گزارش کار پروژه درس هوش مصنوعی

موضوع تشخیص موجودیت های اسمی در متن

استاد گرامی: دکتر بهروز مینایی

محسن ایمانی، سعید عادل‌مهربان

نگارنده: ایوب ایمانی،

شماره دانشجویی:89521089

آدرس الکترونیکی: ayyoobimani@gmail.com

فهرست

چکیده 2
مقدمه 2
روش مبتنی بر واژه‌نامه 2
روش‌های مبتنی بر قواعد زبان 3
روش‌های آماری 3
روش پنهان مارکوف: 3
سه مساله اصلي 6
فرضيات تئوري مدل مخفي ماركوف 6
مساله ارزيابي و الگوريتم پيشرو 7
مساله كد گشايي و الگوريتم ويتربي (Viterbi) 10
مساله يادگيري 11
الگوريتم بام- ولش 12
پیاده سازی 14
نتیجه گیری 15
سپاس و قدردانی 15
منابع 15

چکیده

در این پروژه هدف تشخیص موجودیت های اسمی در متن است، برای این کار روش های متعددی مانند روش های مبتنی بر واژه نامه، روش های بر مبنای عبارات با قاعده و گرامر زبان و روش های آماری وجود دارد، در این گزارش ابتدا به معرفی این روش ها می پردازیم و سپس با دقت بیشتری روش های آماری را بررسی می کنیم.

واژه‌های کلیدی: پردازش زبان‌های طبیعی، تشخیص موجودیت‌های اسمی، مدل به‌هم‌ریختگی بیشینه، مدل پنهان مارک او، هوش مصنوعی

مقدمه

موجودیت نامدار به عبارتی گفته می‌شود که برای ارجاع به نمونه های یک مقوله‌ی مشخص مانند شخص، شرکت یا موسسه، تاریخ، بیماری، گونه‌ای باکتری و غیره بکار می‌رود. نیاز به شناسایی موجودیت‌های نامدار، در دنیای امروز که عصر ارتباطات و اطلاعات است رو به رشد می‌باشد. شناسایی موجودیت‌های نامدار برای جستجوهای معنادار، ترجمه‌ی خودکار، استخراج خودکار مفاهیم متن، کشف ارجاعات درمتن و بسیاری دیگر از زمینه‌های مربوط به پردازش زبان های طبیعی کاربرد دارد.

از سامانه های تشخیص موجودیت های اسمی بیشتر برای تشخیص اشخاص، مکان‌ها، زمان و مراکز یا ارگان‌ها استفاده می‌شود. گاه نیز به طور خاص برای تشخیص ژن‌ها یا مثلا فقط برای کشف اسامی شرکت‌های تولید کننده فولاد از روی متون مربوطه بکار رود.

روش های موجود برای رویارویی با مساله:

روش مبتنی بر واژه‌نامه

در این روش‌ها از واژه‌نامه‌هایی که در آن موجودیت‌ها و نوع آن‌ها در یک زمینه خاص ذکر شده است استفاده می‌شود و پس از پیدا کردن یک واژه در یک مقوله خاص از واژه‌نامه مربوط به آن مقوله استفاده می‌شود.

تهیه و به‌روزنگهداری این واژه نامه‌ها با صرف هزینه و زمان بسیار زیادی ممکن می‌شود و موضوع دیگری که این مشکل را بسیار بزرگتر و وخیم‌تر می‌کند این است که واژه‌نامه‌ها برای هر زبان باید به طور جداگانه استفاده شوند بنابر این هزینه بسیار بالا تر می‌شود.

مشکل دیگر این روش این است که ممکن است کلمه اختصاری که نشان‌دهنده یک سرکلمه مانند هما است به جای اینکه هواپیمایی ملی ایران برگردانده شود به عنوان اسم یک شخص برگردانده شود. و یا مثلا اگر در واژه‌نامه کلمه‌ی

وزارت علوم، تحقیقات و فناوری" موجود باشد، این روش قادر به تشخیص متنی که در آن تنها واژهی وزارت علوم" آورده شده است نخواهد بود.

تشخیص برخی واژهها مانند pink Floyd در متن نیز بدون استفاده از واژهنامهها بسیار سخت میشود، بنابراین میتوان از این روش در کنار روش های دیگر و برای بهبود نتیجه استفاده نمود.

روشهای مبتنی بر قواعد زبان

در این روش از قواعد ثابتی که در یک زبان وجود دارد استفاده میشود به عنوان مثال در زبان انگلیسی اسامی خاص با حروف بزرگ شروع میشوند و در زبان عربی دو حرف "ال" اول هر اسم خاص میآید، ویا در زبان انگلیسی اگر قبل از کلمهای کلمهی Dr و یا Mr بیاید به احتمال قوی آن کلمه اسم است.
مشکل بزرگ این روش این است که در این روش باید تمامی قواعد را برنامهنویسی کرد و در صورتی که تعدادی از قواعد از قلم بیفتد بازده این روش کاهش پیدا میکند، و با گذشت زمان نیز قواعد جدیدی اضافه میشود بنابراین باید همواره به بهروزکردن برنامه پرداخت.

روشهای آماری

در این زمینه چندین روش مانند روش پنهان مارکوف، روش بههمریختگی بیشینه و روش CRFs میتوان نام برد. در این مقاله ما به بررسی روش پنهان مارکوف میپردازیم و نهایتا با استفاده از برنامه stanford ner system کارایی روش CRFs بر روی دادههای فارسی را بررسی میکنیم. این روشها با استفاده از تکنیکهای یادگیری ماشین ابتدا ماشین با استفاده از تعدادی داده تمرینی آموزش داده میشود سپس میتواند از آن برای برچسبزنی دادههای اصلی استفاده نمود.

روش پنهان مارکوف:

این الگوریتم برای برچسب زنی یک دنباله از اشیا استفاده میشود، در این روش ابتدا دنبالهای از اشیا دیده می شود و چون مقولهای که باید اشیا دیده شده را برحسب آن برچسبزنی کرد از دید ماشین پنهان است به آن مدل پنهان میگوییم.
یک ماشین پنهان مارکوف دارای اجزای زیر است:

- مجموعه برچسب ها $Q = q_1 q_2 \dots q_N$
- احتمالات انتقال از یک حالت به حالت دیگر $A = a_{11} a_{12} \dots a_{n1} \dots a_{nn}$

- دنباله مشاهدات(کلمات) $O = o_1 o_2 \dots o_T$

- احتمال اینکه تگ b_i مربوط به کلمه o_t باشد. $B = b_i(o_t)$

- حالات خاص ابتدایی و انتهایی q_0, q_F

$$A = \{a_{ij}\}$$

ماتریس انتقال حالت : یک مجموعه از احتمالات انتقال در بین حالتها

$$a_{ij} = p\{q_{t+1} = j | q_t = i\}, \quad 1 \le i, j \le N,$$

که در آن q_t بیانگر حالت فعلی می باشد. احتمالات انتقال باید محدودیتها طبیعی یک توزیع احتمال تصادفی را برآورده نمایند. این محدودیتها شامل موارد زیر می گردند.

$$a_{ij} \ge 0, \quad 1 \le i, j \le N$$

$$\sum_{j=1}^{N} a_{ij} = 1, \quad 1 \le i \le N$$

برای حالات مدل ارگودیک برای تمام i و j ها مقدار a_{ij} بزرگتر از صفر است و در موردی که اتصالی بین حالات وجود ندارد $a_{ij} = 0$.

$B = \{b_j(k)\}$ توزیع احتمال مشاهدات: یک توزیع احتمال برای هر یک از حالتها

.

$$b_j(k) = p\{o_t = \nu_k | q_t = j\}, \quad 1 \le j \le N, \quad 1 \le k \le M \qquad (۵)$$

که در آن ν_k بیانگر k^{th} سمبل مشاهده شده در الفبا است و o_t بیانگر بردار پارامترهای ورودی فعلی می باشد. در مورد مقادیر احتمال حالتها نیز شرایط موجود در نظریه احتمال باید رعایت گردند.

$$b_j(k) \ge 0, \quad 1 \le j \le N, \quad 1 \le k \le M$$

$$\sum_{k=1}^{M} b_j(k) = 1, \quad 1 \le j \le N$$

مارکوف برای ساده کردن مساله فرض زیر را در نظر گرفت

$$P(q_i | q_1 \ldots q_{i-1}) = P(q_i | q_{i-1})$$

فرض او این بود که احتمال آمدن یک برچسب در دنباله فقط به برچسب قبل از آن بستگی دارد و نه به سایر برچسب‌ها بدون این فرض به پیچیدگی الگوریتم افزوده می‌شد و مرتبه زمانی اجرای آن نیز بالا می‌رود.

واضح است که در این مدل تعدادی برچسب ثابت تعریف می‌شود و بعد از هر برچسب نیز یکی از برچسب‌های موجود در این مجموعه می‌آید:

$$\sum_{j=1}^{n} a_{ij} = 1 \quad \forall i$$

سه مساله اصلي

برای اینکه مدل HMM در دنیای واقعی قابل استفاده باشد باید سه مساله مهم حل شود. این سه مساله به قرار زیرند:

1- مساله ارزیابی(Evaluation Problem)

با داشتن دنباله مشاهدات $O=\{O_1,...,O_t\}$ و مدل $\lambda=\{A,B,\pi\}$ چگونه $P(O\mid\lambda)$، احتمال تولید دنباله مشاهدات توسط λ را محاسبه نماییم؟

2- مساله کدگشایی (Decoding problem)

با داشتن دنباله مشاهدات $O=\{O_1,...,O_t\}$و مدل $\lambda=\{A,B,\pi\}$ چگونه دنباله حالات بهینه $Q=\{q_1,...,q_t\}$ برای تولید$O=\{O_1,...,O_t\}$را بدست آوریم؟

3- مساله آموزش (Learning problem)

چگونه پارامترهای مدل، A,B,π، را بدست آوریم؟

فرضيات تئوري مدل مخفي ماركوف

برای اینکه مدل مخفی مارکوف از لحاظ ریاضی و محاسباتی قابل بیان باشد فرضهای زیر در مورد آن در نظر گرفته می شود.

1- فرض مارکوف

با داشتن یک مدل مخفی مارکوف، احتمال انتقال از حالت i به حالت j به صورت زیر تعریف می شود:

$$a_{ij}=p\{q_{t+1}=j|q_t=i\}.$$

به بیان دیگر فرض می شود که حالت بعدی تنها به حالت فعلی بستگی دارد. مدل حاصل از فرض مارکوف یک مدل HMM مرتبه صفر می باشد.

در حالت کلی، حالت بعدی می تواند با k حالت قبلی وابسته باشد. این مدل که مدل HMM مرتبه k ام گفته می شود، با استفاده از احتمالات انتقال به صورت زیر تعریف می گردد.

$$a_{i_1i_2...i_kj}=p\{q_{t+1}=j|q_t=i_1,q_{t-1}=i_2,...,q_{t-k+1}=i_k\},\quad 1\le i_1,i_2,...,i_k,j\le N.$$

به نظر می رسد که یک مدل HMM از مرتبه بالاتر باعث افزایش پیچیدگی مدل می شود. علی رغم اینکه مدل HMM مرتبه اول متداول ترین مدل است، برخی تلاشها برای استفاده از مدلهای دارای مرتبه بالاتر نیز در حال انجام می باشد.

2- فرض ایستایی(stationarity)

در اینجا فرض می شود که احتمال انتقال در بین حالات از زمان واقعی رخداد انتقال مستقل است. در این صورت می توان برا ی هر t_1 و t_2 نوشت:

$$p\{q_{t_1+1} = j | q_{t_1} = i\} = p\{q_{t_2+1} = j | q_{t_2} = i\},$$

2- فرض استقلال خروجی

در این حالت فرض می شود که خروجی (مشاهدات) فعلی به صورت آماری از خروجی قبلی مستقل است. می توان این فرض را با داشتن دنباله ای از خروجی ها مانند بیان نمود:

$$\boldsymbol{O} = o_1, o_2, \ldots, o_T$$

آنگاه مطابق با این فرض برای مدل HMM با نام λ خواهیم داشت:

$$p\{\boldsymbol{O} | q_1, q_2, \ldots, q_T, \lambda\} = \prod_{t=1}^{T} p(o_t | q_t, \lambda).$$

اگر چه بر خلاف دو فرض دیگر این فرض اعتبار کمتری دارد. در برخی حالات این فرضیه چندان معتبر نیست و موجب می شود که مدل HMM با ضعفهای عمده ای مواجه گردد.

مساله ارزیابي و الگوريتم پيشرو (forward)

در این حالت مساله این است که با داشتن مدل $\lambda = (A, B, \pi)$ و دنباله مشاهدات $O = \{O_1, O_2, ..., O_T\}$ باید مقدار $P(O | \lambda)$ را پیدا نماییم. می توانیم این مقدار را با روشهای آماری مبتنی بر پارامترها محاسبه نماییم. البته این کار به محاسباتی با پیچیدگی $O(N^T)$ احتیاج دارد. این تعداد محاسبات حتی برای مقادیر متوسط t نیز بسیار بزرگ است. به همین دلیل لازم است که راه دیگری برای این محاسبات پیدا نماییم. خوشبختانه روشی ارائه شده است که پیچیدگی محاسباتی کمی دارد و از متغیر کمکی $\alpha_t(i)$ با نام متغیر پیشرو استفاده می کند.

متغیر پیشرو به صورت یک احتمال از دنباله مشاهدات $O = \{O_1, O_2, ..., O_t\}$ تعریف می شود که در حالت ***i*** خاتمه می یابد. به بیان ریاضی:

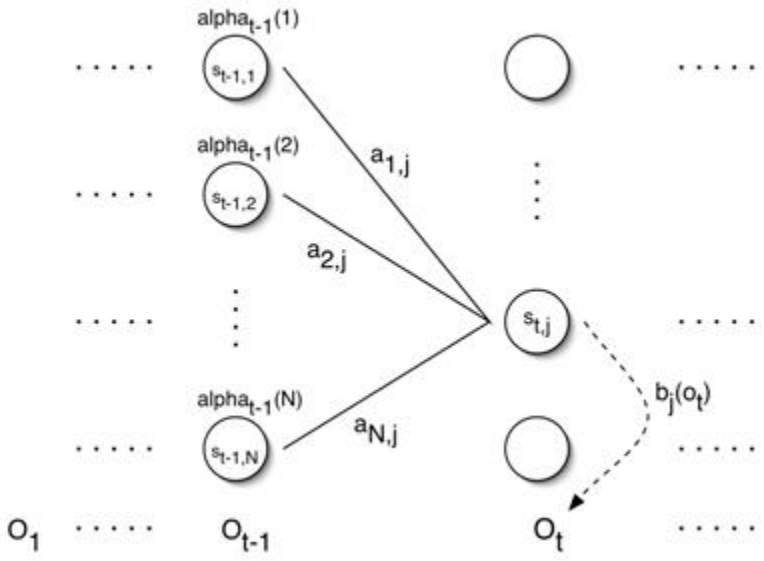
$$\alpha_t(i) = p\{o_1, o_2, \ldots, o_t, q_t = i|\lambda\} \qquad (۱۷)$$

آنگاه به سادگی مشاهده می شود که رابطه بازگشتی زیر برقرار است.

$$\alpha_{t+1}(j) = b_j(o_{t+1}) \sum_{i=1}^{N} \alpha_t(i) a_{ij}, \quad 1 \le j \le N, \quad 1 \le t \le T-1 \qquad (۱۸)$$

که در آن

$$\alpha_1(j) = \pi_j b_j(o_1), \quad 1 \le j \le N$$



شکل 4: احتمالات پیشرو

با داشتن این رابطه بازگشتی می توانیم مقدار زیر را محاسبه نماییم.

$$\alpha_T(i), \quad 1 \le i \le N$$

و آنگاه احتمال $P(O \mid \lambda)$ به صورت زیر محاسبه خواهد شد:

$$p\{\boldsymbol{O}|\lambda\} = \sum_{i=1}^{N} \alpha_T(i). \qquad (۱۹)$$

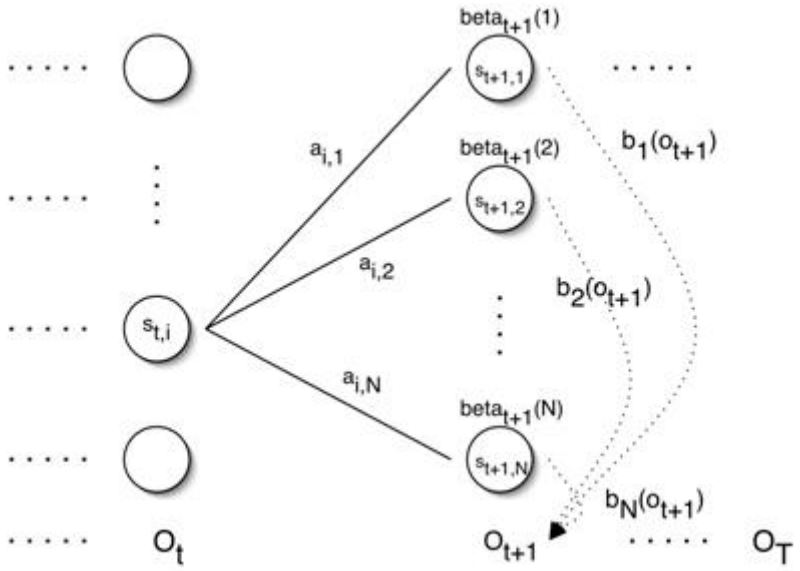
پیچیدگی محاسباتی روش فوق که به الگوریتم پیشرو معروف است برابر با $O(N^2T)$ است، که در مقایسه با حالت محاسبه مستقیم که قبلا گفته شد، و دارای پیچیدگی نمایی بود، بسیار سریعتر است.

روشی مشابه روش فوق را می توان با تعیین متغیر پسرو، $\beta_t(i)$، به عنوان احتمال جزئی دنباله مشاهدات $O = \{O_{t+1}, O_{t+2}, \ldots, O_T\}$ در حالت ***i*** تعریف نمود. متغیر پیشرو را می توان به شکل زیر نمایش داد.

$$\beta_t(i) = p\{o_{t+1}, o_{t+2}, \ldots, o_T | q_t = i, \lambda\} \qquad (۲۰)$$

مانند روش پیشرو یک رابطه بازگشتی به شکل زیر برای محاسبه $\beta_t(i)$ وجود دارد.

$$\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j) a_{ij} b_j(o_{t+1}), \quad 1 \le i \le N, \quad 1 \le t \le T-1 \qquad (۲۱)$$



شکل 5: احتمالات پسرو

که در آن

$$\beta_T(i) = 1, \quad 1 \le i \le N$$

می توان ثابت کرد که

$$\alpha_t(i)\beta_t(i) = p\{O, q_t = i|\lambda\},\ 1 \le i \le N,\ 1 \le t \le T \qquad (۲۲)$$

آنگاه می توان با کمک هر دو روش پیشرو و پسرو مقدار احتمال $P(O|\lambda)$ را محاسبه نمود.

$$p\{O|\lambda\} = \sum_{i=1}^{N} p\{O, q_t = i|\lambda\} = \sum_{i=1}^{N} \alpha_t(i)\beta_t(i) \qquad (۲۳)$$

رابطه فوق بسیار مهم و مفید است و بخصوص برای استخراج روابط آموزش مبتنی بر گرادیان لازم می باشد.

مساله کد گشایي و الگوریتم ویتربي (Viterbi Algorithm)

با فرض داشتن یک ماشین مارکوف اگر بخواهیم بهترین مجموعه برچسب‌های ممکن برای یک دنباله از مشاهده‌ها را پیدا کنیم الگوریتم اولیه‌ای که به ذهن می‌رسد این است که تمام حالت‌های ممکن برای دنباله برچسب‌ها را پیدا کنیم و احتمال وقوع هر حالت را محاسبه کنیم و در انتها حالت با بیشترین امکان وقوع را انتخاب کنیم. در این حالت با فرض اینکه N برچسب و T کلمه داریم پیچیدگی زمانی $O(N^T)$ خواهد بود.

در مدل مارکوف از الگوریتم Viterbi که روشی مانند روش‌های برنامه نویسی پویا است برای پیدا کردن بهترین دنباله برچسب‌ها استفاده می‌شود.

در این حالت می خواهیم با داشتن دنباله مشاهدات $O = \{O_1,...,O_t\}$ و مدل $\lambda = \{A, B, \pi\}$ دنباله حالات بهینه $Q = \{q_1,...,q_t\}$ برای تولید $O = \{O_1,...,O_t\}$ را بدست آوریم.

یک راه حل این است که محتمل ترین حالت در لحظه t را بدست آوریم و تمام حالات را به این شکل برای دنباله ورودی بدست آوریم. اما برخی مواقع این روش به ما یک دنباله معتبر و با معنا از حالات را نمی دهد به همین دلیل باید راهی پیدا نمود که یک چنین مشکلی نداشته باشد.

در یکی از این روشها که با نام الگوریتم Viterbi شناخته می شود، دنباله حالات کامل با بیشترین مقدار نسبت شباهت پیدا می شود. در این روش برای ساده کردن محاسبات متغیر کمکی زیر را تعریف می نماییم.

$$\delta_t(i) = \max_{q_1 q_2 \cdots q_{t-1}} p\{q_1, q_2, \ldots, q_{t-1}, q_t = i, o_1, o_2, \ldots, o_{t-1} | \lambda\},$$

که در شرایطی که حالت فعلی برابر با *i* باشد، بیشترین مقدار احتمال برای دنباله حالات و دنباله مشاهدات در زمان t را می دهد. به همین ترتیب می توان روابط بازگشتی زیر را نیز بدست آورد.

$$\delta_{t+1}(j) = b_j(o_{t+1}) \left[\max_{1 \le i \le N} \delta_t(i) a_{ij} \right], \quad 1 \le i \le N, \quad 1 \le t \le T-1 \qquad (۲۴)$$

که در آن

$$\delta_1(j) = \pi_j b_j(o_1), \quad 1 \le j \le N$$

به همین دلیل روال پیدا کردن دنباله حالات با بیشترین احتمال از محاسبه مقدار $\delta_j(i), i \le j \le N$ و با کمک رابطه فوق شروع می شود. در این روش در هر زمان یک اشاره گر به حالت برنده قبلی خواهیم داشت. در نهایت حالت j^* را با داشتن شرط زیر بدست می آوریم.

$$j^* = \arg \max_{1 \le j \le N} \delta_T(j),$$

و با شروع از حالت j^*، دنباله حالات به شکل بازگشت به عقب و با دنبال کردن اشاره گر به حالات قبلی بدست می آید. با استفاده از این روش می توان مجموعه حالات مورد نظر را بدست آورد. این الگوریتم را می توان به صورت یک جستجو در گراف که نودهای آن برابر با حالتها مدل HMM در هر لحظه از زمان می باشند نیز تفسیر نمود.

مساله یادگیري

به طور کلی مساله یادگیری به این موضوع می پردازد که چگونه می توان پارامترهای مدل HMM را تخمین زد تا مجموعه داده های آموزشی به بهترین نحو به کمک مدل HMM برای یک کاربرد مشخص بازنمایی شوند. به همین دلیل می توان نتیجه گرفت که میزان بهینه بودن مدل HMM برای کاربردهای مختلف، متفاوت است. به بیان دیگر می توان از چندین معیار بهینه سازی متفاوت استفاده نمود، که از این بین یکی برای کاربرد مورد نظر مناسب تر است. دو معیار بهینه سازی مختلف برای آموزش مدل HMMوجود دارد که شامل معیار بیشترین شباهت (ML) و معیار ماکزیمم اطلاعات متقابل ((Maximum Mutual Information (MMI) می باشند. آموزش به کمک هر یک از این معیارها در ادامه توضیح داده شده است.

معیار بیشترین شباهت (Maximum Likelihood (ML))

در معیار ML ما سعی داریم که احتمال یک دنباله ورودی O^w که به کلاس w تعلق دارد را با داشتن مدل HMM همان کلاس بدست آوریم. این میزان احتمال برابر با نسبت شباهت کلی دنباله مشاهدات است و به صورت زیر محاسبه می شود.

$$L_{tot} = p\{\boldsymbol{O}^{\mathbf{w}} | \lambda_w\}$$

با توجه به رابطه فوق در حالت کلی معیار ML به صورت زیر تعریف می شود.

$$L_{tot} = p\{\boldsymbol{O} | \lambda\} \qquad (۲۵)$$

اگر چه هیچ راه حل تحلیلی مناسبی برای مدل $\lambda = \{A, B, \pi\}$ وجود ندارد که مقدار L_{tot} را ماکزیمم نماید، لیکن می توانیم با استفاده از یک روال بازگشتی پارامترهای مدل را به شکلی انتخاب کنیم که مقدار ماکزیمم بدست آید.
روش Baum-Welch و یا روش مبتنی بر گرادیان از جمله این روشها هستند.

الگوریتم بام- ولش

این روش را می توان به سادگی و با محاسبه احتمال رخداد پارامترها و یا با محاسبه حداکثر رابطه زیر بر روی $\bar{\lambda}$ تعریف نمود.

$$Q(\lambda, \bar{\lambda}) = \sum_{q} p\{q | \boldsymbol{O}, \lambda\} \log[p\{\boldsymbol{O}, q, \bar{\lambda}\}]$$

یکی از ویژگیهای مخصوص این الگوریتم این است که همگرایی در آن تضمین شده است. برای توصیف این الگوریتم که به الگوریتم پیشرو- پسرو نیز معروف است، باید علاوه بر متغیرهای کمکی پیشرو و پسرو که قبلا تعریف شده اند، متغیرهای کمکی بیشتری تعریف شود. البته می توان این متغیرها را در قالب متغیرهای پیشرو و پسرو نیز تعریف نمود.

اولین متغیر از این دست احتمال بودن در حالت i در زمان t و در حالت j در زمان t+1 است، که بصورت زیر تعریف می شود.

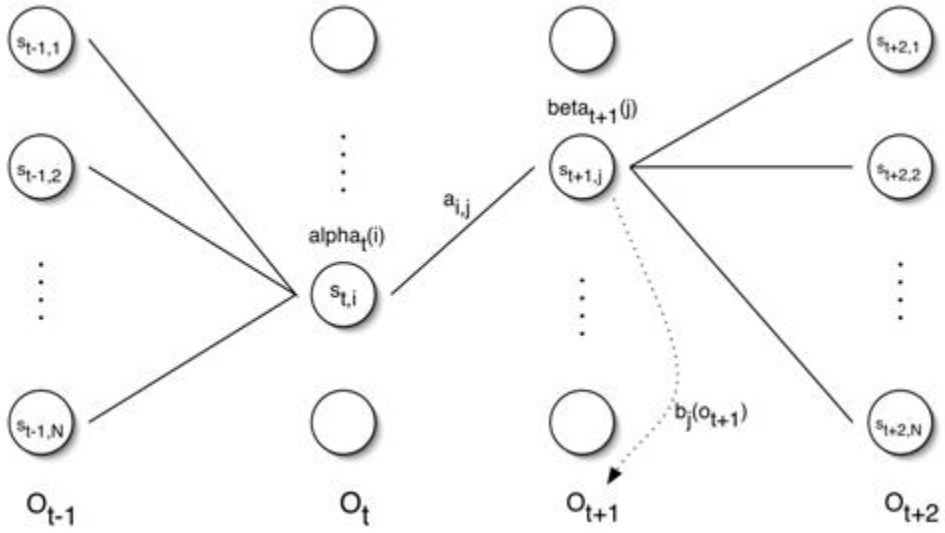
$$\xi_t(i, j) = p\{q_t = i, q_{t+1} = j | \boldsymbol{O}, \lambda\} \qquad (۲۶)$$

این تعریف با تعریف زیر معادل است.

$$\xi_t(i,j) = \frac{p\{q_t = i, q_{t+1} = j, \boldsymbol{O}|\lambda\}}{p\{\boldsymbol{O}|\lambda\}} \qquad (۲۷)$$

می توان این متغیر را با استفاده از متغیرهای پیشرو و پسرو به صورت زیر تعریف نمود.

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}\beta_{t+1}(j)b_j(o_{t+1})}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}\beta_{t+1}(j)b_j(o_{t+1})} \qquad (۲۸)$$



شکل 6: بازتخمین احتمالات انتقال

متغیر دوم بیانگر احتمال پسین حالت i با داشتن دنباله مشاهدات و مدل مخفی مارکوف می باشد و به صورت زیر بیان می شود.

$$\gamma_t(i) = p\{q_t = i|\boldsymbol{O}, \lambda\} \qquad (۲۹)$$

این متغیر را نیز می توان در قالب متغیرهای پیشرو و پسرو تعریف نمود.

$$\gamma_t(i) = \left[\frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N}\alpha_t(i)\beta_t(i)}\right] \qquad (۳۰)$$

رابطه بین دو متغیر فوق بصورت زیر بیان می شود.

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j), \quad 1 \le i \le N, \quad 1 \le t \le M \qquad (۳۱)$$

اکنون می توان الگوریتم آموزش بام – ولش را با ماکزیمم کردن مقدار $P(O \mid \lambda)$ بدست آورد. اگر مدل اولیه ما $\lambda = \{A, B, \pi\}$ باشد، می توانیم متغیرهای پسرو و پیشرو را به استفاده از روابط (18) و (21) و متغیرهای ξ و γ را با استفاده از روابط (28) و (29) تعریف نمود. مرحله بعدی این است که پارامترهای مدل را با توجه به روابط بازتخمین زیر بروزرسانی نماییم.

$$\bar{\pi}_i = \gamma_1(i), \quad 1 \le i \le N \qquad (۳۲)$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \quad 1 \le i \le N, \quad 1 \le j \le N \qquad (۳۳)$$

$$\bar{b}_j(k) = \frac{\sum_{\substack{t=1 \\ O_t = v_k}}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}, \quad 1 \le j \le N, \quad 1 \le k \le M \qquad (۳۴)$$

پیاده سازی

برای بررسی میزان کارایی این الگوریتم‌ها بر روی داده‌های فارسی از برنامه Stanford ner system استفاده شد، این سیستم ابتدا توسط سیستم 15000 داده آموزش داده شد و سپس برای برچسب زنی 70000 داده از آن استفاده شد. برای ارزیابی نیز از precision, recall استفاده شد که نتیجه 20 آزمایش به طور متوسط به

Precision=0.76 recall=0.40

رسید.

نتیجه گیری

همان طور که از نظر تئوری نیز پیش بینی می‌شد هر چند که روش‌های مبتنی بر واژه‌نامه و قواعد بر انتقال از یک زبان به زبان دیگر موفق عمل نمی‌کنند روش‌های آماری بسیار بهتر عمل می کنند و با ویرایش اندک می‌توان آن‌ها را برای اعمال به هر زبانی استفاده کرد.

سپاس و قدردانی

با تشکر فراوان از مجید عسگری که داده‌های مورد نیاز فارسی برای انجام آزمایشات را فراهم آوردند و محسن ایمانی به خاطر راهنمایی‌ها.

منابع

محسن ایمانی، روش‌های تشخیص موجودیت‌های اسمی در متن.

Jurafsky, D., & Martin, J. H. (2007). Speech and Language Processing: An introduction to natural language processing,. Draft.

پونه سادات مرتضوی، مهرنوش شمس‌فرد، شناسایی موجودیت‌های نام‌دار در متون فارسی

The Stanford Natural Language Processing Group