**«بسم رب المهدی»**

**پیش نویس پژوهش نامه درس هوش مصنوعی**

موضوع پژوهش

**Transliterate کردن از انگلیسی به فارسی**

استاد

**جناب آقای دکتر مینایی**

**استاد راهنما**

**جناب آقای مهندس رسولی**

**گردآورندگان**

**سینا سلیمان پور 87522106**

**حامد رضانژاداصل بناب 87521002**

**چکیده**

زبان فارسی یک زبان هندی-اروپایی است که توسط رسم الخط عربی توسعه یافته[[1]](#footnote-2)به نوشتار درمی آید و زبان رسمی ایران ، افغانستان و تاجیکستان می باشد. Transliteration یک پروسه اتوماتیک برای تبدیل رسم الخط یک کلمه از زبان مبدا به رسم الخط زبان مقصد با حفظ ساختار آوایی کلمه می باشد (KARIMI 2009).

در این مقاله ما روش های مختلفی که در رابطه با موضوع Transliteration ارائه شده (با محوریت انگلیسی به فارسی) را به طور خلاصه بررسی می کنیم.

1. **مقدمه**

[[2]](#footnote-3)Machine Translation یک بخش مهم از بسیاری از برنامه های چند زبانه[[3]](#footnote-4) می باشد. در فضای جهانی امروز مهمترین برنامه هایی که MT نیاز دارند (KARIMI 2009) شامل استخراج اطلاعات بین زبانی، پرسش پاسخ های بین زبانی، برنامه های چت چندزبانه، مترجم های سخنگو و ترجمه آنی ایمیل ها و وب سایت ها می باشند.

دیکشنری ها ترجمه توسط انسان را مقدور ساختند و همچنین در ترجمه های ماشینی ای که بر مبنای دیکشنری هستند استفاده می شوند. دیکشنری های معمول حدود 50000 تا 150000 کلمه در خود جای می دهند که در عمل متن ها شامل کلمات خیلی بیشتری می باشند. برای مثال یک مجموعه از متن های جمع آوری شده از Associated Press newswire در طی 10 ماه شامل 44 میلیون کلمه می باشد که خود شامل 300000 کلمه متفاوت می باشد. کلماتی که خارج از دیکشنری هستند شامل اسم های خاص مانند اسم شرکت ها، مردم، مکان ها و محصولات می باشد (DALE 2007). در این موارد که کلمات در دیکشنری موجود نمی باشند به Transliteration نیاز پیدا خواهیم کرد.

Machine Transliteration حدود یک دهه قبل به عنوان قسمتی از Machine Translation برای کار با اسم های خاص و عبارت های تکنیکی که با حفظ تلفظ ترجمه می شوند به وجود آمد (KARIMI 2009).

**در این مقاله ما ابتدا به بررسی مفهوم های پایه ای مربوط به** Machine Translation می پردازیم، سپس به معرفی و بررسی دو روش برای Transliteration کردن می پردازیم.

1. مفهوم های کلیدی

**بعضی از مفهوم های معمول با پیش زمینه زبانی که در این مقاله استفاده می شود را در این بخش توضیح می دهیم. اطلاعات بیشتر با جزییات در مورد سیستم های نوشتاری، حروف نوشتاری و آواهای زبان های مختلف در انتشارات**[[4]](#footnote-5) IPA(International Phonetic Alphabet) **پیدا می شود.**

آواشناسی و واج شناسی[[5]](#footnote-6). **آواشناسی مطالعه ی صداهای گفتار انسان می باشد و در مورد خصوصیات اصلی صدای گفتار بحث می کند. آواشناسی به صورت مستقل با صداها رفتار می کند. اما واج شناسی در طرف مقابل آواشناسی بر روی سیستم های صدایی و واحد های صدایی انتزاعی همانند واج ها، واج آرایی و قوانین واج شناسی کار می کند. ارائه ی آوایی[[6]](#footnote-7) یک صدا توسط [] نشان داده می شود و واج ها توسط // نشان داده می شوند. برای مثال نسخه ی صدایی حرف فارسی “پ” و حرف انگلیسی “P” بصورت [p] نشان داده می شوند.**

واج[[7]](#footnote-8). **یک واج کوچک ترین بخش سخن است که تغییر معنایی ایجاد می کند. واج ها بخش مهمی از هر کلمه هستند و جایگذاری آن ها باعث تغییر معنایی در کلمه می شود. برای مثال اگر ما صدای [b] را با [p] عوض کنیم کلمه “Big” به کلمه ی “Pig” تغییر معنا می دهد پس /b/ یک واج می باشد.**

**بعضی از الگوریتم های Transliteration از واج ها برای تکه تکه کردن کلمات قبل از Transliteration استفاده می کنند.**

حرف[[8]](#footnote-9). **یک حرف جز بنیادی یک زبان نوشتاری است. حروف شامل حروف الفبایی، اعداد، نشانه ها و کل سمبل های تکی هر سیستم نوشتاری می باشد. برای مثال کلمه “ship” شامل 4 حرف (s,h,I,p) می باشد ولی فقط 3 آوا دارد.**

سیلابل[[9]](#footnote-10). **یک سیلابل یک بخش از تلفظ می باشد. یک سیلابل معمولا بزرگتر از یک تک صدا و کوچکتر از یک کلمه می باشد. معمولا یک سیلابل از اوج سیلابل که معمولا یک حرف صدا دار است و چند حرف بی صدای اختیاری (چه در اول چه در آخر سیلابل) تشکیل می شود.**

سیستم نوشتاری[[10]](#footnote-11). **سیستم نوشتاری یک سیستم نمادی برای بیان جملات و حالات در یک زبان است. یک سیستم نوشتاری 4 مجموعه ی خصوصیات دارد:**

1. یک مجموعه از نمادها که به تنهایی کاراکترها یا حروف خوانده می شوند و با هم اسکریپ[[11]](#footnote-12) گفته می شوند.
2. یک سری قوانین اختیاری که به حروف، ترتیبشان و روابطشان معنی می بخشد و توسط یک اجتماع قابل فهم و مشترک می باشد.
3. یک زبان که توسط تفسیر این قوانین و عناصر، ساختارش ارائه می شود.
4. بعضی جنبه های فیزیکی از ارائه ی جداگانه ی سمبل ها، به ضرورت نیاز به یک واسطه دائمی یا نیمه دائمی که شاید تفسیر شوند.
5. **مشکلات معمول در Transliteration :**

مهمترین مشکلات در Transliteration را می توان به 5 دسته تقسیم کرد(KARIMI 2009) :

ویژگی های متنی[[12]](#footnote-13) ، صداهای محذوف[[13]](#footnote-14)، گونه های مختلف Transliteration[[14]](#footnote-15) ، زبان مبدا[[15]](#footnote-16) و تصمیم گیری در مورد اینکه آیا بر روی یک اسم Translation یا Transliteration را انجام دهیم.

**ویژگی های متنی.** امکان وجود متن های مختلف بین زبان مبدا و مقصد اولین چالشی است که سیستم های Transliteration باید با آن مواجه شوند. همانگونه که توضیح داده شد یک متن می تواند ارائه ای از یک یا چند سیستم نوشتاری مختلف باشد و شامل سمبل هایی است که می توانند نوشته ای را ارائه دهند. برای مثال متن لاتین تمامی تمامی زبان های اروپای غربی را پوشش میدهد و متن عربی برای زبان عربی، فارسی، پشتو، اردو، مالایایی و بالتی استفاده می شود. از طرف دیگر بعضی از زبان های نوشتاری نیاز به چند متن دارند همانند ژاپنی که حداقل با سه متن نوشته می شود(هیراگانا ، کاتاکانا و کانجی)[[16]](#footnote-17).

یکی دیگر از جنبه های مهم یک متن، جهت نوشتن آن است. بعضی از زبان ها از راست به چپ نوشته می شوند و بعضی دیگر از چپ به راست. برای مثال زبان فارسی و عربی راست به چپ و زبان انگلیسی و سایر زبان هایی که از متن لاتین استفاده می کنند چپ به راست هستند.

**صداهای محذوف.** زبان های مختلف انسان هر کدام ساختار صدایی و متن زبان مربوط به خود را دارند. اگر یک صدای محذوف در حروف یک زبان موجود باشد صداهای تکی توسط 2حرف های تک صدا یا 3حرف های تک صدا نشان داده می شوند. برای مثال در دو حرف تک صدای انگلیسی “sh” به صدای [ᶴ] اشاره می کند. عمل Transliteration میتواند صدایی جدید را به زبان مقصد که به صورت عادی موجود نمی باشد اضافه کند. برای مثال در زبان انگلیسی صدای [X] موجود نمی باشد اما هنگام Transliterate کردن کلمه "خسرو" از فارسی به انگلیسی مترادف /kh/ را پیدا می کند.



شکل 1. مثال هایی از زوج های Transliterate شده در چهار زبان

**گونه های مختلف Transliteration.** Transliteration یک پروسه خلاقانه می باشد که گونه های مختلفی برای یک کلمه از زبان مبدا به زبان مقصد را امکان پذیر می سازد. لهجه های مختلف در یک زبان می تواند به گونه های مختلف Transliteration ختم شود. در حالی که جمع آوری تمام این گونه ها برای یک کلمه در بستر کلمه ای ما امکان پذیر نمی باشد زیرا به سادگی نمی توان تمامی سخن گوهای یک زبان را در شهرها و کشورهای مختلف جمع آوری کرد. همچنین نام های جدید شرکت ها، محصولات و مردم هر روزه معرفی می شوند که راهی جز داشتن یک استاندارد برای Transliteration را به جای نمی گذارد. بنابراین به دست اوردن سیستم های Transliteration مختلف مشکل ساز می شود که خود عاملی برای مقایسه کردن این سیستم ها می تواند باشد.

**زبان مبدا.** سرراست ترین راه برای طراحی سیستم های Transliteration استفاده از بستری[[17]](#footnote-18) برای تمرین و تست که شامل نام ها در زبان منتخب می باشد. هرچند که این چنین داده مرتبی به ندرت موجود می باشد. بستری که شامل کلمات مبدا که قرار است از مبداهای مختلف Transliterate شوند یک چالش برای سیستم های اتوماتیک می باشد. یکی از این چالش ها این است که کدام حرف را برای ارائه مبدا آن کلمه انتخاب کنیم. برای مثال وقتی می خواهیم کلمه “Josef” را به فارسی برگردان کنیم برای “J” می توانیم هم "ژ" که تلفظ فرانسه آن و هم "ج" که تلفـــــظ انگلیسی آن است را بکار بریم. همچنین وجود کلماتی که از یک زبان سومی وارد این زبان شده نیز باعث وارد کردن یک سری متـــــغیرها برای Transliteration کردن  **(KARIMI** 2008**)** یا رخداد خطا در صورت عدم تعریف مشخص از این متغییرهای جدید (Huang 2005; You 2008)، می شود. برای مثال هنگام Transliterate کردن کلمه “Amid” به فارسی که از عربی به انگلیسی Transliterate شده، با توجه به ریشه عربی کلمه، کاراکتر "ع" برای “A” اصولا باید انتخاب شود در حالی که برای کلمات فارسی کاراکتر "ا" برای “A” انتخاب می شود.

**تصمیم گیری روی Translation یا** **Transliteration.** تصمیــــم گیری روی Translation کردن یا Transliteration کردن یک اسم یا بخشی از یک اسم یک چالش بزرگ برای سیـــستم machine translation محسوب می شود.

اسم مکان ها و سازمان ها معمول ترین کلماتی هستند که روی آنها Translation و Transliteration اجرا می شود. برای مثال وقتی عبارت اسمی مثل “Alborz Mountain” در متن پیدا می شود، بخش اول باید Transliterate و بخش دوم Translate شود. یعنی بخشی از این عبارت اسمی باید Transliterate شود نه همه ی آن. مثال دیگر در مورد کلمات چند اسمی می باشد که امکان دارد بخشی از آن معنادار باشد ولی نباید Translate شود. زبان هایی مثل فارسی و عربی که اسم های خاص را علامت گذاری نمی کنند(بر خلاف زبانی مثل انگلیسی که با حرف بزرگ اسم خاص را علامت گذاری می کنند) برای اینچنین خطاهایی مستعدتر هستند. برای مثال اسم عربی مثل "عبدالحمید" (که تنها حالت Transliterate شدن آن “Abd al-hamid” می باشد) از دو بخش تشکیل شده که نباید بخش اول آن \_عبد\_ Translate شود. برخی مطالعه ها روی نوشته های مختلف (Al-Onaizan 2002; Chen 2006; Hermjakob 2008) این مشکل را بررسی کرده اند.

* در اینجا به بررسی روشی براساس مدل واجی برای Transliterate کردن از انگلیسی به فارسی (Sarvnaz Karimi 2009) می پردازیم. این روش برایعربی هم پیشنهاد داده شده (Larkey 2003) است. این روش با استفاده از کاراکترهای متنی بیشتر از کلمه مبدا S ارتقا داده شده و سپس یک مدل پیچیده تر برای کار با خصوصیات زبان فارسی معرفی شده است. بر روی یک بستر شامل 1857 اسم خاص، سیستم بهینه شده به دقت 66% رسیده است. این اولین روش موفق برای Transliterate کردن انگلیسی به فارسی در ادبیات، گزارش شده است.

1. **پیش زمینه :**

به طور کلی Transliteration های بر اساس واج، یک پروسه ی سه مرحله ای (Sarvnaz Karimi 2009) می باشند.

**انتخاب حروف الفبا.** در مرحله اول حروف الفبایی از سمبل های مبدا و مقصد، ∑S و ∑T ، از یک بستر آموزشی از جفت کلمه های مبدا به مقصد درست می شود. توجه کنید که سمبل ها می توانند کاراکترهای تکی یا گروهی از کاراکترها باشند. برای مثال جفت کاراکتر انگلیسی “CH” امکان دارد به عنوان یک تک سمبل در زبان مبدا باشد. سمبل های الفبایی می توانند از متدهای آماری شبیه مدل های پنهان مارکو (Sung Young Jung 2000) یا مدل های ترجمه ای (Peter F. Brown 1993) بدست آید. این ها در ابزار GIZA++ (Ney 2003) پیاده سازی شده اند که می توان از آن برای پژوهش های مختلف استفاده کرد.

**تولید احتمالات.** در مرحله دوم بستر آموزشی برای جمع آوری آمار بر روی فرکانس اتفاق یک سمبل الفبایی استفاده می شود. هدف کلی محاسبه ی احتمال P(ti|si , ci) که احتمال تولید شدن سمبل مقصد *ti ∈ ΣT به ازای* سمبل مبدا *si ∈ ΣS در متن* *ci که یک رشته* تولید شده از (*ΣT ∪ ΣS*) می باشد را بیان می کند. در ابتدایی ترین مدل هیچ متنی استفاده نمی شود پس *ci* = Ɛ که همان رشته خالی است و Transliteration به راحتی از احتمال *P*(*ti|si*) بدست می آید.

در تمامی متدهایی که از متن غیر تهی استفاده می کنند تدارکات برای عقب گرد به یک متن کوچک تر باید انجام شود. برای مثال اگر بخواهیم سمبل “O” در کلمه “LightHouse” در متن *ci* =*si−*4*si−*3*si−*2*si−*1 = “ghth” را Transliterate کنیم خیلی محتمل است که متن “ghth” در بستر اموزشی خیلی کم تکرار شده باشد. پس آمارهای بدست آمده از متن قابل اطمینان نیستند. در این مورد یک روش عقب گردی[[18]](#footnote-19) می توان از متن *ci* = *si−*3*si−*2*si−*1 =“hth” استفاده کرد که باز خود قابل اطمینان نیست، پس عقب گرد بعدی متن باید اجرا شود.

**Transliteration.** سومین مرحله از Transliteration کلمه مبدا را به سمبل هایی از ΣS تبدیل کرده و احتمال کلمه مبدا T را از فرمول زیر حساب می کند.



که اولین جمله P(T) یک مدل زبان مقصد است. برای مثال کلمه “zxqj” خیلی احتمال کمی برای تولید شدن در زبان انگلیسی از هر زبان دیگری دارد. پس P(“zxqj”) خیلی کوچک خواهد بود. اما در طرف دیگر احتمال P(“THE”) می تواند حدود یک صدم باشد.

کلمات مقصد می توانند توسط احتمال *P*(*T |S*) که در فرمول بالا آورده شد مرتب شوند که این خود یک لیست دارای ارزش برای Transliteration مربوط به s را تولید می کند.

1. **متد های Transliteration :**

در ابتدا سیستم های مبتنی بر خط را معرفی کرده سپسبه یک روش جدید (Sarvnaz Karimi 2009) (مدل حروف صدادار واپاشیده) برای Transliteration می پردازیم.

* 1. سیستم های مبتنی بر خط[[19]](#footnote-20) (Sarvnaz Karimi 2009) :

**انتخاب الفبا.** برای انتخاب الفبای مبدا و مقصد از تطبیق در سطح کاراکتر برای جفت کلمه های مبدا و مقصد تولید شده توسط GIZA++ (Ney 2003) استفاده می کنیم. اگر این کار یک گروه از کاراکترها را به یک کاراکتر خاص مربوط ساخت چه از مبدا به مقصد و چه برعکس،دراینصورت گروه را به عنوان یک تک سمبل به *ΣS یا ΣT ا ضافه می کنیم. به عنوان مثال کلمه ی “Stella” را* هنگام تطبیق دادن با فارسی به صورت زیر خواهیم داشت:



**تولید احتمالات.** Transliteration های قبلی عربی از متن خالی (Larkey 2003) استفاده می کردند، درحالیکه سیستم های موفق برای زبان های رومانتیک (که فارسی به این زبان ها نزدیک تر است) از *ci* = *si−*2*si−*1*si*+1 استفاده می کنند(Linden 2005). نمادگذاری n\m که در آن n سمبل قبلی مبدا و m سمبل بعدی از مبدا متن را می سازد که به صورت زیر تعریف می شود :



برای رفع حالات مرزی فرض می شود که S به اندازه کافی به چپ و راست توسط یک سمبل مخصوص که همیشه به یک سمبل خالی Transliterate می شود گسترش یافته است. بنابراین [*−n, . . . , |S|* + *m*] همگی ایندکس های قابل قبولی در S می باشند.

در ضمن تولید این احتمال تعریف می شود که :



هنگامی که تمامی فرکانس ها جمع آوری شد به راحتی به احتمال های نرمال شده تبدیل می شوند تا احتمال زیر محاسبه شود :





شکل 2 الگوریتم Transliteration

**Transliteration.** وقتی سمبل های مبدا به صورت کلمات متناظر در بستر آموزشی آماده هستند دراینصورت وقتی یک کلمه تنها از مبدا را برای Transliterate کردن داریم می توان چند روش برای تجزیه شدن با استفاده از الفبای مبدا خواهیم داشت. یکی از روش های ممکن این است که به صورت حریصانه[[20]](#footnote-21) طولانی ترین سمبل مبدا را از چپ به راست که تطبیق می کند را انتخاب کرده سپس از این تجزیه در طی Transliteration (Larkey 2003) استفاده کنیم. یک روش دیگر این است که مدل زبان مبدا را که احتمالات بر آن اعمال شده اند را تعریف کنیم و از این احتمال بر**ای** معادله زیر برای P(T|S) (Sung Young Jung 2000) استفاده کنیم.



در اینجا اعمال یک تک تجزیه[[21]](#footnote-22) استفاده نشده است بلکه مبدا را کاراکتر به کاراکتر پردازش می کنیم. همچنین پروسه ی عقب گرد برای متن را نیز کاراکتر به کاراکتر به جای سمبل به سمبل پایه گذاری می کنیم. توجه به این نکته که رد تعداد کاراکترهایی که در طولانی ترین متن از بخش تولید احتمالات تولید شده اند را نگه داریم ضروری است، این در حالی است که می دانیم این متن شامل n + m سمبل است که به l + r کاراکتر معادل خواهد شد که در آن l ماکزیمم تعداد کاراکترها در n سمبل گذشته و r ماکزیمم تعداد کاراکترها در m سمبل بعدی می باشد.

یک تعریف سطح بالا[[22]](#footnote-23) از الگوریتم Transliteration در شکل1 آورده شده است. تلاشی برای Transliterate کردن هر کدام از کاراکترها در طولانی ترین متن ممکن در حالی که خود این متن از دو طرف کاراکتر به کاراکتر کوتاه شده باشد به طوری که این متن حداقل M بار در داده آموزشی تکرار شده باشد، انجام می شود. وقتی که یک متن پیدا شد تمامی سمبل های ممکن مقصد ti  که از کاراکتر si در آن متن بدست می آیند، درون مجموعه ی X جمع می شوند. اگر متنی پیدا نشود یک هم ارزی بر روی کاراکترهای تکی بدون استفاده از متن صورت می گیرد. از آن جایی که |ΣT ||T| کلمه مقصد ممکن موجود است، عملا تولید کردن تمامی نوع ها همانگونه که توضیح داده شد امکان پذیر نیست. در این الگوریتم از k عدد از کوتاه ترین الگوریتم ها طبق Dijkstra (Eppstein 1998) استفاده می شودکه به صورت حریصانه پیشوندهای رشته های درون X را با بیشترین احتمال توسعه می دهند پس تضمین می کند که k عدد از Transliteration هایی با بیشترین امتیاز پیدا شده اند.

* 1. **مدل حروف صدادار واپاشیده[[23]](#footnote-24) (Sarvnaz Karimi 2009) :**

انتخاب کردن متن ها فقط از روی مدل n\m که توضیح داده شد خصوصیاتی که در مورد زبان مبدا و مقصد را می دانیم مد نظر نمی گیرد. برای مثال فارسی زبانان برای Transliterate کردن مصوت های دوآوایی از زبان دیگر ترجیح می دهند آن را به یک واج ساده تبدیل کنند. به علاوه صداهای کوتاه در زبان انگلیسی معمولا در فارسی حذف می شوند. این خصوصیت مدلی را برای تبدیل حروف صدادار انگلیسی در یک متن با کمک حروف بی صدا به فارسی پیشنهاد می دهد. این مدل می تواند در عوض استفاده از یک تکنیک آنالیزی که تماما مربوط به زبان می باشد(همانگونه که برای چینی پیاده سازی (Verspoor 1998) شده است) استفاده شود.

**انتخاب الفبا.** ما *ΣS و ΣT* را به همان صورت که در سیستم های مبتنی بر خط تعریف کردیم، تعریف می کنیم: سمبل های مرکب توسط هم تراز کردن کلمات که بر اساس مدل آماری ترجمه پیاده سازی شده در GIZA++ مشخص می شوند.

زمانی که این تجزیه کردن اولیه به پایان می رسد کلمات را بار دیگر بر اساس مرزهای حروف صدادار و بی صدا بخش بندی می کنیم که سمبل های جدیدی را تولید کنند که بتوان به الفبای موجود اضافه کرد. هر کلمه مبدا به یک رشته از حروف بی صدای تک تجزیه شده که با C نمایش داده می شود و حروف صدادار پشت سرهم توسط V نشان داده می شود سپس این ها به بخش هایی گروه بندی می شوند که هر کدام می توانند عضو یکی از گروه های زیر باشند (Sarvnaz Karimi 2009) :

1. VC، آمدن چند حروف صدادار در اول کلمه که به دنبال آن ها حروف بی صدا بیاید.
2. CC، دو حرف بی صدا که توسط هیچ حرف صداداری جدا نشده باشند.
3. CVC، آمدن چند حرف صدادار که توسط دو حرف بی صدا محدود شده باشد.
4. CV، آمدن چند حرف صدادار در انتهای کلمه که قبل آن یک حرف بی صدا آمده باشد.

اولین حرف بی صدا در هر بخش تداخل دارد با آخرین حرف بی صدای بخش قبلش. در هر صورت کل سمبل مرکب به الفبای مبدا اضافه می شود و سمبل مرکبی که در انتهایش حرف بی صدا ندارد به دو الفبا اضافه می شود. برای مثال کلمه “Baird” در نظر بگیرید که به صورت زیر تجزیه می شود:



که نتیجه آن اضافه شدن “ai” به *ΣS* می باشد.



شکل 3 مثالی از بخش ها، سمبل ها و متن هایی که برای کلمه مبدا "enrique" تولید می شود

در مرحله دوم کلمه به بخش هایی شبیه زیر تجزیه می شود :



که نتیجه آن ایجاد دو متن ممکن “bair” و “rd” می شود که به *ΣS* به صورت CVC و CC اضافه می شود. برای سمبل اضافی مبدا که “bai” می باشد از مشتق کردن CVC که با افتادن C به دست می آید به *ΣS*  اضافه می شود و “r” به صورت پیش فرض در *ΣS*  موجود است. در الفبای مقصد **ب ی**، به صورت یک CVC کوتاه شده به عنوان یک سمبل مقصد و تک کاراکتر "ر" که قبلا موجود است به *ΣT* اضافه می شوند.

**تولید احتمالات.** احتمالات بر اساس تعداد تکرارهای موجود در داده آموزشی همانند سیستم های میتنی بر خط حساب می شود اما متن ها بر اساس بخش ها انتخاب شده اند نسبت به اینکه در سیستم قبلی بر اساس سمبل ها انتخاب می شدند. از دو تکنیک برای انتخاب متون استفاده می شود. در اولین تکنیک که CV نامیده می شود هر بخش یک متن را برای یک سمبل تولید می کند که خود آن سمبل برای یک بخش پیشوند می شود. برای مثال چهارمین ستون از شکل 3 نشان می دهد که چگونه متن ها و سمبل ها برای کلمه مبدا تولید می شوند.

در دومین تکنیک برای انتخاب متون که CV-BROAD نامیده می شود تطابق های سخت گیرانه[[24]](#footnote-25) بر روی اجزای بی صدا را می کاهیم[[25]](#footnote-26) که بتواند تمام حروف بی صدا را تطابق دهد که خود باعث مقدار داده تمرینی برای یک متن می شود. به نوبت خود این باعث حالت کم ندرتی که در هر صورت امکان پذیر است می شود که در آن Transliteration ها امکان پذیر نمی باشد. حروف صدادار فقط در متونی استفاده می شوند که Transliterate کردن یک سمبل شامل یک حرف صدادار باشد که در ستون آخر شکل 3 نمایش داده شده است.

روش عقب گرد برای هر کدام از 4 بخش ذکرشده توسط 3 قانون (Sarvnaz Karimi 2009) زیر بیان می شود که به ترتیب اجرا می شوند :

1. اگر متن یک بخش CV باشد و حداقل M بار در داده تمرینی اتفاق نیفتاده باشد، سمبل را به دو بخش C سپس V در متن جدا می کنیم. در غیر اینصورت،
2. اگر متن یک V می باشد که نماینگر آمدن r حرف صدادار است و حداقل M در داده تمرینی تکرار نشده باشد طول V را به r-1 تقلیل داده و آخرین حرف صدادار را به صورت یک کاراکتر صدادار مجزا بدون متن Transliterate می کنیم. در غیر اینصورت،
3. برای تمامی متن های دیگر اگر کمتر از M بار در داده تمرینی تکرار شده باشند C انتهایی را حذف کرده و از این متن کوتاهتر استفاده می کنیم.

**Transliteration.** پروسه Transliteration دقیقا به همان صورت که در سیستم های مبتنی بر خط توضیح داده شد انجام می شود که از الگوریتم کوتاه ترین k مسیر دایجسترا استفاده می شود برای تولید k عدد از Transliteration هایی که بیشترین امتیاز را دارند.

**6. آزمایش ها بر روی این دو روش (Sarvnaz Karimi 2009) :**

در این بخش محیط آزمایشی[[26]](#footnote-27) از جمله داده های استفاده شده برای آموزش و تست، مدل های Transliteration و معیار های سنجش توضیح داده می شوند.

**6.1. داده ها**

دو مرحله اول از Transliteration به جفت کلمه هایی از مبدا و مقصد نیاز دارد. یک بستر انگلیسی-فارسی توسط 40000 عبارت از اسم های انگلیسی که در اینترنت استفاده می شود تولید شد. این اسم ها، اسم های مربوط به مکان های جغرافیایی، اشخاص و شرکت ها می باشند که از منابع مختلف عمومی استخراج شده اند. این بستر انگلیسی توسط 26 فرد فارسی زبان Transliterate شدند. کاراکترهای اضافی و چندآوایی معمولا برای Transliterate کردن کلمات عربی در این مجموعه استفاده شد. بر اساس تخمین های مختلف که توسط افراد مختلف زده شده است امکان دارد برای یک کلمه چند Transliteration مختلف انتخاب شده باشد. برای مثال اگر فرد مورد نظر فرض کرده باشد که عبارت “John Pierre” فرانسوی است امکان دارد برای کاراکتر “j” حرف فارسی "ژ" را استفاده کرده باشد و اگر فرد این عبارت را انگلیسی فرض کرده باشد می تواند از "ج" استفاده کند. در نهایت این بستر به کلمات مجزا تقسیم شد که 16952 جفت کلمه منحصر به فرد را تولید کرد که به این بستر جدید ENGLISH+ می گوییم.

پروسه انتخاب این بستر تضمینی برای انتخاب شدن کلمات با اصلیت انگلیسی ندارد. در نتیجه تعداد زیادی کلمه در این بستر وجود دارد که اصلیتی فرانسوی، عربی یا چینی دارند. به صور اتفاقی یک زیرمجموعه 2000 عضوی از جفت اسم های مجموعه ENGLISH+ انتخاب شده اند که در آن کل اسم هایی که اصلیت انگلیسی نداشتن حذف شده اند که این عمل باعث تولید یک زیرمجموعه 1857 عضوی شد که آن را ENGLISH می نامیم.

در انجام آزمایشات 10 مرحله الگوریتم اجرا شده که در آن هر بار از 90% مجموعه به عنوان داده تمرینی و از 10% باقی مانده به عنوان داده تست استفاده شده است و نتایج میانگین این 10 بار اجرا می باشد.

**6.2. معیار های سنجش**

نتایج آزمایشات بر روی Transliteration ها بر اساس معیار استاندارد برای دقت کلمه اعلام می شود که به معنای فاصله ی بین کلمه ی Transliterate شده و کلمه درست می باشد. این معیار فاصله ای تعداد کاراکترهایی که باید اضافه، حذف یا جایگزین شوند تا به کلمه درست برسیم محاسبه می شود.

دقت کلمه[[27]](#footnote-28) (WA) که به عنوان دقت Transliteration نیز شناخته می شود به صورت زیر تعریف می شود :



که در آن ED فاصله بین دو رشته می باشد (Dowling 1980).WA برای مقادیر مقطعی مختلف گزارش می شود. به عنوان مثال Top-1 سهم هر یک از کلمات را در مجموعه تست به طوری که Transliteration درست اولین جواب نامزد تولید شده توسط سیستم باشد، نشان می دهد. در حالی که Top-5 سهم کلماتی را نشان می دهد که جواب درست در 5 جواب نامزد اولی بوده است.

**6.3. نتایج**

در این بخش نتایج آزمایش ها بر اساس اندازه متن، مدل زبان مقصد و مدل حروف صدادار واپاشیده اعلام می شود. در جداولی که در ادامه متن می آید نتایج مربوط به متدهایی که بیشترین دقت را دارند به صورت برجسته آورده می شوند.

**متن قبلی و بعدی.** نتایج مربوط به استفاده از n متن قبلی و m متن بعدی (متغیر) در جدول 1 آورده شده است. بیشترین دقت توسط یک متن به حالت 1\0 که از یک تاریخچه یک کاراکتری استفاده می کند. تفاوت بین استفاده از یک متن به این صورت با استفاده کردن از یک متن تهی به صورت آماری بسیار فاحش می باشد.



جدول 1 دقت کلمات بدست آمده(%) بر اساس تغییر اندازه متن های قبلی و بعدی

**مدل های زبان مقصد.** تاثیر تعریف یک مدل برای زبان مقصد در جدول 2 آورده شده است. طرح هایی که از یک مدل زبان مقصد ترتیب-اول استفاده می کنند توسط یک “T” علامت گذاری شده اند.

**روش های حروف صدادار واپاشیده.** نتایج مربوط به روش جدیدی که به عنوان حروف صدادار واپاشیده از آن یاد می کنیم در ستون آخر جدول 2 آورده شده اند که با CV و CV-BROAD علامت گذاری شده اند. نمای CV بهترین نتیجه را برای Transliterationهای Top-1 مربوط به مجموعه های ENGLISH و ENGLISH+ نشان می دهد. یک پیشرفت آماری چشم گیری نسبت به روش مبتنی بر خط با حالت 1\0 ایجاد شده است.



جدول 2 دقت کلمات (%) وقتی از یک مدل برای یک زبان مقصد (T) و مدل حروف صدادار واپاشی شده (CV و CV-BROAD)

**تاثیرات مبتنی بر زبان[[28]](#footnote-29).** تفاوت بزرگی که میان نتایج مربوط به مجموعه های ENGLISH و ENGLISH+ وجود دارد، این را به ذهن می آورد که امکان دارد مجموعه ای بزرگتر برای آموزش بتواند به پروسه Transliteration ما کمک کند. اصلیت زبان مبدا می تواند تاثیر زیان بار بزرگی بر روی دقت داشته باشد. در شکل زیر دقت را برای روش CV که بر روی زیرمجموعه هایی از ENGLISH+ استخراج شده اند، نشان می دهد. یک کلمه به یک کشور خاص بر اساس تعداد دفعات آورده شده در صفحات اینترنتی مربوط به آن کشور، ربط داده شده است. برای مثال کلمه “Groenendijk” بیشتر در صفحات اینترنتی با پسوند .nl آورده شده است بنابراین این کلمه را به کشور آلمان ربط داده می شود. دقت کنید که دقت برای زبان های مستقل و همه زبان ها کاهش بسزایی داشته است.



شکل 4 اصلیت های مختلف زبان

**تاثیرات اندازه مجموعه ها.** یکی از دلایل برای دقت کم CV بر روی داده های مبتنی بر کشور می تواند کوچک بودن مجموعه های داده ای باشد. برای تحقیق در مورد این موضوع به صورت اتفاقی زیر مجموعه هایی کوچک از ENGLISH را تولید کرده و کم کم اندازه آن ها را افزایش داده و دقت مربوط به CV را بدست می آوریم. نتایج نشان می دهند که هر مجموعه داده ای که اندازه اش از 200 بیشتر باشد عملکرد خوبی داشته است. پس می توان نتیجه گرفت که کوچکی مجموعه های داده ای مربوط به کشورهای مختلف باعث ایجاد اختلال نمی شوند. این نتیجه نشان می دهد که عملکرد CV بر روی داده هایی که اصلیتا انگلیسی می باشند بهتر کار کرده در مقابل با داده هایی که قبلا از یک زبان دیگر به انگلیسی Transliterate شده اند.



شکل 5 تاثیرات اندازه مجموعه ها

**7. نتیجه گیری :**

در این مقاله ابتدا مفهوم های پایه ای مربوط به Transliteration را بیان کردیم، سپس به بررسی تعدادی از تکنیک های Transliteration و تاثیرگذاری این روش ها برای تبدیل کلمات از انگلیسی به فارسی پرداختیم. به بررسی عملکرد روش های مبتنی بر واج پرداخته شد و یک بازه مربوط به متن و مدل برای زبان مقصد امتحان کردیم. استفاده از متن امری مهم است بگونه ای که استفاده از یک متن تاریخچه ای کوچک تاثیر بسزایی در عملکرد سیستم دارد. این تاثیر غافلگیر کننده بود چون پیش بینی می شد که اگر این متن را بزرگتر در نظر بگیریم عملکرد بهتر خواهد شد اما اینگونه نبود.

یک مدل جدید برای Transliteration انگلیسی به فارسی بر اساس حروف صدادار واپاشیده معرفی و بررسی شد. این تکنیک جدید دقت را نسبت به روش مبتنی بر واج استاندارد حدود 18% بر روی مجموعه های اطلاعاتی که داده هایش اصلیت انگلیسی دارد و 41% بر روی مجموعه ی اطلاعاتی که عمومی تر است.

منابع :

Al-Onaizan, Y. a. K., K. (2002). "Machine transliteration of names in Arabic text. In Proceedings of the ACL workshop on Computational approaches to semitic languages." Philadelphia, PA, 1–13.

Chen, C. a. C., H.-H (2006). "A high-accurate Chinese-English NE backward translation system combining both lexical information and web statistics. In Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the ACL on Main Conference Poster Sessions." Sydney, Australia, 81–88.

DALE, R. (2007). "Language Technology." In Slides of HCSNet Summer School Course.

Dowling, P. A. V. H. a. G. R. (1980). "Approximate string matching." ACM Comput. Surv., 12(4):381–402.

Eppstein, D. (1998). "Finding the k shortest paths." SIAM J. Computing, 28(2):652–673.

Hermjakob, U., Knight, K., and III, H. D (2008). "Name translation in statistical machine translation - learning when to transliterate. In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human language Technologies." Columbus, OH, 389–397.

Huang, F. (2005). "Cluster-specific named entity transliteration. In Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing." Vancouver, Canada.

KARIMI, S. (2008). "Machine Transliteration of proper names between English and Persian. Ph.D. thesis " RMIT Univesity, Melbourne, Australia.

KARIMI, S. S., F. TURPIN, A (2009). "Machine Transliteration Survey." (translitration): 57.

Larkey, N. A. a. L. S. (2003). "Statistical transliteration for English-Arabic cross language information retrieval." In CIKM, pages 139–146.

Linden, K. (2005). "Multilingual modeling of cross-lingual spelling variants." Inf. Retrieval, 9(3):295–310.

Ney, F. J. O. a. H. (2003). "A systematic comparison of various statistical alignment models." Computational Linguistics, 29(1):19–51.

Peter F. Brown, V. J. D. P., Stephen A. Della Pietra, and Robert L.Mercer (1993). "The mathematics of statistical machine translation: Parameter estimation." Computional Linguistics, 19(2):263–311.

Sarvnaz Karimi, A. T., and Falk Scholer (2009). "English to Persian Transliteration." School of Computer Science and Information Technology RMIT University, GPO Box 2476V, Melbourne 3001, Australia.

Sung Young Jung, S. L. H., and Eunok Paek (2000). "An English to Korean transliteration model of extended markov window." In COLING, pages 383–389.

Verspoor, S. a. C. (1998). "Automatic English-Chinese name transliteration for development of multilingual resources." In COLING-ACL, pages 1352–1356.

You, J.-L., Chen, Y.-N., Chu, M., Soong, F., and Wang, J.-L. (2008). "Identifying language origin of named entity with multiple information sources." IEEE Transactions on Audio, Speech, and Language Processing 16, 6, 1077–1086.

1. Extended [↑](#footnote-ref-2)
2. MT [↑](#footnote-ref-3)
3. Multilingual [↑](#footnote-ref-4)
4. <http://www.omniglot.com/writing/ipa.htm> [↑](#footnote-ref-5)
5. *Phonetics and Phonology* [↑](#footnote-ref-6)
6. Phonetic representation [↑](#footnote-ref-7)
7. *Phoneme* [↑](#footnote-ref-8)
8. *Grapheme* [↑](#footnote-ref-9)
9. *Syllable* [↑](#footnote-ref-10)
10. *Writing system* [↑](#footnote-ref-11)
11. *Script* [↑](#footnote-ref-12)
12. Script Specifications [↑](#footnote-ref-13)
13. Missing Sounds [↑](#footnote-ref-14)
14. Transliteration Variants [↑](#footnote-ref-15)
15. Language of Origin [↑](#footnote-ref-16)
16. Hiragana, Katakana and Kanji [↑](#footnote-ref-17)
17. Corpus [↑](#footnote-ref-18)
18. Backoff [↑](#footnote-ref-19)
19. Baseline Systems [↑](#footnote-ref-20)
20. Greedy [↑](#footnote-ref-21)
21. Single Parsing [↑](#footnote-ref-22)
22. High level [↑](#footnote-ref-23)
23. Collapsed-Vowel Models [↑](#footnote-ref-24)
24. Strict matching [↑](#footnote-ref-25)
25. Relaxe [↑](#footnote-ref-26)
26. Experimental framework [↑](#footnote-ref-27)
27. Word Accuracy [↑](#footnote-ref-28)
28. Language-based Effects [↑](#footnote-ref-29)