دانشگاه صنعتی شاهرود

دانشکده برق و رباتیک

گروه الکترونیک

تشخیص دستنوشته برخط فارسی

به کمک ویژگی‌های مبتنی بر شکل

هادی تقی زاده

استاد راهنما:

دکتر علیرضا احمدی فرد

استاد مشاور:

دکتر علی سلیمانی

پایان نامه ارشد جهت اخذ درجه کارشناسی ارشد

بهمن ۱۳۹۱

بسم الله الرحمن الرحيم

دانشگاه صنعتی شاهرود

دانشکده برق و رباتیک

گروه الکترونیک

تشخیص دستنوشته برخط فارسی

به کمک ویژگی‌های مبتنی بر شکل

هادی تقی زاده

استاد راهنما:

دکتر علیرضا احمدی فرد

استاد مشاور:

دکتر علی سلیمانی

پایان نامه ارشد جهت اخذ درجه کارشناسی ارشد

بهمن ۱۳۹۱

شماره : ۱۰۴۸/آ.ت.ب
تاریخ : ۹۱/۱۱/۲۸
ویرایش : ------

بسمه تعالی

دانشگاه صنعتی شاهرود
مدیریت تحصیلات تکمیلی
فرم شماره (۶)

فرم صورتجلسه دفاع پایان نامه تحصیلی دوره کارشناسی ارشد

با تأییدات خداوند متعال و با استعانت از حضرت ولی عصر (عج) جلسه دفاع از پایان نامه کارشناسی ارشد خانم / آقای :

هادی تقی زاده رشته : **بـــرق** گرایش : **الکترونیک**

تحت عنوان : **تشخیص دست نوشته بر خط فارسی به کمک ویژگیهای مبتنی بر شکل**

که در تاریخ **۱۳۹۱/۱۱/۲۸** با حضور هیأت محترم داوران در دانشگاه صنعتی شاهرود برگزار گردید به شرح زیر است :

قبول (با درجه : بسیار خوب امتیاز ۱۸/۵) ☑ دفاع مجدد ☐ مردود ☐

۱ـ عالی (۲۰ ـ ۱۹)
۲ـ بسیار خوب (۱۸/۹۹ ـ ۱۸) ✓
۳ـ خوب (۱۷/۹۹ ـ ۱۶)
۴ـ قابل قبول (۱۵/۹۹ ـ ۱۴)
۵- نمره کمتر از ۱۴ غیر قابل قبول

عضو هیأت داوران	نام ونام خانوادگی	مرتبهٔ علمی	امضـــاء
۱ـ استـاد راهنمـا	علیرضا اخوان فر	استادیار	
۲ـ استـاد مشـاور	سید مهدی سلیمانی ایوری	استادیار	
۳ـ نماینده شورای تحصیلات تکمیلی	سامان ناصح	استادیار	
۴ـ استـاد ممتحـن	حسین ایزدیان	استادیار	
۵ ـ استـاد ممتحـن	هادی گرایلو	استادیار	

رئیس دانشکده :

تقدیم به پدر و مادرم

نخست خداوند بزرگ و علیم را شاکرم که به من قدرت تفکر و نوشتن داد و در وجودم اشتیاق یافتن پاسخ معماهایی هر چند کوچک از دانش را قرار داد.

بر خود لازم می دانم که از استاد راهنمای فرهیخته و دانشمند آقای دکتر علیرضا احمدی فرد برای تلاش خستگی ناپذیر و صبر و حوصله در پاسخگویی و راهنمایی جهت پیشبرد این پایان نامه کمال تشکر و قدردانی را بجا آورم.

همچنین از استاد مشاور، جناب دکتر علی سلیمانی و دیگر اساتید محترم گروه الکترونیک که با تخصص و تعهد خود راهنمای من در مسیر کسب دانش بوده اند تشکر می کنم و از خداوند متعال برایشان شادی و موفقیت آرزومندم.

در پایان از خانواده خود و همه دوستانم که در انجام این کار مرا یاری نمودند قدردانی می کنم.

تعهـــد نامـــه

اینجانب ..... دانشجوی دوره کارشناسی ارشد رشته ..... دانشکده مهندسی برق و رباتیک دانشگاه صنعتی شاهرود نویسنده پایان نامه با عنوان :

.....

تحت راهنمائی آقای دکتر ..... 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 « **دانشـــگاه صـــنعتی شـــاهرود** » و یـا « 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 پایـان نامـه رعایـت مـی گردد.
- در کلیـه مراحل انجام این پایان نامـــه، در مواردی که از موجـود زنـده (یا بافت های آنها) استفاده شده است ضوابط و اصـول اخـــلاقی 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 : ۱۳۹۱/۱۱/۲۸

امضاء 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

در این تحقیق روشی برای تشخیص دستنوشته برخط فارسی مبتنی بر قطعه‌بندی زیر-کلمه به حروف و شناسایی حروف قطعه‌بندی شده با استفاده از مدل مخفی مارکوف گسسته ارائه شده است. تصویر متن تایپی یا دستنوشته به دلیل این که به صورت یکجا و بعد از نوشتن کامل آن در دسترس است برون خط نامیده می‌شود در حالی که دستنوشته دریافت شده توسط وسایل دیجیتال نظیر تبلت و تلفن همراه با صفحه لمسی به دلیل در دسترس بودن اطلاعات نوشته همزمان با عمل نوشتن، برخط نامیده می‌شود. برای تشخیص دستنوشته برون خط لازم است برای استخراج ویژگی از روش‌های پردازش تصویر استفاده شود ولی ویژگی‌های دستنوشته برخط مستقیماً از اطلاعات آن که شامل مختصات افقی و عمودی نقاط نوشته است استخراج می‌شود.

استخراج حروف در زیر-کلمه با قطعه‌بندی اضافی بر اساس زاویه نوشته با محور افقی انجام شده است و سپس با حذف موارد اشتباه طی چند مرحله، نقاط قطعه‌بندی نهایی مشخص می‌شوند. به دلیل وجود نقاط قطعه‌بندی اضافی، ترکیب‌های مختلف قطعات امتیازبندی می‌شوند که این امتیازبندی بر حسب میانگین احتمال نرمالیزه شده رخداد توالی مشاهده قطعات آن‌ها در مدل مخفی مارکوف است. سپس ترکیب‌های دارای بالاترین امتیاز به عنوان کاندیدهای نهایی تشخیص گروه حروف معرفی می‌شوند، با استفاده از فرهنگ لغت متنی، گزینه‌های با گروه حروف مشابه پیدا می‌شود و گزینه‌هایی که در فرهنگ لغت موجود نیستند حذف می‌شوند. در مرحله بعد گزینه‌هایی که از نظر وجود علامت در بالا و پایین زیر-کلمه با زیر-کلمه ورودی منطبق نیستند حذف می‌شوند و در مرحله نهایی گزینه‌هایی که تعداد حرکت‌های زیر-کلمه در بالا و پایین آن‌ها با زیر-کلمه ورودی مساوی نیست حذف می‌شوند.

صحت قطعه‌بندی با روش این تحقیق با استفاده از Ground Truth ساخته شده آزمایش شده است که ۸۵/۴۷ درصد نقاط قطعه‌بندی به درستی شناسایی شده‌اند و ۶۰/۱ درصد نقاط اشتباه پیدا شده است. تشخیص زیر-کلمه با دقت ۳۷/۸۵ درصد انجام شده است که برای ۱۰ گزینه اول بازشناسی به ۷۳/۳۴ درصد می‌رسد. این مقدار نسبت به کارهایی که با روش کلی نگر که کل زیر-کلمه را به عنوان الگوی شناسایی در نظر می‌گیرند و روی پایگاه داده تحقیق انجام شده‌اند خیلی کمتر است که به دلیل ماهیت روش مبتنی بر قطعه‌بندی است که نیاز به قطعه‌بندی دقیق دارد اما کار مشابهی با همین پایگاه داده مبتنی بر تشخیص حروف وجود ندارد تا مقایسه انجام شود.

مزیت روش مبتنی بر قطعه‌بندی که در این تحقیق از آن استفاده شده، نسبت به روش کلی‌نگر در این است که برای شناختن زیر-کلمات جدید کافی است که متن آن‌ها به سیستم اضافه شود در حالی که روش کلی‌نگر نیاز به نمونه‌های جدید دستنوشته دارد. مهم‌ترین محدودیت این روش، عدم کارایی الگوریتم‌های قطعه‌بندی است که خطای آن به مرحله تشخیص منتقل می‌شود و منجر به کاهش چشمگیر در نرخ شناسایی می‌شود.

کلمات کلیدی: تشخیص دستنوشته برخط فارسی، قطعه‌بندی، مدل مخفی مارکوف

فهرست مطالب

عنوان **صفحه**

فهرست جدول‌ها

فهرست شکل‌ها

عنوان **صفحه**

علائم و اختصارات

DTW	Dynamic Time Warping
EM	Expectation Maximization
FP	False Positive
FEM	Fuzzy Elastic Machine
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
lpi	line per inch
OCR	Optical Character Recognition
SOM	Self-Organizing Maps
SVM	Support Vector Machine
TP	True Positive

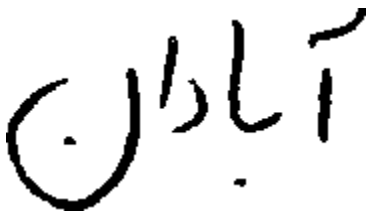
فصل ۱

مقدمه

فصل ۱- مقدمه

۱-۱- پیشگفتار

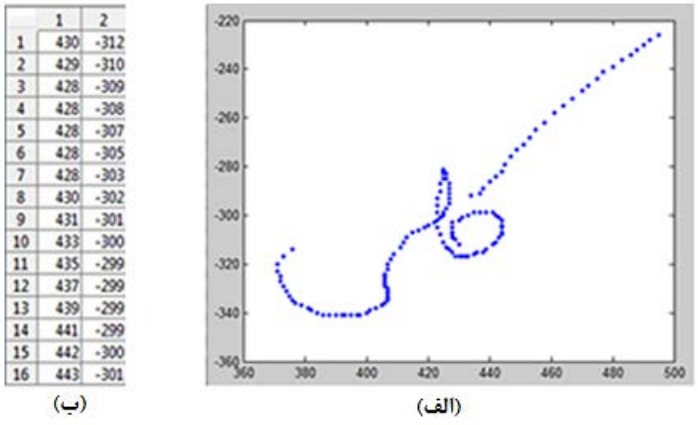
در عصر حاضر مقدار زیادی اطلاعات به صورت نوشته در هر لحظه تولید می‌شود. نوشته‌های تایپ شده با فرمت‌های دیجیتال، امکانات زیادی از قبیل ویرایش آسان، حجم کم و قابلیت جستجو را فراهم می‌کنند اما هنوز هم اطلاعات زیادی به صورت تصویر متن تایپی و دستنوشته وجود دارد. نیاز به جستجو در تصویر متن تایپ شده و یا دستنوشته باعث شد که روش‌هایی برای تشخیص متن داخل تصویر و تبدیل آن به شکل قابل جستجو ارائه شوند. تشخیص متن از تصویر نوشته OCR[1] نام گرفت که در فارسی به نویسه‌خوانی نوری ترجمه شده است [۱]. در این سیستم، ورودی یک تصویر شامل متن تایپی یا دستنوشته است، سیستم پردازش‌های لازم را روی تصویر انجام می‌دهد و خروجی آن متن تشخیص داده شده است. در شکل ۱-۱ یک نمونه دستنوشته برون خط از پایگاه داده ایران‌شهر آورده شده است.



شکل ۱-۱: نمونه‌ای از یک متن دستنویس برون‌خط از پایگاه داده ایران‌شهر

یکی از نتایج پیشرفت الکترونیک، تولید صفحات لمسی بود. استفاده از وسایلی مانند نمایشگرهای لمسی، گوشی‌های تلفن همراه و کامپیوترهای جیبی با صفحه نمایش لمسی، تبلت‌ها و تبلت‌های گرافیکی در حال گسترش است. با گسترش استفاده از این وسایل، تشخیص متن دستنوشته در آن‌ها تبدیل به یک موضوع مورد علاقه در زمینه شناسایی الگو شده است. در متن نوشته شده بر روی صفحه این وسایل علاوه بر مختصات نقاط نوشته، اطلاعات زمانی مربوط به این مختصات هم در دسترس است و خروجی دیجیتال همزمان با ورود داده آماده است به همین دلیل این نوع ورودی را برخط[2] می‌نامند در حالی که تصویر متن که به صورت فایل تصویر است داده برون خط[3] نام دارد.
در شکل ۱-۲ یک نمونه دستنوشته برخط و مختصات تعدادی از نقاط آن آورده شده است.

[1] Optical Character Recognition
[2] Online
[3] Offline



(ب)

	1	2
1	430	-312
2	429	-310
3	428	-309
4	428	-308
5	428	-307
6	428	-305
7	428	-303
8	430	-302
9	431	-301
10	433	-300
11	435	-299
12	437	-299
13	439	-299
14	441	-299
15	442	-300
16	443	-301

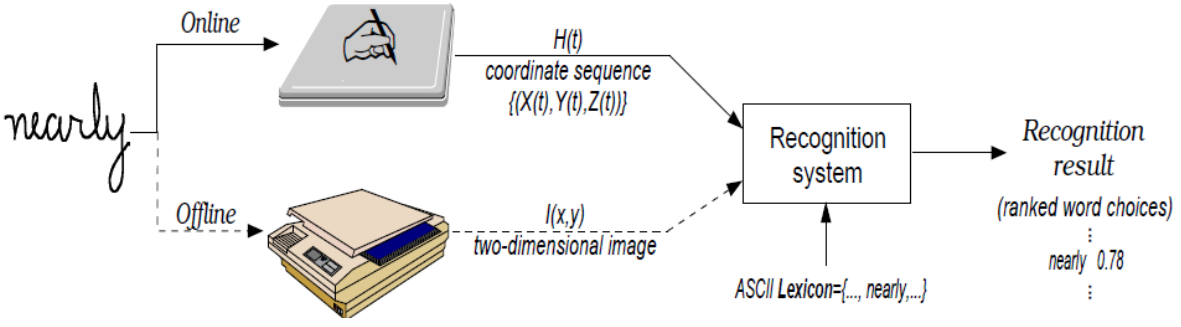
(الف)

شکل ۱-۲: (الف) نمونه‌ای از یک زیر-کلمه دستنویس برخط از پایگاه داده زیر-کلمات برخط دانشگاه تربیت مدرس، (ب) مختصات تعدادی از نقاط مربوط به این زیر-کلمه

۱-۲- شناسایی برون‌خط و شناسایی برخط

در شناسایی برون‌خط، تصویر اسکن شده وارد سیستم می‌شود، سیستم شناسایی، نواحی متن را استخراج می‌کند و با استفاده از روش‌های پردازش تصویر، نوشته را تشخیص داده و به شکل متن قابل ویرایش تبدیل می‌کند.

در شناسایی برخط، متن توسط قلم و صفحه حساس نوشته می‌شود. صفحه حساس به صورت شبکه‌ای از نقاط نزدیک به هم است که به هر کدام یک مختصات (x,y) اختصاص می‌یابد، نقاطی که قلم روی آن‌ها قرار می‌گیرد تحریک می‌شوند و مختصات آن‌ها توسط یک رابط به خروجی وسیله ارسال می‌شود. مختصات نقاط، ورودی سیستم شناسایی برخط هستند، سیستم با انجام پردازش‌های لازم روی این مختصات، متن نوشته شده را شناسایی می‌کند. شکل ۱-۳ مثالی از سیستم‌های تشخیص برخط و برون خط را نشان می‌دهد.



شکل ۱-۳: سیستم‌های تشخیص برخط و برون‌خط[۲]

نوشتار برخط نسبت به نوشتار برون‌خط اطلاعات بیشتر و دقیق‌تری به ما می‌دهد، به عنوان مثال در نوشته برخط ترتیب زمانی نوشتن هر حرکت مشخص است در حالی که نوشته برون‌خط فاقد این اطلاعات است. یکی دیگر از مزایای نوشته برخط این است که در هنگامی که کاربر چیزی نمی‌نویسد

قلم روی صفحه قرار ندارد و در نتیجه هیچ مختصاتی هم به خروجی ارسال نمی‌شود (سیگنال Pen-Up در هنگام برداشتن قلم از صفحه و سیگنال Pen-Down در هنگام گذاشتن قلم روی صفحه تولید می‌شوند)، این مزیت باعث می‌شود که هر تک حرکت[1] قلم (مسیر قلم از لحظه گذاشتن روی صفحه تا لحظه برداشتن) قابل شناسایی باشد.

۱-۳- وسایل نوشتار برخط

۱-۳-۱- وسایل هوشمند با صفحه لمسی

کامپیوترهای جیبی و گوشی‌های تلفن همراه جدید از صفحات لمسی استفاده می‌کنند. مدل‌هایی که بتوان با قلم مخصوص روی صفحه آن‌ها نوشت برای کار دستنوشته مناسب هستند. تعدادی از این وسایل به همراه نرم افزارهای تشخیص دستنوشته ارائه می‌شوند که قابلیت تشخیص دستنوشته لاتین را دارند. تعدادی از مدل‌ها نیز حروف مجزای فارسی را تشخیص می‌دهند که البته دقت و سرعت قابل قبولی ندارند ولی تا کنون سیستم تشخیص دستنوشته پیوسته فارسی برای این وسایل ارائه نشده است.

صفحات این وسایل معمولاً از فناوری مقاومتی یا خازنی استفاده می‌کنند. خروجی این صفحات فقط مختصات نقاط تحریک شده است که با فاصله زمانی ثابت به واحدی که این مختصات را پردازش می‌کند ارسال می‌شود. صفحات خازنی حساسیت بهتری به لمس با انگشت دارند ولی قادر به تشخیص ضربه توسط اشیاء دیگر نیستند در حالی که صفحات مقاومتی با وجود حساسیت کمتر، با قلم معمولی هم تحریک می‌شوند. شکل ۱-۴ نمونه‌ای از دستنوشته لاتین و متن تشخیص داده شده در یک تلفن همراه را نشان می‌دهد.

شکل ۱-۴: تشخیص متن دستنویس لاتین در تلفن همراه

[1] Stroke

۱-۳-۲- تبلت‌های گرافیکی

اساس کار همه صفحات رقومی کننده[1] مشابه است، این صفحات حساس به میدان مغناطیسی هستند و به همراه یک قلم مغناطیسی به فروش می‌رسند. بیشتر مدل‌های صفحات رقومی کننده، تبلت‌های گرافیکی هستند که به عنوان وسیله طراحی گرافیست‌ها جایگزین قلم و کاغذ شده‌اند. این تبلت‌ها همچنین برای بعضی طراحی‌های صنعتی مثل طراحی ظاهر اتومبیل و یا معماری ساختمان کاربرد دارند. علاوه بر تبلت‌های گرافیکی، تبلت‌هایی با کاربرد ویژه یادداشت برداری هم موجودند که با عنوان دفترچه یادداشت دیجیتال[2] شناخته می‌شوند. تفاوت این تبلت‌ها با مدل گرافیکی این است که روی این صفحات، کاغذ معمولی که می‌تواند یک کاغذ خط دار باشد قرار می‌گیرد و می‌توان با قلم معمولی روی آن نوشت و اطلاعات در حافظه داخلی تبلت ذخیره می‌شود، علاوه بر این که نوشتن روی خط زمینه از قبل مشخص شده به روند تشخیص متن کمک می‌کند، نوشتن روی کاغذ مزیت مهم دیگری هم دارد که نیاز به نمایشگر را از بین می‌برد و کاربر دقیقاً به محل نوشتن نگاه می‌کند در نتیجه حرکت دست کاملاً طبیعی است در حالی که بیشتر تبلت‌های گرافیکی فاقد نمایشگر هستند و لازم است به رایانه وصل شوند تا طرح کشیده شده روی تبلت، روی نمایشگر نشان داده شود، این در حالیست که تبلت‌های دارای صفحه نمایش که می‌توان از آن‌ها به عنوان نمایشگر رایانه هم استفاده کرد از تبلت‌های بدون صفحه نمایش به مراتب گران‌تر هستند. یک نمونه دفترچه یادداشت دیجیتال و تبلت گرافیکی در شکل ۱-۵ نشان داده شده است.

شکل ۱-۵: (الف) دفترچه یادداشت دیجیتال CROSSPAD، (ب) تبلت گرافیکی WACOM مدل Bamboo Pen & Touch

[1] Tablet Digitizers
[2] Digital Notepad

تعداد نقاط صفحه رقومی کننده در واحد سطح نشان دهنده قابلیت تفکیک[1] تبلت است، این ویژگی برای تبلت‌ها معمولاً با واحد lpi[2] نشان داده می‌شود. هر چه قابلیت تفکیک تبلت بیشتر باشد، می‌تواند شکل دقیق‌تری از طرح کشیده شده بدهد. این ویژگی تبلت‌ها مشابه ویژگی قابلیت تفکیک نمایشگرهای وسایل دیجیتال است. با توجه به مدل‌های موجود در بازار می‌توان گفت که حتی کم دقت‌ترین تبلت‌ها از نظر قابلیت تفکیک برای کار دستنوشته کافی هستند.

یکی دیگر از ویژگی‌های صفحات رقومی کننده، سرعت پاسخ[3] آن‌هاست بدین معنی که در هر ثانیه حداکثر چه تعدادی از مختصات نقاط تحریک شده قابل دریافت است. به عنوان مثال سرعت پاسخ ۱۳۳ نقطه در ثانیه یعنی در هر ثانیه حداکثر تعداد نقاط لمس شده توسط قلم که به مختصات تبدیل شده و به خروجی می‌رود ۱۳۳ نقطه است.

اندازه ناحیه فعال[4] صفحه نمایش هم از دیگر ویژگی‌های یک تبلت است. معمولاً اندازه‌های استاندارد مثلاً A4 یا نسبت طول به عرض 4:3 کاربرپسندتر هستند.

علاوه بر موارد گفته شده که در همه تبلت‌ها مشترک هستند، ویژگی‌هایی هستند که فقط توسط مدل‌های خاصی پشتیبانی می‌شوند. یکی از این ویژگی‌ها تعداد سطوح فشار است. میزان فشار قلم روی صفحه در کارهای گرافیکی برای ایجاد شدت رنگ قابل استفاده است، همچنین در تشخیص امضا از این خاصیت استفاده می‌شود و در تشخیص هویت نویسنده هم مفید است؛ اما به نظر نمی‌رسد در شناسایی دستنوشته کاربردی داشته باشد هرچند این موضوع قابل بررسی است.

بعضی مدل‌های تبلت علاوه بر تشخیص فشار قادر به تشخیص زاویه قلم تا حدود معینی هستند، این ویژگی هم در کارهای گرافیکی کاربرد دارد به عنوان مثال برای سایه زدن استفاده می‌شود اما در تشخیص دستنوشته کاربردی ندارد.

۱-۴- مسائل مربوط به دستنوشته برخط

۱-۴-۱- زبان نوشته

با توجه به وجود تفاوت‌های زیاد در شکل نوشتار در زبان‌های مختلف، مقایسه روش‌های تشخیص دستنوشته بین زبان‌های مختلف منطقی نیست. در زمینه تشخیص دستنوشته برخط انگلیسی تحقیقات گسترده‌ای انجام شده است[۳] ولی تحقیقات کمی در زمینه تشخیص دستنوشته برخط عربی و فارسی انجام شده است[۴]. در فصل ۲ تعدادی از کارهای انجام شده در این زمینه مرور می‌شود.

[1] Resolution
[2] line per inch (lpi)
[3] Response Rate
[4] Active Area

تشخیص دستنوشته در زبان فارسی و زبان‌های با الفبای مشابه، نسبت به دیگر زبان‌ها پیچیدگی‌های بیشتری دارد. پیچیدگی تشخیص دستنوشته عربی/فارسی را می‌توان در موارد زیر خلاصه کرد[۵]:

- طبیعت پیوسته متن فارسی
- وجود علامت‌های مد، همزه، دسته، سرکش و نقطه
- اشکال مختلف حروف در موقعیت‌های مختلف
- تغییر شکل بعضی حروف در نوشته پیوسته مثل حرف ه در "ها" و ح در "محمد"
- ابهام در شکل بعضی حروف به دلیل شکل‌های مختلف مثل حرف س که می‌تواند بدون دندانه هم نوشته شود.
- شباهت کلی شکل حروف
- جابجا شدن موقعیت علامت حروف در کلمه به دلیل عدم دقت نویسنده

۱-۴-۲- واحد مورد شناسایی

هدف از تشخیص دستنوشته، شناسایی کل کلمه است اما کارهای تحقیقاتی بر روی حروف مجزا، ارقام و بعضی علائم خاص هم انجام شده است. کارهای دیگر در زمینه تحلیل خودکار اسناد[۱] شامل تشخیص خط زمینه(کرسی)[۲]، تشخیص ناحیه متن، تشخیص جدول و بازسازی آن، تشخیص اشکال و علامت‌ها، شناسایی نویسنده[۳]، تصدیق نویسنده[۴]، تصدیق امضا[۵]، کاهش فرهنگ لغت[۶] و ... هستند.

رقم

تمام اعداد با ۱۰ رقم ۰ تا ۹ ساخته می‌شوند و ارقام یک عدد همیشه به صورت جدا نوشته می‌شوند به همین دلیل شناسایی عدد نسبت به حروف جدا و کلمه آسان‌تر است. البته در کاربرد حتی خطای کم هم غیر قابل قبول است به عنوان مثال در کارهای بانکی هیچ اشتباهی قابل قبول نیست مخصوصاً که یک خطای کوچک ممکن است منجر به تغییر زیادی در مبلغ شود در حالی که در شناسایی کلمه چنین حساسیتی وجود ندارد.

[1] Document Analysis
[2] Baseline
[3] Writer Identification
[4] Writer Verification
[5] Signature Verification
[6] Lexicon Reduction

علائم

تشخیص فرمول‌های ریاضی و یا فرمول‌های شیمی نیاز به تشخیص علائم مخصوص مربوط به این مقوله‌ها دارد. در چنین کاربردهایی علاوه بر حروف لاتین، بعضی حروف یونانی، علائم ریاضی مثل رادیکال، نامساوی، مساوی و ... هم باید شناسایی شوند. با توجه به محدودیت این بحث، مقالات کمی در این زمینه وجود دارد.

حروف جدا

بعضی وسایل قادر به تشخیص حروف جدا هستند، این قابلیت چندان کاربردی نیست چون نوشتن کلمات با حروف جدا از هم سرعت نوشتن را به شدت کاهش می‌دهد به ویژه در زبان فارسی که حروف کلمات ذاتاً پیوسته هستند، نوشتن آن‌ها به صورت جدا سخت است. یکی دیگر از کاربردهای تشخیص حروف جدا می‌تواند در تشخیص کلمات بعد از شکستن آن‌ها به حروف تشکیل دهنده باشد.

زیر-کلمه[1]/کلمه

زیر-کلمه قسمتی از کلمه است که بدون در نظر گرفتن علائم حروف می‌تواند در یک حرکت، بدون برداشتن قلم از روی صفحه نوشته شود. منظور از علائم حروف، نقطه، سرکش، دسته و مد است که برای نوشتن آن‌ها لازم است قلم از روی صفحه برداشته شود و دوباره در محل مناسب روی صفحه گذاشته شود. در تشخیص دستنوشته برون‌خط، کلمه شناسایی می‌شود در حالی که در تشخیص دستنوشته برخط، زیر-کلمه شناسایی می‌شود. علت این امر این است که در نوشته برخط، زیرکلمات دقیقاً قابل جداسازی از کلمات هستند در حالی که در نوشته برون‌خط شناسایی زیر-کلمه در کلمه، یک مرحله اضافی به سیستم تحمیل می‌کند. مشخص بودن زیرکلمات به دلیل امکان تشخیص تک حرکت قلم است که قبلاً گفته شد.

دو رویکرد کلی در زمینه تشخیص زیر-کلمه/کلمه وجود دارد:

رویکرد کلی نگر[2] :

در این رویکرد کل زیر-کلمه/کلمه به عنوان یک الگو در نظر گرفته می‌شود. در این رویکرد لازم است یک پایگاه داده از زیرکلمات/کلمات داشته باشیم تا زیر-کلمه/کلمه جدید را به عنوان یک الگو با الگوهای موجود در پایگاه داده مقایسه کنیم.

[1] Sub-word / word part
[2] Holistic

یکی از نقاط ضعف این رویکرد این است که تعداد زیرکلمات/کلمات قابل شناسایی محدود به همان نمونه‌های موجود در پایگاه داده است. در صورتی که بخواهیم تعداد زیرکلمات/کلمات قابل شناسایی را افزایش دهیم باید نمونه‌های بیشتری در پایگاه داده داشته باشیم که این به معنای داشتن تعداد کلاس‌های خروجی بیشتر و سخت‌تر شدن کار شناسایی الگو است.

رویکرد تجزیه‌ای[1]:

در این رویکرد، شناسایی زیر-کلمه با شناسایی تک تک حروف آن انجام می‌شود که خود به دو شکل انجام می‌شود. در رویکرد مبتنی بر قطعه‌بندی، هر زیر-کلمه به حروف سازنده‌اش شکسته می‌شود، که به این کار قطعه‌بندی[2] گفته می‌شود. البته کلمه قطعه‌بندی در مواردی که زیر-کلمه/کلمه به شکل‌های هندسی اولیه[3] مثل خط، کمان و حلقه شکسته می‌شود هم به کار می‌رود که این نوع قطعه‌بندی در رویکرد کلی نگر هم کاربرد دارد.

در روش مبتنی بر جستجوی حروف، برای شناسایی حروف، از مدل آموزش دیده شده آن‌ها استفاده می‌شود، به این صورت که از ابتدای زیر-کلمه به دنبال نقطه‌ای که بیشترین شباهت به یکی از حروف را دارد می‌گردیم و از آن نقطه به بعد به دنبال حرف بعد و به همین ترتیب تا وقتی آخرین حرف را پیدا کنیم. در هر دو روش چون تعداد کلاس‌های خروجی نسبت به رویکرد کلی‌نگر کمتر می‌شود کار شناسایی راحت‌تر است اما به دلیل خطای مرحله قطعه‌بندی، کارایی این رویکرد معمولاً از رویکرد کلی نگر کمتر است ولی در عوض کلمات قابل شناسایی به پایگاه داده محدود نیست و در صورت استفاده از پایگاه داده، افزایش نمونه‌های پایگاه داده منجر به سخت‌تر شدن شناسایی نمی‌شود و روی زمان اجرای الگوریتم هم زیاد تأثیر نمی‌گذارد.

استفاده از این رویکرد در نوشته پیوسته لاتین نسبت به نوشته پیوسته فارسی راحت‌تر است چون در نوشته پیوسته لاتین شکل حروف تغییر زیادی نمی‌کند و فقط حروف یک کلمه به هم متصل می‌شوند به همین دلیل کارهای انجام شده در زمینه قطعه‌بندی کلمات به حروف سازنده در زبان فارسی و عربی خیلی کم است.

رویکرد کلی‌نگر و تجزیه‌ای به صورت خلاصه در جدول ۱-۱ مقایسه شده‌اند.

[1] Analytical
[2] Segmentation
[3] Primitives

جدول ۱-۱: مقایسه رویکردهای کلی‌نگر و تجزیه‌ای

<table>
<tr><td>رویکرد</td><td>کلی نگر</td><td colspan="2">تجزیه‌ای</td></tr>
<tr><td>الگو</td><td>زیر-کلمه</td><td colspan="2">حروف</td></tr>
<tr><td>نمونه آموزشی</td><td>زیر-کلمات</td><td colspan="2">حروف</td></tr>
<tr><td>نمونه آزمایشی</td><td>زیر-کلمات</td><td colspan="2">زیر-کلمات</td></tr>
<tr><td>نمونه های قابل شناسایی</td><td>فقط موجود در پایگاه داده</td><td colspan="2">فرهنگ لغت متنی مستقل از پایگاه داده/نامحدود</td></tr>
<tr><td>روش‌ها</td><td>-</td><td>قطعه‌بندی</td><td>جستجوی حروف</td></tr>
<tr><td rowspan="4">مراحل</td><td>پیش پردازش</td><td colspan="2">پیش پردازش</td></tr>
<tr><td rowspan="2">استخراج ویژگی</td><td>قطعه بندی به حروف</td><td>استخراج ویژگی</td></tr>
<tr><td>استخراج ویژگی</td><td>جستجوی حروف</td></tr>
<tr><td>شناسایی</td><td colspan="2">شناسایی</td></tr>
</table>

۱-۴-۳- نوع ویژگی

ویژگی‌های قابل استفاده برای تشخیص دستنوشته برخط به را می‌توان به ویژگی‌های شکلی، ویژگی‌های ساختاری، ویژگی‌های حوزه تبدیل مثل تبدیل‌های فوریه، موجک و KLT و ویژگی‌های آماری تقسیم کرد.

ویژگی‌های شکلی، رایج‌ترین نوع ویژگی استفاده شده در کارهای مختلف است. این ویژگی نشان دهنده شکل دستنوشته است، به عنوان مثال زاویه بین نقاط متوالی دستنوشته یک ویژگی شکلی است.

ویژگی‌های ساختاری مربوط به ساختارهای نوشته مثل وجود یا عدم وجود حلقه، نقاط کمینه و بیشینه، نقاط تیز و موارد مشابه می‌شود. برای افزایش نرخ بازشناسی از ویژگی‌های ساختاری همراه با ویژگی‌های شکلی استفاده می‌شود.

ویژگی‌های شکلی و ساختاری را می‌توان به دو دسته محلی و سراسری تقسیم کرد. در صورتی برای استخراج ویژگی از کل نوشته استفاده شود، سراسری و در غیر این صورت محلی است. به عنوان مثال، جهت بردار متصل کننده نقطه اول و آخر یک حرف، یک ویژگی سراسری شکلی است و در حالی که وجود نقاط کمینه، بیشینه و تیز ویژگی محلی است اما محل رخ دادن آن‌ها در نوشته یک ویژگی سراسری است.

ویژگی‌های حوزه‌های تبدیل اطلاعات ملموسی ندارند و به همین دلیل کمتر در تشخیص دستوشته برخط مورد استفاده قرار گرفته‌اند. البته بعضی ویژگی‌ها مثل توصیف‌گر ممان‌های هو به دلیل عدم حساسیت به اندازه و چرخش در یکی از کارهای مطالعه شده مورد استفاده قرار گرفته که نتیجه مطلوبی داشته است.

ویژگی‌های آماری، ویژگی‌هایی هستند که با بررسی آماری نمونه‌های متعدد دستنوشته و تجزیه و تحلیل آن به دست می‌آیند، به عنوان مثال میانگین تعداد پیکسل‌ها در یک بلوک از دستنوشته برون خط یک ویژگی آماری است. از ویژگی‌های آماری در تشخیص برون خط استفاده شده است اما در تشخیص برخط به دلیل نوع داده که اطلاعات بیشتری از شکل نوشتن در دسترس است از این نوع ویژگی کمتر استفاده شده است و اغلب کارها با استفاده از ویژگی‌های شکلی و ساختاری انجام شده‌اند.

۱-۴-۴- مسئله انطباق[1] سیستم با نویسنده

دستنوشته یک فرد در زمان‌های مختلف می‌تواند متفاوت باشد یعنی ممکن است یک کلمه مشخص در دو زمان مختلف توسط یک فرد به شکل‌های متفاوتی نوشته شود، اما در حالت کلی کلمات یکسان توسط یک نویسنده نسبت به همان کلمات که توسط فرد دیگری نوشته شود شباهت بیشتری دارند، به همین دلیل در صورتی که برای آموزش سیستم تشخیص دستنوشته، فقط از دستنوشته یک فرد استفاده شود و در مرحله آزمایش هم از دستنوشته همان فرد استفاده شود، میزان تشخیص به طور چشمگیری افزایش می‌یابد.

مزیت سیستم شناسایی دستنوشته مستقل از نویسنده این است که نیاز به آموزش ندارد و هر کاربر دیگری هم می‌تواند بدون صرف وقت برای آموزش مجدد سیستم، از آن استفاده کند در حالی که سیستم شناسایی دستنوشته وابسته به نویسنده، فقط توسط نمونه‌های جمع آوری شده توسط یک نویسنده خاص آموزش می‌بیند و برای همان فرد قابل استفاده است و در صورت استفاده شخص دیگر از این سیستم، میزان تشخیص به شدت افت می‌کند در نتیجه هر کاربر جدیدی باید وقتی را برای آموزش سیستم صرف کند.

[1] Adaptation

برای توجیه پذیری یک سیستم تشخیص دستنوشته وابسته به نویسنده باید موارد زیر تحقق یابد:

۱. افزایش میزان بازشناسی به میزان قابل توجه: معمولاً انطباق سیستم با نویسنده منجر به افزایش قابل توجه میزان بازشناسی می‌شود، اما با در نظر گرفتن شرایط متفاوت سیستم مستقل از نویسنده با سیستم وابسته به نویسنده و اعمال روش‌های مناسب باز هم می‌توان میزان بازشناسی را بهبود داد. به عنوان مثال ممکن است استفاده از ویژگی‌های سرعت و فشار در حالت مستقل از نویسنده، اثر معکوس روی میزان بازشناسی داشته باشد ولی در حالت وابسته به نویسنده منجر به افزایش کارایی سیستم شود.

۲. زمان آموزش کوتاه: شناسایی دستنوشته به میزان نمونه‌های زیادی نیازمند است به همین دلیل ممکن است زمان آموزش طولانی شود و در صورتی که کاربرد سیستم در جایی باشد که نویسنده‌های زیادی وجود دارد، این زمان طولانی غیر قابل توجیه است. یکی از کارهایی که می‌توان برای کاهش زمان آموزش انجام داد ترکیب نمونه‌های آموزشی از قبل آماده شده که توسط نویسنده‌های مختلف نوشته شده‌اند با نمونه‌های گرفته شده در مرحله آموزش است. این کار ممکن است منجر به کاهش بهبود میزان بازشناسی شود، اما زمان آموزش را به شدت کاهش می‌دهد.

برای سیستم‌های تشخیص دستنوشته برخط می‌توان از آموزش سیستم در حین استفاده هم بهره برد، به این صورت که سیستم برای هر کلمه نوشته شده توسط کاربر، علاوه بر نزدیک‌ترین کلمه، لیستی از محتمل‌ترین کلمات شبیه به کلمه نوشته شده را هم بدهد، در صورتی که کلمه شناسایی شده اشتباه باشد ولی کلمه درست در لیست کاندیدهای بعدی باشد، کاربر کلمه درست را از لیست انتخاب می‌کند و سیستم، کلمه نوشته شده توسط کاربر را به پایگاه داده اضافه می‌کند و دورترین نمونه آن کلمه در پایگاه داده نسبت به کلمه وارد شده را حذف می‌کند[۶]. در صورت استفاده از این روش می‌توان مرحله آموزش را کاملاً حذف کرد و اجازه داد تا سیستم به تدریج در حین کار آموزش ببیند. به این ترتیب در اوایل کار با سیستم، خطاهای زیادی رخ می‌دهد ولی به مرور تعداد خطاها کمتر می‌شود.

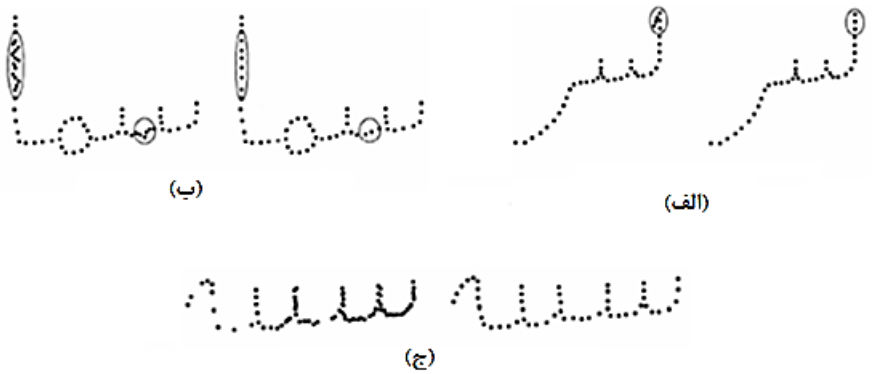
۱-۴-۵- پردازش‌های دیگر

پیش‌پردازش

معمولاً برای کاهش نویز و پیچیدگی‌های مربوط به تنوع دستخط، قبل از استخراج ویژگی یا قطعه‌بندی یک مرحله پیش‌پردازش لازم است. متداول‌ترین پیش‌پردازش، هموارسازی[1] نقاط است

[1] Smoothing

که بیشتر برای حذف نویز مربوط به وسیله دریافت داده برخط به کار می‌رود. در مواردی که طول بردار ویژگی وابسته به تعداد نقاط است، یک مرحله نرمالیزاسیون یا نمونه‌برداری مجدد[1] لازم است که معمولاً با کاهش تعداد نقاط همراه است اما ممکن است در مواردی با درون‌یابی نقطه جا افتاده بین دو نقطه، در آن محل نقطه جدیدی اضافه شود، همچنین در نرمالیزاسیون ممکن است عمل یکسان‌سازی فاصله نقاط هم انجام شود که باعث می‌شود اثر سرعت نوشتن از بین برود. انتقال مختصات و مقیاس‌بندی مجدد هم از دیگر پیش‌پردازش‌های ممکن هستند. یکی دیگر از پیش پردازش‌های مفید، حذف قلاب[2] است. در نوشتار برخط به دلیل حساسیت زیاد صفحه به جنس قلم با نزدیک کردن یا دور کردن قلم از صفحه در ابتدا و انتهای هر زیر-کلمه، تعدادی نقاط ناخواسته ایجاد می‌شود که نام‌گذاری آن به خاطر شکل قلاب مانندش است. همچنین قطعه‌بندی زیر-کلمه به واحدهای کوچک هم ممکن است جزو پیش‌پردازش محسوب شود. سه نوع پیش‌پردازش استفاده شده در [۷] در شکل ۱-۶ نشان داده شده است.



شکل ۱-۶: پیش‌پردازش‌های انجام شده در [۷] (الف) حذف قلاب؛ (ب) هموارسازی، (ج) نرمالیزاسیون

پس‌پردازش

ممکن است بعد از تشخیص حروف یا کلمات برای افزایش میزان بازشناسی کارهایی انجام شود، به عنوان مثال جمله، از نظر قواعد زبان یا مفهوم مورد پردازش قرار گیرد یا پاراگراف بندی نوشته تنظیم شود که این موارد، پس‌پردازش نام دارند.

۱-۴-۶- پیچیدگی محاسباتی و زمان اجرا

در تشخیص دستنوشته برون‌خط، محدودیت جدی در زمینه زمان اجرای الگوریتم وجود ندارد چون تمام متن نوشته شده به صورت یکجا به سیستم داده می‌شود و اصولاً در کاربردهای

[1] Resampling
[2] Dehooking

برون‌خط حتی اگر سرعت عامل مهمی باشد، امکان بلادرنگ[1] بودن سیستم وجود ندارد چون متن قبلاً نوشته شده است اما در تشخیص دستنوشته برخط به محض وارد کردن هر کلمه باید آن کلمه تشخیص داده شود به همین دلیل روش‌های شناسایی باید از نظر حجم محاسباتی طوری باشند که تشخیص به صورت بلادرنگ انجام شود.

معیار مشخصی برای مقایسه زمان اجرای روش‌های مختلف وجود ندارد چون سخت افزارهای استفاده شده متفاوت است. اما می‌توان به صورت بصری، بلادرنگ بودن یک سیستم را تشخیص داد. در صورتی که کاربر با سرعت معمولی بنویسد و سیستم در تشخیص کلمات وارد شده از کاربر عقب نیفتد، این سیستم بلادرنگ است.

۱-۵- هدف از تحقیق

هدف این تحقیق بررسی تشخیص دستنوشته برخط فارسی با قطعه‌بندی زیر-کلمه به حروف سازنده با استفاده از قواعد ساده بر اساس شکل دستنوشته و تشخیص حروف قطعه‌بندی شده با استفاده از مدل مخفی مارکوف است. بدین منظور از پایگاه داده حروف و زیر-کلمات برخط دانشگاه تربیت مدرس[۸] استفاده شده است. این پایگاه داده در [۴] بررسی شده است.

۱-۶- ساختار پایان‌نامه

فصل ۲ به مرور مهم‌ترین کارهای انجام شده دیگران درباره تشخیص دستنوشته برخط عربی و فارسی اختصاص یافته است. این کارها بر اساس الگوی مورد شناسایی و رویکرد تحقیق، در چهار گروه دسته‌بندی شده است و سعی شده از بین کارهای مشابه، بارزترین و جدیدترین آن‌ها انتخاب شود و کارهای نزدیک‌تر به روش این تحقیق با جزئیات بیشتری بیان شود.

در فصل ۳ تئوری‌های مرتبط با موضوع این تحقیق به صورت مختصر بررسی می‌شود. برای مطالعه بیشتر درباره موضوعات این فصل لازم است به مراجع انتهای تحقیق رجوع شود اما اطلاعات آورده شده در این فصل برای درک روش کار ابزارهای استفاده شده و دلیل استفاده آن‌ها کافی است.

در فصل ۴ روش مورد استفاده در این تحقیق به تفصیل توضیح داده شده است. این فصل را می‌توان به بخش‌های آماده‌سازی داده‌ها، روش قطعه‌بندی، روش تشخیص حروف و در نهایت، ترکیب قطعه‌بندی و تشخیص حروف برای تشخیص زیر-کلمه تقسیم کرد.

نتایج شبیه‌سازی، مقایسه با کارهای گذشته و تحلیل نتایج در فصل ۵ می‌آید. این کار برای هر کدام از بخش‌های فصل ۴ به صورت جداگانه انجام شده است. مقایسه با کارهای گذشته به دلیل

[1] Real-time

محدودیت کارهای مشابه و تفاوت در پایگاه‌های داده و تفاوت‌های دیگر فقط برای اطلاع آورده شده است و معیار مناسبی برای مقایسه کارایی روش‌ها نیست.

در نهایت نتیجه‌گیری و پیشنهادات در فصل ۶ بررسی خواهد شد که خلاصه‌ای از کار انجام شده و ویژگی‌های آن، مشکلات پیش رو و پیشنهادها برای رفع این مشکلات و بهبود نتیجه است.

بعضی از جزئیات مورد نیاز که امکان آوردن آن‌ها در فصول نبود، در ضمیمه‌ها آورده شده است.

فصل ۲

مروری بر کارهای گذشته

فصل ۲- مروری بر کارهای گذشته

در این بخش به مرور مهم‌ترین کارهای انجام شده در زمینه تشخیص دستنوشته برخط عربی و فارسی می‌پردازیم. کارهای انجام شده را از نظر رویکرد می‌توان به چند دسته تقسیم کرد:

- قطعه‌بندی زیر-کلمه به حروف یا اجزای اولیه مبتنی بر قواعد ساده
- تشخیص حروف جدا یا قطعه‌بندی شده
- شناسایی زیر-کلمات مبتنی بر تشخیص حروف
- شناسایی زیر-کلمات با در نظر گرفتن هر زیر-کلمه به عنوان یک الگو

۲-۱- قطعه‌بندی زیر-کلمه به حروف یا اجزای اولیه مبتنی بر قواعد ساده

در این رویکرد، ابتدا زیر-کلمه به قطعات کوچک‌تر شکسته می‌شود، این قطعات ممکن است حروف سازنده زیر-کلمه و یا قسمتی از یک حرف باشند.

ضیف الله و همکاران روشی برای قطعه‌بندی اضافی[1] پیشنهاد کرده‌اند که همه نقاط قطعه‌بندی صحیح را شناسایی می‌کند ولی ممکن است بین ۵۰ تا ۶۰ درصد نقاط اضافی هم پیدا کند، این روش روی ۱۵۰ کلمه عربی آزمایش شده است. ابتدا پیش‌پردازش‌های هموارسازی، حذف نقاط تکراری و حذف قلاب روی ورودی انجام شده است. سپس با توجه به این که در نوشته عربی، بین حروف پاره‌خطی افقی از راست به چپ وجود دارد، تمام قطعاتی که بین نقاط متوالی آن جهت نوشته از راست به چپ بوده و زاویه خط اتصال دهنده بین نقاط با محور افقی کمتر از ۳۰± درجه بوده است، در نظر گرفته شده است. سپس بخشی از این قطعه که بالا یا پایین آن نوشته شده باشد، حذف شده است. در ادامه، نقاطی از قطعه که در محدوده ۲۵± درجه خط عمود در آن نقطه، نوشته وجود دارد، حذف شده است. در مرحله بعد، فواصل قطعه‌بندی مجاور هم ادغام شده است سپس نقطه وسط فواصل قطعه‌بندی به عنوان نقاط قطعه‌بندی در نظر گرفته شده است.[۹]

پُتروس و همکاران برای ساده شدن قطعه‌بندی با استفاده از یک مرحله در پیش‌پردازش، ابتدا یک شکل مستطیل‌وار از دستنوشته می‌سازند تا تشخیص نواحی قطعه‌بندی که در این مقاله خط پیوند[2] نامیده می‌شود راحت‌تر شود. این مرحله بعد از یک مرحله هموارسازی و نرمالیزاسیون انجام می‌شود. در مرحله نرمالیزاسیون ابتدا زاویه خط واصل بین هر دو نقطه متوالی به یکی از جهت‌های ۸ گانه تبدیل می‌شود که این بردارها مجموعه S را تشکیل می‌دهند، مجموعه S از زیرمجموعه‌های

[1] Over segmentation
[2] Ligature

S_jتشکیل شده است که $j = \{1,2,\dots,8\}$ شماره مربوط به جهت بردار بین نقاط متوالی دستنوشته است. هر مجموعه S_j خود از زیرمجموعه‌های S_{ij} تشکیل شده که شامل بردارهای هم جهت متوالی است.[۱۰]

مجموعه‌های S_j که احتمال رخداد آن‌ها در کلمه کمتر از λ باشد حذف می‌شوند:

$$\lambda = \frac{1}{10} log(\frac{N}{2}) \qquad (۱-۲)$$

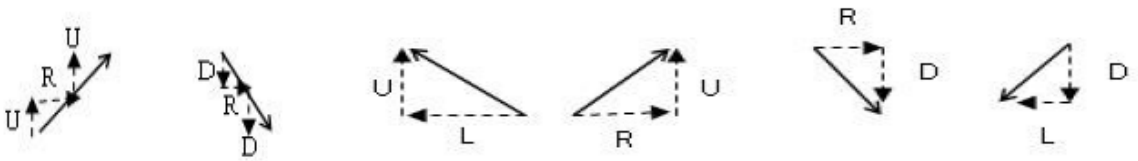
که N تعداد اعضای مجموعه S است. سپس در مرحله اصلاح، مجموعه‌های S_{ij} که کمتر از ε عضو دارند هم حذف می‌شوند. مقدار $\varepsilon = 2$ به صورت تجربی به دست آمده است.[۱۰]

هشت جهت شکل ۲-۱ با توجه به جهت حرکت در هر نقطه تعریف می‌شوند و دستنوشته با این تعریف کد می‌شود.

$$g \in W_{(U,D,L,R)} : g = \begin{cases} U & \Delta x = 0, \Delta y > 0 \\ D & \Delta x = 0, \Delta y < 0 \\ L & \Delta x > 0, \Delta y = 0 \\ R & \Delta x < 0, \Delta y = 0 \\ RU & \Delta x < 0, \Delta y > 0 \\ RD & \Delta x < 0, \Delta y < 0 \\ LU & \Delta x > 0, \Delta y > 0 \\ LD & \Delta x > 0, \Delta y < 0 \end{cases}$$

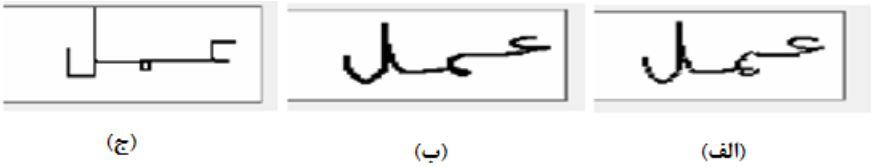
شکل ۲-۱: کدهای جهتی در [۱۰]

سپس مطابق شکل ۲-۲ هر جهت قطری به دو جهت اصلی تبدیل می‌شود، جهت‌های قطری متوالی هم مطابق شکل سمت چپ کد می‌شوند.



شکل ۲-۲: تبدیل جهت‌های قطری به اصلی [۱۰]

به این ترتیب، دستنوشته به صورت کدی شامل جهت‌های اصلی درمی‌آید، در مرحله بعد جهت‌های یکسان متوالی که طول کمتر از ۲ (تجربی) دارند با جهت همسایه‌ای که طول بیشتری دارد جایگزین می‌شود. حاصل این مراحل در شکل ۲-۳ نشان داده شده است.



شکل ۲-۳: (الف) دستنوشته اصلی، (ب) پس از هموارسازی، (ج) شکل مستطیل‌وار[۱۰]

همان طور که دیده می‌شود، حاصل این عملیات شکلی است که از قسمت‌هایی تشکیل شده است که فقط یک جهت دارند و تشخیص حرف و خط پیوند در آن بسیار راحت تر است. تمام حروف عربی بعد از این که به این شکل در آورده شوند در جهت عمودی دارای نقاط مشترک هستند از این رو هر قسمتی که در جهت عمودی با هیچ نقطه دیگری اشتراک نداشته باشد خط پیوند در نظر گرفته می‌شود و بقیه قسمت‌ها هم حرف در نظر گرفته می‌شوند. نقطه وسط هر قطعه به عنوان نقطه قطعه‌بندی در نظر گرفته می‌شود. خروجی این مرحله برای یک نمونه در شکل ۲-۴ آورده شده است.

شکل ۲-۴: خروجی مرحله تشخیص خط پیوند و حرف، (الف) شکل مستطیل‌وار زیر-کلمه، (ب) خط پیوندهای تشخیص داده شده (قسمت‌های توپر) [۱۰]

قسمتی از حرف اول و آخر هم ممکن است به عنوان خط پیوند شناسایی شوند که به خاطر شکل بعضی از حروف عربی است، برای حذف خط پیوند اشتباه در حرف اول، اگر در بازه مربوط به خط پیوند شناسایی شده، تغییرات عمودی نداشته باشد، حرف وگرنه خط پیوند است. برای حذف خط پیوند اشتباه در حرف آخر، در هر نقطه از خط پیوند تشخیص داده شده زاویه بردار واصل آن نقطه تا آخرین نقطه دستنوشته کوچک‌تر از ۷۰ درجه باشد و مختصات عمودی یا کاملاً صعودی و یا کاملاً نزولی باشد.

نتیجه کلی این روش ۹۶/۲ درصد قطعه‌بندی صحیح بوده است که بر روی کلماتی که توسط ۲۰ نویسنده نوشته شده و شامل ۲۳۰۰ حرف است آزمایش شده است.[۱۰]

عِراقی و عبدالعظیم بعد از پیش پردازش‌های هموارسازی و نرمالیزاسیون، روشی برای پیدا کردن محل خط زمینه ارائه کرده‌اند که در تخصیص علامت‌ها به حروف استفاده می‌شود، سپس برای قطعه‌بندی زیر-کلمه‌های پایگاه داده ADAB به نویسه‌هایی که شامل حرف یا قسمتی از حرف هستند یک روش مستقل از خط زمینه ارائه کرده است که شامل مراحل زیر است: [۱۱]
نقاطی که زاویه خط متصل کننده آن به نقطه بعد با محور افقی کمتر از ۲۲ درجه باشد به عنوان نقاط اولیه قطعه‌بندی در نظر گرفته می‌شوند. سپس نقاطی که بالای آن با زاویه ۲۰± درجه نوشته وجود دارد حذف می‌شود. نقاط متوالی به عنوان یک فاصله قطعه‌بندی در نظر گرفته می‌شوند و فواصل قطعه‌بندی که فاصله کمتر از ۵ نقطه دارند به هم متصل می‌شوند تا یک فاصله قطعه‌بندی بزرگ‌تر تشکیل دهند. هر نقطه‌ای که زاویه آن با نقطه بعد بین ۲۱۰ تا ۳۳۰ درجه باشد، رو به پایین و در صورتی که بین ۳۰ تا ۱۵۰ درجه باشد، رو به بالا فرض می‌شود، مراحل ذکر شده برای نقاط قطعه‌بندی برای نقاط رو به پایین و رو به بالا هم انجام می‌شود. فواصل رو به پایین و رو به بالا که کمتر از ۳ نقطه داشته باشند به عنوان نویز تلقی شده و حذف می‌شوند، نقاط وسط فواصل باقی مانده به ترتیب، مرکز رو به پایین و مرکز رو به بالا نامیده می‌شوند.[۱۱]
نقاط قطعه‌بندی که قبل از آن یک مرکز رو به بالا (یا شروع زیر-کلمه) و بعد یک مرکز رو به پایین و بعد از آن یک مرکز رو به بالا و سپس یک مرکز رو به پایین (یا انتهای زیر-کلمه) باشد قبول شده و بقیه حذف می‌شوند. همچنین نقاط قطعه‌بندی که قبل از آن یک مرکز رو به بالا و بعد از آن مرکز رو به پایین باشد به عنوان نقاط تپّه‌ای[1] شناخته شده و حذف می‌شوند، این نقاط روی دندانه‌های حروف قرار دارند. سپس فواصل قطعه‌بندی که در روی حلقه حروف قرار دارند با شمارش نقاطی که ادامه نوشته زیر آن‌ها قرار دارد و قرار دادن یک مقدار آستانه حذف می‌شوند. در انتها هم فواصل قطعه‌بندی که در ابتدا یا انتهای زیر-کلمه قرار دارند حذف می‌شوند.[۱۱]

۲-۲- تشخیص حروف جدا یا قطعه‌بندی شده

در زمینه تشخیص حروف نسبت به قطعه‌بندی و یا تشخیص کل زیر-کلمه به عنوان یک الگوی واحد کارهای بیشتری انجام شده است. در این قسمت به مرور تعدادی از مهم‌ترین این کارها می‌پردازیم.

کلاسن برای تشخیص حروف عربی از ۳۴۰۰ حرف جدای نوشته شده توسط ۲۵ نفر استفاده کرده است. ابتدا نقاط مهم با پیدا کردن محل تغییر جهت تقعر دستنوشته استخراج شده است که در شناسایی حرف مفیدتر است، بعد از پیش‌پردازش‌های نرمالیزاسیون و انتقال، برای استخراج ویژگی از

[1] Hill

شبکه SOM[1] استفاده شده است. ۱۵ کلاس خروجی با استفاده از شبکه عصبی طبقه‌بندی شده‌اند که نتیجه کلی ۹۵ درصد بوده است.[۱۲]

سلیمانی باغشاه و همکاران از ترکیب رویکردهای فازی، ساختاری و یادگیری با استفاده از شبکه عصبی یا روش‌های آماری برای بازشناسی حروف جدای فارسی استفاده می‌کنند. ابتدا یک مرحله پیش‌پردازش روی ورودی انجام می‌شود تا نقاطی که فاصله کمی با نقاط مجاور دارند حذف شوند، سپس قطعه‌بندی حرف در سه مرحله انجام می‌شود، حلقه‌ها با پیدا کردن محل تقاطع دستنوشته به عنوان یک قطعه در نظر گرفته می‌شوند، برای قطعه‌بندی قسمت‌های باقی مانده، نقاطی که بیشینه محلی هستند و تغییر زاویه در آن‌ها زیاد است همچنین نقاط عوض شدن جهت تقعر که تغییر زاویه در آن‌ها زیاد است به عنوان محل قطعه‌بندی معرفی می‌شوند.[۱۳]

ویژگی‌های استخراج شده از هر قطعه شامل این موارد است: جهت بردارهای متصل کننده ابتدا به انتها، ابتدا به مرکز ثقل و انتها به مرکز ثقل قطعه، میزان انحنای قطعه که از اختلاف زاویه بین بردارهای ابتدا و انتها به مرکز ثقل به دست می‌آید، جهت انحنا، میزان حرکت نسبی در جهت افقی و عمودی و نسبت طول به عرض مستطیل دربرگیرنده قطعه.[۱۳]

سپس این ویژگی‌ها با استفاده از مجموعه‌ای از قواعد، فازی شده‌اند که برای تشخیص بدنه حرف استفاده می‌شوند، همچنین علامت حروف هم به عنوان یک ویژگی اضافی در تشخیص حرف استفاده می‌شود. برای طبقه‌بندی از یک شبکه FLVQ[2] استفاده شده است و نتیجه نهایی بازشناسی ۹۰/۲۷ درصد بوده است و ۳ درصد هم وازدگی داشته است.[۱۳]

رضوی و کبیر حروف جدای فارسی را بر اساس علامت به ۱۲ گروه تقسیم کرده‌اند، برای تشخیص یک حرف، ابتدا لازم است علامت آن شناسایی شود تا به یکی از این ۱۲ گروه نسبت داده شود، سپس بدنه حرف وارد شده فقط با حروف آن گروه مقایسه می‌شود. برای تشخیص علامت، ابتدا محل علامت از نظر بالا یا پایین حرف بودن بررسی می‌شود، سپس برای شناسایی علامت از شبکه عصبی استفاده می‌شود. برای تشخیص بدنه حرف، مجموع فاصله اقلیدسی نقاط حرف نرمالیزه شده با نقاط حرف‌های نمونه گروه مقایسه می‌شود و ورودی به کلاس حرفی که کمترین فاصله را دارد نسبت داده می‌شود. میزان صحت این روش ۹۳/۳ درصد است.[۱۴]

ساجدی و همکاران حروف جدای فارسی را بر اساس شکل بدنه به ۱۸ گروه تقسیم کرده‌اند و برای شناسایی علامت از شاخص میانگین گستردگی آن‌ها روی محورهای افقی و عمودی و برای تشخیص مکان نقاط نسبت به بدنه اصلی از شاخص میانگین مؤلفه عمودی نقاط نسبت به میانگین مؤلفه عمودی بدنه اصلی حروف استفاده شده است.[۱۵]

[1] Self-Organizing Maps
[2] Fuzzy Learning Vector Quantization

تفاضل زوایای نقاط متوالی و مشتق‌های زمانی مرتبه اول و دوم مؤلفه‌های محورهای افقی و عمودی به عنوان ویژگی‌های استخراج شده جهت استفاده در طبقه‌بند مدل مخفی مارکوف به کار رفته‌اند. بهترین نتیجه برای مدل مخفی مارکوف با ۵ حالت است که هر حالت از ۳ گاوسی ۳ بعدی (ویژگی‌های گفته شده) تشکیل شده است و برابر با ۹۰/۸۲ درصد برای تشخیص گروه حرف و ۸۸/۸۷ درصد برای تشخیص حرف است.[۱۵]

فرکی و پالهنگ برای بازشناسی حروف جدای فارسی از ۵۳۹ حرف نوشته شده توسط ۹ نفر برای آموزش و ۴۳۱ حرف نوشته شده توسط ۸ نفر دیگر برای آزمایش استفاده کرده‌اند، در این روش حرف‌های جدای فارسی بر اساس تعداد بخش‌ها به ۴ گروه تقسیم شده‌اند. برای ساده کردن بازشناسی بدنه حروف، ابتدا یک گروه بندی بر اساس علامت انجام می‌شود به این ترتیب که حرف ورودی بر اساس تعداد بخش‌ها به یکی از ۴ گروه گفته شده تعلق خواهد داشت، در مرحله بعد با شناسایی علامت‌ها، حروف کاندید کمتر می‌شوند. روند شناسایی علامت‌ها مشابه شناسایی بدنه حروف است و فقط تنظیمات مقادیر متفاوت دارد.[۳]

قبل از استخراج ویژگی، بر روی داده‌ها یک مرحله پیش‌پردازش شامل یکسان کردن اندازه در جهت افقی و دوباره نمونه‌برداری انجام می‌شود. سپس با الگوریتم داگلاس و پکر[1] [۳] نقاط مهم دستنوشته برای مرحله استخراج ویژگی به دست می‌آیند. زاویه بین این نقاط به ۸ جهت رقمی می‌شوند، همچنین نقاط اوج و دره راست و چپ هم به عنوان ویژگی به بردار ویژگی اضافه می‌شوند، برای طبقه‌بندی از مدل مخفی مارکوف استفاده می‌شود، طول بردار مشاهدات و حالت برای هر حرف به صورت جدا محاسبه می‌شود تا بهترین نتیجه به دست آید. همچنین برای کاهش خطا در مرحله آموزش از الگوریتم باوم-ولچ[2] اصلاح شده استفاده می‌شود تا هیچ وزن صفری وجود نداشته باشد. خطای آزمون این روش ۵/۱ درصد است.[۳]

قدس و کبیر روشی مبتنی بر درخت تصمیم‌گیری[3] و با تأکید بر روش استخراج ویژگی ارائه کرده‌اند. در این روش ۲۴ ویژگی برای حروف جدای فارسی استخراج شده است که همه این ویژگی‌ها از روی شکل حروف در گروه‌های ۹ گانه شکلی به دست آمده‌اند. تعداد ویژگی‌ها برای هر گروه از حروف متفاوت است و به میزان پیچیدگی شکل آن گروه بستگی دارد. درخت تصمیم گیری پس از آموزش، گروه‌بندی را با دقت ۹۴ درصد انجام داده است.[۱۶]

[1] Douglas and Peucker
[2] Baum-Welch
[3] Decision Tree

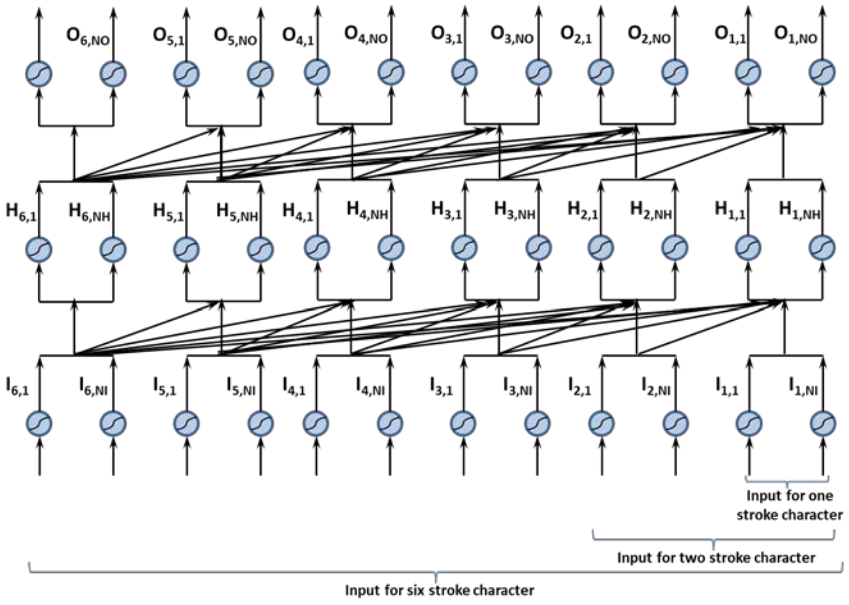
۲-۳- شناسایی زیر-کلمات مبتنی بر تشخیص حروف

در روش متداول این رویکرد، حروف زیر-کلمه بدون نیاز به قطعه‌بندی اولیه شناسایی می‌شوند، به این ترتیب که از شروع زیر-کلمه شروع به جستجوی حرفی می‌کند که بیشترین انطباق را داشته باشد، سپس از پایان قسمتی از زیر-کلمه که منطبق با پایان حرف شناسایی شده است شروع به جستجوی حرف بعدی می‌کند و این کار را تا پایان زیر-کلمه انجام می‌دهد.

مزیت این روش در این است که به قطعه‌بندی جداگانه نیازی ندارد و از مدل آموزش یافته حروف استفاده می‌کند در حالی که در روش دوم که قطعه‌بندی مبتنی بر قواعد است، اطلاعی از شکل حروف ندارد اما مهم‌ترین عیب این روش جستجوی مدل حروف این است که خطا در پیدا کردن اولین حرف تا انتهای زیر-کلمه ادامه پیدا می‌کند، ضمن این که بعضی روش‌ها از نظر زمانی و پیچیدگی محاسباتی برای داده برخط که باید به صورت بلادرنگ شناسایی شود مناسب نیستند.

در میان کارهای گذشته با رویکرد تشخیص حروف زیر-کلمه، هیچ موردی که مبتنی بر قطعه‌بندی باشد پیدا نشد و کارهای مرور شده همگی بر اساس جستجوی مدل حروف هستند.

علیمی با توجه به این که هر حرف عربی می‌تواند حداکثر در ۶ حرکت (که از یک جابجایی ساده با مشخصه سرعت با تابع ناقوسی شکل غیر متقارن به وجود می‌آید) نوشته شود از شبکه عصبی سلسله مراتبی شکل ۲-۵ برای شناسایی حرف استفاده می‌کند. هر حرف دارای ۶ بردار ویژگی با n پارامتر است و بردارهای ویژگی حرکت‌های اضافی با صفر پر می‌شود.[۱۷]



شکل ۲-۵: شبکه عصبی سلسله مراتبی برای شناسایی حرف در [۱۷]

برای هر حرف در کلمه نوشته شده بین ۱ تا ۶ حرکت وجود دارد، هر حرکت ۵۷ حالت ممکن دارد، یا یکی از ۵۶ شکل حروف الفبای عربی (در ۴ حالت جدا، اول، وسط و آخر زیر-کلمه) و یا ۰ به معنای این که این حرکت قسمتی از یک حرف است، این احتمال به صورت یک مقدار فازی است. یک الگوریتم ژنتیک مسئول پیدا کردن بهینه‌ترین حالت در بین ترکیب‌های مختلف ممکن است. نتیجه نهایی بازشناسی بر روی کلمه "عزّالدّین" که ۱۰۰ بار توسط یک نفر نوشته شده است، ۸۹ درصد است. این نتیجه بدون در نظر گرفتن علامت‌ها و اطلاعات جهتی به دست آمده است.[۱۷]

AL-HABIAN و اصاله زیر-کلمه عربی را به حرکت‌هایی که سازنده حروف عربی در مکان‌های مختلف زیر-کلمه هستند تقسیم کرده‌اند که به عنوان واحدهای شناسایی استفاده می‌شوند، از ویژگی‌های Δx، Δy و sin و cos زاویه بین نقاط متوالی حرکت برای آموزش مدل مخفی مارکوف جهت آموزش و شناسایی این حرکت‌ها استفاده شده است. [۱۸]

برای شناسایی این حرکت‌ها یک پنجره به طول ۱۰ نمونه روی ویژگی‌های استخراج شده حرکت می‌کند که دارای ۵۰ درصد همپوشانی است و مجموعه ویژگی‌ها را به طبقه‌بند مدل مخفی مارکوف می‌دهد. در بین ترکیب‌های مختلف، حالتی که بالاترین مقدار را داشته باشد انتخاب می‌شود. سپس با استفاده از یک درخت تصمیم گیری ساده، حالت‌های غیر معتبر حذف می‌شوند (مثل وجود حرکتی در ابتدای زیر-کلمه که در ابتدای حروف اول وجود ندارد) سپس در صورتی که شکل نامعتبری از یک حرف در زیر-کلمه پیدا شود، با نادیده گرفتن این شکل اشتباه، الگوریتم یک بار دیگر برای پیدا کردن شکل درست تکرار می‌شود. نتیجه نهایی ۷۸/۲۵ درصد تشخیص صحیح و ۳۴ درصد درج[1] برای حرکت‌ها و ۷۵/۲۵ درصد تشخیص صحیح و ۳۵/۷۵ درصد درج برای حروف است. منظور از درج، حرکت‌ها یا حروفی است که در نوشته وجود نداشته است اما سیستم تشخیص داده است.[۱۸]

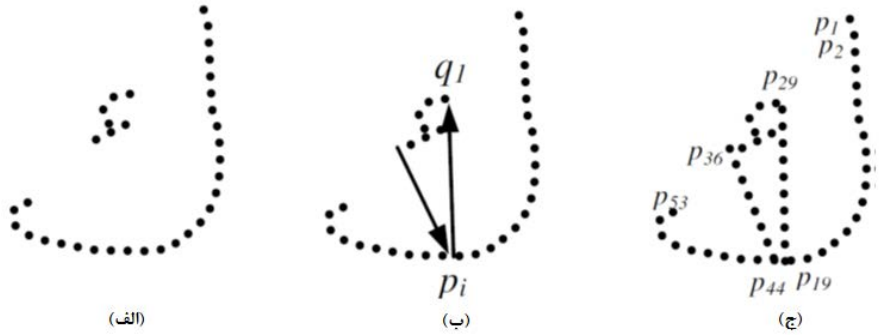
Biadsy و همکاران برای هر حرف یک مدل مخفی مارکوف را با استفاده از حروف از قبل قطعه‌بندی شده آموزش داده‌اند. ابتدا حرف یا زیر-کلمه با استفاده از الگوریتم داگلاس و پکر به قطعات کوچک تقسیم شده است. از هر کدام از این قطعات سه ویژگی استخراج می‌شود، ویژگی اول زاویه بردار اتصال دهنده بین نقاط متوالی قطعه و ویژگی دوم زاویه بردار اتصال دهنده بین نقاط باقی‌مانده بعد از اعمال الگوریتم داگلاس و پکر (با حد آستانه کمتر از مرحله قطعه‌بندی اولیه) با محور افقی است و ویژگی سوم وجود یا عدم وجود حلقه در قطعه است که یک مقدار دودویی[2] است.[۱۹]

چون پایگاه داده این مقاله از کلمات تشکیل شده است، علامت‌های زیر-کلمات شناسایی و به زیر-کلمه مربوط نسبت داده می‌شوند. با فرض این محدودیت که علامت‌های زیر-کلمه بلافاصله بعد از نوشتن آن گذاشته می‌شود، عمل انتساب علامت‌ها به زیر-کلمات انجام می‌شود. برای تشخیص علامت از بدنه زیر-کلمه بعدی، از این واقعیت استفاده می‌شود که اندازه نقاط کوچک است و چهارضلعی

[1] Insertion
[2] Binary

دربرگیرنده نقاط تقریباً مربعی شکل است و بقیه علامت‌ها در سمت راست (قبل) و یا روی بدنه زیر-کلمه قرار می‌گیرند.[۱۹]

سپس حرکت‌های علامت‌ها به بدنه اصلی متصل می‌شوند، بدین منظور نقطه اول حرکت (q_1)، به نقطه‌ای از بدنه زیر-کلمه که مختصات عمودی نزدیک‌تری دارد (p_i) وصل می‌شود، آخرین نقطه حرکت نیز به نقطه بعدی بدنه زیر-کلمه (p_{i+1}) وصل می‌شود. بین نقاط بدنه و حرکت، نقاط جدید اضافه می‌شوند تا بدنه و حرکت‌های آن با هم به یک حرکت تبدیل شوند. این روال در شکل ۲-۶ نشان داده شده است.[۱۹]

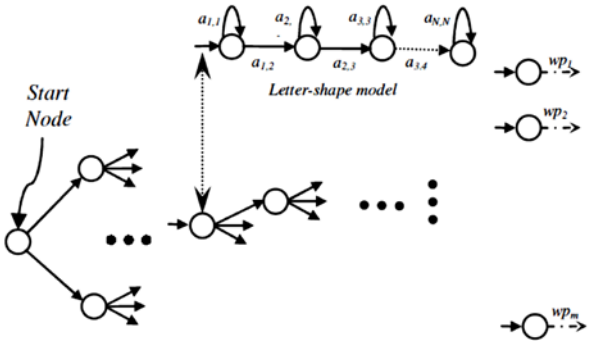


شکل ۲-۶: (الف) زیر-کلمه با یک علامت (ب) اتصال حرکت به بدنه زیر-کلمه (ج) افزودن نقاط متصل کننده حرکت و بدنه زیر-کلمه[۱۹]

با اتصال حرکت‌ها به بدنه زیر-کلمه، علامت‌های حروف هم قسمتی از بدنه جدید آن‌ها هستند که از همین شکل جدید آن‌ها برای آموزش مدل مخفی مارکوف استفاده می‌شود. بردار مشاهده مدل مخفی مارکوف باید دارای مقادیر صحیح مثبت باشد، به همین دلیل یک مرحله گسسته سازی با استفاده از کدهای فریمن ۱۶ و ۸ جهتی برای ویژگی‌های اول و دوم انجام شده است. این جهت‌ها و ویژگی دودویی سوم ۲۵۶ مقدار گسسته و وجود حرکت و موقعیت آن نسبت به بدنه زیر-کلمه هم ۴ مقدار دیگر گسسته دیگر دارند که در مجموع ۲۶۰ مشاهده مختلف را تشکیل می‌دهند. مدل مخفی مارکوف دارای توپولوژی چپ به راست است و تعداد حالت‌ها برای هر حرف بسته به میزان پیچیدگی هندسی آن بین ۵ تا ۱۱ انتخاب شده است.[۱۹]

با استفاده از فرهنگ لغت متنی، دو مجموعه فرهنگ کلمات D_i و فرهنگ زیر-کلمات $WPD_{i,j}$ تشکیل شده است. هر مجموعه D_i شامل تمام کلمات دارای i زیر-کلمه است و هر مجموعه $WPD_{i,j}$ شامل زیر-کلمات j ام D_i است. برای تشخیص کلمه ورودی با k زیر-کلمه، زیر-کلمه n ام، فقط با زیر-کلمات مجموعه $WPD_{k,n}$ مقایسه می‌شود.[۱۹]

تشخیص زیر-کلمه با استفاده از مدل حروف آن انجام می‌شود. به این ترتیب برای هر زیر-کلمه، یک شبکه از مدل‌های مخفی مارکوف حروف آن زیر-کلمه تشکیل می‌شود. این شبکه در شکل ۲-۷ نشان داده شده است.



شکل ۲-۷: مجموعه شبکه‌های زیر-کلمات، هر گره در این شبکه، یک مدل مخفی مارکوف مربوط به حرف زیر-کلمه است و اتصالات بین گره‌ها دارای وزن نیستند.[۱۹]

زیر-کلمه‌ای که بردار مشاهده زیر-کلمه ورودی، خروجی شبکه مربوط به آن را بیشینه کند انتخاب می‌شود. بردار ویژگی زیر-کلمه از اولین مشاهده تا زمانی که مدل اولین حرف بیشترین احتمال را تولید کند به عنوان بردار مشاهده اولی حرف زیر-کلمه در نظر گرفته می‌شود، از ادامه بردار مشاهده تا زمانی که مدل حرف بعد در شبکه زیر-کلمه، بیشترین احتمال را تولید کند به عنوان بردار مشاهده دومین حرف در نظر مگرفته می‌شود و این روال تا پایان بردار مشاهده ادامه پیدا می‌کند. احتمال تولید زیر-کلمه برابر با حاصل‌ضرب احتمال خروجی مدل حروف آن تعریف می‌شود و کلمات مجموعه D_k با حاصل‌ضرب خروجی شبکه‌های زیر-کلمات آن‌ها امتیازبندی می‌شوند.[۱۹]

حلاوتی و باغشاه با استفاده از ماشین مقایسه کشسان فازی (فم[1])، روشی برای تشخیص دستنوشته برخط فارسی ارائه کرده‌اند که بر خلاف مدل مخفی مارکوف نیاز به پیش‌فرض‌های احتمالاتی ندارد و با وجود دقت بازشناسی تقریباً مساوی، مقاومت بیشتری در مقابل تغییرات محیطی از خود نشان می‌دهد و سرعت مناسبی نیز دارد.[۲۰]

با استفاده از زاویه بردارهای بین نقاط متوالی، هر حرکت به قسمت‌هایی تقسیم شده است که سه نوع کلی خط، کمان و نیم حلقه هستند. با توجه به فاصله چند نقطه اول از نوشته کاربر، واحدی برای طول مشخص می‌شود که بر اساس این واحد به هر قطعه یک طول فازی داده می‌شود. یک ویژگی فازی دیگر، جهت حرکت است که از زاویه خط بین نقاط ابتدایی و انتهایی به دست می‌آید و دارای ۸ مقدار است. ویژگی غیر فازی انحنا هم برای کمان‌ها و نیم حلقه‌ها تعریف می‌شود به این صورت که اگر نقطه ثقل در سمت چپ نقطه شروع قرار داشته باشد، ساعت‌گرد و در غیر این صورت پادساعت‌گرد است.[۲۱]

چون تعداد قطعات حرکت‌ها لزوماً مساوی نیستند، به ساختاری برای مقایسه آن‌ها برای تطبیق الگو نیاز است. مطابق شکل ۲-۸، هر قطعه در سه بخش با قطعه بعدی مقایسه می‌شود.

[1] Fuzzy Elastic Machine (FEM)

شکل ۲-۸: مقایسه شباهت دو قطعه[۲۱]

شباهت A به B به صورت ۷۰، ۳۰، ۰ نشان داده می‌شود به این معنا که ۳۰ درصد از A شبیه به B است اما بعد از ۷۰ درصد دیگر که شباهتی ندارد قرار گرفته است. به این ترتیب یک جدول جستجو ساخته می‌شود که از مقایسه انواع مختلف قطعه‌ها با هم به دست می‌آید. در شکل ۲-۹ قسمتی از جدول مربوط به مقایسه خط با انواع مختلف قطعه‌ها آمده است.

	Input Type	Angle Difference[a]	Extra Before	Matching	Extra After
1	Line	0	0%	100%	0%
2	Line	+1 or −1	25%	50%	25%
3	Arc	None	5%	90%	5%
4	Arc-C.	+1	0%	75%	25%

شکل ۲-۹: جدول جستجوی مقایسه انواع قطعات[۲۱]

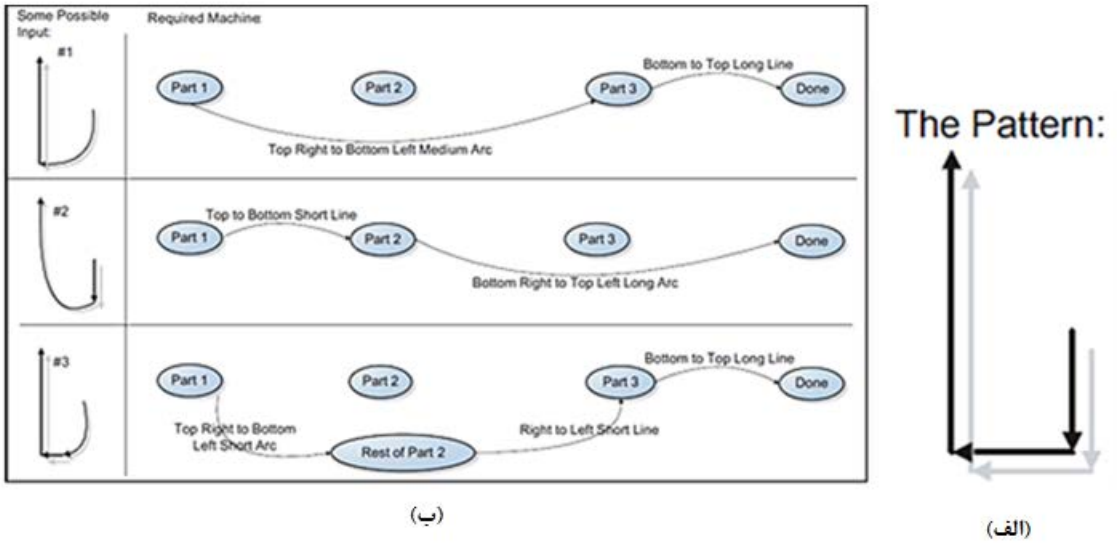
اختلاف زاویه ۱+ یعنی خط یا کمان نسبت به خطی که مقایسه می‌شود به اندازه یک واحد جهت جلوتر باشد مثلاً اگر خط Top to Bottom باشد، خط یا کمان ورودی Top Left to Bottom Right باشد.

با توجه به تنوع در شکل نوشتن، مدل سیستم باید طوری تعریف شود که مانند شکل ۲-۱۰ قابلیت انطباق الگوهای با تعداد نابرابر اجزا را داشته باشد.

این روش تطبیق الگو، این خاصیت را دارد که می‌تواند بهترین تطبیق بین ورودی با الگو را پیدا کند و نیازی به هم طول بودن آن‌ها نیست. پایگاه داده شامل ۱۲۵۰ کلمه از کتاب‌های اول دبستان نوشته شده توسط ۲۰ نفر است و نتیجه نهایی ۷۸ درصد بدون فرهنگ لغت و ۹۶ درصد با فرهنگ لغت است.[۲۱]

۲-۴- شناسایی زیر-کلمات با در نظر گرفتن هر زیر-کلمه به عنوان یک الگو

در این رویکرد هر زیر-کلمه به عنوان یک الگوی غیر قابل تفکیک در نظر گرفته می‌شود. مزیت این روش در این است که پیچیدگی‌های قطعه‌بندی را دور می‌زند اما در مقابل قادر نیست زیر-کلمه‌ای که در پایگاه داده موجود نباشد را شناسایی کند، ضمن این که چون تعداد الگوها زیاد است کار طبقه‌بندی نسبت به شناسایی حروف معدود خیلی سخت‌تر است.



شکل ۲-۱۰: (الف) الگو، (ب) مقایسه ورودی با الگو با ماشین کشسان فازی[۲۱]

رضوی و کبیر برای بازشناسی زیر-کلمات فارسی، ابتدا کل پایگاه داده را با توجه به علامت‌های زیر-کلمات گروه‌بندی کرده‌اند. با این فرض که ترتیب قرار دادن علامت‌ها از نظر مکان رعایت شود و نویسنده‌ها محدودیت‌های مربوط به شکل علامت‌ها را نیز رعایت کرده باشند، زیر-کلمات با توجه به ترتیب انواع علامت‌ها دسته‌بندی می‌شوند. به عنوان مثال زیر-کلماتی که دارای به ترتیب یک نقطه بالا و دو نقطه پایین باشند در یک گروه قرار می‌گیرند. برای شناسایی نقطه از اندازه آن استفاده می‌شود و برای تشخیص علامت‌های دیگر مشابه با روش تشخیص بدنه زیر-کلمات استفاده می‌شود با این تفاوت که علامت به ۱۰ نقطه نرمالیزه می‌شود.[۶]

برای تشخیص بدنه، زیر-کلمه به ۵۰ نقطه نرمالیزه می‌شود و زاویه بردارهای متصل کننده نقاط به دست می‌آید، مجموع وزن‌داری از مجموع قدرمطلق اختلاف زوایا و مجموع قدرمطلق فاصله اقلیدسی بین نقاط به عنوان معیار فاصله در نظر گرفته شده است. این روش بر روی پایگاه داده زیر-کلمات برخط دانشگاه تربیت مدرس آزمایش شده است و نرخ بازشناسی برای گزینه اول برابر با ۷۴/۹۵ درصد است.[۶]

در تکمیل کار قبلی، سیستمی برای بازشناسی کلمات فارسی پیشنهاد شده است که بعد از به دست آوردن کاندیدهای هر زیر-کلمه، امتیاز هر زیر-کلمه برابر با کمترین فاصله تقسیم بر فاصله آن زیر-کلمه در نظر گرفته شده است و امتیاز کلمه از حاصل‌ضربِ امتیاز زیر-کلمات کاندید به دست می‌آید، با توجه به فرهنگ کلمات فارسی موجود، کلمات کاندید غیر موجود، حذف می‌شوند. این سیستم روی یک متن که توسط یک نفر نوشته شده آزمایش شده که دقت بازشناسی آن در سطح کلمه ۱۳۱ از ۱۵۰ و در سطح زیر-کلمه ۳۲۹ از ۳۵۳ است.[۲۲]

فرجی و همکاران برای بازشناسی زیر-کلمات پایگاه داده تربیت مدرس، ابتدا آن‌ها را بر اساس علامت گروه‌بندی کرده‌اند، در این گروه‌بندی با توجه به این که نویسنده‌ها ترتیب علامت‌ها را رعایت

نمی‌کنند، فقط وجود علامت‌های مختلف در نظر گرفته شده است به عنوان مثال تمام زیر-کلمات دارای یک سرکش و دو نقطه بالا در یک گروه قرار می‌گیرند و ترتیب وقوع علامت‌ها اهمیت ندارد.[۲۳]

مراحل پیش‌پردازش بدنه اصلی شامل حذف قلاب، هموارسازی، نرمالیزاسیون و همسان‌سازی کادر نوشته است. از ویژگی جهت بردار ابتدا به انتهای نوشته که یا رو به پایین و یا رو به بالا است برای زیرطبقه‌بندی قبل از طبقه‌بندی نهایی استفاده می‌شود، سپس زاویه بردارها به ۸ جهت اصلی رقمی شده و به عنوان ویژگی در طبقه‌بند مدل مخفی مارکوف استفاده می‌شود. نتیجه این روش برای بهترین پارامترها، ۸۲/۴۳ درصد است.[۲۳]

کار قبلی با استفاده از ۳ مرحله طبقه‌بند شبکه عصبی RBF و افزودن ویژگی جهت رقمی شده بردارهای متصل کننده نقاط تیز و بیشینه محلی متوالی ۸۱/۳۱ درصد گزارش شده است.[۷]

فصل ۳

تئوری‌های مرتبط با تشخیص دستنوشته برخط فارسی

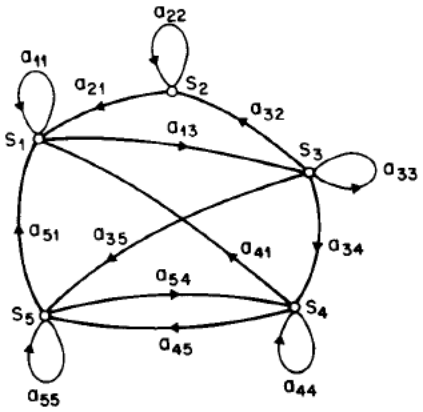
فصل ۳- تئوری‌های مرتبط با تشخیص دستنوشته برخط فارسی

۳-۱- مدل مخفی مارکوف[1] [۲۴]

مدل‌سازی سیگنال‌ها یکی از مهم‌ترین روش‌ها در پردازش سیگنال‌های پیچیده که از یک فرآیند آماری نتیجه می‌شوند است. یکی از روش‌های مدل‌سازی سیگنال‌های آماری، مدل مخفی مارکوف است که اولین بار توسط باوم[2] در دهه ۱۹۷۰ معرفی شد. این روش ابتدا برای تشخیص صحبت استفاده شد اما بعداً در دیگر زمینه‌هایی که با سیگنال‌های زمانی یا ترتیبی، از جمله تشخیص دستنوشته برخط، سروکار داشت هم مورد استفاده قرار گرفت و به دلیل نتایج خوب، به یکی از روش‌های پرکاربرد در مدل‌سازی این سیگنال‌ها تبدیل شد.

۳-۱-۱- فرایند مارکوف گسسته

سیستمی را در نظر بگیرید که در هر لحظه، در یکی از c حالت[3] مجزای $S_1, S_2, \dots, S_c$ قرار دارد و بعد از هر فاصله گسسته زمانی بر اساس مجموعه احتمالات مربوط به حالت فعلی، به یک حالت جدید (که می‌تواند حالت فعلی هم باشد) منتقل می‌شود. در شکل ۳-۱ یک زنجیره مارکوف با ۵ حالت (گره‌ها) و تعدادی از انتقال‌های حالت ممکن نمایش داده شده است.



شکل ۳-۱: یک زنجیره مارکوف با ۵ حالت و تعدادی انتقال حالت[۲۴]

لحظات زمانی را با $t = 1,2, \dots$ و حالت در زمان t را با q_t نشان می‌دهیم. توصیف احتمالاتی کامل از این سیستم در حالت کلی با داشتن مشخصات احتمالاتی در زمان فعلی t و حالت‌های قبلی امکان

[1] Hidden Markov Model (HMM)
[2] Baum
[3] State

پذیر است. برای مورد خاص یک زنجیره مارکوف[1] مرتبه اول گسسته، این توصیف احتمالاتی فقط به حالت فعلی و یک حالت قبل محدود می‌شود، که به صورت زیر قابل نمایش است:

$$P(q_t = S_j | q_{t-1} = S_i, q_{t-2} = S_k, \dots) = P(q_t = S_j | q_{t-1} = S_i) \quad (۱-۳)$$

با در نظر گرفتن این که سمت راست (۳-۱) مستقل از زمان است، می‌توانیم مجموعه احتمال‌های انتقال حالت a_{ij} به صورت

$$a_{ij} = P(q_t = S_j | q_{t-1} = S_i) \quad 1 \le i, j \le c \quad (۲-۳)$$

و با شرایط

$$a_{ij} \ge 0$$
$$\sum_j a_{ij} = 1 \quad (۳-۳)$$

را تعریف کنیم.

این فرایند را می‌توان یک مدل مارکوف قابل مشاهده نامید چون خروجی فرایند، مجموعه‌ای از حالت‌ها در هر لحظه از زمان است که هر حالت مربوط به فرایند فیزیکی قابل مشاهده است. برای روشن شدن موضوع، یک مدل مارکوف سه حالته ساده وضعیت هوا را در نظر بگیرید. فرض می‌کنیم که یک بار در روز (مثلاً ظهر) وضعیت هوا مطابق زیر مشاهده شود:

- وضعیت ۱: بارانی (یا برفی)
- وضعیت ۲: ابری
- وضعیت ۳: آفتابی.

فرض می‌کنیم که هوا در روز t توسط فقط یکی از حالت‌های بالا مشخص شود و ماتریس A احتمالات انتقال حالت[2] را به صورت زیر داشته باشیم:

$$A = \{a_{ij}\} = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix} \quad (۴-۳)$$

اگر بدانیم هوا در روز ۱ ($t = 1$) آفتابی (حالت ۳) است، می‌توانیم این سؤال را مطرح کنیم: (بر اساس مدل) احتمال این که در ۷ روز آینده وضعیت هوا به صورت "آفتابی، آفتابی، بارانی، بارانی، آفتابی، ابری، آفتابی" باشد، چقدر است؟ با بیان رسمی‌تر رشته مشاهدات O را به صورت $O = \{S_3, S_3, S_3, S_1, S_1, S_3, S_2, S_3\}$ مربوط به $t = 1,2, \dots, 8$ تعریف می‌کنیم و به دنبال احتمال رخداد O با داشتن مدل بالا هستیم. این احتمال به صورت زیرمحاسبه می‌شود:

[1] Markov Chain
[2] Transition Matrix

$$P(O|Model) = P(S_3, S_3, S_3, S_1, S_1, S_3, S_2, S_3|Model)$$
$$= P(S_3).P(S_3|S_3).P(S_3|S_3).P(S_1|S_3).P(S_1|S_1).P(S_3|S_1).P(S_2|S_3).P($$
$$= \pi_3.a_{33}.a_{33}.a_{31}.a_{11}.a_{13}.a_{32}.a_{23}$$
$$= 1.(0.8)(0.8)(0.1)(0.4)(0.3)(0.1)(0.2) = 1.536 \times 10^{-4} \quad (۵-۳)$$

که π_i احتمال قرار گرفتن در حالت اولیه iام به صورت زیر تعریف می‌شود:

$$\pi_i = P(q_1 = S_i), \quad 1 \leq i \leq c \quad (۶-۳)$$

۳-۱-۲- تعمیم به مدل مخفی مارکوف

تا این جا مدل‌های مارکوفی را در نظر گرفتیم که هر حالت مربوط به یک رویداد قابل مشاهده (فیزیکی) بود. این مدل قابلیت اعمال بر روی بسیاری از مسائل مورد علاقه را ندارد. در این بخش، مفهوم مدل مارکوف را به شکلی که مشاهده یک تابع احتمالی از حالت است تعمیم می‌دهیم. مدل به دست آمده که مدل مخفی مارکوف نامیده می‌شود یک فرایند تصادفی دو لایه است که دارای یک فرایند تصادفی زیرین غیر قابل مشاهده (مخفی) است که از طریق مجموعه‌ای از فرایندهای تصادفی دیگر که همان رشته مشاهدات[1] هستند، قابل دیدن است. برای درک موضوع مثال زیر را در نظر بگیرید.

فرض کنید که در اتاقی قرار دارید و از وضعیت هوای بیرون اطلاع ندارید ولی می‌دانید که فردی که غذای روزانه شما را می‌آورد در روزهای بارانی به احتمال ۰/۸، در روزهای ابری به احتمال ۰/۳ و در روزهای آفتابی با احتمال ۰/۱ با خودش چتر می‌آورد. در این صورت، آوردن چتر یا نیاوردن آن را می‌توان یک مشاهده تلقی نموده و وضعیت هوا را حالت مخفی نامید و ماتریس توزیع احتمال مشاهدات[2] را به صورت زیر تشکیل داد:

$$B = \{b_{jk}\} = \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \\ 0.1 & 0.9 \end{bmatrix} \quad (۷-۳)$$

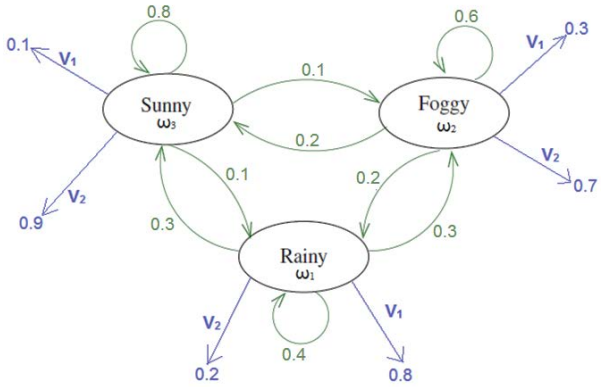
به طوری که b_{jk} احتمال این است که در حالت j ام مشاهده k ام را داشته باشیم. مثلاً b_{jk} احتمال این است که در یک روز آفتابی (حالت مخفی ۳)، چتر آورده نشده است (مشاهده ۲).

بدیهی است که مقادیر ماتریس توزیع احتمال مشاهدات باید در شرط زیر صدق کنند:

$$b_{jk} \geq 0$$
$$\sum_k b_{jk} = 1 \quad (۸-۳)$$

[1] Observations
[2] Emission Matrix

این مدل را می‌توان با شکل ۳-۲ نشان داد، که ω نماد حالت و V نماد مشاهده است که از کلمه Visible به معنای قابل مشاهده گرفته شده است.



شکل ۳-۲: مدل مخفی مارکوف برای مثال وضعیت هوا

۳-۱-۳- مسائل اساسی مدل مخفی مارکوف[۲۵]

در مدل مخفی مارکوف با سه مسئله اساسی روبرو هستیم:

۱. ارزیابی[1]: احتمال رخداد یک رشته مشخص از مشاهدات با داشتن مدل

به عنوان مثال در مثال وضعیت هوا، احتمال رخداد $V^5 = \{V_1, V_1, V_2, V_2, V_1\}$ با داشتن ماتریس‌های احتمالات اولیه π، انتقال حالت A و احتمال مشاهدات B که از این پس این مجموعه را با λ نشان می‌دهیم.

احتمال رخداد رشته مشاهده $V^T = \{V(1), V(2), \dots, V(T)\}$ را می‌توان به صورت زیر محاسبه کرد:

$$P(V^T) = \sum_{r=1}^{r_{max}} P(V^T|\omega_r^T) P(\omega_r^T) \qquad (۹-۳)$$

در (۳-۹) از قاعده احتمال متوسط استفاده شده است. اگر c حالت مخفی $\omega_1, \omega_2, \dots, \omega_c$ داشته باشیم و طول رشته مشاهدات هم T باشد برای $\omega_r^T = \{\omega(1), \omega(2), \dots, \omega(T)\}$ تعداد c^T ترکیب مختلف خواهیم داشت یعنی $r_{max} = c^T$، همچنین قسمت اول مجموع بالا را می‌توان به صورت زیر نوشت:

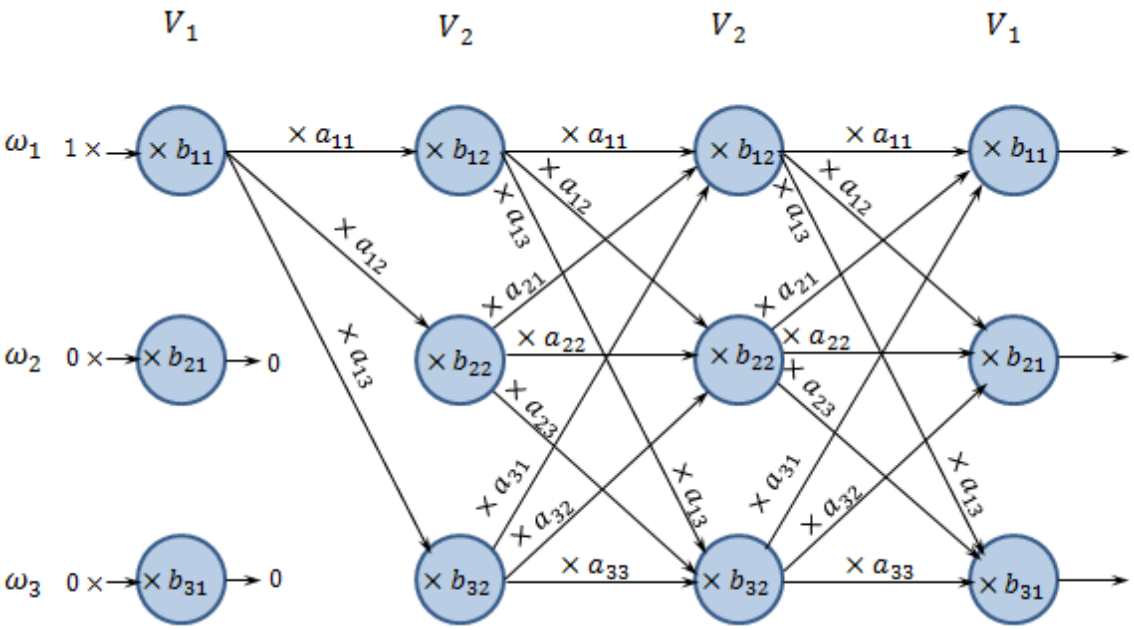
$$P(V^T|\omega_r^T) = \prod_{t=1}^{T} P(V(t)|\omega(t)) \qquad (۱۰-۳)$$

این روش به علت پیچیدگی محاسبه این احتمال که از درجه $\Theta(T.c^T)$ است، غیرعملی است. با استفاده از الگوریتم پیشرو[1] درجه پیچیدگی محاسبه به $\Theta(T.c^2)$ تقلیل می‌یابد. در محاسبه

[1] Evaluation

احتمال (۳-۱۰)، بعضی از حاصل‌ضرب‌ها در ترکیب‌های مختلف ω_r^T تکرار می‌شوند که با استفاده از الگوریتم پیشرو این حاصل‌ضرب‌ها فقط یک بار محاسبه می‌شوند و از این نظر در محاسبات صرفه‌جویی می‌شود. این الگوریتم را با یک مثال توضیح می‌دهیم.

در مثال وضعیت هوا می‌خواهیم احتمال رخداد رشته مشاهدات $V^4 = \{V_1, V_2, V_2, V_1\}$ را محاسبه کنیم. با فرض $\pi = [1,0,0]$ و مقادیر A و B، شبکه شکل ۳-۳ را تشکیل می‌دهیم.



شکل ۳-۳: الگوریتم پیشرو

اگر احتمال رسیدن به یک نقطه (حالت j در زمان t) از تمام مسیرهای ممکن را $\alpha_j(t)$ بنامیم، بنا بر الگوریتم پیشرو به صورت زیر محاسبه می‌شود:

$$\alpha_j(t) = \begin{cases} 0 & t = 0; j \neq \text{حالت آغازی} \\ 1 & t = 0; j = \text{حالت آغازی} \\ \sum_i \alpha_i(t-1) a_{ij} b_{jk_t} & t \neq 0 \end{cases} \qquad (۱۱-۳)$$

۲. رمزگشایی[2]: به پیدا کردن محتمل‌ترین رشته حالت‌ها که منجر به توالی مشاهده V^T می‌شود گفته می‌شود.

حل این مسئله به صورتی که احتمال تمام حالت‌ها به ازای رخداد V^T را ابتدا محاسبه و سپس مقایسه کنیم، مشابه روش کلاسیک در مسئله ارزیابی است، اما با استفاده از الگوریتم رمزگشایی

[1] HMM Forward
[2] Decoding

پیچیدگی محاسباتی به اندازه الگوریتم پیشرو است. در این روش از زمان $t = 1$ تا $t = T$، حالتی که بیشترین مقدار $\alpha_j(t)$ را می‌دهد به عنوان حالت بهینه در زمان t در نظر می‌گیریم. اشکالی که ممکن است در این روش پیش بیاید این است اگر a_{ij} صفر باشد اما در زمان‌های متوالی t و $t+1$، حالت‌های ω_i و ω_j در مسیر بهینه وجود داشته باشند، احتمال رخداد این رشته صفر می‌شود که باید در یک مرحله اضافی بررسی شود که چنین موردی وجود نداشته باشد.

الگوریتم رمزگشایی در شکل ۳-۴ نشان داده شده است.

```
1  begin initialize  Path = {}, t = 0
2          for  t ← t + 1
4              k = 0, α_0 = 0
5              for  k ← k + 1
7                  α_k(t) ← b_jk v(t) Σ_{i=1}^{c} α_i(t − 1) a_ij
8              until k = c
10             j' ← arg max_j α_j(t)
11             AppendTo Path ω_j'
12         until t = T
13   return Path
14 end
```

شکل ۳-۴: الگوریتم رمزگشایی

۳. یادگیری[1]: هدف مسئله یادگیری، تخمین پارامترهای مدل مخفی مارکوف (λ) با استفاده از نمونه‌های آموزشی است.

در حالت کلی برای حل چنین مسئله‌ای باید از شکل عمومی الگوریتم بیشینه‌سازی امید ریاضی[2] استفاده کرد. ولی برای مسئله مدل مخفی مارکوف مرتبه اول می‌توان از یک حالت خاص این الگوریتم به نام الگوریتم پیشرو-پسرو[3] یا باوم-ولچ استفاده کرد.

مقدار $\alpha_i(t)$ همان‌طور که قبلاً گفته شد، احتمال بودن مدل در حالت ω_i در زمان t وقتی که رشته مشاهده تا زمان t تولید شده باشد تعریف می‌کنیم، به طور مشابه مقدار $\beta_i(t)$ را هم احتمال بودن مدل در زمان t در حالت ω_i و تولید ادامه رشته مشاهده داده شده از $t+1$ تا T تعریف می‌کنیم. به صورت مشابه الگوریتم پیشرو، می‌توان مقدار $\beta_i(t)$ در الگوریتم پسرو را به صورت زیر نوشت:

[1] Learning
[2] Expectation Maximization (EM)
[3] Forward-Backward

$$\beta_i(t) = \begin{cases} 0 & t = T; i \neq \text{حالت پایانی} \\ 1 & t = T; i = \text{حالت پایانی} \\ \sum_j a_{ij} b_{jk_{t+1}} \beta_j(t+1) & t \neq T \end{cases} \quad (۱۲-۳)$$

با توجه به این که در مسئله یادگیری، پارامترهای مدل مشخص نیستند، ابتدا به صورت تصادفی و با توجه به شروط (۳-۳)، (۳-۶) و (۳-۸) در نظر گرفته می‌شوند، سپس با کمک نمونه‌های آموزشی و با تکرار الگوریتم، این مقادیر به روز رسانی می‌شوند تا تخمین مناسبی از پارامترها به دست آید.

مقدار $\gamma_{ij}(t)$ را به عنوان احتمال انتقال از $\omega_i(t-1)$ به $\omega_j(t)$ در صورت رخداد کامل نمونه آموزشی رشته مشاهدات V^T و با هر مسیر از رشته حالت‌ها، تعریف می‌کنیم. می‌توان نشان داد:[۲۷]

$$\gamma_{ij}(t) = \frac{\alpha_i(t-1) a_{ij} b_{jk_t} \beta_j(t)}{P(V^T|\lambda)} \quad (۱۳-۳)$$

که $P(V^T|\lambda)$ احتمال تولید رشته مشاهده V^T از تمام مسیرهاست. امید ریاضی انتقال از $\omega_i(t-1)$ به $\omega_j(t)$ به ازای تمام زمان‌ها، برابر است با $\sum_{t=1}^{T} \gamma_{ij}(t)$ و امید ریاضی انتقال از $\omega_i(t-1)$ به تمام حالت‌ها در زمان $t+1$ برابر است با $\sum_{t=1}^{T} \sum_k \gamma_{ik}(t)$، بنابراین مقدار تخمین بعدی برای a_{ij} که با $\hat{a}_{ij}$ نشان می‌دهیم را می‌توانیم با نسبت امید ریاضی انتقال از $\omega_i(t-1)$ به $\omega_j(t)$ به امید ریاضی انتقال از $\omega_i(t-1)$ به تمام حالت‌ها در زمان $t+1$ به دست بیاوریم:

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T} \gamma_{ij}(t)}{\sum_{t=1}^{T} \sum_k \gamma_{ik}(t)} \quad (۱۴-۳)$$

به طور مشابه برای به‌روزرسانی b_{jk} هم با استدلال مشابه خواهیم داشت:

$$\hat{b}_{jk} = \frac{\sum_{t=1}^{T} \gamma_{jk}(t) | V(t) = V_k}{\sum_{t=1}^{T} \gamma_{ik}(t)} \quad (۱۵-۳)$$

روال تکرار الگوریتم برای داده‌های آموزش تا جایی ادامه پیدا می‌کند که قدرمطلق تفاضل نتیجه به‌روزرسانی دو مرحله متوالی از مقدار آستانه مشخصی کمتر شود.

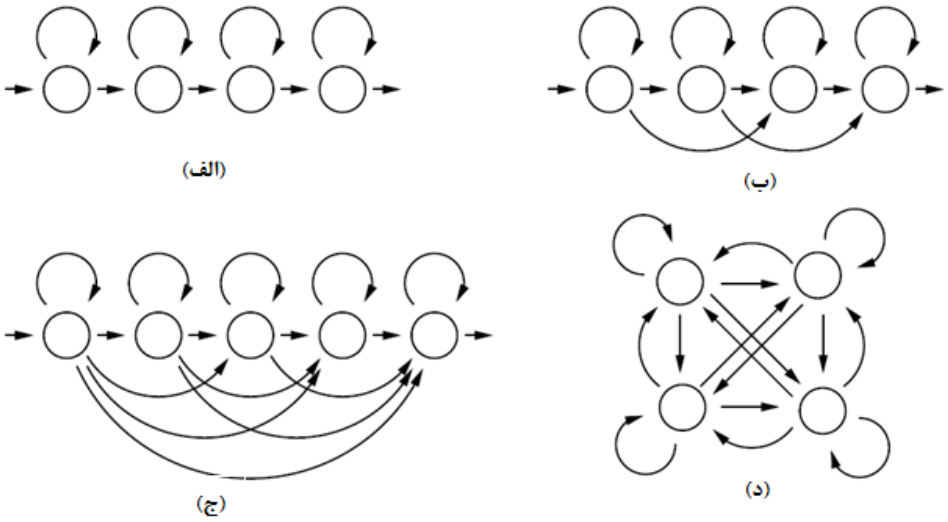
۳-۱-۴- توپولوژی‌های مدل مخفی مارکوف[۲۴]

تا اینجا فقط توپولوژی ارگادیک[1] مدل مخفی مارکوف را در نظر گرفتیم که در آن رسیدن به هر حالت مدل فقط با یک یا تعداد محدودی انتقال ممکن است. در این توپولوژی تمام ضرایب ماتریس انتقال حالت می‌توانند غیر صفر باشند.

اگر فقط انتقال حالت از حالت پایین‌تر (با شماره کمتر) به خود آن حالت یا حالت‌های بالاتر قابل قبول باشد، در این صورت از توپولوژی چپ به راست[2] استفاده می‌شود. در صورتی که تعداد حالت‌های

[1] Ergodic
[2] Left-Right topology

بالاتر قابل قبول برابر با ۲ باشد، توپولوژی Bakis و اگر ۱ باشد، توپولوژی خطی[1] نامیده می‌شود. بلوک دیاگرام توپولوژی‌های ذکر شده در شکل ۳-۵ نشان داده شده است.

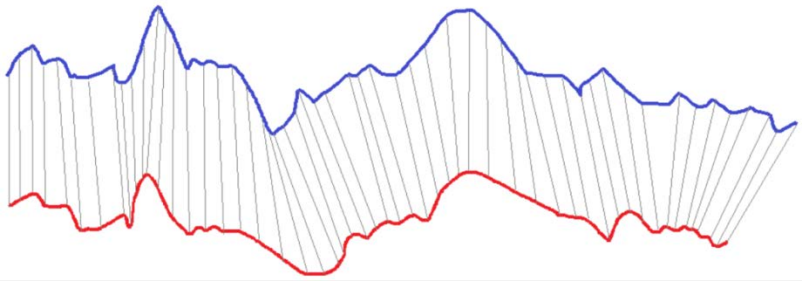


شکل ۳-۵: توپولوژی‌های مدل مخفی مارکوف: (الف) خطی، (ب) Bakis، (ج) چپ به راست، (د) ارگادیک[۲۶]

۳-۲- پیچش زمانی پویا[2][۲۷]

پیچش زمانی پویا (DTW) یک الگوریتم پویا است که همترازی[3] بین دو سیگنال زمانی که به صورت غیر یکنواخت کشیده یا فشرده شده‌اند را بهینه می‌کند. این روش معیاری عددی برای میزان اختلاف بین دو سیگنال ارائه می‌کند.

منظور از نقاط همترازی، نقاطی از دو سیگنال است که دو سیگنال در آن نقاط به هم شبیه هستند. شکل ۳-۶ نقاط همترازی بین یک سیگنال با سیگنال مشابهی که دارای کشیدگی و فشردگی غیریکنواخت است را نشان می‌دهد.



شکل ۳-۶: نقاط همترازی بین دو سیگنال

[1] Linear topology
[2] Dynamic Time Warping (DTW)
[3] Alignment

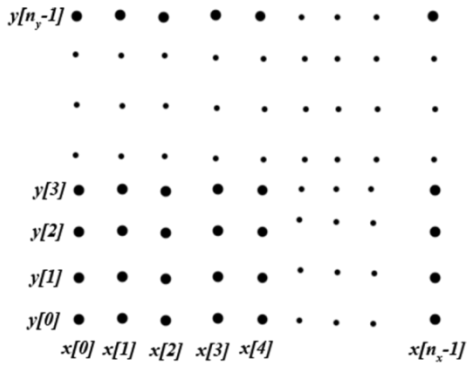
علاوه بر پیدا کردن نقاط همترازی بین دو سیگنال، الگوریتم DTW یک عدد به عنوان میزان اختلاف بین دو سیگنال مورد مطالعه می‌دهد که در کارهای تشخیص الگو، برای مقایسه الگوهای مرجع با الگوهای آزمایشی استفاده می‌شود، مزیت این الگوریتم به مقایسه نقطه به نقطه در این است که علاوه بر پیدا کردن مسیر همترازی، نیازی به هم طول بودن دو سیگنال نیست و مهم‌ترین محدودیت‌های این روش هم این است که وقتی الگوها پیچیده باشند یا تعداد کلاس‌های مرجع زیاد باشد، دقت آن کاهش زیادی می‌یابد.

فرض کنید دو بردار ویژگی مربوط به دو نمونه، یکی نمونه مرجع و دیگری نمونه ورودی به صورت زیر داریم:

$$x[n] = \{x[0], x[1], \dots, x[n_x - 1]\}$$
$$y[n] = \{y[0], y[1], \dots, y[n_y - 1]\} \quad (۱۶-۳)$$

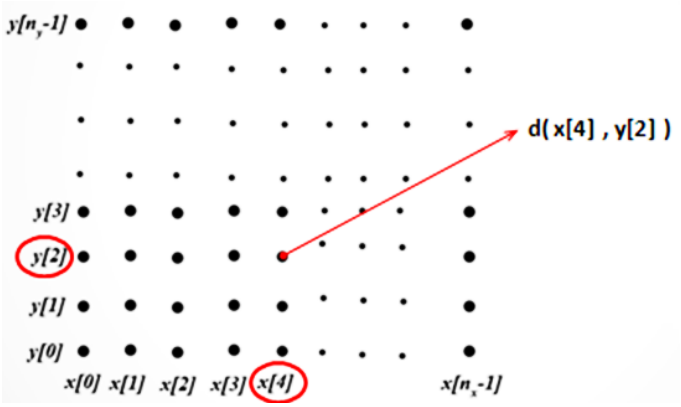
هر کدام از عناصر داخل آکولادها، خود یک بردار هستند که ابعاد آن‌ها مساوی و برابر با تعداد ویژگی‌هاست. اندیس داخل کروشه‌ها هم نشان دهنده زمان است. همان طور که مشخص است طول x و y به ترتیب برابر با n_x و n_y است که می‌توانند برابر نباشند. در نتیجه x و y دو ماتریس دو بعدی با تعداد ستون‌های مساوی هستند.

با استفاده از ماتریس‌های گفته شده، شبکه شکل ۳-۷ را تشکیل می‌دهیم.



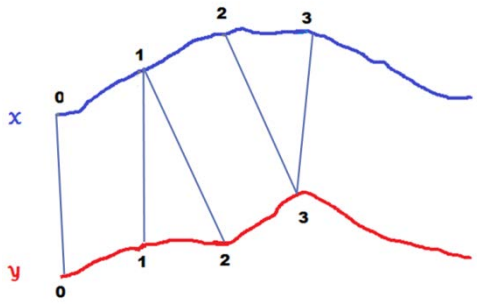
شکل ۳-۷: شبکه تشکیل شده از دو سیگنال

در اولین مرحله شروع به پر کردن گره‌های شبکه می‌کنیم. در هر کدام از گره‌ها مطابق شکل ۳-۸، فاصله بین دو بردار ویژگی مربوط به همان گره را وارد می‌کنیم.



شکل ۳-۸: مقداردهی به گره‌های شبکه

فاصله بین دو بردار ویژگی را می‌توان با تعاریف مختلف فاصله بسته به کاربرد و حجم محاسباتی از نوع فاصله بلوک شهری[1]، فاصله اقلیدسی[2] و ... در نظر گرفت. مقادیر گره‌های شبکه که با استفاده از محاسبه فاصله بین بردارهای ویژگی به دست آمد را ارزش محلی گره می‌نامیم. مقادیر ارزش‌های محلی را در یک ماتریس ذخیره می‌کنیم، نام این ماتریس را d می‌گذاریم و هر عضو آن با اندیس به صورت d[i,j] مشخص می‌شود که i زمان مربوط به سیگنال x و j زمان مربوط به سیگنال y است.
حال می‌خواهیم جزئی تر به مفهوم همترازی نگاه کنیم. شکل ۳-۹ را در نظر بگیرید.

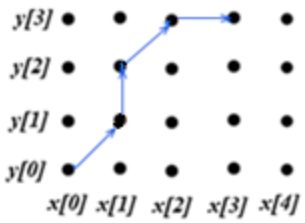


شکل ۳-۹: نقاط همترازی بین دو سیگنال

در این شکل، دو سیگنال یک بعدی x و y که در زمان‌های ۰ تا ۳ نمونه برداری شده‌اند می‌بینید. با توجه به مقدار سیگنال‌ها یک همترازی بین آن‌ها پیدا کرده‌ایم که با خطوط واصل بین نقاط دو سیگنال نشان داده شده است. نقاط اولیه $x[0]$ و $y[0]$ را همتراز فرض می‌کنیم. حال اگر به نمونه بعدی دو سیگنال نگاه کنیم می‌بینیم که هر دو تقریباً به اندازه یکسانی زیاد شده‌اند پس نقاط بعدی را هم با هم همتراز می‌گیریم یعنی نقاط $x[1]$ با $y[1]$.

[1] City-Block
[2] Euclidean

اگر باز هم در زمان جلو برویم مشاهده می‌کنیم که سیگنال x باز هم مقدارش زیاد شده در حالی که سیگنال y در جای قبلی باقی مانده است پس نقطه بعدی در سیگنال y یعنی y[2] با نقطه فعلی در سیگنال x یعنی x[1] همتراز است. تا اینجا سیگنال x در زمان ۱ و سیگنال y در زمان ۲ قرار دارند. به نقاط بعدی می‌رویم؛ هم x و هم y در زمان بعدی خودشان افزایش یافته‌اند پس x[2] با y[3] همتراز است. به همین ترتیب نقطه بعدی را هم بررسی می‌کنیم؛ این بار در حالی که سیگنال y کاهش پیدا کرده است، سیگنال x ثابت مانده است پس نقاط x[3] و y[3] همترازند. این روند را در شبکه ایجاد شده به صورت شکل ۳-۱۰ می‌توان نشان داد.



شکل ۳-۱۰: مسیر همترازی

در شبکه، از نقطه اولیه که مربوط به بردارهای x[0] و y[0] است و همتراز فرض می‌شوند، شروع شده است و با بردارهای آبی رنگ مسیر نقاط همترازی که مسیر همترازی نامیده می‌شود مشخص شده است. مسیر همترازی باید طوری ادامه پیدا کند که در نقطه پایانی مربوط به بردارهای $x[n_x - 1]$ و $y[n_y - 1]$ ختم شود.

حال یک ماتریس جدید تعریف می‌کنیم که مشابه ماتریس d روی سطر و ستون‌های شبکه تعریف می‌شود که آن را مقدار گره می‌نامیم و با D نشان می‌دهیم. از مقادیر این ماتریس برای تعیین مسیر همترازی و تعیین مقدار کمّی فاصله دو سیگنال استفاده می‌شود. مقادیر D به صورت زیر تعریف می‌شوند.

$$D[i,j] = \min_{(a,b)\in P}(D[i-a, j-b] + d[i,j] * w(a,b)) \qquad (۳-۱۶)$$

که w تابع وزن و P مجموعه ضرایب مربوط به قیدهای محلی هستند و منظور از عبارت min در فرمول بالا این است که مقدار گره در نقطه D[i, j] از مسیری به دست می‌آید که کمترین مقدار را در آن گره ایجاد کند. این الگوریتم از این نظر که مسیری را برای یافتن مینیمم مقدار پیدا می‌کند شبیه به مدل مخفی مارکوف است. برای پیدا کردن این مسیر بهینه، باید کلیه مسیرهای مجاز بر اساس قید محلی آزمایش شوند، مسیری که منجر به کمترین مقدار گره شود را در نظر گرفته و مقدار به دست آمده را به D[i, j] نسبت می‌دهیم.

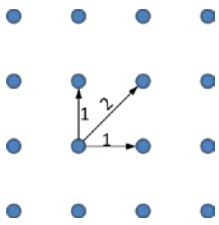
۳-۲-۱- قیدهای محلی

تعدادی از قیدهای محلی که کاربرد بیشتری دارند در این قسمت توضیح داده می‌شوند. این که از کدام قید محلی استفاده شود بستگی به بار محاسباتی و کاربرد دارد.

اولین قید محلی به صورت زیر تعریف می‌شود:

$$D[0,0] = 2d[0,0]$$
$$D[i,j] = min\begin{cases} D[i,j-1] + d[i,j] \\ D[i-1,j-1] + 2d[i,j] \\ D[i-1,j] + d[i,j] \end{cases} \quad (۱۷-۳)$$

این قید را می‌توان به صورت شکل ۳-۱۱ نشان داد:



شکل ۳-۱۱: یک قید محلی برای DTW

در این قید محلی مقدار گره اول را دو برابر ارزش محلی آن در نظر می‌گیریم. سپس مقدار گره‌های دیگر را به ترتیب حساب می‌کنیم. گره‌های مجاور گره ابتدایی D[0,0]، گره‌های D[0,1] و D[1,0] هستند که بلافاصله بعد از مشخص شدن مقدار D[0,0] قابل محاسبه هستند. چون برای رفتن به هر کدام از این گره‌ها فقط یک راه وجود دارد، بنابراین مقادیر آن‌ها هم از اول معلوم است:

$$\mathrm{D[0,1]} = \mathrm{D[0,0]} + \mathrm{d[0,1]} = 2 * \mathrm{d[0,0]} + \mathrm{d[0,1]}$$
$$\mathrm{D[1,0]} = \mathrm{D[0,0]} + \mathrm{d[1,0]} = 2 * \mathrm{d[0,0]} + \mathrm{d[1,0]} \quad (۱۷-۳)$$

اما برای محاسبه بقیه گره‌ها نیاز به محاسبه بیش از یک مقدار است که باید مینیمم مقدار حاصل را در آن گره قرار دهیم. به عنوان مثال برای محاسبه D[1,1]، سه مسیر وجود دارد:

۱- مستقیم از D[0,0] که با توجه به این قاعده محلی به این صورت محاسبه می‌شود:

$$\mathrm{D[0,0]} + 2 * \mathrm{d[1,1]} \quad (۱۸-۳)$$

۲- از D[0,1] یعنی از گره [0,0] به گره [0,1] و بعد به گره [1,1]. در این صورت مقدار مسیر به صورت زیر است:

$$\mathrm{D[0,1]} + 1 * \mathrm{d[1,1]} = \mathrm{D[0,0]} + \mathrm{d[0,1]} + \mathrm{d[1,1]} \quad (۱۹-۳)$$

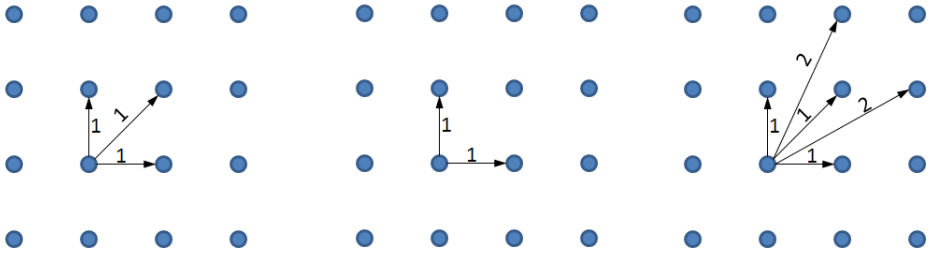
۳- از D[1,0] یعنی از گره [0,0] به گره [1,0] و بعد به گره [1,1]. در این صورت مقدار مسیر به صورت زیر است:

$$D[1,0] + 1 * d[1,1] = D[0,0] + d[1,0] + d[1,1] \quad (۲۰-۳)$$

حال کمترین این سه مقدار را به عنوان مقدار گره در $D[1,1]$ انتخاب می‌کنیم. مسیر رسیدن به این گره تا این لحظه قسمتی از مسیر همترازی است. با ادامه همین روال برای بقیه گره‌ها، علاوه بر مقادیر گره‌ها، مسیر همترازی هم به دست می‌آید. معیار کمی که بیانگر میزان فاصله (اختلاف) دو سیگنال است توسط آخرین گره مشخص می‌شود و برابر با مقدار آن گره است؛ یعنی:

$$Dist = D[n_x - 1, n_y - 1] \quad (۲۱-۳)$$

در شکل ۳-۱۲ تعداد دیگری از قیدهای محلی متداول آورده شده است، برای افزایش انعطاف پذیری می‌توان قید محلی را طوری تعریف کرد که بتواند از حداکثر یک نقطه (یک نمونه زمانی) در یکی از سیگنال‌ها پرش کند.



شکل ۳-۱۲: تعدادی از قیدهای محلی DTW

۳-۲-۲- مثال

سیگنال‌های زیر را در نظر بگیرید.

$$x[n] = \{0,1,2,3,2,1,0,1,2\}$$
$$y[n] = \{0,1,1,2,2,3,1,0\} \quad (۲۲-۳)$$

می‌خواهیم الگوریتم DTW را برای مقایسه این دو سیگنال به کار ببریم.

در این مثال از معیار فاصله بلوک شهری با تعریف زیر برای محاسبه مقادیر ارزش‌های محلی استفاده شده است.

$$d[i,j] = \sum_{\text{تمام ابعاد}} |x[i] - y[j]| \quad (۲۳-۳)$$

در (۳-۲۳) جمع روی همه ابعاد بردارهای ویژگی انجام شده است.

با محاسبه فاصله از فرمول بالا و قرار دادن ارزش‌های محلی هر گره در جای خود به ماتریس شکل ۳-۱۳ می‌رسیم.

0	0	1	2	3	2	1	0	1	2
1	1	0	1	2	1	0	1	0	1
3	3	2	1	0	1	2	3	2	1
2	2	1	0	1	0	1	2	1	0
2	2	1	0	1	0	1	2	1	0
1	1	0	1	2	1	0	1	0	1
1	1	0	1	2	1	0	1	0	1
0	0	1	2	3	2	1	0	1	2
d	0	1	2	3	2	1	0	1	2

شکل ۳-۱۳: محاسبه ارزش محلی گره‌ها

در این مرحله باید مقادیر گره‌ها را با توجه به قید محلی شکل ۳-۱۴ محاسبه کنیم.

شکل ۳-۱۴: قید محلی مورد استفاده در مثال

در این صورت به نتیجه شکل ۳-۱۵ خواهیم رسید.

0	10	5	4	5	3	2	1	2	4
1	10	4	2	2	1	1	2	2	3
3	9	4	1	0	1	3	5	6	6
2	6	2	0	1	1	2	4	5	5
2	4	1	0	1	1	2	4	5	5
1	1	0	1	3	4	4	5	5	6
1	1	0	1	3	4	4	5	5	6
0	0	1	3	6	8	9	9	10	12
D	0	1	2	3	2	1	0	1	2

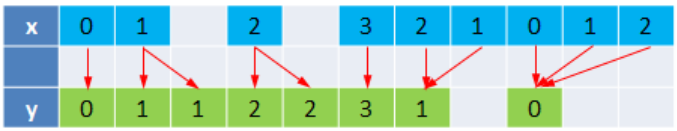
شکل ۳-۱۵: محاسبه مقادیر گره‌ها

علاوه بر روش گفته شده برای پیدا کردن مسیر همترازی می‌توان از روی ماتریس D به دست آمده هم مسیر همترازی را به صورت نشان داده شده در شکل ۳-۱۶ مشخص کرد.

0	10	5	4	5	3	2	1	2	4
1	10	4	2	2	1	1	2	2	3
3	9	4	1	0	1	3	5	6	6
2	6	2	0	1	1	2	4	5	5
2	4	1	0	1	1	2	4	5	5
1	1	0	1	3	4	4	5	5	6
1	1	0	1	3	4	4	5	5	6
0	0	1	3	6	8	9	9	10	12
D	0	1	2	3	2	1	0	1	2

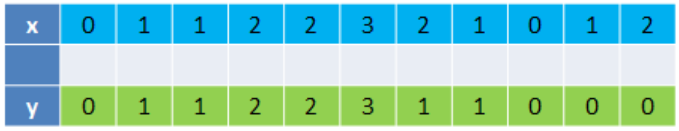
شکل ۳-۱۶: مسیر همترازی در ماتریس D

اگر دو سیگنال را به صورت عددی نشان دهیم، بر اساس توضیحات داده شده می‌توان نقاط همتراز دو سیگنال را به صورت نشان داده شده در شکل ۳-۱۷ به هم وصل کرد.



شکل ۳-۱۷: نقاط هم‌تراز در دو سیگنال مثال

در شکل ۳-۱۷ هر نقطه از سیگنال اول که با بیش از یک نقطه از سیگنال دیگر همتراز است به تعداد نقاط اضافی تکرار می‌شود. این کار برای سیگنال دوم هم تکرار می‌شود. نتیجه یکسان‌سازی طول سیگنال‌های مثال در شکل ۳-۱۸ نشان داده شده است.



شکل ۳-۱۸: سیگنال‌های گسترش یافته

فصل ۴

روش پیشنهادی

فصل ۴- روش پیشنهادی

۴-۱- پیشگفتار

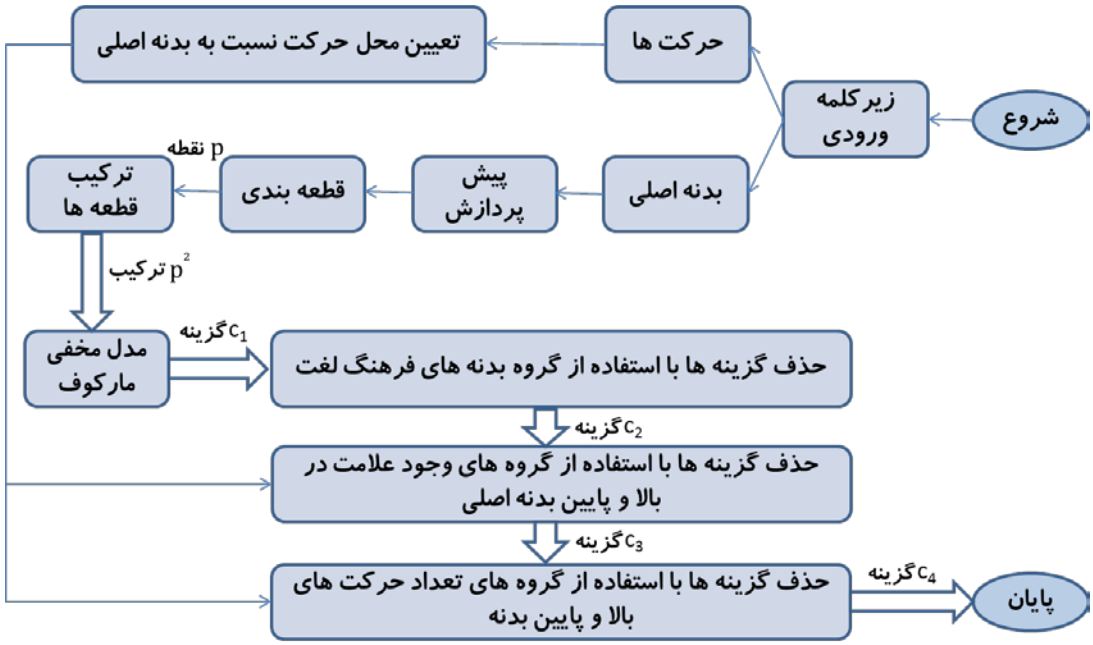
در این تحقیق برای تشخیص زیر-کلمات برخط فارسی مستقل از نویسنده، از روش قطعه‌بندی زیر-کلمه به حروف و سپس تشخیص حروف استفاده شده است. برای آزمایش صحت روش پیشنهادی قطعه‌بندی و همچنین برای آموزش حروف، باید ابتدا قطعه‌بندی زیر-کلمات به صورت دستی انجام شود و نواحی قطعه‌بندی به عنوان Ground Truth ذخیره شود. این قطعه‌بندی بر روی داده‌های پیش‌پردازش شده انجام شده است.

تخصیص علامت‌ها به حروف مربوطه در یک زیر-کلمه دارای خطای زیادی است. این علامت‌ها شامل کلاه "آ"، تک‌نقطه، دونقطه، سه نقطه، همزه، سرکش های "ک/گ" و دسته "ط/ظ" هستند. خوشبختانه به دلیل برخط بودن دستنوشته، جدا کردن علامت‌ها از بدنه اصلی امکان‌پذیر است. در مرحله آموزش و آزمون، علامت حروف در نظر گرفته نمی‌شوند، از این جهت حروف "بـ"، "پـ"، "تـ"، "ثـ"، "نـ" و "یـ" که دارای شکل بدنه یکسان هستند و فقط در علامت فرق دارند در یک گروه قرار می‌گیرند.

بعد از قطعه‌بندی زیر-کلمه به حروف سازنده آن، باید این حروف، شناسایی شوند. برای شناسایی طبقه‌بند مدل مخفی مارکوف پیشنهاد شده است. چون شکل حروف به موقعیت آن‌ها در زیر-کلمه بستگی دارد شناسایی حروف با توجه به موقعیت حرف در زیر-کلمه انجام می‌شود، از این نظر چهار گروه حروف جدا، حروف اول، حروف وسط و حروف آخر وجود دارند که تعداد گروه‌های آن‌ها هم متفاوت است.

خروجی مرحله شناسایی، یک رشته از بدنه‌های حروف است که باید علامت‌های زیر-کلمه به آن اعمال شود، به دلیل مشکلاتی که در تخصیص علامت‌ها به حروف و همچنین تشخیص علامت‌ها وجود دارد، نتایج را با استفاده از علامت‌های زیر-کلمه و مجموعه زیر-کلمات متنی، فیلتر می‌کنیم. از متن زیر-کلمات پایگاه داده به عنوان مجموعه زیر-کلمات متنی استفاده شده است.

بلوک دیاگرام کلی طرح را در شکل ۴-۱ می‌بینید که تا پایان این فصل تمام قسمت‌های آن توضیح داده می‌شود.



شکل ۴-۱: بلوک دیاگرام کلی طرح

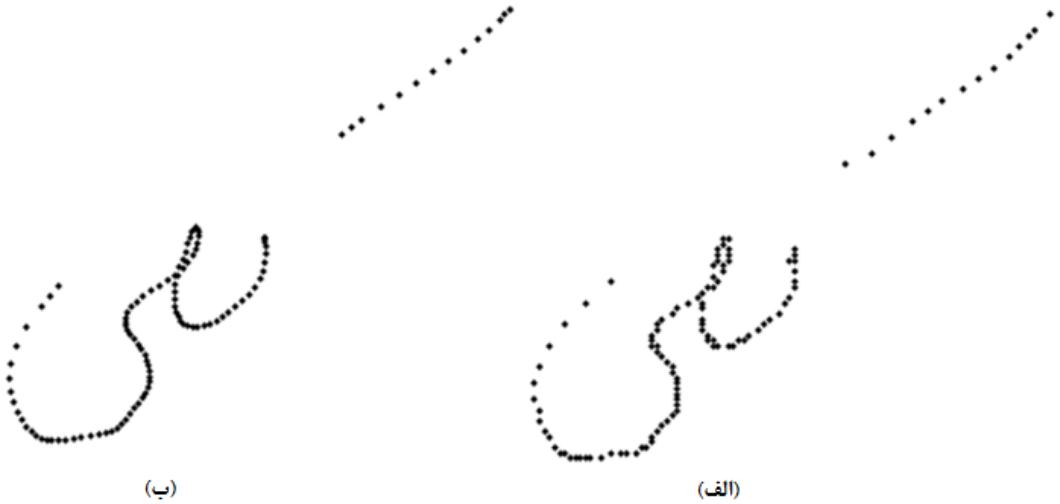
۴-۲- پیش‌پردازش

برای راحت‌تر شدن کار با داده‌ها، ابتدا محورهای مختصات را به گونه‌ای منتقل می‌کنیم تا کوچک‌ترین مقادیر مختصات به نقطه [0,0] منتقل شود، برداشتن قلم را هم با مختصات [-1,-1] در پایان هر حرکت قلم مشخص می‌کنیم. سپس پیش‌پردازش‌های هموارسازی و نرمالیزاسیون روی داده‌ها انجام شده است.

۴-۲-۱- هموارسازی

برای حذف نویز مربوط به حرکت قلم، لازم است رشته داده‌ها با یک فیلتر پایین‌گذر هموارسازی شود تا شکل یکنواخت‌تری پیدا کند. برای هموارسازی داده از یک فیلتر میانگین متحرک[1] استفاده شده است. این فیلتر، مختصات x (افقی) و y (عمودی) هر نقطه را با میانگین مختصات چند نقطه مجاور جایگزین می‌کند. برای این که شکل دستنوشته در اثر هموارسازی دچار تغییر نامناسب نشود، از ۲ نقطه قبل و ۲ نقطه بعد از هر نقطه برای میانگین‌گیری استفاده شده است. شکل ۴-۲ نتیجه هموارسازی را برای یک مثال نشان می‌دهد.

[1] Moving-average filter



شکل ۴-۲: (الف) دستنوشته موجود در پایگاه داده (زیر-کلمه "کلی")، (ب) شکل هموارسازی شده

با توجه به این که شکل علامت‌ها در سیستم تشخیص دستنوشته پیشنهادی نقشی ندارد، نیازی به انجام پیش‌پردازش روی علامت‌ها نیست.

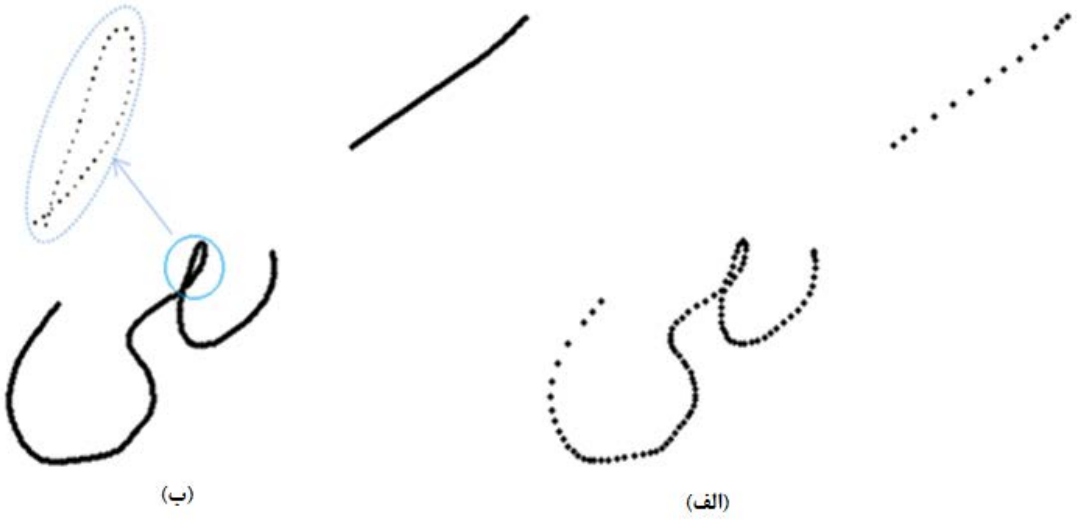
۴-۲-۲- نمونه‌برداری مجدد

سرعت حرکت قلم در یک کلمه تغییر می‌کند، همچنین نویسنده‌های مختلف سرعت نوشتن متفاوت دارند. چون سیستم پیشنهادی برای تشخیص دستنوشته مستقل از نویسنده طراحی شده است، اطلاعات سرعت قلم نه تنها کمکی به حل مسئله نمی‌کند بلکه آن را پیچیده‌تر می‌کند به همین دلیل برای حذف اثر سرعت نوشتن، یک مرحله پیش‌پردازش نرمالیزاسیون از نوع نمونه‌برداری مجدد روی داده‌ها انجام شده است.

نمونه‌برداری مجدد باعث می‌شود که فاصله بین هر دو نقطه متوالی دستنوشته مساوی شود. این فاصله را $L0 = 0.5$ در نظر گرفته‌ایم تا مطمئن شویم که هیچ جزئیاتی از دست نمی‌رود. نقاط، قبل از مرحله نمونه‌برداری مجدد در مختصات صحیح قرار گرفته‌اند، اما بعد از نمونه‌برداری مجدد مختصات اعشاری دارند که به دلیل رعایت فاصله مساوی بین هر دو نقطه است.

روش پیشنهادی به این صورت است که اولین نقطه برخورد قلم با صفحه را در نظر می‌گیریم، اگر فاصله اقلیدسی نقطه دوم تا این نقطه کمتر از $L0$ باشد، یک نقطه در فاصله نقطه $L0$ اول روی امتداد خط واصل بین دو نقطه قرار می‌دهیم و نقطه دوم را حذف می‌کنیم و اگر فاصله نقطه اول تا دوم بیشتر از $L0$ باشد، تا وقتی به فاصله کمتر از $L0$ از نقطه دوم برسیم، نقاط جدید با فاصله $L0$ از هم، روی خط واصل بین نقطه اول و دوم قرار می‌دهیم. این کار را تا وقتی به آخرین نقطه بدنه زیر-کلمه برسیم ادامه می‌دهیم.

بررسی خروجی این روش نشان می‌دهد که استفاده از خط واصل بین دو نقطه یا امتداد آن برای قرار دادن نقاط جدید، نتیجه مطلوبی دارد و نیازی به استفاده از روش‌های پیچیده‌تر مثل B-Spline یا Bezier [۲۸] نمی‌باشد. شکل ۴-۳ یک نمونه دستنوشته و شکل نرمالیزه شده آن را نشان می‌دهد.



شکل ۴-۳: راست: داده هموارسازی شده، چپ: بعد از نمونه‌برداری مجدد

۴-۳- علامت زدن نواحی قطعه‌بندی

نقطه‌ای که دو حرف متصل به هم در یک زیر-کلمه با بیش از دو حرف را جدا می‌کند، نقطه قطعه‌بندی نامیده می‌شود. در عمل نمی‌توان مرز دو حرف متصل در زیر-کلمه را به صورت منحصر به فرد مشخص ولی می‌توان تعدادی از نقاط متصل که کاندید برای نقطه قطعه‌بندی هستند مشخص کرد. هر کدام از این نقاط می‌تواند نقطه قطعه‌بندی باشد.

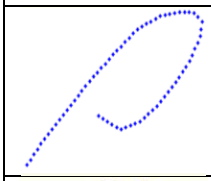
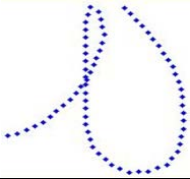
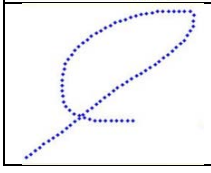
چون سیستم تشخیص زیر-کلمه بر اساس تشخیص بدنه حروف کار می‌کند، لازم است یک پایگاه داده از حروف در اختیار داشته باشیم. پایگاه داده حروف برخط تربیت مدرس [۸] فقط شامل حروف جدای فارسی است و چون هیچ پایگاه داده‌ای برای حروف اول، وسط و آخر فارسی موجود نیست، لازم بود حروف پایگاه داده زیر-کلمات برخط تربیت مدرس که تنها پایگاه داده زیر-کلمه برخط فارسی موجود است استخراج شود. بدین منظور برای تمام زیر-کلمات این پایگاه داده، نواحی قطعه‌بندی به صورت دستی نشانه‌گذاری شده است. این نشانه‌گذاری روی داده‌های پیش‌پردازش شده انجام شده است اما می‌توان با استفاده از مختصات محل نواحی قطعه‌بندی، از این Grand Truth برای مشخص کردن نواحی قطعه‌بندی در داده‌های خام هم استفاده کرد. در شکل ۴-۴ نواحی قطعه‌بندی برای دو زیر-کلمه نمونه مشخص شده‌اند.

شکل ۴-۴: نواحی قطعه‌بندی در دو زیر-کلمه پایگاه داده

۴-۴- گروه‌بندی بدنه حروف

در این پایان نامه حروف با توجه به شکل بدنه و محل قرار گرفتن در زیر-کلمه به ۸۱ گروه تقسیم شده‌اند که در پیوست أ آورده شده است. طبق جدول أ-۱ بعضی از حروف در بعضی دسته‌ها (جدا، ابتدا، وسط و انتها) بیش از یک شکل دارند. شکل‌های مختلف هر حرف به صورت دستی جدا و به گروه خود منتقل شده است. گروه‌های مربوط به اشکال مختلف نوشتن حرف "مـ" در جدول ۴-۱ نشان داده شده است. این سه شکل مربوط به انواع متداول نوشتن این حرف است و شکل‌های با تکرار کم در پایگاه داده و همچنین شکل‌های ناخوانا در هیچ یک از این گروه‌ها قرار نگرفته‌اند.

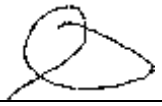
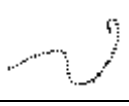
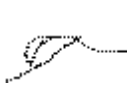
جدول ۴-۱: گروه‌های حرف "مـ" و یک نمونه تصویر از پایگاه داده

شماره گروه در دسته حروف اول	تصویر نمونه
۱۵	
۱۶	
۱۷	

در طراحی یک سیستم تشخیص دستنوشته، می‌توان اساس را بر این گذاشت که آن سیستم قادر به تشخیص انواع دستخط اما با خطای بیشتر باشد یا این که بتواند نمونه‌های رایج دستخط را با دقت بیشتری تشخیص دهد ولی میزان خطا برای تشخیص نمونه‌های غیر رایج اهمیتی نداشته باشد. در این تحقیق، تصمیم گرفتیم که رویکرد دوم را دنبال کنیم از این رو بعضی از نمونه‌های حروف که

ناخوانا بودند و یا شکل نوشتن آن‌ها در کل مجموعه دارای فراوانی کمی بود، جدا شدند تا در مرحله آموزش از آن‌ها استفاده نشود. تعدادی از این نمونه‌ها در جدول ۴-۲ آورده شده است.
در این جدول، حرف "ا" در نمونه نادرست، دارای قلاب بزرگ‌تر از حد معمول است، حرف "هـ" در نمونه نادرست شباهتی به این حرف ندارد و حروف "ـجـ" و "ـی" در نمونه نادرست به شکل غیر معمول نوشته شده‌اند که فراوانی بسیار کمی در پایگاه داده دارند.

جدول ۴-۲: مثال‌هایی از نمونه‌های درست و نادرست حروف

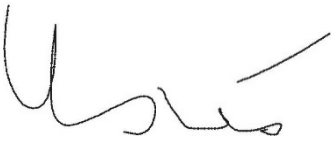
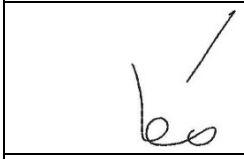
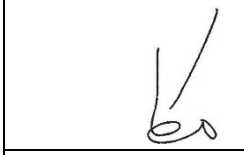
حرف	نمونه درست	نمونه نادرست
ا		
هـ		
ـجـ		
ـی		

۴-۵- گروه‌بندی زیر-کلمات بر اساس بدنه اصلی و علامت

گروه‌های حرف "مـ" در جدول ۴-۱ نشان داده شد. حرف "ـکـ" هم دارای دو شکل است، در صورتی که قبل از "ـا" یا "ـلـ/ـل" بیاید می‌تواند به صورت حلقه‌ای نوشته شود (گروه ۱۴ حروف وسط) و در غیر این صورت دارای شکل مشابه حرف "ـلـ" است (گروه ۱۳ حروف وسط). حرف "ـا" هم فقط یک شکل دارد (گروه ۱ حروف آخر). به این ترتیب زیر-کلمه "ملا" را می‌توان به ۳ شکل مختلف و زیر-کلمه "مکا" را به ۶ شکل مختلف نوشت که ۳ شکل آن با شکل‌های زیر-کلمه "ملا" مشترک است. این شکل‌ها در جدول ۴-۳ آورده شده است. به این ترتیب ۱۰۰۱ زیر-کلمه پایگاه داده با توجه به اشکال مختلف حروف را می‌توان به ۴۷۱۴ شکل بدنه زیر-کلمه مختلف نوشت که زیر-کلمات "ملا" و "مکا" گروه‌های ۱۲۹ تا ۱۳۴ را تشکیل می‌دهند.

همان طور که می‌بینید بعضی از شکل‌های ممکن بدنه زیر-کلمات در پایگاه داده وجود ندارند. سیستم شناسایی با رویکرد تجزیه‌ای می‌تواند چنین نمونه‌هایی را علی رغم این که در پایگاه داده زیر-کلمات وجود ندارند شناسایی کند.

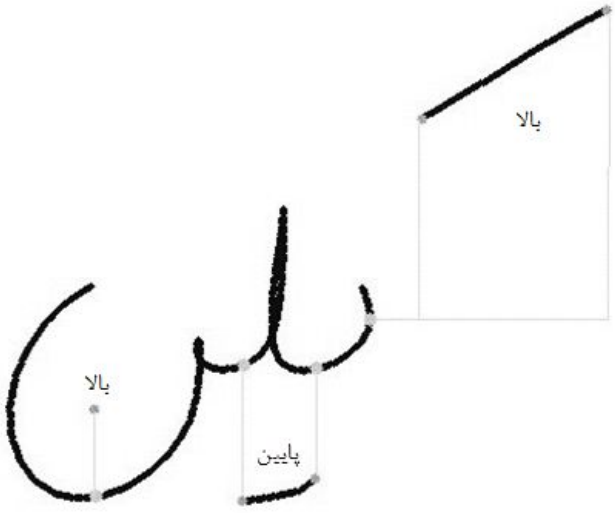
جدول ۴-۳: گروه‌های شکل بدنه زیر-کلمات "مکا" و "ملا"

گروه	زیر-کلمات	گروه بدنه حروف (راست به چپ)	مثال از پایگاه داده
۱۲۹	مکا و ملا	۱-۱۳-۱۵	
۱۳۰	مکا	۱-۱۴-۱۵	
۱۳۱	مکا و ملا	۱-۱۳-۱۶	
۱۳۲	مکا	۱-۱۴-۱۶	
۱۳۳	مکا و ملا	۱-۱۳-۱۷	موجود نیست
۱۳۴	مکا	۱-۱۴-۱۷	موجود نیست

همان طور که در بخش ۴-۱ ذکر شد، به دلیل دشواری تشخیص علامت‌ها، طراحی سیستم تشخیص دستنوشته مستقل از مجموعه لغات امکان پذیر نیست از این رو نتایج خروجی قسمت تشخیص گروه حروف را بر اساس علامت فیلتر می‌کنیم. این فیلتر بر اساس تعداد حرکت‌های بالا و پایین بدنه اصلی زیر-کلمه عمل می‌کند، لذا الگوریتمی برای تشخیص محل حرکت‌های علامت‌های زیر-کلمه نیاز داریم.

۴-۵-۱- روش پیشنهادی برای تعیین محل حرکت نسبت به بدنه اصلی زیر-کلمه

اگر اولین نقطه حرکت (به ترتیب نوشتن) پایین‌تر از اولین نقطه از بدنه اصلی که با علامت مذکور همپوشانی افقی دارد باشد، حرکت را زیر بدنه اصلی در نظر می‌گیریم، در غیر این صورت، حرکت، بالای بدنه اصلی است. در صورتی که حرکت با بدنه زیر-کلمه همپوشانی افقی نداشته باشد، نزدیک‌ترین نقطه از نظر فاصله افقی با حرکت را در نظر می‌گیریم. تعیین محل حرکت‌ها با این روش تقریباً بدون خطا انجام می‌شود. سپس تعداد حرکت‌های بالا و پایین بدنه اصلی برای زیر-کلمه به صورت جداگانه محاسبه می‌شوند تا در مرحله فیلتر کردن نتایج که در بخش ۴-۹ بررسی می‌شود مورد استفاده قرار بگیرد. روش تعیین محل حرکت‌ها نسبت به بدنه زیر-کلمه در شکل ۴-۵ نشان داده شده است.



شکل ۴-۵: مثال تعیین محل علامت زیر-کلمه "یکن"

در این شکل، سرکش "ـکـ" با بدنه زیر-کلمه همپوشانی ندارد به همین دلیل نزدیک‌ترین نقطه با فاصله افقی از سرکش در نظر گرفته می‌شود، چون اولین نقطه حرکت سرکش (نقطه قرمز) از نقطه مذکور پایین‌تر نیست، سرکش بالای بدنه اصلی است.

برای رفع ابهام محل نقطه حروف "ج/چ" که نقطه با بدنه اصلی در دو محل همپوشانی افقی دارد، کافی است که اولین محلی همپوشانی (به ترتیب نوشته شدن) را در نظر بگیریم. همچنین اگر نقطه این حروف طوری گذاشته شود که فقط با قسمت پایین حرف همپوشانی داشته باشد می‌تواند منجر به خطا شود که برای رفع این مشکل، ۲۰٪ آخر حرف (زیر-کلمه) را در نظر نمی‌گیریم تا کشیدگی انتهای حرف "ج/چ" به عنوان محل همپوشانی افقی با نقطه در نظر گرفته نشود. به این ترتیب این روش ساده قادر است محل حرکت‌ها را به درستی تشخیص دهد.

۴-۵-۲- گروه‌بندی مجموعه لغات بر اساس تعداد و محل حرکت‌ها

برای مثال زیر-کلمه "تشخیص" را در نظر بگیرید. این زیر-کلمه دارای یک دونقطه، یک تک‌نقطه و یک سه‌نقطه در بالای بدنه اصلی و یک دونقطه در زیر بدنه اصلی است. دونقطه را می‌توان در یک حرکت به صورت یک خط از راست به چپ (یا چپ به راست) یا در دو حرکت به صورت دو تک‌نقطه و سه‌نقطه را می‌توان در یک حرکت به صورت یک کمان، دو حرکت به صورت یک دونقطه به شکل خط و یک نقطه بالا یا پایین آن یا سه حرکت به صورت سه تک‌نقطه نوشت. بر این اساس زیر-کلمه "تشخیص" می‌تواند دارای ۳، ۴، ۵ یا ۶ حرکت در بالا و ۱ یا ۲ حرکت در پایین باشد. بنابراین زیر-کلمه "تشخیص" در تمام گروه‌های جدول ۴-۴ قرار خواهد گرفت.

جدول ۴-۴: گروه‌های زیر-کلمه "تشخیص" بر اساس تعداد حرکت‌ها در بالا و پایین بدنه

شماره گروه	تعداد حرکت‌های بالا	تعداد حرکت‌های پایین
۱	۳	۱
۲	۳	۲
۳	۴	۱
۴	۴	۲
۵	۵	۱
۶	۵	۲
۷	۶	۱
۸	۶	۲

این گروه‌بندی را برای تمام زیر-کلماتِ مجموعه متنی انجام می‌دهیم و هر زیر-کلمه را در گروه‌های مربوط اضافه می‌کنیم.

۴-۵-۳- گروه‌بندی زیر-کلمات بر اساس وجود علامت در بالا یا پایین بدنه

زیر-کلمات مجموعه متنی را بر اساس وجود حرکت در بالا یا پایین بدنه هم تقسیم‌بندی می‌کنیم. هر زیر-کلمه بر این اساس می‌تواند در یکی یا بیشتر از یکی از گروه‌های زیر قرار بگیرد:

۱. بدون علامت
۲. علامت(ها) فقط در پایین بدنه اصلی
۳. علامت(ها) فقط در بالای بدنه اصلی
۴. دارای علامت هم در بالا و هم پایین بدنه اصلی

برای این گروه‌بندی از اطلاعات مرحله قبل استفاده می‌شود. مثلاً تعداد حرکت‌های بالا و پایین در زیر-کلمه "تشخیص" همواره غیر صفر است پس این زیر-کلمه فقط در گروه ۴ قرار می‌گیرد.

۴-۵-۴- گروه‌بندی مجموعه متنی زیر-کلمات بر اساس بدنه اصلی

زیر-کلمات "ببر، تبر، تیر، بیر، پیر" دارای شکل بدنه اصلی یکسان هستند و فقط در علامت فرق دارند، چنین زیر-کلماتی همه در یک گروه قرار می‌گیرند. از این مجموعه زیر-کلمات همگروه بر اساس بدنه اصلی در مرحله فیلتر کردن نتیجه تشخیص گروه بدنه اصلی زیر-کلمات استفاده می‌شود.

۴-۶- روش پیشنهادی برای ایجاد داده‌های مصنوعی

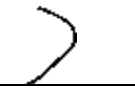
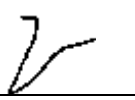
پایگاه‌های داده موجود شامل حروف جدا و زیر-کلمات متداول فارسی هستند، به همین دلیل فراوانی بعضی از حروف برای آموزش مدل مخفی مارکوف کافی نیستند. مدل مخفی مارکوف هم مشابه شبکه عصبی دارای وزن‌هایی است که در ابتدا به صورت تصادفی مقداردهی می‌شوند و با آموزش، به تخمین دقیق‌تری می‌رسند. اگر تعداد نمونه‌های آموزشی کم باشد، مدل مخفی مارکوف عملاً بی‌استفاده می‌شود چون قادر به بازسازی نمونه‌های متنوع نیست و در مرحله تست، احتمال رویداد دنباله مشاهده داده تست مقدار کمی می‌شود.

برای حل این مشکل، برای گروه‌هایی که کمتر از ۱۰۰ نمونه آموزشی دارند، نمونه‌های مصنوعی می‌سازیم تا با نمونه‌های آموزشی به ۱۰۰ نمونه برسد. برای ایجاد داده‌های مصنوعی فقط از داده‌های آموزشی استفاده می‌شود. روش کار به این صورت است که مسیر همترازی بین زاویه نقاط متوالی هر دو نمونه آموزشی را با الگوریتم DTW محاسبه می‌کنیم و با گسترش سیگنال مشابه آنچه در بخش ۳-۲-۲ نشان داده شد، سیگنال مصنوعی را به دست می‌آوریم. در این روش با داشتن n نمونه آموزشی، می‌توان C_2^{n-1} نمونه مصنوعی تولید کرد که تعداد مورد نیاز به صورت تصادفی از کل نمونه‌های مصنوعی قابل تولید انتخاب می‌شوند.

تعدادی از نمونه‌های آموزشی و نمونه مصنوعی ایجاد شده از آن‌ها در جدول ۴-۵ آورده شده است. ممکن است بعضی نمونه‌های مصنوعی تولید شده، شبیه حرف‌های سازنده آن نباشد و در بدترین حالت مثل سطر آخر جدول ۴-۵، نمونه مصنوعی تولید شده شبیه به یکی دیگر از گروه‌های همان دسته (حروف جدا، ابتدا، وسط و انتها) باشد، به همین دلیل لازم است فقط از نمونه‌های مصنوعی معتبر استفاده شود. تشخیص اعتبار نمونه‌های مصنوعی با استفاده از مشاهده شکل آن‌ها ممکن است که نیاز به یک مرحله جداسازی دستی نمونه‌های مصنوعی نامناسب دارد. با توجه به این که بعضی شکل‌های حروف در بعضی دستخط‌ها کمیاب هستند بهتر است به جای تولید نمونه مصنوعی برای این شکل حروف، از آن‌ها صرف نظر کرد و روی دستخط متداول تمرکز کرد.

الگوریتم DTW به این دلیل که قادر است نقاط همتراز دو سیگنال غیر هم طول را پیدا کند برای ترکیب آن‌ها مناسب است در حالی که دیگر معیارهای فاصله نظیر فاصله اقلیدسی این ویژگی مثبت را ندارند. افزودن نویز به یک سیگنال دستنوشته برای ایجاد داده مصنوعی نیز روش مناسبی نیست چون داده نویزی شده علاوه بر این که از نظر داشتن نویز برای آموزش طبقه‌بند، مناسب نیست، نمی‌توان آن را یک نمونه متفاوت دانست زیرا شکل آن مشابه نمونه اصلی است. همچنین تعداد نمونه‌های مصنوعی قابل تولید با این روش محدود است.

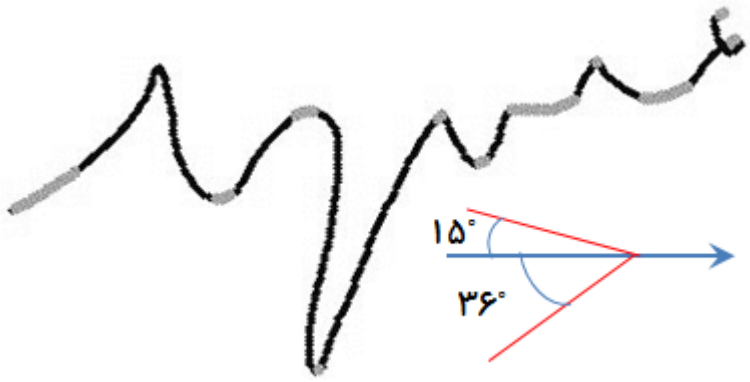
جدول ۴-۵: مثال‌هایی از نمونه‌های مصنوعی تولید شده

حرف	نمونه آموزشی ۱	نمونه آموزشی ۲	نمونه مصنوعی
د			
ط			
ـهـ			
ـج			

۴-۷- الگوریتم قطعه‌بندی

برای قطعه‌بندی زیر-کلمه به حروف، روش [۹] را مبنا قرار داده‌ایم، ولی با توجه به نتایج ضعیف این روش روی پایگاه داده مورد آزمایش اصلاحاتی داده شده است تا نتیجه قطعه‌بندی بهبود داده شود. تمام مقادیر آستانه به صورت تجربی به دست آمده‌اند و طوری در نظر گرفته شده‌اند که بین پیدا کردن بیشترین نقاط صحیح قطعه‌بندی و به وجود آمدن نقاط قطعه‌بندی اضافی کمتر، یک مصالحه منطقی وجود داشته باشد.

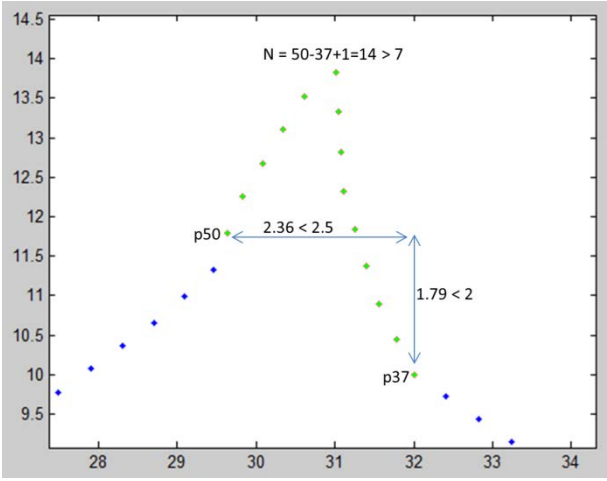
نقاطی که زاویه خط متصل کننده به نقطه بعد بین ۱۵-۱۸۰ تا ۳۶+۱۸۰ درجه باشد، به عنوان نقاط اولیه کاندید قطعه‌بندی انتخاب می‌شوند. در شکل ۴-۶ نقاط اولیه قطعه‌بندی در زیر-کلمه "مشهد" نشان داده شده است.



شکل ۴-۶: نقاط اولیه کاندید قطعه‌بندی در زیر-کلمه مشهد (نقاط ضخیم) با توجه به زاویه با محور افقی

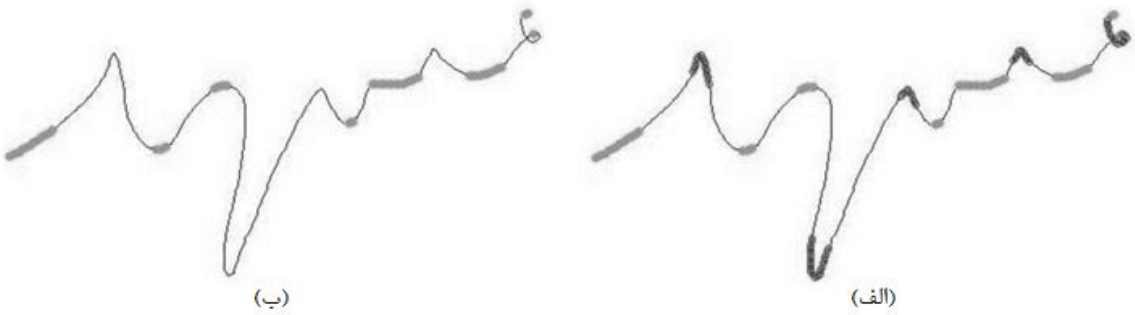
سپس نقاطی که در محدوده حلقه‌ها قرار گرفته‌اند، شناسایی و از نقاط کاندید قطعه‌بندی حذف می‌شوند. روش پیشنهادی برای پیدا کردن حلقه‌ها به این صورت است که از اولین نقطه شروع می‌کنیم و نقاط نزدیک آن از نظر فاصله را پیدا می‌کنیم. نقاط نزدیک یک نقطه را نقاطی تعریف

کرده‌ایم که در جهت x، ۲/۵ واحد و در جهت y، ۲ واحد از آن نقطه فاصله دارند. در پیدا کردن نقاط نزدیک یک نقطه، نقاطی که از نظر زمانی قبل از آن هستند و همچنین تا ۷ نقطه بعد از آن نقطه را در نظر نمی‌گیریم. بین نقطه شروع تا آخرین نقطه نزدیک آن را به عنوان حلقه در نظر می‌گیریم و نقطه شروع بعدی را از نقطه بعد از آخرین نقطه حلقه می‌گیریم. اگر یک نقطه هیچ نقطه نزدیکی نداشته باشد، نمی‌تواند نقطه شروع حلقه باشد و نقطه بعدی به عنوان نقطه شروع در نظر گرفته می‌شود. شکل ۴-۷ نحوه تعیین این نقاط را نشان می‌دهد.



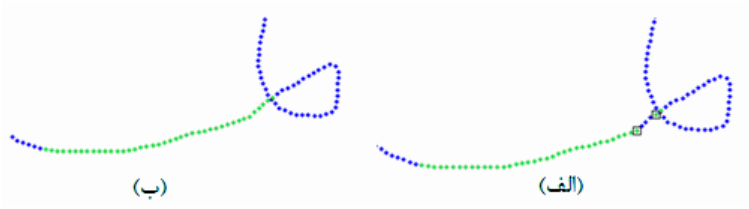
شکل ۴-۷: نحوه تعیین محل حلقه‌ها و نقاط تیز، با شروع از نقطه ۳۷، نقطه ۵۰ آخرین نقطه نزدیک به این نقطه است که چون بیش‌تر از ۷ نقطه فاصله دارد نقاط این فاصله به عنوان حلقه/نقاط تیز شناخته می‌شوند.

نقاطی از کاندیدهای قطعه‌بندی که جزو حلقه هستند، حذف می‌شوند. قسمت‌های نوک تیز مثل دندانه‌ها نیز در این روش به عنوان حلقه شناخته می‌شوند که این امر نه تنها هیچ مشکلی را ایجاد نمی‌کند بلکه مفید نیز هست چون این قسمت‌ها نمی‌توانند جزو نقاط قطعه‌بندی باشند. شکل ۴-۸ نقاط تشخیص داده شده به عنوان حلقه و نقاط نوک تیز و نتیجه قطعه‌بندی بعد از حذف این نقاط را نشان می‌دهد.



شکل ۴-۸: (الف) نقاط پررنگ به عنوان حلقه شناسایی شده‌اند. (ب) نقاط کاندید قطعه‌بندی بعد از حذف حلقه‌ها

در مرحله بعد، نقاط کاندید متوالی با حداقل طول ۱، را به عنوان ناحیه قطعه‌بندی تعریف می‌کنیم و نواحی قطعه‌بندی که کمتر از ۵ نقطه با هم فاصله دارند را ادغام می‌کنیم. مثالی از ادغام نواحی نزدیک در شکل ۴-۹ دیده می‌شود.

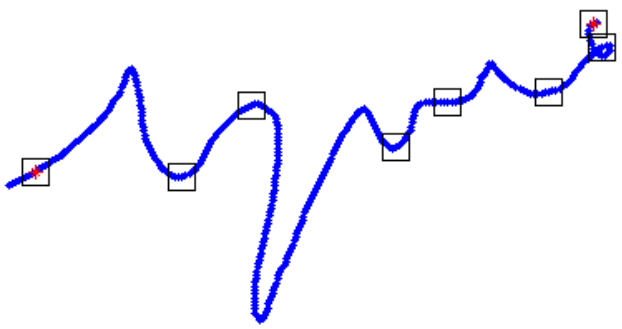


شکل ۴-۹: مثالی از ادغام نواحی نزدیک. (الف) دو ناحیه نزدیک به هم (ب) ناحیه ادغام شده

خروجی مرحله حذف نواحی کوچک و ادغام نواحی نزدیک برای زیر-کلمه "مشهد" در شکل ۴-۱۰ دیده می‌شود.

شکل ۴-۱۰: نواحی قطعه‌بندی در زیر-کلمه "مشهد" بعد از ادغام نواحی نزدیک و حذف نواحی کوچک

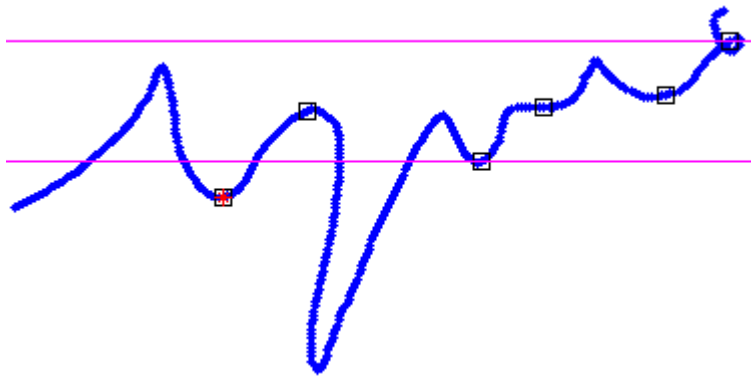
نقطه وسط هر ناحیه قطعه‌بندی، نقطه قطعه‌بندی در نظر می‌گیریم. نقاط قطعه‌بندی که در ۵ درصد اول یا ۱۰ درصد انتهای زیر-کلمه قرار دارند، حذف می‌شوند. این کار به این دلیل انجام می‌شود که غالباً این قسمت‌ها جزو حرف اول یا آخر هستند و در بعضی حروف، شبیه به نواحی قطعه‌بندی هستند. این مراحل در شکل ۴-۱۱ نشان داده شده است.



شکل ۴-۱۱: مربع توخالی: نقاط قطعه‌بندی، نقاط داخل مربع اول و آخر: نقاط قطعه‌بندی حذف شده در ابتدا و انتهای زیر-کلمه

همه نقاط قطعه‌بندی باید روی خط زمینه قرار داشته باشند اما با توجه به این که در دستنوشته این مورد چندان رعایت نمی‌شود و این مسئله در مورد نمونه‌های پایگاه داده که با استفاده از یک تبلت گرافیکی بدون صفحه نمایش جمع‌آوری شده‌اند، بیشتر به چشم می‌آید، به جای خط زمینه، یک بازه روی محور y را به عنوان بازه وجود نقاط قطعه‌بندی در نظر گرفته‌ایم. برای به دست آوردن این بازه به صورت زیر عمل شده است:

برای زیر-کلماتی که بیشتر از ۵ نقطه قطعه‌بندی برای آن‌ها پیدا شده است، نقطه قطعه‌بندی که در جهت y وسط است را در نظر می‌گیریم. در صورتی که تعداد نقاط قطعه‌بندی که با این نقطه اختلاف ارتفاعی کمتر از ۲۵ درصد ارتفاع بدنه اصلی زیر-کلمه دارند، بیشتر از ۷۵ درصد کل نقاط قطعه‌بندی باشد، بازه مربوط به پایین‌ترین تا بالاترین این نقاط را به عنوان بازه وجود نقاط قطعه‌بندی در نظر می‌گیریم، در غیر این صورت، از این مرحله صرف نظر می‌شود. در آخرین مرحله قطعه‌بندی، نقاط قطعه‌بندی خارج از بازه وجود نقاط قطعه‌بندی حذف می‌شوند که برای زیر-کلمه نمونه "مشهد" در شکل ۴-۱۲ نشان داده شده است.



شکل ۴-۱۲: بازه وجود نقاط قطعه‌بندی (ناحیه عمودی بین دو خط) و نقطه قطعه‌بندی حذف شده (مربع آخر)

۴-۸- تشخیص گروه حروف

برای تشخیص گروه هر حرف، استفاده از مدل مخفی مارکوف را پیشنهاد می‌دهیم. برای هر گروه یک مدل مخفی مارکوف با استفاده از داده‌های آموزشی آن گروه، آموزش داده می‌شود، به این ترتیب ۸۱ مدل به تعداد گروه‌ها خواهیم داشت. از ۷۰ درصد داده‌های هر گروه برای آموزش و از ۳۰ درصد باقی‌مانده برای تست استفاده شده است. در مرحله تست، با توجه به محل قرارگیری حرف در زیر-کلمه، بردار ویژگی استخراج شده از حرف به تمام مدل‌های آن دسته (جدا، ابتدا، وسط یا انتها) داده می‌شود که خروجی هر مدل متناسب با میزان تعلق حرف به آن مدل است. این خروجی لگاریتم احتمال تولید دنباله مشاهده (بردار ویژگی) حرف است.

چون مدل مخفی مارکوف دارای دنباله مشاهدات با مقادیر گسسته است، یا باید بردار ویژگی دارای مقادیر گسسته باشد که در این صورت همان دنباله مشاهده در مدل مخفی مارکوف خواهد بود و یا از طریق یک الگوریتم خوشه‌بندی[1] مثل مدل مخلوط گوسی[2] یا k-means، بردار ویژگی دارای مقادیر پیوسته را به یک بردار دارای مقادیر گسسته تبدیل کرد. ما از بردار ویژگی دارای مقادیر گسسته استفاده کرده‌ایم تا نیاز به یک مرحله اضافی نباشد. به این ترتیب این بردار ویژگی، همان دنباله مشاهدات در مدل مخفی مارکوف است که در بخش بعد توضیح داده می‌شود.

استفاده از طبقه‌بند مدل مخفی مارکوف به دلایل زیر است:

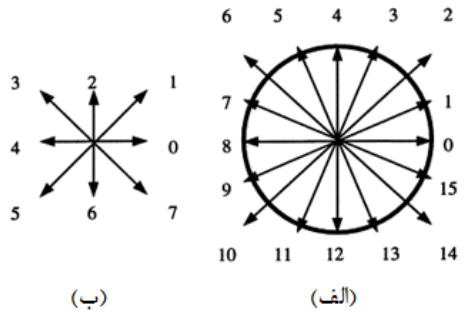
- مدل مخفی مارکوف در کارهای گذشته نتیجه قابل قبولی داشته است و بیشتر از انواع دیگر طبقه‌بندها استفاده شده است.
- این مدل، ابزاری مخصوص سیگنال‌های زمانی است و سیگنال دستنوشته برخط نیز یک سیگنال زمانی است، از این رو استفاده از مدل مخفی مارکوف برای مدل کردن دستنوشته برخط مناسب است.
- پارامترهای مدل مخفی مارکوف برای مدل هر حرف به صورت جداگانه قابل تنظیم است. از این رو می‌توان برای حروفی که پیچیدگی کمتری دارند مدل‌های ساده‌تر(با تعداد مشاهدات و حالت‌های کمتر) و برای حروف پیچیده‌تر از مدل‌های پیچیده‌تر استفاده کرد.
- طول بردار مشاهده می‌تواند متغیر باشد. در صورتی که بخواهیم در آینده، از یک ویژگی استفاده کنیم که باعث شود طول بردار ویژگی برای نمونه‌های مختلف ثابت نباشد این خاصیت مدل مخفی مارکوف بسیار مفید است چون نیاز به هم طول کردن بردارهای ویژگی را از بین می‌برد در حالی که طبقه‌بندهای پر استفاده دیگر مثل شبکه عصبی و ماشین بردار پشتیبان[3] نیاز به طول بردار ویژگی ثابت دارند.

۴-۸-۱- استخراج ویژگی

ویژگی استخراج شده مشابه ویژگی‌های به کار رفته در [۲۳] است. ابتدا حرف با روش گفته شده در بخش ۴-۲-۲ به n نقطه نرمالیزه می‌شود که با آزمایش مقادیر ۸ تا ۲۰ برای دسته حروف جدا، ابتدا، وسط و انتها به ترتیب برابر با ۱۴،۱۳،۱۱ و ۱۴ به دست آمده است، سپس زاویه بردارهای متصل کننده نقاط متوالی محاسبه می‌شود. برای گسسته‌سازی مقدار زوایا، از زنجیره کدهای فریمن ۱۶ تایی استفاده می‌شود که با آزمایش جهت‌های شکل ۴-۱۳نتیجه بهتری داشته است. کد جهت‌ها، به عنوان

[1] Clustering
[2] Gaussian Mixture Model (GMM)
[3] Support Vector Machine (SVM)

ویژگی در طبقه‌بند، مورد استفاده قرار می‌گیرد. کد نزدیک‌ترین جهت به هر زاویه به جای مقدار زاویه قرار می‌گیرد. طول بردار کدهای جهتی یکی کمتر از تعداد نقاط در حرف نرمالیزه شده است.



شکل ۴-۱۳: کدهای (الف) ۱۶ و (ب) ۸ جهتی فریمن

۴-۸-۲- پارامترهای مدل مخفی مارکوف

پارامترهای قابل تنظیم مدل مخفی مارکوف عبارتند از:

- تعداد مشاهدات
- طول بردار مشاهده
- تعداد حالت‌های مخفی
- توپولوژی مدل مخفی مارکوف
- وزن‌های اولیه ماتریس انتقال حالت

هر کدام از این پارامترها به صورت جداگانه برای مدل هر گروه قابل تنظیم است، اما ما تمام این پارامترها را در هر دسته برای تمام مدل‌ها ثابت در نظر گرفتیم، با این کار ممکن است بهترین نتیجه شناسایی به دست نیاید ولی نیازی به بهینه کردن پارامترها برای تک‌تک گروه‌ها که وقت زیادی می‌گیرد نیست، ضمن این که باعث سهولت پیاده‌سازی نرم‌افزاری می‌شود.

تعداد مشاهدات

در بخش ۴-۸-۱ تعداد مشاهدات را برابر ۱۶ در نظر گرفتیم که عبارتند از مقادیر گسسته ۱ تا ۱۶.

طول بردار مشاهده

در بخش ۴-۸-۱ این مقدار را ۱۳،۱۲،۱۰ و ۱۳ برای دسته‌های حروف جدا، ابتدا، وسط و انتها گرفتیم.

تعداد حالت‌های مخفی

تعداد بهینه حالت‌های مخفی معمولاً از طریق آزمایش به دست می‌آید. وقتی تعداد مشاهدات و طول بردار مشاهده کوچک باشد، تعداد حالت‌های مخفی هم مقدار کوچکی است که راحت‌ترین راه به دست آوردن مقدار بهینه آن از طریق آزمایش است. آزمایش مقادیر ۶ تا ۳۰، نشان داد که مقدار ۲۹ برای دسته حروف جدا و ۲۷ برای بقیه دسته‌ها به بهترین نتیجه منجر می‌شود.

توپولوژی مدل مخفی مارکوف

با آزمایش توپولوژی‌های معرفی شده در بخش ۳-۱-۴ توپولوژی ارگادیک به دلیل نتیجه بهتر، انتخاب شده است.

وزن‌های اولیه ماتریس انتقال حالت

مقادیر اولیه وزن‌ها معمولاً به صورت تصادفی انتخاب می‌شوند. چون این مقادیر، احتمال هستند باید بین صفر و یک باشند و طبق (۳-۳) باید مجموع هر سطر برابر یک باشد ضمن این که باید توجه شود که وزن‌هایی که مقدار صفر داشته باشند، هر چقدر که آموزش ببینند، تغییر نمی‌کنند، از این رو باید تمام مقادیر وزن‌ها غیر صفر باشند. به همین دلیل مشابه [۳] وزن‌هایی که صفر می‌شوند را برابر یک صدم کوچک‌ترین وزن در آن سطر گرفتیم و از بقیه وزن‌ها مقداری کم شده است که مجموع سطر یک شود.

۴-۸-۳- مرحله تست با استفاده از مدل مخفی مارکوف

ورودی سیستم بازشناسی، قطعاتی از زیر-کلمه است که توسط الگوریتم قطعه‌بندی استخراج شده است، این روش دارای خطاست که بسته به نوع خطا ممکن است یکی از حالت‌های زیر اتفاق بیفتد:

۱. قطعه استخراج شده، قسمتی از یک حرف باشد.

۲. قطعه استخراج شده شامل بیش از یک حرف باشد به عنوان مثال، یک حرف به اضافه قسمت اول حرف بعد.

خطاهای مربوط به حالت اول تا حدودی قابل جبران هستند، اما خطای نوع دوم که مربوط به نقاط قطعه‌بندی از دست رفته است در نهایت منجر به ایجاد خطا در بازشناسی زیر-کلمه خواهد شد. از این رو الگوریتم قطعه‌بندی باید حداکثر نقاط قطعه‌بندی را پیدا کند هر چند منجر به ایجاد نقاط قطعه‌بندی اضافی شود. البته چون وجود نقاط قطعه‌بندی اضافی، بار محاسباتی را زیاد و دقت مرحله شناسایی را کم می‌کند، تعداد این نقاط نباید خیلی بیشتر از تعداد نقاط درست قطعه‌بندی باشد.

جبران خطای نوع اول در الگوریتم قطعه‌بندی به این صورت است که ترکیب‌های مختلف وجود نقاط قطعه‌بندی را در نظر می‌گیریم. به عنوان مثال برای یک زیر-کلمه که ۳ نقطه قطعه‌بندی برای آن به دست آمده است، $۲^۳=۸$ حالت نشان داده شده در جدول ۴-۶ وجود دارد:

جدول ۴-۶: ترکیب‌های مختلف وجود نقاط قطعه‌بندی برای زیر-کلمه با ۳ نقطه قطعه‌بندی

شماره	تعداد حرف زیر-کلمه خروجی	نقطه قطعه‌بندی اول	نقطه قطعه‌بندی دوم	نقطه قطعه‌بندی سوم
۱	یک حرفی	حذف	حذف	حذف
۲	دو حرفی	وجود دارد	حذف	حذف
۳		حذف	وجود دارد	حذف
۴		حذف	حذف	وجود دارد
۵	سه حرفی	وجود دارد	وجود دارد	حذف
۶		وجود دارد	حذف	وجود دارد
۷		حذف	وجود دارد	وجود دارد
۸	چهار حرفی	وجود دارد	وجود دارد	وجود دارد

برای هر زیر-کلمه، تمام این حالت‌های ممکن را در نظر می‌گیریم، در هر کدام از این حالت‌ها، برای هر قطعه، ۱۰ گزینه اول بازشناسی را به دست می‌آوریم. خروجی مدل مخفی مارکوف برای هر قطعه، لگاریتم مقداری است که از (۳-۹) به دست می‌آید، چون مقدار احتمال همیشه بین ۰ تا ۱ است، لگاریتم آن هم مقداری بین منفی بی‌نهایت تا صفر است.

۱۰ گزینه اولی که بیشترین مقدار درست‌نمایی[1] را دارند برای هر قطعه ذخیره می‌کنیم. برای یک زیر-کلمه با n قطعه، با ترکیب حالت‌های مختلف ۱۰ گزینه اول بازشناسی هر قطعه، n^{10} حالت خواهیم داشت که فقط ۱۰ حالتی که بیشترین امتیاز را دارند در نظر می‌گیریم. امتیاز هر حالت برابر با میانگین هندسی مقدار درست‌نمایی قطعه‌های آن است.

تعداد ترکیب‌های وجود نقاط قطعه‌بندی با p نقطه، برابر با p^2 است که برای هر کدام ۱۰ حالت از ترکیب گروه‌های مختلف مربوط به گزینه‌های بازشناسی قطعه را داریم. از این مجموعه فقط ۱۰ گزینه دارای بالاترین امتیاز را نگه می‌داریم.

[1] Likelihood

۴-۹- فیلتر کردن نتیجه تشخیص گروه قطعات با استفاده از علامت‌ها

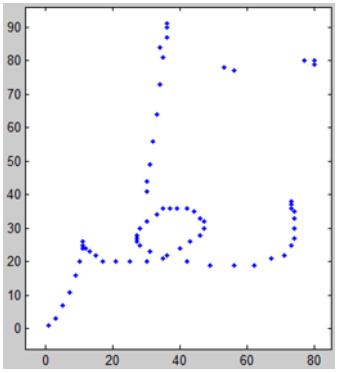
فیلتر کردن نتایج مرحله قبل با استفاده از علامت‌ها را در دو مرحله انجام دادیم، در مرحله اول فقط وجود علامت در بالا یا پایین زیر-کلمه را در نظر می‌گیریم و در مرحله دوم تعداد حرکت‌ها را هم در نظر می‌گیریم. نحوه به دست آوردن تعداد حرکت‌های بالا و پایین بدنه اصلی زیر-کلمه و تشکیل مجموعه زیر-کلمات همگروه بر اساس تعداد حرکت‌های بالا و پایین بدنه (GNUD)، مجموعه زیر-کلمات همگروه بر اساس وجود علامت در بالا یا پایین بدنه (GUD) و همچنین مجموعه زیر-کلمات همگروه بر اساس بدنه اصلی (GM) در بخش ۴-۵ توضیح داده شد.

خروجی مرحله قبل، تعدادی حرف بی‌علامت است که ابتدا با توجه به گروه بدنه حروف، زیر-کلمات منطبق از GM استخراج می‌شوند. سپس با استفاده از علامت‌های زیر-کلمه، این نتایج را محدود می‌کنیم. بدین منظور در مرحله اول با توجه به وجود علامت‌ها در پایین و بالای بدنه، زیر-کلماتی که در گروه مربوطه در GUD قرار ندارند، حذف می‌شوند. این مرحله بدون خطا انجام می‌شود.

سپس با توجه به تعداد حرکت‌ها در بالا و پایین بدنه اصلی، زیر-کلمات باقی‌مانده که در گروه مربوطه در GNUD نیستند حذف می‌شوند. این مرحله هم در صورتی که نویسنده، تعداد حرکت‌های علامت‌ها را رعایت کرده باشد، خطایی ندارد.

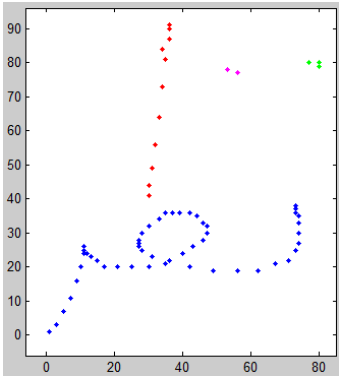
بعد از اعمال این دو فیلتر، ممکن است باز هم اولین در گزینه بازشناسی بیش از یک زیر-کلمه موجود باشد که در این صورت، این زیر-کلمات در شکل بدنه اصلی، وجود علامت در پایین یا بالای بدنه و تعداد حرکت‌های بالا و پایین بدنه اصلی مشترک هستند. برای مثال دو زیر-کلمه "تبر" و "تیر" در صورتی که هر علامت فقط با یک حرکت نوشته شود چنین حالتی دارند. در این صورت باید روشی برای تفکیک علامت‌ها ارائه کرد، این کار در این تحقیق مورد مطالعه قرار نگرفته است و در صورت بروز چنین موردی، ممکن است در یک گزینه بازشناسی، بیش از یک زیر-کلمه منطبق وجود داشته باشد.

با یک مثال واقعی از شبیه‌سازی انجام شده، روال تشخیص زیر-کلمه که در شکل ۴-۱نشان داده شد را توضیح می‌دهیم. زیر-کلمه ورودی به سیستم "نظر" در شکل ۴-۱۴است.



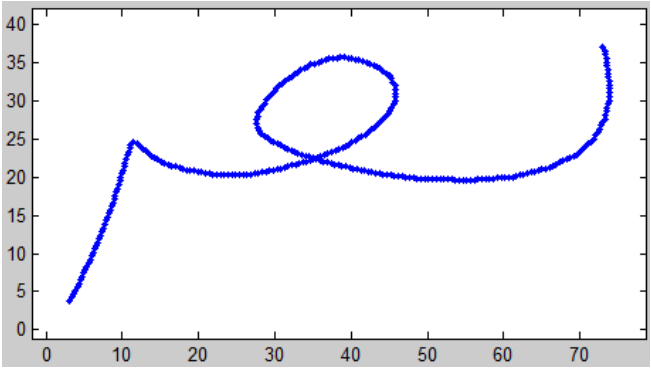
شکل ۴-۱۴: زیر-کلمه ورودی به سیستم شناسایی

این زیر-کلمه دارای ۳ حرکت در بالای بدنه اصلی است که در شکل ۴-۱۵با رنگ متفاوت نشان داده شده‌اند.



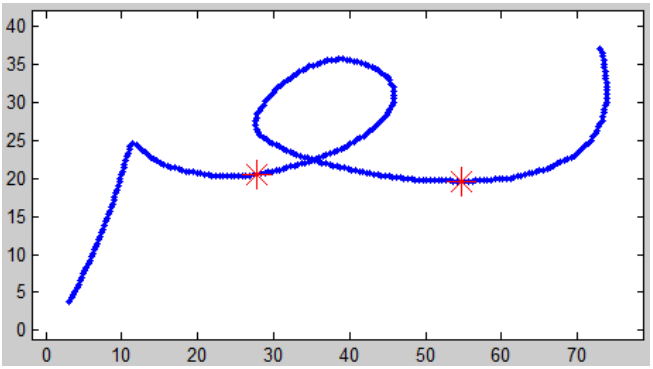
شکل ۴-۱۵: حرکت‌های زیر-کلمه "نظر" در مرحله دوم از بدنه اصلی جدا شده‌اند.

سپس بدنه اصلی زیر-کلمه، پیش‌پردازش می‌شود. نتیجه پیش‌پردازش در شکل ۴-۱۶نشان داده شده است.



شکل ۴-۱۶: بدنه پیش‌پردازش شده زیر-کلمه "نظر"

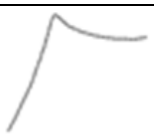
در مرحله بعد باید نقاط قطعه‌بندی با روش پیشنهادی پیدا شوند. همان طور که در شکل ۴-۱۷دیده می‌شود، نقاط قطعه‌بندی به درستی شناسایی شده‌اند.



شکل ۴-۱۷: نقاط قطعه‌بندی پیدا شده در بدنه زیر-کلمه "نظر"

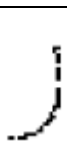
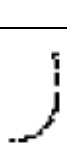
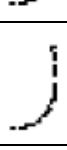
با در نظر گرفتن این که این نقاط می‌توانند اشتباه باشند، ترکیب‌های مختلف نقاط قطعه‌بندی از نظر وجود داشتن را به دست می‌آوریم. با دو نقطه قطعه‌بندی چهار حالت جدول ۴-۶ می‌تواند وجود داشته باشد.

جدول ۴-۷: ترکیب‌های مختلف وجود نقاط قطعه‌بندی برای زیر-کلمه "نظر"

قطعات	قطعه اول	قطعه دوم	قطعه سوم
ترکیب ۱			
ترکیب ۲			
ترکیب ۳			
ترکیب ۴			

ترکیب‌هایی که متوسط احتمال تعلق قطعات آن به گروه حروف بیشتر است را در نظر می‌گیریم. ۱۱ ترکیب اول در جدول ۴-۷ آورده شده است که گزینه ۱۱ صحیح است. گزینه‌های ۴، ۹ و ۱۰ هیچ زیر-کلمه منطبقی در مجموعه زیر-کلمات متنی ندارند و حذف می‌شوند. گزینه‌های ۱، ۲، ۶ و ۸ دارای علامت در زیر بدنه اصلی هستند و به این دلیل حذف می‌شوند. گزینه‌های ۳، ۵ و ۷ دارای ۳ حرکت در بالای بدنه نیستند و از لیست گزینه‌ها حذف می‌شوند و گزینه ۱۱ که زیر-کلمه منطبق نظر است تبدیل به اولین گزینه بازشناسی می‌شود.

جدول ۴-۸: گزینه‌های بازشناسی برای زیر-کلمه "نظر" و مراحل حذف گزینه‌های اشتباه

گزینه های بازشناسی	قطعه اول	قطعه دوم	قطعه سوم	زیر-کلمات منطبق	فیلتر GUD (فقط بالا)	فیلتر GNUD (۳ حرکت در بالا)
۱				بط - یط	-	
۲				بق - یق	-	
۳				یو – بو – پو – نو – تو - ئو	نو – تو – ئو	-
۴				-		
۵				یم – نم	نم	-
۶				بطه	-	
۷				نکه	نکه	-
۸				بقه	-	
۹				-		
۱۰				-		
۱۱				نظر	نظر	نظر

فصل ۵

نتایج شبیه‌سازی و تحلیل نتایج

فصل ۵- نتایج شبیه‌سازی و تحلیل نتایج

۵-۱- نتایج شبیه‌سازی

برای شبیه‌سازی الگوریتم‌های اشاره شده در فصل ۴ از نرم افزار MATLAB و در مرحله شناسایی حروف از جعبه ابزار مدل مخفی مارکوف دانشگاه MIT [۲۹] استفاده کردیم. مزیت این جعبه ابزار نسبت به مدل مخفی مارکوف در جعبه ابزار آمار نرم افزار MATLAB در تخمین احتمالات اولیه و قابلیت خوشه‌بندی ویژگی‌های پیوسته توسط مدل مخلوط گوسی همراه با نرم افزار است. در روش پیشنهادی ویژگی استخراج شده از حروف، گسسته و یک بعدی است و نیازی به خوشه‌بندی آن نیست ولی ممکن است در کارهای آینده از ویژگی دیگری استفاده شود که نیاز به خوشه‌بندی بردار ویژگی برای به دست آوردن بردار مشاهده باشد. سخت افزار مورد استفاده دارای مشخصات 3 GB RAM DDR III و CPU Core i3 2.58 GHz است.

در این تحقیق از پایگاه‌های داده حروف و زیر-کلمات برخط دانشگاه تربیت مدرس استفاده شده است که پایگاه داده زیر-کلمات برخط تربیت مدرس شامل ۱۰۰۱ زیر-کلمه متداول زبان فارسی است که از آرشیو الکترونیکی روزنامه همشهری استخراج شده‌اند. نمونه‌های این پایگاه داده توسط ۱۲۴ نفر نوشته شده است که هر کدام حدود ۱۰۰ زیر-کلمه را نوشته‌اند. برای هر زیر-کلمه به طور متوسط حدود ۱۱ نمونه جمع‌آوری شده است و تعداد کل نمونه‌ها بیش‌تر از ۱۱ هزار است. کوتاه‌ترین نمونه‌ها همان حروف جدا هستند که یک حرف دارند و بلندترین زیر-کلمات دارای ۷ حرف هستند. برای جمع‌آوری داده از صفحه و قلم Wacom Graphire استفاده شده است که دارای اطلاعات فشار و زاویه نیست و تنها محدودیتی که برای نویسنده‌ها گذاشته شده است، این است که ابتدا بدنه اصلی زیر-کلمه و سپس علامت‌ها نوشته شوند.[۴و۸]

از پایگاه داده حروف برخط تربیت مدرس برای افزایش تعداد نمونه‌های حروف جدا استفاده شده است. این پایگاه داده شامل حروف جدای فارسی است که شامل حدود ۴ هزار نمونه است که توسط ۱۲۰ نفر نوشته شده است. محدودیت نوشتن در این پایگاه داده مربوط به شکل علامت‌هاست.[۸]

۵-۱-۱- داده‌های مصنوعی

نمونه‌های مصنوعی برای گروه بدنه حروف دارای کمتر از ۱۰۰ نمونه آموزشی تولید شده است تا مدل مخفی مارکوف به خوبی آموزش ببینند. افزودن تعداد زیاد داده‌های مصنوعی باعث یادگیری بیش از حد طبقه‌بند مدل مخفی مارکوف با داده‌های آموزشی می‌شود به همین دلیل تعداد نمونه‌های مصنوعی به اندازه‌ای که کل نمونه‌های آموزشی به ۱۰۰ برسند در نظر گرفته شده است. در جدول ۵-۱ تعداد گروه‌هایی که نمونه‌های آموزشی کمی داشته‌اند و نسبت کل نمونه‌های مصنوعی در هر دسته به کل نمونه‌های آموزشی آمده است.

جدول ۵-۱: تعداد گروه‌ها و نسبت نمونه‌های مصنوعی به کل نمونه‌ها برای هر دسته از حروف

دسته	جدا	اول	وسط	آخر
تعداد کل گروه‌ها	۲۲	۱۸	۱۷	۲۴
تعداد گروه‌ها با کمتر از ۱۰۰ نمونه آموزشی	۱۲	۶	۵	۳
درصد کل نمونه‌های مصنوعی به کل نمونه‌های آموزشی	۱۰/۶	۶/۲	۱/۳	۹/۲

۵-۱-۲- الگوریتم قطعه‌بندی

نتایج بخش ۴-۷ در جدول ۵-۲ آورده شده است، در این جدول، درصدها از نسبت تعداد نقاط قطعه‌بندی به دست آمده به تعداد نقاط قطعه‌بندی صحیح به دست آمده‌اند.

جدول ۵-۲: نتایج قطعه‌بندی

نتایج قطعه‌بندی		صحیح (درصد)	اضافی (درصد)		زمان (ثانیه)
			در ناحیه قطعه‌بندی	روی حرف	
فقط زاویه با محور افقی		۹۵/۵۵	۲۴۱۵/۴۱	۳۲۰۷/۹۹	۳۸/۶۷
حذف حلقه		۹۴/۸۷	۲۲۹۷/۸۶	۲۶۷۵/۳۴	۹۵/۴۳
ترکیب نواحی نزدیک به هم		۸۶/۰۸	۱/۹۸	۸۰/۲۳	۹۸/۵۶
حذف نقاط خیلی نزدیک به ابتدا یا انتها		۸۵/۷۱	۱/۹۴	۶۰/۹۰	۹۴/۵۴
حذف نقاط خارج از محدوده عمودی وجود نقطه قطعه‌بندی		۸۵/۴۷	۱/۹۳	۵۹/۶۷	۹۲/۴۴
تعداد کل زیر-کلمات	۱۰۸۹۴	زمان متوسط قطعه‌بندی زیر-کلمه برای کل الگوریتم			۰/۰۰۸
تعداد کل نقاط قطعه-بندی صحیح	۲۱۳۵۴				
تعداد زیر-کلماتی که تمام نقاط قطعه‌بندی برای آن‌ها پیدا شده است.					۷۷۳۷
تعداد زیر-کلماتی که تمام نقاط قطعه‌بندی شده پیدا شده در ناحیه قطعه‌بندی است.					۲۶۸۴

در سطر اول نتایج حاصل از الگوریتم قطعه بندی با در نظر گرفتن معیار زاویه بردار متصل کننده نقاط متوالی مطابق بخش ۴-۷ آورده شده است. همانطور که ملاحظه می گردد ۹۵/۵۵٪ از نقاط قطعه بندی در پایگاه داده به درستی تشخیص داده می شوند، گرچه نقاط اضافی زیادی ۲۴۱۵/۴۱٪ و ۳۲۰۷/۹۹٪ به ترتیب در ناحیه قطعه بندی و روی حروف به عنوان نقاط قطعه‌بندی بدست آمده‌اند. با اعمال الگوریتم های حذف حلقه، ترکیب نواحی قطعه بندی نزدیک و حذف نقاط نزدیک ابتدا و انتها در زیر-کلمه درصد نقاط اضافی در قطعه بندی به ۶۰٪ کاهش می یابد. گرچه با اعمال این روشها درصد نقاط درست تشخیص داده شده برای قطعه‌بندی نیز به حدود ۸۵٪ افت می کند.

روش پیشنهادی برای بیش از ۷۱٪ از زیر-کلمات، تمام نقاط قطعه‌بندی آن‌ها را استخراج کرده است و برای حدود ۲۵٪ زیر-کلمات، تمام نقاط قطعه‌بندی استخراج شده صحیح (در ناحیه قطعه‌بندی) بوده‌اند.

۵-۱-۳- تشخیص گروه حرف

نتایج آموزش و تست گروه حروف در جدول ۵-۳ آمده است.

جدول ۵-۳: درصد تشخیص گروه حروف

دسته	آموزش (٪)	تست (٪)	تست (مؤثر*)(٪)	تعداد نمونه‌های تست	زمان اجرا (ثانیه)			
					کل	کمترین	بیشترین	متوسط
جدا	۹۵/۸۷	۹۰/۱۷	۹۰/۳۹	۱۳۳۲	۱۶/۹۰۸	۰/۰۱۱	۰/۰۳۲	۰/۰۱۳
اول	۸۱/۸۸	۷۵/۲۷	۷۵/۸۵	۲۹۲۷	۳۲/۹۲۶	۰/۰۰۹	۰/۰۲۴	۰/۰۱۱
وسط	۸۷/۵۴	۸۴/۱۵	۸۹/۶۳	۳۰۰۹	۳۳/۳۳۶	۰/۰۰۹	۰/۱۲۳	۰/۰۱۱
آخر	۹۳/۸۳	۸۹/۵۹	۹۰/۶۳	۲۹۶۸	۴۴/۹۷۸	۰/۰۱۳	۰/۰۴۰	۰/۰۱۵

*: در صورتی که از خطای بین گروه‌های یکسان مثل شکل‌های مختلف حرف "مـ" صرف نظر شود.

این مقادیر با استفاده از پارامترهای گفته شده در بخش ۴-۸-۲ به دست آمده‌اند. نتایج جزئی شامل جدول تداخل[1] و مقدار مثبت صحیح[2] و مثبت کاذب[3] برای هر گروه در ضمیمه ب -آمده است.

در این جدول نتایج تشخیص حروف زیر کلمه به کمک الگوریتم مدل مخفی مارکوف آورده شده است. همانطور که می‌توان انتظار داشت نرخ تشخیص برای تمام حالت‌های حروف (جدا، اول، وسط و آخر) برای نمونه‌های آموزشی بالاتر از نمونه‌های آزمایشی است. همچنین نرخ تشخیص برای حروف جدا نسبت به حروف به هم چسبیده بالاتر است. عدم قطعیت در تعیین دقیق نقطه شروع و پایانی حروف داخل یک زیر کلمه تشخیص این حروف را دشوار می‌سازد.

جدول تداخل هر دسته از حروف می‌تواند نشان‌دهنده دقت انتخاب گروه‌های حروف باشد. مقدار سطر i و ستون j نشان‌دهنده تعداد داده‌هایی است که عضو گروه i هستند و در گروه j طبقه‌بندی شده‌اند. با توجه به نتایج جدول تداخل، ممکن است لازم باشد گروه‌های بدنه حروف مجدداً انتخاب شوند. به عنوان مثال، گروه ۱ و گروه ۱۱ از دسته حروف اول که مربوط به بدنه "بـ پـ تـ ثـ نـ یـ" و "کـ (بدون سرکش)، گـ (بدون سرکش)، لـ" هستند، شباهت زیادی دارند و باید بررسی شود که

[1] Confusion table
[2] True Positive (TP)
[3] False Positive (FP)

ترکیب این دو گروه و یا افزودن یک مرحله اضافی برای طبقه‌بندی بین آن‌ها مفید است یا خیر. به عنوان مثال طول قسمت رو به پایین این دو گروه متفاوت است که می‌تواند به عنوان یک ویژگی برای مقایسه بین این دو گروه استفاده شود.

۵-۱-۴- تشخیص زیر-کلمه

نتایج تشخیص زیر-کلمه در جدول ۵-۴ آمده است. سطر اول این جدول، نتیجه تشخیص گروه حروف زیر-کلمه که حروف آن با استفاده از Ground Truth جدا شده است را نشان می‌دهد. چون خطای الگوریتم قطعه‌بندی در این مورد وجود ندارد، تشخیص با درصد بالایی انجام گرفته است. در تشخیص زیر-کلمات حتی اگر یکی از حروف یک زیر-کلمه اشتباه تشخیص داده شده باشد، تشخیص کل زیر-کلمه اشتباه فرض شده می‌شود. در صورتی که از روش پیشنهادی برای قطعه‌بندی استفاده شود، نرخ تشخیص زیر-کلمه به شدت کاهش می‌یابد اما استفاده از فرهنگ لغت متنی و دسته‌بندی زیر-کلمات آن بر اساس وجود علامت و تعداد حرکت‌ها در بالا و پایین بدنه زیر-کلمه، تعدادی از نتایج اشتباه حذف می‌شوند و نرخ تشخیص بهبود می‌یابد. همچنین سطرهای دوم و سوم جدول نشان می‌دهد که در صورت پیدا کردن تمام نقاط قطعه‌بندی، نتیجه تشخیص زیر-کلمه بهبود زیادی می‌یابد.

جدول ۵-۴: نتایج شناسایی زیر-کلمه (درصد)

مجموعه مورد شناسایی	گزینه اول	۲ گزینه اول	۵ گزینه اول	۱۰ گزینه اول	وازدگی[1]
از قبل قطعه‌بندی شده	۵۶/۴۸	۶۵/۲۸	۸۱/۶۹	۸۵/۷۸	۳/۸۸
قطعه‌بندی شده با روش پیشنهادی	۲۹/۲۸	۳۹/۸۲	۵۹/۰۰	۷۰/۳۹	۲/۳۷
قطعه‌بندی شده با روش پیشنهادی*	۴۲/۱۰	۵۹/۵۸	۷۷/۶۷	۸۲/۳۶	۱۴/۲۸
قطعه‌بندی شده با روش پیشنهادی**	۵۵/۳۷	۶۵/۶۱	۸۳/۵۳	۸۷/۱۱	۱۰/۵۸
بعد از فیلتر کردن با GUD	۳۵/۱۴	۴۹/۵۸	۶۶/۶۷	۷۳/۷۴	۳/۸۰
بعد از فیلتر کردن با GNUD	۳۷/۸۵	۵۲/۵۸	۶۸/۶۹	۷۳/۳۴	۴/۸۵

*: فقط برای زیر-کلماتی که تمام نقاط قطعه‌بندی شده برای آن‌ها پیدا شده است.

**: فقط برای زیر-کلماتی که تمام نقاط قطعه‌بندی شده پیدا شده در ناحیه قطعه‌بندی است.

زمان اجرای برنامه کامل به تعداد گزینه‌های انتخاب شده بستگی دارد که با ۵۰ گزینه بازشناسی برای مراحل تشخیص گروه حروف، فیلتر کردن با گروه حروف زیر-کلمات فرهنگ لغت و فیلتر کردن با وجود علامت در بالا و پایین بدنه و ۱۰ گزینه برای فیلتر کردن با تعداد و محل حرکت‌ها (گزینه‌های نهایی) برای هر زیر-کلمه به طور متوسط ۰/۸۱ ثانیه است.

[1] Rejection

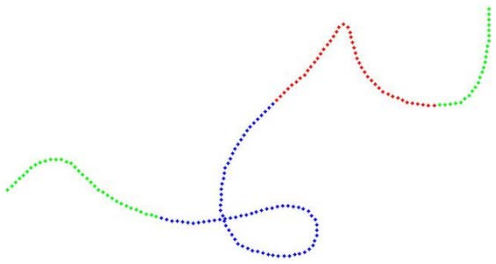
۵-۲- مقایسه و تحلیل نتایج

در این بخش نتایج روش پیشنهادی با کارهای مشابه که در فصل ۲ بیان شد، مقایسه می‌شود. البته چون در بیشتر این کارها، پایگاه داده مورد استفاده متفاوت است نتایج قابل مقایسه نیستند و صرفاً برای اطلاع آورده شده‌اند.

۵-۲-۱- مقایسه نتایج قطعه‌بندی

نتیجه روش قطعه‌بندی کارهای ذکر شده در فصل مرور کارهای گذشته و روش پیشنهادی در جدول ۵-۵ آورده شده است.

در نوشتن زیر-کلمات پایگاه داده استفاده شده در این تحقیق از صفحه رقومی کننده بدون صفحه نمایش استفاده شده است، به همین دلیل نسبت به نوشته‌های عادی روی کاغذ ناخواناتر هستند و خط زمینه هم در آن‌ها رعایت نشده است. ضمن این که در گذاشتن علامت‌ها بعد از نوشتن بدنه اصلی زیر-کلمه امکان جابجایی علامت‌ها وجود دارد که تخصیص علامت‌ها به حروف زیر-کلمه را با مشکل مواجه می‌کند. با مشاهده نتایج الگوریتم قطعه‌بندی مشاهده می‌شود که بیشتر خطاها مربوط به حروفی است که از روی خط زمینه نوشته نشده‌اند به عنوان مثال حرف "ج" در وسط زیر-کلمه یا دندانه‌های متوالی که در حین نوشتن به سمت بالا یا پایین متمایل شده‌اند. شکل ۵-۱ یک نمونه از پایگاه داده این تحقیق را نشان می‌دهد که به دلیل نوشته نشدن روی خط زمینه، فرایند قطعه‌بندی و در نتیجه تشخیص زیر-کلمه را با مشکل مواجه می‌کند.



شکل ۵-۱: یک نمونه دشوار برای قطعه‌بندی به حروف از پایگاه داده زیر-کلمات برخط دانشگاه تربیت مدرس

جدول ۵-۵: مقایسه نتایج قطعه‌بندی

مرجع	قطعه‌بندی صحیح	قطعه‌بندی اضافی	پایگاه داده
[۹]	۱۰۰	۵۰ تا ۶۰	۱۵۰ کلمه عربی
[۱۰]	۹۶/۲	ذکر نشده	کلمات عربی شامل ۳۲۰۰ حرف نوشته شده توسط ۲۰ نفر
روش پیشنهادی	۸۵/۴۷	۶۰/۱	زیر-کلمات برخط دانشگاه تربیت مدرس

۵-۲-۲- مقایسه تشخیص حروف

بررسی نتیجه کارهای مشابه در زمینه تشخیص حروف در جدول ۵-۶ آمده است. در زمینه تشخیص حروف، فقط روی حروف جدا کار شده است، به همین دلیل فقط نتیجه تشخیص حروف جدا ذکر شده است.

در این تحقیق علاوه بر حروف پایگاه داده مربوط به حروف جدای فارسی، حروف جدای پایگاه داده زیر-کلمات فارسی هم برای آموزش و تست سیستم استفاده شده است. نتایج جدول ۵-۶ نشان می‌دهد که استفاده از ویژگی‌های پیچیده‌تر می‌تواند تشخیص گروه حرف را بهبود دهد.

جدول ۵-۶: مقایسه تشخیص حروف

مرجع	نتیجه	روش	پایگاه داده
[۳]	٪۹۴/۹	HMM	حروف برخط دانشگاه تربیت مدرس
[۱۲]	٪۹۵	SOM برای انتخاب ویژگی و شبکه عصبی برای طبقه‌بندی	۳۴۰۰ حرف جدای عربی نوشته شده توسط ۲۵ نفر
[۱۳]	٪۹۰/۲۷ ٪۳ وازدگی	FLVQ	حروف برخط دانشگاه تربیت مدرس
[۱۴]	٪۹۳/۳	NN برای تشخیص بدنه و شبکه عصبی برای تشخیص علامت	حروف برخط دانشگاه تربیت مدرس
[۱۵]	٪۹۰/۸۲ تشخیص گروه حرف ٪۸۸/۸۷ تشخیص حرف	HMM	حروف برخط دانشگاه تربیت مدرس
[۱۶]	٪۹۴ برای تشخیص گروه حرف	درخت تصمیم گیری	حروف برخط دانشگاه تربیت مدرس
روش پیشنهادی	۹۰/۱۷ تشخیص گروه حرف	HMM	حروف و زیر-کلمات برخط دانشگاه تربیت مدرس

۵-۲-۳- مقایسه تشخیص زیر-کلمه

در این بخش فقط کارهایی که در زمینه تشخیص کلمه/زیر-کلمه بر مبنای تشخیص حروف سازنده آن انجام شده است، آورده شده است. این نتایج در جدول ۵-۷ نشان داده شده است. هیچ کاری در زمینه تشخیص دستنوشته برخط عربی/فارسی با قطعه‌بندی زیر-کلمات به حروف پیدا نکردیم لذا این کار را با کارهای نزدیک‌تر که بر مبنای جستجوی مدل حروف در زیر-کلمه هستند مقایسه کرده‌ایم.

تنوع زیاد نویسنده‌ها و دستخط‌ها در پایگاه داده استفاده شده در این تحقیق، باعث سخت شدن پیشنهاد یک روش مبتنی بر قواعد ساده برای قطعه‌بندی زیر-کلمه به حروف شده است. خطای الگوریتم قطعه‌بندی باعث کاهش چشمگیر تشخیص زیر-کلمه می‌شود. در [۱۷]، [۱۹] و [۲۱] برای

شناسایی حروف زیر-کلمه از قطعه‌بندی استفاده نشده است بلکه تشخیص حروف از ابتدای زیر-کلمه بر اساس مدل حروف صورت گرفته است. نتایج جدول ۵-۴ نشان می‌دهد که قطعه‌بندی درست تا چه حد در تشخیص زیر-کلمه مؤثر است.

جدول ۵-۷: مقایسه تشخیص زیر-کلمه

مرجع	نتیجه	روش	پایگاه داده
[۱۷]	٪۸۹	شبکه عصبی سلسله مراتبی	کلمه "عزّالدّین" که ۱۰۰ بار توسط یک نفر نوشته شده است.
[۱۹]	٪۹۸/۴۹ فرهنگ لغت ۵۰۰۰ کلمه‌ای، مستقل از نویسنده	HMM	۶۲۲۰ زیر-کلمه از ۸۰۰ کلمه عربی نوشته شده توسط ۱۰ نفر
[۲۱]	٪۷۸ بدون فرهنگ لغت و ٪۹۶ با فرهنگ لغت	FEM	۱۲۵۰ کلمه از کتاب‌های اول دبستان نوشته شده توسط ۱۰ بزرگ‌سال و ۱۰ دانش آموز ابتدایی
روش پیشنهادی	٪۳۷/۸۵ برای گزینه اول، ٪۷۳/۳۴ برای ۱۰ گزینه اول و ٪۴/۸۵ وازدگی	HMM	زیر-کلمات برخط دانشگاه تربیت مدرس

فصل ۶

نتیجه‌گیری و کارهای آینده

فصل ۶- نتیجه‌گیری و کارهای آینده

۶-۱- نتیجه‌گیری

با گسترش استفاده از تلفن همراه دارای صفحه لمسی و تبلت، تشخیص متن دستنوشته در آن‌ها اهمیت بیشتری پیدا کرده است. با حرکت قلم روی صفحه، حسگرهای[1] قرار گرفته روی صفحه نمایش تحریک می‌شوند و مختصات نقطه لمس شده را به خروجی صفحه لمسی ارسال می‌کنند که این اطلاعات برای پردازش در واحد بعدی ذخیره می‌شوند. اطلاعات حرکت قلم در هر لحظه در دسترس است به همین دلیل، داده تولید شده برخط نامیده می‌شود در حالی که تصویر متن دستنویس که به صورت یکجا و پس از پایان نوشتن در دسترس است، نوعی داده برون خط است.
نوشتن با قلم مخصوص بر روی صفحه لمسی، بسیار راحت‌تر و سریع‌تر از کار با صفحه کلید مجازی این وسایل است از این رو تشخیص دستنوشته برخط می‌تواند انقلابی در وارد کردن داده‌های متنی به وسایل دیجیتال ایجاد کند. در زمینه تشخیص دستنوشته برخط لاتین کارهای زیادی انجام شده است ولی کارهای کمی بر روی دستنوشته برخط فارسی انجام شده است. تشخیص دستنوشته فارسی به دلیل طبیعت پیوسته متن، وجود علامت‌های حروف و تغییر شکل حروف در موقعیت‌های مختلف پیچیدگی‌های بیشتری نسبت به دستنوشته لاتین دارد.
در تشخیص دستنوشته برخط می‌توان هر زیر-کلمه را به عنوان یک الگوی مستقل در نظر گرفت و اقدام به شناسایی آن کرد یا حروف زیر-کلمه را به عنوان الگوهای مورد شناسایی در نظر گرفت. در صورتی که از رویکرد دوم استفاده شود، لازم است ابتدا هر زیر-کلمه به حروف سازنده تجزیه شود که به این کار قطعه‌بندی گفته می‌شود. روش قطعه‌بندی دارای این مزیت است که می‌تواند هر زیر-کلمه‌ای را تشخیص دهد ولی در روش کلی‌نگر، تشخیص محدود به زیر-کلمات موجود در پایگاه داده است.
در این تحقیق، روشی برای تشخیص دستنوشته برخط فارسی مبتنی بر قطعه‌بندی پیشنهاد شده است. برای قطعه‌بندی زیر-کلمه به حروف از این واقعیت استفاده شده است که معمولاً در نقاط اتصال دو حرف یک زیر-کلمه، جهت دستنوشته از راست به چپ و زاویه آن با محور افقی کم است. در این روش، نقاطی که با زاویه بردار اتصال دهنده آن به نقطه بعد در حدود مشخصی باشد به عنوان کاندیدهای اولیه قطعه‌بندی مشخص می‌شوند. سپس با استفاده از قواعد ساده‌ای نظیر این که نقاط قطعه‌بندی نمی‌توانند خیلی به هم نزدیک باشند، نمی‌توانند در حلقه یا دندانه‌های حروف قرار داشته باشند و نمی‌توانند خیلی نزدیک به ابتدا یا انتهای نوشته باشند، نقاط اضافی حذف می‌شوند.

[1] Sensor

برای هر زیر-کلمه ممکن است تعدادی نقطه قطعه‌بندی اضافی تولید شده باشد. چون نمی‌دانیم کدام نقاط قطعه‌بندی اضافی هستند تمام ترکیب‌های ممکن از نظر وجود نقاط قطعه‌بندی را بررسی می‌کنیم و بهترین ترکیب را انتخاب می‌کنیم. بدین منظور به هر ترکیب امتیازی نسبت می‌دهیم و ترکیب‌های با بالاترین امتیاز را انتخاب می‌کنیم. امتیاز هر ترکیب را متوسط احتمال نرمالیزه شده تولید بردار مشاهده قطعات توسط مدل مخفی مارکوف در نظر می‌گیریم. بعد از به دست آوردن بهترین ترکیب گروه بدنه حروف، تعداد حرکت‌های زیر-کلمه در بالا و پایین آن را محاسبه می‌کنیم. با استفاده از فرهنگ لغات متنی با استفاده از فیلترهای وجود علامت در بالا و پایین زیر-کلمه و تعداد حرکت‌های بالا و پایین زیر-کلمه، گزینه‌های اشتباه حذف می‌شوند.

آزمایش‌ها بر روی پایگاه داده زیر-کلمات برخط دانشگاه تربیت مدرس انجام شده است. در تهیه این پایگاه داده، تنها محدودیت نویسنده‌ها نوشتن بدنه اصلی زیر-کلمه قبل از گذاشتن حرکت‌های علامت‌ها بوده است و به همین دلیل دارای تنوع زیاد دستخط است، نویسنده‌ها نوشتن روی خط زمینه را رعایت نکرده‌اند و در بعضی نمونه‌ها از شکل‌های غیر متداول حروف استفاده شده است. بدون خط زمینه، نوشته ممکن است شکل رو به بالا یا پایین پیدا کند که در این صورت واقعیتی که درباره نقاط قطعه‌بندی گفته شد نادرست می‌شود و باعث بالا رفتن خطای الگوریتم قطعه‌بندی می‌شود. برای کاهش خطای الگوریتم قطعه‌بندی ناچاریم نقاط قطعه‌بندی اضافی را بپذیریم که منجر به افزایش خطا و زمان محاسبات می‌شود. همچنین چون برای آموزش مدل هر حرف، فقط از نمونه‌های متداول حروف استفاده شده است، این مدل‌ها قادر به شناسایی نمونه‌های غیر متداول نیستند، که باعث افزایش خطای تشخیص گروه حروف می‌شود.

کار مشابهی در زمینه قطعه‌بندی زیر-کلمه به حروف بر روی پایگاه داده استفاده شده در این پایان نامه، موجود نیست اما مقایسه با کارهای انجام شده در زبان عربی نشان می‌دهد که تنوع کلمات/زیر-کلمات در پایگاه داده و طرز نوشتن نویسنده‌ها تأثیر زیادی در نتیجه دارد.

در زمینه تشخیص حروف، فقط حروف جدا مقایسه شده‌اند چون کاری بر روی شکل‌های دیگر حروف انجام نشده است. مقایسه با کارهای گذشته نشان می‌دهد استفاده از ویژگی‌های پیچیده‌تر و انتخاب طول متغیر بردار ویژگی در بهبود تشخیص حروف مؤثر است.

در زمینه تشخیص زیر-کلمه قطعه‌بندی شده هیچ کار مشابهی حتی در زبان عربی پیدا نشد اما نتیجه روش پیشنهادی به همراه نتیجه کارهایی که مبتنی بر تشخیص زیر-کلمه با استفاده از مدل حرف هستند آورده شده است. برای مقایسه کارایی دو روش ذکر شده لازم است تشخیص زیر-کلمه بر مبنای مدل حروف بر روی پایگاه داده این تحقیق پیاده‌سازی شود.

به دلایل زیر رسیدن به نرخ تشخیص بالا با روش مبتنی بر قطعه‌بندی دشوار است:

- مشکل بودن نمونه های پایگاه داده برای پیدا کردن خط زمینه و نقاط قطعه بندی
- وجود تعدادی از حروف با شکل غیر متداول
- وجود تعدادی از زیر-کلمات که بدنه اصلی در بیش از یک حرکت نوشته می شود. (کلا)

۶-۲- کارهای آینده

با توجه به نتایج ذکر شده در فصل قبل برای بهبود و تکمیل این تحقیق پیشنهاد می‌شود موارد زیر در آینده انجام شود:

۱- تهیه پایگاه داده جدید

تنها پایگاه داده موجود برای زیر-کلمات دستنوشته برخط فارسی، پایگاه داده زیر-کلمات برخط دانشگاه تربیت مدرس است. متأسفانه در این پایگاه داده تعداد نمونه کافی از هر حرف در جایگاه‌های مختلف زیر-کلمه برای آموزش گروه‌های بدنه حروف وجود ندارد؛ لذا لازم است پایگاه داده‌ای از زیر-کلمات فارسی که تعداد تکرار قابل قبولی از هر حرف داشته باشد تهیه شود. علاوه بر این بهتر است برای تهیه پایگاه داده جدید از یک وسیله دارای صفحه نمایش استفاده شود تا خوانایی نمونه‌های دستنویس افزایش یابد و امکان نوشتن روی خط زمینه هم برای افزایش صحت قطعه‌بندی و نرخ تشخیص نهایی در تهیه این پایگاه داده لحاظ شود.

۲- انتخاب مستقل پارامترهای مدل مخفی مارکوف برای هر گروه

چون پیچیدگی و طول حروف مختلف متفاوت است بهتر است طول دنباله مشاهده برای هر حرف به صورت جداگانه انتخاب شود. با تغییر طول دنباله مشاهده، لازم است تعداد حالت‌های مخفی هم تنظیم شود تا بهترین نتیجه ممکن حاصل شود.

۳- پیشنهاد روشی برای مقایسه حرکت‌های زیر-کلمات یک گزینه بازشناسی

نتیجه نهایی سیستم پیشنهادی، حداکثر ۱۰ گزینه بازشناسی است که ممکن است هر کدام شامل بیش از یک زیر-کلمه باشند. زیر-کلمات هر گزینه از نظر شکل بدنه و تعداد حرکت‌ها در پایین و بالای بدنه مشابه هستند و فقط در نوع حرکت‌ها متفاوتند. برای تکمیل سیستم لازم است روشی برای مقایسه حرکت‌های زیر-کلمه پیشنهاد شود.

۴- استفاده از تطبیق رشته[1]

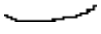
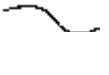
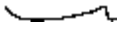
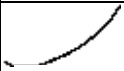
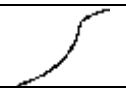
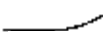
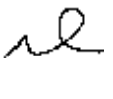
استفاده از یکی از روش‌های تطبیق رشته مثل حداقل فاصله ویرایشی[2] به عنوان یک مرحله پس‌پردازش، می‌تواند تأثیر بسزایی در کاهش خطای ناشی از خطای قطعه‌بندی داشته باشد[۲]. برای رفع مشکل نقاط قطعه‌بندی اضافی، ترکیب حالت‌های مختلف وجود نقاط قطعه‌بندی پیدا شده را امتحان کردیم اما برای نقاط قطعه‌بندی از دست رفته، هیچ راهی وجود ندارد ولی با استفاده از تطبیق رشته می‌توان حالت‌های احتمالی از دست رفتن نقاط قطعه‌بندی در موارد پرتکرار را هم در نظر گرفت. همچنین در صورت امکان شناسایی علامت‌ها اگر این مرحله را بعد از تشخیص گروه قطعات استفاده کنیم، می‌توانیم یک سیستم مستقل از فرهنگ لغت متنی داشته باشیم.

---

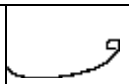
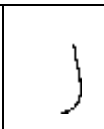
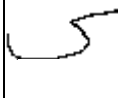
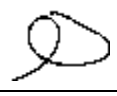
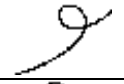
[1] String Matching
[2] Minimum Edit Distance

ضمیمه أ - گروه‌های بدنه حروف

جدول أ-۱: گروه‌های بدنه حروف در ۴ دسته حروف جدا، ابتدایی، وسط و انتهایی به همراه یک نمونه تصویر از پایگاه داده

شماره	جدا		شماره	ابتدا		شماره	وسط		شماره	انتها	
	حروف	مثال		حروف	مثال		حروف	مثال		حروف	مثال
۱	آ ا أ		۲۳	ب پ ت ث ن ی		۴۱	ب پ ت ث ن ی (دندانه)		۵۸	ـا ـأ	
۲	ب پ ت ث		۲۴	ج چ ح خ		۴۲	ب پ ت ث ن ی (انتها به بالا)		۵۹	ـب ـپ ـت ـث	
۳	ج چ ح خ		۲۵	س ش (با دندانه)		۴۳	ب پ ت ث ن ی (انتها به پایین)		۶۰	ـج ـچ ـح ـخ	
۴	د ذ		۲۶	س ش (کشیده)		۴۴	ج چ ح خ		۶۱	ـد ـذ	
۵	ر ز ژ		۲۷	ص ض		۴۵	س ش		۶۲	ـر ـز ـژ	
۶	س ش		۲۸	ط ظ (بدون دسته)		۴۶	س ش (کشیده)		۶۳	ـس ـش	
۷	ص ض		۲۹	ط ظ (با دسته- شروع از حلقه)		۴۷	ص ض		۶۴	ـس ـش (کشیده)	
۸	ط ظ (بدون دسته)		۳۰	ط ظ (با دسته- شروع از دسته)		۴۸	ط ظ (بدون دسته)		۶۵	ـص ـض	
۹	ط ظ (با دسته)		۳۱	ع غ		۴۹	ط ظ (با دسته)		۶۶	ـط ـظ (بدون دسته)	
۱۰	ع غ		۳۲	ف ق		۵۰	ع غ (حلقه پادساعت‌گرد)		۶۷	ـط ـظ (با دسته)	

ادامه جدول أ-۱

۱۱	ف		۳۳	کـ گـ لـ (بدون سرکش)		۵۱	ـعـ ـغـ (حلقه ساعت‌گرد)		۶۸	ـع ـغ (حلقه پادساعت گرد)	
۱۲	ق		۳۴	کـ گـ (بدون سرکش- قبل از ـا و ـل و ـک/ـگ)		۵۲	ـفـ ـقـ		۶۹	ـع ـغ (حلقه ساعت‌گرد)	
۱۳	ک گ (بدون سرکش)		۳۵	کـ گـ (بدون سرکش- کج)		۵۳	ـکـ ـگـ ـلـ*		۷۰	ـف	
۱۴	ک گ (با سرکش)		۳۶	کـ گـ (با سرکش)		۵۴	ـکـ ـگـ (قبل از ـا و ـل و ـک/ـگ)		۷۱	ـق	
۱۵	ل		۳۷	مـ (پادساعت گرد)		۵۵	ـمـ		۷۲	ـک ـگ	
۱۶	م (حلقه از بالا)		۳۸	مـ (ساعتگرد- شروع از بالا)		۵۶	ـهـ (ـهـ)		۷۳	ـل	
۱۷	م (حلقه از پایین)		۳۹	مـ (ساعتگرد- شروع از پایین)		۵۷	ـهـ (ـهـ)		۷۴	ـم (حلقه بالا)	
۱۸	ن		۴۰	هـ					۷۵	ـم (حلقه پایین)	
۱۹	و								۷۶	ـم (حلقه کوچک)	
۲۰	ه								۷۷	ـن	
۲۱	ی								۷۸	ـو	
۲۲	ء								۷۹	ـه (به)	
۲۳									۸۰	ـه	
۲۴									۸۱	ـی	

ضمیمه ب - نتایج جزئی

جدول ب-۱: جدول تداخل حروف جدا (تست)

FP (%)	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲	۱۳	۱۴	۱۵	۱۶	۱۷	۱۸	۱۹	۲۰	۲۱	۲۲	TP (%)
۱	۱۰۹	۰	۰	۰	۱	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۱	۰	۰	۰	۰	۰	۰	۹۱/۶۰
۲	۰	۱۳۵	۰	۱	۱	۰	۰	۱	۰	۰	۳	۰	۲۰	۰	۰	۰	۰	۱	۰	۰	۰	۰	۸۶/۵۴
۳	۰	۰	۱۳۴	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۱	۰	۱	۹۵/۷۱
۴	۱	۰	۰	۷۰	۰	۰	۰	۰	۰	۰	۰	۰	۱	۰	۰	۰	۰	۰	۱	۱	۰	۰	۸۸/۶۱
۵	۰	۰	۰	۰	۱۱۵	۰	۰	۰	۰	۰	۰	۰	۳	۰	۰	۰	۰	۰	۰	۰	۰	۱	۹۶/۶۴
۶	۰	۰	۰	۰	۰	۷۰	۲	۰	۱	۰	۰	۱	۰	۰	۰	۰	۰	۰	۰	۳	۰	۰	۹۰/۹۱
۷	۰	۰	۰	۰	۰	۲	۷۴	۰	۰	۰	۱	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۹۶/۱۰
۸	۰	۰	۰	۰	۰	۰	۰	۶۲	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۱۰۰
۹	۰	۰	۰	۰	۰	۱	۰	۰	۱۴	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۸۷/۵۰
۱۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۷۴	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۳	۹۶/۱۰
۱۱	۰	۰	۰	۰	۰	۰	۰	۱	۰	۰	۲۹	۱	۰	۰	۰	۰	۰	۰	۱	۰	۰	۰	۷۶/۳۲
۱۲	۰	۰	۰	۰	۰	۱	۰	۰	۰	۰	۲	۳۱	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۸۸/۵۷
۱۳	۰	۰	۰	۰	۳	۰	۰	۰	۰	۰	۱	۰	۵۱	۰	۰	۰	۰	۱	۰	۰	۰	۰	۶۸/۹۲
۱۴	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۲	۰	۰	۰	۰	۰	۰	۰	۰	۱۰۰
۱۵	۰	۰	۰	۰	۰	۰	۰	۰	۱	۰	۰	۰	۰	۰	۳۶	۰	۰	۱	۰	۰	۰	۰	۹۲/۳۱
۱۶	۱	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۱	۳۲	۱	۰	۰	۰	۰	۰	۹۱/۴۳
۱۷	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۲	۱	۰	۰	۰	۰	۰	۳۳/۳۳
۱۸	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۲	۱	۰	۲	۰	۰	۲۳	۰	۲	۱	۰	۷۴/۱۹
۱۹	۰	۰	۰	۰	۰	۰	۰	۲	۰	۰	۱	۰	۰	۰	۰	۰	۰	۰	۲۶	۰	۰	۰	۹۲/۳۱
۲۰	۰	۰	۰	۰	۰	۱	۰	۰	۱	۰	۰	۰	۰	۰	۲	۰	۰	۰	۰	۳۰	۰	۲	۸۱/۰۸
۲۱	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۱	۰	۰	۳۷	۰	۹۴/۸۷
۲۲	۰	۰	۱	۱	۱	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۱	۰	۳۶	۹۴/۷۴
TP (%)	۱/۶۸	۱۴/۷۴	۰	۲/۵۳	۵/۰۴	۷/۷۹	۰	۶/۴۵	۲۵	۵/۱۹	۲۳/۶۸	۱۴/۲۹	۳۵/۶۸	۵۰	۲۸/۲۱	۲۰	۱۰۰	۱۹/۳۵	۵/۱۳	۱۳/۵۱	۱۰/۲۶	۲۱/۰۵	۹۰/۱۷

جدول ب-۲: جدول تداخل حروف اول (تست)

FP (%)	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲	۱۳	۱۴	۱۵	۱۶	۱۷	۱۸	TP (%)
۱	۵۱۶	۲	۱	۷۲	۰	۰	۰	۰	۰	۰	۳۲۷	۰	۲۰	۰	۰	۳	۶	۰	۵۴/۴۹
۲	۰	۳۱۵	۰	۰	۰	۱۰	۰	۰	۰	۵	۰	۱۲	۲	۰	۱	۱	۱۶	۰	۸۷/۰۲
۳	۱	۰	۲۱۴	۵	۱	۰	۱	۱	۰	۰	۰	۰	۰	۰	۱	۵	۰	۱	۹۳/۴۵
۴	۲۰	۰	۲	۸۹	۰	۰	۰	۱	۰	۰	۰	۱	۰	۰	۱	۰	۰	۰	۷۸/۷۶
۵	۰	۰	۰	۰	۸۸	۳	۰	۱	۰	۳	۰	۲	۰	۰	۱	۰	۱	۰	۹۰/۷۲
۶	۰	۱	۰	۰	۱	۱۹	۰	۰	۰	۲	۰	۴	۰	۰	۱	۰	۲	۰	۶۵/۵۲
۷	۰	۰	۰	۰	۱	۰	۱۰	۰	۰	۰	۰	۰	۰	۰	۱	۰	۰	۰	۹۰/۹۱
۸	۰	۰	۰	۰	۰	۰	۰	۴	۰	۰	۱	۰	۱	۰	۰	۰	۰	۰	۸۰
۹	۰	۰	۰	۰	۰	۰	۰	۱	۱۵۱	۰	۰	۰	۱	۰	۱	۰	۰	۰	۹۷/۴۲
۱۰	۱	۳	۰	۰	۰	۱۰	۰	۰	۰	۱۴۵	۲	۱	۰	۰	۰	۰	۳۰	۰	۷۵/۵۲
۱۱	۷۱	۱	۰	۰	۰	۰	۰	۰	۰	۱	۲۲۰	۰	۶	۰	۰	۰	۰	۰	۷۳/۵۸
۱۲	۰	۰	۰	۰	۰	۴	۰	۰	۰	۳	۰	۴	۰	۰	۰	۰	۱	۰	۳۳/۳۳
۱۳	۰	۱	۰	۰	۰	۰	۰	۰	۰	۰	۱	۱	۴	۰	۰	۰	۱	۰	۵۰
۱۴	۰	۰	۰	۰	۰	۰	۰	۰	۱	۰	۰	۰	۰	۴	۰	۰	۰	۰	۸۰
۱۵	۲	۰	۰	۲	۰	۰	۱	۰	۲	۰	۰	۰	۰	۰	۲۲۱	۰	۲	۰	۹۶/۰۹
۱۶	۰	۰	۰	۰	۰	۰	۰	۱	۰	۰	۰	۰	۰	۰	۰	۵۱	۰	۳	۹۲/۷۳
۱۷	۰	۰	۰	۰	۰	۳	۰	۰	۰	۱۵	۰	۳	۰	۱	۰	۰	۳۰	۰	۵۴/۵۵
۱۸	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۴	۰	۱۱۸	۹۵/۹۳
۷۵/۲۷	۱۰/۰۳	۳/۰۴	۱/۳۱	۶۹/۹۱	۳/۰۹	۱۰۳/۴۵	۱۸/۱۸	۱۲۰	۱/۹۴	۱۴/۵۸	۱۱۰/۷۰	۲۰۰	۳۶۲/۵۰	۴۰	۰/۸۷	۲۳/۶۴	۱۰۷/۲۷	۳/۲۵	

جدول ب-۳: جدول تداخل حروف وسط (تست)

FP (%)	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲	۱۳	۱۴	۱۵	۱۶	۱۷	TP (%)
۱	۹۵۴	۵۲	۷۵	۰	۱	۳	۰	۱	۰	۳	۱	۱۵	۹۵	۱	۱	۰	۰	۷۹/۳۷
۲	۱۱	۸۸	۲	۰	۰	۰	۰	۰	۰	۱	۰	۰	۰	۰	۱	۰	۱	۸۴/۶۲
۳	۱۶	۱	۴۳	۰	۱	۰	۰	۰	۰	۰	۰	۰	۰	۰	۱	۰	۰	۶۹/۳۵
۴	۰	۰	۰	۱۵۳	۰	۰	۰	۰	۰	۰	۱	۱	۰	۰	۱	۰	۰	۹۸/۱۱
۵	۰	۲	۰	۰	۱۹۴	۰	۱	۰	۰	۰	۰	۰	۰	۰	۰	۰	۱	۹۷/۹۸
۶	۰	۱	۰	۰	۱	۶۴	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۹۶/۹۷
۷	۰	۰	۰	۰	۰	۰	۶۵	۲	۲	۰	۰	۲	۰	۰	۰	۰	۰	۹۱/۵۵
۸	۰	۰	۰	۰	۰	۰	۰	۲۰	۰	۰	۰	۱۵	۱	۳	۰	۰	۰	۵۱/۲۸
۹	۰	۰	۰	۰	۰	۰	۰	۰	۱۸	۰	۰	۰	۰	۰	۰	۱	۰	۹۴/۷۴
۱۰	۱	۰	۰	۱	۰	۰	۰	۰	۰	۸۹	۳	۰	۲	۰	۰	۰	۰	۹۲/۷۱
۱۱	۱	۰	۰	۰	۰	۰	۰	۴	۰	۲	۱۷	۱	۰	۱	۰	۰	۰	۶۵/۳۸
۱۲	۸	۰	۰	۰	۰	۰	۲	۳۸	۰	۰	۴	۱۳۷	۴	۱۶	۰	۰	۰	۶۵/۵۵
۱۳	۲۶	۰	۰	۰	۰	۰	۰	۰	۲	۱	۰	۲	۳۹۰	۱	۰	۰	۰	۹۲/۴۲
۱۴	۰	۰	۰	۰	۰	۰	۰	۷	۰	۰	۲	۱۱	۰	۱۰	۰	۰	۰	۳۳/۳۳
۱۵	۰	۱	۱	۶	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۱۶۵	۰	۵	۹۲/۷۰
۱۶	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۸۷	۰	۱۰۰
۱۷	۰	۰	۲	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۲	۰	۳۶	۸۷/۸۰
۸۴/۱۵	۵/۲۴	۵۵/۷۷	۱۲۹/۰۳	۴/۴	۱/۵۲	۴/۵۵	۴/۲۳	۱۳۳/۳۳	۲۱/۰۵	۷/۲۹	۴۲/۳۱	۲۲/۴۹	۲۴/۱۷	۷۳/۳۳	۳/۳۷	۱/۱۵	۱۷/۰۷	

جدول ب-۴: جدول تداخل حروف آخر (تست)

	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲	۱۳	۱۴	۱۵	۱۶	۱۷	۱۸	۱۹	۲۰	۲۱	۲۲	۲۳	۲۴	TP (%)
۱	۶۶۰	۰	۰	۱	۰	۰	۱	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۳	۴	۳	۹۸/۲۱
۲	۰	۱۸۶	۰	۲	۳	۰	۲	۰	۳	۰	۰	۰	۰	۲	۹	۰	۰	۰	۰	۱۱	۰	۰	۰	۰	۸۵/۳۲
۳	۰	۰	۱۵	۰	۰	۰	۰	۰	۰	۰	۱	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۹۳/۷۵
۴	۰	۰	۰	۲۶۴	۲	۰	۰	۰	۰	۰	۰	۰	۰	۱	۷	۱	۰	۰	۰	۰	۷	۶	۰	۰	۹۱/۶۷
۵	۰	۱	۰	۳۹	۳۷۲	۰	۰	۰	۰	۰	۰	۰	۰	۰	۳	۰	۰	۰	۰	۱	۱۹	۱۹	۰	۱	۸۱/۷۶
۶	۰	۱	۰	۰	۰	۴۷	۱	۱	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۵	۰	۰	۰	۳	۸۱/۰۳
۷	۰	۰	۰	۰	۰	۴	۶	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۶۰
۸	۰	۰	۰	۰	۰	۱	۰	۲	۰	۰	۰	۰	۱	۱	۰	۱	۰	۰	۰	۰	۰	۰	۰	۰	۳۳/۳۳
۹	۰	۰	۰	۰	۰	۰	۰	۰	۱۸	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۳	۰	۰	۰	۸۵/۷۱
۱۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۱	۰	۱	۰	۰	۰	۰	۱	۰	۰	۰	۰
۱۱	۰	۰	۱	۰	۰	۰	۰	۰	۰	۰	۲۱	۴	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۸۰/۷۷
۱۲	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۴	۵	۰	۰	۰	۰	۰	۰	۰	۰	۰	۱	۰	۰	۵۰
۱۳	۰	۱	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۹	۲	۱	۰	۰	۰	۰	۰	۱	۰	۰	۰	۶۴/۲۹
۱۴	۰	۰	۰	۰	۰	۱	۰	۰	۰	۰	۰	۰	۰	۲۷	۰	۰	۰	۰	۰	۲	۲	۰	۰	۰	۸۴/۳۸
۱۵	۰	۲	۰	۵	۱	۰	۰	۰	۱	۰	۰	۰	۰	۱	۱۹	۰	۰	۰	۰	۱	۰	۰	۰	۰	۶۳/۳۳
۱۶	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۸۱	۰	۰	۰	۴	۰	۰	۰	۰	۹۵/۲۹
۱۷	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۱	۰	۰	۰	۱	۳۰	۱	۱	۰	۰	۱	۰	۰	۸۵/۷۱
۱۸	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۲۱	۵	۰	۰	۰	۰	۰	۸۰/۷۷
۱۹	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۱	۸	۱۸	۰	۰	۰	۰	۰	۶۶/۶۷
۲۰	۰	۱	۰	۲	۰	۰	۰	۰	۰	۰	۰	۰	۰	۵	۱	۰	۰	۰	۰	۱۰۱	۰	۰	۰	۹	۸۴/۸۷
۲۱	۰	۰	۰	۹	۷	۰	۰	۰	۴	۱	۰	۰	۰	۱	۱	۰	۰	۰	۰	۰	۲۷۳	۰	۰	۰	۹۲/۲۳
۲۲	۰	۰	۰	۱۳	۹	۰	۰	۰	۰	۰	۱	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۲۰۶	۰	۰	۸۹/۵۷
۲۳	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۲	۶۱	۰	۹۶/۸۳
۲۴	۰	۰	۰	۰	۰	۷	۱	۰	۰	۰	۰	۰	۰	۱	۰	۰	۰	۰	۰	۲	۰	۰	۰	۲۱۷	۹۵/۱۸
FP (%)	۰	۲/۷۵	۱۲/۵۰	۲۴/۶۵	۴/۸۴	۳۲/۴۱	۵۰	۱۶/۶۷	۳۸/۱۰	۳۳/۳۳	۳۲/۰۸	۵۰	۷/۱۴	۴۶/۸۸	۷۷/۳۳	۴/۷۱	۲/۸۶	۳۴/۶۲	۳۲/۲۲	۳۱/۸۵	۱۱/۱۵	۱۳/۴۸	۷/۹۴	۷/۰۲	۸۹/۵۹

فهرست مراجع

[۱] شورای پژوهشی (OCR) کارگروه خط و زبان فارسی(۱۳۸۸)، پژوهشنامه نویسه‌خوان نوری (OCR) فارسی، دبیرخانه شورای عالی اطلاع رسانی، تهران.

[2] Seni G. (1995) “Large Vocabulary Recognition of On-Line Handwritten Cursive Words” PhD Thesis, University of New York at Buffalo, USA.

[۳] فرکی م. و پالهنگ م. (۱۳۸۹) "بازشناسی برخط حروف فارسی بر پایه مدل مخفی مارکوف" *مجله مهندسی برق دانشگاه تبریز*، شماره ۱، ج ۴۰: صص ۲۳-۳۴.

[۴] قدس و. و کبیر ا. (۱۳۹۰) "بررسی شیوه‌های متداول دستنوشته‌های برخط فارسی به منظور استفاده در بازشناسی آن‌ها" *مجله مهندسی برق دانشگاه تبریز*، شماره ۱، ج ۴۱: صص ۲۲-۳۲.

[5] AlHamada H.A. and AbuZitar R. (2010) “Development of an Efficient Neural-Based Segmentation Technique for Arabic Handwriting Recognition” *Journal of Pattern Recognition*, No.43, pp. 2773-2798.

[۶] رضوی س.م. و کبیر ا. (۱۳۸۴) "**روشی ساده برای بازشناسی زیر-کلمات فارسی**" *نشریه مهندسی برق و مهندسی کامپیوتر ایران*، سال ۳، شماره ۲، پاییز و زمستان ۱۳۸۴، صص ۶۳-۷۲.

[7] Faradji F. , Faez K. and Nosrati M. S. (2007) “Online Farsi Handwritten Words Recognition Using a Combination of 3 Cascaded RBF Neural Networks” International Conference on Intelligent and Advanced Systems, pp. 134-138 , Kuala Lumpur , Malaysia.

[۸] رضوی م. و کبیر ا. (۱۳۸۳) "**یک پایگاه داده برای دستنوشته‌های برخط فارسی**" ششمین کنفرانس سیستم‌های هوشمند، صص ۲۱۸-۲۲۵، کرمان.

[9] Daifallah Kh., Zarka N. and Jamous H. (2009) “Recognition-Based Segmentation Algorithm for On-Line Arabic Handwriting” 10th International Conference on Document Analysis and Recognition, pp. 886-890 , Barcelona , Spain

[10] Potrus, M.Y., Ngah, U.K. and Sakim H.A.M. (2010) “An Effective Segmentation Method for Single Stroke Online Cursive Arabic Words” International Conference on Computer Applications and Industrial Electronics, pp. 217-221, Kuala Lumpur, Malaysia.

[11] Eraqi H. M. and Abdel Azeem Sh. (2011) “An On-Line Arabic Handwriting Recognition System Based on a new On-line Graphemes Segmentation Technique” International Conference on Document Analysis and Recognition, pp. 409-413 , Beijing , China.

[12] Klassen T.J. (2001) “Towards Neural Network Recognition of Handwritten Arabic Letters” Master of Science Thesis, Dalhousie University at Halifax, Nova Scotia, Canada.

[۱۳] سلیمانی باغشاه م. ، باقری شورکی س. و کسائی ش. (۱۳۸۵) "بازشناسی و یادگیری حروف مجزای برخط فارسی به روش فازی" چهاردهمین کنفرانس مهندسی برق ایران، تهران.

[۱۴] رضوی س.م. و کبیر ا. (۱۳۸۴) "روشی ساده برای بازشناسی برخط حروف مجزای فارسی" *نشریه دانشکده مهندسی دانشگاه فردوسی مشهد*، سال هفدهم، شماره دوم، صص ۶۳-۷۲.

[۱۵] ساجدی ه. ، جم زاد م. ، ثامتی ح. و باباعلی ب. (۱۳۸۵) "ارائه یک روش مبتنی بر گروه‌بندی برای بازشناسی حروف مجزای برخط فارسی به کمک مدل مخفی مارکوف" دوازدهمین کنفرانس بین‌المللی انجمن کامپیوتر ایران، صص ۴۱۹-۴۲۶، تهران.

[16] Ghods V. and Kabir E. (2010) “Feature Extraction for Online Farsi Characters” 12th International Conference on Frontiers in Handwriting Recognition, pp. 477-482, Bari, Italy.

[17] Alimi A. M. (1997) “An Evolutionary Neuro-Fuzzy Approach to Recognize On-Line Arabic Handwriting” Fourth International Conference on Document Analysis and Recognition, pp. 382-386, Kathmandu, Nepal.

[18] Al-Habian Gh. and Assaleh Kh. (2007) “Online Arabic Handwriting Recognition Using Continuous Gaussian Mixture HMMs” International Conference on Intelligent and Advanced Systems, pp. 1183-1186, Kuala Lumpur, Malaysia.

[19] F. Biadsy, J. El-Sana, and N. Habash. (2006) “Online Arabic handwriting recognition using Hidden Markov Models” 10th International Workshop on Frontiers in Handwriting Recognition, La Baule, France.

[۲۰] حلاوتی ر. (۱۳۸۸) "استخراج خودکار مفاهیم با هدف افزایش کارایی بازشناسی الگوهای ترتیبی" پایان‌نامه دکتری، دانشگاه صنعتی شریف، تهران

[21] Halavati R. and Bagheri Shouraki S. (2007) “Recognition of Persian Online Handwriting Using Elastic Fuzzy Pattern Recognition” *International Journal of Pattern Recognition and Artificial Intelligence* , Vol. 21, No. 3 , pp. 491-513.

[۲۲] رضوی س.م. و کبیر ا. (۱۳۸۷) "بازشناسی برخط کلمات دستنویس فارسی با واژگانی گسترده" پنجمین کنفرانس ماشین بینایی و پردازش تصویر، تبریز.

[23] Faradji F., Faez K. and Mousavi M.H. (2007) “An HMM-based Online Recognition System for Farsi Handwritten Words” International Conference on Intelligent and Advanced Systems, pp. 1187-1192, Kuala Lumpur, Malaysia.

[24] Rabiner L. R. (1989) “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition” *Proceeding of the IEEE*, Vol. 77, No. 2, pp. 257-287.

[25] Duda R. O. , Hart P. E. and Stork D. G. (2001) “*Pattern Classification*” 2nd edition , Wiley , pp. 130-139.

[26] Fink G. A. (2008) "*Markov Models for Pattern Recognition*" Springer, Berlin, pp. 127-136.

[27] Müller M. (2007) “*Information Retrieval for Music and Motion*” Springer, London, pp. 69-84.

[28] Andersson F. (2003) “Bezier and B-Spline Technology” M.Sc. Thesis, Umeå University, Sweden.

[29] website: http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html

واژه نامه فارسی به انگلیسی

ابتدایی	Primitives
ارزیابی	Evaluation
ارگادیک	Ergodic
اقلیدسی	Euclidean
انطباق	Adaptation
بازشناسی نوری نویسه‌ها	Optical Character Recognition (OCR)
برخط	Online
برون خط	Offline
بلادرنگ	Real-time
بیشینه سازی امید ریاضی	Expectation Maximization (EM)
پیچش زمانی پویا	Dynamic Time Warping (DTW)
پیشرو-پسرو	Forward-Backward
تپه	Hill
تجزیه‌ای	Analytical
تحلیل سند	Document analysis
تصدیق امضا	Signature verification
تصدیق نویسنده	Writer verification
تطبیق رشته	String matching
تفکیک	Resolution
توپولوژی چپ به راست	Left-Right topology
توپولوژی خطی	Linear topology
جدول تداخل	Confusion table
حالت	State
حداقل فاصله ویرایشی	Minimum edit distance
حذف قلاب	Dehooking
حرکت	Stroke
خط پیوند	Ligature
خط در اینچ	line per inch (lpi)
خط زمینه	Baseline
خوشه بندی	Clustering
درج	Insertion
درخت تصمیم	Decision tree
درست نمایی	Likelihood
دودویی	Binary
رمزگشایی	Decoding
زنجیره مارکوف	Markov chain
زیر-کلمه	Sub-word
سرعت پاسخ	Response rate
شناسائی نویسنده	Writer identification
صفحه رقومی کننده	Tablet digitizers
فاصله بلوک شهری	City-block distance
فیلتر میانگین متحرک	Moving-average filter
قطعه بندی	Segmentation
قطعه بندی اضافی	Over-segmentation
کاهش فرهنگ لغت	Lexicon reduction
کلی نگر	Holistic
ماتریس انتشار	Emission matrix
ماتریس انتقال	Transition matrix
ماشین بردار پشتیبان	Support Vector Machine (SVM)
ماشین کشسان فازی	Fuzzy Elastic Machine (FEM)
مثبت صحیح	True Positive (TP)
مثبت کاذب	False Positive (FP)
مدل مخفی مارکوف	Hidden Markov Model (HMM)
مدل مخلوط گوسی	Gaussian Mixture Model (GMM)
مشاهده	Observation
ناحیه فعال	Active area
نقشه‌های خود سازمانده	Self-Organizing Maps (SOM)
نمونه برداری مجدد	Resampling
همترازی	Alignment
هموارسازی	Smoothing
وازدگی	Rejection
یادگیری	Learning

واژه نامه انگلیسی به فارسی

Active area	ناحیه فعال
Adaptation	انطباق
Alignment	همترازی
Analytical	تجزیه‌ای
Baseline	خط زمینه
Binary	دودویی
City-block distance	فاصله بلوک شهری
Confusion table	جدول تداخل
Clustering	خوشه بندی
Decision tree	درخت تصمیم
Decoding	رمزگشایی
Dehooking	حذف قلاب
Document analysis	تحلیل سند
Dynamic Time Warping (DTW)	پیچش زمانی پویا
Emission matrix	ماتریس انتشار
Ergodic	ارگادیک
Euclidean	اقلیدسی
Evaluation	ارزیابی
Expectation Maximization (EM)	بیشینه سازی امید ریاضی
False Positive (FP)	مثبت کاذب
Forward-Backward	پیشرو-پسرو
Fuzzy Elastic Machine (FEM)	ماشین کشسان فازی
Gaussian Mixture Model (GMM)	مدل مخلوط گوسی
Hidden Markov Model (HMM)	مدل مخفی مارکوف
Hill	تپه
Holistic	کلی نگر
Insertion	درج
Learning	یادگیری
Left-Right topology	توپولوژی چپ به راست
Lexicon reduction	کاهش فرهنگ لغت
Ligature	خط پیوند
Likelihood	درست نمایی
line per inch (lpi)	خط در اینچ
Linear topology	توپولوژی خطی
Markov chain	زنجیره مارکوف
Minimum edit distance	حداقل فاصله ویرایشی
Moving-average filter	فیلتر میانگین متحرک
Observation	مشاهده
Offline	برون خط
Online	برخط
Optical Character Recognition (OCR)	بازشناسی نوری نویسه‌ها
Over-segmentation	قطعه بندی اضافی
Primitives	ابتدایی
Real-time	بلادرنگ
Rejection	وازدگی
Resampling	نمونه برداری مجدد
Resolution	تفکیک
Response rate	سرعت پاسخ
Segmentation	قطعه بندی
Self-Organizing Maps (SOM)	نقشه‌های خود سازمانده
Signature verification	تصدیق امضا
Smoothing	هموارسازی
State	حالت
String matching	تطبیق رشته
Stroke	حرکت
Sub-word	زیر-کلمه
Support Vector Machine (SVM)	ماشین بردار پشتیبان
Tablet digitizers	صفحه رقومی کننده
Transition matrix	ماتریس انتقال
True Positive (TP)	مثبت صحیح
Writer identification	شناسائی نویسنده
Writer verification	تصدیق نویسنده

Abstract

In this thesis, a method for online Persian handwriting recognition based on letter segmentation and recognition using HMM is proposed. Typed or handwritten image is called offline because its data is available after writing is completed whiles handwritten on digital devices like touch cell phones and tablets is called online since its data is available simultaneous with writing. Feature extraction from offline handwritten needs image processing but features of online handwritten are extracted directly from its information including horizontal and vertical co-ordinates.

Sub-words were initially over-segmented based on the vector connecting consecutive point's angle with x-axis and then filtered for some false results using simple rules. Regarding existence of some extra segmentation points, each combination of sub-word's segments is scored as the mean of HMM normalized probabilities of its segments and their top scores are chosen as the final candidates for letter group recognition, candidates which don't match any sub-word in lexicon are removed, then remained candidates which are not the same as input sub-word regarding existence of up and down signs are deleted. Then those candidates with non-identical number of strokes up and down to the input sub-word are deleted.

Segmentation had accuracy of 85.47% and 60.1% of found segmentation points were false. Sub-word recognition was 37.85% for first candidate and 73.34% for first-10 candidates. This accuracy is very lower than holistic methods previously performed on the same database. It's because of segmentation-based method nature which is heavily dependent to segmentation stage and poor segmentation leads to low accuracy of sub-word recognition. No similar research is performed on the same database yet for comparison.

The advantage of segmentation-based method over holistic method is that it can recognize each sub-word in the text dictionary, but the holistic method recognizes only handwritten samples available in the database; in this method, however, segmentation stage errors extends to the recognition stage degrading dramatically the whole accuracy.

Keywords: Online Persian Handwriting Recognition, Segmentation, HMM

Shahrood University of Technology

Faculty of Electrical and Robotic Engineering

Online Persian Handwriting Recognition

Hadi Taghizadeh

Supervisor**:**

Dr. Alireza Ahmadifard

Advisor:

Dr. Ali Soleymani

FOR THE DEGREE OF

MASTER OF SCIENCE

January 2013