Algorithm of different spelling

معرفی و کاربرد :

وظیفه الگوریتم تولید اسپل های انگلیسی مختلف یک اسم فارسی که به زبان انگلیسی نوشته شده است می باشد. بنابراین باید یک اسپل تقریبا صحیح به ورودی داده شود تا بتواند اسپل های مختلف دیگر را تولید کند.

این الگوریتم از داده های آماری یک دیتاست از اسامی فارسی به زبان انگلیسی استفاده میکند. برای مثال؛ اسم hosein در داخل دیتاست ۱۰۰۰ بار و اسم Hossein در آن ۲۰۰۰ بار استفاده شده است. پس دیتاست شامل اسامی مختلف با اسپل های مختلف و تعداد تکرار آنهاست.

فاز اول (تولید احتمالات) :

روشی که مد نظر این الگوریتم است؛ استفاده از ngram بر روی اسامی میباشد. با 2gram کردن یک اسم مثلا Hossein داریم :

(ho, os, ss, se, ei, in) ؛ از این مجموعه میتوان متوجه شد که جفت os بعد از جفت ho می آید. همچنین ss بعد از os و se بعد از ss و به همین ترتیب میتوان این اطلاعات را از سایر اسامی استخراج کرد و مطابق تعداد تکرار اتفاق os بعد از ho، یک احتمال از اینکه os بعد از ho بیاید را بیرون اورد.

این کار را برای تمامی اسامی دیتاست انجام داده و تعداد اتفاقات یک جفت کاراکتر را بعد از جفت کاراکتر دیگر بدست می آوریم. سپس برای بدست اوردن احتمال از فرمول زیر استفاده میکنیم :

$$probility('os'\ after\ 'ho') = \frac{count('os'\ after\ 'ho')}{count('ho')}$$

به این شکل میتوان احتمال اتفاق یک جفت کاراکتر بعد از جفت کاراکتری دیگر را بدست اورد.

حال یک دیشکنری داریم که این احتمالات را در خود نگه داشته است و در مرحله‌ی بعد مورد استفاده قرار میگیرد.

فاز دوم (تولید درختی از اسپل های متفاوت) :

تولید ترکیبات از ابتدای کلمه :

اگر یک اسپل از اسم را داشته باشیم ؛ با برداشتن n=2 کاراکتر اول آن و پیدا کردن اینکه چه جفت هایی و با چه احتمالاتی بعد از آن می‌آید میتوان یک قسمت از اسپل اسم را تولید کرد. برای مثال اگر اسم hosein را به

الگوریتم بدهیم؛ ho از آن برداشته شده و در دیکشنری که در قسمت قبل تولید شد سرچ میشود که پس از ho چه جفت کاراکتری امده است و با چه احتمالی.

برای مثال احتمال اینکه بعد از ho ؛ os قرار بگیرد بالاترین عدد را دارد. و ابتدا کلمه‌ی hos ساخته میشود. ممکن است احتمال بعدی ou باشد که کلمه‌ی hou ساخته میشود و به همین ترتیب کلمات چند احتمال برتر ساخته میشود.

حال در ردیف اول، hos را داریم؛ باید بررسی شود که بعد از os (یعنی دو کاراکتر اخر کلمه‌ی ساخته شده) چه جفت کاراکتری آمده است. ممکن است ss امده باشد و یا مثلا se . پس کلمات hoss و hose ساخته میشوند.

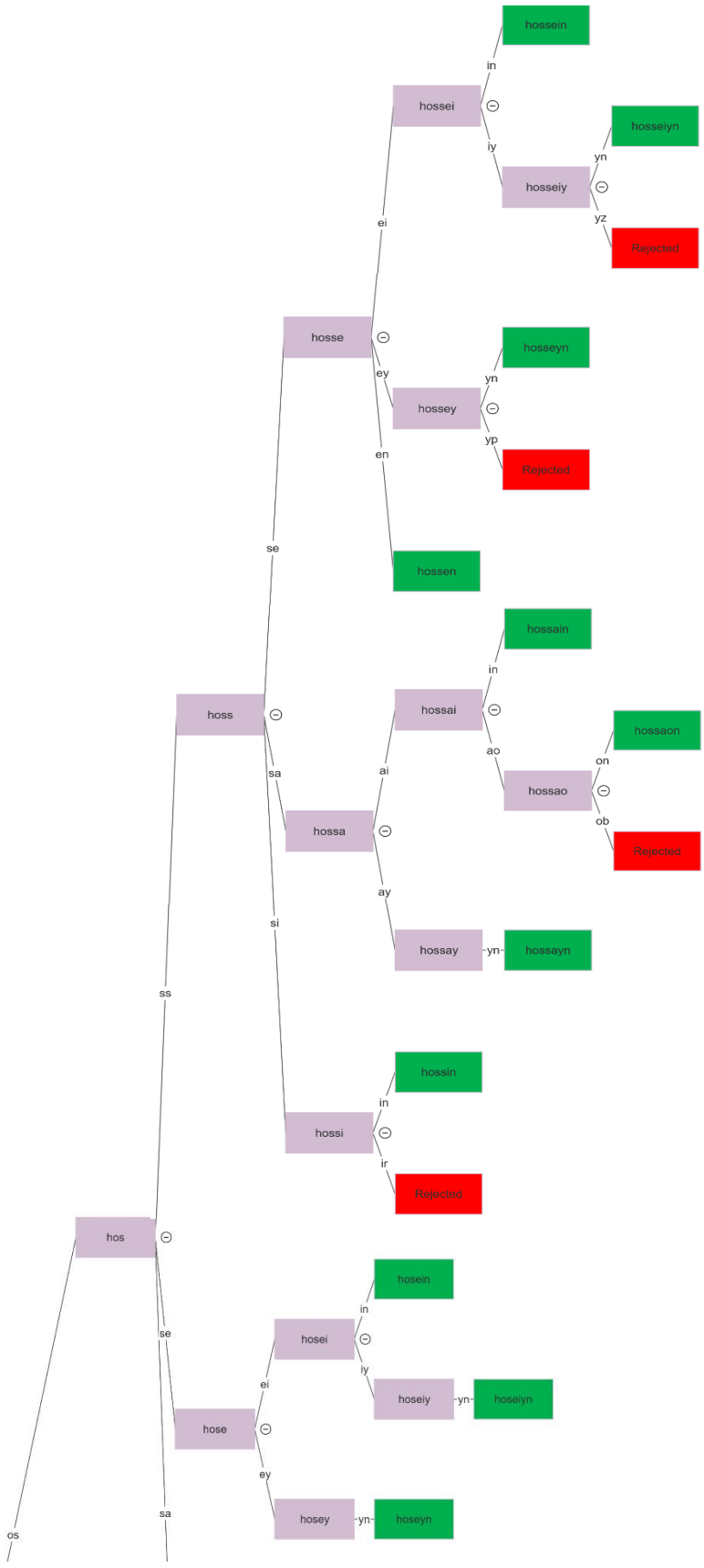
مجددا n=2 حرف **آخر** کلمه‌ی ساخته شده را برداشته و داخل دیکشنری سرچ میشود؛ یعنی سرچ میشود که پس از ss چه جفتی آمده است(برای کلمه‌ی hoss). مثلا se امده پس کلمه‌ی جدید hosse است. به همین ترتیب پیش میرود و ei و in را نیز به کلمه ی تولید شده قبلی متصل می شود و Hossein ساخته میشود. به این ترتیب با دادن یک اسپل از hosein به Hossein رسیدیم.

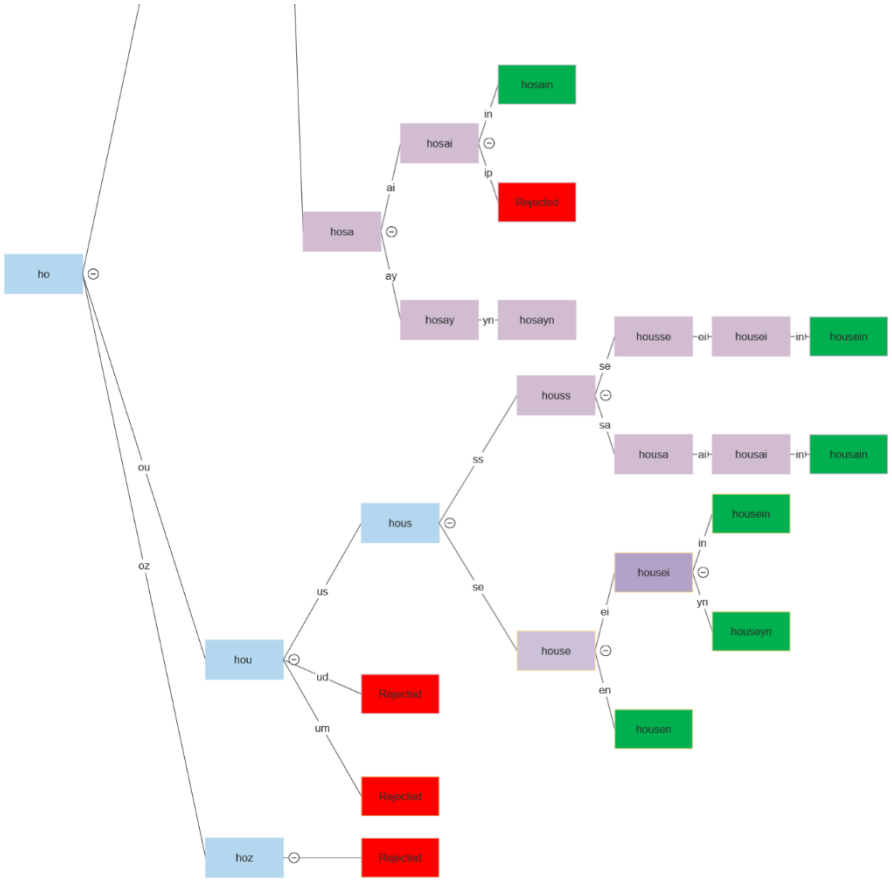
همانگونه که توضیح داده شد، این عملیات بصورت درختی پخش میشود و کلمات بسیاری تولید میکند که تعداد زیادی از آنها نیز ممکن است نامربوط و اشتباه باشند اما تا آنجایی که میسر باشد بخش قابل توجهی از اسپل های متفاوت یک اسم را هم تولید میکند.

برای کاهش تعداد خروجی ها میتوان از میانگین هندسی احتمالات استفاده کرد و بر اساس آن، کلمات تولید شده را مرتب کرد و صرفا چند احتمال برتر را در نظر گرفت. همچنین از فیلترهایی استفاده میشود تا کلمات تولیدی پرت و دور از اسم اصلی نباشند. یعنی مثلا اگر بعد از ho ؛ oz آمد، ترکیب جدید ساخته نشود چرا که در کلمه‌ی اصلی (hosein) ، z وجود ندارد.

در دو صفحه ی بعد درخت بخشی از نحوه‌ی عملکرد الگوریتم روی اسم hosein رسم شده است. باید توجه داشت که درخت ایجاد شده توسط الگریتم بسیار بزرگ تر از چیزی که نشان داده شده است میباشد و این شکل صرفا برای درک بهتر است.

همچنین برای اینکه الگوریتم متوجه شود که تا چه عمقی از درخت جلو رود؛ در فاز یک (تولید احتمالات) و هنگام خواندن از دیتاست؛ به اول و آخر هر اسم یک * و # اضافه میکند تا در فاز دوم بدانیم ابتدا و انتهای هر اسم کجاست. در این صورت درخت تا عمقی پیش میرود که به اسمی برسد که آن اسم ستاره و مربع را باهم داشته باشد. (اگر ترکیب ایجاد شده ستاره و مربع را باهم نداشته باشد به کار ادامه میدهد).


hos
os
ss
se
hose
ei
hosei
in
hosein
iy
hoseiy
yn
hoseiyn
ey
hosey
yn
hoseyn
sa
hossa
ai
hossai
in
hossain
ao
hossao
on
hossaon
ob
Rejected
ay
hossay
yn
hossayn
si
hossi
in
hossin
ir
Rejected
se
hosse
ei
hossei
in
hossein
iy
hosseiy
yn
hosseiyn
yz
Rejected
ey
hossey
yn
hosseyn
yp
Rejected
en
hossen
hoss



تولید ترکیبات از انتهای کلمه :

اگر به طریقه فوق پیش رویم میتوانیم انواع اسپل های متفاوت اسم حسین که با ho آغاز میشوند را تولید کنیم اما هرگز نمیتوان Hussein را تولید کرد. چرا که در حالت گفته شده یک پیش فرض داریم (ho) و بر اساس آن، اسپل های متفاوت را میسازیم. برای حل این مشکل دقیقا عکس الگوریتم ذکر شده را اجرا میکنیم. یعنی از انتهای hosein؛ اسم hussein را بسازیم. به این شکل که ابتدا in را به عنوان پیش فرض در نظر گرفته و از داخل دیکشنری ایجاد شده در فاز ۱ سرچ میشود که قبل از in چه حرفی آمده است. احتمال امدن e بالاست و ein ساخته میشود. سپس دو حرف **اول** آن گرفته شده (چون مسیر حرکت برعکس حالت قبل است) و سرچ میشود که قبل از ei چه آمده است. زمانی که به ترکیب sein رسیدیم سرچ میشود که قبل از s چه آمده است؛ ممکن از o و u بالاترین احتمالات باشند و ترکیبات ossein و ussein ساخته میشوند. به این طریق میتوانیم دو کلمه ی پیش فرض در حالت قبل (شروع از راست) را که توانایی تغییر آن را نداشتیم را تغییر دهیم.

این الگوریتم با چالشی مواجه است و آن، تولید ترکیبات ناخواسته است. که باعث تولید بیش از حد ترکیبات میشود و این عملکرد cpu و حافظه رم را مختل میکند.

اگر بحث محدودیت تولید وجود نداشت، این الگوریتم توانایی تولید ۹۹ درصد اسپل های متفاوت را دارا بود.

اما چون پیش روی در هر عمق درخت بصورت تصاعدی تعداد نتایج را بالا میبرد؛ نهایتا تا عمق ۵ و ۶ پیش میرویم. و بازهم این الگوریتم تا حد بالایی از اسپل های متفاوت را میتواند تولید کند.