به نام خدا

جداسازی کلمات و علائم از متن و تعیین تعداد تکرار آنها بوسیلهٔ DFS و n-gramها

استاد: آقای دکتر فیضی درخشی

دانشجو: امیر محسن یوسفی واقف

بهار ۹۰

فهرست:

مقدمه:

در این گزارش قصد آشنایی کلی با نرم افزار موجود را داریم که در پیرامون این موضوع ما ابتدا بررسی کلی پیرامون DFAها و n-gramها را داریم که بوسیله آنها می خواهیم متن را از فایل HTML جدا نموده و سپس توکن ها و جملات آنرا از متن پالایش شده مشخص نمائیم، سپس bigram و trigramهای آن متن را پیدا نموده و نیز تعداد آنها را یافته تا بعداً برای پیشگویی کلمات و جایگاه آنها از این گرامها استفاده شود. حال با مقدمه ای درباره نظریه زبانها و توضیحاتی پیرامون ماشینهای متناهی قطعی و n-gramها به روند پیاده سازی الگوریتم رسیده و یا نمایی از DFA این نرم افزار این گزارش را به پایان می رسانیم.

نظریه زبانها:

براساس نظریه زبان ها ۴ نوع زبان وجود دارد که برای هر یک از آنها ۴ نوع ماشین موجود می باشد:

۱- زبانهای منظم	نوع سوم	ماشینهای متناهی FA
۲- زبانهای مستقل از متن	نوع دوم	ماشینهای پائین فشردنی PD
۳- زبانهای حساس به متن	نوع اول	ماشینهای کراندار خطی LBA
۴- زبانهای بدون محدودیت	نوع صفر	ماشینهای تورینگ

که می توان از تمامی این ماشینها برای جداسازی و تطبیق در پردازش گفتار و زبان طبیعی استفاده کرد، در ماشینهای قطعی دو گونه پذیرنده وجود دارد که عبارتند از، پذیرنده متناهی قطعی DFA و پذیرنده متناهی غیرقطعی NFA که به طور غیرقراردادی به ماشینی پذیرنده متناهی قطعی می گویند که بطور قطعی بتواند بگوید که رشته ورودی را میپذیرد و یا نمیپذیرد و به ماشینی پذیرنده متناهی غیرقطعی می گویند که بطور غیرقطعی بتواند بگوید که رشته ورودی را میپذیرد و یا نمیپذیرد که این عدم قطعیت را خود بصورت یک تئوری بیان می کنند که در زیر بیان شده است.

نظریه عدم قطعیت:

در نظریه عدم قطعیت ما تصور می کنیم که ماشین به صورت غیرقطعی مسیر درست را می پیماید و رشته مورد نظر را اگر قابل پذیرش باشد می پذیرد و در غیر اینصورت به صورت غیرقطعی رد می کند اما باید در نظر داشت که اگر ماشین رشته ای را به صورت غیرقطعی نپذیرفت این رشته توسط این ماشین قابل پذیرش نمی باشد.

n-gramها:

بوسیله n-gramها که شامل unigram(که همان جداسازی ساده توکن ها می باشد) bigram(که جداسازی کلمات با توجه به کلمه بعدی) و trigram(که جداسازی کلمات با توجه به دو کلمه قبلی می باشد) و ... n-gram می باشد می توان کلمه بعدی یک متن را حدس زد.

روش کار بدین صورت می باشد که بوسیله bigram فقط میتوان با توجه به کلمه قبلی، کلمه بعدی را حدس زد و با trigram می توان با توجه به دو کلمه قبلی، کلمه بعدی را حدس زد و به همین ترتیب در n-gram بوسیله n-1 کلمه قبلی می توان کلمه بعدی را حدس زد، در زیر نحوه محاسبه چند n-gram به خلاصه آورده شده است:

$p(w_n|w_{n-1})$ ⟵ ۱- مدل bigram (مارکوف)

P(rabbit |Just the other I day I saw a) ⟶ P(rabbit |a)

$p(w_n|w_{n-2}w_{n-1})$ ⟵ 2- مدل trigram

P(rabbit |Just the other I day I saw a) ⟶ P(rabbit | saw a)

$p(w_n|w_1^{n-1}) \approx p(w_n|w_{n-N+1}^{n-1})$ ⟵ 3- مدل n-gram

پیاده سازی الگوریتم:

در پیاده سازی عملیات برای زبان فارسی و انگلیسی در نظر گرفته شده است و مدل سازی بوسیلۀ ماشین های DFA انجام شده است، اما در طی طراحی مشکلاتی مشاهده شده است که به معرفی و بررسی برخی از آنها در ادامه می پردازیم.

یکی از بارزترین مشکلات در عمل نشانۀ نقطه می باشد زیرا نقطه از جمله نشانه هایی است که چند منظوره بوده و همین امر باعث بوجود آمدن مشکلاتی در طراحی می شود. نقطه می تواند در نقش هایی همچون نشانه آخر جمله، در اسم های کوتاه (A.mohammadi) در اعداد اعشاری برای تمایز قسمت صحیح عدد از قسمت اعشاری و ... می باشد.

مشکل دیگری که از بارزترین مشکلات در طراحی می باشد علامت ها می باشند، از جمله علمات های چند منظوره که مشکلاتی مشابه با نشانه نقطه دارند. برای مثال علامتی مانند منها (-) می تواند در پشت یک کلمه یا در در پشت یک عدد ظاهر شود، این علامت می تواند نقش یک منفی یا نقش یک تفکیک کننده یا شاید هم یک دسته بندی کننده(...,2-,1-) را بازی کند. این مشکل چند منظوره بودن یک علامت یا نشانه تداخلاتی در طراحی بوجود می آورد . برای رفع این تداخلها می بایست قرار دادهایی را در نظر بگیریم تا بتوانیم تا حدی این مشکلات(تداخل ها) را برطرف کنیم. در ادامه به رفع تعدادی از این مشکلات می پردازیم.

برای نشانه نقطه قرارداد به این صورت در نظر گرفته شده که اگر یک حرف بیاید و بعد از آن نقطه قرار گیرد آن شروع یک اسم اختصاری(A.Ali,A.M.Rattz) در نظر می شود و در ماشین به دنبال مابقی حروف اسم اختصاری رفته و از رفتن به حالت پذیرش جلوگیری می شود تا ماشین به یکی از علائم جداساز برسد.

در اعداد اعشاری اگرعدد دیده شود ماشین به خواندن خود ادامه داده تا به نقطه برسد، در اینجا نقطه را به عنوان مشخصه قسمت اعشاری در نظر می گیرد و اگر علامتی غیر از نقطه ببیند آن را به عنوان عدد صحیح در نظر می گیرد و به حالت پذیرش می رود.

در طراحی ماشین سه حالت پذیرش در نظر گرفته شده است که هر حالت مشخص کننده پذیرش نوع های مختلف ورودی می باشد که براساس گروه بندی کاراکتر ها مشخص شده است. حالت پذیرش برای کلمات می باشد، حالت دوم برای پذیرش اعداد و نهایتا

حالت سوم حالت پذیرش trap می باشد. هدف از حالت پذیرش trap در زمانی اتفاق می افتد که چند علامت نشانه گذاری با هم آمده باشند(مانند آمدن دو نقطه پشت سر هم).

قراردادها:

همان طور که قبل تر گفته شد کاراکتر ها به دسته های مختلف تقسیم شده اند. در زیر جدول گروه بندی کاراکتر ها نشان داده شده است.

حروف	a-zA-z	Word
اعداد	0-9	Digit
جاخالی	Space, newline	Space
نشانه های ریاضی	+,-,*,/,=,^,%,<,>	MathMark
نشانه های پایانی	; ,! , : , . , ,	EndMark
توضیحات	() , {} , [] , " , ' , < , >	Comment
نشانه نقطه	.	Point
نشانه	Others	Mark

جدول ۱) جدول گروه بندی کاراکترها

برای پیاده سازی ماشین DFS، قراردادهایی جهت عملکرد ماشین در نظر گرفته شده است که در طی پیاده سازی می بایست لحاظ گردند. جدول زیر قرارداد های پیاده سازی ماشین را نشان می دهد:

حالت غیر پایانی به پایانی	حرکت غیر پایانی به غیر پایانی	حالت پایانی به غیر پایانی	حالت پایانی به پایانی
• خواندن هد • درج ch در لیست trap • برو به حالت پایانی	• خواندن هد • درج current در لیست digit یا word • برو به حالت غیر پایانی	• درج current در لیست digit یا word • خواندن هد • درج در current • برو به حالت غیر پایانی	• درج current در لیست digit یا word • درج ch در لیست trap • خواندن هد • برو به حالت پایانی

جدول ۲) جدول قراردهای پیاده سازی

(ch) متغیری که مقدار خوانده شده از هد در آن قرار دارد (current) متغیری که کاراکترهای خوانده شده از ماشین در آن ذخیره می شود تا در حالت پایانی در لیست مربوطه ذخیره شود

ماشین متناهی قطعی(DFA):

در پایان دیاگرام طراحی شدهٔ ماشین DFS در شکل زیر نشان داده شده است.