رشته کامپیوتر

گرایش تکنولوژی نرم‌افزار

نرم افزار موبایل تبدیل متن فارسی به صوت

نگارنده

احمد منوچهری

استاد راهنما

دکتر محمد طاهری

تابستان ۱۳۹۸

به نام دوست

چکیده

با توجه به کاربرد های بسیار زیاد سیستم های تبدیل متن به گفتار در همه زبان های دنیا ، و نظر به تعداد کم سیستم های اینچنین برای زبان فارسی، تحقیقاتی در رابطه با چگونگی پیاده سازی این سیستم ها در زبان های غربی (به خصوص زبان انگلیسی) انجام گرفت و سعی بر آن شد تا با در نظر گرفتن ویژگی های منحصر به فرد زبان فارسی یک سیستم اولیه طراحی و پیاده سازی شود.

بررسی مقالات مختلف و نحوه عملکرد سیستم تکلم در انسان به این دسته بندی منتهی شد که سیستم های خواننده متن به طور کلی به دو دسته «سنتز پیوندی» - که بر پایه صدای ضبط شده توسط انسان برنامه ریزی می شود – و همچنین «سنتز قالبی» - که بر پایه ایجاد فرکانس های صوتی توسط سیستم کامپیوتری عمل می‌کند – قابل پیاده سازی است.

با توجه به اینکه «سنتز پیوندی» به علت استفاده از صدای ضبط شده انسانی ، نزدیک ترین خروجی به صدای انسان را دارد ، مبنای این پروژه قرار گرفته است. برای این منظور با مطالعه مباحثی از علم «واج شناسی در زبان فارسی» به چارچوبی دست یافته شد که به واسطه آن ترکیبی از دوحرفی های «صامت+مصوت» و صداهای خاص «مصوت» و صدا های کوتاه «صامت» قابلیت تولید دامنه بیشماری از کلمات و جملات در زبان فارسی را دارند.

برای محقق ساختن این چارچوب تعداد ۱۶۷ صدا به شرح ۶ مصوت ، ۲۳ صامت و ۱۳۸ صدای مرکب «صامت+مصوت» با دوسرعت آهسته برای تکلم شمرده و سریع ضبط شد. همچنین دو صدای فاصله کوتاه و بلند نیز اضافه شد که جمعا ۱۶۹ صدای ضبط شده را آماده عملیات سنتز پیوندی نمود..

تحقیقات انجام شده نیاز به دارا بودن فرهنگ تلفظ لغات را بدیهی نمود که به این منظور با به کارگیری منبع کدهای NLP و خروجی آنها ۵۰ هزار کلمه پرکاربرد در محاوره فارسی استخراج شد و برای تست اولیه پر کاربرد ترین ۵۰۰ لغت اول آن آوانگاری شد.

نتیجه روال طی شده دستیابی به سیستمی است که توانایی خواندن جملات نامحدود در زبان فارسی را دارد و با افزایش دامنه لغات آن تلفظ صحیح کلمات در آن به سادگی قابل بهبود است. همچنین با اعمال تنظیمات مختلف در مرحله سنتز پیوندی و ضبط صدای حرفه ای توسط صداپیشگان امکان بهبود صدای خروجی آن نیز قابل دستیابی است.

کلید واژه ها: تبدیل متن به صوت – سنتز پیوندی – واج شناسی فارسی – نرم افزار گویای فارسی

فهرست مطالب

مقدمه **۱**

بخش اول: مراحل تبدیل نوشتار به گفتار فارسی **۲**

۱.۱ تشریح وظایف هر یک از مراحل **۳**

۱.۱.۱ دریافت کننده جملات ۳

۱.۱.۲ نرمال کننده کلمات ۳

۱.۱.۳ دیکشنری تلفظ ها ۳

۱.۱.۴ قواعد تبدیل کلمه به آوانگاری ٤

۱.۱.۴.۱ شرح قواعد الگوریتم تبدیل کلمه به آوانگاری ٥

۱.۱.۴.۲ استاندارد آوانگاری فارسی ایرانیک ٦

۱.۱.۵ قوانین تکیه صدا ۸

۱.۱.۶ تولید کننده صدا ۸

۱.۱.۷ پخش کننده صدا ۱۰

بخش دوم : پیاده سازی نرم افزار و توضیحات فنی **۱۱**

۱.۲ محیط پیاده سازی و تکنولوژی ها **۱۱**

1.2.1 زبان برنامه نویسی JavaScript و تکنولوژی های React و ReactNative ۱۱

۱.۲.۲ روش ضبط صدا و انتخاب فرمت wav در مقابل دیگر فرمت های موجود ۱۲

۱.۲.۳ پیاده سازی سنتز پیوندی با FFMpeg و mobile-ffmpeg ۱۲

۱.۲.۴ مراحل طراحی و پیاده سازی ۱٤

۱.۲.۵ پیکر بندی پروژه ۱٦

۱.۲.۶ راه اندازی و کامپایل اولیه پروژه ۱۷

۱.۳ کد منبع و توسعه بیشتر **۱۸**

منابع ..... **I**

فهرست شکل ها

شکل ۱ - مراحل تبدیل متن به صوت فارسی ۲

شکل ۲- مراحل تولید صدای سنتز پیوندی ۹

شکل3 - نحوه عملکرد فلیتر پیوند ۱۳

شکل ۴ - رابط کاربری و جریان کاربر ۱۵

فهرست جدول ها

جدول ۱ - مصوت های زبان فارسی ٦

جدول ۲- صامت های هم صدای زبان فارسی ۷

جدول ۳ - waveform هجاها ۹

جدول ۴ - نحوه پیوند هجاها ۱۰

فهرست قطعه کد ها

قطعه کد ۱ ..... ٤

قطعه کد ۲ – نحوه فراخوانی سنتز پیوندی در FFMpeg ۱۳

مقدمه

سیستم های تبدیل متن به گفتار[1] از دیرباز کاربرد های بسیار فراوانی داشته اند و به انحاء مختلف مورد تحلیل ، بررسی و پیاده سازی قرار گرفته اند.

این سیستم ها موارد استفاده بسیار زیادی از جمله در بیمارستان ها ، نرم افزار های کمک به نابینایان ، سیستم های ترجمه ، تلفظ لغات ، تلفن های گویا و ... داشته و دارند.

علی‌رغم وجود مدل های علمی و تجاری بسیار پیشرفته و پیچیده برای زبان انگلیسی و زبان های دارای خط نوشتاری لاتین ، متاسفانه مدل های علمی و تجاری کمی برای زبان فارسی وجود دارند و این مساله نگارنده را بر آن داشت تا تحقیقی هرچند خرد در راستای پیاده سازی این سیستم انجام دهد.

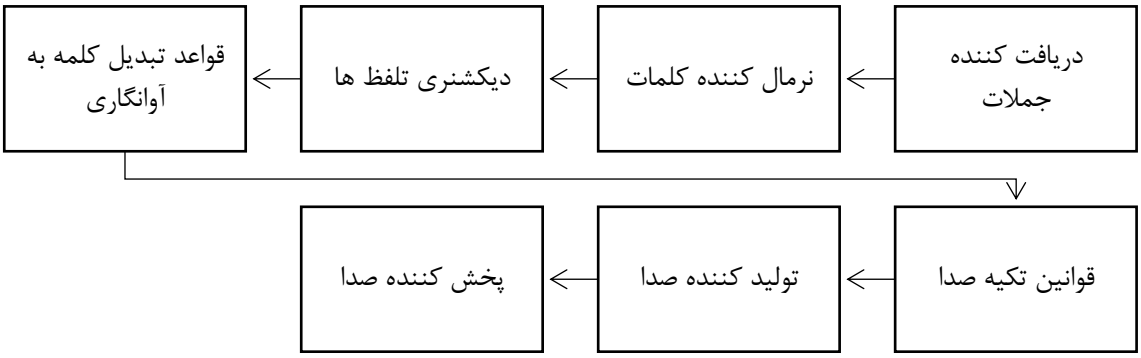
متن پیش رو بر آن است روش تحقیق و اجرای کامل نمونه ساده ای از یک سیستم تبدیل متن فارسی به گفتار فارسی را بصورت تشریحی به رشته تحریر درآورد.

[1] Text to Speech (TTS)

بخش اول: مراحل تبدیل نوشتار به گفتار فارسی

تبدیل یک متن فارسی به گفتار به طور کلی می تواند از مراحل مختلفی تشکیل شود. یک عامل انسانی به طور معمول برای صحبت کردن با عبور هوای موجود در ریه ها از تارهای صوتی و ترکیب آنها با شکل دهان ، دندان ها و بینی اصوات را ترکیب کرده و تولید صدا میکند.

با استناد بر تحقیقات پایه ای تکنولوژی های صحبتی بکرلی [1] ، یک سیستم تبدیل نوشتار به متن می تواند از مراحل توضیح داده شده در شکل ۱ - مراحل تبدیل متن به صوت فارسی تشکیل شود که برای بهینگی هر یک از آنها ساختمان داده و الگوریتم های متناسب در این پروژه پیشنهاد و پیاده سازی شده اند.



شکل ۱ ـ مراحل تبدیل متن به صوت فارسی

از آنجا که ساز و کار یک نرم‌افزار کامپیوتری با فرکانس های صوتی دیجیتال است، طبق تحقیقات به عمل آمده [2] روش تولید صوت در آن می تواند به یکی از روش های «سنتز پیوندی[۲]» یا «سنتز قالبی[۳]» صورت گیرد.

از آنجا که سنتز قالبی نیازمند تولید فرکانس های صوتی بصورت پویا می باشد و صداهای تولید شده در آن شباهت کمتری به صدای انسانی دارند، تحقیقات موضوع این پروژه بر مبنای سنتز پیوندی استوار گشته است.

[۲] Concatenative synthesis
[۳] Formant synthesis

۱/۱ تشریح وظایف هر یک از مراحل

برای رفع ابهام عناوین مطرح شده در شکل ۱ - مراحل تبدیل متن به صوت فارسی و همچنین نحوه اعمال آنها در نرم‌افزار اجرا شده این پروژه در ادامه به تشریح خلاصه ای از وظایف و نحوه پیاده سازی آنها در پروژه می پرازیم.

۱/۱/۱ دریافت کننده جملات

مراد از این بخش هر روشی از ورودی خروجی است که جملات را بصورت متن از کاربر دریافت کنند. در پروژه موبایل قطعا این اتفاق به کمک یک فیلد ورود متن[۴] شکل می‌گیرد. جملات بصورت رشته ای از کاراکتر ها به سیستم داده می شوند و نرم‌افزار باید بتواند وظایف زیر را روی آنها انجام دهد:

۱. حذف حروف ناخوانا ، علائم و نشانه ها از رشته کاراکتر ها
۲. تبدیل علائم خط بعد ، نیم خط بعد ، نیم فاصله و غیره به فاصله استاندارد
۳. شکستن رشته کاراکتر ها به آرایه ای از کلمات مجزا
۴. ارسال کلمات عضو آرایه منتج از مرحله قبل به مراحل بعد

۱/۱/۲ نرمال کننده کلمات

این بخش در زبان هایی مثل انگلیسی کلماتی همچون .Mr را به Mister تبدیل میکند. با توجه به اینکه در زبان فارسی استفاده از مخفف ها چندان مرسوم نیست، وجود چنین بخشی در این پروژه الزامی به نظر نمی‌رسد. با این حال در صورت نیاز کاربر میتواند با استفاده از امکانات تعبیه شده در بخش دیکشنری تلفظ ها این نیاز را مرتفع سازد.

۱/۱/۳ دیکشنری تلفظ ها

با توجه به اینکه در هر زبانی در موارد بسیار زیادی تلفظ کلمات با حالت نوشتاری آنها متفاوت است می بایست یک دیکشنری در سیستم موجود باشد که کلمات قبل از عبور از قواعد تبدیل کلمه به آوانگاری در آن جستجو شود. همچنین برای پویایی نرم افزار ، می بایست دیکشنری مخصوص به کاربر[۵]

---

[۴] Text Field
[۵] User-specific dictionary

نیز وجود داشته باشد که کاربر توسط آن قادر باشد کلمات پر تکرار خود را به آن افزوده و تلفظ صحیح آن را به نرم افزار یاد دهد. بهترین ساختمان داده برای این منظور میتواند یک دیتابیس قابل حمل مثل SQLite و یا Realm باشد.

در این پروژه برای دیکشنری داخلی نرم افزار ، به منظور استفاده از پر کاربردترین کلمات ، در نتیجه انجام تحقیقات مختلف از خروجی کتابخانه NLP تحت عنوان FrequencyWords [3] استفاده شد. این کتابخانه تعداد نزدیک به ۳ میلیون زیرنویس فیلم مشتمل بر بیشتر از ۷ میلیون جمله فارسی [4] را پردازش کرده و لیست ۵۰ هزار کلمات پرکاربرد را بر اساس تعداد استفاده مرتب می کند. در حال حاظر کلیه نتایج برای تمامی زبان های دنیا در منبع-کد[۶] این کتابخانه موجود می‌باشد.

برای تست اولیه این پروژه تعداد ۵۰۰ کلمه اول این لیست به استفاده از قواعد ذکر شده در بخش قواعد تبدیل کلمه به آوانگاری بصورت دستی آوانگاری شدند و در یک فایل JSON با ساختمان داده Hashmap قرار گرفتند تا با بهینگی O(1) بتوان به اطلاعات آن دسترسی پیدا کرد.

۱/۱/۴ قواعد تبدیل کلمه به آوانگاری

نظر به اینکه ساخت یک دیکشنری که در برگیرنده تمامی کلمات یک زبان و تلفظ صحیح آنها باشد – علی‌الخصوص با توجه به پویایی زبان و زایش کلمات نو در آن – کاری امکان ناپذیر است ، می بایست قواعدی وجود داشته باشند تا بر مبنای تجربیات یک بومی زبان[۷] بتواند کلمات جدید را تبدیل به آوانگاری کند.

قواعد آوانگاری استفاده شده در این پروژه به شکل زیر عمل میکند:

عملکرد سیستم پس از آزمون و خطاهای گونانون و تحقیقات مختلف بصورت ابتکاری به این نحو رسیده است که ترکیب های دو حرفی متشکل از «صامت+مصوت» از کلمات ساخته می شوند و در آرایه پشت سر هم قرار میگیرند. برای مثال به جمله زیر در قطعه کد ۱ توجه بفرمایید:

```
1  const input = "سلام"
2  const output = ["sa", "lā", "m"]
```

قطعه کد ۱

[۶] Repository
[۷] Native Speaker

با پیاده سازی الگوریتم به نحوی که ورودی و خروجی آن به شکل بالا باشد ، میتوان با مجموع ۱۶۷ صدای مرکب از دو حرفی های صامت+مصوت ، تمام کلمات ممکن در زبان فارسی را پوشش داد و جملات مختلف را به کمک آن بصورت صوتی تولید نمود.

۱/۱/۴/۱ شرح قواعد الگوریتم تبدیل کلمه به آوانگاری

۱. به کمک یک حلقه تک تک کاراکتر های کلمه را میخواند
۲. اگر کاراکتر اول (اندیس صفر) الف باشد ، با در نظر گرفتن کاراکتر بعد از خود نگاشت های استثناء زیر رخ می دهد:
   a. اگر کاراکتر «آ» کلاه دار باشد ، نگاشت به حرف ā صورت میگیرد.
   b. اگر کاراکتر بعد «و» باشد صدای «او» با کلید u نگاشت می‌شود.
   c. اگر کاراکتر بعد «ی» باشد ، صدای «ای» با کلید i نگاشت می‌گردد
   d. اگر کاراکتر بعد ، کاراکتر شفاف[۸] «ـُ» باشد ، کلید o نگاشت می‌گردد.
   e. و نهایتا اگر کاراکتر بعد صامت باشد(حالت پیش‌فرض) صدای «اَ» با کلید a نگاشت میگردد.
۳. اگر کاراکتر یکی از صامت های زبان فارسی باشد کاراکتر بعدی آن بررسی می شود:
   a. اگر کاراکتر بعدی مصوت باشد ، دوحرفی «صامت+مصوت» برای آن تولید شده و در آرایه push می‌شود.
   b. اگر کاراکتر بعدی نیز صامت باشد، بطور پیش فرض صامت فعلی به مصوت ـَ جمع شده و در آرایه قرار داده می شود.
   c. حالت خاص برای کاراکتر بعد، حرف «ه» آخر است که صامت فعلی را با مصوت ـِ جمع میزند و این حالت نیز در حلقه مورد بررسی قرار میگیرد.
   d. حالت خاص دیگر وجود کاراکتر شفاف ساکن ـْ به عنوان کاراکتر بعد است ، که در این حالت نیست الگوریتم به جای ترکیب دوتایی صامت+مصوت تنها تک مصوت فعلی را در آرایه قرار می دهد.

پس از انجام تمامی مراحل و اعمال قواعد ذکر شده در بالا ، نهایتا آرایه ای مشابه آرایه مثال در . خروجی الگوریتم خواهد بود که با پیمایش آن می توان به فایل های صوتی دسترسی یافت و آنها را به روش سنتز پیوندی با هدف تولید جملات منسجم و به هم پیوسته با یکدیگر ترکیب کرد.

---

[۸] Transparent character

۱/۱/۴/۲ استاندارد آوانگاری فارسی ایرانیک

نظر به اینکه تلفظ لغات در زبان های مختلف با حالت نوشتاری آنها در اغلب موارد متفاوت است ، در علم زبان شناسی بحث الفبای آوانگاری[۹] مطرح میشود. این سیستم الفبایی که برپایه رسم الخط لاتین بیانگزاری شده [5] معمولا در آوانگاری های نوشته شده در فرهنگ لغات همه زبان های دنیا به کار گرفته شده است.

با وجود اینکه استفاده از استاندارد IPA در زبان فارسی موافقان و مخالفان بسیاری دارد ، تحقیقات اخیر استاندارد خلاصه تر و ساده تری به نام استاندارد ایرانیک[۱۰] [6] یا به اختصار IRPA ارائه نموده است. در پروژه پیش رو نتیجه این تحقیقات برای یافتن مصوت ها و صامت های زبان فارسی به کار گرفته شده و خلاصه مهمترین بخش این تحقیقات به شکل جدول ۱ - مصوت های زبان فارسی و جدول ۲- صامت های هم صدای زبان فارسی مبنای تقسیم بندی مصوت و صامت های این پروژه می باشد:

نام مصوت	سمبل معادل در IRPA	کد Unicode
«َ» مثل اَبر ، بَرف	A a	0x064e
«آ» مثل آب ، باد	Ā ā	0x627
«ُ» مثل اُردو	O o	0x064f
«او» مثل کودک	U u	0x0648
«ِ» مثل استخر	E e	0x064e
«ای» مثل ایران	I i	0x06cc

جدول ۱ - مصوت های زبان فارسی

[۹] International Phonetic Alphabet (IPA)
[۱۰] IRanic Phonetic Alphabet (IRPA)

نام صامت هم‌صدا	سمبل معادل در IRPA	کد Unicode
«ب» مثل باد	B b	0x0061
«پ» مثل پروانه	P p	0x0070
«ت» و «ط» مثل تیر	T t	0x0074
«ج» مثل جشن	J j	0x006A
«چ» مثل چراغ	Č č	0x010D
«ح» و «هـ» مثل هفت	H h	0x0068
«خ» مثل خرداد	X x	0x0078
«د» مثل دادگر	D d	0x0064
«ر» مثل رود	R r	0x0072
«ز» ، «ذ» ، «ض» و «ظ» مثل زبان	Z z	0x007A
«ژ» مثل ژاله	Ž ž	0x017E
«س» ، «ث» و «ص» مثل سارا	S s	0x0073
«ع» و «ئـ» مثل عائد	Ø ø	0x00F8
«ع» و «ئـ» مثل عائد	Ø ø	0x00F8
«غ» و «ق» مثل عائد	Ğ ğ	0x011F
«ف» مثل فردا	F f	0x0066
«ک» مثل کتاب	K k	0x006B
«گ» مثل گلاب	G g	0x0067
«ل» مثل لیوان	L l	0x006C
«م» مثل ماه	M m	0x006D
«ن» مثل نان	N n	0x006E
«و» مثل وال	V v	0x0076
«ی» مثل یار	Y y	0x0079

جدول ۲- صامت های هم صدای زبان فارسی

پیاده سازی جداول فوق در پروژه پیش رو توسط دو Hashmap به نام های vowels و consonants صورت میگیرد که هرکدام از آنها کاراکتر را به Unicode معادل آن در استاندارد IRPA نگاشت می‌کند.

۱/۱/۵ قوانین تکیه صدا

با توجه به اینکه در مکالمات روزمره نسبت به پرسشی ، سوالی ، حالت شادی ، غم ، خشم و ... تکیه صدا[۱۱] و نحوه بیان[۱۲] و زیر و بمی صدا[۱۳] جملات دستخوش تغییر می‌شود ، حالت ایده آل برای یک سیستم کامپیوتری این است که بتواند این حالات را نیز شبیه سازی کند. تحقیقات انجام شده برای زبان های غربی علی‌الخصوص زبان انگلیسی تا حد زیادی به این مقصود دست یافته است.

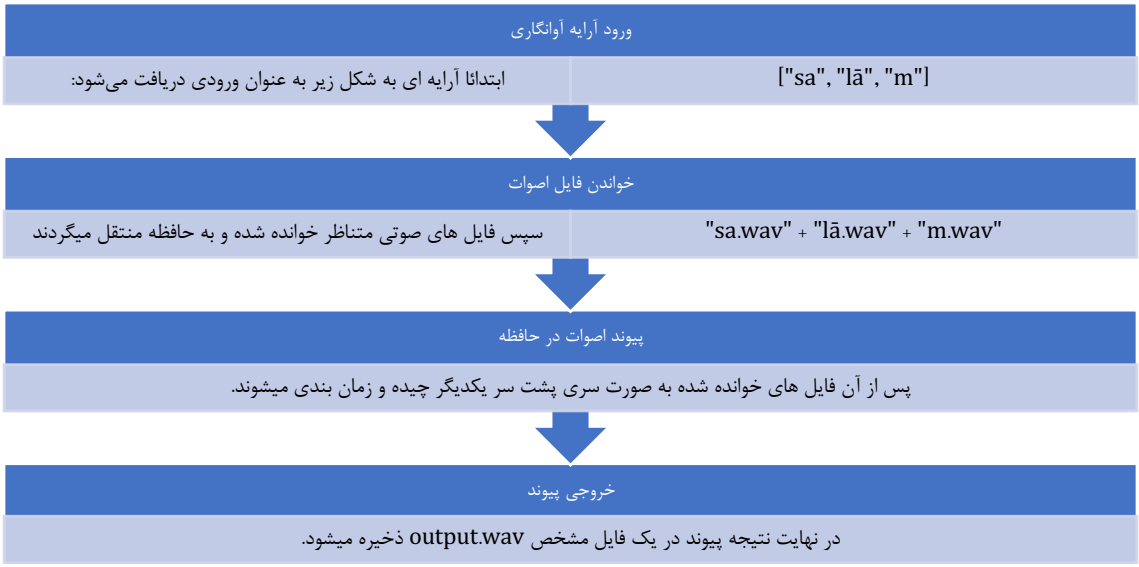
با این حال با توجه به پیچیدگی های زیاد این مبحث و همچنین نیاز به ضبط ماتریس گسترده ای از دوحرفی های صامت+مصوت در سیستم سنتز پیوندی ، در پروژه فعلی به این مبحث پرداخته نشده و تنها بیان جملات خبری به فرم ساده مورد پیاده سازی قرار گرفته است.

۱/۱/۶ تولید کننده صدا

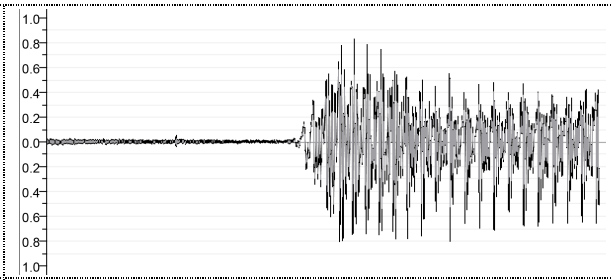
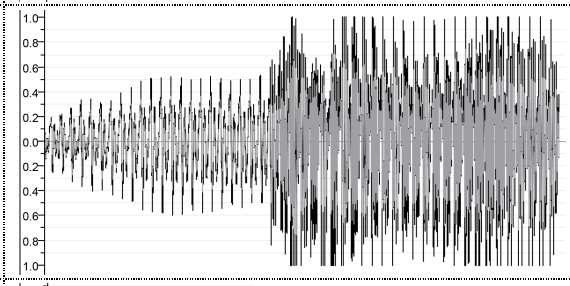
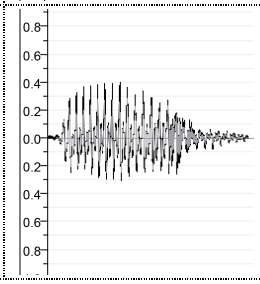
همانطور که پیشتر گفته شد، در سیستم سنتز پیوندی صدای نهایی از ترکیب (اتصال) صدای ضبط شده دو حرفی های صامت+مصوت به یکدیگر به وجود می آیند. در ادامه مثال شرح داده شده در قطعه کد ۱ بر مبنای تحقیقات صورت گرفته [7] مراحل تولید صوت کلمه «سلام» به شرح زیر اتفاق می افتد:

---

[۱۱] Stress
[۱۲] Prosody
[۱۳] Intonation



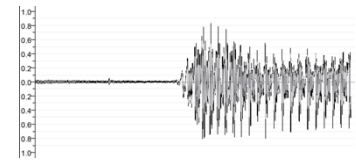
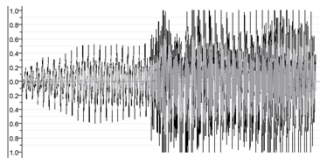
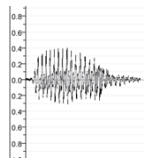
شکل ۲- مراحل تولید صدای سنتز پیوندی

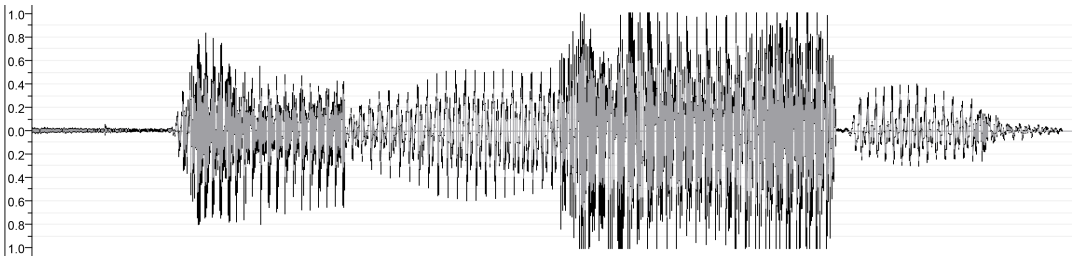
sa.wav	
lā.wav	
m.wav	

جدول ۳ - waveform هجاها

تحقیقات صورت گرفته [7] حاکی از آن است که برای ضبط هجاها بهترین روش ، ضبط تلفظ یک کلمه بطور کامل و نهایتا برش زدن هجای مورد نظر از کلمات می باشد. در پروژه فعلی تعداد ۱۶۷ کلمه برای برش زدن ۱۶۷ هجای فارسی دو مرتبه با سرعت های آهسته و سریع به ضبط رسیدند.

پس از پیوند صدای هجاها ، خروجی مثال بالا به شکل زیر خواهد بود:

تک هجاهای پیوند نخورده:		
		
sa.wav	lā.wav	m.wav

هجا های پیوند زده شده:

output.wav

جدول ۴ - نحوه پیوند هجاها

۱/۱/۷ پخش کننده صدا

در حالت کلی پخش صدا به صورت سخت افزاری و نرم‌افزاری امکان پذیر است که در پروژه فعلی با توجه به مقصد بودن بستر موبایل صداهای خروجی در اندروید و iOS از طریق framework های داخلی سیستم عامل ها به سمع کاربر نرم‌افزار می‌رسد.

بخش دوم : پیاده سازی نرم افزار و توضیحات فنی

در این بخش به تکنولوژی های استفاده شده در جهت پیاده سازی نرم افزار و دلایل انتخاب آنها می پردازیم و همچنین راهنمای خلاصه ای به منظور راه اندازی نرم‌افزار ، کامپایل و توسعه بیشتر آن را به رشته تحریر در می‌آوریم.

۱/۲ محیط پیاده سازی و تکنولوژی ها

برای پیاده سازی این نرم‌افزار از ویرایشگر های کد Visual Studio Code ، Vim و Xcode استفاده شده است. همچنین محیط پیاده سازی نرم‌افزار سیستم عامل MacOS X 10.14.5 بوده است و برای کامپایل گرفتن نسخه iOS مناسب برای دستگاه های iPhone و iPad داشتن این سیستم عامل و نصب بودن Xcode و Xcode command line tools ضروری می باشد.

برای کامپایل کردن نسخه Android تنها نصب بودن Android SDK و Android command line tools روی سیستم عامل های پشتیبانی شده این ابزار (از جمله ترجیحا یکی از توزیع های Linux یا سیستم عامل Microsoft windows) کافی خواهد بود.

همچنین برای تست سیستم سنتز پیوندی، اینترفیس دستوری FFMpeg را با انجام اقدامات مرحله به مرحله راهنمای سایت رسمی این کتابخانه میتوانید نصب کنید.

۱/۲/۱ زبان برنامه نویسی JavaScript و تکنولوژی های React و ReactNative

زبان برنامه نویسی JavaScript عموما به عنوان زبان اسکریپت نویسی سمت کلاینت در توسعه وب شناخته می شود اما با ورود موتور مفسر V8 که شرکت Google اولین بار در سال ۲۰۰۸ در پروژه Chromium آنرا توسعه داد موجب شد این زبان قدرت بالایی (قابل رقابت با زبان های کامپایلری) بیابد و نهایتا با توسعه پروژه Node.js این زبان به زبان برتر برنامه نویسی سمت سرور نیز تبدیل گشت.

شرکت Facebook در سال ۲۰۱۳ کتابخانه React را جهت تولید نرم افزار های کلاینت وب (Single Page Applications – SPA) شروع به توسعه نمود و پس از موفقیت بسیار زیاد آن از سال ۲۰۱۵ پروژه ReactNative را استارت زد که نهایتا به موجب آن تولید اپلیکیشن های موبایل با قابلیت های

بسیار زیاد کتابخانه React و سازگاری آن با هر دو سیستم عامل اصلی موبایل با بستر native و بازدهی یکسان انجام پذیر شد.

پروژه ReactNative به دلیل اینکه برای اندروید به زبان Java و برای iOS به زبان Objective-C پیاده سازی شده در هر دو پلتفرم بصورت کاملا native تهیه خروجی میکند و به هیچ وجه با تکنولوژی های هیبریدی مثل Apache Cordova و ... قابل مقایسه نیست.

۱/۲/۲ روش ضبط صدا و انتخاب فرمت wav در مقابل دیگر فرمت های موجود

با توجه به بازمتن بودن این پروژه و از آنجا که مبنای پروژه تا حد زیادی بر پایه تکنولوژی های بازمتن[۱۴] می باشد ، برای ضبط کلمات کامل و برش هجاها نیز از نرم افزار بازمتن Audacity و برای تولید Waveform spectogram ها از نرم‌افزار بازمتن Sonic Visualiser استفاده شده است.

با تحقیقات درباره ماهیت فرمت wav نسبت به فرمت های دیگر مثل mp3 یا flac و ... با توجه به غیر فشرده بودن این فرمت صوتی و نیاز بسیار پایین آن به منابع سخت افزاری جهت پردازش و ویرایش ، نهایتا این فرمت صوتی مبنای اصوات ضبط شده و خروجی پردازش های درون نرم‌افزار قرار گرفت.

۱/۲/۳ پیاده سازی سنتز پیوندی با FFMpeg و mobile-ffmpeg

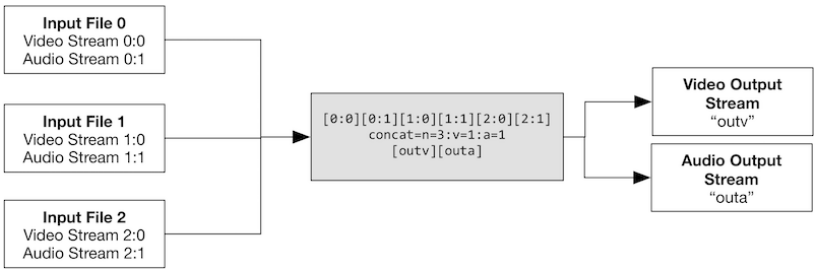
مهم ترین مرحله تولید صدای نهایی ، سنتز پیوندی آن است که برای این مساله از کتابخانه بازمتن FFMpeg استفاده می گردد. با توجه به اینکه این کتابخانه به زبان C و Assembly پیاده سازی شده است علاوه بر قابلیت حمل[۱۵] ایده آل از بهینگی بسیار بالایی نیز برخوردار است.

طبق مستندات آنلاین این کتابخانه به هر یک از پردازش های انجام شده توسط آن filter گفته میشود و filter مورد نیاز سنتز پیوندی به همین نام (concat) نام گذاری شده است [8]. این فیلتر تنها مختص فایل های صوتی نیست و قابلیت پیوند قطعات فیلم به یکدیگر را نیز دارا می باشد.

طبق مستندات آنلاین این ابزار نحوه عملکرد این فیلتر به نحو زیر می باشد:

---

[۱۴] Open Source
[۱۵] Portability



شکل۳ - نحوه عملکرد فلیتر پیوند

فرم کلی فراخوانی دستور در محیط bash بدون استفاده از کانال تصویر و تنها برای کانال صوت برای مثال ذکر شده در قطعه کد ۱ به صورت زیر می باشد:

```
ffmpeg \
-I sa.wav -I lā.wav -I m.wav \
-filter_complex '[0:0][1:0][2:0]concat=n=3:v=0:a=1[out]' \
-map '[out]' output.wav
```

قطعه کد ۲ – نحوه فراخوانی سنتز پیوندی در FFMpeg

در مشاهده می شود که برای ساخت کلمه «سلام» می بایست ۳ ورودی مطابق جدول ۴ - نحوه پیوند هجاها به حافظه منتقل شود. سپس با پارامتر filtex_complex نگاشت پشت سر هم ورودی ها را به فرمت [audioIndex:videoIndex] وارد کرده و آپشن های فیلتر concat به شرح زیر ست می شود:

۱. آپشن n تعداد ورودی ها را ست می کند که در این مثال ۳ ورودی داده شده
۲. آپشن v تعداد خروجی های کانال ویدیو را ست میکند که بطور بدیهی صفر ویدیو ست شده
۳. آپشن a تعداد خروجی های کانال صوت را ست میکند که بطور بدیهی باید ۱ ست شود.

نهایتا خروجی فیلتر در لیبل out نگاشت می شود و توسط لایبرری این نگاشت جهت ارسال stream به فایل خروجی به کار گرفته می شود.

همانطور که پیشتر نیز آورده شد لایبرری FFMpeg قابلیت حمل بالایی دارد و در حال حاضر برای پلتفرم موبایل نیز توسط توسعه دهندگان دیگر port شده است. کد منبع نسخه سبک آن که برای این پروژه استفاده شده mobile-ffmpeg [9] نام دارد و wrapper مخصوص ReactNative آن نیز تحت عنوان react-native-ffmpeg [10] توسط همین توسعه دهنده بصورت رسمی در اختیار برنامه نویسان قرار گرفته است.

با توجه به فرمت کلی فراخوانی فیلتر پیوند در FFMpeg الگوریتم تولید پویای این دستور با یک حلقه و با O(n) نسبت به نگاشت ورودی ها به رشته فراخوانی دستور اقدام میکند. کد این الگوریتم در فایل PhonemsToFFMpeg/index.js از ریشه utils جود دارد.

با توجه به اینکه در محیط موبایل منابع قبل از استفاده می بایست به sandbox مختص خود انتقال داده شوند این انتقال بصورت غیر هم زمان[۱۶] منتقل می گردند و پس از انتقال و به دست آمدن آدرس مطلق[۱۷] فایل های صوتی الگوریتم تولید رشته فراخوانی فیلتر پیوند کار خود را آغاز میکند و خروجی آن یک رشته مطابق قطعه کد ۲ – نحوه فراخوانی سنتز پیوندی در FFMpeg خواهد بود.

۱/۲/۴ مراحل طراحی و پیاده سازی

در مرحله ابتدایی پیاده سازی اپلیکیشن موبایل طراحی و آنالیز امکانات آن و همچنین دیزاین ساده رابط کاربری انجام پذیرفت. طراحی رابط کاربری و جریان کاربر۱۸ با استفاده از نرم افزار Sketch در سیستم عامل MacOS انجام گرفته است و شمایل کلی امکانات و نحوه اتصال بخش های مختلف آن در شکل ۴ - رابط کاربری و جریان کاربر آورده شده است.

امکانات در نظر گرفته شده برای نرم افزار به طور کلی به این روال می باشد:

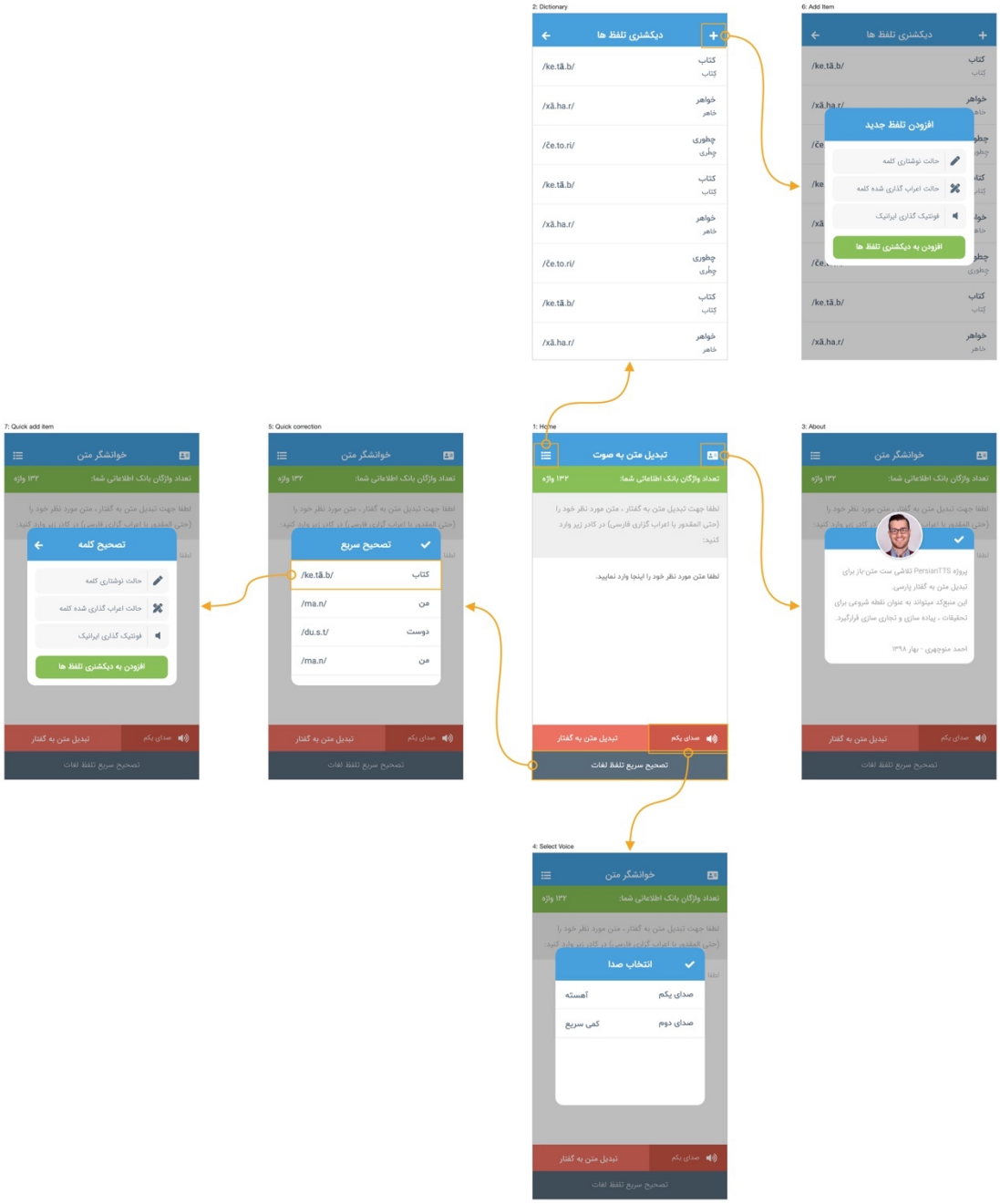
۱. صفحه اصلی
   a. ورودی متن به همراه قابلیت اعراب گذاری برای متنی که کاربر قصد شنیدن صدای آنرا دارد
   b. دکمه انتخاب صوت خواندن متن (دو صدا در حال حاضر با سرعت آهسته و بسیار سریع ضبط شده و آماده انتخاب است)
   c. دکمه نمایش صفحه تصحیح سریع کلمات وارد شده توسط کاربر
   d. دکمه نمایش صفحه فرهنگ لغات شخصی کاربر
   e. دکمه نمایش اطلاعات جانبی درباره نرم افزار
۲. صفحه فرهنگ لغات شخصی

---

[۱۶] Asynchronous
[۱۷] Absolute path
[۱۸] User flow

a. قابلیت مشاهده لیست لغات تعریف شده توسط شخص کاربر

b. قابلیت اضافه ، ویرایش یا حذف لغت و تلفظ صحیح آن توسط کاربر



شکل ۴ - رابط کاربری و جریان کاربر

از میان صفحات بالا در زمان نگارش این مستندات صفحات اصلی و صفحه انتخاب صدای گوینده به طور کامل پیاده سازی شده و اجرایی هستند. دیتابیس درنظر گرفته شده برای پیاده سازی دیکشنری تلفظ ها در این اپلیکیشن به جهت سبک تر بودن SQLite نسبت به Realm مورد اول می باشد.

۱/۲/۵ پیکر بندی پروژه

سورس کد پروژه از پیکر بندی استاندارد ReactNative استفاده می کند. جهت کنترل کیفیت کدنویسی[۱۹] و همچنین کنترل استفاده از بهترین عادات[۲۰] از کتابخانه ESLint استفاده شده است و قوانین کنترل کیفیت کدنویسی آن با استاندارد های شرکت AirBnB منطبق گشته است.

همچنین برای تست های واحد[۲۱] از ابزار Jest توسعه یافته توسط شرکت Facebook استفاده گشته و برای کلاس های پایه ای پروژه تست های واحد نوشته شده و پاس شده اند.

پیکر بندی استاندارد شده توسط این پروژه به شرح زیر می باشد:

- پوشه ریشه (src)
  - App.js کامپوننت پایه نرم افزار و در برگیرنده مسیریاب[۲۲] ریشه نرم افزار
  - Router.js شامل Hashmap نگاشت صفحات نرم افزار و تنظیمات مسیریابی به آنها
  - Voices.json شامل نگاشت کلید به مقدار صداهای موجود در نرم افزار
  - پوشه utils – الگوریتم های پایه ای نرم افزار
    - پوشه TextToPhonems
      - vowels.js شامل حروف صدادار (مصوت ها) در زبان فارسی
      - consonants.js شامل صامت های زبای فارسی
      - frequent500.json دیکشنری تلفظ ۵۰۰ لغت پرتکرار زبان فارسی
      - reshaper.js الگوریتم تبدیل نوشتار فارسی به معادل unicode آن
      - index.js الگوریتم اصلی تبدیل متن به آوانگاری ایرانیک
    - پوشه PhonemsToFFMpeg
      - index.js الگوریتم اصلی تبدیل آوانگاری به دستورات FFMpeg

---

[۱۹] Code quality
[۲۰] Best practices
[۲۱] Unit test
[۲۲] Router

- پوشه screens – شامل کامپوننت های صفحات مختلف نرم افزار
  - پوشه Home – کامپوننت صفحه اصلی نرم افزار
  - پوشه SelectSoundModal – کامپوننت صفحه انتخاب گر صدا
  - پوشه DictionaryList – کامپوننت لیست فرهنگ لغات کاربر
  - پوشه AddWordForm – کامپوننت فرم ویرایشگر تلفظ لغت
  - پوشه QuickWordsList – کامپوننت افزودن سریع تلفظ به فرهنگ لغت
- پوشه components – شامل کامپوننت های عمومی استفاده مجدد پذیر[23]
  - پوشه ColorButton – کامپوننت دکمه با قابلیت تنظیم رنگ ، آیکن ، عنوان ، اندازه و نحوه نمایش

- پوشه ios: شامل کدهای Objective-C جهت کامپایل برای iOS
- پوشه android: شامل کدهای Java جهت کامپایل برای Android
- پوشه assets: شامل فونت های باز متن استفاده شده در نرم افزار
- پوشته node_modules: حاوی کد منبع های کتابخانه های مختلف استفاده شده در پروژه
- پوشه __tests__: شامل تست های واحد الگوریتم های اصلی نرم افزار

۱/۲/۶ راه اندازی و کامپایل اولیه پروژه

پس از رفتن به مسیر پروژه در محیط دستوری[۲۴] می بایست مراحل زیر برای راه اندازی پروژه انجام شود:

۱. ابتدا کتابخانه های اصلی مورد نیاز توسط دستور npm install یا yarn install نصب شود
۲. سپس برای راه اندازی پروژه iOS در صورت نیاز با رفتن به مسیر پوشه ios دستور pod install جهت نصب نیازمندی های پروژه iOS اجرا شود
۳. نهایتا برای اجرای پروژه روی android دستور react-native run-android ویا برای راه اندازی پروژه iOS دستور react-native run-ios می بایست اجرا شود.

لازم به ذکر است ابزار های npm یا yarn ، CocoaPods و ReactNative می بایست قبلا نصب باشند.

---

[23] Reusable general React components
[۲۴] Command line

۱/۳ کد منبع[۲۵] و توسعه بیشتر

کد های منبع این پروژه به عنوان تلاش کوچکی در برای بستر تبدیل متن به صوت فارسی بصورت باز متن در یک مخزن github به آدرس http://github.com/amfolio/persian-tts قرار گرفته است و مجوز آن GPLv3 تعیین گشته است که به موجب آن نرم افزار های استفاده کننده از آن موظف به ذکر منبع و همچنین باز متن کردن کدهای خود هستند.

به دلیل علاقه مندی زیاد به این مبحث نگارنده قصد ادامه توسعه آن را دارد و کدهای آن همیشه بصورت بازمتن در اختیار توسعه دهندگان دیگر قرار خواهد گرفت.

بدون شک قابلیت های بیشمار زیادی میتواند به این پروژه اضافه شده و کیفیت آن را بهبود ببخشند. برای مثال صداهای بهتر توسط صدا پیشگان ضبط شده و به نرم‌افزار افزوده شود. یا الگوریتم تولید صدا از تکیه گذاری روی متن ، جملات پرسشی ، حالات شادی ، خشم ، پر انرژی و ... پشتیبانی کند و یا با بهبود الگوریتم فعلی و محاسبه دوطرفه از اول به آخر و از آخر به اول کلمات، ترکیبی از صامت+مصوت ها ارائه کند که وقفه کمتر و هماهنگی بیشتری بین اجزاء صوتی ایجاد گردد.

همچنین میتوان با استفاده و تنظیم دقیق فیلتر هایی مثل acrossfade در FFMpeg بخش های مختلف صدا را در یکدیگر به نحوی ادغام کرد که وقفه بین آنها کمتر شده و همخوانی بیشتری بیابد.

و یا با ساخت یک اپلیکیشن سمت سرور و همگام سازی داده های دیکشنری های فردی کاربران با پایگاه داده ابری ، با اشتراک گیری از پر اصلاح ترین لغات ، آنها را با دیکشنری های داخلی کاربران دیگر همگام سازی کرد و با یک هم افزاریی همگانی به تصحیح خودکار نرم افزار کمک نمود.

علاوه بر آن تحقیقات انجام شده نمایانگر آن است که علوم هوش مصنوعی مثل NLP و یادگیری ماشین در راستای موضوع این مقاله قدم های بسیار زیادی برداشته اند که با مطالعه و به کارگیری آنها بهبود این نرم افزار دستخوش کیفیت بسیار بالاتری خواهد شد.

امید است باز متن بودن این نرم افزار منجر به کمک دیگر توسعه دهندگان نیز بشود و در آینده شاهد کتابخانه ای جامع و قدرتمند برای تبدیل متن به گفتار فارسی باشیم.

احمد منوچهری
تابستان ۱۳۹۸

---

[۲۵] Source code

منابع

[1] Michael H. O'Malley, "Text To Speech Conversion Technology," *IEEE,* p. 21, 1990.

[2] Mrs. S. D. Suryawanshi, Mrs. R. R. Itkarkar and Mr. D. T. Mane, "High Quality Text to Speech Synthesizer using Phonetic Integration," *International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE),* p. 133, 2014.

[۳] H. Dave, "FrequencyWords Github repository," February 2018 [درون خطی]. Available: https://github.com/hermitdave/FrequencyWords.

[۴] OpenSubtitles, "New collection of translated movie subtitles," 7 January 2016 [درون خطی]. Available: http://opus.nlpl.eu/OpenSubtitles2016.php.

[۵] Wikipedia, "International Phonetic Alphabet" [درون خطی]. Available: https://en.wikipedia.org/wiki/International_Phonetic_Alphabet.

[۶] س. ع. هاشمی, "الفبای آوانگاری ایرانیک (IRPA)," 06 April 2016 [درون خطی]. Available: http://amerpage.org/web/works/phonetics/IRPA.htm.

[۷] Mr. D. T. Mane, "High Quality Text و Mrs. S. D. Suryawanshi, Mrs. R. R. Itkarkar to Speech Synthesizer using Phonetic Integration", *International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)*, جلد ۳, شماره ۲, 2014, p. 134.

[۸] FFMpeg, "Concatnation Documentation," 2019 [درون خطی]. Available: http://ffmpeg.org/ffmpeg-filters.html#concat.

[۹] T. Sener, “FFmpeg for Android and IOS,” 2019 [درون خطی]. Available: https://github.com/tanersener/mobile-ffmpeg.

[۱۰] T. Sener, “FFmpeg for react-native,” 2019 [درون خطی]. Available: https://github.com/tanersener/react-native-ffmpeg.