TTS Reengineered.

A Novel Approach

MEHUL SINGH

2023

Introduction

In the past few years, it has emerged as a significant newly growing field in voice synthesis owing to the rising demand for speech sounding as natural as possible. Although conventional methods have proven effective, the Vointhez project relies on using a new algorithm, making it much more efficient, simpler, and more flexible.

The heart of Vointhez lies in a high performance and flexible algorithm that quite precisely and efficiently models the complexities underlying human speech production. Capturing the nuances of pronunciation and rhythm. This can be an innovative approach with radical implications for ventures such as customer service and virtual assistants, for language learning, and also for access technologies.

One important characteristic of Vointhez is its incredible flexibility. The system can be easily customized to allow different applications and languages, making it adaptable to further changing needs. In addition, the seamless integration with many platforms and devices adds to its functionality and accessibility.

Vointhez-I represents a step very much into the future in voice synthesis. The features of innovation, efficiency, simplicity, and adaptability make this one promising solution to a variety of applications. The versatility of the upcoming demands for natural-sounding speech generation positions Vointhez as key in shaping the future of voice technology.

Mehul Singh

ALGORITHM

1. Algorithm Explained.

1.01 Alphabetical Pattern Breakdown.

This essential aspect can be considered the back bone of this first type of approach. The phonetic realization of a word can be segmented into a specific sequence of phonemes, along predictable lines, such as:

Case 1: Words Having Even Number of Characters

```
Word: abacus    Pronunciation: ab-ac-us
```

Case 2: Words Having Odd Number of Characters

```
Word: steel    Pronunciation: st-ee-l
```

These examples bring out the different combinations of characters needed for the accurate pronunciations of different words.

The character sets are as follows:

1.01.01. Character Sets starting with a

```
aa ab ac ad ae af ag ah ai aj ak al am an ao ap aq ar as at au av aw ax ay az
```

1.01.02. Character Sets starting with b

```
ba bb bc bd be bf bg bh bi bj bk bl bm bn bo bp bq br bs bt bu bv bw bx by bz
```

1.01.03. Character Sets starting with c

```
ca cb cc cd ce cf cg ch ci cj ck cl cm cn co cp cq cr cs ct cu cv cw cx cy cz
```

1.01.04. Character Sets starting with d

```
da db dc dd de df dg dh di dj dk dl dm dn do dp dq dr ds dt du dv dw dx dy dz
```

1.01.05. Character Sets starting with e

```
ea eb ec ed ee ef eg eh ei ej ek el em en eo ep eq er es et eu ev ew ex ey ez
```

1.01.06. Character Sets starting with f

```
fa fb fc fd fe ff fg fh fi fj fk fl fm fn fo fp fq fr fs ft fu fv fw fx fy fz
```

1.01.07. Character Sets starting with g

```
ga gb gc gd ge gf gg gh gi gj gk gl gm gn go gp gq gr gs gt gu gv gw gx gy gz
```

1.01.08. Character Sets starting with h

```
ha hb hc hd he hf hg hh hi hj hk el em en eo ep eq er es et eu ev ew ex ey ez
```

1.01.09. Character Sets starting with i

```
ia ib ic id ie if ig ih ii ij ik il im in io ip iq ir is it iu iv iw ix iy iz
```

1.01.10. Character Sets starting with j

```
ea eb ec ed ee ef eg eh ei ej ek el em en eo ep eq er es et eu ev ew ex ey ez
```

1.01.11. Character Sets starting with k

```
ka kb kc kd ke kf kg kh ki kj kk kl km kn ko kp kq kr ks kt ku kv kw kx ky kz
```

1.01.12. Character Sets starting with l

```
la lb lc ld le lf lg lh li lj lk ll lm ln lo lp lq lr ls lt lu lv lw lx ly lz
```

1.01.13. Character Sets starting with m

```
ma mb mc md me mf mg mh mi mj mk ml mm mn mo mp mq mr ms mt mu mv mw mx my mz
```

1.01.14. Character Sets starting with n

```
na nb nc nd ne nf ng nh ni nj nk nl nm nn no np nq nr ns nt nu nv nw nx ny nz
```

1.01.15. Character Sets starting with o

```
oa ob oc od oe of og oh oi oj ok ol om on oo op oq or os ot ou ov ow ox oy oz
```

1.01.16. Character Sets starting with p

```
pa pb pc pd pe pf pg ph pi pj pk pl pm pn po pp pq pr ps pt pu pv pw px py pz
```

1.01.17. Character Sets starting with q

```
qa qb qc qd qe qf qg qh qi qj qk ql qm qn qo qp qq qr qs qt qu qv qw qx qy qz
```

1.01.18. Character Sets starting with r

```
ra rb rc rd re rf rg rh ri rj rk rl rm rn ro rp rq rr rs rt ru rv rw rx ry rz
```

1.01.19. Character Sets starting with s

```
sa sb sc sd se sf sg sh si sj sk sl sm sn so sp sq sr ss st su sv sw sx sy sz
```

1.01.20. Character Sets starting with t

```
ta tb tc td te tf tg th ti tj tk tl tm tn to tp tq tr ts tt tu tv tw tx ty tz
```

1.01.21. Character Sets starting with u

```
ua ub uc ud ue uf ug uh ui uj uk ul um un uo up uq ur us ut uu uv uw ux uy uz
```

1.01.22. Character Sets starting with v

```
va vb vc vd ve vf vg vh vi vj vk vl vm vn vo vp vq vr vs vt vu vv vw vx vy vz
```

1.01.23. Character Sets starting with w

```
wa wb wc wd we wf wg wh wi wj wk wl wm wn wo wp wq wr ws wt wu wv ww wx wy wz
```

1.01.24. Character Sets starting with x

```
xa xb xc xd xe xf xg xh xi xj xk xl xm xn xo xp xq xr xs xt xu xv xw xx xy xz
```

1.01.25. Character Sets starting with y

```
ya yb yc yd ye yf yg yh yi yj yk yl ym yn yo yp yq yr ys yt yu yv yw yx yy yz
```

1.01.26. Character Sets starting with z

```
za zb zc zd ze zf zg zh zi zj zk zl zm zn zo zp zq zr zs zt zu zv zw zx zy zz
```

A separate library for special characters is also required, as detailed below:

```
! ~ # $ % ^ & * ( ) { } [ ] \ , . < > ? / "" '' ; : _ - = +
```

All the Alphabets also need to have multiple types of libraries separately:

1. Standalone
2. As the last character of a word

There also needs to be a separate library for numbers:

```
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34
35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65
66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96
97 98 99
```

Apart from these numbers, hundred, thousand, lakh, crore are to be recorded separately.

Words separated by a space & dashes will be treated as two different words.[2]

There needs to be a separate library for words with exceptional pronunciations which needs to be updated from time to time.

Example: Psychology

[1] There are certain words with exceptional pronunciations that cannot be managed by this algorithm and must be addressed separately.

[2] Can be modified if any better condition comes on our table.

IMPLEMENTATION

1.02 Exceptional Pronunciations.

1.03 High Level Language.

In practice, it is developed using Python-based programming language owing to portability, rich libraries, and strong community support for program development.

1.04 Vointhez.

Vointhez embodies a perfect albeit flawless implementation of the algorithm, displaying not just flawlessness and efficiency in its original design but also being seamlessly incorporated into separate systems. The implementation strictly follows the parameters for best practices; due care is given in robustness, maintainability and scalability. This also adopts efficiently all the features of the used programming language while proving knowledge of both the algorithmic principles employed and the technical environment in which it stands.

1.05 Compatibility.

Vointhez API would facilitate re-imagined ways of utilizing Vointhez TTS Engine. It would, therefore, tend towards improved efficiency in operations from various industries with relatively robust compatibility and high-speed processing.

1.06 Voice Modulation.

The Vointhez TTS Engine incorporates voice modulation together with synthesised speech that might be slightly different in all its renderings.