه در شکل ۴-۱ دیده می‌شود، ایجاد می‌شود. آنگاه هر کدام از واژه‌های ناشناس ورودی با هرکدام از مدل‌های شبکه مقایسه می‌شوند تا مشخص شود واژه‌ی ناشناس، یک کلمه‌ی کلیدی خواهد بود یا خیر. در ادامه با نحوه‌ی آموزش این شبکه و روش مورد استفاده آشنا خواهیم شد.



شکل ۴-۱ شبکه‌ی کلمات کلیدی و زوائد. هر کدام از بلوک‌های مستطیلی، مدل ایجاد شده برای یک کلمه‌ی کلیدی و یا یک کلمه‌ی زائد را نشان می‌دهند.

زمانی که یک کلمه‌ی ناشناس وارد سیستم شود، با هر کدام از این مدل‌ها مقایسه شده و یک امتیاز را در خروجی قرار می‌دهد. هر مدلی که امتیاز بیشتری را کسب کند، برنده‌ی فرایند تشخیص می‌شود. مدل کلمات

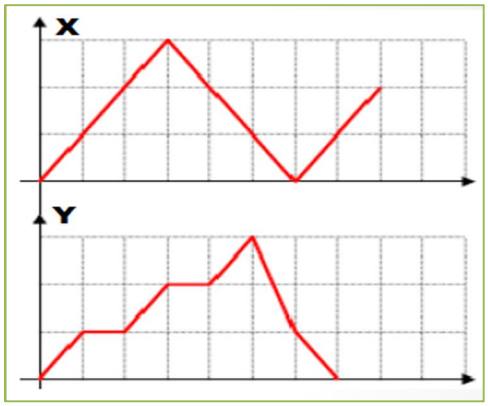
زائد به منظور حذف کلمات غیر کلیدی در شبکه قرار داده می‌شوند. کلمات غیر کلیدی کلماتی هستند که هدف فرایند تشخیص نیستد ولی احتمال اینکه بیان شوند، وجود دارد.

۴-۲ الگوریتم Dynamic Time Warping

در فصل دوم الگوریتم Dynamic Time Warping(DTW) به صورت کلی و مختصر شرح داده شد. از الگوریتم DTW به منظور مقایسه‌ی دو سیگنال با دارای طول نابرابر و اندازه‌گیری میزان اختلاف بین آنها استفاده می‌شود. در پردازش گفتار، از این روش برای مقایسه‌ی دو سیگنال صوتی با طول نابرابر استفاده می‌شود. در این بخش با نحوه‌ی بکارگیری عملی این روش آشنا خواهیم شد. به منظور فهم ساده‌تر این الگوریتم، از یک مثال عددی ساده استفاده می‌کنیم. فرض کنید سیگنال‌های X و Y، دو تابع زمانی با طول نابرابر و به صورت زیر باشند:

X[t]={۰ , ۱ , ۲ , ۳, ۲ ,۱ , ۰ , ۱ , ۲}

Y[t]={۰ , ۱ , ۱ , ۲, ۲ ,۳ , ۱ , ۰ }



شکل ۴-۲ نمایش دو تابع زمانی با طول نابرابر

همانطور که در شکل ۴-۲ مشخص شده است، دو تابع X و Y داراط طول نابرابر هستند و نمی‌توان از طریق محاسبه فاصله اقلیدسی، اختلاف آنها را بدست آورد. برای استفاده از معیار فاصله DTW، ابتدا باید یک ماتریس

فاصله‌ی محلی را تشکیل دهیم. ماتریس فاصله‌ی محلی، یک ماتریس دو بعدی است که در آن فاصله اقلیدسی تمام مقادیر دو تابع ورودی به صورت دو به دو محاسبه می‌شود. مطابق جدول ۴-۱، ابتدا دو تابع ورودی را در دو سمت بالا و چپ ماتریس قرار می‌دهیم.

جدول ۴-۱ ماتریس فاصله‌ی محلی میان دو تابع X و Y

	۰	۱	۲	۳	۲	۱	۰	۱	۲
۰	۰	۱	۲	۳	۲	۱	۰	۱	۲
۱	۱	۰	۱	۲	۱	۰	۱	۰	۱
۱	۱	۰	۱	۲	۱	۰	۱	۰	۱
۲	۲	۱	۰	۱	۰	۱	۲	۱	۰
۲	۲	۱	۰	۱	۰	۱	۲	۱	۰
۳	۳	۲	۱	۰	۱	۲	۳	۲	۱
۱	۱	۰	۱	۲	۱	۰	۱	۰	۱
۰	۰	۱	۲	۳	۲	۱	۰	۱	۲

بالایی‌ترین ردیف جدول محتوی اطلاعات سیگنال X و اولین ردیف از سمت چپ، محتوی اطلاعات سیگنال Y است. سایر خانه‌های ماتریس را با محاسبه‌ی فاصله عناصر متناظر آن در دو تابع پر می‌کنیم. به عنوان مثال المان (۴و۵) جدول، محتوی اختلاف چهارمین مقدار تابع X و پنجمین مقدار تابع Y است. سایر المان‌های ماتریس نیز به همین ترتیب محاسبه خواهند شد. بعد از محاسبه‌ی ماتریس فاصله‌ی محلی، نوبت به محاسبه‌ی

ماتریس فاصله‌ی سراسری می‌رسد. از ماتریس فاصله‌ی سراسری به منظور بدست آوردن مسیر همترازی و اختلاف دو سیگنال استفاده می‌شود. از رابطه‌ی ۴-۱ به منظور محاسبه ماتریس فاصله سراسری استفاده می‌شود:

$$Global\_dist(i,j) = Local\_dist(i , j) + \min \{ Global\_dist(i-1 , j-1), Global\_dist(i-1 , j), Global\_dist(i,j-1)\} \quad (۱-۴)$$

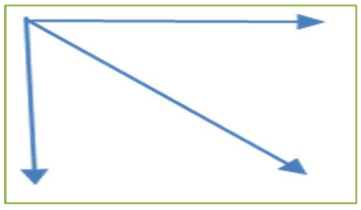
با استفاده از رابطه ۴-۱ می توان ماتریس فاصله‌ی سراسری را بدین صورت محاسبه کرد.

جدول ۴-۲ ماتریس سراسری فاصله‌ی میان دو تابع X و Y

	۰	۱	۲	۳	۲	۱	۰	۱	۲
۰	۰	۱	۳	۶	۸	۹	۹	۱۰	۱۲
۱	۱	۰	۱	۳	۴	۴	۵	۵	۶
۱	۲	۰	۱	۳	۴	۴	۵	۵	۶
۲	۴	۱	۰	۱	۱	۲	۴	۵	۵
۲	۶	۲	۰	۱	۱	۲	۴	۵	۵
۳	۹	۴	۱	۰	۱	۳	۵	۶	۶
۱	۱۰	۴	۲	۲	۱	۱	۲	۲	۳
۰	۱۰	۵	۴	۵	۳	۲	۱	۲	۴

ماتریس فاصله‌ی محلی را با d ماتریس فاصله‌ی سراسری را با D نشان می‌دهیم. در رابطه‌ی ۴-۱، ماتریس فاصله‌ی محلی با عنوانی Local_dist و ماتریس فاصله‌ی سراسری با عنوان Global_dist نمایش داده شده‌اند. به منظور بکار بردن رابطه‌ی ۴-۱، نیازمند این هستیم تا شرایط اولیه را بدین صورت تعریف کنیم:D(۱,۱)=d(۱,۱)

با استفاده از رابطه‌ی ۴-۱ و اعمال شرط مرزی می‌توان سایر عناصر ماتریس فاصله‌ی سراسری را محاسبه کرد. آخرین خانه از ماتریس فاصله‌ی سراسری (پایین ترین ردیف، سمت راست) بیانگر مقدار نهایی فاصله‌ی بین دو سیگنال خواهد بود. علاوه بر محاسبه‌ی اختلاف میان دو سیگنال، نیازمند بدست آوردن مسیر همترازی بین دو سیگنال هستیم. مسیر همترازی مشخص می‌کند که کدام نقاط از تابع X همتراز با کدام نقطه از تابع Y خواهد بود. در فصل دوم در مورد محدودیت‌های حاکم بر مسیر همترازی به صورت مفصل بحث شد. مسیر همترازی بسته به نوع چینش عناصر در ماتریس فاصله‌ی سراسری، باید اکیدا صعودی و یا اکیدا نزولی باشد. انتخاب مسیر همترازی با استفاده از اعمال برخی قیود بر روی ماتریس فاصله صورت می‌گیرد. برای آشنایی با این قیود، به [۱۷] مراجعه شود. یک قید ساده برای انتخاب مسیر همترازی به این صورت نمایش داده می‌شود:



شکل ۴-۳ نحوه حرکت در ماتریس فاصله برای بدست آوردن مسیر همترازی

این بدان معنی است که اگر خانه‌ی اول ماتریس فاصله‌ی سراسری، نقطه‌ی شروع مسیر همترازی باشد، خانه‌ی بعدی این مسیر، یکی از سه خانه‌ی پایین، سمت راست و یا رو به روی خانه‌ی فعلی خواهد بود. هر کدام از این سه خانه که مقدار کمتری داشته باشد، نقطه‌ی بعدی ما در مسیر همترازی خواهد بود. برای فهم بیشتر این مساله، مسیر همترازی را برای مثالی که بحث شد، بدست می‌آوریم.

جدول ۴-۳

نمایش مسیر همترازی میان دو تابع X و Y

	۰	۱	۲	۳	۲	۱	۰	۱	۲
۰	۰	۱	۳	۶	۸	۹	۹	۱۰	۱۲
۱	۱	۰	۱	۳	۴	۴	۵	۵	۶
۱	۲	۰	۱	۳	۴	۴	۵	۵	۶
۲	۴	۱	۰	۱	۱	۲	۴	۵	۵
۲	۶	۲	۰	۱	۱	۲	۴	۵	۵
۳	۹	۴	۱	۰	۱	۳	۵	۶	۶
۱	۱۰	۴	۲	۲	۱	۱	۲	۲	۳
۰	۱۰	۵	۴	۵	۳	۲	۱	۲	۴

مسیر همترازی که به صورت پر رنگ تر در جدول دیده می‌شود، از اولین خانه ی ماتریس فاصله ی سراسری آغاز شده و در آخرین خانه نیز خاتمه می‌یابد. برای بررسی تفصیلی الگوریتم DTW به [۱۷] مراجعه شود.

۴-۳ آموزش سیستم:

۴-۳-۱ ایجاد لیست کلمات کلیدی و کلمات زائد

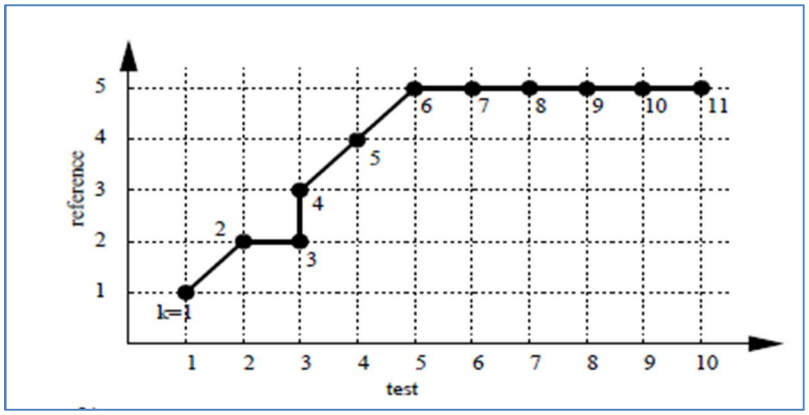
در این پایان نامه برای انجام آزمایشات خود از روش DTW استفاده کرده‌ایم. نحوه‌ی انجام این کار به این صورت است که فاصله‌ی DTW کلمه‌ی ورودی ناشناس با هرکدام از کلمات کلیدی و کلمات زائد که از قبل در شبکه ذخیره کرده ایم محاسبه می‌شود و آن کلمه‌ای که کمترین فاصله را با کلمه‌ی ورودی ناشناس دارد، به عنوان کاندیدای تشخیص معرفی می‌شود. برای اینکه بتوانیم از روش DTW استفاده کنیم، باید ابتدا لیستی از کلمات کلیدی را تهیه کنیم. فرض کنید در یک سیستم کنترل صوتی تلویزیون، کلمات ((روشن، خاموش، کم(صدا) و زیاد)) به عنوان ۴ کلمه‌ی کلیدی مورد نظر باشند. تعریف کلمات زائد بستگی به نوع کاربرد ما دارد. اگر اطمینان داریم که کاربر جز ۴ کلمه‌ی مزبور کلمه‌ی دیگری را در هنگام استفاده به زبان نمی‌آورد، نیازی به تعریف مدل زوائد نخواهد بود. زمانی که نیازی به استفاده از مدل زوائد نباشد، یکی از کلمات کلیدی به عنوان کاندیدای تشخیص کلمه‌ی ناشناس ورودی معرفی می‌شود.

۴-۳-۲ آموزش:

برای تهیه‌ی لیست کلمات کلیدی، باید حداقل یک نمونه از هر کلمه‌ی کلیدی موجود باشد. این نمونه، یک فایل صوتی است که در آن یک گوینده کلمه‌ی کلیدی را بیان می‌کند. یک فایل صوتی در واقع یک ماتریس یک بعدی است که محتوی اطلاعات سیگنال گفتار است. از آنجا که هر سیگنال گفتار دارای حجم زیادی از اطلاعات است، نیازمند این هستیم که اطلاعات مهم و ارزشمند را از سیگنال گفتار استخراج کنیم. آزمایشات محققان در طول سالیان نشان داده است که در پردازش

گفتار، ضرایب MFCC نسبت به سایر روش‌ها از کارایی بیشتری برخوردار هستند. ما نیز از ضرایب MFCC به عنوان ویژگی‌های استخراجی از صوت استفاده کرده‌ایم. در صورتی که بیش از یک نمونه از هر کلمه‌ی کلیدی موجود باشد، می‌توان توسط الگوریتم DTW، تمام این نمونه‌ها را به یک نمونه تبدیل کرد. این نمونه که نمونه‌ی عام نامیده می‌شود، محتوی یک میانگین کلی از تمام فایل‌های موجود از یک کلمه‌ی کلیدی خاص است. اگر گوینده‌های مختلف با لهجه‌ها و سنین متفاوت یک کلمه‌ی کلیدی را بیان کنند، آنگاه تمام این کلمات از نظر طولی با یکدیگر تفاوت خواهند داشت. از آنجا که هر فایل صوتی به صورت یک بردار است، در صورتی که طول بردارها با هم یکسان نباشد، امکان گرفتن میانگین بین آنها در حالت عادی وجود ندارد. برای حل این مشکل از روش تغییر طول بردارها بر اساس الگوریتم DTW استفاده می‌کنیم.

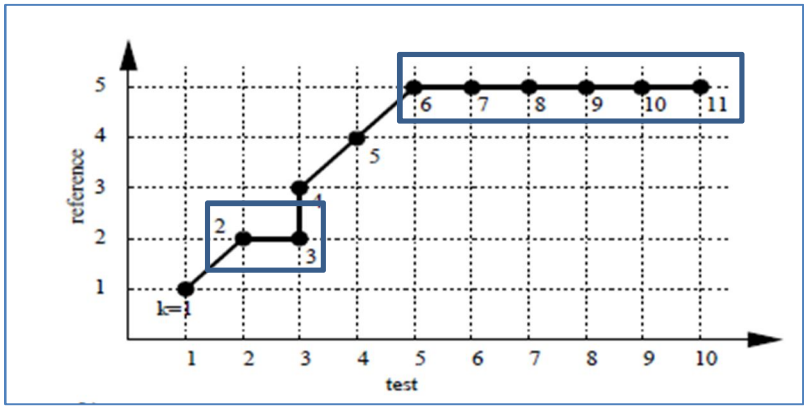
فاصله‌ی میان دو بردار توسط مسیر همترازی مشخص می‌شود. یک نمونه از مسیر همترازی در شکل ۴-۲ برای بردار مرجع و بردار آزمون دیده می‌شود.



شکل ۴-۴ مسیر همترازی برای بردار مرجع و بردار آزمون

مسیر همترازی مشخص می‌کند که کدام نقطه از بردار مرجع، همتراز با کدام نقطه یا نقاط از بردار آزمون است. در صورتی که تعداد چندین نمونه از هرکلمه‌ی کلیدی موجود باشد، از مسیر همترازی

می‌توان برای میانگین گیری و تولید یک کلمه‌ی عام استفاده کرد. به منظور تولید کلمه‌ی عام که خواص تمام کلمات را در خود دارد، باید کلمه‌ای که کوچکترین طول نسبت به سایرین دارد را انتخاب کنیم. این کلمه همان کلمه‌ی مرجع است. پس از انتخاب کلمه‌ی مرجع، هرکدام از کلمات دیگر به نوبت با کلمه‌ی مرجع مقایسه شده و مسیر همترازی آنها مشخص می‌شود. بعد از مشخص شدن مسیر همترازی، بخش‌هایی از مسیر که به صورت افقی است را انتخاب کرده و از بردارهای متناظر در آن نقاط میانگین می‌گیریم. این کار باعث می‌شود که دو بردار مرجع و آزمون که طول نابرابر داشته‌اند، با یکدیگر هم اندازه شوند. شکل ۴-۵ نحوه‌ی انجام این کار را نشان می‌دهد.



شکل ۴-۵ یکسان سازی ابعاد بردار آزمون و بردار مرجع. بخش‌های افقی در طول مسیر همترازی را پیدا می‌کنیم و از مقادیر متناظر آنها میانگین می‌گیریم.

همانطور که در شکل۴-۵ نشان داده شده است، نقاط ۲ و ۳ از بردار آزمون (test) با نقطه‌ی ۲ از بردار مرجع(reference) هم‌تراز است. همچنین نقاط ۵، ۶، ۷، ۸، ۹ و ۱۰ از بردار آزمون، هم‌تراز با نقطه‌ی ۵ از بردار مرجع است. برای یکسان سازی طول بردارهای آزمون و مرجع، در صورتی که همانند شکل بالا چندین نقطه از بردار آزمون هم‌تراز با یک نقطه از بردار مرجع باشد، از تمامی آن نقاط میانگین گرفته و مقدار میانگین را به جای کل آن نقاط قرار می‌دهیم. در شکل ۴-۵ میانگین نقاط ۲ و ۳ از بردار آزمون را هم‌تراز با نقطه‌ی ۲ از بردار مرجع قرار می‌دهیم. همین عمل را در مورد

نقاط ۵، ۶، ۷، ۸، ۹،۱۰ نیز انجام می‌دهیم و میانگین این نقاط را همتراز با نقطه‌ی ۵ از بردار مرجع قرار می‌دهیم. بعد از اتمام این کار طول بردار مرجع و آزمون با هم برابر می‌شود. به عنوان مثال ۱۰ نمونه از فایل صوتی کلمه‌ی کلیدی کتابخانه وجود دارد، بعد از استخراج ویژگی از تمام ۱۰ نمونه، آن نمونه‌ای که طول کمتری از سایرین دارد به عنوان مرجع انتخاب شده، و طول سایر ۹ نمونه‌ی دیگر توسط روشی که گفته شد، با طول نمونه‌ی مرجع برابر می‌شود. بعد از اتمام این کار، ۱۰ ماتریس ویژگی از کلمه‌ی کلیدی کتابخانه خواهیم داشت که طول همه‌ی آنها برابر است. وقتی طول این ماتریس‌های ویژگی برابر باشد، به سادگی یک میانگین کلی از آن ۱۰ نمونه بدست آورده و به عنوان مدل کلمه‌ی کلیدی کتابخانه، در شبکه قرار می‌دهیم.

۴-۴ آماده سازی برای انجام آزمایشات:

۴-۴-۱ معرفی دادگان گفتاری:

به منظور انجام آزمایشات، از دادگان گفتاری فارسی ((فارس دات)) استفاده شده است. دادگان فارس دات، در لابراتوار پردازش گفتار دانشگاه تهران تهیه شده است و شامل جملات ضبط شده ای از گویندگان ایرانی با لهجه‌ها و سنین متفاوت می‌باشد. در اولین آزمایش، حدود ۱۰ کلمه‌ی کلیدی که هرکدام توسط ۲۰ گوینده‌ی مختلف بیان شده‌اند را از این جملات استخراج کردیم. به علت محدودیت در دسترسی به دادگان گفتاری استاندارد، از مدل زوائد استفاده نشده است. استفاده از مدل زوائد، کاملا اختیاری است و صرفا برای کاهش خطای سیستم بکار می‌رود. بعد از مشخص شدن نوع کاربرد، می‌توان در کنار کلمات کلیدی، از کلمات زائد نیز استفاده کرد. برای انجام این کار باید ابتدا بررسی کنیم که گوینده‌ها ممکن است چه کلماتی را غیر از هر کلمه‌ی کلیدی بر زبان بیاورند. این کلمات، کلمات زائد هستند و در صورتی که مدل زوائد نداشته باشیم، نهایتا یکی از کلمات

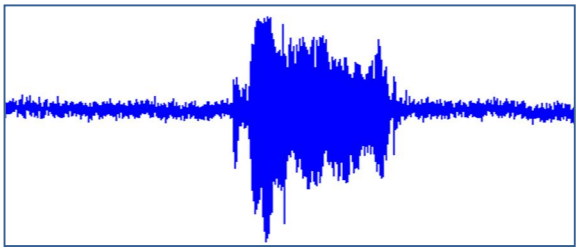
کلیدی شبکه به عنوان کاندیدای تشخیص معرفی می‌شود و این مساله باعث ایجاد خطا و کاهش دقت سیستم می‌گردد.

کلمات ژاپن، شاهرخ، راهپیمایی، ماشین، کتاب، اسلامی، ایران، جنگ، خاموش و خسرو، ۱۰ کلمه‌ی کلیدی هستند که در اولین آزمایش استفاده شده‌اند. این کلمات به این علت انتخاب شده‌اند که دارای تکرار بیشتری در دادگان فارس دات هستند. اکثر کلماتی که در دادگان فارس دات وجود دارند، دارای تکرارهایی کمتر از ۱۵ عدد می‌باشند. وقتی تعداد نمونه‌های بیان شده از یک کلمه کم باشد، مدل آن کلمه با خوبی آموزش نمی‌بینند. هر یک از ۱۰ کلمه‌ی انتخابی دارای ۲۰ تکرار است که توسط ۲۰ گوینده‌ی مختلف بیان شده است. گوینده‌های یک کلمه‌ی نیز کلیدی لزوما با گوینده‌های کلمه‌ی کلیدی دیگر یکسان نمی‌باشند. یکسان نبودن گوینده‌ها و کمبود داده‌های آموزشی برای گفتار گسسته، باعث افزایش خطای سیستم می‌شود. مشکل دیگری که برای داده‌های آموزشی ما وجود دارد این است که تمامی این کلمات گسسته، از گفتار پیوسته برش داده شده‌اند. تلفظ اکثر کلمات در حالت پیوسته با حالت گسسته متفاوت است و تحت تاثیر کلمات قبل و بعد از خود قرار می‌گیرد. به منظور کاهش خطای مزبور، از کلماتی استفاده شده است که نحوه‌ی تلفظ آنها تاثیر پذیری کمتری از کلمات قبل و بعد خود دارند. علاوه بر استفاده از کلمات استخراج شده از دادگان فارس دات، از یک دادگان دیگر که شامل اعداد ۰ تا ۹ به زبان انگلیسی است نیز استفاده شده است.

۴-۴-۲ پیش پرداز کلمات کلیدی:

زمانی که گوینده یک کلمه‌ی را بیان می‌کند، معمولا چند لحظه سکوت یا مکث قبل و بعد از بیان آن کلمه به وجود می‌آید. این مقدار مکث باعث به وجود آمدن اطلاعات حشو و اضافه می‌گردد و در نتیجه سبب می‌شود که مدلی که از یک کلمه‌ی کلیدی ایجاد شده، بهینه نباشد و تحت تاثیر نویز و

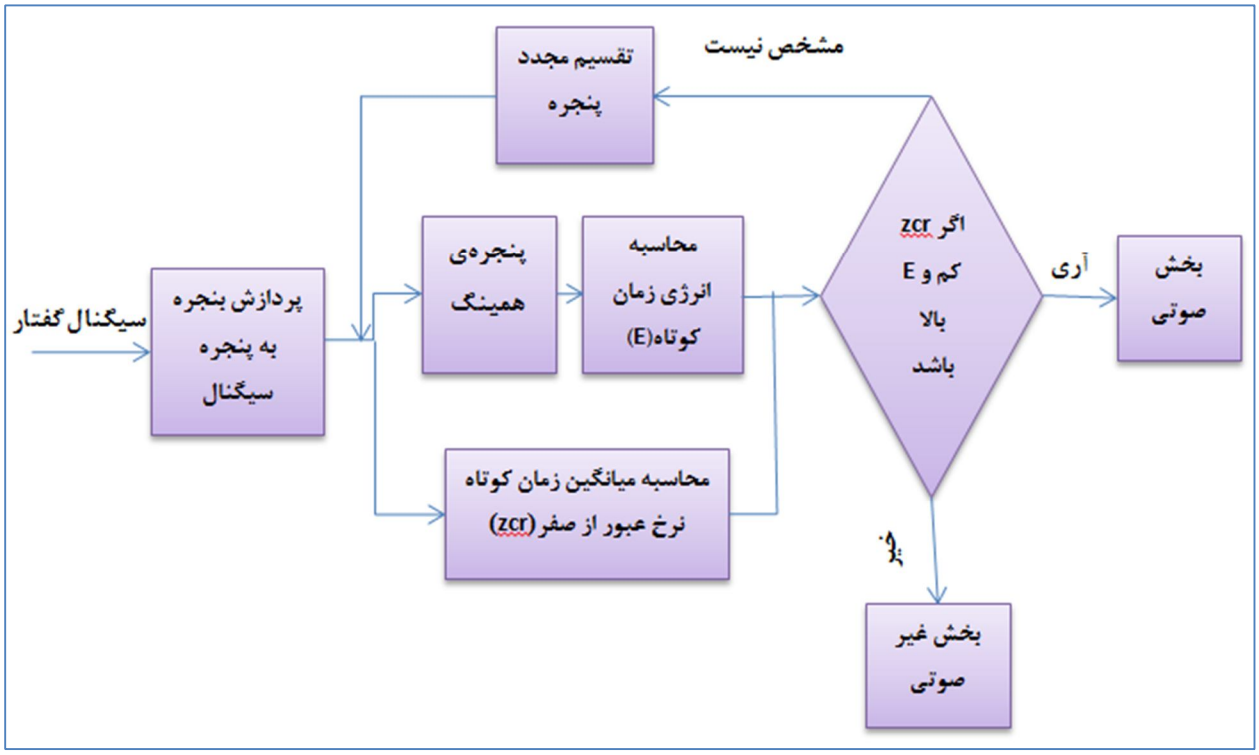
مکث آغازی و پایانی قرار بگیرد. برای حل این مشکل از الگوریتمی به نام Voice Activity Detection(VAD) یا تشخیص فعالیت صوتی استفاده می‌کنیم.



شکل ۴-۶ وجود مکث طولانی قبل و بعد از بیان یک کلمه. مکث طولانی قبل و بعد از بیان یک کلمه باعث گم شدن یا کمرنگ شدن نقش اطلاعات بخش اصلی و دور شدن مدل ایجاد شده از حالت ایده‌آل می‌گردد.

روش‌های زیادی توسط محققان برای تفکیک بخش‌های صوتی و غیر صوتی در یک سیگنال گفتار وجود دارد. بررسی و پیاده سازی این روش‌ها خارج از محدوده‌ی این تحقیق است. ما از ترکیب دو الگوریتم ساده‌ی انرژی و نرخ عبور از صفر استفاده کرده‌ایم. اگر بخواهیم به طور خلاصه بیان کنیم، قسمت صوتی سیگنال که به دلیل تحریک vocal tract توسط جریان متناوب هوا ایجاد شده است، نرخ عبور از صفر پایینی دارد. در حالی که قسمت‌های غیر صوتی سیگنال که توسط انقباض vocal tract ایجاد شده است، نرخ عبور از صفر بالایی دارد.

انرژی، پارامتر دیگری برای دسته بندی قسمت‌های صوتی و غیر صوتی است. قسمت‌های صوتی سیگنال به علت دوره‌ی تناوب و دامنه‌ی کمتر نسبت قسمتهای غیر صوتی، دارای انرژی بیشتری هستند. در شکل ۴-۷ بلوک دیاگرام الگوریتم استفاده شده برای جداسازی قسمتهای غیر گفتاری که در این تحقیق استفاده شده است، رسم شده است.



شکل ۴-۷ بلوک دیاگرام الگوریتم Voice Activity Detection(VAD)

۴-۴-۳ استخراج ویژگی:

برای استخراج ویژگی‌های صوتی از ماژول نرم افزاری Hcopy.exe واقع در بسته‌ی نرم افزاری HTK استفاده شده است. به دلیل کارایی بهتر ویژگی‌های MFCC نسبت به سایر ویژگی‌ها، از ضرایب MFCC در این تحقیق استفاده شده است. نحوه‌ی انتخاب ضرایب به این صورت است:

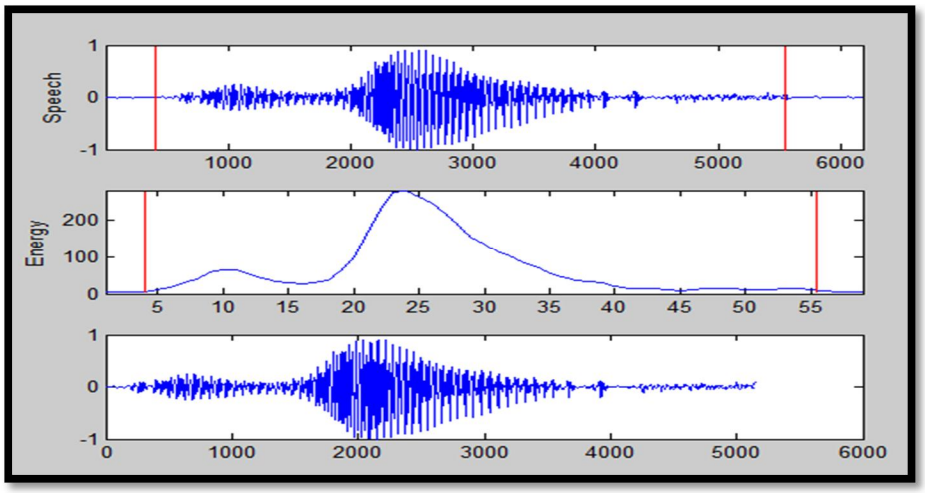
۳۹ ضریب = ۱۳ MFCCs + ۱۳ Δ-MFCCs + ۱۳ Δ^2-MFCCs

استفاده از فریم‌هایی با طول ۲۵ میلی ثانیه به صورت پنجره‌ی همینگ با همپوشانی ۵۰ درصد و استفاده از ۲۲ فیلتربانک نیز دیگر مشخصه‌های قابل ملاحظه در اینجا می‌باشد.

۴-۵ نتایج و ارزیابی :

به منظور ارزیابی عملکرد الگوریتم پیشنهادی، سه آزمایش مختلف با استفاده از سه دادگان گفتاری انجام شده است. در آزمایش اول که با استفاده از دادگان گفتاری فارس دات انجام شده است، از ۱۰ کلمه‌ی کلیدی مختلف که هر کدام دارای ۲۰ تکرار هستند استفاده کردیم. در ابتدا فایل‌های صوتی کلمات را به ۴ دسته تقسیم کردیم. در هر دسته ۵ تکرار از هر کلمه‌ی کلیدی وجود دارد. از ۴ تکرار اول به منظور آموزش و از تکرار پنجم به عنوان داده‌ی آزمون استفاده کردیم. این بدان معنی است که در حالت کلی برای هر کدام از ۱۰ کلمه‌ی کلیدی از ۱۶ نمونه برای آموزش و ۴ نمونه برای آزمون و تست سیستم استفاده شده است.

به منظور آموزش یک مدل برای هر کلمه‌ی کلیدی، ابتدا بخش گفتاری تمام نمونه‌های آموزشی را توسط الگوریتم VAD از بخش غیر گفتاری جدا کردیم. شکل ۴-۸ بکار گیری الگوریتم VAD را برای کلمه‌ی ((Zero)) از آزمایش دوم نشان می‌دهد.



شکل ۴-۸ بکارگیری الگوریتم VAD برای کلمه‌ی zero. اولین شکل از بالا سیگنال اصلی را نشان می‌دهد. شکل میانی منحنی انرژی و شکل پایین سیگنال خروجی را نشان می‌دهد.

در ادامه از تمام نمونه‌های آموزشی هر کدام از کلمات کلیدی ضرایب MFCC را استخراج می‌کنیم. نمونه‌ای که طول کمتری داشته باشد، دارای ماتریس ضرایب با عرض کمتری خواهد بود. ماتریس با عرض کوچکتر را به عنوان نمونه‌ی مرجع انتخاب کرده و طبق روشی که در بخش پیش توضیح دادیم سایر نمونه‌های آموزشی را هم بعد با ماتریس مرجع می‌سازیم. سپس از تمام نمونه‌های آموزشی میانگین گرفته و این میانگین حاصل را در شبکه‌ی تشخیص قرار می‌دهیم. سپس تمام نمونه‌هایی که به عنوان نمونه‌ی تست کنار گذاشته بودیم را توسط الگوریتم DTW با مدل هر کلمه‌ی کلیدی در شبکه مقایسه کرده و مدلی که کمترین فاصله با نمونه‌ی آزمون دارد را به عنوان کاندیدای تشخیص معرفی می‌کنیم.

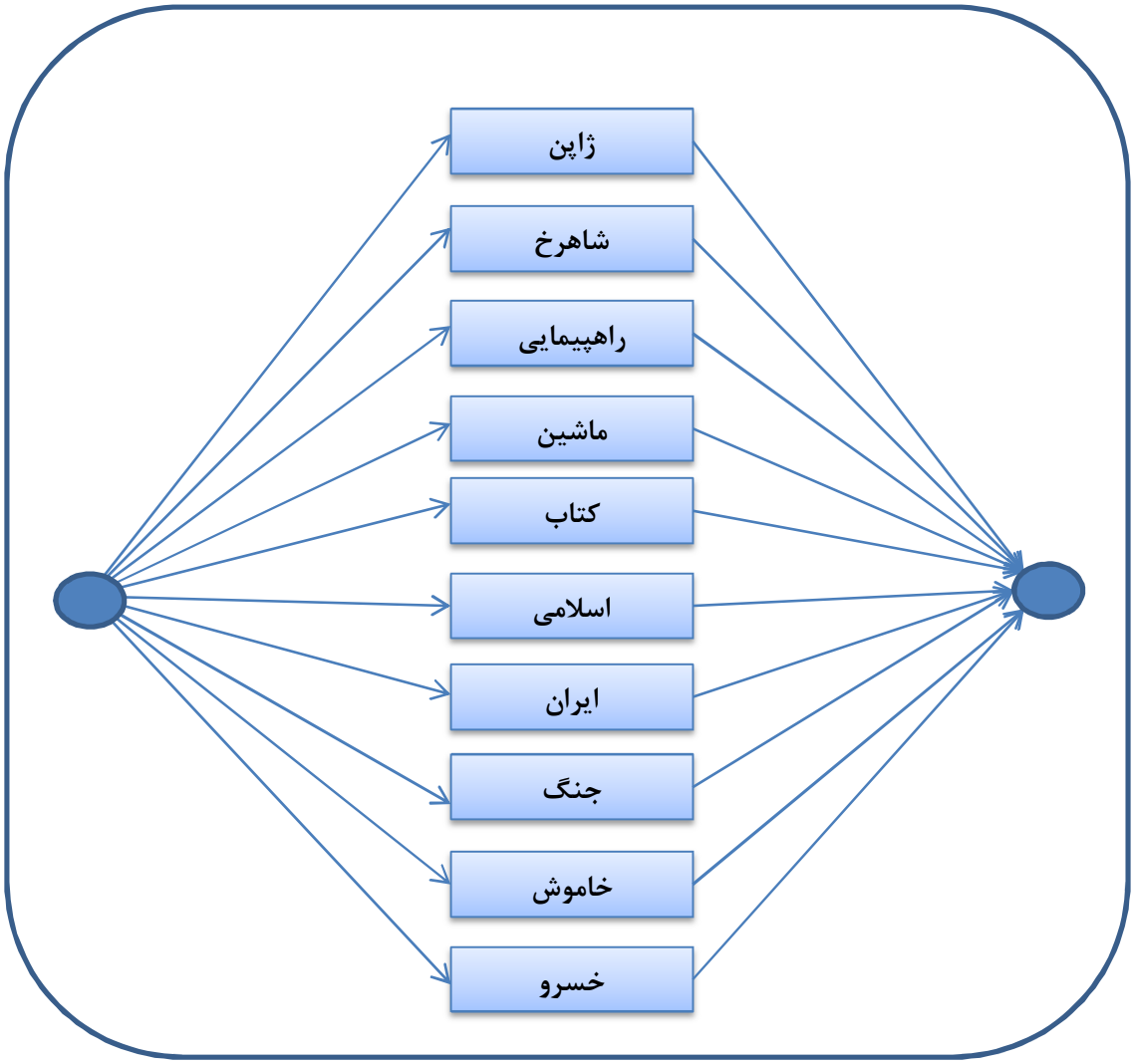
آزمایشات این بخش محدود به آموزش و تست ساده می‌شوند و برخلاف حالت پیوسته که در فصل بعد به آن خواهیم پرداخت، دارای ملاحظات و پیچیدگی زیادی نیستند. در ادامه توپولوژی شبکه و درصد تشخیص صحیح برای دو دادگان فارس دات و اعداد زبان انگلیسی آورده شده است.

آزمایش اول: شناسایی کلمات کلیدی تصادفی

تعداد نمونه‌های آموزشی برای هر کلمه‌ی کلیدی:۱۶

تعداد نمونه‌های تست برای هر کلمه‌ی کلیدی: ۴

دادگان : فارس دات



شکل ۴-۹ شبکه تشخیص کلمات کلیدی فارسی. از مدل زوائد استفاده نشده است.

در این شبکه، مدل‌های تمام کلمات کلیدی به صورت موازی با یکدیگر قرار می‌گیرند. در حین فرایند تشخیص، تمام کلمات تست پشت سر هم به شبکه نشان داده می‌شوند و فاصله‌ی هر کدام از آنها طبق معیار DTW با هر کدام از مدل‌ها محاسبه می‌شود. مدلی که کمترین فاصله را با گفتار تست ورودی داشته باشد به عنوان کاندیدای تشخیص معرفی می‌شود. بعد از اتمام فرایند آموزش و تشخیص، خروجی برنامه به این صورت در اختیار ما قرار می‌گیرد:

جدول ۴-۴ ماتریس تشخیص. ماتریس تشخیص نشان دهنده میزان تشخیص کلمات مختلف است

	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰
۱	۴	۰	۰	۰	۰	۰	۰	۰	۰	۰
۲	۰	۴	۰	۰	۰	۰	۰	۰	۰	۰
۳	۰	۰	۴	۰	۰	۰	۰	۰	۰	۰
۴	۰	۰	۰	۴	۰	۰	۰	۰	۰	۰
۵	۰	۰	۰	۰	۴	۱	۰	۰	۰	۰
۶	۰	۰	۰	۰	۰	۳	۰	۰	۰	۰
۷	۰	۰	۰	۰	۰	۰	۴	۰	۰	۰
۸	۰	۰	۰	۰	۰	۰	۰	۴	۰	۰
۹	۰	۰	۰	۰	۰	۰	۰	۰	۴	۰
۱۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۴

۴-۵-۱ تحلیل ماتریس تشخیص:

ماتریس تشخیص یک ماتریس مربعی N * N است که مشخص می‌کند که هرکدام از کلمات تست به عنوان کدام کلمه کلیدی تشخیص داده شده‌اند. عناصر قطری نشان دهنده‌ی تعداد تشخیص صحیح هر کلمه هستند. عناصر غیر قطری، نمایانگر کلمه‌ای هستند که به اشتباه بجای کلمه‌ی اصلی تشخیص داده شده است. در این آزمایش، تعداد کلمات آزمون برای هر کلمه‌ی کلیدی ۴ عدد می‌باشد. به عنوان مثال، ۴ کلمه‌ی تست مربوط به کلمه‌ی کلیدی اول به درستی کلمه‌ی کلیدی اول را به عنوان کاندید تشخیص معرفی کرده‌اند (ستون اول را ملاحظه کنید). اما از ۴ نمونه‌ی تست کلمه‌ی کلیدی ششم، ۳ نمونه، کلمه‌ی کلیدی ششم را به درستی تشخیص داده‌اند ولی ۱ نمونه به اشتباه کلمه‌ی کلیدی پنجم را به عنوان کاندید تشخیص معرفی کرده است (ستون ششم را ملاحظه کنید). به طور مشابه برای ستون هفتم که مربوط به نمونه‌های تست هفتمین کلمه‌ی کلیدی است، ۴ نمونه کلمه‌ی کلیدی هفتم را به درستی تشخیص داده‌اند. برای بدست آوردن درصد تشخیص صحیح کلی مطابق فرمول زیر عمل می‌کنیم:

درصد تشخیص صحیح کلی = (تعداد کل تشخیص/تعداد تشخیص صحیح) * ۱۰۰ (۴-۲)

با مراجعه به جدول و جمع عناصر قطری، تعداد تشخیص‌های صحیح برابر با ۳۹ بدست می‌آید. تعداد کل تشخیص‌ها نیز برابر با ۴۰ است. در نتیجه دقت سیستم بدین صورت است:

درصد تشخیص صحیح = (۳۹/۴۰) * ۱۰۰ = ۹۷.۵ %

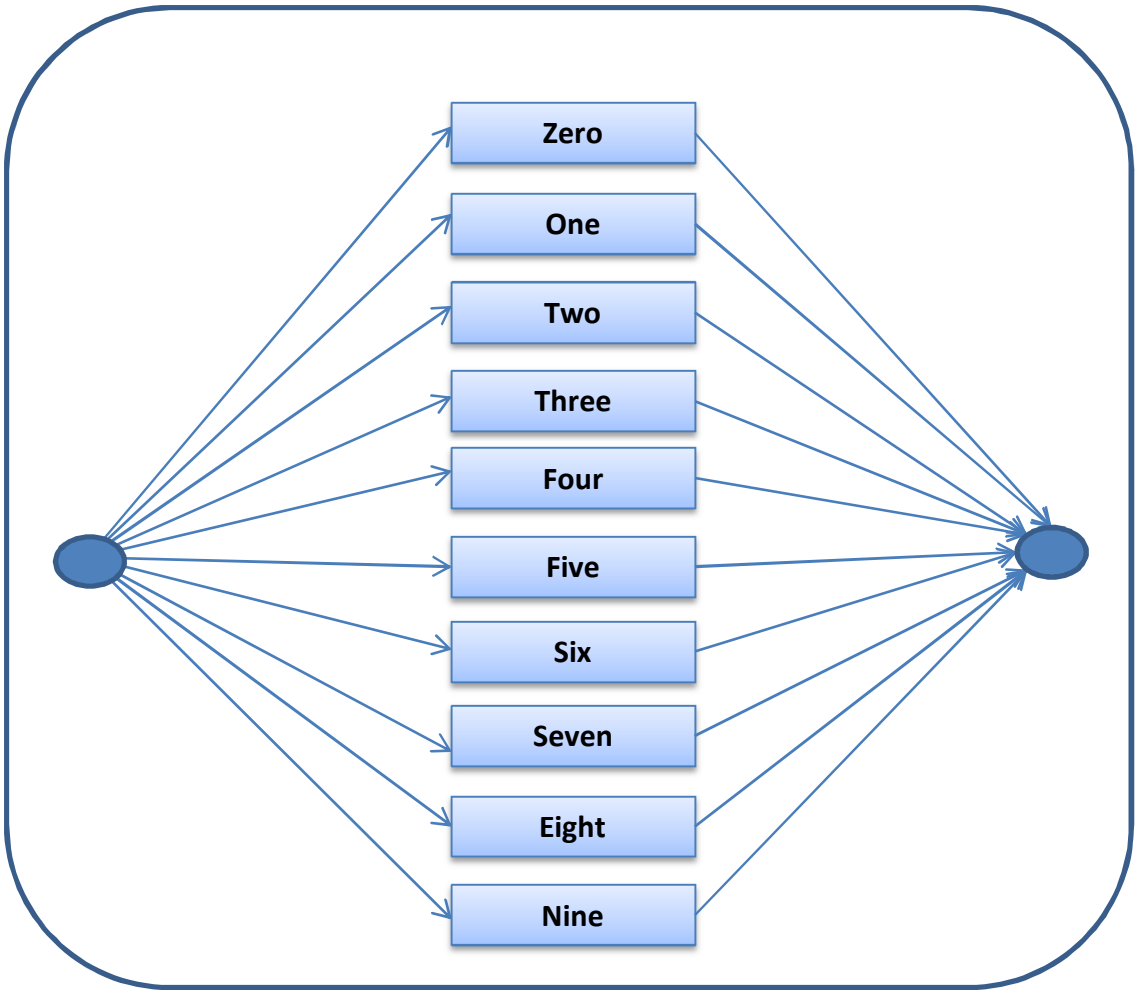
یک بار نیز برای تشخیص اعداد ۰ تا ۹ در زبان انگلیسی تکرار کردیم که نتایج آن در ادامه آورده شده است.

آزمایش دوم: سیستم شماره گیر صوتی

تعداد نمونه‌های آموزشی برای هر کلمه‌ی کلیدی:۴۰

تعداد نمونه‌های تست برای هر کلمه‌ی کلیدی: ۱۰

دادگان: اعداد انگلیسی ضبط شده



شکل ۴-۱۰ شبکه تشخیص اعداد در زبان انگلیسی

جدول ۴-۵ ماتریس تشخیص برای اعداد ۰ تا ۹ در زبان انگلیسی

	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰
۱	۱۰	۰	۰	۰	۰	۰	۰	۰	۰	۰
۲	۰	۱۰	۰	۰	۰	۰	۰	۰	۰	۰
۳	۰	۰	۸	۰	۰	۰	۰	۰	۰	۰
۴	۰	۰	۰	۸	۰	۰	۰	۰	۰	۰
۵	۰	۰	۰	۰	۹	۰	۰	۰	۰	۰
۶	۰	۰	۰	۰	۱	۱۰	۰	۰	۰	۰
۷	۰	۰	۲	۱	۰	۰	۱۰	۰	۰	۰
۸	۰	۰	۰	۰	۰	۰	۰	۱۰	۰	۱
۹	۰	۰	۰	۱	۰	۰	۰	۰	۱۰	۰
۱۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۹

با مراجعه به جدول و جمع عناصر قطری، تعداد تشخیص‌های صحیح برابر با ۹۴ بدست می‌آید. تعداد کل تشخیص‌ها نیز برابر با ۱۰۰ است. در نتیجه دقت سیستم بدین صورت است:

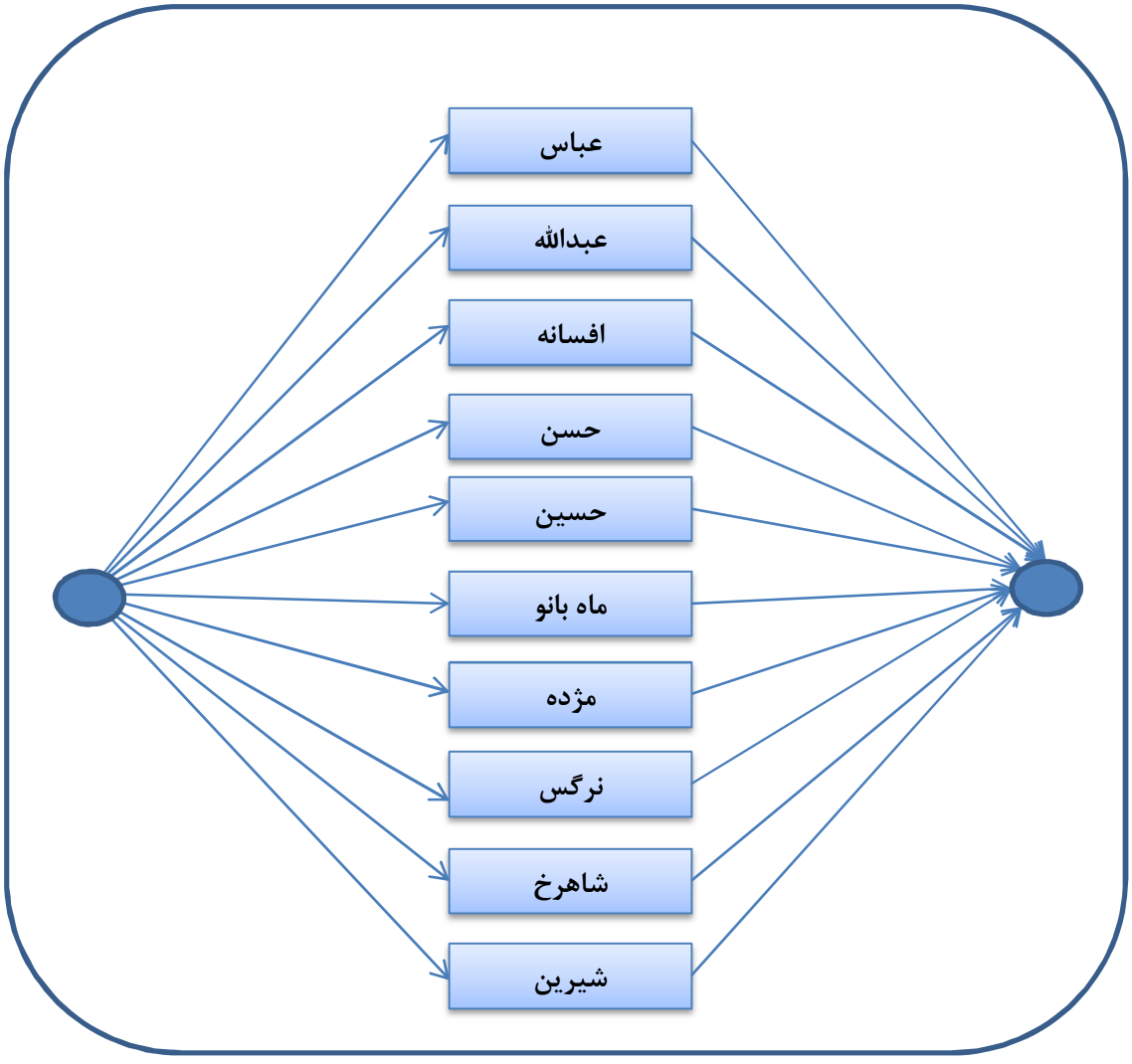
درصد تشخیص صحیح = (۹۴/۱۰۰) * ۱۰۰ = ۹۴%

آزمایش سوم : سیستم دفترچه تلفن صوتی

تعداد نمونه‌های آموزشی برای هر کلمه‌ی کلیدی :۸

تعداد نمونه‌های تست برای هر کلمه‌ی کلیدی : ۱

دادگان : فارس دات



شکل ۴-۱۱ دفترچه‌ی تلفن صوتی

جدول ۴-۶ ماتریس تشخیص دفترچه تلفن صوتی

	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰
۱	۱	۰	۰	۰	۰	۰	۰	۰	۰	۰
۲	۰	۱	۰	۰	۰	۰	۰	۰	۰	۰
۳	۰	۰	۱	۰	۰	۰	۰	۰	۰	۰
۴	۰	۰	۰	۱	۰	۰	۰	۰	۰	۰
۵	۰	۰	۰	۰	۱	۰	۰	۰	۰	۰
۶	۰	۰	۰	۰	۰	۱	۰	۰	۰	۰
۷	۰	۰	۰	۰	۰	۰	۱	۰	۰	۰
۸	۰	۰	۰	۰	۰	۰	۰	۱	۰	۰
۹	۰	۰	۰	۰	۰	۰	۰	۰	۱	۰
۱۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۱

با مراجعه به جدول و جمع عناصر قطری، تعداد تشخیص‌های صحیح برابر با ۱۰ بدست می‌آید. تعداد کل تشخیص‌ها نیز برابر با ۱۰ است. در نتیجه دقت سیستم بدین صورت است:

درصد تشخیص صحیح = (۱۰/۱۰) * ۱۰۰ = ۱۰۰%

همانطور که ملاحظه می‌شود با وجود داده‌های آموزشی کافی و مناسب، دستیابی به دقت بالای تشخیص دور از دسترس نخواهد بود.

فصل پنجم

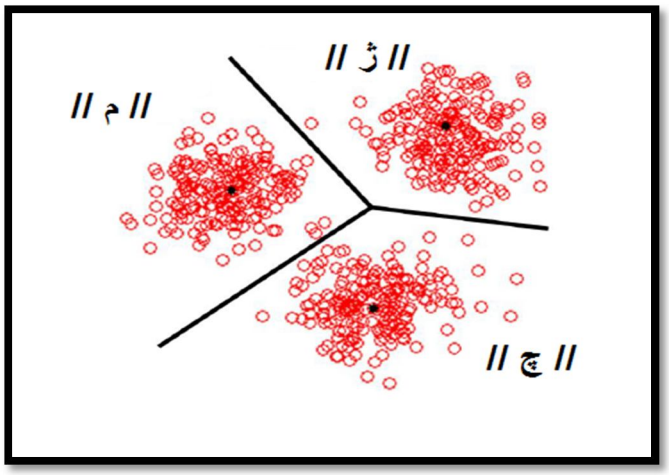
روش پیشنهادی ب

جستجوی کلمات کلیدی در گفتار پیوسته

۵-۱ نگاه کلی:

یک سیستم عمومی تشخیص گفتار، از لحاظ کارکرد می‌تواند به دو فاز عملکردی عمده تقسیم شود؛ مرحله‌ی آموزش و مرحله‌ی آزمون(تست). روش ارائه شده در این بخش بر پایه‌ی مقایسه‌ی ویژگی‌های کوانتیزه شده‌ی جمله یا جملات با کلمه و یا کلمات کلیدی، با استفاده از الگوریتم DTW است. به منظور کوانتیزه کردن یک بردار ویژگی گفتار، ابتدا باید یک فضای ویژگی که شامل تمام تغییرات آوایی ممکن در یک زبان هست را ایجاد کنیم. در ادامه این فضای ویژگی به چندین خوشه تقسیم می‌شود به نحوی که هر خوشه محتوی ویژگی‌های مربوط به یک صدای واحد است. این خوشه بندی در واقع برای ایجاد تمایز بین صداهای مختلف است. یک کلمه‌ی کلیدی یا یک جمله را می‌توان به صورت دنباله‌ای از صداهای مختلف فرض کرد. زمانی که یک جمله و یک کلمه که به صورت گفتاری هستند را بتوانیم به صورت یک دنباله ای از صداها بیان کنیم، آنگاه با استفاده از یک الگوریتم تطبق نمونه‌ها می‌توانیم میزان شباهت کلمه‌ی کلیدی نمونه و گفتار آزمون را اندازه بگیریم. در شکل ۵-۱، نحوه‌ی خوشه بندی صداها مشخص شده است. البته این خوشه بندی دقیقا به این شکل انجام نمی‌شود وتعداد خوشه‌ها محدود به تعداد واج‌ها در یک زبان نیست و می‌تواند توسط کاربر تغییر کند. اما برای نمایش ساده تر و بهتر در اینجا فرض بر این است که هر خوشه نمایانگر یک واج یا سیلابس در زبان فارسی است. با استفاده از خوشه‌های نشان داده شده در شکل، می‌توان کلمه‌ی کلیدی را از طریق جستجوی جمله برای یافتن دنباله‌ای مشابه با دنباله‌ی کلمه‌ی کلیدی در جمله پیدا کرد، به این صورت که باید بررسی کنیم ببینیم آیا همان دنباله از صداها که در کلمه‌ی کلیدی وجود دارد؛ در جمله نیز وجود دارد یا خیر. برای فهم بهتر این موضوع، فرض کنید دنباله‌ای از اعداد به صورت مقابل وجود دارند:

دنباله:{۱ و ۵ و ۷۴ و۱۲ و ۵ و ۴۱ و ۱۳ و ۸ و ۲۳ و ۸ و ۱ و ۲۳ و ۹ و ۱۱ و ۲ و ۱ و ۷ }



شکل ۵-۱ خوشه‌های واجی. فضای ویژگی به سه واج // چ //، // م // و // ژ // تقسیم شده است. هیچ جداسازی آشکاری روی شرایط مرزی وجود ندارد و ویژگی‌های متعلق به یک دسته ممکن است در دسته‌ی دیگر واقع شوند.

این دنباله، دنباله‌ی اصلی است و ما می‌خواهیم جستجو را درون آن انجام دهیم. دنباله‌ی جستجوی ما {۱۳ و ۸ و ۲۲ و ۷ و ۱} است. حالا باید تعیین کنیم که کدام بخش از دنباله‌ی اصلی شباهت بیشتری به دنباله‌ی جستجو دارد. سیستم‌های کلاسیک تشخیص کلمات کلیدی که عموما از مدل مخفی مارکوف استفاده می‌کنند، دارای محدودیت‌های فراوانی هستند و عین حال بازدهی و دقت آنها در صورت عدم وجود داده‌های آموزشی فراوان، بسیار کم می‌شود. علاوه بر این، برای هر زبان خاص باید از داده‌های آموزشی همان زبان استفاده شود. در این روش‌ها باید متن نوشتاری از داده‌های آموزشی با دقت زیادی تهیه شود که مستلزوم انجام دستی است و این امر از لحاظ عملی بسیار دشوار است. سیستم‌های تشخیص گفتار تجاری از لحاظ پوشش زبانی به حدود ۵۰ تا ۱۰۰ زبان محدود می‌شوند. این میزان پوشش برای حدود ۷۰۰۰ زبانی که در دنیا وجود دارد بسیار کم است[۶]. به منظور پوشش زبان‌های دیگر، ایجاد یک روش برای آموزش بدون نظارت خوشه‌ها، ضروری است. علاوه براین در تشخیص کلمات کلیدی، گفتار باید قبل از بکارگیری الگوریتم تطبیق نمونه‌ها، به یک رشته‌ی متنی تبدیل شود که این کار نیازمند بکارگیری روشهای پیچیده‌ای نظیر

HMM است. دقت تشخیص در این روش به دقت الگوریتم تشخیص آواها بستگی دارد که دقت کلی سیستم را پایین می‌آورد.

مشکلات و محدویت‌های فراوانی که روش‌های قدیمی در حوزه‌ی تشخیص کلمات کلیدی دارند، محققان را بر آن داشته است تا از روش‌هایی با انعطاف بیشتر استفاده کنند که در آنها مشکل کمبود داده‌های آموزشی و ازدیاد بار محاسباتی تا حدود زیادی مرتفع شود. اخیرا تحقیقاتی برپایه‌ی مدلهای مخلوط گوسی[1] انجام شده است[۲]. در اینجا خوشه بندی توسط مدل‌های مخلوط گوسی انجام می‌شود و احتمال اینکه هر بردار متعلق به یک خوشه‌ی خاص باشد توسط یک بردار احتمالاتی گوسی معین می‌شود. اختلاف بین بردارهای احتمالات پیشین کلمه‌ی کلیدی و جمله، به صورت ضرب نقطه ای آنها محاسبه می‌شود: $D(p,q)=-\log(p,q)$ که در آن p و q به ترتیب متعلق به کلمه‌ی کلیدی و جمله است. در این روش نیازی به موجود بودن رونوشت متنی از فایل‌های صوتی آموزشی وجود ندارد. در ادامه مدل‌های مخلوط گوسی برای تمام داده‌های آموزشی محاسبه می‌شوند و تشخیص کلمه‌ی کلیدی بر پایه‌ی اندازه گیری احتمالات پیشین روی یک مسیر غیر خطی انجام می‌شود. علی رغم مرتفع شدن مشکل وجود داده‌های آموزشی دارای رونوشت، از لحاظ محاسباتی، محاسبه‌ی مدلهای مخلوط گوسی و احتمالات پیشین و ضرب نقطه‌ای برای هر فریم، امری زمان بر و پرهزینه است. از این رو در مطالعه‌ای که ما انجام داده‌ایم، سعی شده است از یک روش ساده‌تر و دارای بار محاسباتی و پیچیدگی کمتر و در عین حال کارامد، استفاده شود.

۵-۲ آموزش سیستم :

[1] - Gaussian Mixture Models(GMM)

۵-۲-۱ آموزش خوشه:

یک کلمه‌ی کلیدی را می‌توان به صورت رشته‌ای از صداها در نظر گرفت. بنابراین جستجوی یک کلمه‌ی کلیدی در یک جمله را می‌توان به صورت جستجوی یک رشته‌ی کوتاه در یک رشته‌ی بلند فرض کرد. نکته‌ی مهمی‌که باید همیشه در خاطر داشت این است که یک کلمه اگر چند بار تکرار شود، حتی اگر این تکرارها توسط یک نفر خاص انجام شوند، باز هم لزوما از لحاظ صوتی یکسان نخواهند بود. وجود این تغییرات باعث ایجاد تغییراتی در ویژگی‌های استخراج شده از صوت می‌گردد. و زمانی که از دو کلمه‌ی یکسان که از لحاظ صوتی تغییراتی باهم دارند، استخراج ویژگی انجام دهیم، این ویژگی‌ها دارای تغییراتی خواهند بود. به منظور منعطف کردن سیستم در مقابل تغییرات کوچک، از تکنیک کوانتیزاسیون استفاده می‌شود. به این صورت که کل فضای ویژگی‌ها به تعداد محدودی از کلاسترها تقسیم می‌شود. از آنجا که ویژگی‌های استخراج شده از صوت به صورت برداری هستند، به این فرایند کوانتیزاسیون بردار[۱] گفته می‌شود.

این ایده در حالت کلی به این صورت است که در ابتدا یک فضای گسترده از ویژگی‌های صوتی را ایجاد می‌کنیم. در ادامه این فضای ویژگی‌ها باید به تعداد محدودی کلاس تقسیم شود به نحوی که صداهای یکسان در یک کلاس قرار بگیرند. بدین منظور، یک فضای ویژگی را از ویژگی‌های صوتی استخراج شده از جملات طولانی گفتاری شامل گوینده‌های مختلف تشکیل می‌دهیم. این فضا باید اندازه‌ای بزرگ باشد که تمام صداهای ممکن را در خود جای دهد. هر نقطه در فضای ویژگی، یک بردار ویژگی واحد را نشان می‌دهد. بردارهای ویژگی مربوط به اصوات یکسان لزوما روی هم واقع نمی‌شوند و ممکن است دارای تغییراتی نسبت به هم باشند. از این رو کل فضای ویژگی به تعداد محدودی کلاستر تقسیم می‌شود(۵۱۲-۶۴ برای آزمایشهای ما). هر کلاستر شامل بردارهای ویژگی

[۱] - Vector Quantization(VQ)

اصوات مشابه است. از آنجا که هدف ما پیدا کردن ناحیه‌ای در جمله است که توالی صوتی مشابهی با توالی صوتی کلمه‌ی کلیدی دارد، میزان تشخیص کلی بیشتر از آنکه به دقت یک کلاستر خاص وابسته باشد، به نحوه‌ی توالی کلاسترها بستگی دارد.

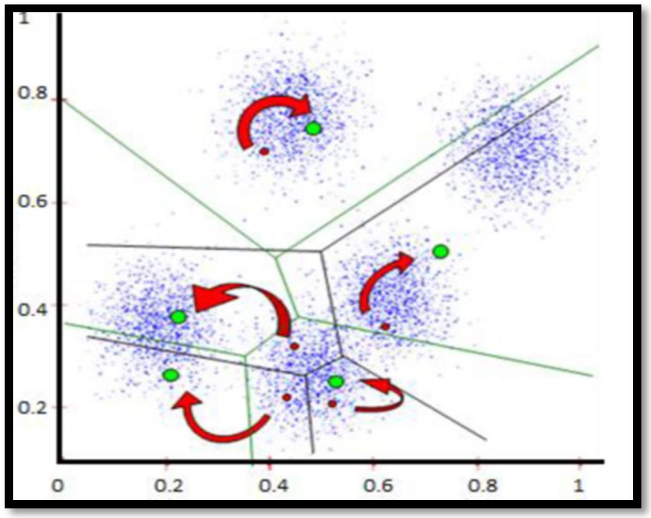
از الگوریتم k-means clustering [۱۸]، در مرحله‌ی خوشه بندی استفاده می‌شود. در این الگوریتم سعی بر این است که تا حد ممکن فاصله‌ی بردارهای درون هر کلاستر از مرکز کلاستر کمتر شوند. یعنی فضا را به نحوی دسته بندی کنیم که بعد اتمام دسته بندی، فاصله‌ی برداری که در یک دسته‌ی خاص قرار گرفته است، از مرکز همان دسته، کمتر از فاصله تا مراکز سایر کلاسترها باشد. در اینجا مراحل خوشه بندی بر اساس الگوریتم k-means بیان شده است:

- فضایی شامل تمام ویژگی‌های ممکن ایجاد می‌کنیم. هر کلاستر دارای یک مرکز است که این مرکز در واقع میانگین تمام داده‌های آن کلاستر است. از آنجا که هنوز داده‌ها را کلاستر بندی نکرده‌ایم، مراکز را هم در اختیار نداریم. از این رو در ابتدا به صورت تصادفی K عدد از داده‌های موجود در فضای ویژگی را به عنوان مراکز اولیه انتخاب می‌کنیم.(K تعداد کلاسترهای مورد نیاز است).
- فاصله‌ی هر داده در فضای ویژگی‌ها را با هرکدام از مراکز اولیه حساب می‌کنیم. هر داده را در دسته‌ای قرار می‌دهیم که فاصله‌کمتری با مرکز آن دسته دارد. به عبارت دیگر، ما K مرکز کلاستر و n نقطه در فضای ویژگی داریم. در این مرحله این n نقطه بر حسب فاصله‌ی خود از مرکز دسته‌ها، که در مرحله‌ی اول به صورت تصادفی انتخاب شده‌اند، به K دسته تقسیم می‌شوند.
- مرکز هر کلاستر را از طریق محاسبه‌ی مجدد میانگین داده‌ها در آن کلاستر، مجددا محاسبه می‌کنیم:

- $M_k = \frac{1}{N_k} . \sum_{j=1}^{N_k} X_{jk}$

در اینجا، M_k، میانگین جدید، N_k، تعداد بردارها در کلاستر k و X_{jk}، در واقع k-امین بردار متعلق به آن کلاستر است.

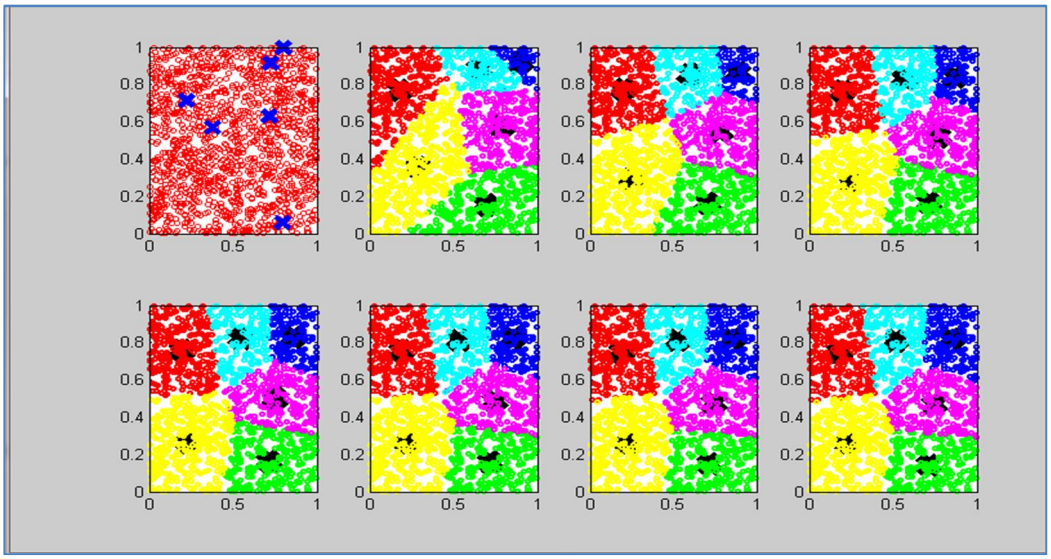
- مراکز بدست آمده‌ی جدید را با مراکز قدیمی‌جایگزین می‌کنیم.
- مراحل ۲ تا ۴ را آنقدر تکرار می‌کنیم تا جایی که اختلاف میانگین‌های بدست آمده در یک مرحله و مرحله‌ی قبلی آن از یک حد خاص، مثلا ۰.۰۱، کمتر شود.



شکل ۵-۲ کوانتیزاسیون بردار. فضای ویژگی به ۵ کلاستر تقسیم شده است. هرکدام از آنها توسط یک مرکز نمایش داده می‌شوند. دایره‌های کوچک و بزرگ، مراکز را بعد از اولین و دومین تکرار نشان می‌دهند.

شکل ۵-۲ به صورت تصویری، کوانتیزاسیون بردار را نشان می‌دهد. برای شروع، K مرکز به صورت تصادفی انتخاب می‌شوند. سپس هر کدام از بردارهایی که در فضای ویژگی پراکنده هستند به نزدیکترین مرکز نسبت داده می‌شوند. این کار باعث تقسیم کل فضا به k کلاستر می‌شود. آنگاه میانگین هرکدام از کلاسترها محاسبه می‌شود و جایگزین مقدار میانگین پیشین می‌شود. این عمل تا رسیدن به همگرایی لازم ادامه می‌یابد.

هر بردار دوبعدی که در یک ناحیه‌ی خاص قرار می‌گیرد، توسط مرکز ناحیه تخمین زده می‌شود. در نتیجه کل فضای ویژگی که به K ناحیه تقسیم شده است، توسط k مرکز بیان می‌شود. هر مرکز دارای یک اندیس خاص است؛ بنابراین دنباله ای از بردارها را می‌توان به صورت دنباله ای از اندیس‌ها نشان داد که می‌تواند تا حدود زیادی از بار محاسباتی مساله کم کند.



شکل ۵-۳ آموزش کلاسترها. نحوه‌ی آرایش کلاسترها بعد از ۷ تکرار. در ردیف بالا سمت چپ داده‌ها در کل فضا پراکنده شده‌اند.۶ نقطه به صورت تصادفی در فضا به عنوان مراکز اولیه انتخاب شده است.

ویژگی‌های صوتی، بردارهایی با ابعاد زیاد هستند و این باعث می‌شود فضای ویژگی ما نیز ابعاد بالایی داشته باشد. برای حل این مشکل، از الگوریتمی که توضیح داده شد، به راحتی می‌توانیم استفاده کنیم. هر فریم را می‌توان توسط یک اندیس نمایش داد و در نهایت برای دنباله ای از فریم‌ها، یک کتاب کد[۱] ایجاد خواهد شد که نه تنها ابعاد، بلکه حجم محاسبات را نیز تا حد زیادی کاهش می‌دهد. همچنین تغییرات کوچک در بردار های ویژگی را از طریق کوانتیزه کردن آنها به مرکز کلاسترهایشان می‌توان از بین برد.

[۱] - Codebook

۵-۲-۲ تولید کتاب کد:

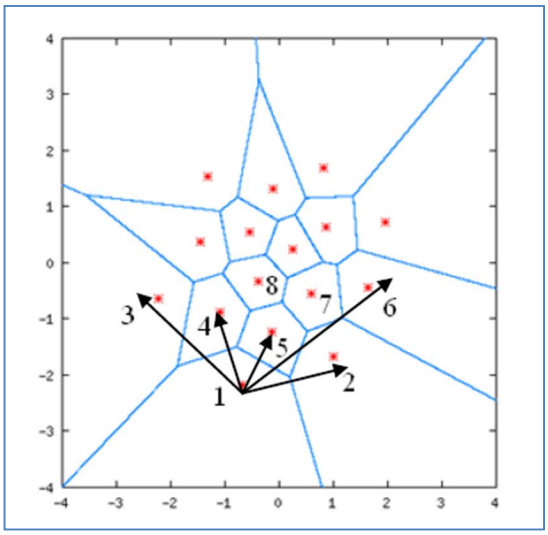
سیستم ما با این فرض طراحی شده است که حداقل یک و یا چندین نمونه از هر کلمه کلیدی موجود باشد. زمانی که آموزش کلاسترها تمام شد، فضای ویژگی به K کلاستر کوانتیزه شده، که با K مرکز نشان داده می‌شود. هرکدام از این مراکز، یک بردار ویژگی N بعدی هستند. تمام کلمات کلیدی موجود در ادامه توسط این کلاسترها کوانتیزه می‌شوند و هر فریم از یک کلمه‌کلیدی، توسط مرکز کلاستری که کمترین فاصله را با آن فریم دارد، نمایش داده می‌شود. در این صورت ما می‌توانیم هر کلمه‌ی کلیدی را به جای آنکه با بردارهای ویژگی N بعدی نمایش دهیم، با رشته‌ی یک بعدی از کتاب کد نمایش دهیم. این کار باعث نمایش ساده تر و افزایش کارایی در محاسبات می‌شود. اگر از یک کلمه‌ی کلیدی بیشتر از یک نمونه موجود باشد، بهترین راه این است که از تمامی ویژگیها میانگین بگیریم و یک کلمه‌ی عام را ایجاد کنیم.

به همان روشی که برای کلمات کلیدی بیان شد، جمله‌ی اصلی را نیز به یک دنباله از اندیسهای کتاب کد تبدیل می‌کنیم. نتیجه‌ی همه‌ی این مراحل، تولید یک رشته‌ی کوتاه از اندیس‌ها برای کلمات کلیدی و تولید یک رشته‌ی بزرگتر از اندیس‌ها برای نمایش جمله‌ی اصلی است.

۵-۲-۳ محاسبه‌ی ماتریس فاصله:

قبل از اینکه در مورد نحوه‌ی محاسبه‌ی ماتریس فاصله صحبت کنیم، باید ابتدا هدف از محاسبه‌ی ماتریس فاصله را شرح دهیم. ماتریس فاصله در واقع یک تخمین از احتمال تعلق نقطه‌ی مرکزی به هرکدام از کلاسترها است. یک فریم گفتار که متعلق به یک کلاستر خاص با تغییرات بیشتر است، ممکن است به جای اینکه در کلاستر خود قرار گیرد، در یکی از کلاسترهای همسایه قرار بگیرد.

بنابراین محاسبه احتمال تعلق یک بردار به سایر کلاسترها منطقی به نظر می‌رسد. این کار را می‌توان با اندازه گیری فاصله بین بردار مشاهده و مراکز سایر کلاسترها انجام داد. بکارگیری این روش باعث افزایش انعطاف سیستم و مقاومت بیشتر در برابر تغییرات آوایی خواهد شد.



شکل ۵-۴ . تخمین احتمال تعلق داده‌ی موجود در یک خوشه به سایر خوشه‌ها

همانطور که در شکل ۵-۴ نشان داده شده است، احتمال اینکه بردارهای نواحی ۲، ۳، ۴ و یا ۵ متعلق به ناحیه‌ی ۱ باشند، بیشتر از این است که این بردارها متعلق به ناحیه‌ی ۶ باشند. این کار را می‌توان با استفاده از فاصله‌ی بین مراکز نواحی مربوط به هر بردار تخمین زد. از آنجا که بردار ویژگی هم اکنون توسط مرکز کلاستر تخمین زده شده است، احتمال این که یک بردار متعلق به کلاستر دیگری باشد، توسط فاصله‌ی مراکز این دو کلاستر تخمین زده می‌شود. برای مثال، بردار V۱ در کلاستر R۱(Region) به C۱(مرکز ناحیه‌ی ۱) کوانتیزه می‌شود. بنابراین احتمال اینکه این بردار متعلق به ناحیه‌ی R۲ باشد، به صورت نسبتی از معیار فاصله‌ی |C۲-C۱| است. برای یک کتاب کد با ۶ کلاستر(K=٦) ماتریس فاصله به صورت شکل ۵-۵ است:

	C1	C2	C3	C4	C5	C6
C1	0	|C2-C1|	|C3-C1|	|C4-C1|	|C5-C1|	|C6-C1|
C2	|C2-C1|	0	|C3-C2|	|C4-C2|	|C5-C2|	|C6-C2|
C3	|C3-C1|	|C3-C2|	0	|C4-C3|	|C5-C3|	|C6-C3|
C4	|C4-C1|	|C4-C2|	|C4-C3|	0	|C5-C4|	|C6-C4|
C5	|C5-C1|	|C5-C2|	|C5-C3|	|C5-C4|	0	|C6-C5|
C6	|C6-C1|	|C6-C2|	|C6-C3|	|C6-C4|	|C6-C5|	0

شکل ۵-۵ ماتریس فاصله برای کتاب کد با ۶ کلاستر

همانطور که در شکل ۵-۵ مشخص است، ماتریس فاصله، یک ماتریس متقارن است که همه‌ی عناصر قطری آن برابر با صفر هستند.($D_{ij}=0$ اگر $i=j$ و $D_{ij}=D_{ji}$ اگر $i\neq j$). دلیل آنکه این ماتریس از قبل محاسبه می‌شود، در واقع سرعت بخشیدن به فرایند تشخیص است. به محض اینکه کلمه‌ی کلیدی نمونه و جمله، کوانتیزه و تبدیل به رشته‌ای از اندیس‌های کتاب کد شوند، احتمال محل وقوع هر فریم را می‌توان مستقیما با استفاده از ماتریس فاصله محاسبه کرد و دیگر نیازی به مراجعه‌ی مجدد به کلاسترهای اصلی نیست. بعد از کوانتیزاسیون بردار، کلمه‌ی کلیدی و جمله به صورت کتاب کد تبدیل می‌شوند. به عبارت دیگر دنباله‌ای از اندیس‌های کلاستر به این شکل نمایش داده می‌شوند:

کلمه‌ی کلیدی: C۱-C۲-C٤-C٦-C۱-C۳-C۲-C۱

بخشی از جمله : C۲-C٤-C٥-C٦-C۱-C۳-C٤-C۱-C۲-C۳-C٦

در شکل ۵-۶ یک نمونه از ماتریس احتمال آمده است:

C1	D_{12}	D_{14}								
C2	D_{22}	D_{24}								
C3	D_{32}	D_{34}								
C1	D_{12}	D_{14}								
C6	D_{62}	D_{64}								
C4	D_{42}	D_{44}	D_{45}							
C2	D_{22}	D_{24}	D_{25}							
C1	D_{12}	D_{14}	D_{15}	D_{16}	D_{11}					
↑Keyword Utterance→	C2	C4	C5	C6	C1	C3	C4	C1	C2	

شکل ۵-۶ ماتریس احتمال نمونه برای کلمه‌ی کلیدی و جمله کوانتیزه شده

شکل ۵-۶، محاسبه‌ی ماتریس احتمال برای کلمه‌ی کلیدی و جمله‌ی کوانتیزه شده را نشان می‌دهد. هر کدام از سلول‌های این ماتریس را می‌توان از ماتریس فاصله(ماتریس D) به دست آورد. برای مثال فاصله‌ی اولین فریم از کلمه‌ی کلیدی که به کلاستر C۱ تعلق دارد و اولین فریم از جمله که به کلاستر C۲ تعلق دارد، برابر با $D_{۱۲}$ است؛ است که در آن $D_{۱۲}=|C۱-C۲|$. این کار باعث جلوگیری از مقدار زیاد محاسبات در حین فرایند تشخیص خواهد شد.

۵-۳ تشخیص کلمه‌ی کلیدی:

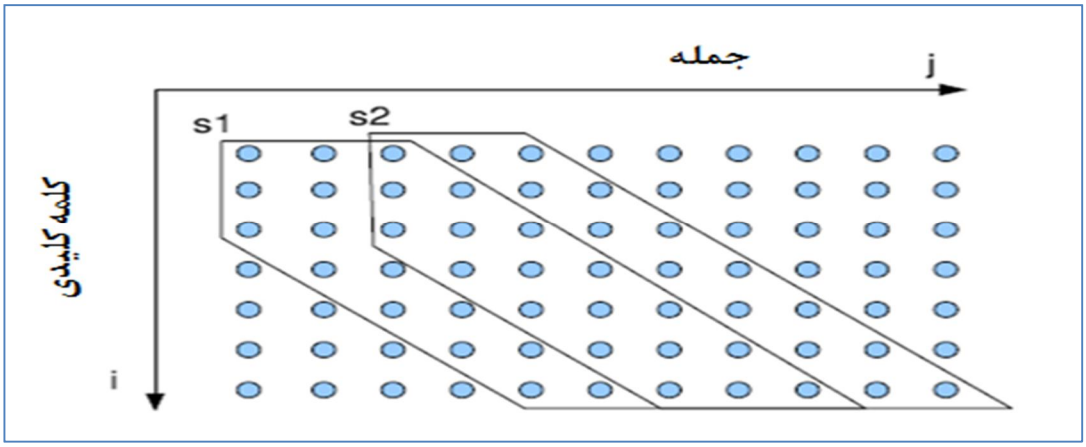
کوانتیزاسیون بردار منجر به ایجاد کتاب کد با تعداد کلاسترهای محدود می‌شود. هر کدام از این کلاسترها توسط یک مرکز واحد نشان داده می‌شوند. توسط این کتاب کد، هر کدام از کلمات کلیدی به یک رشته از اندیس‌های کتاب کد تبدیل خواهند شد. برای تشخیص کلمه‌ی کلیدی در یک گفتار،

آن گفتار یا جمله نیز ابتدا مانند کلمه‌ی کلیدی به رشته‌ای از اندیسهای کتاب کد تبدیل می‌شود. و ما در واقع دو رشته‌ی یک بعدی خواهیم داشت که نماینده‌ی جمله و کلمه‌ی کلیدی خواهند بود. اگر کلمه‌ی کلیدی در جمله‌ی مورد نظر وجود داشته باشد، آنگاه در قسمتی جمله، الگویی مشابه با الگویی که در کلمه‌ی کلیدی داشتیم وجود خواهد داشت و در آن ناحیه نمودار فاصله به طور یکنوا کاهش خواهد داشت. برای اینکه بتوانیم در برابر برخی از تغیرات ناشی از سرعت بیان و نحوه‌ی تلفظ، سیستمی مقاوم تر داشته باشیم، از یک تکنیک مقایسه‌ی انعطاف پذیر و قوی به نام ((Segmental-Dynamic Time Warping)) استفاده می‌کنیم.

۵-۳-۱ الگوریتم Segmental-Dynamic Time Warping :

الگوریتم Dynamic Time Warping ملقب به DTW، یک الگوریتم برای مقایسه‌ی دو کلمه توسط انحراف زمانی به منظور همترازی بهینه است، در حالی که Segmental-DTW دو گفتار پیوسته را گرفته و زیر دنباله‌های آنها را با هم مقایسه می‌کند. به عبارت دیگر در دو گفتار پیوسته بخش‌هایی از دو گفتار را که به هم شباهت دارند را پیدا می‌کند. Segmental-DTW در زمینه‌ی کشف بدون نظارت الگوها بسیار موفق عمل کرده است.[۱۲] و [۱] تحقیقاتی که اخیرا انجام شده است نشان می‌دهد که این روش می‌تواند در تشخیص بدون نظارت کلمات کلیدی نیز بکار رود.[۲][۳] . عمده‌ترین تغییری که این الگوریتم نسبت به روش سنتی DTW کرده است، تحمیل دو محدودیت بر روش سنتی است. اولین محدودیت، ایجاد یک پنجره‌ی تنظیم است که مسیری که الگوریتم DTW اتخاذ می‌کند را محدود به دو مرز در طرفین می‌کند. تغییر دوم این روش نسبت به روش DTW سنتی در این است که در بر خلاف روش سنتی که شروع و پایان مسیر محدود به دو نقطه‌ی خاص است، در این روش نقاط شروع و پایان متفاوتی را برای هر فریم می‌توانیم داشته باشیم. با بکارگیری

پنجره‌ی تنظیم، می‌توانیم فضای جستجو را به تعداد محدودی بخش تقسیم کنیم و مسیرهای همترازی چندگانه را تولید نماییم. همانطور که در شکل ۵-۷ نشان داده شده است، شرایط پنجره‌ی



شکل ۵-۷ Segmental-DTW با پنجره‌ی تنظیم به اندازه ۲(R=۲)

تنظیم نه تنها فرم مسیر انحراف را محدود می‌کند، بلکه نقطه‌ی پایانی نیز بسته به محل نقطه‌ی شروع، به یک بازه محدود می‌شود. برای مثال اگر نقطه‌ی شروع $(i_1,j_1) = (1,1)$ باشد، و مقدار $i_{end}=m$ باشد، آنگاه $j_{end} \in (1+m-R, 1+m+R)$. در نتیجه ماتریس اختلاف می‌تواند به چندین قطعه‌ی قطری پیوسته با عرض ۲R+۱ تقسیم شود. همانطور که در شکل ۵.۷ مشخص است، هر کدام ازین بخش‌های قطری دارای یک همپوشانی با همدیگر هستند که مقدار این همپوشانی قابل تنظیم است. بکارگیری الگوریتم S-DTW(segmental DTW) روی گفتار آزمون، منجر به ایجاد (n-۱)/R مسیر انحراف (warp) می‌شود که در آن n برابر تعداد فریم‌ها در گفتار و R برابر با اندازهِی پنجره است. هر مسیر دارای اندیس‌هایی به صورت i و j، طول مسیر و امتیاز اعوجاج است. بخش‌هایی که دارای کمترین اعوجاج باشند به عنوان ناحیه‌ی کاندید حضور کلمه‌ی کلیدی در نظر گرفته می‌شوند. از آنجا که هدف، پیدا کردن بخشی است که به کلمه‌ی کلیدی شبیه است، در قدم بعد، بخش‌هایی که دارای اعوجاج زیاد و امتیاز کم باشند حذف خواهند شد.

۵-۳-۲ منطق تصمیم گیری:

به محض اینکه امتیاز اعوجاج برای کل گفتار آزمون محاسبه شد، فرایند تشخیص را می‌توان بر اساس تعریف یک مقدار آستانه انجام داد. مقدار آستانه بدین معنی است که اگر امتیاز اعوجاج بدست آمده از الگوریتم DTW در یک محل از جمله، از یک مقدار خاص کمتر شود، آن محل احتمالا محل حضور کلمه‌ی کلیدی خواهد بود. کلمات کلیدی مختلف ممکن است دارای آستانه‌ی اعوجاج مختلفی در فرایند تشخیص باشند. محاسبه‌ی یک مقدار اعوجاج عمومی و فراگیر کار سخت و دشواری است و شاید اصلا وجود نداشته باشد. اگر بیشتر از یک نمونه برای یک کلمه‌ی کلیدی موجود باشد، یک راه خوب برای بهره بردن از این مساله این است که تمام این کلمات را نرمالیزه کرده و یک کلمه‌ی عمومی را از آنها استخراج کنیم تا تمام این کلمات را در بر بگیرد. از لحاظ تئوری این را می‌توان با بهینه کردن الگوریتم k-means برای کوانتیزاسیون بردار انجام داد. برای مثال ۳ اگر نمونه از یک کلمه‌ی کلیدی به شکل دنباله‌ای از اندیسهای کلاستری به شکل زیر در دسترس باشند؛

K۱: C۱ , C۵ , C۲۰ , C۵۳ , C۲۰۰ , C۱۵۰ , C۱۰,...

K۲: C۲ , C۶ , C۲۰ , C۵۲ , C۱۹۸ , C۱۵۰ , C۹ ,...

K۳: : C۱ , C۶ , C۲۲ , C۵۰ , C۲۰۰ , C۱۴۸ , C۸ ,...

هدف، چینش مجدد کلاسترها است به نحوی که اولین فریم تمام این ۳ کلمه‌ی کلیدی در یک کلاستر واحد قرار بگیرند و در نتیجه با یک مرکز کلاستر واحد نشان داده شوند. در این صورت قابلیت جداسازی آن کلاستر برای این فریم‌ها بسیار بیشتر از کلاستر معمولی بدست آمده توسط الگوریتم K-means خواهد بود. با این وجود، برخی از کلمات کلیدی می‌توانند تغییرات بیشتری از ۳ نمونه‌ی فوق داشته باشند. برای مثال اگر کتاب کد برای یک نمونه‌ی دیگر از همان کلمه‌ی کلیدی را

در نظر بگیریم، C۱۰ ,C۱۵۰ ,C۲۰۰ ,C۵۰ ,C۱۰۰ ,C۵ :K۴ آنگاه قرار دادن آنها در یک کلاستر واحد باعث نویزی شدن سیستم و کاهش دقت تشخیص کلی سیستم می‌شود. از این روکلاستر بهینه شده باید سعی کند اکثر و البته نه تمام فریم‌ها را در خود جای دهد. برای حل این مساله می‌توان از روش‌های SOM(Self Organizing Vector Maps)[۵] و SVM(Support Vector Machine)[۴] استفاده کرد که خارج از محدوده‌ی این تحقیق هستند.

یک راه دیگر برای بهره بردن از وجود چندین نمونه از هر کلمه‌ی کلیدی این است که یک امتیاز قطعه‌ای (Segmental Score) را برای هر کلمه‌ی کلیدی بدست آوریم و تمام این امتیازات را باهم ترکیب کنیم. یک بخش با میانگین امتیازی بالا، با احتمال بالاتری نسبت به بقیه، محل وقوع کلمه‌ی کلیدی خواهد بود. این روش از لحاظ محاسباتی دارای هزینه‌ی بالایی است اما از نظر پیاده سازی نسبت به روش اول که تمام نمونه‌های یک کلمه کلیدی را در یک کلمه‌ی عام مدل می‌کردیم، ساده‌تر است.

روش دیگر بر پایه رای گیری است. روش رای گیری به ما می‌گوید که اگر N نمونه از یک کلمه‌ی کلیدی داشته باشید، در صورتی که بیش از نصف آنها، یک محل را به عنوان محل حضور کلمه‌ی کلیدی تشخیص دهند، آن محل به احتمال فراوان، محل واقعی حضور کلمه‌ی کلیدی خواهد بود. از آنجا که این روش دارای سرعت و دقت بسیار بالایی است و در عین حال پیچیدگی کمتری نسبت به روش‌های بالا دارد، در این تحقیق از روش بر اساس رای گیری استفاده شده است.

۵-۴ آماده سازی برای انجام آزمایشات :

۵-۴-۱ معرفی دادگان:

انجام آزمایشات و نتیجه گیری، مهمترین بخش هر پروژه‌ی تحقیقاتی عملی است. در تحقیقاتی که در حوزه‌ی گفتار انجام می‌شود، وجود یک منبع گسترده‌ی دادگان صوتی استاندارد ضروری می‌باشد. یک منبع دادگان صوتی استاندارد باید شامل گفتارهای متنوع از افرادی با سنین و لهجه‌های مختلف و جنسیت متفاوت باشد که در یک محیط عاری از سروصدا و نویز و با ابزارهای استاندارد ضبط شده‌اند. برای آزمون عملی روش پیشنهادی در این بخش نیز از پایگاه استاندارد دادگان فارسی، **فارس دات(FARSDAT)** استفاده شده است. مجموعه دادگان فارس دات شامل داده‌ها صوتی متنوعی است که توسط ۳۰۴ گوینده فارسی زبان بیان شده است. تمامی این گوینده‌ها، از لحاظ سن، جنسیت، لهجه و سطح آموزش با یکدیگر متفاوت هستند. هر گوینده ۲۰ جمله را در دو فصل بیان می‌کند. دادگان فارس دات در مجموع دارای ۶۰۸ فایل صوتی است که هر فایل شامل ۱۰ جمله است که توسط یک گوینده بیان شده است. تمامی این داده‌های صوتی در یک اطاقک مخصوص واقع در لابراتوار زبان شناسی دانشگاه تهران ضبط شده‌اند. همچنین از یک میکروفون سونی با پاسخ فرکانسی ۸۰ هرتز تا ۱۶ کیلوهرتز و فاصله‌ی حدود ۱۲ سانتی متر با لب گوینده، استفاده شده است. یک نمونه از جملاتی که در یک فایل صوتی در دادگان فارس دات وجود دارد در اینجا آورده شده است:

فایل : "S۱۳۷.WAV"

((این نوع موتورها وزنشان زیاد است. کوپه‌ی قطار پر شده است. این حکم در مورد جعل اسناد صدق نمی‌کند.تنها نصف اموال او غصبی است. زیبای خفته نام یک افسانه است. حوض را با ظرف پر از آب کرد. صحاف گفت این کتاب چاپی است نه خطی. قطر این دایره چقدر است؟. حتی خرج دفن اون در وسع آنها نیست. هیچکس طنز را نفی نمی‌کند.))

نام فایلهای صوتی در دادگان فارس دات به صورت ترکیب حرف انگلیسی S و یک شماره است که به این صورت انتخاب می‌شود:

''S'' + <شماره‌ی فصل> + <کد گوینده> + ''.wav''

میزان استفاده‌ی ما از دادگان گفتاری فارس دات برای آزمایش‌های مختلف متفاوت بوده است. همانطور که قبلا بیان شد، فضای ویژگی که در ادامه ایجاد می‌شود باید آنقدر بزرگ باشد تا تمام تنوع‌های آوایی و انواع صداها در زبان فارسی را در بر بگیرد. کل تعداد فایلهای صوتی در دادگان فارس دات حدود ۶۰۸ فایل است. برنامه به کاربر این امکان را می‌دهد تا تعداد فایل‌های مورد نیاز برای فرایند آموزش را انتخاب کند، از ۱ تا ۶۰۸. هرچقدر تعداد فایل‌ها برای آموزش سیستم بیشتر باشد، فضای ویژگی تنوع آوایی بیشتری خواهد داشت؛ ولی در مقابل سرعت آموزش کلاسترها افت خواهد کرد. ما بخش اصلی کار خود را با حدود ۲۰۰ فایل صوتی که به صورت تصادفی از بین ۶۰۸ فایل انتخاب شده و شامل لهجه‌های مختلف هستند، انجام داده‌ایم. در هنگام آموزش کلاسترها باید دقت کنیم که این فایلهای صوتی نباید حاوی مقادیر زیادی سکوت باشند. وجود بخشهای زیاد غیر گفتاری و سکوت در داده‌های آموزشی باعث به وجود آمدن کلاسترهای نامتعادل می‌شود و افزایش خطا در عملکرد کلی سیستم می‌شود. خوشبختانه هنگام بیان جملات توسط گوینده‌ها، مکث قابل توجهی وجود ندارد اما در ابتدا و انتهای هر فایل مقادیری سکوت وجود دارد که البته توسط برنامه‌ی ما حذف می‌شوند. با اعمال این ملاحظات، در حالت کلی ما از ۲۰۰ فایل استفاده کرده‌ایم که به طور میانگین هر کدام محتوی ۳۰ ثانیه گفتار خالص هستند. با انجام یک محاسبه‌ی ساده داریم:

۱۰۰ دقیقه=۶۰۰۰/۶۰ ثانیه ۶۰۰۰ ثانیه= ۲۰۰ * (۳۰ ثانیه)

از ۱۰۰ دقیقه گفتار خالص بدون سکوت که توسط گوینده‌های مختلف بیان شده، برای آموزش سیستم استفاده شده است.

۵-۴-۲ پیش تاکید:

خواص سیگنال گفتار به گونه‌ای است که انرژی آن در فرکانس‌های بالا کم می‌شود. در برخی از کاربردها به دنبال این هستیم تا این افت را جبران کنیم. به عبارت دیگر، سیگنال صوتی اصلی، انرژی بالایی در فرکانس‌های پایین‌تر دارد و پردازش سیگنال برای برجسته‌تر کردن انرژی در فرکانس‌های بالا ضروری است. فرکانس‌های بالا، جایی است که جزئیات سیگنال صوتی در آنجا نهفته شده است. اجرای فیلتر پیش تاکید بر روی یک سیگنال باعث می‌شود که سیگنال از حالت بم بودن خارج شود و به صورت شارپ‌تر و تیز تر(زیر) شنیده شود. به عبارت دیگر جزئیات سیگنال تقویت می‌شوند. یک راه ساده برای پیش تاکید یک سیگنال صوتی، استفاده از یک فیلتر بالاگذر مرتبه‌ی اول است که به این صورت داده می‌شود: $H(z)=1-\alpha Z^{-1}$ $S_2[n]=S[n]-\alpha S[n-1]$

که در آن $S[n]$ سیگنال ورودی و $S_2[n]$، سیگنال خروجی پیش تاکید شده است. α نیز پارامتر تنظیم است. هرچقدر α به ۱ نزدیک‌تر باشد، روی فرکانس‌های بالا تاکید بیشتری می‌شود. مقدار α معمولا بین ۰.۹ تا ۱ است که در آزمایشات ما مقدار ۰.۹۷ برای α در نظر گرفته شده است.

۵-۴-۳ استخراج ویژگی:

همانند حالت گسسته، برای استخراج ویژگی‌های صوتی، از ماژول نرم افزاری Hcopy.exe واقع در بسته‌ی نرم افزاری HTK استفاده شده است. به دلیل کارایی بهتر ویژگی‌های MFCC نسبت به سایر ویژگی‌ها، ازضرایب MFCC در این تحقیق استفاده شده است. نحوه‌ی انتخاب ضرایب به این صورت است:

۳۹ ضریب = ۱۳ MFCCs + ۱۳ Δ-MFCCs + ۱۳ Δ^2-MFCCs

استفاده از فریم‌هایی با طول ۲۵ میلی ثانیه به صورت پنجره‌ی همینگ با همپوشانی ۵۰ درصد و استفاده از ۲۲ فیلتربانک هم دیگر مشخصه‌های قابل ملاحظه در ویژگی‌های استخراجی می‌باشد.

۵-۴-۴ کوانتیزاسیون:

از هر کدام از فایل‌های صوتی، یک فایل حاوی ضرایب MFCC استخراج می‌شود. سپس کل فایل‌های MFCC به محیط نرم افزار متلب فراخوانی شده و در یک متغیر قرار می‌گیرند. پس از فراخوانی تمام فایل‌ها، کل ویژگیهای استخراجی، در فضای ویژگی گسترده می‌شوند. سپس با استفاده از الگوریتم K-means، این فضای ویژگی به K کلاستر کوانتیزه می‌شود. تعداد کلاسترها توسط کاربر قابل تنظیم است. برای بررسی اثر تعداد کلاسترها روی میزان تشخیص، آزمایشاتی برای مقادیر مختلف K انجام شده است. بعد از اتمام کوانتیزاسیون، به هر کلاستر یک اندیس اختصاص داده می‌شود که منجر به تشکیل یک کتاب کد با K اندیس می‌گردد. به منظور سرعت بخشیدن به محاسبات بعدی، یک ماتریس فاصله از ویژگی‌ها در کتاب کد از قبل محاسبه می‌شود. ماتریس فاصله، یک ماتریس مربعی با ابعاد K * K است. برای مثال فرض کنید که بردار ویژگی‌ها از مرتبه‌ی ۳ را از سیگنال گفتار استخراج کرده‌ایم و این بردارها را به ۱۰ کلاستر کوانتیزه کرده‌ایم(K=۱۰). در نتیجه ۱۰ مرکز با ابعاد (۱۰*۳) برای این ۱۰ کلاستر به وجود می‌آید:

جدول ۵-۱ نمونه‌ی مراکز کلاسترها (k=۱۰)

C۱	C۲	C۳	C۴	C۵	C۶	C۷	C۸	C۹	C۱۰
۱	۵	۷	۹	۲۰	۱۲	۲۵	۸	۹	**۱۷**
۵	۳	۶	۴	۹	۱	۲	۳	۹	**۷**
۲	۱	۶	۲	۵	۱	۳	۴	۶	**۴**

اختصاص یک اندیس به هر مرکز منجر به ایجاد یک کتاب کد با اندازه‌ی ۱۰ خواهد شد (بگویید : C۱-C۱۰). ماتریس فاصله حاوی فاصله‌ی بین هرکدام از این مراکز خواهد بود. بنابراین ماتریس فاصله یک ماتریس ۱۰ * ۱۰ است که در جدول۵-۲ نشان داده شده است:

$D[۱][۱]=|۱-۱|^۲ + |۵-۵|^۲ + |۲-۲|^۲ = ۰$

$D[۱][۲]=|۱-۵|^۲ + |۵-۳|^۲ + |۲-۱|^۲ = ۲۱ = D[۲][۱]$

$D[۱][۳]=|۱-۷|^۲ + |۵-۶|^۲ + |۲-۶|^۲ = ۵۳ = D[۳][۱]$

در زمانی که عملیات تشخیص کلمات کلیدی انجام می‌شود، سیستم به طور مداوم باید فاصله‌ی بین مراکز کلاسترها را دو به دو حساب کند. از آنجا که بردار ویژگی‌های MFCC دارای ۳۹ بعد است، این کار فشار محاسباتی زیادی را به سیستم تحمیل می‌کند. به عنوان مثال فرض کنید که فاصله‌ی بین مراکز کلاسترهای ۵ و ۹ مورد نیاز باشد. راه پر هزینه این است که ما دو بردار ۳۹ بعدی را از هم کم کنیم و به توان ۲ برسانیم و سپس عناصر آن را با هم جمع کنیم. اما با داشتن ماتریس فاصله فقط کافی است به آرایه‌ی ۵و۹ رجوع کنیم.

جدول ۵-۲ ماتریس فاصله برای کلاسترهای نشان داده شده در جدول ۵-۱

۰	۲۱	۵۳	۶۵	۳۸۶	۱۳۸	۵۸۶	۵۷	۹۶	۶۴
۲۱	۰	۳۸	۱۸	۲۷۷	۵۳	۴۰۵	۱۸	۷۷	۱۶۹
۵۳	۳۸	۰	۲۴	۱۷۹	۷۵	۳۴۹	۱۴	۱۳	۱۰۵
۶۵	۱۸	۲۴	۰	۱۵۵	۱۹	۲۶۱	۶	۴۱	۷۷
۳۸۶	۲۷۷	۱۷۹	۱۵۵	۰	۱۴۴	۷۸	۱۸۱	۱۲۲	۱۴
۱۳۸	۵۳	۷۵	۱۹	۱۴۴	۰	۱۷۴	۲۹	۹۸	۷۰
۵۸۶	۴۰۵	۳۴۹	۲۶۱	۷۸	۱۷۴	۰	۲۹۱	۳۱۴	۹۰
۵۷	۱۸	۱۴	۶	۱۸۱	۲۹	۲۹۱	۰	۴۱	۹۷

۹۶	۷۷	۱۳	۴۱	۱۲۲	۹۸	۳۱۴	۴۱	۰	۷۲
۲۶۴	۱۶۹	۱۰۵	۷۷	۱۴	۷۰	۹۰	۹۷	۷۲	۰

۵-۴-۵ پردازش کلمات کلیدی(تبدیل به کتاب کد):

به مجرد اینکه مرکز هر کلاستر از داده‌های آموزشی بدست آمد، بردار ویژگی‌ها برای هر فریم از کلمه‌ی کلیدی، به یکی از این مراکز که کمترین فاصله را با آن دارد کوانتیزه می‌شود. سپس اندیس آن کلاستر (مثلا کلاستر ۸) به عنوان نماینده‌ی آن فریم از کلمه‌ی کلیدی معرفی می‌شود. اگر این کار را برای تمام فریم‌ها از کلمه‌ی کلیدی انجام دهیم، در نهایت یک رشته از اعداد را خواهیم داشت که نمایانگر آن کلمه کلیدی است. به عنوان مثال:

۵ : کلمه کلیدی ۳ ۲ ۱ ۶

هر کدام از این اعداد اندیس آن کلاستری هستند که بردار ویژگی هر کدام از فریم‌های کلمه‌ی کلیدی کمترین فاصله را با آن داشته است. همانطور که قبلا اشاره شد، به این نوع نمایش، نمایش **کتاب کد** گفته می‌شود. البته طول کتاب کد برای یک کلمه‌ی کلیدی خیلی بیشتر از نمونه‌ی نشان داده است. در اینجا برای سادگی، طول کتاب کد را ۵ در نظرگرفته‌ایم . به عنوان مثال کلمه‌ی کلیدی ((اقتصاد)) در یکی از آزمایشات دارای طول کتاب کد ۸۰ است .هر چه طول زمانی کلمه بیشتر باشد، کتاب کد هم بزرگتر است.

۵-۴-۶ پردازش جملات (تبدیل به کتاب کد):

دقیقا همان فرایندی که برای تبدیل یک کلمه‌ی کلیدی به کتاب کد انجام شد، برای گفتار اصلی که جستجو در آن صورت می‌گیرد نیز تکرار می‌شود. از آنجا که طول جملات معمولا زیاد است، یک کتاب کد طولانی نیز برای هر جمله حاصل خواهد شد:

جمله : ۳ ۲ ۵ ۲ ۷ ۹ ۹ ۲ ۳ ۲ ۵ ۲ ۷ ۹ ۹ ۲ ۳ ۲ ۵ ۲ ۷ ۹.....

۵-۴-۷ فرایند تشخیص(بررسی احتمال حضورکلمه کلیدی):

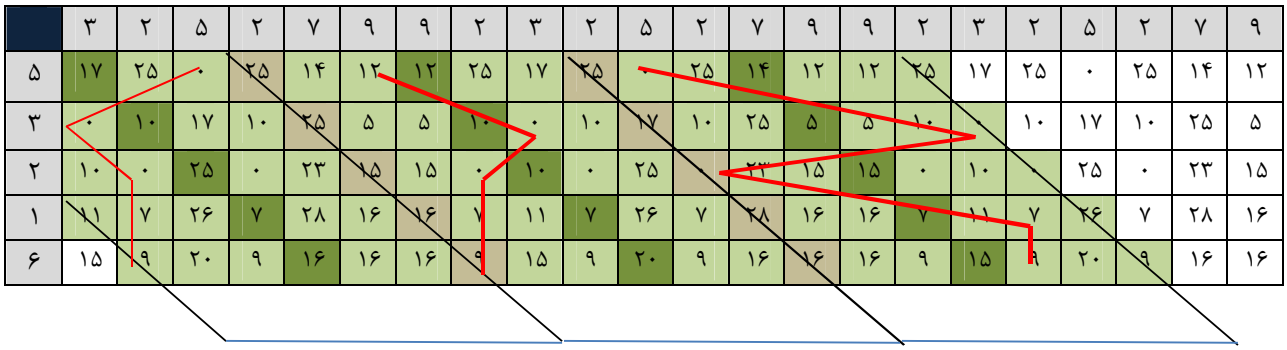
برای اینکه احتمال حضور یک کلمه‌ی کلیدی را در گفتار بررسی کنیم، باید گفتار اصلی ما به بخش‌هایی که با همدیگر دارای همپوشانی هستند تقسیم شود. طریقه‌ی این کار در شکل ۵-۸ توضیح داده شده است. هر کدام از بخش‌ها در ادامه با کلمه‌ی کلیدی که در حال جستجوی آن هستیم، مقایسه می‌شود. به منظور لحاظ تغییرات احتمالی در کلمه‌ی مورد جستجو، بجای استفاده از فاصله‌ی خطی از معیار فاصله‌ی (DTW) بین هر بخش از جمله و کلمه کلیدی استفاده شده است. بعد از بدست آوردن کتاب کد برای کلمه‌ی کلیدی و گفتار اصلی، آنها را در دو طرف یک شبکه(grid) قرار می‌دهیم. به عنوان مثال برای کتاب کدی که قبلا محاسبه کردیم، در شکل ۵-۸ به صورت تصویری نحوه‌ی انجام این کار مشخص شده است. مقادیر اعدادی که در ماتریس زیر وجود دارند از ماتریس فاصله بدست آمده‌اند.

	۳	۲	۵	۲	۷	۹	۹	۲	۳	۲	۵	۲	۷	۹	۹	۲	۳	۲	۵	۲	۷	۹
۵	۱۷	۲۵	۰	۲۵	۱۴	۱۲	۱۲	۲۵	۱۷	۲۵	۰	۲۵	۱۴	۱۲	۱۲	۲۵	۱۷	۲۵	۰	۲۵	۱۴	۱۲
۳	۰	۱۰	۱۷	۱۰	۲۵	۵	۵	۱۰	۰	۱۰	۱۲	۱۰	۲۵	۵	۵	۱۰	۰	۱۰	۱۷	۱۰	۲۵	۵
۲	۱۰	۰	۲۵	۰	۲۳	۱۵	۱۵	۰	۱۰	۰	۲۵	۰	۲۳	۱۵	۱۵	۰	۱۰	۰	۲۵	۰	۲۳	۱۵
۱	۱۱	۷	۲۶	۷	۲۸	۱۶	۱۶	۷	۱۱	۷	۲۶	۷	۲۸	۱۶	۱۶	۷	۱۱	۷	۲۶	۷	۲۸	۱۶
۶	۱۵	۹	۲۰	۹	۱۶	۱۶	۱۶	۹	۱۵	۹	۲۰	۹	۱۶	۱۶	۱۶	۹	۱۵	۹	۲۰	۹	۱۶	۱۶

قطعه‌ی ۱ قطعه‌ی ۲ قطعه‌ی ۳

شکل ۵-۸ فاصله بر مبنای فریم، بین یک کلمه کلیدی و گفتار پنجره گذاری شده. قسمتی که به صورت تیره‌تر نمایش داده شده است، مرکز هرکدام از بخش‌ها را نشان می‌دهد. بالاترین ردیف، کتاب کد جمله اصلی و اولین ردیف سمت چپ، کتاب کد کلمه‌ی کلیدی را نشان می‌دهد.

بلوک‌های تاریک‌تر مرکز هر قطعه را نشان می‌دهند. با توجه به شکل، می‌توان دید که عرض هر بخش در اینجا برابر ۷ واحد است. برای مقایسه‌ی احتمال حضور هر کلمه کلیدی، فاصله‌ی مسیر با کمترین اعوجاج محاسبه شده و بر طول کتاب کد کلمه‌ی کلیدی تقسیم می‌شود.



شکل ۵-۹ برای هر بخش، مسیر با کمترین فاصله با خط قرمز مشخص شده است. فاصله‌ی تجمعی کل، برابر با مجموع فاصله‌هایی است که در هر مسیر وجود دارند تقسیم بر طول کتاب کد کلمه‌ی کلیدی.

امتیاز اعوجاج در بخش اول: (۰ + ۰ + ۰ + ۷ + ۹) /۵=۲.۴

امتیاز اعوجاج در بخش دوم: (۱۲ + ۰ + ۰ + ۷ + ۹) /۵=۵.۶

هر قطعه، یک امتیاز شباهت دارد. قطعاتی که بیشترین امتیاز شباهت را دارند(کمترین امتیاز اعوجاج را دارند)، کاندیدای حضور کلمه‌ی کلیدی در آن نقطه می‌شوند. اگر از هر کلمه‌ی کلیدی چندین نمونه در اختیار داریم، بهترین راه برای بهره گیری از آنها، این است که از الگوریتم رای گیری استفاده کنیم. همانطور که قبلا بیان شد، الگوریتم رای گیری به این شکل است که اگر ما آزمایش تشخیص را برای N نمونه‌ی موجود از یک کلمه‌ی کلیدی خاص را انجام دهیم، اگر نصف این تعداد یعنی N/۲ یک قطعه را به عنوان کاندید حضور کلمه کلیدی معرفی کنند، آنگاه آن قطعه محل واقعی حضور کلمه‌ی کلیدی خواهد بود. البته در طول انجام آزمایش‌ها به این نتیجه رسیدیم که برخی کلمات کلیدی در چند فریم قبل و یا بعد از محل اصلی خود شناسایی می‌شوند. اگر این خطا

کم باشد، می‌توان از آن چشم پوشی کرد. ما در آزمایش‌های خود، اختلاف ۵ واحد قبل و بعد از محل اصلی را قابل اغماض فرض کرده‌ایم. یعنی اگر محل اصلی کلمه کلیدی در سگمنت ۳۳۰ باشد، آنگاه تشخیص در سگمنت‌های (۳۲۵ تا ۳۳۵) نیز به عنوان تشخیص صحیح در نظر گرفته می‌شود.

در انتها لازم است تا محل وقوع کلمه‌ی کلیدی را بر حسب زمان بیان کنیم، بدین منظور لازم است که طول زمانی فایل اصلی که در آن جستجو انجام داده‌ایم را داشته باشیم. برای مثال اگر طول فایل بر حسب زمان برابر با ۳۱ ثانیه، تعداد کل سگمنت‌ها برابر ۱۵۰۰ سگمنت و محل وقوع کلمه کلیدی در سگمنت شماره‌ی ۶۵۰ باشد، آنگاه :

$$\text{محل وقوع کلمه‌ی کلیدی} = (۶۵۰)/۱۵۰۰ * ۳۱ = ۱۳.۴۳s$$

بنابراین کلمه‌ی کلیدی مورد نظر در ثانیه ۱۳.۴۳ بیان شده است.

فصل ششم

نتایج و ارزیابی

۶-۱ نتایج:

در ایـن قسـمت بـه بررسـی آزمـایش‌هـای اولیـه انجـام شـده روی الگـوریتم مـورد بحـث در فصـل پـنجم خـواهیم پرداخـت. در ادامـه، تعـدادی از آزمایشـات بـه ازای کلمـات کلیـدی مختلـف، تعـداد کلاسـترهـای مختلف و طـول متفـاوت پنجـره‌ی DTW، انجـام و نتـایج آنهـا آورده شـده اسـت. بـه علـت وجـود محـدودیت در دسترسـی بـه دادگـان گفتـار فارسـی، محـدودیت‌هـایی در انتخـاب کلمـات داشـته‌ایـم. معمـولا از کلمـاتی استفاده شده است که حداقل ۸ نمونه از آنها موجود باشد.

کلمه‌ی کلیدی: **لوکوموتیو**

تعداد نمونه‌های موجود: ۱۴ نمونه

اندازه‌ی پنجره (R):۳

اندازه‌ی کلاستر:۵۱۲

جنسیت گوینده‌ی جمله: مذکر

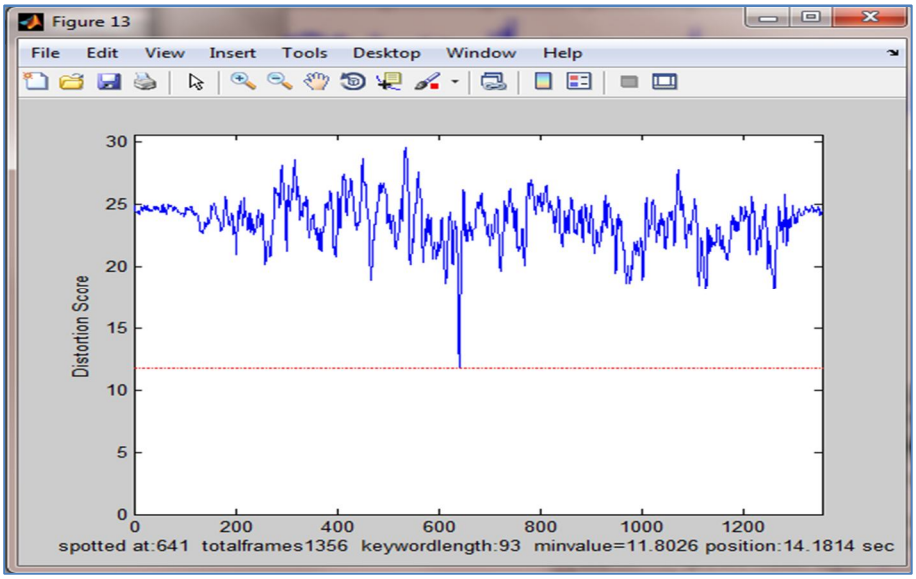
محل حضور کلمه‌ی کلیدی بر اساس شهود مستقیم توسط نرم افزار مدیا پلیر: حدودا ثانیه ۱۴

نام فایل صوتی: S۲۹.wav

طول فایل:۳۰ ثانیه

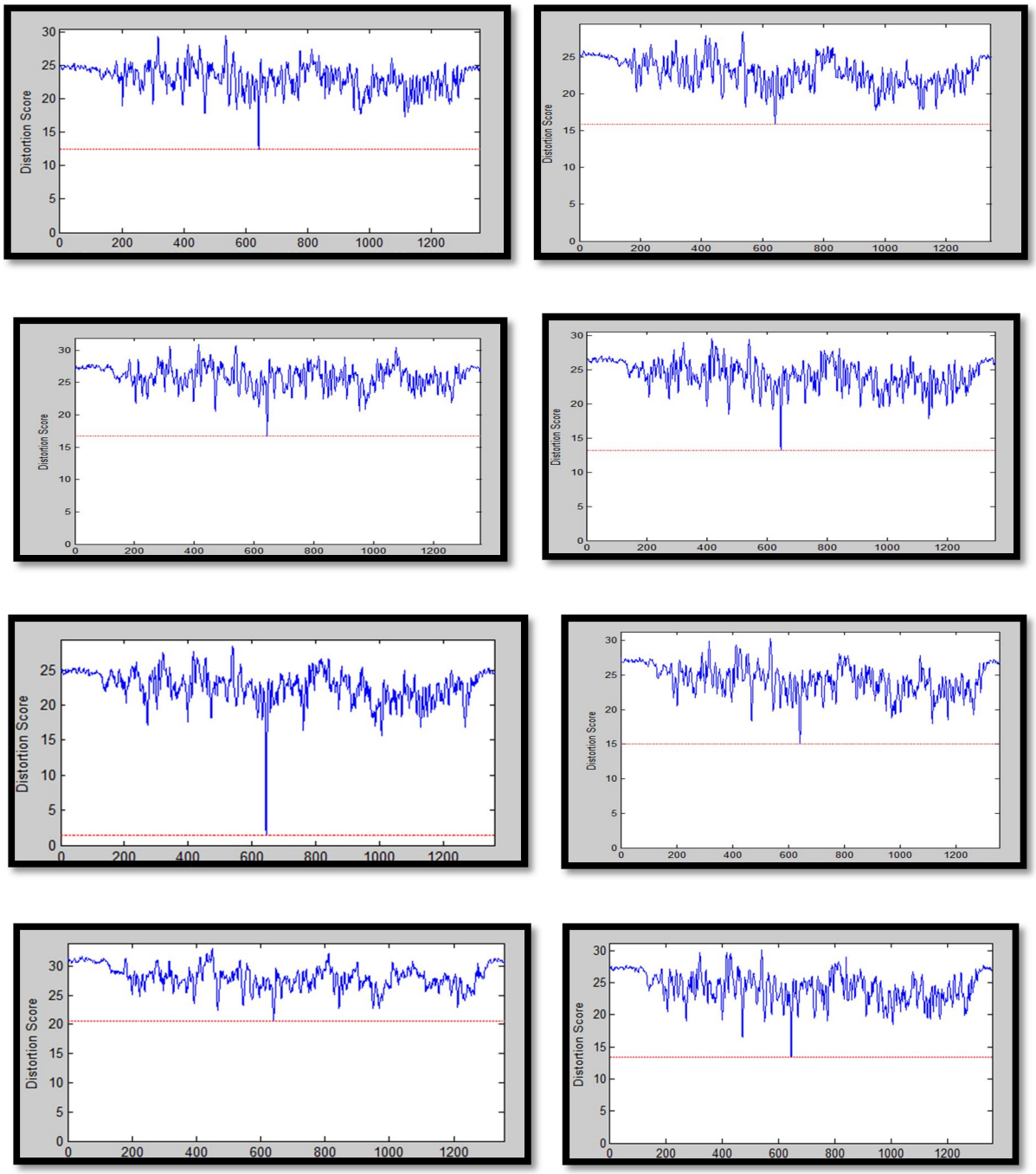
متن فایل صوتی: ((دو رکعت نماز خواند. به چه علت سگ فرار کرد؟.پایش از مچ شکست. رنگ پارچ قرمز است. شیخ ما خیر ما را می‌خواهد. این **لوکوموتیو** راننده ندارد. وای ژاله مچ پایش شکست. از خسرو وحشت ندارم. با روشن شدن هوا، تظاهرکنندگان به سوی مجلس شورای ملی شروع به راهپیمایی کردند. مگر مژده اول چراغ قوه را خاموش نکرد؟.))

شکل ۶-۱ نمایش تصویری تشخیص محل حضور کلمه‌ی کلیدی را نشان می‌دهد. قسمتی از منحنی که دارای کمترین مقدار است، در واقع فریمی را نشان می‌دهد که با کلمه‌ی کلیدی نمونه کمترین اختلاف را دارد. در قسمت پایین شکل اطلاعات مفیدی وجود دارد که در حین فرایند تشخیص بدست می‌آیند.

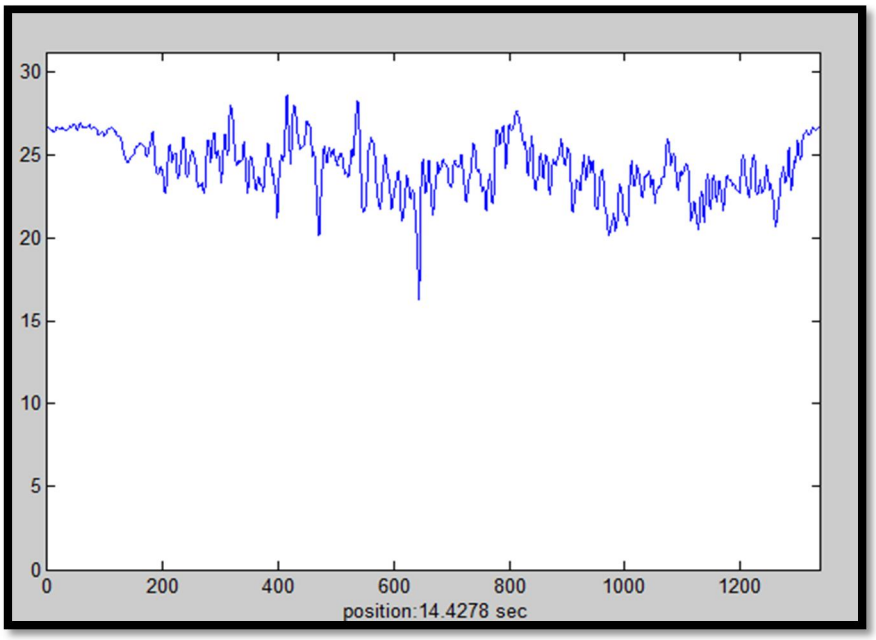


شکل ۶-۱ نمودار تشخیص محل کلمه‌ی کلیدی ((لوکوموتیو)). در قسمت پایین شکل اطلاعاتی وجود دارد که از سمت چپ به راست به ترتیب بیانگر این موارد هستند: شماره فریم نقطه‌ی تشخیص- تعداد کل فریم‌ها - طول کتاب کد کلمه‌ی کلیدی - مقدار کمینه‌ی منحنی و محل تشخیص داده شده‌ی حضور کلمه‌ی کلیدی بر حسب ثانیه

همانطور که در شکل ۶-۱ مشخص است، محل حضور کلمه کلیدی در ثانیه‌ی ۱۴.۱۸۱۴ تشخیص داده شده است که صحیح است. این جمله برای سایر نمونه‌های موجود از این کلمه‌ی کلیدی نیز آزمایش شده است که نتایج آن در ادامه آورده شده است.



شکل ۶-۲ منحنی‌های امتیاز اعوجاج برای تشخیص محل حضور کلمه کلیدی لوکوموتیو. در تمامی شکل‌ها ثانیه ۱۴، محل حضور کلمه‌ی کلیدی تشخیص داده شده است.



شکل ۶-۳ نمودار میانگین امتیاز اعوجاج. از ۱۴ نمودار بدست آمده (در بالا فقط ۹ نمودار آورده شده است) برای کلمه لوکوموتیو میانگین گیری شده است. نقطه‌ای که کمترین امتیاز اعوجاج را دارد، نقطه‌ی حضور کلمه‌ی کلیدی است.

از میان ۱۴ نمونه‌ی موجود، تصویر حاصل از ۹ نمونه در اینجا نشان داده شده است. تمامی ۱۴ نمودار، حوالی ثانیه‌ی ۱۴ را به عنوان نقطه‌ی تشخیصی خود معرفی کرده‌اند. با اینکه این نتیجه کاملا صحیح است اما باید دقت کنیم که ما قبلا از محل حضور کلمه‌ی کلیدی اطلاع داشته‌ایم. اگر به شکلی که آخرین ردیف سمت چپ وجود دارد دقت کنید، متوجه خواهید شد که علیرغم تشخیص صحیح محل کلمه‌ی کلیدی، مقدار دامنه در این نقطه نسبت به سایر نقاط کمی بیشتر است. در سایر شکل‌ها، مقدار دامنه در نقطه‌ی تشخیص حول و حوش عدد ۱۵ است اما در شکل مزبور این مقدار نزدیک به ۲۰ است. در زمانی که تشخیص انجام می‌شود از دو راهکار می‌توان بهره برد. راهکار اول این است که فرض کنیم مانند این مثال چندین نمونه از یک کلمه‌ی کلیدی موجود است. آنگاه اگر تعداد بیشتر از نصف این نمونه‌ها، یک محل را به عنوان کاندیدای حضور کلمه کلیدی معرفی

کنند، آن نقطه به عنوان نقطه‌ی تشخیص معرفی می‌شود. در راهکار دوم، ما باید یک حد آستانه را برای پذیرش و یا رد کردن یک کلمه‌ی کلیدی معرفی کنیم. این حد آستانه برای کلمات کلیدی مختلف متفاوت است. آنگاه اگر نقطه‌ای که به عنوان مینیمم معرفی شده، از حد آستانه کمتر باشد، سیستم به ما هشدار خواهد داد که کلمه‌ی کلیدی یافت شده است؛ و اگر نقطه‌ای که از حد آستانه کمتر باشد پیدا نشود، سیستم به ما هشداری نمی‌دهد. در اینجا دو نوع خطا ممکن است رخ دهد. خطای اول، خطای هشدار اشتباه و یا False Alarm(FA) نامیده می‌شود. این خطا زمانی رخ می‌دهد که در نقطه‌ای که از حد آستانه کمتر بوده و به عنوان محل وقوع کلمه‌ی کلیدی معرفی شده است، هیچ کلمه‌ی کلیدی موجود نباشد. خطای دوم خطای از دست رفتن و یا Miss نامیده می‌شود. در این خطا، با وجود این که کلمه‌ی کلیدی در یک محل وجود دارد، اما مقدار آن از حد آستانه بیشتر است و سیستم آنرا لحاظ نمی‌کند. در آزمایش بعدی، همین فرایند تشخیص روی یک فایل صوتی انجام می‌شود که کلمه‌ی کلیدی مورد نظر در آن وجود ندارد.

کلمه‌ی کلیدی: **شاه عباس**

تعداد نمونه‌های موجود: ۱۴ نمونه

اندازه‌ی پنجره (R):۳

اندازه‌ی کلاستر:۵۱۲

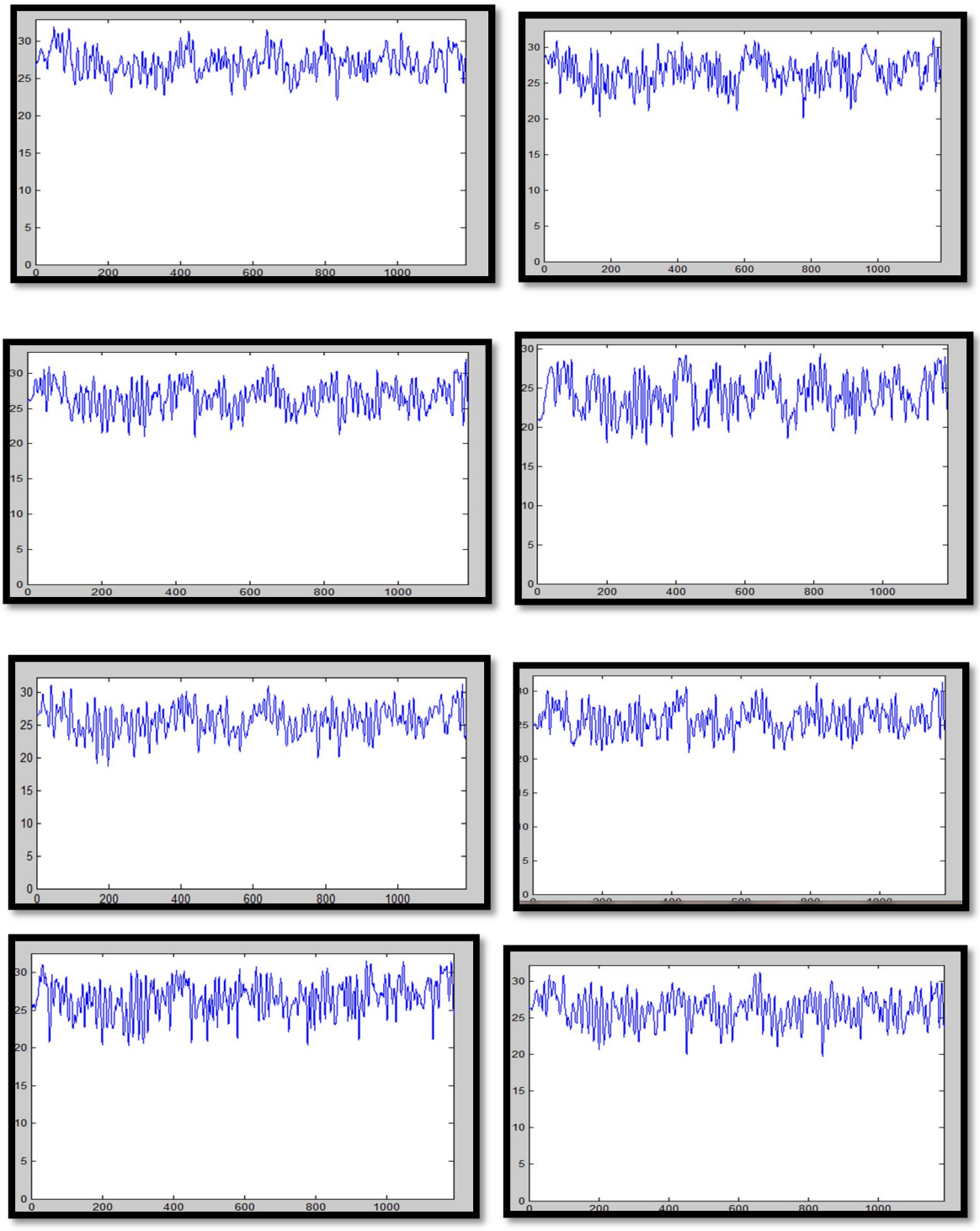
جنسیت گوینده‌ی جمله: مونث

محل حضور کلمه‌ی کلیدی بر اساس شهود مستقیم توسط نرم افزار مدیا پلیر: کلمه کلیدی وجود ندارد

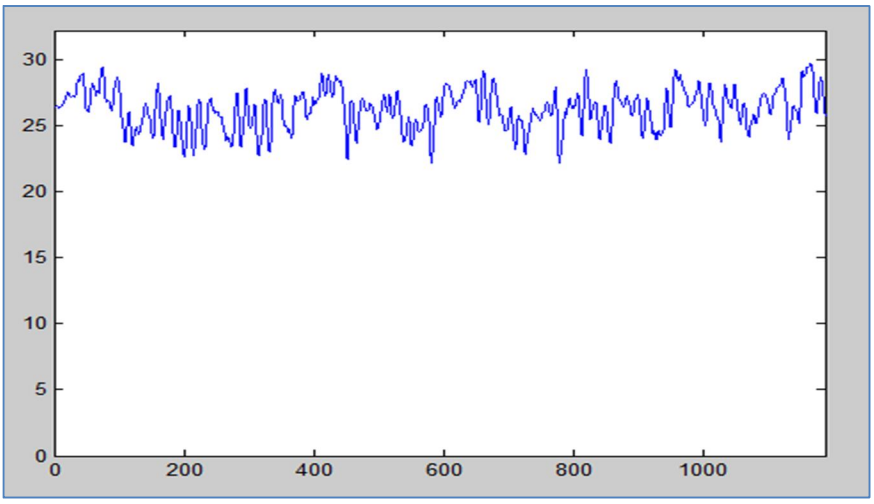
نام فایل صوتی: S۱۸۰.wav

طول فایل: ۲۶ ثانیه

متن فایل صوتی: ((زیبای خفته نام یک افسانه است. حوض را با ظرف پر از آب کرد. صحاف گفت این کتاب چاپی است نه خطی. قطر این دایره چقدر است؟حتی خرج دفن او در وسع آنها نیست. هیچکس طنز را نفی نمی‌کند. ترشی‌ها را گذاشتم کنار رادیو. مگر سرتیپ در پادگان نیست؟ فقط یک ربع کره‌ی زمین مسکونی است. شعرت را بنویس روی این برگه!.))



شکل ۶-۴ هشت نمودار حاصل از آزمایش بررسی حضور کلمه‌ی کلیدی شاه عباس در یک فایل صوتی. همانطور که در شکل‌ها مشخص است، هیچ نقطه‌ی مینیمم برجسته‌ای در نمودارها که بیانگر حضور کلمه کلیدی باشد وجود ندارد.



شکل ۶-۵ نمودار میانگین حاصل از ۱۴ نمودار در آزمون کلمه‌ی کلیدی شاه عباس. هیچ نقطه‌ی مینیمم برجسته‌ی مجزایی در شکل وجود ندارد.

۶-۲ ضوابط تصمیم گیری:

نمودار میانگین اعوجاج، کمترین مقدار اعوجاج در محلی که کلمه‌ی کلیدی قرار دارد را نشان می‌دهد. هرچند که کمترین میزان اعوجاج همیشه از روی نمودار قابل تشخیص نیست. علاوه بر این، میانگین امتیاز اعوجاج، به یک مقدار آستانه برای تصمیم گیری نیز نیازمند است. برای جلوگیری از این مشکل، الگوریتم رای گیری که در قسمت‌های قبل توضیح داده شد استفاده شده است. اگر بیشتر از نیمی از کلمات کلیدی موجود، یک نقطه را به عنوان محل حضور کلمه‌ی کلیدی تشخیص دهند، آن نقطه به عنوان نقطه‌ی تشخیص معرفی می‌شود.

۶-۳ مشخصات عملکردی سیستم:

پارامترهای مختلفی هستند که تنظیمات عملکردی سیستم را تغییر می‌دهند. در ادامه این پارامترها مورد بحث و بررسی قرار می‌گیرند:

۶-۳-۱ تعداد کلمات کلیدی:

این پارامتر که با N نشان داده می‌شود، برابر با تعداد کل نمونه‌های موجود از یک کلمه‌ی کلیدی خاص است.

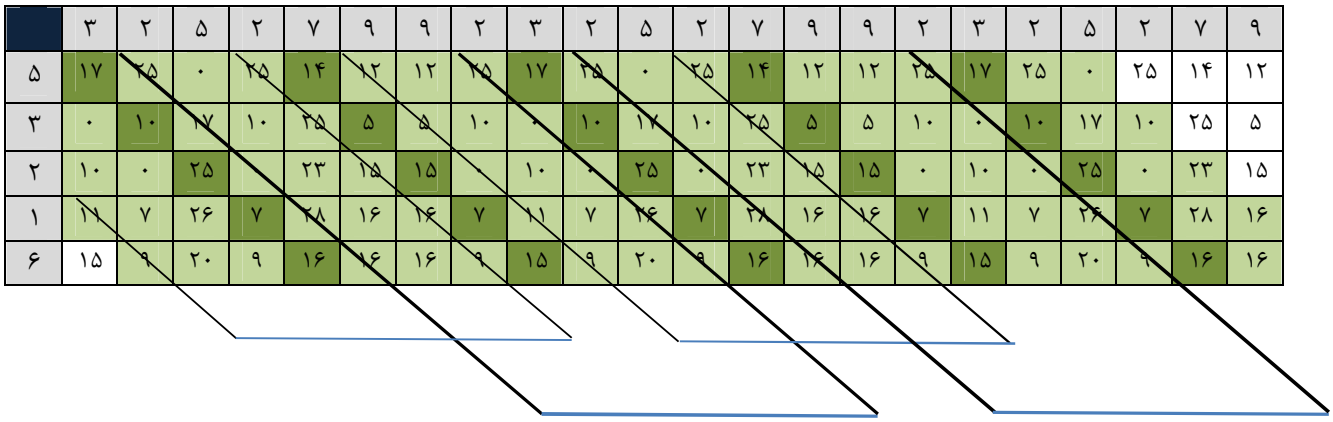
۶-۳-۲ اندازه‌ی پنجره:

در زمانی که از الگوریتم Segmental-Dynamic time Warping استفاده می‌کنیم، گفتار اصلی ما به بخش‌هایی دارای همپوشانی تقسیم می‌شود و هر بخش با کلمه‌ی کلیدی مورد نظر ما مقایسه می‌شود. برای اینکه گفتار را به صورت بخش بخش درآوریم، باید عرض این پنجره در ادامه مشخص شود. برای این کار باید یک نقطه را به عنوان نقطه‌ی شروع در نظر گرفته و به اندازه‌ی نصف طول پنجره در دو طرف این نقطه به عنوان حاشیه در نظر بگیریم. نقطه‌ی واقعی شروع کلمه‌ی کلیدی ممکن است درون عرض یک پنجره‌ی خاص واقع شود.

ما عرض پنجره را به صورت تابعی از R در نظر می‌گیریم. رابطه‌ی R و عرض پنجره به این شکل است:

عرض پنجره = ۲ * R + ۱

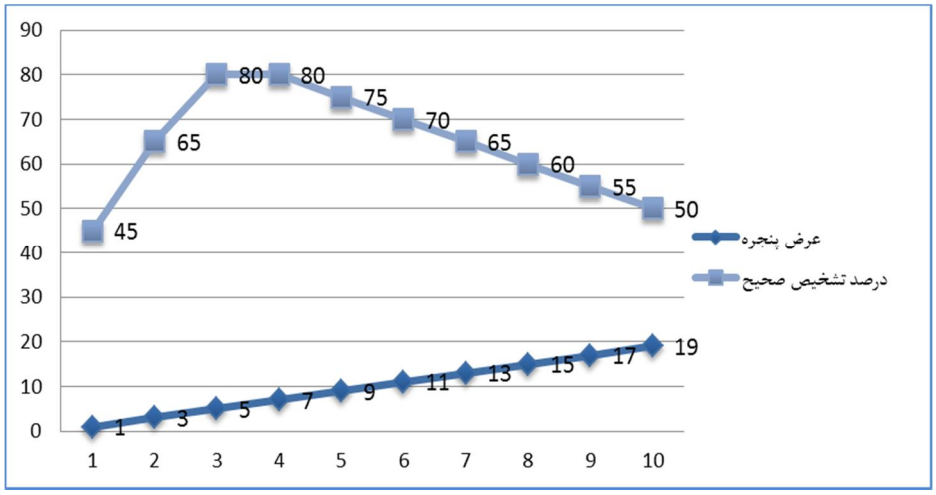
همانطور که در شکل ۶-۶ دیده می‌شود، قرار دادن R=۳ بدین معنا است که از مرکز هر پنجره به اندازه‌ی ۳ واحد در طرفین حاشیه وجود دارد. در نتیجه عرض پنجره برابر با W=۲ * ۳ + ۱=۷ است. مرکز پنجره‌ی بعدی به اندازه‌ی R + ۱ واحد از مرکز پنجره‌ی فعلی فاصله دارد. پس در نتیجه یک همپوشانی به اندازه‌ی R بین هر پنجره و پنجره‌ی قبل و بعد از آن وجود دارد.



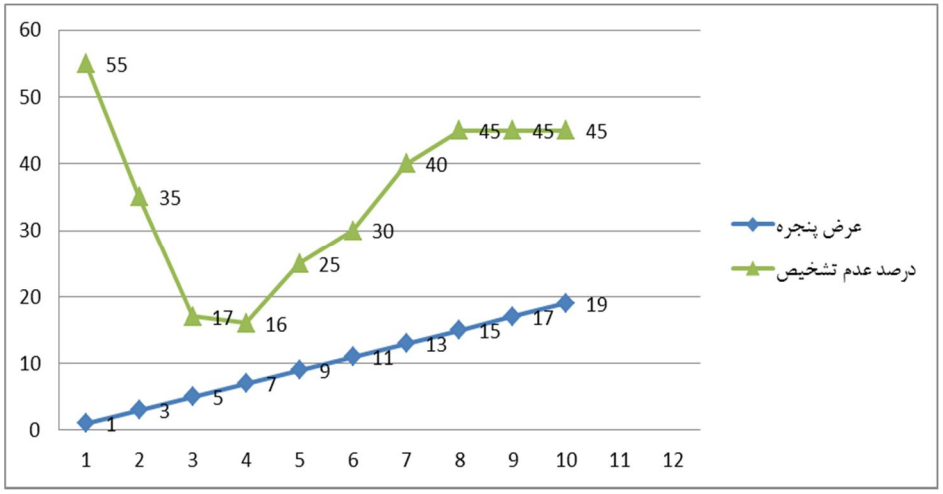
شکل ۶-۶ پنجره گذاری ماتریس فاصله‌ی گفتار و کلمه‌ی کلیدی برای R=۳. عرض پنجره برابر با W=۲ * R +۱=۷ است.

همانطور که در شکل مشخص است، بخش‌های ابتدایی و انتهایی که بدون رنگ هستند را در محاسبات دخالت نمی‌دهیم، چون آنقدر بزرگ نیستند که بتوانند یک کلمه‌ی کلیدی باشند.

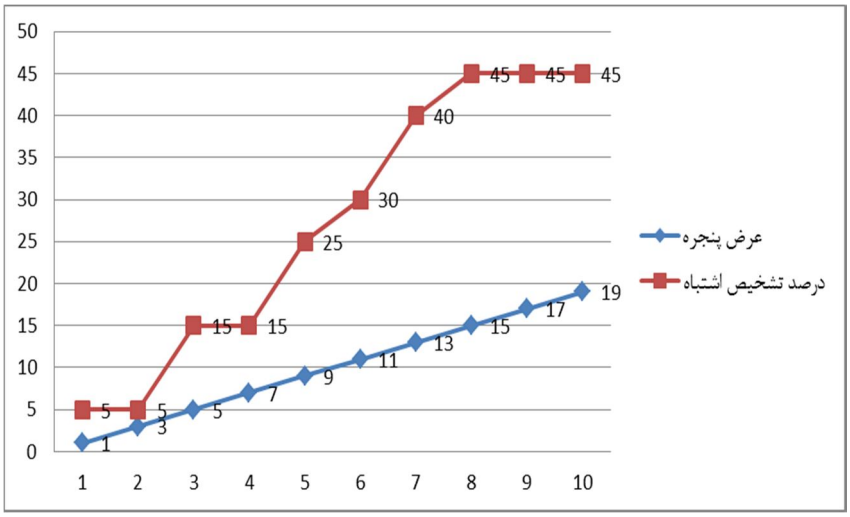
طول پنجره علاوه بر کنترل دقت سیستم، روی سرعت محاسبات سیستم نیز اثرگذار است. در ادامه آزمایشی برای تشخیص دقت عملکرد سیستم برای مقادیر مختلف R انجام شده است. در اینجا از ۵ کلمه‌ی کلیدی مختلف و مقادیر متغیر R برای بررسی پاسخ سیستم استفاده شده است.



شکل ۶-۷ درصد تشخیص کلمات کلیدی بر حسب مقادیر مختلف R



شکل ۶-۸ درصد عدم تشخیص کلمات کلیدی بر حسب مقادیر مختلف R

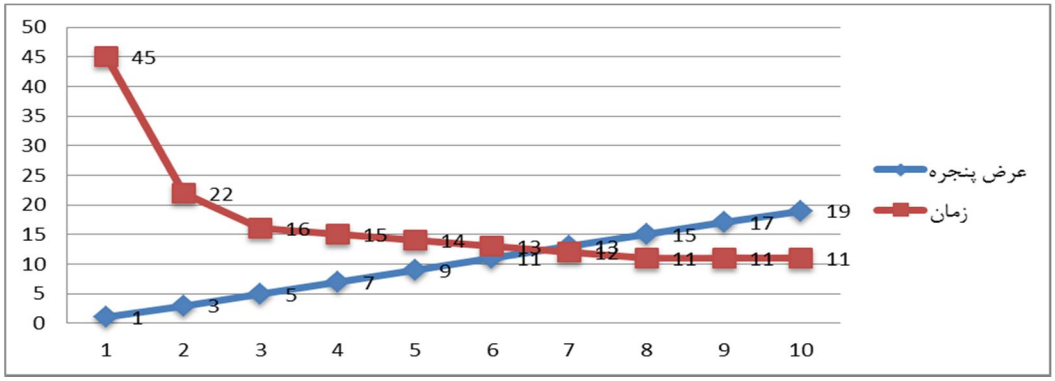


شکل ۶-۹ درصد تشخیص اشتباه کلمات کلیدی بر حسب مقادیر مختلف R

همانطور که در نمودارهای بالا دیده می‌شود، مقدار عرض پنجره تاثیر زیادی در عملکرد و دقت سیستم دارد. هرچه عرض پنجره بیشتر شود، انعطاف سیستم بیشتر می‌شود و اجازه‌ی تغییرات بیشتری را در هنگام همترازی به برنامه می‌دهد. طبیعی است که زمانی که عرض پنجره کمتر باشد، سخت گیری سیستم در انجام مقایسه بیشتر می‌شود و فاصله‌ی اکثر فریم‌ها با کلمه‌ی کلیدی در هنگام مقایسه مقدار بیشتری بدست خواهد آمد. این مساله باعث افزایش نرخ عدم تشخیص(Miss) می‌شود. عکس این قضیه برای نرخ هشدار یا تشخیص اشتباه(False Alaram) برقرار است. زمانی که عرض بیشتری به پنجره داده شود، آنگاه انعطاف بیشتری در هنگام مقایسه با کلمه‌ی کلیدی وجود دارد و ممکن است فریم‌های بیشتری به عنوان کاندیدای محل حضور کلمه‌ی کلیدی معرفی شوند.

پارامتر مهمی دیگری که تحت تاثیر عرض پنجره قرار می‌گیرد، زمان است. زمان تاثیر مهمی در کارکرد سیستم دارد. اگر جستجویی که انجام می‌دهیم مستلزم صرف زمان زیاد باشد، آن جستجو

بهینه نخواهد بود. منحنی‌ای که در شکل ۶-۱۰ ملاحظه می‌کنید، منحنی زمان جستجو نسبت به عرض پنجره است. این آزمایش برای حدود ۱۴ نمونه از یک کلمه‌ی کلیدی و جمله‌ای به طول ۳۰ ثانیه انجام شده است.همانطور که در شکل نشان داده شده است، با افزایش عرض پنجره، زمان محاسبه نیز کاهش می یابد. این امر بدان دلیل است که هر چه طور پنجره بیشتر شود، تعداد پنجره‌ها کمتر شده و در نتیجه تعداد محاسبات نیز کاهش می‌یابد.



شکل ۶-۱۰ رابطه‌ی عرض پنجره با زمان

۶-۴ ارزیابی سیستم:

با استفاده از مواردی که از مباحث انجام شده، استخراج می‌شود، آزمایشاتی به صورت جامع و با استفاده از دادگان گفتاری فارس دات انجام شده است. در این آزمایشات از ۱۴ کلمه‌ی کلیدی مختلف که دارای طول و تنوع آوایی متفاوت هستند، استفاده شده است. تمامی آزمایشات بر روی دادگان گفتاری فارس دات، انجام شده‌اند. این نتایج آن در جدول ۶-۱ آورده شده است.

جدول ۶-۱ نتایج تشخیص کلمات کلیدی مختلف برای کلاستر با اندازه‌ی ۵۱۲ و عرض پنجره ۵

کلمه کلیدی	تعداد نمونه موجود	تعداد تشخیص صحیح (Hit)	تعداد هشدار اشتباه (False Alarm)	تعداد عدم تشخیص (Miss)	میانگین درصد وزنی تشخیص صحیح
شاه عباس	۱۴	۹	۰	۱	٪۶۱
اژدر انداز	۱۳	۹	۰	۱	٪۷۰
چمدان	۱۵	۸	۱	۱	٪۸۰
اقتصادی	۱۴	۱۰	۰	۰	٪۸۷
فانوس	۱۲	۱۰	۰	۰	٪۷۵
جوانان	۱۶	۵	۲	۳	٪۵۵
خشمگین	۱۲	۹	۰	۱	٪۹۱
لوکوموتیو	۱۴	۱۰	۰	۰	٪۹۵
پلنگ	۱۴	۵	۲	۳	٪۶۰
پیچگوشتی	۱۴	۱۰	۰	۰	٪۹۷
راهپیمایی	۱۴	۸	۰	۲	٪۶۵
تظاهرکنندگان	۱۴	۱۰	۰	۰	٪۸۱
آهنگری	۱۴	۷	۱	۲	٪۷۷
فوتبال	۱۳	۱۰	۰	۰	٪۷۶
میانگین کل	۱۳.۷۸	۸.۵	۰.۴۲	۱	٪۷۶.۴۲

در این جدول اطلاعات مختلفی از جمله تعداد نمونه‌های موجود از هر کلمه‌ی کلیدی، میزان تشخیص صحیح، تعداد هشدار اشتباه و تعداد عدم تشخیص کلمه‌ی کلیدی، آورده شده است. هر کدام از کلمات موجود در جدول، بر روی ۱۰ جمله‌ی مختلف تست شده‌اند. ستون تعداد تشخیص صحیح، بیانگر این است که در ۱۰ جمله‌ی تست، در چند جمله کلمات کلیدی به درستی شناسایی شده‌اند. ستون هشدار اشتباه، به ما می‌گوید که در چند جمله از ۱۰ جمله، محل حضور کلمه‌ی کلیدی به اشتباه شناسایی شده است. و در نهایت ستون عدم

تشخیص، بیانگر این است که در چند جمله از ۱۰ جمله‌ی تست، کلمه‌ی کلیدی در محلی که حضور داشته است، تشخیص داده نشده است. در آخرین ستون، اطلاعاتی با نام درصد وزنی تشخیص صحیح موجود است. همانطور که قبلا بیان شد، در صورتی که بیش از یک نمونه از هر کلمه‌ی کلیدی موجود باشد، از الگوریتم رای گیری برای تشخیص محل حضور کلمه‌ی کلیدی استفاده می‌کنیم. این الگوریتم بیان می‌کند که در صورتی که نصف تعداد کلمات موجود، یک فریم و اطراف آن را به عنوان محل حضور کلمه‌ی کلیدی معرفی کنند، آن فریم به عنوان کاندیدای تشخیص معرفی خواهد شد. درصد وزنی که ما می‌گوید که در طول این ۱۰ آزمایش که به ازای هر کلمه‌ی کلیدی انجام شده است، به طور میانگین چند درصد نمونه‌ها به درستی یک محل را به عنوان محل حضور کلمه‌ی کلیدی معرفی کرده‌اند. هر چه میانگین درصد وزنی برای یک کلمه‌ی کلیدی بیشتر باشد، تشخیص آن کلمه با قاطعیت بیشتری انجام می‌شود. به عنوان مثال در کلمه‌ی **لوکوموتیو**، میانگین وزنی ۹۵ درصد وجود دارد. این بدان معنی است که ۹۵ درصد نمونه‌های موجود، در تمام جملات، به درستی یک محل را به عنوان محل حضور کلمه‌ی کلیدی معرفی کرده‌اند. اما در کلمه‌ای مانند **جوانان** این میزان ۵۵ درصد است و قاطعیت تشخیص کمتری نسبت به کلمه‌ی لوکوموتیو حاصل شده است.

همان طور که در بخش‌های گذشته بیان شد، دقت سیستم به این صورت تعریف می‌شود:

$$\text{دقت} = \frac{hits}{hits+misses+false\ alarms}$$

با استفاده از اطلاعات جدول ۶-۱ و رابطه‌ی مزبور، دقت نهایی سیستم در تشخیص ۱۴ کلمه‌ی کلیدی مختلف در جدول ۶-۲ آورده شده است.

جدول ۶-۲ عملکرد سیستم بر اساس دقت تشخیص. از اطلاعات جدول می‌توان دریافت که درحالت کلی کلماتی که طول بیشتری دارند، دقت تشخیص بالاتری نیز دارند.

کلمه کلیدی	تعداد نمونه موجود	میانگین درصد وزنی تشخیص صحیح	دقت تشخیص
شاه عباس	۱۴	۶۱٪	۹۰٪
اژدر انداز	۱۳	۷۰٪	۹۰٪
چمدان	۱۵	۸۰٪	۸۰٪
اقتصادی	۱۴	۸۷٪	۱۰۰٪
فانوس	۱۲	۷۵٪	۱۰۰٪
جوانان	۱۶	۵۵٪	۵۰٪
خشمگین	۱۲	۹۱٪	۹۰٪
لوکوموتیو	۱۴	۹۵٪	۱۰۰٪
پلنگ	۱۴	۶۰٪	۵۰٪
پیچگوشتی	۱۴	۹۷٪	۱۰۰٪
راهپیمایی	۱۴	۶۵٪	۸۰٪
تظاهرکنندگان	۱۴	۸۱٪	۱۰۰٪
آهنگری	۱۴	۷۷٪	۷۰٪
فوتبال	۱۳	۷۶٪	۱۰۰٪
میانگین کل	۱۳.۷۸	۷۶.۴۲٪	۹۲٪

با توجه به جدول ۶-۲ مشخص می‌شود که دقت تشخیص برای کلمات کلیدی مختلف در زبان فارسی متفاوت می‌باشد. بخشی از این تفاوت در دقت، مربوط به خصوصیات ذاتی کلمات و بخشی از آن مربوط به نحوه‌ی بیان آنها توسط گوینده‌ها و همچنین هویت گوینده می‌باشد. اگرچه در الگوریتم پیشنهادی در این پایان نامه سعی

شده است که میزان تاثیر اعوجاج و نحوه‌ی بیان، به حداقل برسد، اما باز هم نمی‌توان از اثر آن چشم پوشی کرد. در ادامه، موارد مختلفی که در عملکرد کلی سیستم تشخیص کلمات کلیدی در روش ارائه شده تاثیرگذار هستند را بررسی می‌کنیم.

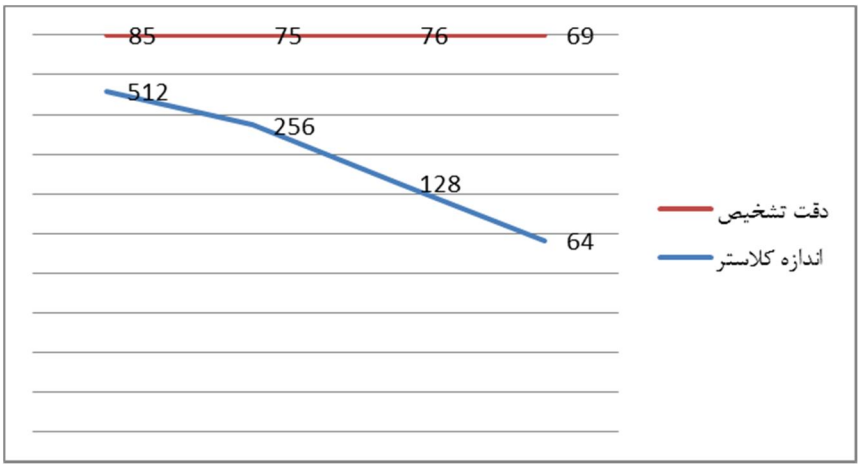
مواردی که در ادامه مطرح می‌شود تاثیر زیادی در عملکرد کلی سیستم دارند:

کلمات کلیدی نمونه: داشتن چندین نمونه از هر کلمه‌ی کلیدی برای انجام آزمایش‌های ما ضروری است. کلماتی که در آزمایشات این تحقیق استفاده شده‌اند، همگی از دادگان پیوسته‌ی گفتار فارسی فارس دات استخراج شده‌اند. این مساله باعث شده است این کلمات بهینه نباشند. از آنجا که تمام این کلمات از بخشی از یک گفتار پیوسته استخراج شده‌اند، در برخی از آنها شکل تلفظ کلمه نسبت به حالتی که ما آنرا به صورت منفرد بیان می‌کنیم تغییر یافته است. این مساله باعث وجود قسمتی از خطای سیستم شده است.

تعداد نمونه‌ها:در آزمایشات ما، از الگوریتم رای گیری برای تشخیص کلمات کلیدی استفاده شده است. البته این بدان معنی نیست که ما هرچه تعداد نمونه‌های بیشتری از هر کلمه‌ی کلیدی در اختیار داشته باشیم، دقت تشخیص بیشتر می‌شود. در واقع حتما باید نصف تعداد نمونه‌ها به یک فریم خاص رای بدهند تا آن فریم به عنوان محل حضور کلمه‌ی کلیدی معرفی شود.

خصوصیات ذاتی کلمات: کلماتی که در هنگام تلفظ در جمله برجستگی بیشتری دارند و همچنین کلماتی که از لحاظ آوایی ارتباط کمتری با قبل و بعد خود در جمله دارند، دارای نرخ تشخیص بالاتری هستند. همچنین کلماتی که صامت‌های بیشتری مانند ((پ ...ژ....ک ...ص...ط...)) در آنها وجود دارند، بیشتر از سایرین قابل جداسازی و تشخیص هستند.

انـــدازه‌ی کلاســـتر: بـــرای بررســـی اثـــر انـــدازه‌ی کلاســـتر روی کـــارایی سیســـتم، آزمایشـــی بـــا ۴ کلمـــه‌ی کلیـــدی مختلـــف انجـــام شـــد. هرچـــه انـــدازه‌ی کلاســـتر بیشـــتر باشـــد، زمـــان و بـــار محاســـباتی نیـــز بیشـــتر مـــی‌شـــود. نمـــودار میـــزان دقـــت سیســـتم نســـبت بـــه انـــدازه‌ی کلاســـتر در شـــکل ۶-۱۱ آورده شـــده اســـت. در این آزمایش، از ۴ اندازه‌ی مختلف ۵۱۲، ۲۵۶، ۱۲۸، ۶۴ برای تعداد کلاسترها استفاده شده است.



شکل ۶-۱۱ نمودار میزان دقت سیستم نسبت به اندازه‌ی کلاستر

سایر ملاحظـات: سـاختار فایلهـای صـوتی دادگـان فـارس دات بـه گونـه‌ای اسـت کـه در یـک فایـل صـوتی روی آواهای مشـابه تاکیـد زیـادی شـده اسـت. بـه عنـوان مثـال در یـک فایـل صـوتی حـرف ((چ)) و یـا ((ژ)) به دفعات بکار رفته است. این عمـل بـه منظـور بهتـر کـردن فراینـد آمـوزش صـورت گرفتـه اسـت؛ ولـی ایـن خـود باعـث بـروز خطـا در هنگـام تشـخیص کلمـات کلیـدی مـی‌شـود. مـثلا اگـر در کلمـه‌ی کلیـدی حـرف ((چ)) بکـار رفتـه باشـد، بـه علـت وجـود کلمـات متعـدد در همـان فایـل کـه دارای حـرف ((چ)) هسـتند، ممکـن اسـت محـل رخ دادن کلمـه‌ی کلیـدی، بـه اشـتباه در مکـان دیگـری تشـخیص داده شـود. معمـولا در سایر جملات فارسی وجود آواهای مشابه فراوان در یک جمله کمتر رخ می‌دهد.

نام فایل : S۱۱۲۵.wav

مگر گاراژ بسته نشده است؟..... طرف ژاله را کتک زد. بخش اعظم دژ خراب شده است. اژدر انداز در دجله غرق شد.

وجود موارد زیاد از یک آوای خاص در فایلهای دادگان فارس دات. در اینجا حرف ((ژ))

فصل هفتم

نتیجه گیری و اقدامات آینده

۷-۱ نتیجه گیری:

جستجوی کلمات کلیدی در گفتار پیوسته توسط الگوریتم Segmental-DTW روی بردارهای ویژگی کوانتیزه شده، در این تحقیق، بررسی، پیاده سازی و ارزیابی شد. در ابتدا با استفاده از داده‌های آموزشی، یک فضای گسترده از ویژگی‌های صوتی را ایجاد کرده و آنرا توسط الگوریتم k-means به تعداد محدودی کلاستر کوانتیزه کردیم. این فضای ویژگی به گونه‌ای بود که تمام تغییرات آوایی ممکن در زبان فارسی را در خود جای دهد. سپس گفتار اصلی و کلمات کلیدی، با استفاده از این بردارهای کوانتیزه شده به یک رشته‌ی یک بعدی از کتاب کد تبدیل شدند. در ادامه گفتار اصلی به تعدادی جزء که با یکدیگر همپوشانی دارند تقسیم شد. هر کدام از این جزءها در ادامه توسط الگوریتم DTW با کتاب کد کلمه‌ی کلیدی مقایسه شدند. نتایج این مقایسه را برای هر جزء به صورت یک امتیاز برای آن جزء در نظر گرفتیم. سپس از روش رای گیری برای یافتن محل حضور کلمه‌ی کلیدی استفاده کردیم و این طور قرار داد کردیم که اگر نصف تعداد نمونه‌های موجود از کلمات کلیدی، یک محل خاص را به عنوان کاندیدای حضور کلمه‌ی کلیدی معرفی کنند آن محل را به عنوان محل واقعی حضور کلمه‌ی کلیدی معرفی می‌کنیم.

الگوریتمی که ما استفاده کردیم، نیازی به وجود فایلهای متنی از داده‌های صوتی (label) ندارد. همچنین از این الگوریتم به راحتی می‌توان در زبان‌های دیگر نیز استفاده کرد. از هیچ هیچگونه مدل خاصی نیز استفاده نشده است و در محاسبات نیز فقط از ۴ عمل اصلی ریاضی و جذر گیری استفاده شده است. از این رو، این روش به راحتی روی سخت افزار نیز قابل پیاده سازی است.

دقت سیستم با روش دیگری که برپایه‌ی الگوریتم Segmental-DTW است[۲] قابل مقایسه است. مهمترین تفاوت روش ما و روش‌های مزبور این است که آنها بر پایه‌ی استفاده از مدل‌های گوسی هستند. الگوریتم کلاسترها بسیار ساده‌تر از الگوریتم مدل‌های گوسی است و از لحاظ محاسباتی نیز

منابع سیستم را بسیار کمتر اشغال می‌کند. تنها پارامتر مورد نیاز در اینجا، بردار میانگین هر کلاستر است. محاسبه‌ی فاصله‌ی بردارهای ویژگی نیز توسط فاصله‌ی اقلیدسی انجام می‌شود. علاوه بر اینها، یک ماتریس فاصله نیز ایجاد کردیم تا محاسبات بعدی با حداکثر سرعت و حداقل محاسبات انجام شود.

فرایند آموزش در روش ما بسیار ساده تر و سریع تر از روش‌هایی مانند مدل مخفی مارکوف است. برای حدود ۱۰۰ دقیقه از داده‌های آموزشی و تعداد کلاستر ۵۱۲، زمان تقریبی ۹۰ دقیقه برای آموزش سیستم صرف شد. سرعت تشخیص کلمه‌ی کلیدی نیز به عرض پنجره بستگی دارد. هرچه عرض پنجره بیشتر باشد، سرعت محاسبه نیز بیشتر است که البته افزایش عرض پنجره باعث کاهش کارایی سیستم خواهد شد. می‌توان این امکان را به کاربر داد تا با توجه به نوع کاربرد خود، بین زمان و دقت، فاکتور مورد نظر خود را اعمال نماید.

تمامی آزمایشات انجام شده در این تحقیق، توسط یک کامپیوتر قابل حمل وایو سونی با میزان حافظه فعال ۴ گیگابایت و یک پردازنده‌ی دو هسته ای با سرعت پردازشی ۲.۵۶ گیگاهرتز و سیستم عامل ویندوز سون، در محیط نرم افزار متلب انجام شده است.

۷-۲ اقدامات آینده:

همانطور که آزمایشات ملاحظه شد، در حالت پیوسته برای برخی از کلمات دقت بسیار بالایی بدست آمد. در برخی از کلمات هم دقت بسیار پایینی داشتیم. به عنوان مثال در کلمه ((انقلاب)) که در جدول کلمات به آن اشاره نشده است، دقت ۳۰ ٪ حاصل شد. علت حاصل شدن دقت پایین برای برخی از کلمات در اینجا ذکر می‌شود.

بخش زیادی از این دقت پایین، به این علت است که کلمات کلیدی نمونه از گفتار پیوسته استخراج شده‌اند. نحوه‌ی تلفظ برخی از این کلمات تحت تاثیر کلمات قبلی و بعدی و سرعت و استرس در بیان گوینده قرار دارد. می‌توان برای افزایش دقت سیستم از دادگان گفتاری دقیق‌تر و حتی از دادگان گفتاری گسسته برای کلمات کلیدی استفاده کرد. علاوه بر این، دادگان گفتاری فارس دات دارای نویز زیادی است، به طوری که این نویز به راحتی قابل شنیدن است. این مساله، خود باعث کاهش کارایی سیستم خواهد شد.

می‌توان برای ایجاد فضای ویژگیها، از سایر انواع ویژگی‌های صوتی مانند تبدیل موجک و یا PLP نیز بهره برد. همچنین می‌توان از شبکه‌های عصبی برای انجام فرایند تشخیص استفاده کرد.

در آینده قصد داریم سیستم تشخیص کلمات گسسته را به صورت سخت افزاری پیاده سازی کنیم. با توجه به الگوریتم بیان شده، در پیاده سازی سخت افزاری کار بسیار راحتی خواهیم داشت. شاید مهمترین چالش در پیاده سازی سخت افزاری، محاسبه‌ی ضرایب ویژگی باشد. اینکار پیچیدگی زیادی در هنگام پیاده سازی دارد. می‌توان برای انجام راحت‌تر این کار، از آی سی‌هایی که خصوصا برای پردازش سیگنال طراحی شده‌اند، بهره برد.

منابع و مآخذ

[۱] PARK, A.S. AND GLASS, J.R., ۲۰۰۸. Unsupervised Pattern Discovery in Speech. *IEEE Transactions on Audio, Speech, and Language Processing*.

[۲] ZHANG, Y. AND GLASS, J.R., ۲۰۰۹. Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams. *IEEE Workshop on Automatic Speech Recognition & Understanding, ۲۰۰۹. ASRU ۲۰۰۹.*

[۳] HAZEN, T.J., SHEN, W. AND WHITE, C., ۲۰۰۹. Query-by-example spoken term detection using phonetic posteriorgram templates. *IEEE Workshop on Automatic Speech Recognition & Understanding, ۲۰۰۹. ASRU* .

[٤] KESHET, J., GRANGIER, D. AND BENGIO, S., ۲۰۰۹. Discriminative keyword spotting. *Speech Communication* .

[٥] SOMERVUO, P. AND KOHONEN, T., ۱۹۹۹. Self-organizing maps and learning vector quantization for feature sequences. *Neural Processing Letters*.

[٦] ZHANG, Y. AND GLASS, J.R., ۲۰۱۰. Towards multi-speaker unsupervised speech pattern discovery. *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP).*

[۷] DONGZHI HE, YIBIN HOU, YUANYUAN LI AND ZHI-HAO DING. ۲۰۱۰. Key technologies of pre - processing and post-processing methods for embedded automatic speech recognition systems. In *Mechatronics and Embedded Systems and Applications (MESA), ۲۰۱۲ IEEE/ASME International Conference*, ۷٦-۸۰.

[۸] ZAHARIA, T., SEGARCEANU, S., COTESCU, M. AND SPATARU, A., ۲۰۱۰. Quantized Dynamic Time Warping (DTW) algorithm. *Communications (COMM), ۲۰۱۰ ۸th International Conference.*

[۹] GRANGIER, D. AND BENGIO, S., ۲۰۰۷. Learning the Inter-frame Distance for Discriminative Template-based Keyword. International Conference on Speech Communication and Technology (INTERSPEECH).

[۱۰] SAKOE, H. AND CHIBA, S., ۱۹۷۸. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*.

[۱۱] MYERS, C., RABINER, L. AND ROSENBERG, A., ۱۹۸۰. An investigation of the use of dynamic time warping for word spotting and connected speech recognition. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP* .

[۱۲] ANGUERA, X., MACRAE, R. AND OLIVER, N., ۲۰۱۰. Partial sequence matching using an Unbounded Dynamic Time Warping algorithm. *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), ۲۰۱۰.*

[۱۳] Chunsheng Fang " From Dynamic Time Warping (DTW) to Hidden Markov Model (HMM)" University of Cincinnati, ۲۰۰۹/۳/۱۹

[۱٤] Hadi Naderi "Hidden Markov Models-final Report for speech processing course" .Shahrood university of technology ۲۰۱۱"

*[۱٥] Jan NOUZA, Jan SILOVSKY "*Fast Keyword Spotting in Telephone Speech" RADIOENGINEERING, VOL. ۱۸, NO. ٤, DECEMBER ۲۰۰۹

[۱٦] Berlin Chen "Keyword Spotting & Utterance Verification" Department of Computer Science & Information Engineering National Taiwan Normal University

[١٧] Hadi Taghizade " Dynamic time wrapping approaches-final report for speech processing course " Shahrood university of technology ٢٠١١

[١٨] LLOYD, S., ١٩٨٢. Least squares quantization in PCM. *IEEE Transactions on Information Theory,* ٢٨, ١٢٩-١٣٧.

[١٩] SILAGHI, M.C., ٢٠٠٨. A new evaluation criteria for keyword spotting techniques and a new algorithm. Proceedings of InterSpeech, ٢٠٠٥.

[٢٠] HTK handbook by Cambridge University

[٢١] HERMANSKY, H., MORGAN, N., BAYYA, A. AND KOHN, P., ١٩٩٢. RASTA-PLP speech analysis technique. In *Acoustics, Speech, and Signal Processing,* ١٩٩٢. *ICASSP-٩٢., ١٩٩٢ IEEE International Conference*.

[٢٢] REAL TIME SPEAKER RECOGNITION,USING MFCC AND VQ,A THESIS SUBMITTED IN PARTIAL FULFILLMENT,OF THE REQUIREMENTS FOR THE DEGREE OF Master of Technology In Telematics and Signal Processing,٢٠٠٨

[٢٣] Ye-Yi Wang, Dong Yu, Yun-Cheng Ju, and Alex Acero An Introduction to Voice Search [A look at the technology, the technological challenges, and the solutions.

[٢٤] KEYWORD-SPOTTING USING SRI'S DECIPHERm LARGE-VOCABUARLY,SPEECH-RECOGNITION SYSTEM,Mitchel Weintraub,SRI International,Speech Research and Technology Program.Menlo Park, CA, ٩٤٠٢٥

[٢٥] GOPALAN, K., CHU, T. AND MIAO, X.,٢٠٠٩. An Utterance Recognition Technique for Keyword Spotting by Fusion of Bark Energy and MFCC. International conference on signal, speech and image processing.

[٢٦] MORGAN, D.P., SCOFIELD, C.L., LORENZO, T.M., REAL, E.C. AND LOCONTO, D.P١٩٩٠. A keyword spotter which incorporates neural networks for secondary processing. *International Conference on Acoustics, Speech, and Signal Processing, ICASSP-١٩٩٠., ١٩٩٠*.

[٢٧] Acoustic keyword spotting in speech with applications to data mining A.j.kishan Thambiratnam. March٢٠٠٥

[٢٨] Fast Keyword Spotting in Telephone Speech, *Jan NOUZA, Jan SILOVSKY,* SpeechLab, Faculty of Mechatronics, Technical University of Liberec, Studentska ٢, ٤٦١١٧ Liberec, Czech Republic.jan.nouza@tul.cz, jan.silovsky@tul.cz

[٢٩] Phoneme based acoustics keyword spotting in informal continuous speech, Igor Szoke, Petr Schwarz, Lukas Burget, Martin Karafiat, Jan ·Cernocky,Faculty of Information Technology, Brno University of Technology,E-mail: szoke@¯t.vutbr.cz

[٣٠] Keyword Detection for Spontaneous Speech, Weifeng Li, Aude Billard, and Herv´e ourlard, Swiss Federal Institute of Technology, Lausanne (EPFL), Switzerland,Idiap Research Institute, CH-١٩٢٠, Martigny, Switzerland Email: {weifeng.li}@epfl.ch

[٣١] KEYWORD RECOGNITION WITH PHONE CONFUSION NETWORKS AND PHONOLOGICAL FEATURES BASED KEYWORD THRESHOLD DETECTION Abhijeet Sangwan and John H. L. Hansen Center for Robust Speech Systems (CRSS),Department of Electrical Engineering, University of Texas at Dallas, Richardson, Texas, U.S.A

[٣٢] A NEW KEYWORD SPOTTING ALGORITHM WITH PRE-CALCULATED, OPTIMAL THRESHOLDS,J. Junkawitsch, L. Neubauer, H. Höge, and G. Ruske, Technical University of Munich, Arcisstr. ٢١, D-٨٠٢٩٠ Munich, Germany, Siemens Corp., Otto-Hahn-Ring ٦, D-٨١٧٣٠ Munich, Germany.

[۳۳] A NEW KEYWORD SPOTTING APPROACH BASED ON REWARD FUNCTION, Y. Benayed, D. Fohr, J. P. Haton and G. Chollet, LORIA-CNRS/INRIA-Lorraine, BP۲۳۹, F٥٤٥٠٦, Vandoeuvre, France, ENST, CNRS-LTCI, ٤٦ rue Barrault, F٧٥٦٣٤, Paris, France.

[۳٤] Key-Word Spotting-The Base Technology, for Speech Analytics,By Guy Alon,NSC - Natural Speech communication Ltd.۳۳ Lazarov St. Rishon Lezion Israel, e - m a i l : i n f o @ n s c s p e e c h . c o m.

[۳٥] Keyword Spotting in Continuous Speech Utterances,Yong Ling, School of Computer Science ,McGill University, Montreal, Yong Ling, ۱۹۹۹, Febniary ۱, ۱۹۹۹

[36] ارائه روشی جدید برای تعیین معیار اطمینان در پذیرش کلمات کلیدی در زبان فارسی, محمد مهدی همایون پور دانشکده مهندسی کامپیوتر و فن آوری، اطلاعات، دانشگاه صنعتی امیرکبیر homayoun@aut.ac.ir مریم اسدی، دانشکده مهندسی کامپیوتر و فن آوری اطلاعات، دانشگاه صنعتی امیرکبیرMa_asadi@aut.ac.ir

Shahrood University of Technology

Faculty of Electrical and Robotic Engineering

Keyword Spotting in Speech Utterance

Hadi Naderi

Supervisor

Dr. Hossein Marvi

February ۲۰۱۳

Abstract

Detecting Keywords or Keyword Spotting, generally means finding a keyword in a written or spoken document. In this research, a new approach for detecting or recognizing keywords in Persian language is proposed, to spot keywords in continues and discrete utterances. Dynamic Time Warping approach is used in the both parts which is different from Hidden Markov Model based approaches that is widely used today.

The proposed algorithm for keyword spotting in continues utterance is based on a modified version on DTW, named Segmental DTW. DTW is a classic distance measure for calculating the similarity between two sequences which are different in length. In the processing stage, the utterance is divided into short frames. Each frame is represented in forms of quantized feature vectors. Both keywords and utterance are converted into a sequence of indices of codebooks.

After that, the sequence of codebook indices of the utterance is divided into some segments. Later on, the system calculates the DTW distance of each of segment with query keyword(s). finally the segment with the minimum of distortion score, is most likely to be a keyword.

We also have proposed an approach for keyword spotting in isolated words. In this part, we first prepare a dataset from desired keywords. We compose a generic model for any of desired keywords, using alignment path in Dynamic Time Warping algorithm. In the test stage, we calculated DTW the distance between generic models of keywords and the incoming words. The model with the minimum distortion score is candidate to be a keyword.

Keywords: Keyword Spotting, Speech Recognition, Isolated Word Recognition, Dynamic Time Warping, Segmental-DTW, Hidden Markov Model, HMM