به نام خدا

گزارش پروژه درس پردازش زبان طبیعی

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 و با محاسبه احتمال آنها، برای پیشگویی کلمات و جایگاه آنها از این احتمالات استفاده شود. حال با مقدمه ای درباره نظریه زبانها و توضیحاتی پیرامون ماشینهای متناهی فطعی و n-gramها و طریقه محاسبه احتمالات به روند پیاده سازی الگوریتم رسیده و با 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 نمیپذبرد 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

محاسبه احتمالات:

در bigram محاسبات برای دو کلمه پشت سر هم(دو کلمه با هم) انجام می شود. برای محاسبه، ابتدا تعداد تکرار دو کلمه پشت سر هم در متن را بدست آورده سپس بر تعداد تکرار کلمه ی اول در دو کلمه مورد نظر در متن تقسیم می شود. فرمول به صورت زیر می باشد:

$$p(w_n|w_{n-1}w_n) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$$

در trigram محاسبه مشابه با bigram است اما به صورتی که در فرمول زیر نشان داده می شود، ابتدا تعداد سه کلمه پشت سرهم را بدست آورده و برتعداد دو کلمه اول پشت سرهم تقسیم می نمائیم:

$$p(w_n|w_{n-2}w_{n-1}) = \frac{C(w_{n-2}w_{n-1}w_n)}{C(w_{n-2}w_{n-1})}$$

منظور از C تعداد تکرار کلمات داخل پرانتز در متن ما می باشد.

روند پیاده سازی:

در طراحی ماشین چهار حالت پدیرش 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 اول پذیرش برای کلمات می باشد، حالت دوم برای پذیرش اعداد، حالت سوم برای علامتها Mark ها و نهایتا حالت چهارم 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).

ابتدا متن را از داخل فایل HTML بیرون کشیده و پس از آن متن را به کلمات و جملات تقسیم کرده و تعداد هرکدام را بدست آورده و احتمال کلمات، علامتها و اعداد را محاسبه می کنیم.

سپس bigram و trigram جملات را پیدا کرده و با استفاده از قوانین احتمال برای bigramها و trigramها احتمال آنها را نیز محاسبه نموده. پس از آن با استفاده از این احتمالات و استخراج توکن ها، bigramها و trigramها از جملات احتمال را بطور نرمال سازی شده حساب نموده و در جلوی هر جمله نمایش می دهیم.

در انتها با استفاده از قابلیت ویرایش امکان حذف، درج و جابجایی کلمات در جملات و یا درج جمله ای جدید را داریم که با استفاده از Corpus قدیمی احتمال جمله جدید را محاسبه می نمائیم.

برای پیاده سازی ماشین DFA، 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:

جدول ۲) جدول قراردادهای پیاده سازی

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

(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.

مشکلات پیاده سازی:

در پیاده سازی عملیات برای زبان فارسی و انگلیسی در نظر گرفته شده است و مدل سازی بوسیلهٔ 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 نشانهٔ 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.Yousefi) در اعداد اعشاری برای تمایز قسمت صحیح عدد از قسمت اعشاری(مخصوص زبان انگلیسی) و ... می باشد.

یکی دیگر از مشکلات مفهومی در پایان جملات می باشد، زیرا بطور مثال برخی از جملات با علامت ویرگول به پایان می رسد و در برخی دیگر از جملات ویرگول علامت جدا کننده کلمات و مکث در جمله می باشد. همین مشکل را میتوان در دیگر علامتهای پایان نیز یافت، مانند علامت ; و : و از این دست علامت ها که در همه جا نشان پایان جمله نمی باشد و این موضوع باعث ایجاد جملاتی گاه طولانی که شامل چندین جمله است و گاه جملاتی بی معنا و حتی در برخی موارد باعث ایجاد جملاتی که تنها یک کلمه می باشد، می شود.

مشکل دیگری که از بارزترین مشکلات در طراحی می باشد علامت ها ریاضی 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 که تا حدی نادرست هستند در نظر بگیریم تا بتوانیم برخی از این مشکلات(تداخل ها) را برطرف کنیم. در ادامه به رفع تعدادی از این مشکلات می پردازیم.

۱- 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.

۲- 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.

۳- ...

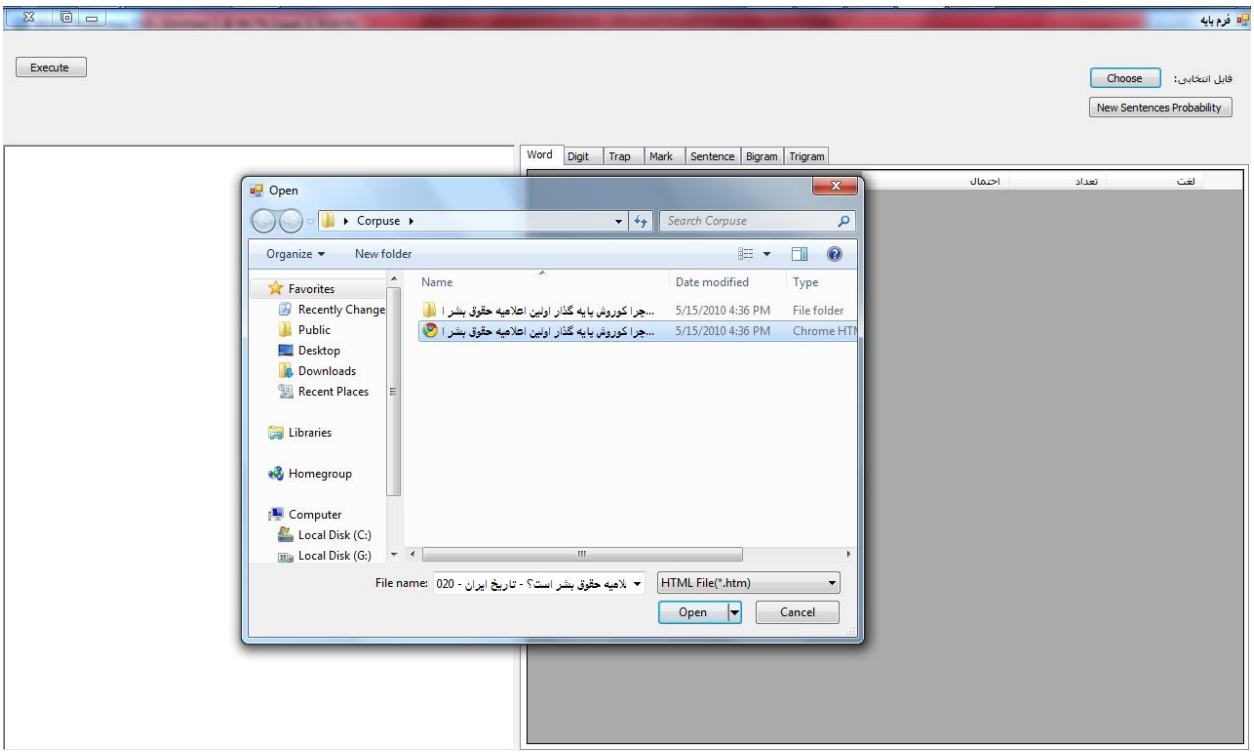
که بدون ذکر تمامی قراردادها با قدری تأمل در نمودار DFA این برنامه می توان به قراردادهای تنظیم شده برای این نرم اقزار، پی برد.

برخی از مشکلات که تقریباً مشکلاتی حل نشدنی هستند به خاطر وجود ماشین متناهی می باشد زیرا ماشین متناهی قادر به پذیرش گرامرهای منظم می باشد و اغلب قواعد زبان فارسی و انگلیسی خارج از گرامرهای منظم می باشد که از آن جمله میتوان به متون داخل پرانتز، نقل قول و کلاً کامنت ها اشاره نمود، زیرا برای نقل قول ها ما باید بدانیم که اگر علامت نقل قول دیگری هم در انتهای جمله وجود دارد آنگاه آنرا مانند نقل قول در نظر گرفت در غیر اینصورت آن نقل قول ممکن است علامت " به معنی ایضاً و یا صدها چیز دیگر باشد و موارد دیگری از این دست که در قالب زبانهای منظم نمی گنجد.

حتی در قرارداد شماره۲ که قراردادی کامل نمی باشد مشاهده می کنیم که اگر بعد از عدد نقطه بیاید آنرا به منزله عدد می پذیرد (در زبان فارسی به جای نقطه از اسلش استفاده می شود) در حالیکه ممکن است نقطه پایان جمله باشد که آخرین توکن آن عدد است و منظور از این نقطه ممیز جدا کننده قسمت صحیح و اعشاری نباشد.

روند اجرای برنامه:

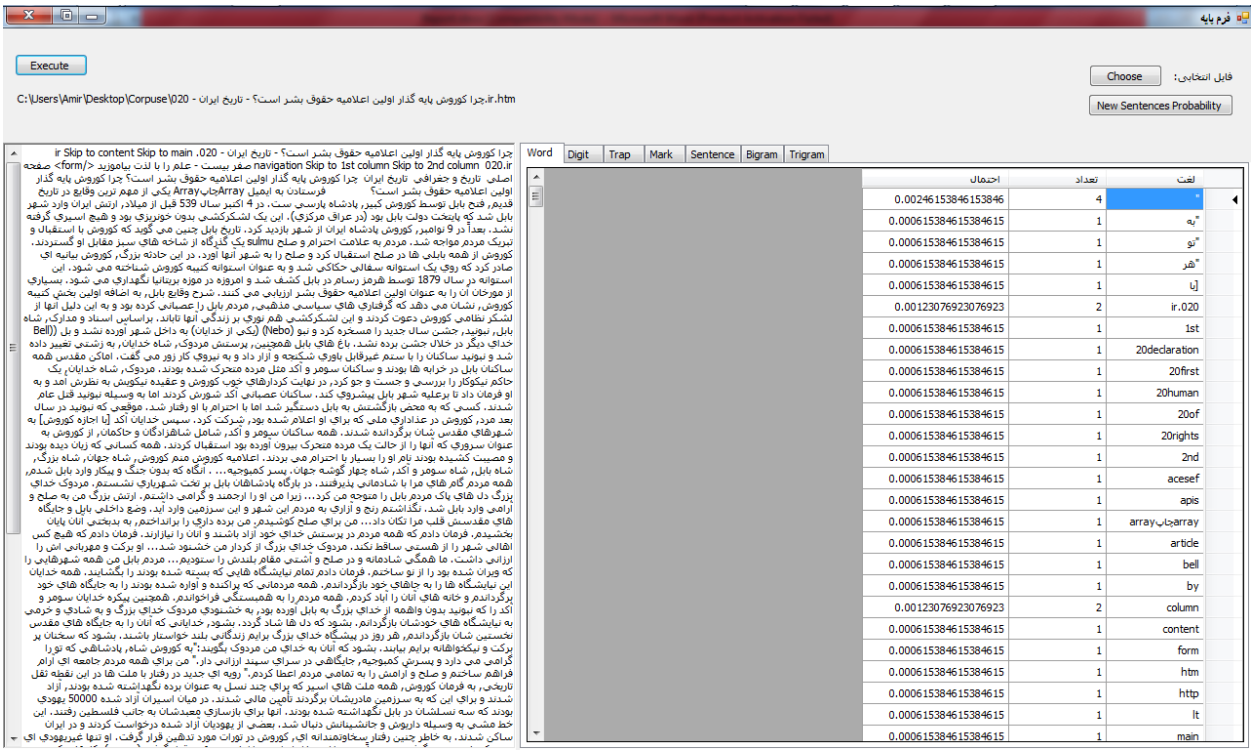
ابتدا فایل HTML مورد نظر را انتخاب می کنیم.(شکل ۱) بعد از انتخاب فایل نگ های HTML آن حذف شده و متن را از داخل نگ های HTML استخراج می کنیم و نتیجه را در پنجره سمت چپ نمایش می دهیم.(شکل ۲)



شکل ۱- انتخاب فایل HTML مورد نظر

شکل ۲- استخراج متن از داخل فایل HTML

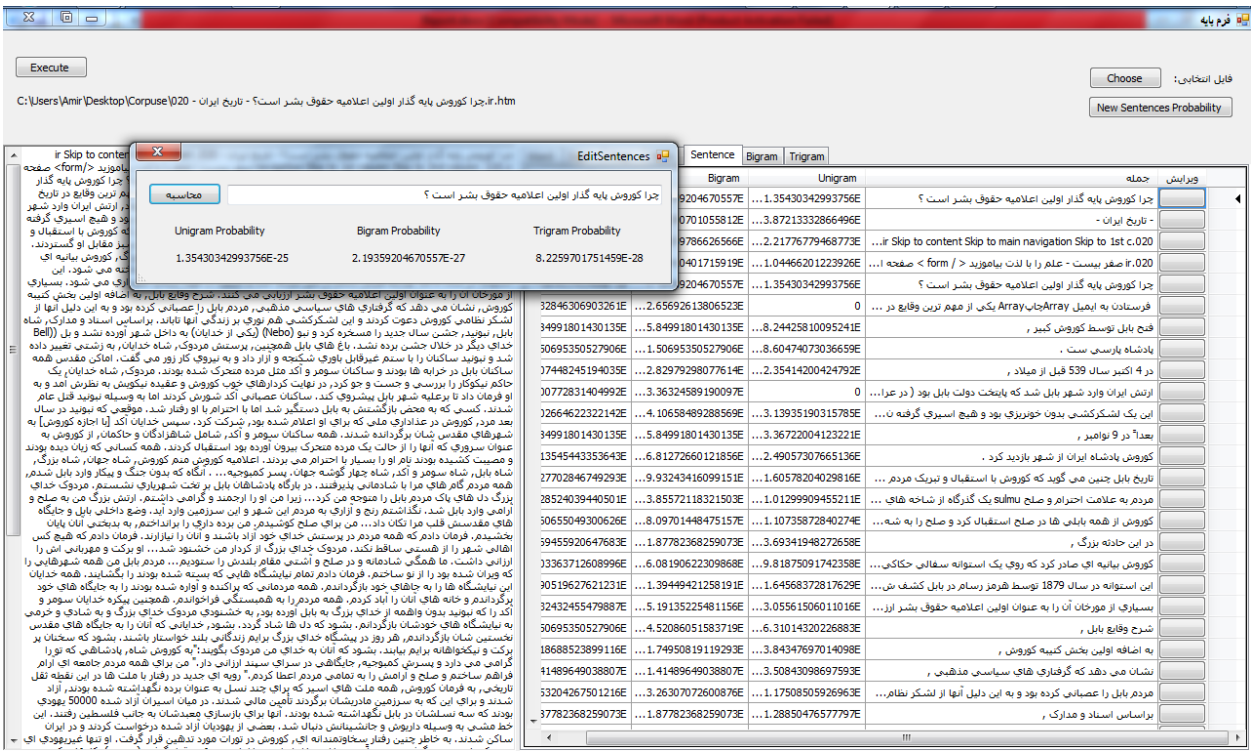
سپس برای جداسازی توکن ها و جملات از متن و یافتن Bigramها و Trigramهای جملات و بعد از آن محاسبه احتمال توکن ها و Bigramها و Trigramها بطور جداگانه و بوسیله احتمالات بدست آمده یافتن احتمال جملات روی دکمه Execute کلیک نموده و پس از چند دقیقه (بسته به اندازه Corpuse) نتایج در tabهای جداگانه نمایش داده می شوند.(شکل ۳)



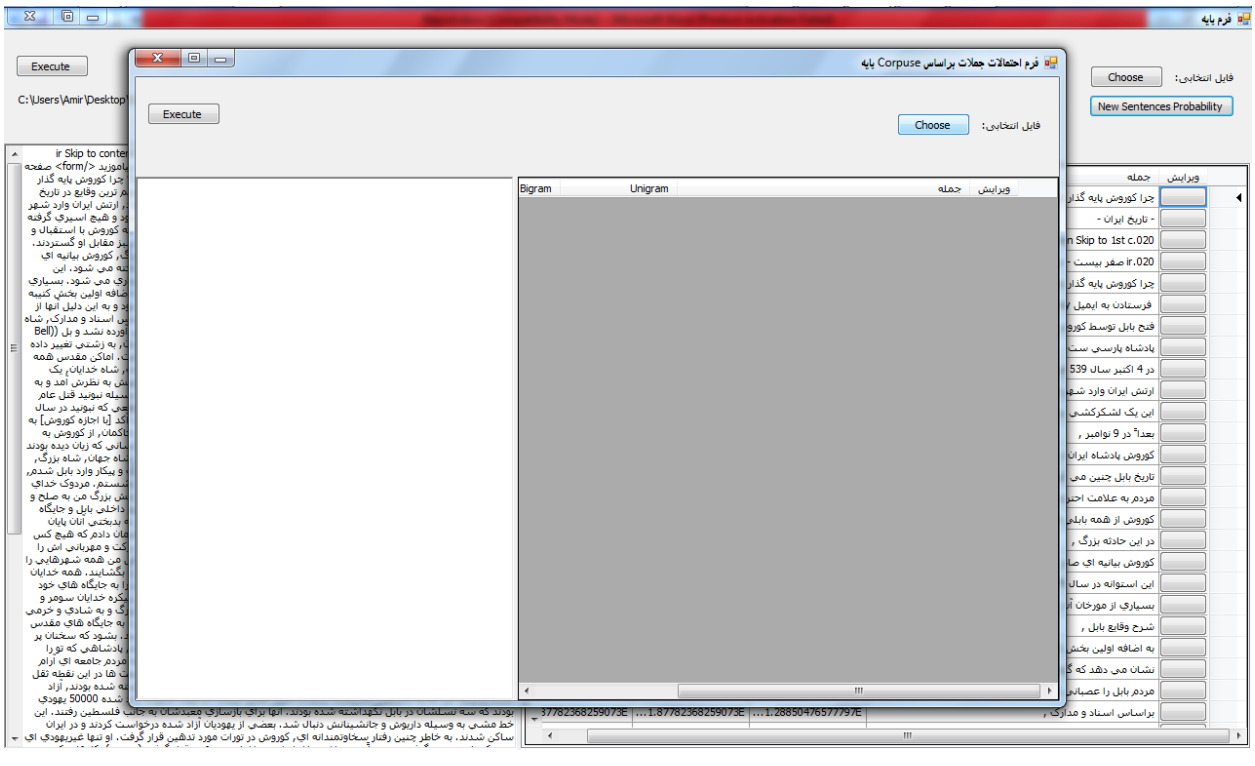
شکل ۳- استخراج توکن ها پس از Execute برنامه

حال می توانیم برای ویرایش جملات یافت شده و سپس محاسبه احتمال آنها روی دکمه در نظر گرفته شده در کنار هر جمله در tab جملات کلیک نموده تا پنجره جدید باز شود.(شکل ۴)

برای یافتن احتمالات جملات جدید دیگری که توسط Corpuse دیگری وارد شده اند(با استفاده از احتمالات Corpuse قدیمی) روی دکمه New Sentences Probability کلیک می نمائیم تا پنجره جدیدی باز شود.(شکل ۵)

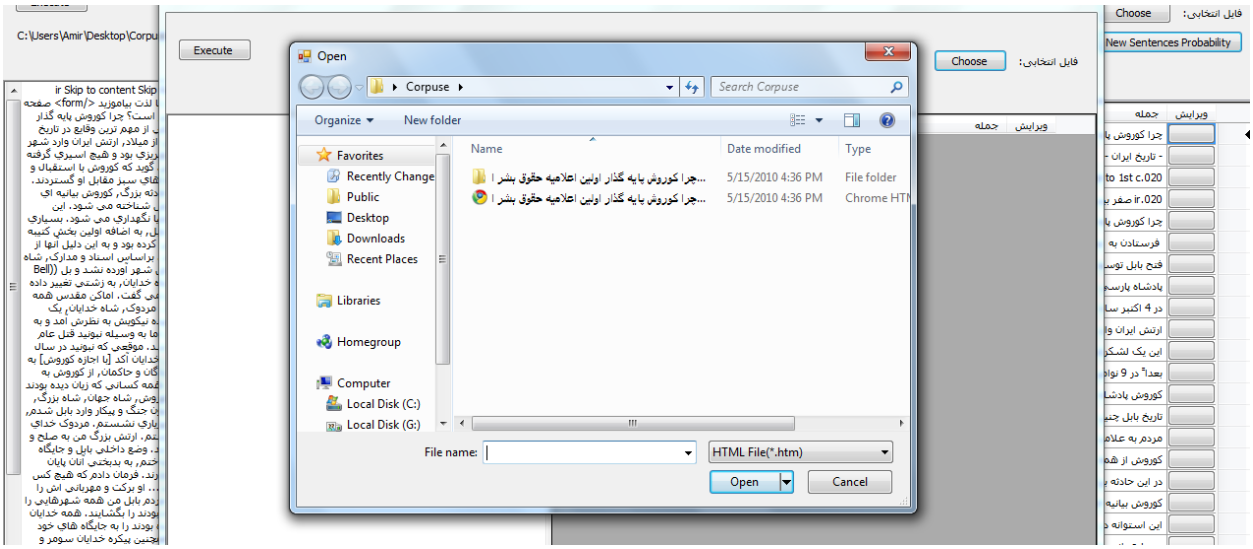


شکل ۴- ویرایش جملات موجود و محاسبه احتمالات آنها

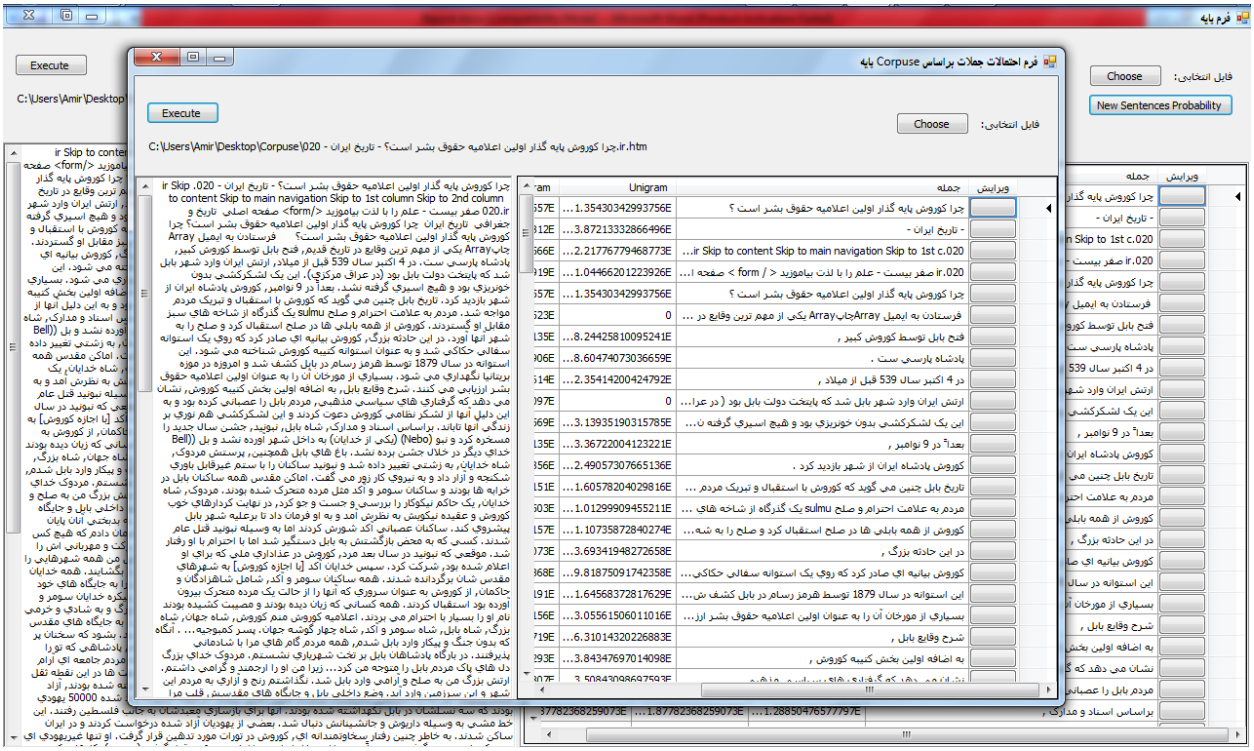


شکل ۵- پنجره تست احتمالات Corpuse جدید با استفاده از احتمالات Corpuse اولیه

حال دوباره مانند مراحل اول فایل جدید دیگری را باز کرده و پردازش را روی این فایل جدید(با کلیک روی دکمه Execute) انجام می دهیم که نتیجه را در شکل زیر مشاهده می نمائید:(شکل ۶ و ۷)



شکل ۶- انتخاب قایل Corpuse جدید



شکل ۷- نتایج حاصل از احتمالات Corpuse قدیم بر روی متن جدید

قراردادها:

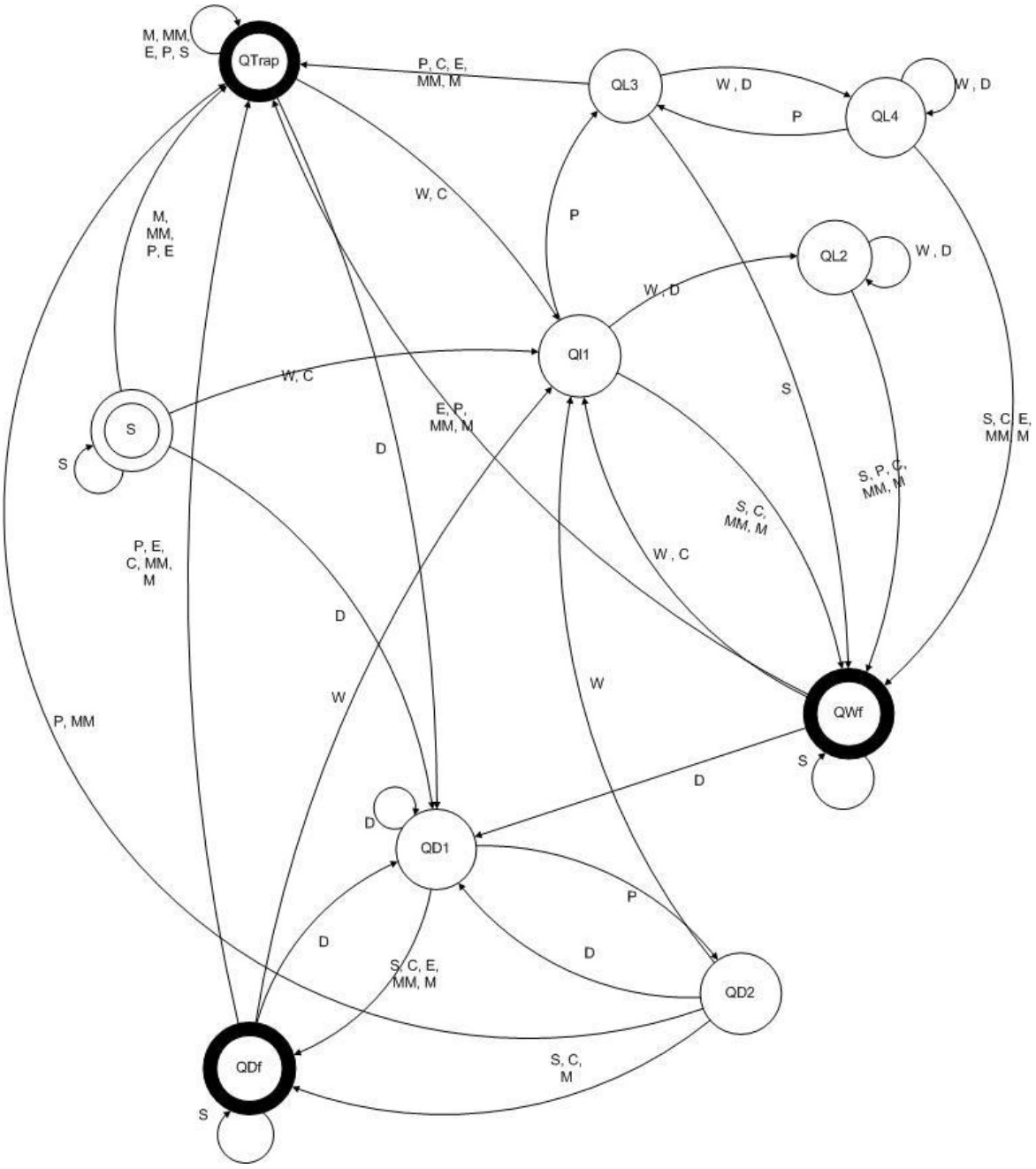
همان طور که قبل تر گفته شد کاراکتر ها به دسته های مختلف تقسیم شده اند. در زیر جدول گروه بندی کاراکتر ها نشان داده شده است.

جدول ۱) جدول گروه بندی کاراکترها

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

ماشین متناهی قطعی(DFA):

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



شکل ۸- ماشین DFA که کلمات اعداد و نشانه ها را از هم جدا می کند