دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده مهندسی برق و کامپیوتر

**عنوان**

# جداسازی بخش‌های نام در اسامی بهم‌چسبیده

**پایان‌نامه برای دریافت درجه کارشناسی**

**در رشته مهندسی کامپیوتر گرایش نرم‌افزار**

**نام و نام خانوادگی**

**نیکی بیات**

**شماره دانشجویی**

۸۱۰۱۹۳۳۶۸

**استاد راهنما:**

**جناب آقای دکتر مسعود اسدپور**

**بهمن ماه ۱۳۹۷**

**تعهدنامه اصالت اثر**

**باسمه تعالی**

**اینجانب نیکی بیات تائید می‌کنم که مطالب مندرج در این پایان نامه حاصل کار پژوهشی اینجانب است و به دستاوردهای پژوهشی دیگران که در این نوشته از آن‌ها استفاده شده است مطابق مقررات ارجاع گردیده است. این پایان نامه قبلا برای احراز هیچ مدرک هم سطح یا بالاتر ارائه نشده است.**
**کلیه حقوق مادی و معنوی این اثر متعلق به دانشکده فنی دانشگاه تهران می باشد.**

**نام و نام خانوادگی دانشجو :**
**نیکی بیات**

**امضای دانشجو :**

**تقدیم[1] به:**

**پدر و مادر عزیزم،**

**که در این راه سایه به سایه همراه من بودند و دعای خیرشان را بدرقه راه اینجانب کردند.**

[1] Dedication

## تشکر و قدردانی[1]:

**سپاس‌گزار معلمی هستم که**

**اندیشیدن را به من آموخت نه اندیشه‌ها را**

از استاد گرامی جناب آقای دکتر مسعود اسدپور که از محضر پرفیض تدریسشان بهره‌ها برده‌ام تشکر و قدردانی می‌کنم. دلسوزی، تلاش و کوشش حضرتعالی در تعلیم و تربیت و انتقال دانش و تجربه قابل ستایش است.

با سپاس بی‌دریغ خدمت راهنما و مشاورم جناب آقای مهندس محسن رئیسی که مرا سایه به سایه در این راه یاری داده‌اند و بدون یاری و راهنمایی‌هایشان، تامین این پژوهش بسیار مشکل می‌نمود.

با تشکر خالصانه خدمت همه دوستان و عزیزانی که اینجانب را در این راه پر فراز و نشیب یاری نموده‌اند.

[1] Acknowledgements

## چکیده[1]

مسأله جداسازی کلمات بهم‌چسبیده در دهه‌های اخیر مورد توجه محققین بسیاری قرار گرفته است و از جمله مباحث حائز اهمیت در حوزه پردازش زبان‌های طبیعی محسوب می‌شود. جداسازی در نگاشت دامنه‌های اینترنتی به پرس‌وجو، تشخیص دامنه‌های مخرب، ترجمه ماشینی، تشخیص گفتار، تصحیح خطاهای نوشتاری و... بسیار کارامد است. از جمله خطاهای نام حذف فاصله میان بخش‌های مختلف آن می‌باشد. نام بهم‌چسبیده به یک نام که فاصله میان بخش‌های مختلف آن حذف شده است گفته می‌شود. بخش‌های نام شامل نام کوچک، نام میانی، نام خانوادگی و... می‌باشد. در این پژوهش هدف ما جداسازی بخش‌های نام در نام‌های بهم‌چسبیده است. جداسازی بخش‌های نام در یک نام بدون فاصله در زمینه‌های متعددی از جمله بازیابی اطلاعات و تطبیق موجودیت کاربرد دارد. در گذشته برای جداسازی کلمات از روش‌های مبتنی بر مدل‌های زبانی استفاده شده است و روش این پژوهش نیز استفاده از مدل‌های زبانی ۱حرفی و ۲حرفی است. جداسازی به صورت خاص بر روی نام در این پژوهش برای اولین بار انجام می‌شود.

از آن جایی که روش استفاده شده باناظر می‌باشد، برای ارزیابی عملکرد الگوریتم جداسازی نیازمند مجموعه دادگان آموزش و آزمون هستیم. دادگان این پژوهش از نام‌های موجود در مجموعه داده‌های متعدد که به صورت رایگان در اینترنت در اختیار محققین قرار گرفته است، استخراج شده است. در مجموع بیش از ۱۲۰ میلیون نام از زبان‌های مختلف برای تولید ۱حرفی و ۲حرفی این پژوهش جمع‌آوری شده‌اند و توکن‌های یکتای موجود در این مجموعه و فرکانس رخداد هر یک محاسبه شده است. عملکرد الگوریتم جداسازی نام بر روی دادگان آزمون با دقت بالایی عمل می‌کند.



## کلمات کلیدی

پردازش زبان‌های طبیعی، شکستن کلمات[2]، تقسیم‌بندی کلمه[3]


[1] Abstract

[2] Word Breaking

[3] Word Segmentation

# فهرست مطالب

# فهرست شکل‌ها

# فهرست جدول‌ها

# فهرست علائم اختصاری

| | |
|---|---|
| OOV | Out of Vocabulary |
| OCR | Optical Character Recognition |
| URL | Uniform Resource Locator |
| MAP | Maximum a Posteriori |

# فصل ۱:

# مقدمه و تعریف مسئله

---

مسأله جداسازی کلمات بهم‌چسبیده[۱]، جداسازی یک رشته حروف بهم‌چسبیده به واحدهای معنادار، مانند کلمات گفته می‌شود. این جداسازی در ذهن انسان هنگام مشاهده یک رشته بهم‌چسبیده از حروف الفبا رخ می‌دهد و در کامپیوتر توسط فرآیندها و الگوریتم‌های پیشرفته قابل پیاده‌سازی می‌باشد. در برخی زبان‌های دنیا میان واحدهای معنادار زبان مانند کلمات، فاصله وجود ندارد، به طور مثال در زبان چینی، برخلاف‌های زبان‌هایی مانند فارسی و عربی، کلمات توسط فاصله از یکدیگر جدا نمی‌شوند در نتیجه تشخیص اجزای معنادار زبان توسط کامپیوتر بدیهی نمی‌باشد، کاربردهای فراوان شکستن رشته حروف بدون فاصله منجر معرفی روش‌های جداسازی کلمات بهم‌چسبیده شدند تا این اجزای معنادار را از رشته‌های بهم‌چسبیده استخراج کنند.

## ۱-۱- تاریخچه‌ای از موضوع تحقیق

جداسازی کلمات بهم‌چسبیده در ۳ دسته کلی بررسی شده است. دسته اول به مسأله تقسیم‌بندی کلمات برای زبان های آسیایی مانند چینی و ژاپنی بازمی گردد. در نوشتار این زبان‌ها، فاصله مرز کلمات در حروف متوالی را مشخص نمی کند به همین دلیل برای تشخیص بخش‌های معنادار نوشتار نیاز به تقسیم-بندی متن به واحدهای سازنده آن هم‌چون کلمات احساس می‌شد. دسته دوم، مسأله تقسیم ترکیبات اضافی[۲] است که سعی در جداسازی ترکیبات اضافی ساخته شده در زبان های اروپایی مانند آلمانی و داچ را دارد[۱]. دسته سوم با پیشرفت دنیای اینترنت و معرفی مفهوم دامنه‌های وب و آدرس سایت‌های اینترنتی، گسترش یافت. قانون ثبت دامنه‌های اینترنتی[۳] به کاربران اجازه ثبت فاصله[۴] میان کلمات سازنده دامنه را نمی‌دهد، همین امر موجب شده است که ترکیب‌های فراوانی از کلمات معنادار بهم‌چسبیده برای ساخت دامنه‌های اینترنتی تولید شوند. پردازش و تحلیل محتویات دامنه‌ها، طبقه‌بندی سایت‌ها، بهبود پرس‌وجوهای سطح وب در هنگام تطبیق مرتبط ترین دامنه‌ها به عبارت جست‌وجو شده، تصحیح خطاهای نوشتاری و املایی کاربران حین جست‌وجو و هم‌چنین تأمین امنیت کاربران در دنیای وب، از جمله کاربرد های جداسازی صحیح عبارات بدون فاصله به کلمات سازنده است و بیش از یک دهه است که مورد بررسی محققین زیادی در این زمنیه قرار گرفته است[۲].

---

۱ Word Breaking

۲ Compound Splitting

۳ URL

۴ Space

جداسازی کلمات بهم‌چسبیده در زمینه‌های بسیاری کاربرد دارد و همین امر موجب شده است مقالات زیادی در این مورد نوشته شوند و روش‌های متنوعی طی سالیان ابداع شوند. این مسأله برای تشخیص ریشه[1] کلمات بیش از ۴۰ سال است که مورد بررسی قرار گرفته است [۳].

## ۲-۱- شرح مسئله تحقیق

ما در این پروژه قصد داریم روش‌های مختلف جداسازی نام‌های بهم‌چسبیده را بررسی کرده و بهترین روش را برای جداسازی بخش‌های نام در اسامی بهم‌چسبیده بیابیم. تحقیق روی این دامنه خاص، نیازمند مجموعه دادگان مخصوص به خود، روش‌های یادگیری متناسب با آن و روش‌های هموارسازی ویژه می‌باشد که در ادامه این گزارش به آن‌ها می‌پردازیم.

## ۳-۱- اهداف کلی تحقیق

هدف این تحقیق جداسازی اسامی بدون فاصله می‌باشد، به عبارت دیگر، در این پژوهش می‌خواهیم سیستمی را طراحی کنیم که اگر یک اسم بهم‌چسبیده را به آن به عنوان ورودی بدهیم، بخش‌های مختلف نام به همراه فاصله صحیح میان آن‌ها را به عنوان خروجی دریافت کنیم، که به آن جداسازی صحیح اسم می‌گوییم.

## ۴-۱- روش انجام تحقیق

در این پژوهش ابتدا تمامی کارهای پیشین مرتبط با جداسازی رشته کلمات بهم‌چسبیده در زبان‌های مختلف مطالعه و بررسی می‌شود، سپس یک مجموعه دادگان غنی از اسامی بین‌المللی جمع آوری شده و نهایتا الگوریتم جداسازی روی این مجموعه دادگان پیاده سازی می‌شود و اصلاحات در الگوریتم اعمال می‌شود تا بهترین دقت و درستی مطلوب در جداسازی اسامی بدون فاصله حاصل شود.

[1] Stem

## ۵-۱- ساختار پایان‌نامه

فصل دوم، شامل مرور ادبیات و بررسی و دسته‌بندی کارهای پیشین مرتبط با این تحقیق و پیش‌زمینه‌های مورد نیاز برای درک هرچه بهتر الگوریتم‌های جداسازی رشته کلمات بدون فاصله خواهد بود.

در فصل سوم ابتدا مسئله مورد پژوهش توضیح داده خواهد شد و سپس روش پیشنهادی این پایان‌نامه مطرح می‌شود و نحوه پیاده‌سازی آن ارائه خواهد شد.

فصل چهارم در برگیرنده‌ی توضیحاتی جامع درمورد چگونگی جمع‌آوری و تولید مجموعه دادگان آموزش و آزمون این پژوهش می‌باشد.

در فصل پنجم، معیارهای ارزیابی معرفی می‌شوند و سپس آزمایش‌های انجام شده و نتایج هریک مشخص شده و نتایج دریافت شده به تفصیل تحلیل و بررسی خواهند شد.

فصل ششم حاوی نتیجه‌گیری کلی حاصل شده از این تحقیق است و محدودیت‌های پیش‌رو مورد بحث قرار می‌گیرد و پیشنهادهایی برای ادامه‌ی مسیر به علاقه‌مندان به این حوزه‌ی ارائه خواهد شد.

در نهایت، در فصل هفتم منابع این پژوهش ذکر خواهد شد.

# فصل ۲:

# مرور ادبیات و کارهای پیشین

پردازش زبان‌های طبیعی شاخه‌ای از علوم کامپیوتر و هوش مصنوعی است که به تعاملات میان کامپیوتر و انسان در حوزه زبان‌های طبیعی می‌پردازد. هدف نهایی این شاخه از علوم کامپیوتر این است که فهم کامپیوتر ها از زبان‌طبیعی را هم‌سطح با انسان کند و نهایتا کامپیوتر را قادر سازد که به زبان‌طبیعی صحبت کند. از جمله حوزه‌های فعالیت این شاخه از علوم کامپیوتر می‌توان به تشخیص گفتار[۱]، خلاصه‌سازی اتوماتیک متن[۲]، ترجمه ماشین[۳] و... اشاره کرد. پردازش زبان‌های طبیعی شاخه‌ای از علم است که منجر به همکاری زبان‌شناس‌ها، متخصصین یادگیری ماشین و متخصصین علوم کامپیوتر می‌شود. پردازش زبان‌های طبیعی به سه بخش عمده تشخیص گفتار، فهم زبان طبیعی و تولید زبان طبیعی تقسیم می‌شود که در این پژوهش بخش فهم زبان طبیعی توسعه داده خواهد شد. فهم و تحلیل زبان انسان، با وجود این که حتی کودکان خردسال قادر به استفاده از آن هستند، دشوار است چرا که سرشار از نمادهای گوناگون است و می‌توان با استفاده از آن یک مفهوم یکسان را به روش‌های مختلف منتقل کرد، هم چنین بی‌نهایت روش برای قرار دادن کلمات در یک جمله وجود دارد و هر یک از این کلمات با توجه به سایر اجزای جمله و متن می‌توانند معانی مختلف داشته باشند در نتیجه فهم معنای متنی که کلمه در آن به کار برده شده است نیز ضروری است.

## ۱-۲- مقدمه‌ای بر تحلیل معنایی متن

وقتی یک انسان صحبت می‌کند، مغز شنونده به صورت ناخودآگاه با استفاده از حس و غریزه درونی و هم‌چنین دانش زبان طبیعی مورد صحبت، متوجه می‌شود که گوینده چه چیزی گفته است. اگر انسان نتواند یک جمله را تحلیل معنایی کند، مکالمه به درستی انجام نمی‌شود، چرا که شنونده در واقع اصلا متوجه مفهوم مورد انتقال نشده است. در عمل جداسازی متن ورودی، نیاز داریم تحلیل معنایی انجام دهیم تا رشته حروف بدون فاصله را به اجزای معنادار بشکنیم، برای تشخیص معنادار بودن یا نبودن یک رشته‌ای از حروف تحت عنوان یک کلمه، نیاز داریم تحلیل معنایی زبان را به درستی انجام دهیم.

---

[۱] Speech Recognition

[۲] Text Summarization

[۳] Machine Translation

## ۲-۲- مرور پژوهش‌های پیشین

مسئله جداسازی رشته کلماتی که بدون فاصله پشت یکدیگر قرار گرفته‌اند، در زبان‌های مختلف و برای موضوعات و کاربردهای گوناگون با استفاده از روش‌های متنوع مورد بررسی قرار گرفته است که نشان از اهمیت این موضوع در حوزه پردازش زبان‌های طبیعی دارد. در ادامه این فصل پژوهش‌های پیشین مرتبط را طبقه‌بندی کرده و هر یک را به تفصیل شرح خواهیم داد.

### ۱-۲-۲- یادگیری

پژوهش‌های پیشین از حیث نوع یادگیری به دو دسته زیر تقسیم می‌شوند:

- **یادگیری با نظارت**

در این روش، کامپیوتر یک نگاشت از ورودی به خروجی‌های صحیح مطابق با رابطه ۱ دارد که بر اساس آن آموزش می‌بیند، سپس تلاش می‌کند دادگان آزمون را نیز با استفاده از برچسب‌های موجود تحلیل و بررسی کند، در واقع در یادگیری با نظارت دادگان دارای برچسب‌هایی هستند که جواب صحیح را برای هر یک مشخص می‌کند.

$$y = f(x)$$ **رابطه ۱**

هدف یادگیری با نظارت این است که با دادن یک ورودی x جدید که خارج از دادگان آموزش الگوریتم است، خروجی صحیح را دریافت کنیم. به این الگوریتم به این دلیل با نظارت گفته می‌شود که برچسب‌های صحیح دادگان آموزش، مانند یک ناظر و راهنما به بهبود عملکرد الگوریتم کمک می‌کنند.

- **یادگیری بدون نظارت**

در این روش تنها دادگان آموزش ورودی را داریم و برچسب صحیح و خروجی متناظر دادگان آموزش در اختیار نیست. هدف یادگیری بدون نظارت پیدا کردن ساختار و توزیع دادگان ورودی

جهت فهم بیشتر داده‌هاست. نام این نوع یادگیری بدون نظارت است چون همانند یادگیری با نظارت برچسب صحیحی از خروجی متناظر هر داده ورودی در اختیار نیست و ناظر راهنمایی وجود ندارد. ماشین در این روش به تنهایی سعی در پیدا کردن یک الگو یا ساختار قابل توجه در دادگان دارد که منجر به نتیجه‌گیری مفید و مطلوب شود. روش‌های یادگیری بدون نظارت اغلب در مواردی که حجم داده‌ها و یا مقیاس پروژه بسیار عظیم است و امکان برچسب زدن تمامی دادگان آموزش را نداریم، استفاده می‌شوند.

## ۲-۲-۲- کاربردهای جداسازی کلمات بهم‌چسبیده

در این بخش زمینه‌های کاربرد جداسازی کلمات بهم‌چسبیده در پژوهش‌های پیشین ذکر خواهد شد. کاربردهای فراوان این حوزه منجر به پیشرفت روش‌ها و الگوریتم‌های مربوط به آن شده و فعالیت‌های این حوزه را هدفمندتر ساخته است.

- **بازیابی اطلاعات[۱] و ارتقا کیفیت جست‌وجو در وب**

موتورهای جست‌وجو در وب یکی از مهم‌ترین کاربردهای الگوریتم‌های بازیابی اطلاعات هستند. برای نمایش بهترین و مرتبط‌ترین نتیجه، پرس‌وجو های کاربران در موتورهای جست‌وجو نیازمند پیش‌پردازش‌های تخصصی می‌باشد، از جمله آن می‌توان به تصحیح خطاهای نوشتاری و املایی و هم‌چنین شکستن آدرس وب‌سایت‌های اینترنتی به منظور تطبیق بهتر با پرس‌وجو کاربر اشاره کرد. هنگامی که کاربران موضوعی را در موتورهای جست‌وجو می‌پرسند، به دفعات رخ می‌دهد که بر اثر عجله و نوشتن با سرعت بالا، فاصله کلمات نوشته نمی‌شود و کلمات بهم‌چسبیده نوشته می‌شوند، موتور جست‌وجو به منظور ارتقاء کیفیت جست‌وجو نیاز به جداسازی این کلمات دارند تا هر یک را جداگانه تحلیل کرده و امتیاز وب‌سایت های مرتبط را بر اساس آن به دست بیاورند، در این موارد، با استفاده از الگوریتم‌های موجود برای جداسازی رشته کلمات بهم‌چسبیده به واحدهای معنادار، این خطاها اصلاح می‌شوند. از طرف دیگر آدرس صفحات اینترنتی بدون فاصله نوشته می‌شود، از این رو برای تبطیق بهینه سایت‌ها به عبارت جست‌وجو شده توسط کاربر،

[۱] Information Retrieval

ضروری است که آدرس به واحدهای معنادار سازنده خود شکسته شود و سپس میزان شباهت و ارتباط آن با پرس‌وجو سنجیده شود، در این مورد نیز با استفاده از الگوریتم‌های شکستن رشته کلمات بدون فاصله، عمل جداسازی را انجام می‌دهیم.[2, 4, 5]

- **ترجمه ماشینی**

در چند دهه اخیر، ماشین‌های ترجمه معرفی و رونق یافتند، وظیفه ماشین ترجمه، ترجمه یک متن از زبان مبدأ به زبان مقصد بدون دخالت انسان می‌باشد. معرفی ماشین ترجمه، چالش‌های جدیدی نیز با خود به همراه داشت. زبان‌های شرق اسیا همانند چینی و ژاپنی از فاصله میان ترکیبات معنادار زبان خود استفاده نمی‌کنند، از طرفی می‌دانیم که در ترجمه ماشین برای ترجمه لغوی کلمات (در این مقاله وارد تحلیل‌های نحوی ترجمه ماشین نمی‌شویم چرا که از حوزه کاری این مقاله خارج می‌باشد) نیاز داریم یک کلمه را جدا کرده و با استفاده از تناظر میان آن کلمه و کلمات زبان هدف ترجمه، معنای آن را متوجه شویم، اکنون اگر مرز کلمات به درستی تشخیص داده نشوند، نمی‌توان با استفاده از دیکشنری یا سایر روش‌های موجود، آن کلمه را ترجمه کرد و عملکرد ماشین ترجمه به شدّت افت خواهد کرد. از طرف دیگر، زبان‌هایی مانند آلمانی، سوئدی، نروژی با استفاده از کلمات مختلف، ترکیبات اضافی بهم‌چسبیده می‌سازند که اگرچه برای کسانی که به این زبان صحبت می‌کنند تشخیص معنای این ترکیبات بدیهی است، در کامپیوتر نیازمند شکستن این ترکیبات اضافی و یافتن اجزای معنادار سازنده آن هستیم تا بتوانیم این عبارات را به درستی در زبان مقصد ترجمه کنیم. به طور مثال ترکیب اضافی بطریِ آب، در زبان آلمانی تبدیل به بطری‌آب می‌شود و برای ترجمه آن نیاز داریم این عبارت را ابتدا به دو واژه آلمانی بطری و آب شکسته و سپس ترجمه کنیم.[۸-٦]

- **شبکه‌های اجتماعی**

همزمان با رشد دنیای اینترنت، شبکه‌های اجتماعی مجازی نیز معرفی شدند و مورد استقبال جمع کثیری از کاربران اینترنت قرار گرفتند، با افزایش بی‌رویه مطالب منتشر شده در این شبکه‌ها، جست‌وجوی میان مطالب نیازمند ابزارهایی برای یافتن بهینه مرتبط‌ترین مطالب شد و همین مهم موجب به‌وجودآمدن برچسب‌های جست‌وجو در شبکه‌های اجتماعی گردید. کاربران با استفاده از این برچسب‌ها مطالب خود را هنگام انتشار دسته‌بندی می‌کنند و زمانی که کاربر دیگری بخواهد در

خصوص آن موضوع اطلاعاتی کسب کند، کافی است برچسب آن را جستجو کند تا تمامی مطالب مرتبط با آن نمایش داده شود. هنگام ایجاد این برچسب‌ها، همانند دامنه‌های اینترنتی، اجازه استفاده از فاصله را نداریم، از این رو کاربران ترکیباتی از کلمات معنادار را به عنوان برچسب به مطلب خود اضافه می‌کنند، در این مرحله، شبکه‌های اجتماعی موظف هستند برای بهبود عملکرد جستجوگرها این برچسب‌ها را به واحدهای سازنده بشکنند و با استفاده از تابع‌های امتیازدهی مربوطه، مرتبط‌ترین برچسب‌ها به عبارت جستجوشده توسط کاربر را نمایش دهند .به طور مثال شبکه‌اجتماعی توییتر می‌تواند با استفاده از این روش، کیفیت موتور جستجو توییت‌های خود را ارتقا بدهد.[۹]

- **طبقه‌بندی وبلاگ‌های اینترنتی و تشخیص دامنه های مخرب**

تشخیص دامنه‌های مخرب اینترنتی و سایت‌های ویروسی و هم‌چنین دامنه‌های ساخته‌شده توسط الگوریتم‌های کامپیوتری نیز از جمله حوزه‌های کاربرد جداسازی کلمات بدون فاصله می‌باشد. دامنه‌های مخرب و ساخته‌شده توسط کامپیوتر اغلب از کلماتی استفاده می‌کنند که در گفت‌وگو و یا نوشتار انسان‌ها فرکانس رخداد بسیار کمی دارند، به همین دلیل چنانچه پس از جداسازی صحیح یک دامنه به کلماتی نادر با فرکانس رخداد کم در زبان برسیم، آن دامنه مشکوک خواهد بود و با احتمال بالاتری به یک وب‌سایت مخرب تعلق دارد. این روش یقینا بدون خطا نیست و ممکن است برخی آدرس‌هایی که به سایت‌های سالم تعلق دارند نیز مخرب شمرنده شوند اما نتایج بررسی‌های انجام شده در این موضوع نشان می‌دهد که حذف دامنه‌های مخرب با این روش (با در نظر گرفتن احتمال خطا) در مجموع دقت بالاتری را برای طبقه‌بندها به ارمغان می‌آورد و امنیت سیستم‌ها و کاربران وب را نیز، بالاتر می‌برد[10, 11].

- **تشخیص گفتار**

در سال‌های اخیر، جداسازی رشته کلمات بدون فاصله در حوزه تشخیص گفتار نیز کاربرد پیدا کرده است. در گفتار کلمات با فاصله از هم جدا نمی‌شوند بلکه سکوت گوینده است که مرز کلمات را برای شنونده مشخص می‌کند. اگر گوینده با سرعت بالایی صحبت کند، گاهی تشخیص این فاصله دشوار می‌شود و هنگام نوشتن متن گفتار، کلمات بهم چسبیده نوشته می‌شوند. به عبارت دیگر کلمات در گفتار رشته‌ای از حروف یا آواها هستند که مرزهای کلمات از میان آن حذف شده

است،جهت بهبود عملکرد الگوریتم‌های تشخیص گفتار، نیازمند شکستن این رشته کلمات هستیم و همین موضوع موجب توسعه الگوریتم‌های جداسازی کلمات بهم‌چسبیده گردیده است.[۱۲, ۱۳]

## ۲-۲-۳- پیاده سازی

مقالات مرتبط هر یک با توجه به محدودیت‌ها و چالش‌های حوزه کاری خویش و زمینه کاربردی و آرمان و هدف تحقیق خود، الگوریتم جداسازی را متفاوت پیاده‌سازی کرده‌اند، در ادامه به انواع پیاده‌سازی‌های موجود در مقالات پیشین اشاره خواهم کرد.

- **مدل‌های زبانی**

مدل زبانی یک توزیع احتمال بر روی یک زبان طبیعی می‌باشد، به عبارت دیگر مدل‌های زبانی تعیین می‌کنند که احتمال رخداد یک رشته از کلمات در زبان طبیعی چقدر می‌باشد. به طور مثال در زبان فارسی احتمال رخداد "او رفت "بیشتر از "نیکی رفت "است چرا که "نیکی "یک اسم خاص است و خیلی کمتر از "او "که ضمیر سوم شخص مفرد است استفاده می‌شود.

- احتمالات nحرفی[۱]

این احتمالات فرکانس رخداد تمامی ترکیب‌های مختلف کنار هم قرار گرفتن توکن‌ها در یک متن مرجع را نمایش می‌دهند. تعداد این توکن‌ها مقدار حرف n در nحرفی را مشخص می‌کند. اگر توکن مورد استفاده حرف باشد، فرکانس رخداد ترکیبات مختلفی از قرار گرفتن حروف در جوار یکدیگر را نشان می‌دهد و چنانچه یک توکن معادل یک کلمه باشد، فرکانس رخداد کلمات متوالی در زبان نمایش داده می‌شوند. به عنوان مثال اگر توکن کلمه باشد، رشته "سلام من نیکی هستم" یک۴حرفی[۲] می‌باشد.

[۱] Ngram

[۲] 4gram

از آن‌جایی که سبک های مختلف زبان مانند متن سند، عنوان سند، لینک و... دارای خواص آماری بسیار متفاوت می‌باشند، عملکرد الگوریتم های پردازش زبان‌های طبیعی برای هر یک از این سبک‌ها کاملا متفاوت می‌باشد در نتیجه nحرفی باید با توجه به هر یک از این سبک‌ها ساخته و یاد گرفته شود. در این پژوهش چون تمامی دادگان ورودی اسامی می‌باشند و همگی یک سبک یکسان دارند، نگران تفاوت سبک‌ها و تاثیر آن‌ها در یادگیری نیستیم.

در nحرفی احتمال رخداد هر توکن به n-1 توکن قبل بستگی دارد، حال اگر n برابر با ۱ باشد، احتمال رخداد توکن‌ها مستقل از یکدیگر در نظر گرفته می‌شود، در این حالت هر کلمه به هیچ کلمه قبل و بعد از خود بستگی ندارد. در مدل ۱حرفی[۱] چون توکن‌ها مستقل از یکدیگر هستند، احتمال رخداد یک رشته‌ای از توکن‌ها در زبان طبیعی تحت بررسی از حاصلضرب احتمال رخداد هر یک از توکن‌های آن رشته مطابق با رابطه ۲ به دست می‌آید.

$$P(Phrase) = \prod_{i=1}^{n} P(token_i) = P(token_۱) \times P(token_۲) \dots P(token_n)$$ **رابطه ۲**

در مدل ۲حرفی[۲] احتمال رخداد یک کلمه را بر اساس کلمه قبل بررسی می‌کنیم. از آن‌جایی که در این مدل کلمات وابسته به کلمات پیشین خود هستند، ترتیب رخداد کلمات حائز اهمیت می‌باشد و احتمالات بر اساس رابطه ۳ محاسبه می‌شوند.

$$P(W_i|W_۰ \dots W_{i-۱}) \approx P(W_i|W_{i-۱})$$ **رابطه ۳**

هنگامی که می‌خواهیم اولین کلمه موجود در یک عبارت را به ۲حرفی اضافه کنیم، یعنی کلمه‌ای را اضافه کنیم که هیچ کلمه‌ای پیش از آن رخ نداده است، از عبارت <S> به عنوان توکن پیشین (توکن آغاز کننده جمله یا عبارت) استفاده می‌کنیم. احتمال شرطی به روش زیر محاسبه خواهد شد:

[۱] Unigram

[۲] Bigram

$$P(w_i|w_{i-1}) = \frac{P(w_i and w_{i-1})}{P(w_{i-1})}$$ **رابطه ۴**

در ادامه دو نمونه از معروف‌ترین nحرفی‌ها توضیح داده خواهند شد.

- nحرفی منتشر شده توسط شرکت ماکروسافت[۱]

این شرکت با استفاده از دادگان جمع‌آوری شده توسط موتور جست‌وجو بینگ[۲] که در حدود صدها میلیارد سطر داده است، nحرفی خود را ساخته است. این دادگان ابتدا دانلود شده، سپس تحلیل نحوی و توکن‌بندی شده اند و در نهایت تبدیل به حروف کوچک[۳] شده و در آخرین مرحله علائم نگارشی از آن‌ها حذف شده است سپس به nحرفی اضافه شده‌اند. توجه داشته باشید که خطاهای املایی در دادگان جمع‌آوری شده اصلاح نشده است.

برخلاف nحرفی شرکت گوگل[۴]، که تعداد رخداد توکن‌ها را به صورت خام گزارش کرده است، ماکروسافت از الگوریتم CALMاستفاده کرده است که به صورت پویا دادگان را بر اساس وب به‌روزرسانی می‌کند. این الگوریتم اطمینان حاصل می‌کند که توکن‌هایی که به تازگی در وب معرفی شده‌اند، پس از کسب فرکانس رخداد معقول، به nحرفی اضافه شوند در نتیجه مدل همواره با وب به روزرسانی‌شده و هیچ‌گاه قدیمی نمی‌شود. الگوریتم CALM هم‌چنین اطمینان حاصل می‌کند که دادگان تکراری بر روی خروجی nحرفی جهت‌دهی بیش از حد ندارند. هم‌اکنون بالاترین مرتبه nحرفی موجود ۵ می‌باشد. از آن‌جایی که این دادگان از وب جمع‌آوری می‌شوند و دنیای پویای وب به همه کشورها تعلق دارد، nحرفی حاصل چندزبانه می‌باشد. از طرف دیگر چون هر روز در دنیای وب اختصارات و کلمات جدیدی تولید می‌شوند و این دادگان از وب جمع‌آوری شده‌اند، فرکانس رخداد این عبارات نیز به درستی محاسبه می‌شود. شرکت ماکروسافت تا ماه اگوست سال ۲۰۱۸ میلادی یک رابط برنامه‌نویسی[۵] رایگان در اختیار عموم قرار داده بود که با استفاده از آن

[۱] Microsoft

[۲] Bing

[۳] Lowercase

[۴] Google

[۵] API

می‌توانستیم از nحرفی این شرکت استفاده کنیم و عمل جداسازی را انجام دهیم، اما این قابلیت هم‌اکنون دیگر به صورت رایگان در اختیار عموم نمی‌باشد.[۱٤]

- nحرفی منتشر شده توسط شرکت گوگل

این شرکت یک ابزار تحت عنوان نمایشگر nحرفی طراحی کرده است که به ازای هر ورودی کاربر، فرکانس رخداد آن ورودی در متن اصلی را طی چندین سال به دست آورده و سپس یک نمودار خطی از فرکانس در سال‌های مختلف رسم می‌کند. متنی که این nحرفی از آن استخراج شده است صفحات اسکن شده بیش از دوازده کتاب‌خانه دانشگاهی در زبان‌های انگلیسی، آلمانی، فرانسوی، چینی، عبری، ایتالیایی و اسپانیایی هستند. در واقع این ابزار به ازای هر کلمه ورودی، میزان محبوبیت آن در کتب را به صورت یک نمودار نمایش می‌دهد. این مجموعه دارای خطاهایی نیز می‌باشد، اول از همه روند تبدیل یک تصویر اسکن شده از کتاب به کاراکتر[۱] یک روند کامل و بدون نقص نمی‌باشد و همواره دارای خطا است، عملکرد این الگوریتم زمانی که کتب قدیمی بررسی می‌شوند به مراتب دشوارتر هم خواهد شد. به طور مثال در خیلی از موارد تبدیل کتب به کاراکتر، حروف sو fبه اشتباه مشابه یکدیگر خوانده می‌شوند و خطای زیادی به وجود می‌آید. نکته دیگر این است که بخش عظیمی از منبع این nحرفی کتب علمی هستند، این موضوع باعث می‌شود رخداد کلمات و عبارات علمی به مراتب بیشتر از سایر عبارات شود که موجب می‌شود فرکانس رخداد بسیاری از کلماتی که در روزمره و جامعه به فراوانی کاربرد دارند، در مقابل این کلمات علمی به طرز محسوسی کاهش یابد. یکی دیگر از مهم‌ترین نقص‌های مفهومی منبعی که گوگل از آن استفاده می‌کند این است که هر کتاب تنها یک‌بار شمرده می‌شود، این در حالی است که برخی کتب میلیون‌ها بار خوانده می‌شوند و برخی دیگر هرگز ورق نخورده‌اند، یکسان گرفتن وزن این دو کتاب منطقی به نظر نمی‌رسد.

- **فرآیند Dirichlet**

این روش برای بهبود عملکرد الگوریتم‌های پردازش گفتار و هم‌چنین ارتقاء کیفیت جداسازی زبان‌های آسیایی به کار رفته است. توزیع این فرآیند در تحلیل‌های غیرپارامتری بیزی به کار می‌رود و به ما اجازه می‌دهد که بخش‌های مختلف مدل را با انعطاف بیشتری اصلاح کنیم.

[۱] OCR

الگوریتم‌های جستجو در برخی مقالات ضعیف بودند و نشان می‌دادند که وابستگی میان کلمات مهم نیست، به طور مثال در نتایج نهایی میان ۲حرفی و ۱حرفی فرق چندانی وجود نداشت. در حالی که تحلیل وابستگی میان کلمات امری کاملا ضروری برای تقسیم‌بندی صحیح کلمات است و با در نظر گرفتن صحیح این وابستگی‌های متنی، عملکرد بهبود شایان توجهی پیدا کند.[۱۵]

- **تطبیق دوجهته بیشینه**[۱]

در این روش از جلو (اولین حرف) و از عقب (آخرین حرف) شروع به حرکت کرده و حرف به حرف جلو می‌رویم تا زمانی که به کلمه‌ای برسیم که در لغت‌نامه وجود دارد، چنانچه چندین کلمه وجود داشته باشد که همگی از منظر دیکشنری معتبر باشند، آن کلمه‌ای انتخاب می‌شود که طول بیشتری داشته باشد. از طرفی اگر در جایی از رشته از حروف کوچک به حروف بزرگ برویم، این نقطه نیز یک نقطه جداسازی محسوب خواهد شد. هنگامی که تطبیق از انتها انجام می‌دهیم از یک لغت‌نامه وارونه استفاده می‌کنیم. برای بهبود عملکرد الگوریتم برای هر سند یک مجموعه مرجع[۲] تعریف می‌شود که شامل مواردی از جمله ترکیب کاراکتر اول همه کلمات عنوان داکیومنت وب و.. می‌شود. اگر کاندیدایی فقط از کلمات مجموعه مرجع تشکیل شده باشد، به سایر کاندیداها اولویت دارد. سپس در اولویت دوم، جداسازی‌ای اهمیت دارد که ترکیبی از مجموعه مرجع و کلمات دیکشنری باشد. اگر دو کاندید داشته باشیم که هر دو تماما از کلمات دیکشنری باشند، آن کاندید که تعداد کلمات کمتری دارد، اولویت دارد. در اولویت بعد، کاندیدایی که ترکیبی از مجموعه مرجع و کلمات غیر دیکشنری باشد، به عنوان جداسازی صحیح برگزیده می‌شود. اگر دو کاندید داشته باشیم که هر دو ترکیبی از کلمات دیکشنری و کلمات غیر دیکشنری باشد، آن کاندید که تعداد کلمات غیر دیکشنری کمتری دارد، انتخاب می‌شود. اگر دو کاندید داشته باشیم که هر دو ترکیبی از کلمات غیر دیکشنری باشند، آن کاندید که تعداد کلمات کمتری دارد، انتخاب می‌شود. برای آموزش صحیح‌تر، یک جدول اضافه می‌شود که کلمات شناخته‌شده غیر دیکشنری را به آن اضافه می‌کنیم تا جایی که ثابت و پایدار شود.[۱۶]

[۱] Maximal Bidirectional Maching

[۲] Reference Base Set

- **جداسازی بر مبنای دادگان آموزش[۱]**

در زبان آلمانی، معمولا چندین کلمه به یکدیگر می‌چسبند و تشکیل یک کلمه مرکب می‌دهند. تحلیل متن‌ها در صورتی که بتوانیم این کلمات را از مرکب‌بودن خارج کنیم، بسیار آسان‌تر و بهتر خواهد بود. در این روش با استفاده از میزان شباهت بخشی از کلمه مرکب با کلمه متناظر در زبان دیگر عمل جداسازی انجام می‌شود، سپس این روش بهبود داده می‌شود تا حتی زمانی که رابطه شناختی وجود ندارد هم جداسازی را انجام بدهد. ابتدا رابطه تناظر میان حروف و کلمات آلمانی و انگلیسی را در یک جدول نگه‌داری می‌کنیم. پس از آن کلمات نامرکب را به یکدیگر می‌چسبانیم و میزان شباهت آن را با کلمه مرکب ورودی در نظر می‌گیریم، اگر این شباهت از یک آستانه بیشتر شود، میزان شباهت زیررشته‌های مختلف کلمه مرکب را با هر یک از کلمات غیرمرکب می‌سنجیم تا نقطه جداسازی را بیابیم. هر یک از دو تیکه به دست آمده، ممکن است مجددا به صورت بازگشتی به الگوریتم جداسازی کلمات داده شوند. اگر دو کاراکتر یکسان باشند، وزن ۱ است و اگر کاراکتر ها یکسان نیستند اما مرتبط هستند وزن ۰/۹ است و نهایتا امتیاز ۰/۵ برای حالتی که دو کاراکتر نامرتبط هستند اما جد یکی از آن‌ها با کاراکتر دیگر مرتبط است.[۸]

- **طبقه‌بندی تقسیم‌بندی[۲]**

اگر یک پرس‌وجو از n توکن مختلف تشکیل شده باشد و بخواهیم آن را به جداسازی صحیح آن نگاشت کنیم، می‌توانیم از الگوریتم‌های یادگیری با نظارت استفاده کرده و با نمایش تعدادی جداسازی صحیح، عملکرد جداسازی را آموزش دهیم. در هنگام آموزش، پارامترهای یادگیری به گونه‌ای انتخاب می‌شوند که برای جداسازی صحیح مقدار بیشینه داشته باشند، سپس هنگام مواجهه با دادگان آزمون مقدار این پارامترها محاسبه می‌شود و آن کاندیدایی که بیشترین مقدار را کسب کند، مطابق با رابطه ۵، به عنوان جداسازی نهایی انتخاب می‌شود.

$$\hat{y} = argmax_y Score_w(x,y)$$ **رابطه ۵**

[۱] Corpus

[۲] Segmetation Classification

برای فهم ساختار طبقه‌بند می‌توان از ماشین‌های بردار پشتیبان[۱] استفاده کرد.

- **شبکه عصبی**

تقسیم‌کننده کلمات با استفاده از روش شبکه عصبی بازگشت‌کننده[۲] برای زبان تایلندی پیاده‌سازی شده است. این تقسیم‌کننده بر روی یک مجموعه داده ۵ میلیونی از کلمات تایلندی آموزش دیده است و هم‌اکنون در حال پیشرفت و توسعه می‌باشد و دارای دقت ۹۶٫۳٪ می‌باشد. از آن‌جایی که الگوریتم‌های به کار گرفته شده در زبان‌های خیلی متفاوت از انگلیسی همانند چینی،ژاپنی، تایلندی و ... مگر پس از ارزیابی و اصلاحات دقیق در زبان انگلیسی قابل استفاده نیستند، به همین میزان توضیح پیرامون این روش کفایت می‌کنیم.[۳]

- **روش تانگو[۴]**

یک الگوریتم آماری بدون نظارت است که نیازی به دیکشنری ندارد، از آن‌جایی که این روش در مقایسه با سایر روش‌های معرفی‌شده دقت بسیار پایین‌تری دارد، از ذکر جزئیات آن چشم‌پوشی خواهم کرد.[٤]

- **استفاده از چارچوب کاهش ریسک برای شکست کلمات**

در این روش چنانچه یک رشته حروف u و یک تابع ریسک R داشته باشیم، هدف الگوریتم این است که عبارت زیر را بیابد:

**رابطه ۶**

$$\hat{S} = argmin_s E[R(u|s)]$$

در رابطه فوق، s یک جداسازی از رشته بدون فاصله u می‌باشد که تابع ریسک را کمینه می‌کند. تابع ریسک به گونه ای تعریف می‌شود که اگر s یک جداسازی صحیح نباشد عدد یک و در

[۱] Support Vector Machine

[۲] Recurrent Neural Network

[۳] https://github.com/pucktada/cutkum accessed on January 3th 2019

[۴] Tango

غیر این صورت عدد صفر را برگرداند. برای پیدا کردن جواب بهینه‌ای که ریسک را کمینه می‌کند، می‌توانیم از تابع قانون تصمیم‌گیری MAP [۱] استفاده کنیم.

$$\hat{s} = argmax_{s\in\Omega} P(s|u) = argmax_{s\in\Omega} P(u|s)P(s)$$ **رابطه ۷**

فضای جست‌وجو $\Omega$ مجموعه تمامی رشته حروفی است که اگر فاصله‌های آن‌ها برداشته شود، رشته u را تولید می‌کنند. در واقعیت احتمالات p(s) و p(u|s) را نمی‌دانیم و باید تخمین بزنیم. این روش بسته به روش محاسبه احتمالات می‌تواند باناظر و یا بدون ناظر انجام شود.[۱]

## ۴-۲-۲- روش هموارسازی

یکی از بزرگترین مشکلاتی که در مواجهه با nحرفی ها با آن مواجه هستیم، کامل نبودن مجموعه دادگان است، به عبارت دیگر فرکانس رخداد بسیاری از کلمات صفر می‌شود تنها به این دلیل که در دادگان آموزش وجود نداشتند و نه به این معنا که در واقعیت وجود خارجی و کاربرد ندارد. اگر احتمال حضور این کلمات صفر در نظر گرفته شود حاصل کلی احتمال نیز صفر می‌شود در صورتی که این احتمال بدیهی است که صفر نمی‌باشد بلکه نقص دادگان آموزش منجر به رخ دادن این عارضه شده است. این مشکل اغلب به دلیل پراکندگی دادگان جمع‌اوری شده رخ می‌دهد. فقط به این خاطر که یک کلمه هرگز در هنگام آموزش دیده نشده است نمی‌توان نتیجه گرفت که هیچ‌گاه در دادگان آزمون هم رخ نمی‌دهد. سوال اصلی که منجر به ابداع روش‌های هموارسازی گردید این است که زمانی که فرکانس رخداد یک کلمه صفر است، احتمال آن باید چه عددی در نظر گرفته شود.

روش‌های هموارسازی تنها با اندکی تلاش عملکرد الگوریتم‌هایی که از nحرفی استفاده می‌کنند را بهبود می‌بخشند. حتی اگر مجموعه دادگان آموزش به حدی کامل باشد که فرکانس رخداد هیچ کلمه‌ای در دادگان آزمون صفر نشود، باز هم برای کسب دقت بیشتر اگر به مراتب بالاتر برویم، داده‌ها پراکندگی بیشتری پیدا می‌کنند و نیاز به هموارسازی خواهیم داشت.[۲]

[۱] Map Decision Rule

[۲] https://nlp.stanford.edu/~wcmac/papers/20050421-smoothing-tutorial.pdf accessed on January 5th 2019

در ادامه چند نمونه از معروف‌ترین روش‌های هموارسازی پرکاربرد در این حوزه معرفی می‌شوند:

- **هموارسازی افزودنی**

در این روش به تمامی فرکانس‌های موجود یک واحد اضافه می‌شود که در نتیجه آن فرکانس صفر در nحرفی وجود نخواهد داشت، این روش بسیار ساده می‌باشد و مشکل احتمالات صفر را نیز برطرف می‌کند اما به شدت به نتیجه‌گیری‌های خروجی جهت‌دهی می‌دهد و عملکرد الگوریتم را تحت شعاع قرار می‌دهد.[۱]

- **هموارسازی تعامل**[۲]

در این روش ابتدا یک پارامتر ثابت $\lambda$ در نظر می‌گیریم، انتخاب صحیح این پارامتر در عملکرد روش هموارسازی به شدت موثر است. این روش یک تعامل خطی از مدل ماکسیمم احتمال[۳] می‌باشد و پارامتر $\lambda$ میزان تاثیرگذاری هریک را مشخص می‌کند.

$$p_\lambda(w|d) = \lambda p_{ML}(w|d) + (1-\lambda)p(w|C) \qquad \text{رابطه ۸}$$

در رابطه ۸ پارامتر d کل سندی[۴] است که در آن به دنبال کلمه w می‌گردیم و پارامتر C کل مجموعه مدل زبانی است.[۱۱]

- **هموارسازی Witten and Bell**

این روش یک نمونه از روش هموارسازی تعامل می‌باشد. اگر یک نمونه ۲حرفی بسازیم که در آن زوج کلمه $w_{i-1}w_i$ هرگز رخ نداده باشد، فرکانس رخداد آن صفر خواهد بود و در نتیجه احتمال کلمه $w_i$ مشروط به رخداد کلمه $w_{i-1}$ پیش از آن صفر خواهد شد. یکی از مهم‌ترین چالش‌ها

۱ https://jeremykun.com/2012/01/15/word-segmentation/ accessed at August 15th 2018

۲ Jelinek–Mercer (interpolation)

۳ Maximum Likelihood Model

۴ Document

تخمین این احتمالات است به نحوی که خروجی صفر نشود. در روش Witten and Bell با توجه به تخمین Good-Turing احتمال اختصاص داده شده به عباراتی که فرکانس رخداد آن‌ها صفر است برابر با عبارت $\frac{N_1(w_{i-1})}{c_h(w_{i-1})}$ است که در آن صورت برابر با تعداد کلماتی است که دقیقا یک‌بار پس از کلمه $w_{i-1}$ رخ داده اند. این روش رابطه زیر را برای هموارسازی پیشنهاد می‌دهد:

$$P_{WB}(w_i|w_{i-۱}) = \lambda P_{MLE}(w_i|w_{i-۱}) + \frac{N_۱(w_{i-۱})}{c_h(w_{i-۱})} P_{backoff}(W_i)$$ **رابطه ۹**

پارامتر $\lambda$ به گونه‌ای انتخاب می‌شود که مجموع احتمال بالا به عدد ۱ برسد. از آن‌جایی که تعداد کلماتی که دقیقا یک‌بار پس از کلمه $w_{i-1}$ رخ داده‌اند نیز ممکن است در خیلی از مجموعه‌های دادگان برابر با صفر باشد، می‌توان $N_{1+}(w_{i-1})$ را محاسبه کرده که تعداد کلماتی است که یک‌بار یا بیشتر پس از کلمه $w_{i-1}$ رخ داده اند.[۹]

## ۳-۲- خلاصه و جمع بندی

در این فصل با مفاهیم اولیه و پیش‌زمینه‌هایی که جهت درک هرچه بهتر روش‌ها و مفاهیم موجود در حوزه پردازش زبان‌های طبیعی که در مسئله جداسازی کلمات بهم‌چسبیده به کار برده می‌شوند، آشنا شدیم. مفاهیم موجود در این فصل هر یک به اختصار توضیح داده شده‌اند و توصیه می‌شود برای درک و یادگیری عمیق‌تر به منابع معرفی شده مراجعه شود. در این فصل هنگامی که پیشینه پژوهش را بررسی کردیم، متوجه شدیم که جداسازی کلمات بدون فاصله کاربردهای گوناگونی دارد و در زمنیه‌های مختلف به کار گرفته می‌شود، هم چنین با روش‌های مختلفی برای جمع‌آوری دادگان آموزش، الگوریتم جداسازی و هموارسازی دادگان وجود دارند آشنا شدیم، به منظور انجام بهترین جداسازی، هریک از این روش‌ها می‌بایست با توجه به دامنه کاربرد جداسازی انتخاب شوند.

# فصل ۳:

# روش پیشنهادی حل مسئله

## ۳-۱- روش پیشنهادی

روش پیشنهادی برای جداسازی بخش‌های نام در اسامی بهم‌چسبیده، استفاده از مدل‌های زبانی برای انتخاب بهترین نامزد جداسازی می‌باشد. در این پژوهش تنها مدل‌های زبانی ۱حرفی و ۲حرفی پیاده‌سازی شده‌اند چرا که مدل‌های زبانی مرتبه‌های بالاتر، به دلیل پراکندگی بالا، عملکرد مناسبی ندارند و هم‌چنین حجم بسیار بالایی داشته و بارگیری آن‌ها بسیار زمان‌بر می‌باشد. با استفاده از مجموعه دادگان جمع‌آوری شده مجموعه دادگان ۱حرفی و ۲حرفی را می‌سازیم، سپس با استفاده از الگوریتم‌های جداسازی متکی بر nحرفی-ها، عمل جداسازی را انجام می‌دهیم. هدف نهایی این است که با دریافت یک نام بدون فاصله در رشته ورودی، بخش‌های نام به درستی از یکدیگر جدا شده و در خروجی نوشته شوند.

## ۳-۲- معرفی نرم‌افزارهای استفاده شده برای پیاده‌سازی پروژه

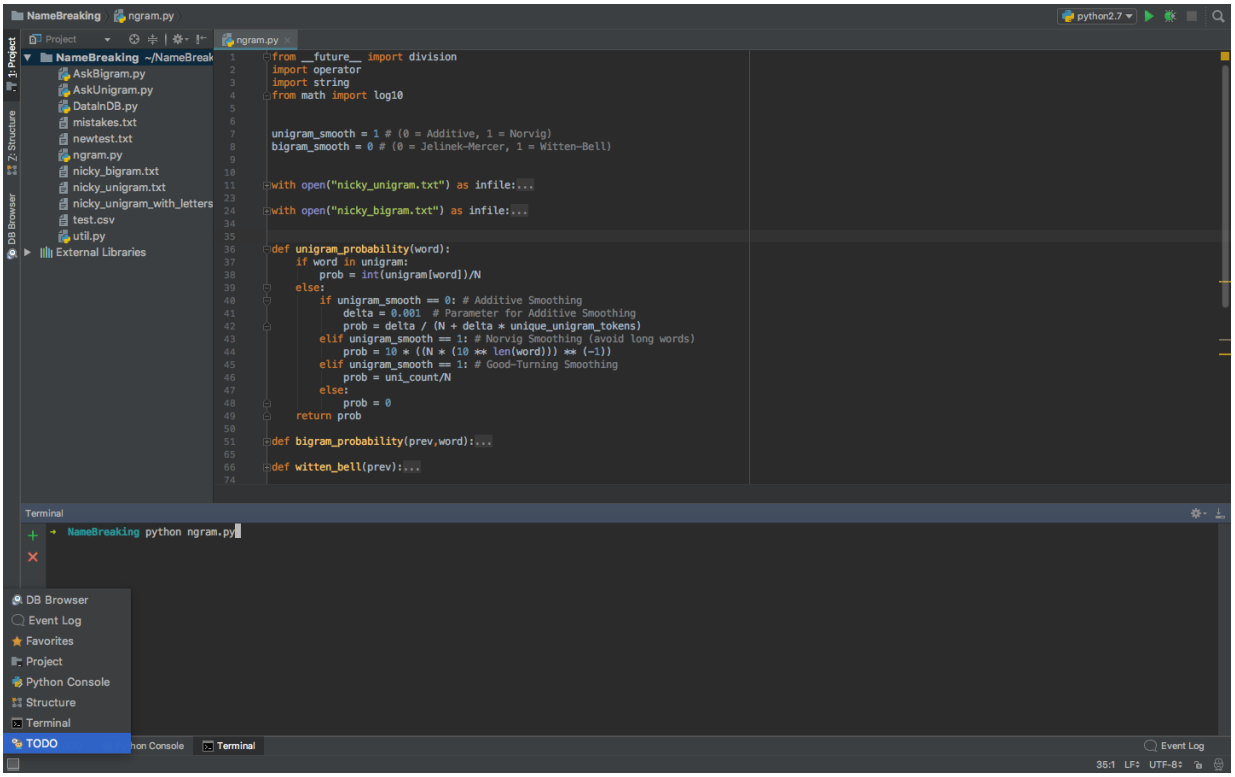
### ۳-۲-۱- زبان برنامه‌نویسی و ویرایشگر

در این پژوهش از زبان برنامه‌نویسی پایتون[۱] نسخه ۲٫۷ استفاده شده است .زبان پایتون یک زبان همه‌منظوره و بسیار منعطف است، کار کردن با این زبان ساده است و در بسیاری زمینه‌ها استفاده می‌شود. پایتون بسته‌ها و کتابخانه‌های فراوانی دارد و به طور خاص برای مقاصد تحلیل داده و هوش مصنوعی بسیار کارآمد است.


**شکل ۱ - نماد تجاری زبان برنامه‌نویسی پایتون**

[۱] http://python.org

در این پژوهش از ویرایشگر PyCharm استفاده شده است. PyCharm یک محیط توسعه یکپارچه برای برنامه نویسی به زبان پایتون در کامپیوتر است. این ویرایشگر توسط شرکت JetBrains در کشور چک توسعه یاقته است. با استفاده از این ویرایشگر می‌توانیم کدها را تحلیل و بررسی و اشکال‌زدایی نماییم و صحت برنامه‌ها اطمینان حاصل کنیم. این ویرایشگر در همه پلتفرم‌ها از جمله macOS، ویندوز و لینوکس نصب می‌شود. با استفاده از این ویرایشگر در هنگام نوشتن کد از خطاهای نوشتاری آگاه شده و هم‌چنین آن‌ها را اصلاح کنیم.

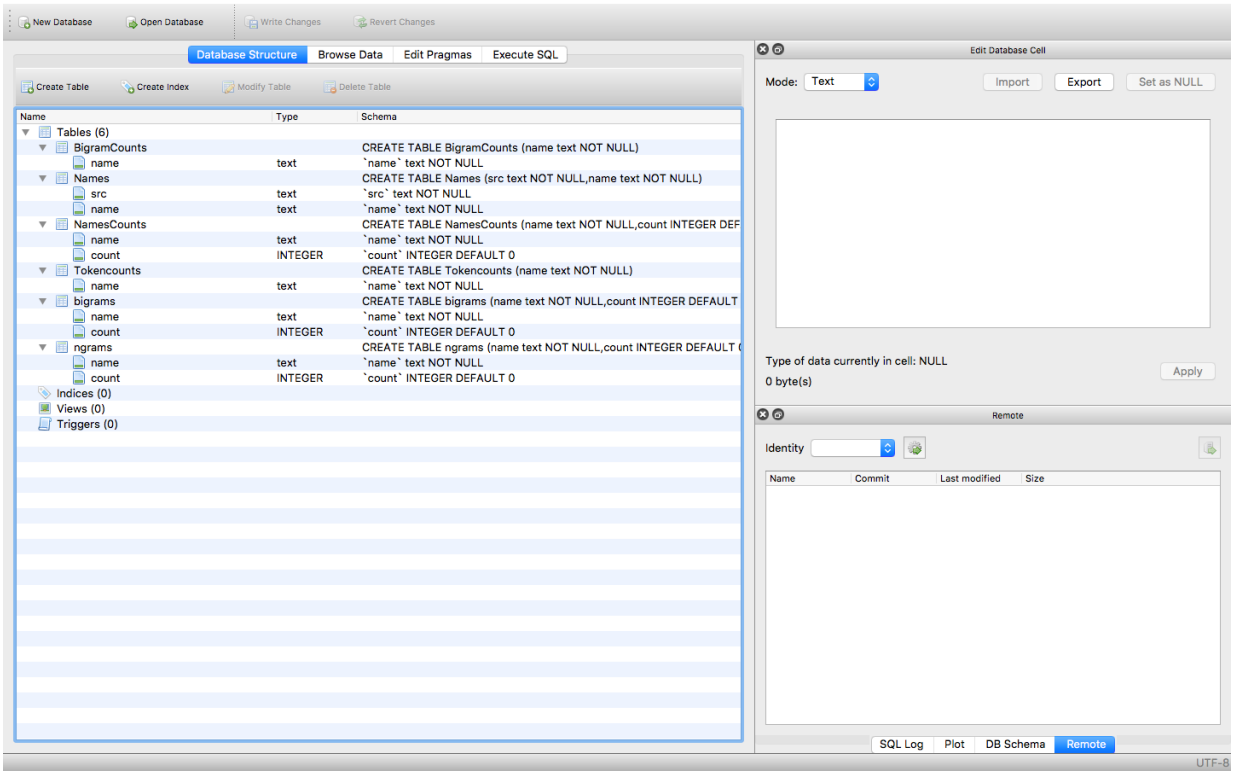



**شکل ۲ - شمایی از رابط کاربری نرم‌افزار pycharm**

## ۲-۲-۳- ابزار ارتباط با پایگاه داده

برای اجرای پرس‌وجوهای مربوط به پایگاه داده و ذخیره یا اضافه کردن اطلاعات به آن، نیازمند ابزاری برای ارتباط بهینه و موثر با پایگاه داده sqlite هستیم. برای این منظور از نرم‌افزار DB Browser for SQLITE استفاده شده است. این نرم‌افزار یک واسط کاربری برای مشاهده ساختار پایگاه داده، اجرای دستورات SQL، استخراج دادگان پایگاه داده و هم‌چنین اضافه کردن، حذف کردن، اصلاح جدول‌ها در اختیار کاربر قرار می‌دهد.

**شکل ۳ - نماد تجاری[۱] DB Browser for SQLITE**




**شکل ٤ - شمایی از رابط کاربری نرم‌افزار DB Browser for SQLITE**

[۱] http://sqlitebrowser.org

## ۳-۳- پیاده‌سازی

دو مدل زبانی ۱حرفی و ۲حرفی پیاده‌سازی شده اند. هم‌چنین روش‌های هموارسازی افزایشی[۱] و روش معرفی شده در کتاب Norvig [۱۷] برای هموار کردن الگوریتم ۱حرفی و روش‌های هموارسازی بازگشت به عقب[۲] و تعامل[۳] برای هموار کردن الگوریتم۲حرفی پیاده‌سازی شده‌اند.

### ۱-۳-۳- بارگیری داده‌ها[۴]

برای استفاده از دادگان ۱حرفی و ۲حرفی می‌بایست ابتدا فایل مربوط به آن‌ها را در برنامه بارگیری کنیم. بدین منظور به ازای هر خط از فایل ورودی، ابتدا قسمت‌های مختلف رشته نوشته شده در هر خط را که با استفاده از فاصله از هم جدا شده‌اند را با استفاده از تابع Split() جدا کرده و در یک لیست می‌نویسیم. سپس یک دیکشنری برای ۱حرفی تحت عنوان unigram و یک دیکشنری برای ۲حرفی تحت عنوان bigram تعریف کرده و هر توکن موجود در رشته ورودی را یک کلید این دیکشنری در نظر گرفته و فرکانس مربوط به آن مقدار متناظر با آن کلید خواهد بود. مقدار پیش‌فرض برای کلیدهایی که تعریف نشده‌اند صفر است. همزمان تعداد سطرهای فایل که برابر با تعداد توکن‌های یکتا در فایل است را محاسبه نموده و هم‌چنین با محاسبه مجموع فرکانس‌ها در فایل ورودی، تعداد کل توکن‌ها را به دست آورده و به ترتیب برای ۱حرفی و ۲حرفی در متغیرهای N و N2 ذخیره می‌کنیم.

---

[۱] Additive

[۲] Back off

[۳] Interpolation

[۴] Load data

```
with open("new_unigram.txt") as infile:
    unigram = {}
    unigram = defaultdict(lambda: 0, unigram)
    unique_unigram_tokens = 0   #num of unique tokens
    N = 0   #num of tokens
    uni_count = 0 #num of tokens with frequency 1
    for line in infile:
        token_freq = line.split()
        unigram[token_freq[0]] = token_freq[1]
        if int(token_freq[1]) == 1:
            uni_count += 1
        unique_unigram_tokens += 1
        N += int(token_freq[1])

with open("nicky_bigram.txt") as infile:
    bigram = {}
    bigram = defaultdict(lambda: 0, bigram)
    unique_bigram_tokens = 0    #num of unique tokens
    N2 = 0  #num of tokens
    for line in infile:
        token_freq = line.split()
        token = token_freq[0] + token_freq[1]
        bigram[token] = token_freq[2]
        unique_unigram_tokens += 1
        N2 += int(token_freq[2])
```

**شکل ۵- نحوه پیاده‌سازی بارگیری دیتا**

## ۲-۳-۳- محاسبه احتمالات ۱حرفی

برای محاسبه احتمال یک توکن در دیکشنری ۱حرفی از تابع unigram_probability استفاده می کنیم. این تابع به عنوان ورودی یک کلمه را می‌گیرد و سپس در خروجی احتمال آن را برمی‌گرداند. برای محاسبه احتمال کلمه ورودی، به دیکشنری ۱حرفی مراجعه کرده و چنانچه کلمه در دیکشنری موجود باشد، فرکانس رخداد آن بخش‌بر تعداد کل توکن‌های موجود در ۱حرفی را به عنوان خروجی برمی‌گرداند. اگر کلمه در دیکشنری موجود نباشد، با استفاده از پارامتر unigram_smooth نوع روش هموارسازی را انتخاب می‌کنیم، اگر این پارامتر برابر با صفر باشد، هموارسازی افزایشی انتخاب می‌شود. احتمال در این روش با استفاده از رابطه زیر به دست می‌آید:

$$probability = \frac{C(w_i)+\delta}{N+\delta \times C(UniqueTokens)}$$ **رابطه ۱۰**

چنانچه پارامتر unigram_smooth برابر با یک باشد، از روش هموارسازی معرفی شده در کتاب norvig

[17]استفاده خواهد شد. این روش در صورت حضور کلمه در دیکشنری فرکانس رخداد آن بخش‌بر تعداد کل توکن‌ها را به عنوان احتمال برمی‌گرداند و در صورت عدم حضور کلمه، از رابطه زیر احتمال را محاسبه می‌کند. در این رابطه از کلمات طولانی اجتناب می‌شود.

رابطه ۱۱

$$probability = \frac{۱۰}{N \times ۱۰^{len(word)}}$$

اگر پارامتر unigram_smooth برابر با ۲ انتخاب شود، هموارسازی نخواهیم داشت و در صورت عدم حضور کلمه در دیکشنری، مقدار احتمال صفر خواهد بود.

```
def unigram_probability(word):
    if unigram_smooth == 0: # Additive Smoothing
        delta = 0.001  # Parameter for Additive Smoothing
        prob = int(unigram[word])+delta/ (N+ delta *
        unique_unigram_tokens)
    elif unigram_smooth == 1: # Norvig Smoothing (avoid long words)
        if word in unigram:
            prob = int(unigram[word]) / N
        else:
            prob = 10 * ((N * (10 ** len(word))) ** (-1))
    else: # No Smoothing
        if word in unigram:
            prob = int(unigram[word]) / N
        else:
            prob = 0
    return prob
```

شکل ٦- نحوه محاسبه احتمال یک توکن در ۱حرفی

## ۳-۳-۳- جداسازی رشته ورودی با استفاده از مدل ۱حرفی

برای انجام عمل جداسازی ابتدا می‌بایست تمامی نامزدهای جداسازی یک رشته ورودی را به دست بیاوریم. برای یک رشته ورودی nحرفی، n-1 نقطه وجود دارد که می‌تواند محل قرارگیری فاصله باشد، به همین دلیل $۲^{n-۱}$ نامزد جداسازی خواهیم داشت که با استفاده از تابع candidates پیدا می‌شوند. سپس در تابع unigram_breaker به ازای تمامی کاندیدها مجموع لگاریتم احتمال تکه‌ها را به دست آورده و کاندیدایی که بیشترین امتیاز را کسب کند به عنوان بهترین جداسازی انتخاب می‌شود.

```
def candidates(name):
    splits = []
    for i in range(len(name)):
        splits.append([])
        splits[i].append(name[:i])
        splits[i].append(name[i:])
    return splits

def unigram_breaker(name):
    if not name:
        return ""
    splits = candidates(name)
    score = []
    for first,remaining in splits:
        score.append(log10(unigram_probability(first))+log10(
            unigram_probability(remaining)))
    index, value = max(enumerate(score), key=operator.itemgetter(1))
    return splits[index]
```

**شکل ۷ - نحوه جداسازی رشته ورودی با استفاده از مدل ۱حرفی**

## ۴-۳-۳- محاسبه احتمالات ۲حرفی

برای محاسبه احتمال یک جفت توکن در دیکشنری ۲حرفی از تابع bigram_probability استفاده می‌کنیم. این تابع به عنوان ورودی یک کلمه و کلمه پیشین آن را می‌گیرد و سپس در خروجی احتمال رخداد آن کلمه مشروط بر کلمه پیشین را برمی‌گرداند. برای محاسبه احتمال کلمه ورودی، به دیکشنری ۲حرفی مراجعه کرده و چنانچه کلمه در دیکشنری موجود باشد، فرکانس رخداد آن مشروط بر کلمه پیشین به عنوان خروجی برمی‌گرداند. اگر کلمه در دیکشنری موجود نباشد، با استفاده از پارامتر bigram_smooth نوع روش هموارسازی را انتخاب می‌کنیم، اگر این پارامتر برابر با یک باشد، هموارسازی تعامل انتخاب می‌شود. احتمال در این روش با استفاده از رابطه زیر به دست می‌آید:

$$probability = \lambda \times P(word|prev) + (1 - \lambda) \times P(word)$$ **رابطه ۱۲**

پارامتر $\lambda$ در رابطه فوق در این پژوهش بر اساس آزمون و خطا به نحوی انتخاب می‌شود که بهترین نتیجه حاصل شود اگرچه الگوریتم‌هایی برای انتخاب بهترین پارامتر نیز وجود دارند، به طور مثال در روش witten-bell پارامتر $\lambda$ با استفاده از روش زیر محاسبه می‌شود:

$$\lambda_{w_{i-1}} = 1 - \frac{u(w_{i-1})}{u(w_{i-1})+c(w_{i-1})}$$ **رابطه ۱۳**

در رابطه فوق $u(w_{i-1})$ تعداد کلمات یکتایی است که در دیکشنری دو حرفی پس از کلمه ورودی رخ داده‌اند.

در صورتی که پارامتر bigram_smooth مقدار صفر داشته باشد با استفاده از روش بازگشت به عقب چنانچه فرکانس رخداد یک جفت توکن در دو حرفی صفر باشد، احتمال ۱حرفی کلمه محاسبه می‌شود.

```
def bigram_probability(word, prev):
    # Back off
    if bigram_smooth == 0:
        try:
            prob = bigram[prev + ' ' + word] / float(unigram_probability(prev))
            return prob
        except KeyError:
            return unigram_probability(word)

    # Interpolation
    elif bigram_smooth == 1:
        landa2 = 0.9
        try:
            prob = ((landa2 * bigram[prev + " " + word]) / float(
                unigram_probability(prev))) + (
                            1 - landa2) * float(unigram_probability(word))
            return prob
        except KeyError:
            return (1 - landa2) * float(unigram_probability(word))
```

**شکل ۸- نحوه محاسبه احتمال یک زوج توکن در ۲حرفی**

## ۵-۳-۳- جداسازی رشته ورودی با استفاده از مدل ۲حرفی

برای انجام عمل جداسازی ابتدا می‌بایست تمامی کاندیدهای جداسازی رشته ورودی را با استفاده از تابع candidates به دست آوریم. سپس در تابع bigram_breaker به ازای تمامی کاندیدها مجموع لگاریتم احتمال تکه‌ها به ازای هر جفت تکه را به دست آورده و با هم جمع می‌کنیم. کاندیدایی که بیشترین امتیاز را کسب کند به عنوان بهترین جداسازی انتخاب می‌شود.

## ۳-۴- خلاصه و جمع‌بندی

فصل سوم به طور عمده در برگیرنده‌ی چگونگی پیاده‌سازی روش پیشنهادی برای حل مسئله جداسازی بخش‌های نام در اسامی بهم‌چسبیده می‌شود. مدل‌های زبانی ۱حرفی و ۲حرفی و روش‌های هموارسازی متعدد برای هر یک پیاده‌سازی شده‌اند. در فصل بعد نحوه جمع‌آوری دادگان استفاده شده برای آموزش و آزمون الگوریتم را توضیح خواهیم داد.

# فصل ۴:

# تولید مجموعه داده

در این فصل، نحوه جمع‌آوری دادگان آموزش و آزمون، الگوریتم پیاده‌سازی شده برای انجام عمل جداسازی و منابع استفاده شده و نحوه ذخیره دادگان در پایگاه داده معرفی می‌شود. در فصل بعد نتایج حاصل از اعمال الگوریتم پیاده‌سازی شده در فصل قبل بر روی دادگان این فصل تحلیل و بررسی خواهند شد.

## ۴-۱- مجموعه دادگان یادگیری

میزان تاثیرگذاری و کیفیت عملکرد روش‌های آماری پردازش زبان‌های طبیعی به شدت وابسته به اندازه مجموعه داده‌ای است که برای توسعه آن‌ها به کار رفته است. همانطور که مطالعات تجربی به تکرار نشان داده‌اند، الگوریتم‌های ساده‌تر، در بسیاری از کاربردها و حوزه‌های پردازش زبان‌های طبیعی، اغلب با در اختیار داشتن مجموعه دادگان عظیم، می توانند عملکرد بهتری از سایر جایگزین‌های پیچیده‌تر داشته باشند.[۱٤] بسیاری باور دارند که این اندازه مجموعه دادگان آموزش است و نه پیچیدگی الگوریتم استفاده شده که نقش اصلی را در پردازش زبان‌های طبیعی امروزی بازی می‌کند.[۱۷] همین امر موجب شده است که در این پژوهش عمده زمان و انرژی بر روی جمع‌آوری دادگان کامل‌تر و غنی‌تر گذاشته شده و اسامی بین‌المللی فراوانی جمع‌آوری شود. این مجموعه اسامی متعلق به جمع کثیری از کشورهای جهان بوده و شامل نام کوچک، نام میانی و نام‌خانوادگی می‌شود، البته نام میانی در برخی اسامی مانند اسامی ایرانی وجود ندارد. در مجموع ۱۲۲ میلیون نام کامل از وب جمع‌آوری و در پایگاه داده برای پردازش‌های آتی ذخیره شد. نحوه جمع‌آوری این اسامی به تفصیل در ادامه ذکر خواهد شد.

## ۱-۱-۴- مجموعه دادگان آموزش

- **دادگان پرونده‌های مرگ**[۱]

این مجموعه شامل اطلاعات افرادی است که از دنیا رفته‌اند، در گذشته این اطلاعات به صورت محرمانه در اختیار دولت‌ها قرار داشت و به همین دلیل وسیله‌ای برای توسعه صنعت مرگ تقلبی شده بود، اما هم‌اکنون به صورت عمومی در اختیار همگان قرار دارد تا هم مانع ادعای دروغین مرگ برخی افراد از جمله خلافکاران شود و هم به عنوان مجموعه دادگان آموزش در تحقیقات استفاده شود. سطرهای داده در این فایل بر اساس شماره امنیت اجتماعی[۲] مرتب شده‌اند. در مجموع تعداد ۸۷۳۲۹۷۲۵ نام و نام‌خانوادگی از این مجموعه داده به دادگان آموزش این پژوهش اضافه شد.[۳]

- **دادگان خانم‌های سیاه‌پوست آفریقایی آمریکایی**

این مجموعه شامل نام، نام خانوادگی، جنسیت و نژاد خانم‌های سیاه‌پوست آفریقایی آمریکایی می‌باشد که از سوابق عمومی زندانیان ایالات متحده جمع آوری شده است. تعداد رکورد های این مجموعه ۲۴۳۸ می‌باشد که به دادگان آموزش اضافه شده اند.[۴]

- **دادگان آقایان سیاه‌پوست آفریقایی آمریکایی**

این مجموعه شامل نام، نام خانوادگی، جنسیت و نژاد حدودا ۵۰۰۰۰ مرد سیاه‌پوست آفریقایی آمریکایی است که از سوابق عمومی زندانیان ایالت متحده به منظور تحلیل و بررسی‌های پردازش زبان‌های طبیعی جمع آوری شده است و تعداد ۳۵۰۸۱ رکورد از آن به دادگان آموزش افزوده شده است.[۵]

---

۱ Death Master File

۲ Social Security Number

۳ https://archive.org/details/DeathMasterFile accessed on January 5th 2019

۴ https://gist.github.com/mbejda/9dc89056005a689a6456 accessed on January 9th 2019

۵ https://gist.github.com/mbejda/61eb488cec271086632d accessed on January 9th 2019

- **دادگان آقایان سفیدپوست قفقازی**

دادگان شامل نام، نام خانوادگی، جنسیت و نژاد حدودا ۴۰۰۰۰ مرد سفیدپوست قفقازی است که از سوابق عمومی زندانیان ایالت متحده به منظور تحلیل و بررسی‌های پردازش زبان‌های طبیعی جمع آوری شده است و تعداد ۴۴۰۴۸ رکورد از آن به دادگان آموزش افزوده شده است.[۱]

- **دادگان خانم‌های سفیدپوست قفقازی**

دادگان شامل نام، نام خانوادگی، جنسیت و نژاد حدودا ۴۵۰۰ زن سفیدپوست قفقازی است که از سوابق عمومی زندانیان ایالت متحده به منظور تحلیل و بررسی‌های پردازش زبان‌های طبیعی جمع آوری شده است و تعداد ۴۶۰۰ رکورد از آن به دادگان آموزش افزوده شده است.[۲]

- **دادگان آقایان اسپانیایی**

دادگان شامل نام، نام خانوادگی، جنسیت و نژاد حدودا ۴۰۰۰ مرد اسپانیایی است که از سوابق عمومی زندانیان ایالت متحده به منظور تحلیل و بررسی‌های پردازش زبان‌های طبیعی جمع آوری شده است و تعداد ۴۱۶۶ رکورد از آن به دادگان آموزش افزوده شده است.[۳]

- **دادگان خانم‌های اسپانیایی**

دادگان شامل نام، نام خانوادگی، جنسیت و نژاد حدودا ۲۰۰ زن اسپانیایی است که از سوابق عمومی زندانیان ایالت متحده به منظور تحلیل و بررسی‌های پردازش زبان‌های طبیعی جمع آوری شده است و تعداد ۲۱۷ رکورد از آن به دادگان آموزش افزوده شده است.[۴]

- **دادگان آقایان هندی**

دادگان شامل نام، نام خانوادگی، جنسیت و نژاد حدودا ۱۴۰۰۰ مرد هندی است که از سوابق عمومی زندانیان ایالت متحده به منظور تحلیل و بررسی‌های پردازش زبان‌های طبیعی جمع آوری شده است و تعداد ۱۴۸۴۶ رکورد از آن به دادگان آموزش افزوده شده است.[۵]

۱ https://gist.github.com/mbejda/6c2293ba3333b7e76269 accessed on January 9th 2019

۲ https://gist.github.com/mbejda/26ad0574eda7fca78573 accessed on January 9th 2019

۳ https://gist.github.com/mbejda/21fbbfe24efd2a114800 accessed on January 9th 2019

۴ https://gist.github.com/mbejda/1e77ee4ad268916142a6 accessed on January 9th 2019

۵ https://gist.github.com/mbejda/7f86ca901fe41bc14a63 accessed at January 9th 2019

- **دادگان خانم‌های هندی**

دادگان شامل نام، نام خانوادگی، جنسیت و نژاد حدودا ۱۴۰۰۰ زن هندی است که از سوابق عمومی زندانیان ایالت متحده به منظور تحلیل و بررسی‌های پردازش زبان‌های طبیعی جمع آوری شده است و تعداد ۱۵۳۸۳ رکورد از آن به دادگان آموزش افزوده شده است.[۱]

- **دادگان اسامی اصولی**

این مجموعه شامل یک مشخص کننده یکتا برای هر فرد، نامی که فرد عمدتا با آن شناخته می‌شود، سال تولد، سال مرگ، حرفه اصلی و القاب و عناوینی که فرد با آن شناخته می‌شود، می‌باشد. در مجموع تعداد ۷۸۸۱۲۱۹ نام از این مجموعه به دادگان آموزش پژوهش افزوده شده است.[۲]

- **دادگان اطلاعات ثبت نام رأی دهندگان ایالت فلوریدا**[۳]

مجموعه فوق از سیستم ثبت‌نام رأی‌دهی ایالت فلوریدا ماهانه جمع‌آوری می‌شود و شامل اطلاعات رأی‌دهندگانی است که تا قبل از شروع ماه آتی، رسما ثبت‌نام و یا پیش ثبت‌نام کرده‌اند. اطلاعات کسانی که تقاضا کرده‌اند اطلاعاتشان محرمانه باقی بماند در این مجموعه موجود نمی‌باشد. هم‌چنین بخش غیررسمی دادگان از گزارش‌های مستقل ثبت‌شده از ۶۷ شهرستان ناظر بر انتخابات جمع‌آوری شده است که تاریخچه رأی را در یک برهه خاص زمانی ثبت کرده‌اند. در مجموع تعداد ۱۳۷۱۰۲۷۸ نام و نام‌خانوادگی از این مجموعه به دادگان آموزش اضافه شده است.[۴]

- **دادگان نژاد**[۵]

داده‌های نژادی فوق برای انجام تحقیقات محدود و مطالعات شخصی و تحصیلی در اختیار عموم قرار گرفته است و تنها شامل نام خانوادگی می‌باشد، در مجموع تعداد ۳۰۰۰۰۰ نام خانوادگی از این مجموعه به دادگان آموزش افزوده شدند.[۶]

---

۱ https://gist.github.com/mbejda/9b93c7545c9dd93060bd accessed on January 9th 2019

۲ https://datasets.imdbws.com/name.basics.tsv.gz accessed on January 9th 2019

۳ Florida Voter Registration Data

۴ https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/UBIG3F accessed on January 9th 2019

۵ http://www.ancestry.com/ accessed on January 9th 2019

۶ https://github.com/jeffsicdm14/name_pairs accessed on January 9th 2019

- **دادگان رشته دوم**[۱]

داده‌های گردآوری شده از این طریق، شامل ۱۴ مجموعه تک فیلد با استفاده از بسته‌بندی[۲] رشته دوم است که توسط آقای ویلیام کوئِن جمع‌آوری شده است. ۴۴ نام و نام‌خانوادگی با استفاده از این داده به مجموعه دادگان آموزش افزوده شدند.

- **دادگان کتاب‌شناسی علوم رایانه**[۳]

داده‌های گردآوری شده از مجموعه فوق با استفاده از کتاب‌شناسی علوم‌رایانه که یک مرجع آنلاین برای اطلاعات کتابشناسی در نشریات علمی کامپیوتر است تهیه شده است.کتاب‌شناسی dblp[۴] از یک وب سرور اولیه وب کوچک به یک سرویس اطلاعاتی محبوب برای جامعه علمی کامپیوتر تکامل یافته است. مأموریت این وب‌سایت این است که محققان علوم کامپیوتری را در تلاش های روزانه خود با ارائه دسترسی آزاد به متاداده های کتاب‌شناختی با کیفیت بالا و پیوندهایی به نسخه های الکترونیکی انتشارات حمایت کند و تا کنون شامل ۴،۳ میلیون مقاله که توسط ۲،۱ میلیون محقق منتشر شده‌اند، می‌باشد. آقای پاتریک رویتر[۵]با استفاده از این اطلاعات، یک مجموعه داده ارجاعات میان مقالات تهیه کرده است که در این پژوهش با استفاده از نام نویسندگان موجود در آن ۷۸۸۸۹۷ نام و نام خانوادگی به دادگان آموزش اضافه شد.[۶]

- **دادگان سوابق مخترع‌ها**

این مجموعه شامل شماره ثبت اختراع، نام و نام خانوادگی مخترع و آدرس وی شامل شهر، ایالت و کشور(در صورت وجود) می‌باشد، از آن‌جایی که یک فرد ممکن است چندین محصول یا روش مختلف را اختراع کرده باشد، در این مجموعه نام تکراری نیز مشاهده می‌شود. با استفاده از این اسامی ۴۳۰۱۲۲۹ رکورد به مجموعه آموزش تحقیق افزوده شد.[۷]

۱ Second String

۲ Package

۳ Computer Science Bibliography

۴ https://dblp.uni-trier.de/faq/What+is+dblp.html accessed on January 9th 2019

۵ Patrick Reuther

۶ http://www.cs.utexas.edu/users/ml/riddle/data.html accessed on January 9th 2019

۷ http://www.nber.org/patents/ accessed on January 9th 2019

- **دادگان citeseer**

این مجموعه یک خوشه‌بندی از مقالات است و هر سطر آن شامل شماره نویسنده مقاله، شماره خوشه‌ای که نویسنده به آن تعلق دارد، نام نویسنده و شماره نویسنده در مقاله، شماره مقاله و شماره خوشه‌ای که مقاله به آن تعلق دارد و عنوان مقاله می‌شود. با استفاده از این مجموعه ۲۸۹۲ نام به دادگان آموزش افزوده شده است.[۱]

- **دادگان ncvoter**

این مجموعه شامل اطلاعاتی از قبیل کد شهرستان، نوع و... در مورد انتخابات می‌باشد. از اسامی این مجموعه برای تکمیل مجموعه دادگان آموزش استفاده شده است که در مجموع ۸۱۲۶۱۷۱ نام و نام‌خانوادگی اضافه شد.[۲]

- **دادگان وابستگی میان نام و نژاد[۳]**

یک مجموعه داده جمع‌آوری‌شده توسط تیم استیون اسکینا[۴] با استفاده از ویکیپدیا[۵] می‌باشد که برای بخشی از پروژه‌ای در راستای طبقه بندی نژاد و قومیت بر اساس نام، تهیه شده است. بیش از ۱۴۰۰۰۰ نام و نژاد و وابستگی میان آن‌ها در این مجموعه موجود است. به طور دقیق، ۱۴۸۲۷۵ نام و نام‌خانوادگی از این مجموعه به دادگان آموزش اضافه شدند.[۶]

## ۲-۱-۴- مجموعه دادگان آزمون

- **دادگان حساب‌های کاربری مشهور در توییتر[۷]**

یک مجموعه داده برای تحقیقات جمع‌آوری شده است که در آن نام‌کاربری، دامنه وب‌سایت، نام و نام‌خانوادگی کامل و نوع حساب کاربری (فرد مشهور) ۱۰۰۰ نفر از مشهورترین

۱ https://linqs-data.soe.ucsc.edu/public/citeseer-mrdm05/ accessed on January 9th 2019

۲ https://dl.ncsbe.gov/index.html?prefix=data accessed on January 9th 2019

۳ Name Race Associations

۴ Steven Skiena

۵ Wikipedia

۶ https://github.com/appeler/ethnicolr/tree/master/ethnicolr/data/wiki accessed on January 9th 2019

۷ Twitter

کاربران توییتر در آن ذکر شده است. اسامی این هزار نفر جدا شده و به مجموعه دادگان آزمون این پژوهش اضافه شده است.[۱]

- **دادگان هنرمندان موسیقی**

این مجموعه داده شامل نام و نام خانوادگی و آدرس صفحه فیس‌بوک و آدرس صفحه توییتر ۱۰۰۰۰ خواننده معروف جهان می‌باشد. از ستون نام این مجموعه داده برای تکمیل دادگان آزمون تحقیق استفاده شده است.[۲]

- **دادگان شبکه اجتماعی آکادمیک**[۳]

مجموعه فوق شامل اطلاعات مقالات، ارجاعات مقالات، اطلاعات نویسندگان و همکاری میان نویسندگان است که ۲۰۹۲۳۵۶ مقاله و ۸۰۲۴۸۹۶ ارجاع میان آن‌ها را بررسی کرده است. در این دادگان ٦١٨١٩٠ نویسنده موجود است که اسامی آن‌ها به دادگان آزمون این پژوهش افزوده شده اند.[۴]

## ۲-۴- پیش‌پردازش مجموعه داده

پس از جمع‌آوری اسامی موجود در اینترنت، نیاز به انجام پیش‌پردازش‌هایی خواهیم داشت که استفاده از این دادگان را در الگوریتم مورد نظر ممکن سازد. از آن‌جایی که اسامی در مجموعه داده‌های مختلف به سبک‌های مختلف نوشته شده اند، ابتدا می‌بایست اطمینان حاصل شود که هر اسم، تنها از حروف الفبا ساخته شده است. بدین منظور یک رشته از علائم نگارشی تعریف کرده و در هر اسم، در صورت حضور هر یک از علائم نگارشی ممنوع، آن علامت حذف شده و به جای آن فاصله قرار می‌دهیم. به عنوان مثال اگر اسم "Roger Ten-Valls" در ورودی موجود باشد، خط فاصله یک علامت نگارشی است که می‌بایست حذف شود و سپس اسم مذکور به صورت " Roger Valls Ten" نوشته خواهد شد. سپس در مرحله بعد، برای حفظ یکپارچگی دادگان و بهبود دقت

۱ https://gist.github.com/mbejda/9c3353780270e7298763 accessed on January 9th 2019

۲ https://gist.github.com/mbejda/9912f7a366c62c1f296c accessed on January 9th 2019

۳ Academic Social Network

۴ https://aminer.org/data accessed on January 9th 2019

عملکرد الگوریتم، تمامی حروف اسم، تبدیل به حروف کوچک خواهند شد و چنانچه خطایی در داده از جمله حظور عدد وجود داشته باشد، آن داده حذف خواهد شد. هنگامی که یک اسم جدید برای آزمون به الگوریتم داده می‌شود، همین مراحل بر روی اسم جدید تکرار خواهد شد. از آن‌جایی که فرکانس رخداد حروف خیلی بالا و منجر به هدایت الگوریتم جداسازی به شکستن رشته‌های ورودی به حروف سازنده می‌شد، در این پژوهش فرکانس رخداد حروف در ۱حرفی را از رابطه ۱۴ به دست آوردیم.

$$Frequency(x) = round(\frac{C(x)}{N} \times 26 \times 100)$$ **رابطه ١٤**

## ۱-۲-۴- انتقال اطلاعات به پایگاه داده

در این پروژه به منظور ذخیره دادگان از پایگاه داده SQLITE استفاده شده است.SQLITE یک کتابخانه به زبان Cاست که یک موتور کوچک، سریع، خودمختار، قابل اعتماد و با ویژگی‌های فراوان پیاده‌سازی می‌کند. این پایگاه داده بسیار محبوب بوده و کاربران زیادی از سراسر دنیا دارد. این پایگاه داده متن‌باز بوده و کد مرجع آن به صورت رایگان در اختیار همگان قرار دارد و به همین دلیل در این پروژه از این پایگاه داده استفاده شده است. تمامی اسامی جمع‌آوری شده در این پایگاه داده ذخیره و سپس پردازش شده‌اند.

| Names | |
|---|---|
| name | text |
| src | text |
| rows | 122715171 |

| unigrams | |
|---|---|
| name | text |
| PK,FK2 | integer |
| rows | 5339514 |

| NamesCounts | |
|---|---|
| name | text |
| count | integer |
| rows | 64022352 |

| bigrams | |
|---|---|
| name | text |
| count | integer |
| rows | 41668548 |

**شکل ۹- جداول پایگاه داده**

### ۲-۲-۴- شمای پایگاه داده SQLITE

جدول Names شامل تمامی اسامی جمع‌آوری شده به همراه منبع نام می‌باشد. صفت name نام کامل فرد و صفت src منبع نام را مشخص می‌کند. این جدول در مجموع شامل ۱۲۲۷۱۵۱۷۱ نام می‌باشد که از این میان عده‌ای ممکن است تکراری باشند. با حذف داده‌های تکراری این جدول و اضافه کردن صفت countکه تعداد رخداد یک نام را مشخص می‌کند، جدول NamesCountsساخته می‌شود. این جدول ۶۴۰۲۲۳۵۲ سطر دارد که تعداد اسامی یکتا جمع‌آوری شده است.

جدول unigrams به ازای هر بخش نام، فرکانس رخداد آن در مجموعه داده را در صفت count ذخیره می‌کند. نام کوچک، نام میانی و نام خانوادگی از جمله بخش‌های نام می‌باشند که در مجموع ۲۱۵۳۳۸۴ بخش نام یکتا داریم.

جدول bigrams یک مجموعه دوتایی از بخش‌های نام را که در مجموعه داده پشت‌سرهم آمده‌اند را در صفت name ذخیره کرده و سپس در صفت count فرکانس رخداد آن را ذکر می‌کند. چنانچه یک بخش نام در ابتدای نام آمده باشد و توکن پیشین نداشته باشد، توکن <S> به عنوان توکن پیشین آن در نظر گرفته خواهد شد. جدول مذکور شامل ۴۱۶۶۸۵۴۸ زوج بخش نام و تعداد رخداد هر یک می‌باشد. برای محاسبه فرکانس رخداد برای هر یک از جداول، ابتدا با استفاده از دستور group by در SQL توکن‌های تکراری را در یک گروه قرار داده و سپس تعداد اعضای آن گروه را به عنوان فرکانس رخداد آن توکن یکتا در نظر می‌گیریم.

# فصل ۵:

# معیارهای ارزیابی، نتایج و تحلیل نتایج

الگوریتم یادگیری ابتدا با استفاده از مجموعه دادگان آموزش روند الگوریتم را یاد می گیرد، سپس یک مجموعه داده آزمون جمع‌آوری می‌شود که متفاوت از دادگان آموزش است و هم‌چنین پاسخ صحیح هر یک از سطرهای آن نیز موجود است. هدف در این مرحله بررسی عملکرد الگوریتم یادگیری می‌باشد، به همین دلیل هر یک از دادگان آزمون را به عنوان ورودی به ماشین می‌دهیم سپس خروجی حاصل را با خروجی صحیح مجموعه دادگان آزمون مقایسه می‌کنیم، چنانچه تعداد دفعاتی که نتیجه این مقایسه مثبت باشد (یعنی دو خروجی یکسان باشند) را تقسیم بر تعداد کل دادگان آزمون کنیم، دقت عملکرد الگوریتم یادگیری حاصل می‌شود. بدیهی است که در حالت ایده‌ال انتظار می‌رود این دقت ۱۰۰ درصد باشد، به عبارت دیگر، بهترین عملکرد الگوریتم یادگیری زمانی است که خروجی تمامی ورودی‌های مجموعه دادگان آزمون، مطابق با برچسب صحیح آن در مجموعه دادگان آزمون باشد.

در این فصل هدف ما این است که با استفاده از روش فوق، دقت عملکرد مدل‌های زبانی ۱حرفی و ۲حرفی را بر روی دادگان آزمون و با روش‌های هموارسازی مختلف مشاهده، مقایسه و ارزیابی کنیم.

## ۱-۵- آزمایشات و تحلیل نتایج

### ۱-۱-۵- آزمایش اول: جداسازی بر اساس مدل زبانی

مدل زبانی یک توزیع احتمالاتی بر روی کلمات و یا حروف زبان است و با استفاده از فرکانس رخداد توکن‌ها در یک متن آموزش، احتمال قرارگیری یک مجموعه توکن به صورت پشت سر هم در یک زبان را مشخص می‌کند. در این آزمایش عملیات جداسازی را بر روی دادگان آزمون که شامل ۶۱۸۱۹۰ نام بهم‌چسبیده و جداسازی صحیح هر یک است، با استفاده از مدل زبانی ۱حرفی و ۲حرفی انجام داده و نتایج را با یکدیگر مقایسه کرده و تحلیل می‌کنیم.

مدل ۱حرفی هنگام جداسازی رشته ورودی احتمالات تکه‌های مختلف را مستقل از هم در نظر گرفته و برای محاسبه امتیاز کاندیدا حاصلضرب احتمالات را به دست می‌آورد. دقت جداسازی مدل ۱حرفی بدون استفاده از روش هموارسازی، ۷۵% است.

سه علت برای عدم جداسازی صحیح برخی اسامی وجود دارد:

۱. برخی اسامی چندین جداسازی صحیح دارند و چنانچه جداسازی که فرکانس رخداد بالاتری دارد مطابق با برچسب جداسازی صحیح موجود در مجموعه آزمون نباشد، این جداسازی غلط شمرده می‌شود حال آن که الگوریتم با استفاده از دانش خود بهترین جداسازی صحیح را انتخاب کرده است. در این گونه موارد ایرادی بر عملکرد الگوریتم وارد نیست.
۲. در زمان جمع‌آوری توکن‌های یکتا از اسامی جمع‌آوری شده، فرکانس رخداد حروف الفبا به تنهایی بسیار زیاد بود و موجب جداسازی تمامی رشته‌های ورودی به حروف سازنده می‌شد. برای رویارویی با این مشکل، فرکانس رخداد حروف برابر با عددی متناسب با فرکانس حقیقی حرف اما به مراتب کوچکتر در نظر گرفته شده است، این روش اگر چه از صفر در نظر گرفتن فرکانس حروف به مراتب بهتر بود، در برخی موارد هنگام مواجهه با جداسازی‌هایی که نیاز به جدا کردن یک حرف به عنوان یک بخش از نام داشتند، دچار مشکل می‌شود چرا که فرکانس رخداد حقیقی آن حرف را در محاسبه احتمال نامزد جداسازی دخیل نمی‌کند.
۳. هنگام محاسبه احتمال رخداد یک توکن، فرکانس آن تقسیم بر مجموع توکن‌های موجود در ۱حرفی می‌شود. در زمان محاسبه امتیاز یک کاندید جداسازی، احتمال تکه‌ها در یکدیگر ضرب می‌شوند. مخرج امتیاز جداسازی همواره شامل تعداد کل توکن‌ها به توان تعداد تکه‌ها در جداسازی مورد نظر است. این موضوع باعث می‌شود الگوریتم تمایل داشته باشد جداسازی‌هایی که تکه‌های کمتری دارند را انتخاب کند. به عبارت دیگر تعداد کل توکن‌ها در جداسازی یک رشته ورودی تاثیر مستقیم دارد. برای روشن‌تر شدن موضوع به مثال زیر توجه کنید:

**مثال:** می‌خواهیم رشته ورودی "HongJiang" را به بخش‌های "Hong" و "Jiang" بشکنیم. ابتدا فرکانس رخداد هر یک در مجموعه ۱حرفی را به دست می‌آوریم سپس احتمالات را محاسبه کرده و امتیاز جداسازی را محاسبه می‌کنیم.

C(“HongJiang”) = 63  C(“Jiang”) = 3972  C(“Hong”) = 14451

P(“Hong Jiang”) = P(“Hong”) * P(”Jiang”) = C(“Hong”)/N * C(“Jiang”)/N  **رابطه ۱۵**

از طرفی احتمال یکی دیگر از نامزدهای جداسازی به روش زیر محاسبه می‌شود:

P(“HongJiang”) = C(“HongJiang”)/N  **رابطه ۱۶**

همانطور که در مثال فوق مشاهده می‌کنید، توکن‌های "Hong" و "Jiang" فرکانس بسیار بیشتری از توکن "HongJiang" دارند اما به دلیل حضور N به توان دو در مخرج کاندید اول، کاندید دوم به عنوان بهترین جداسازی انتخاب می‌شود.

در جداسازی با استفاده از مدل ۲حرفی احتمال توکن‌ها وابسته به توکن پیشین آن‌ها می‌باشد. این مدل به وابستگی‌های متنی توجه بیشتری دارد اما دادگان آن بسیار پراکنده می‌باشد و احتمال رخداد جفت توکنی که در مجموعه ۲حرفی فرکانس صفر داشته باشند زیاد است. به همین دلیل در مدل ۲حرفی در صورت مواجهه با فرکانس صفر، یک قدم به عقب رفته و احتمال ۱حرفی را محاسبه می‌کنیم. با استفاده از مدل ۲حرفی دقت جداسازی ۷۲٫۷% است.

علت کاهش دقت نسبت به مدل ۱حرفی پراکندگی داده در این مدل می‌باشد. برای مقایسه عملکرد مدل ۱حرفی و ۲حرفی و علت جداسازی غلط برخی اسامی در مدل ۲حرفی به مثال زیر که در آن نام در مدل ۱حرفی به درستی جدا شده و در مدل ۲حرفی دارای خطای جداسازی است، توجه کنید:

رشته ورودی: erzenhyko

جداسازی صحیح مدل ۱حرفی: erzen hyko

جداسازی مدل ۲حرفی: erzen hy ko

- جداسازی توسط مدل ۱حرفی

c(“erzen”) = 25
c(“hyko”) = 0

با استفاده از روش هموارسازی کتاب norvig، احتمال hyko برابر با مقدار زیر خواهد بود:

$$probability = \frac{10}{N\times10^4} = \frac{10}{71507429\times10^4} = \frac{1}{71507429\times10^3}$$ **رابطه ۱۷**

امتیاز erzen hyko که بهترین جداسازی از میان نامزدهای ممکن است، برابر است با:

$$score = \frac{25}{71507429} \times 10^{-3} = 34 \times 10^{-11}$$

- جداسازی توسط مدل ۲حرفی

ابتدا می‌بایست احتمال hyko به شرط erzen را محاسبه کنیم، از آن‌جایی که جفت erzen hyko در ۲حرفی جمع‌آوری شده وجود ندارد، احتمال hyko در مدل ۱حرفی را به دست می‌آوریم. از آن جایی که فرکانس رخداد hyko در مدل ۱حرفی صفر است، پس با استفاده از روش هموارسازی احتمال آن را مطابق رابطه ۱۷ محاسبه می‌کنیم.
دلیل انتخاب جداسازی "erzen hy ko" توسط مدل ۲حرفی این است که جفت "erzen hy" در مدل ۲حرفی موجود نیست و باید احتمال "hy" در مدل ۱حرفی محاسبه شود که فرکانس رخداد آن ۱۸۵ است. از طرف دیگر جفت "hy ko" نیز در مدل ۲حرفی موجود نیست و باید احتمال "ko" در مدل ۱حرفی محاسبه شود و فرکانس رخداد "ko" ۲۷۵۸ است. همین امر موجب می‌شود "erzen hy ko" به اشتباه به عنوان نامزد جداسازی برتر انتخاب شود.

## ۲-۱-۵- آزمایش دوم: ارزیابی روش‌های هموارسازی

در این آزمایش عملکرد روش‌های هموارسازی مختلف در دو مدل زبانی ۱حرفی و ۲حرفی با یکدیگر مقایسه و ارزیابی می‌شوند.

دقت عملکرد هموارسازی افزایشی در مدل زبانی ۱حرفی با پارامترهای مختلف بررسی شده و در جدول شماره ۱ قابل مشاهده است.

جدول ۱ - محاسبه دقت جداسازی در پارامترهای مختلف هموارسازی‌های افزایشی

| **پارامتر دلتا** | ۰٫۹ | ۰٫۱ | ۰٫۰۵ | ۰٫۰۱ | ۰٫۰۰۱ | ۰٫۰۰۰۱ |
|---|---|---|---|---|---|---|
| **دقت جداسازی** | ۷۰٫۸% | ۷۵% | ۷۵٫۸% | ۷۶٫۸% | ۷۷٫۵% | ۷۷٫۶% |

در مدل ۱حرفی چنانچه از روش هموارسازی معرفی شده در کتاب norvig که در فصل سوم نحوه پیاده‌سازی آن را توضیح داده استفاده شود، دقت جداسازی ۷۹٫۴۸% خواهد بود که بهترین نتیجه جداسازی‌ها از میان روش-های مختلف این پژوهش است.

در جدول ۲ دقت روش‌های هموارسازی پیاده‌سازی شده برای مدل ۱حرفی در این پژوهش را مشاهده می‌کنید:

جدول ۲- مقایسه دقت هموارسازی های مختلف در ۱حرفی

| روش هموارسازی | دقت جداسازی |
|---|---|
| بدون هموارسازی | ۷۵% |
| هموارسازی افزایشی | ۷۷٫۶% |
| هموارسازی norvig | ۷۹٫۴۸% |

در مدل ۲حرفی ابتدا با استفاده از روش بازگشت به عقب عمل جداسازی را انجام دادیم . دقت جداسازی ۷۲٫۷% حاصل شد. سپس با استفاده از روش هموارسازی تعامل که در فصل سوم نحوه پیاده‌سازی آن را توضیح دادیم، اسامی بهم‌چسبیده ورودی را جدا کردیم. جدول ۳ دقت جداسازی مدل ۲حرفی با استفاده از روش هموارسازی تعامل را با پارامترهای لاندا متفاوت نشان می‌دهد. همانطور که مشاهده می‌کنید دقت جداسازی در این روش با تغییر پارامتر لاندا تغییر زیادی نمی‌کند و دقت جداسازی بسیار پایین است.

جدول ۳- محاسبه دقت جداسازی برای پارامترهای مختلف هموارسازی تعامل

| **پارامتر لاندا** | ۰٫۹۵ | ۰٫۹ | ۰٫۵ | ۰٫۱ |
|---|---|---|---|---|
| **دقت جداسازی** | ۴۷٫۷% | ۴۷% | ۴۶٫۷% | ۴۲٫۲% |

در مثال زیر علت تفاوت چشم‌گیر جداسازی توسط روش بازگشت به عقب و روش هموارسازی تعامل شرح داده خواهد شد. این مثال توسط روش بازگشت به عقب به درستی جدا شده اما در روش تعامل دچار خطا می‌شود. پارامتر لاندا در این مثال عدد ۰٫۹ در نظر گرفته خواهد شد.

رشته ورودی: wenhuwu

جداسازی صحیح توسط مدل ۲حرفی بازگشت به عقب: wenhu wu

جداسازی مدل ۲حرفی تعامل: wen hu wu

- بررسی نحوه جداسازی روش بازگشت به عقب

$$SCORE(wenhu\ wu) = \frac{C(<S>wenhu)}{N_2} \times \frac{C(wenhu\ wu)}{N_2} = \frac{11}{N_2} \times \frac{1}{N_2}$$ **رابطه ۱۸**

- بررسی نحوه جداسازی روش تعامل

محاسبه احتمال جداسازی نامزد wenhu wu از روش زیر انجام می‌پذیرد:

$$SCORE(wenhu\ wu) = \frac{0.9\times C(<S>\ wenhu)}{N_2} + \frac{0.1\times C(wenhu)}{N_1} \times \frac{0.9\times C(wenhu\ wu)}{N_2} + \frac{0.1\times C(wu)}{N_1}$$ **رابطه ۱۹**

$$SCORE(wenhu\ wu) = \frac{0.9 \times 11}{N_2} + \frac{0.1 \times 23}{N_1} \times \frac{0.9 \times 1}{N_2} + \frac{0.1 \times 13437}{N_1}$$

محاسبه احتمال جداسازی نامزد wen hu wu از روش زیر انجام می‌پذیرد:

$$SCORE(wen\ hu\ wu) = \frac{0.9\times C(<S>\ wen)}{N_2} + \frac{0.1\times C(wen)}{N_1} \times \frac{0.9\times C(<wen\ hu)}{N_2} + \frac{0.1\times C(hu)}{N_1} \times \frac{0.9\times C(hu\ wu)}{N_2} + \frac{0.1\times C(wu)}{N_1}$$ **رابطه ۲۰**

$$SCORE(wen\ hu\ wu) = \frac{0.9\times 747}{N_2} + \frac{0.1\times 6491}{N_1} \times \frac{0.9\times 4}{N_2} + \frac{0.1\times 4498}{N_1} \times \frac{0.9\times 1}{N_2} + \frac{0.1\times 13437}{N_1}$$

از مقایسه روابط ۱۹ و ۲۰ درمی‌یابیم که روش هموارسازی تعامل به اشتباه جداسازی wen hu wu را انتخاب می‌کند. روش‌های هموارسازی بازگشت به عقب برای مواردی که اندازه داده بزرگ باشد بهتر عمل می‌کنند این درحالی است که روش‌های تعامل برای داده‌های کوچک بهینه هستند[۱] و این موضوع در نتایج آزمایش نیز کاملا مشهود است.

## ۲-۵- جمع‌بندی

در این فصل با نحوه ارزیابی عملکرد الگوریتم جداسازی برای مدل‌های ۱حرفی و ۲حرفی و نتایج حاصل از پیاده‌سازی هر یک آشنا شدیم. سپس نتایج را تحلیل کرده و علت جداسازی‌های غلط را بررسی کردیم. سپس

۱ https://nlp.stanford.edu/~wcmac/papers/20050421-smoothing-tutorial.pdf

روش‌های هموارسازی مختلف را بررسی کرده و برای هر یک پارامتر بهینه را پیدا کردیم.

# فصل ۶:

# نتیجه‌گیری و کارهای آتی

## ۶-۱- نتیجه‌گیری

مسأله جداسازی کلمات بهم‌چسبیده از جمله مباحث حائز اهمیت در حوزه پردازش زبان‌های طبیعی محسوب شده و مقالات متعددی پیرامون روش‌های متنوع انجام عمل جداسازی برای مقاصد گوناگون در دهه‌های اخیر منتشر شده‌اند. مقالات پیشین در زمینه‌های نگاشت پرس‌و-جو کاربران به دامنه‌های اینترنتی، تشخیص دامنه‌های مخرب، ترجمه ماشینی، تشخیص گفتار و... با استفاده از روش‌های متنوع سعی در جداکردن کلمات بدون فاصله به بخش‌های معنادار داشته-اند. همانطور که پیش‌تر مطرح کردیم، حذف فاصله میان بخش‌های مختلف نام یک خطای رایج است. در این پژوهش هدف ما جداسازی بخش‌های نام در نام‌های بهم‌چسبیده بود که از جمله کاربردهای آن تطبیق موجودیت است. روش این پژوهش برای جداسازی بخش‌های نام استفاده از مدل‌های زبانی ۱حرفی و ۲حرفی بود. در این روش، امتیازدهی به نامزدهای مختلف جداسازی با استفاده از احتمال رخداد آن جداسازی به‌خصوص در مدل زبانی انجام شد. جداسازی به صورت خاص بر روی نام در این پژوهش برای اولین بار انجام شده است.

روش جداسازی این پژوهش باناظر بود، پس به منظور ارزیابی عملکرد الگوریتم جداسازی نیازمند مجموعه دادگان آموزش و آزمون بودیم که با استفاده از نام‌های موجود در مجموعه داده‌های متعدد موجود در اینترنت بیش از ۱۲۲ میلیون نام کامل را استخراج کردیم. الگوریتم جداسازی نام بر روی دادگان آزمون با استفاده از مدل ۱حرفی و روش هموارسازی معرفی شده در کتاب norvig با دقت ۷۹٫۸٤% عمل کرد که بهترین درصد درستی به دست آمده در این پژوهش است.

## ۶-۲- کارهای آتی

### ۶-۲-۱- جمع‌آوری داده و بهبود فایل‌های ۱حرفی و ۲حرفی

جمع‌آوری مجموعه دادگان بیشتر و کامل‌تر به بهبود عملکرد الگوریتم جداسازی کمک شایانی می‌کند. هرچقدر ۱حرفی و ۲حرفی کامل‌تر باشند، در هنگام ارزیابی عملکرد الگوریتم،

احتمال مواجه شدن با بخش‌های نامی که در دادگان آموزش وجود ندارند، کمتر می‌شود. هنگامی که با یک توکن روبه‌رو می‌شویم که در مجموعه دادگان آموزش وجود نداشته است، احتمال رخداد آن صفر در نظر گرفته می‌شود که با ضرب کردن آن در سایر احتمالات، امتیاز آن جداسازی به‌خصوص صفر خواهد شد. برای مقابله با این مشکل از روش‌های هموارسازی استفاده می‌کنیم تا این احتمالات را مقداری مخالف صفر گزارش کنیم. روش‌های هموارسازی اگرچه عملکرد را بهبود می‌بخشند، هرگز به خوبی زمانی که توکن مورد نظر در دادگان آموزش وجود داشته باشد نخواهند بود، به همین دلیل جمع‌آوری اسامی بیشتر برای تشکیل ۱حرفی کامل‌تر همواره به بهبود عملکرد این پژوهش کمک خواهد کرد.

### ۶-۲-۲- کاهش تاثیر کلمات غیر مرتبط در محاسبه امتیاز یک کاندید

یکی از مشکلات پیش‌رو هنگام محاسبه امتیاز یک کاندید جداسازی، تاثیر بیش‌ازحد تعداد توکن‌های موجود در ۱حرفی در عمل جداسازی است. هنگام محاسبه احتمال یک توکن، فرکانس آن تقسیم بر تعداد کل توکن‌های آموزش می‌شود. این موضوع موجب می‌شود الگوریتم جداسازی به سمت جداسازی با تعداد تکه‌های کمتر جهت‌گیری کند و این سیاست در برخی موارد منجر به هدایت الگوریتم به سمت انتخاب یک جداسازی غلط می‌شود.

### ۶-۲-۳- بهبود روش‌های هموارسازی

روش‌های هموارسازی در مدل‌های زبانی بسیار موثر هستند به این دلیل که هرگز دادگان آموزش کامل نخواهد بود و همواره احتمال مواجه شدن با توکن جدید در زمان آزمون وجود دارد. بهبود این روش‌ها عملکرد الگوریتم را در این شرایط به میزان قابل توجهی ارتقا می‌دهد. در این پژوهش روش‌های هموارسازی متعددی بررسی شده‌اند اما هم‌چنان روش‌هایی وجود دارند که نیاز به بررسی بیشتر دارند. از سوی دیگر علاوه بر روش‌های هموارسازی معروف، می‌توان با تفکر و خلاقیت مبدع روش‌هایی باشیم که در حوزه به خصوص جداسازی نام خوب عمل می‌کنند.

### ۴-۲-۶- پیاده‌سازی روش‌های مقالات پیشین

در فصل دوم روش‌های پیشین را از بعد پیاده‌سازی طبقه‌بندی کردیم. در این پژوهش تنها یکی از روش‌های موجود پیاده‌سازی شده است. با اجرای الگوریتم‌های دیگر بر روی دادگان اسامی جمع‌آوری شده می‌توانیم به بهبود دقت جداسازی بخش‌های نام در اسامی بهم‌چسبیده امیدوار باشیم.

### ۵-۲-۶- مشخص کردن فرکانس رخداد حروف الفبا

پس از جمع آوری داده و مشخص کردن توکن‌های یکتا و فرکانس رخداد هر یک، متوجه می‌شویم که حروف الفبا انگلیسی فرکانس بسیار زیادی دارند، این موضوع موجب می‌شود هنگام انجام عمل جداسازی شکستن رشته ورودی به تمام حروف سازنده بیش‌ترین امتیاز را کسب کند. در این پژوهش برای مقابله با این مشکل فرکانس رخداد حروف الفبا برابر با نسبت فرکانس به تعداد کل توکن‌ها ضربدر تعداد کل حروف الفبا به دست آوردیم. این کار مشکل پیشین را برطرف نمود اما در مواردی که نیاز به جداسازی یک حرف یکتا در رشته داریم، جداسازی ممکن است اشتباه انجام شود. در پژوهش‌های آتی می‌بایست روش بهتری برای تعیین مناسب‌ترین مقدار برای فرکانس حروف الفبا انگلیسی انتخاب شود.

# فصل ۷:

# منابع

# ۱-۷- مراجع


۱. Wang, K., C. Thrasher, and B.-J.P. Hsu. *Web scale nlp: a case study on url word breaking*. in *Proceedings of the ۲۰th international conference on World wide web*. ۲۰۱۱. ACM.

۲. Bergsma, S. and Q.I. Wang. *Learning noun phrase query segmentation*. in *Proceedings of the ۲۰۰۷Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. ۲۰۰۷.

۳. Hafer, M.A. and S.F. Weiss, *Word segmentation by letter successor varieties.* Information storage and retrieval, ۱۹۷۴. ۱۰(۱۱-۱۲): p. ۳۷۱-۳۸۵.

۴. Esch, M., et al., *A query suggestion workflow for life science IR-systems.* J Integr Bioinform, ۲۰۱۴. ۱۱(۲): p. ۲۳۷.

۵. Alfonseca, E., S. Bilac, and S. Pharies. *Decompounding query keywords from compounding languages*. in *Proceedings of the ۴۶th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*. ۲۰۰۸. Association for Computational Linguistics.

۶. Macherey, K., et al. *Language-independent compound splitting with morphological operations*. in *Proceedings of the ۴۹th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume ۱*. ۲۰۱۱. Association for Computational Linguistics.

۷. Koehn, P. and K. Knight. *Empirical methods for compound splitting*. in *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume ۱*. ۲۰۰۳. Association for Computational Linguistics.

۸. Brown, R.D. *Corpus-driven splitting of compound words*. in *Proceedings of the ۹th International Conference on Theoretical and Methodological Issues in Machine Translation*. ۲۰۰۲.

۹. Srinivasan, S., S. Bhattacharya, and R. Chakraborty. *Segmenting web-domains and hashtags using length specific models*. in *Proceedings of the ۲۱st ACM international conference on Information and knowledge management*. ۲۰۱۲. ACM.

۱۰. Salvetti, F. and N. Nicolov. *Weblog classification for fast splog filtering: A url language model segmentation approach*. in *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*. ۲۰۰۶. Association for Computational Linguistics.

۱۱. Raghuram, J., D.J. Miller, and G. Kesidis, *Unsupervised, low latency anomaly detection of algorithmically generated domain names by generative probabilistic modeling.* J Adv Res, ۲۰۱۴. ۵(۴): p. ۴۲۳-۳۳.

۱۲. Brent, M.R., *An efficient, probabilistically sound algorithm for segmentation and word discovery.* Machine Learning, ۱۹۹۹. ۳۴(۱-۳): p. ۷۱-۱۰۵.

۱۳. Venkataraman, A.J.C.L., *A statistical model for word discovery in transcribed speech.* ۲۰۰۱. ۲۷(۳): p. ۳۵۱-۳۷۲.

.۱۴ Wang, K., et al. *An overview of Microsoft Web N-gram corpus and applications*. in *Proceedings of the NAACL HLT ۲۰۱۰Demonstration Session*. ۲۰۱۰. Association for Computational Linguistics.

.۱۵ Goldwater, S., T.L. Griffiths, and M. Johnson. *Contextual dependencies in unsupervised word segmentation*. in *Proceedings of the ۲۱st International Conference on Computational Linguistics and the ۴۴th annual meeting of the Association for Computational Linguistics*. ۲۰۰۶. Association for Computational Linguistics.

.۱۶ Chi, C.-H., C. Ding, and A. Lim. *Word segmentation and recognition for web document framework*. in *Proceedings of the eighth international conference on Information and knowledge management*. ۱۹۹۹. ACM.

.۱۷ Norvig, P.J.B.D., *Natural language corpus data.* ۲۰۰۹: p. ۲۱۹-۲۴۲.

## Abstract:

The problem of word breaking has attracted many scholars to study this field for last decades and is one of the most important topics in the field of natural language processing. Word breaking is useful in mapping of Internet domains to a query, detection of malicious domains, machine translation, speech recognition, correction of written errors, etc. One of the noises that names suffer from is the removal of the spaces between parts of name. Names without space are names that the space between different parts of them is removed. Different parts of name consist of first name, middle name, last name, etc. Our goal in this project is to break names without space into constructive parts.

Breaking parts of name in names without space is useful in a variety of fields, including information retrieval and entity resolution. In the past, methods based on language models have been used in word breaking, the approach of this study is also using unigram and bigram language models. Breaking on a name-specific basis is done in this research for the first time.

Since the approach used is supervised, to evaluate the performance of the breaking algorithm, train and test datasets are required. The data used in this research is extracted from names in the numerous data collections provided to researchers for free on the Internet. Overall, more that 120 million names from different languages have been collected. Unique tokens in the dataset and the frequency of each occurrence is calculated to produce unigram and bigrams used in this project. Our breaking algorithm performs well with a high accuracy on the test dataset.

**Keywords:**

**Natural Language Processing, Word Breaking, Word Segmentation**

**University of Tehran**

**College of Engineering**
**School of Electrical and Computer Engineering**

**Thesis Title**

# Breaking Parts of Name in Names without Space

**A thesis submitted to the Undergraduate Studies Office**
**In partial fulfillment of the requirements for**
**The degree of Bachelor of Science in**
**Software Engineering**

**By:**
**Nicky Bayat**
**Supervisor:**
**Dr. Masoud Asadpoor**
**February 2019**