Bilaga 1: Huvudrapport

Wikispeech –

en användargenererad talsyntes på Wikipedia

Innehållsförteckning

Introduktion

Wikipedia är en av världens mest använda webbplatser med ca 500 miljoner besökare varje månad och ca 20 miljarder sidvisningar.[1] Wikipedia är en s.k. wiki och använder mjukvaran MediaWiki i bakgrunden. MediaWiki används av många tusen andra webbplatser och detta projekt syftar till att skapa den programvara som behövs för att talsyntes skall kunna användas på alla dessa och optimerad på Wikipedia.

Med hjälp av navigering och uppläsning med syntetiskt tal, kan personer som bättre tillgodogör sig tal än skrift få likvärdig tillgång till informationen. I förlängningen gör projektets öppna natur det möjligt att utveckla nya sätt att presentera den talade informationen, exempelvis i en spelare avsedd för mobiltelefoner. Det kan handla om de som har synnedsättningar, som har dyslexi eller är analfabeter. Även de ca. 25% som lär sig bäst av uppläst text ska kunna nyttja funktionen samt de som vill lära sig samtidigt som de gör något annat (ex. kör bil). 25% av Wikipedias läsare innebär att för närvarande skulle runt 115-125 miljoner människor kunna få nytta av projektets arbete på sikt.

De som har fått en medicinsk diagnos gällande begränsningar i läsförståelse (exempelvis dyslexi, synnedsättningar eller kognitiva nedsättningar) har ofta tillgång till hjälpmedel. Det krävs dock ofta en diagnos, att du bor i ett rikt land samt att det språk du talar har en fungerande talsyntes för att det ska vara en lösning på tillgänglighetsproblemet. Även personer med svag läsförståelse (från ovana läsare till analfabeter) har en begränsad tillgång till kommersiella verktyg. Trots att det skulle öka deras förståelse. Inte minst om de inte vill dela all sin data med någon av IT-jättarna. Sammanfattningsvis blir bedömningen att en mycket stor grupp skulle gynnas av en inbyggd talsyntes på Wikipedia.

Att göra alla de webbplatser som använder MediaWiki mer tillgängliga för de som av olika orsaker har svårt att ta till sig skriven text är därför oerhört viktigt. Projektet kommer att bredda tillgängligheten för en av de viktigaste webbplatserna. Alla andra plattformar som använder MediaWiki kommer att kunna dra nytta av de tekniska lösningar som utvecklas under projektet. Det rör sig om tusentals webbplatser som snabbt och enkelt kommer att kunna aktivera talsyntes.

Wikipedia och många andra wikis innehåller många specialiserade texter vilket gör att talsyntesens uttalslexikon måste vara mycket omfattande för att fungera tillfredsställande. Därtill finns Wikipedia på 288 olika språk, och plattformen skall vara skalbar till alla de språken samt alla framtida. Projektet gör det lättare att utveckla talsyntes för språk som ännu saknar tekniken, vilket är intressant då det kan finnas åtskilliga talare i Sverige, men inte tillräckligt många för att göra det kommersiellt attraktivt/prioriterat. Kommersiella lösningar finns endast för en bråkdel av alla språk.

Flexibilitet är därför centralt. Svenska, engelska samt ett höger-till-vänster-språk (arabiska) kommer att inkluderas i genomförandeprojektet. Ett sätt att nå en flexibel plattform är att tillgodogöra sig den språkliga expertis som finns hos de tiotusentals volontärer som är

[1] http://reportcard.wmflabs.org

involverade i Wikimedias olika projekt. Genom att användargenerera talsyntesen med inspelningar av de specialiserade texterna kan vi nå en förfinad och högkvalitativ talsyntes även i obskyra ämnen på språk som tidigare helt saknat en fungerande talsyntes. Metoderna runt användargenererad talsyntes kommer att vara möjliga att använda även för andra typer av texter då även dessa fritt kommer att delas. Till skillnad från stängda lösningar gör den användargenererade open source-lösningen det även möjligt för de som använder talsyntesen att själva förbättra den och slippa störande fel (på samma sätt som många läsare uppskattar att kunna rätta exempelvis stavfel i texterna).

Det är ett vanligt misstag att utgå från att det finns en lösning som passar alla som exempelvis är blinda. Även om de delar en funktionsnedsättning kan personerna i övrigt ha helt olika förutsättningar och olika behov. Exempelvis kan en person som varit blind hela livet ofta spela upp texten i en väldigt hög takt, men om personen förutom att vara blind även har en kognitiv nedsättning så vill de istället kunna spela upp texten långsammare än vanligt. Möjligheten att på Wikipedia skapa ett personligt konto där egna inställningar kan sparas är därför centralt.

Projektet kommer att gynna både forskare och företag då materialet som genereras av volontärer för att förbättra talsyntes kommer att vara fritt att återanvända – även för kommersiellt bruk. En liknande lösning saknas idag på marknaden.

Inom forskningen ger projektet unika möjligheter. Nyttan för forskning runt talteknologi och talsyntes är uppenbar: tillgång till stora datamängder är en av huvudförutsättningarna för modern talteknologi, och här genereras detta på alla plan, med inläsningar, användardata, och användargenererade utvärderingar. Projektet inte bara stöder utan kan gå i bräschen för nya metoder inom användarcentrerad iterativ forskning och utveckling.

Speciellt intressant ur ett talteknologiskt forskningsperpspektiv är att projektet arbetar med uppläsning av längre, sammanhängande text om brett varierande ämnen. Befintliga talsynteser är normalt inte utvecklade för den typen av uppläsning, trots att den är av stor betydelse ur tillgänglighetsperspektiv. Projektet bidrar till forskningen även utanför talteknologin. Exempelvis kan den återkoppling från användare som genereras betraktas som en form av perceptionstester som kan ge insikter om hur verkliga lyssnare uppfattar tal av olika slag, och ur användargenerarade kontinuerliga uppdateringar och tillägg av ord och uttal kan vi lära oss saker som tidigare inte var tillgängliga om hur språk utvecklas och hur olika språk förhåller sig till varandra.

Användarscenarion

Här nedan ges tre olika användarscenarion för att visa hur vi tänker oss att en användare nyttjar de tre olika processer vi har identifierat: att lyssna på talsyntesen, att förbättra talsyntesen samt att utveckla talsyntesen med nya språk.

Kim lär sig av texten

Kim, som har svårt att ta till sig skriven text, är intresserad av att lära sig mer om demokrati och besöker Wikipedia för att lära sig mer om parlamentarism. Kim har alltid haft problem med att läsa, men av olika anledningar blev det hela aldrig utrett och Kim har inte fått några verktyg som stöd. Idag är skolorna i Sverige rätt duktiga på att utreda barnens behov och ge dem hjälpmedel, men det är en rätt ny företeelse. På webbplatsen finns det dock en knapp för att få artikeln uppläst via talsyntes som alla kan använda.

Kim har lämnat laptopen hemma och bestämmer sig idag för att använda sin mobil för att läsa på. Det är ju trots allt lika enkelt att navigera i det praktiska mobilgränssnittet på alla de 70 olika språk som artikeln (för närvarande) finns på. Kim går idag till den svenskspråkiga artikeln då den är väl utbyggd, men hade i annat fall kanske tittat på den engelskspråkiga artikeln (där det också finns en talsyntes). I artikeln klickar Kim på knappen och texten börjar läsas upp.

Flera existerande uttalslexikon har inkluderats, men artikel innehåller dock många fackuttryck vilket kräver ett specialiserat uttalslexikon för att talsyntesen skall bli korrekt (ex. ord som "misstroendevotum" och "folksuveränitetsprincipen"). I den svenska texten nämns även Gustav III vilket kräver att det finns en förståelse hur detta skall uttalas ("Gustav den tredje"). För en vecka sedan hade detta ställt till problem, men som tur är har engagerade volontärer just hjälpt till att utveckla lexikonet och talsyntesen inom detta ämnesområde är nu mycket välutvecklad på svenska.

Efter att ha gått igenom en del av artikeln tar Kim en paus och stannar talsyntesen men återkommer dagen efter och fortsätter uppläsningen på rätt plats. Behagligt nog försvinner inte alla de tillgänglighetsinställningar som Kim gjort gällande val av röst, uppspelningshastighet m.m. då dessa sparats ned som personliga inställningar då Kim valt att vara inloggad på Wikipedia. Då Kim blir störd vid ett par tillfällen behöver hen gå tillbaka och läsa om, vilket smidigt sker genom att antingen använda tangentbord eller mus.

Kim rättar uttalet i texten

När Kim går in för att titta på artikeln om M/S Teaterskeppet upptäcker Kim att uttalet för skeppets tidigare namn, Vágbingur, inte uttalas på ett bra sätt. Kim väljer då att fixa detta genom att i det här fallet hänvisa till det färöiska uttalslexikonet direkt i artikeln vilket Kim smidigt kan göra genom ett inbyggt verktyg.

Kim lyssnar vidare och upptäcker att den nautiska termen “föröver” uttalas felaktigt (som “för över”) och som gammal seglare känner hen att det inte går för sig utan går in för att rätta i uttalslexikonet. Kim har fått en speciell användarrättighet och kan som inloggad på Wikimedias projekt uppdatera uttalslexikonet och rättningen kommer, på wiki-vis, att gå live direkt.

Att uppdatera den fonetiska texten går snabbt tack vare den praktiska verktygslådan som gör det lätt att välja IPA-tecken och lyssna på om det låter rätt. När allt ser bra ut passar Kim

även på att spela in uttalet på en ljudfil som laddas upp och berikar både Wiktionary och Wikidata. Kim tycker att det är kul att arbetet gynnar flera projekt! Speciellt då inspelningarna i framtiden kan användas för att utveckla en ännu bättre talsyntes.

Kim passar även på att gå in och kontrollera några rättningar som andra användare gjort för att säkerställa att de håller en hög kvalité.

Kim lägger till sitt favoritspråk

Kim har under många år fördjupat sig i Esperanto och upptäcker med förskräckelse att det saknas en talsyntes för språket. Efter att ha tittat runt en del inser Kim att de viktigaste komponenterna faktiskt finns tillgängliga som fri programvara och som borde gå att sätta samman för att få en fungerande grund för talsyntesen.

Även om det existerande lexikonet som Kim hittat ännu är rätt svagt så tänker Kim att när allt väl är på plats så vore det ju värt att lägga någon timme per dag och utöka lexikonet. Efter att ha mailat med sin Esperanto-förening hittar Kim ytterligare tre personer som vill hjälpa till att utveckla lexikonet. Tillsammans börjar de gå igenom den välstrukturerade processen som finns för att aktivera en ny talsyntes på en språkversion av Wikipedia genom att göra utvecklarna medvetna om intresset och om de existerande komponenterna genom att lägga till all informationen på en wikisida. Så fort de känner sig redo kan de börja bygga upp lexikonet. (En del ord på Esperanto finns dock redan i lexikonet då Esperanto-ord som finns nämnda på andra språkversioner redan har lästs in). De håller en god takt och den lilla gruppen laddar upp över 200 nya ord till lexikonet per dag.

Några månader senare har de olika komponenterna för Wikispeech på Wikipedia på Esperanto anpassats av Wikimedias utvecklare för att passa in i den befintliga infrastrukturen. Det tog lite längre tid än förväntat då några av komponenternas licenser inte var tydligt angivna och de som skapat dem behövde kontaktas och några existerande komponenter behövde anpassas då de var byggda med föråldrad teknik. Glädjande nog beslutade de att släppa sitt material med en fri licens.

Bakgrund och definitioner

Minimum Viable Product

I detta projekt använder vi två olika definitioner: *Walking skeleton* och *Minimum Viable Product (MVP)*. Ett Walking skeleton är den minsta funktionalitet som behöver finnas för att det ska gå att se vad systemet egentligen gör. Inga avancerade funktioner är alltså med utan bara det allra mest grundläggande. Minimum Viable Product innehåller lite mer, nämligen den minsta mängd funktioner som krävs för att Wikimediagemenskapen och svenska funktionshinderorganisationer ska tycka att det är värt att aktivera talsyntesen. Dessa krav är alltså lite högre då det inte räcker med att det fungerar, det ska ha en viss mån av användarvänlighet också.

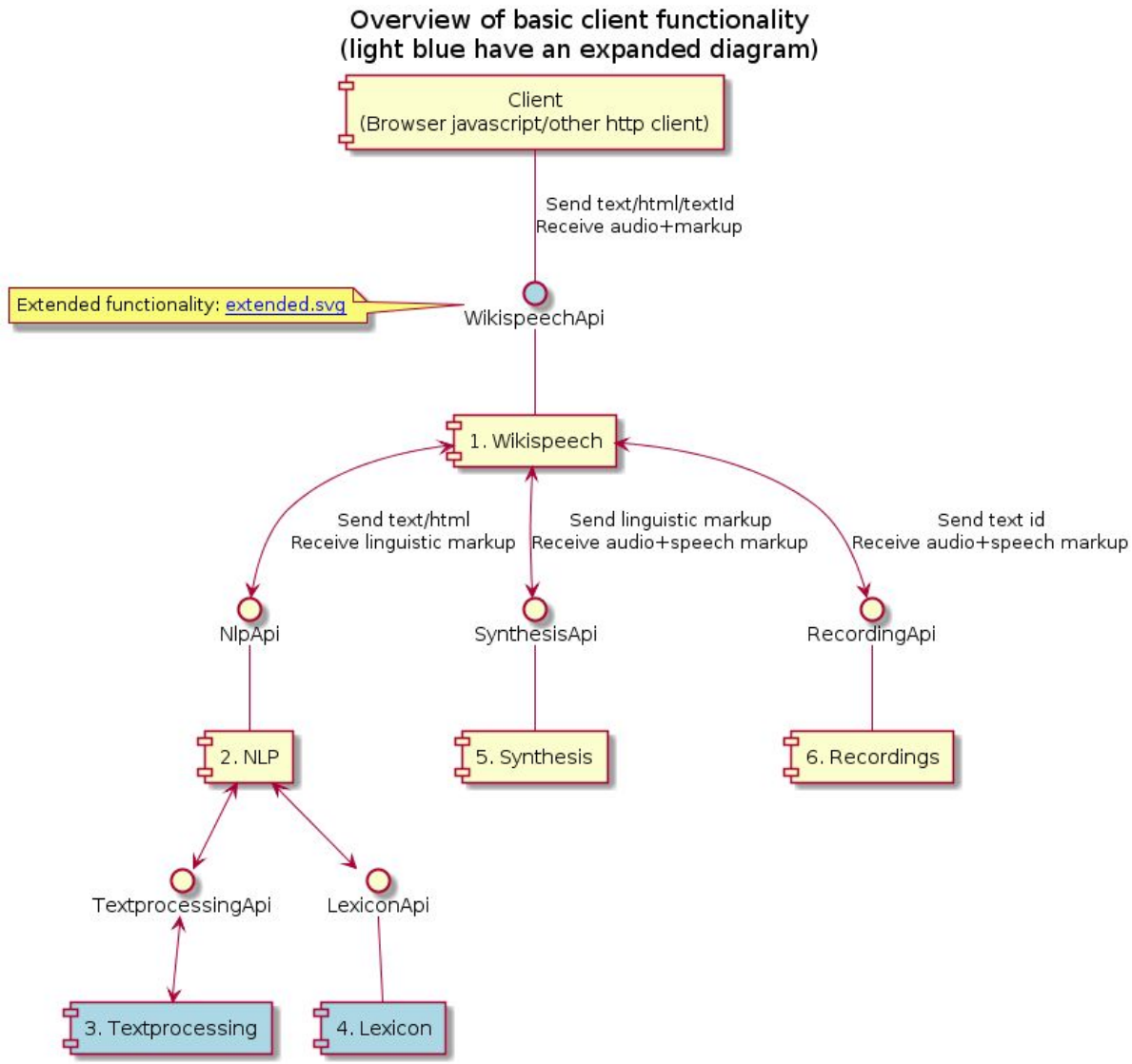
Dessa två begrepp används för att klassificera systemets olika egenskaper nedan. Vårt mål är att uppfylla alla egenskaper för MVP:n under projektet.

Övergripande systembeskrivning

Systemet kommer att vara byggt för att kunna hantera olika befintliga språkresurser, eftersom att tillgången till fria sådana och sätten de är gjorda på varierar så stort idag. Systemet ska också vara generellt nog för att kunna användas av alla Wikimedias projekt[2] och även tredjepartsinstallationer av MediaWiki. Därför kommer systemet bestå av ett flertal väldefinierade API:er som kan utbyta information med varandra, vi kallar alla dem tillsammans i texten för en “wrapper”. I bilden nedan är det det som är alla cirklar och pilar, medan de olika rutorna är delar som ska kunna vara utbytbara.

Wikispeech kommer att ligga som en serverlösning vilket innebär att den inte behöver laddas hem av användaren.

[2] Detta exkluderar sidor som inte är i standard wikiformat, t. ex. specialsidor, wikibase-specifika sidor m.m.



De olika rutorna 2, 3, 4, 5 och 6 kan innehålla olika delar för olika språk men har alla det gemensamt att de levererar in- och utdata på ett väldefinierat sätt.

Nytt språk

Förutom att det på några olika sätt skall gå att lyssna på Wikipedia med talsyntes, skall det vara möjligt att lägga till nya röster, för ett befintligt eller helt nytt språk. Det kommer inte att finnas ett komplett gränssnitt för att skapa en ny syntes(röst), med det kommer att finnas instruktioner som kan följas för att göra det.

MediaWiki och Wikimedias servermiljö

Mjukvaran MediaWiki är skrivet i PHP, men tillägg kan använda andra språk. För att något ska aktiveras i produktionsmiljön på Wikimedias servrar krävs det att alla delar är fritt licensierade samt att de kan användas på flera språk (MediaWiki finns på 371 språk och Wikipedia på 291). I Wikimedias miljöer finns runt 800 installationer av MediaWiki, vilka alla

potentiellt ska ha stöd för Wikispeech. Hela Wikispeech kommer att installeras på Wikimedias servermiljö för att stödja de 800 MediaWiki-installationerna.

Wikispeech kommer att skapas som ett *Tillägg* (en. *Extension*). Denna ska kunna aktiveras och konfigureras separat på varje wiki. För att detta ska tillåtas måste tillägget vara översättningsbart via plattformen Translatewiki.net. All kod måste skrivas anpassat efter det och ett visst jobb för installation där behöver göras.

För uppmärkningen av ljud, och för att göra det lätt redigerbart för alla som redigerar på Wikipedia har olika möjligheter att lagra detta undersökts. Den lösning vi tror mest på kommer inte att använda sig av uppmärkning i det traditionella wikitext-läget utan istället använda sig av en liknande teknik som den nya editorn till MediaWiki (VisualEditor) använder sig av (Parsoid). Denna lösning gör att koden för uppmärkningen inte är i vägen för alla som vill redigera en artikel, och därigenom inte irriterar volontärerna, men ändå är lätt att redigera för den som vill förbättra talsyntesen.

GUI:t i sig kommer att definieras allt eftersom. Det kommer att innehålla ett antal olika separata delar där den mest påtagliga för slutanvändaren är ljudspelaren. Den näst viktigaste biten blir gränssnittet för att göra förbättringar. Wikimedia Foundation har redan ett utvecklat designbibliotek med alla komponenter väl beskrivna (hur de ska se ut och hur de ska användas), så det kommer inte att behöva designas på den nivån, utan det blir mer layoutval och UX som blir viktigt.

Att vi har valt en serverlösning beror på att vi vill uppnå största möjliga tillgänglighet. I stället för att vi förlitar oss på att läsarna har egna program installerade säkerställer vi att alla kan tillgodogöra sig resultatet. Den alternativa lösningen är ju uppenbarligen att skapa en klient som kan installeras av läsaren, men det är en extra tröskel samt innebär också hindret att göra läsarna medveten om att den finns tillgänglig. En serverlösning gör också att talsyntesen kan användas av tredje part.

Valet att göra den så modulär som vi har valt beror på att det finns en stor mängd resurser tillgängliga ute i världen, men de är i sig inte standardiserade. Alternativet skulle vara att specificera en befintlig standard, men det skulle innebära att majoriteten av tidigare arbete inte kommer att passa in och göra att Wikispeech utvecklas i en mycket långsammare takt.

Processer

I projektet kommer tre processer att sättas upp: för att läsa upp texten med talsyntes, för att förbättra talsyntesen samt för att lägga till talsyntes för ett nytt språk. Här ges först en schematisk skiss och därefter presenteras alla de egenskaper som de olika processerna består av.

Process för uppläsning inkluderar:
Navigering → NLP→ Syntesmotor → Ljudspelare

Process för förbättring av talsyntesen inkluderar:

Rättning sker i uttalslexikon → Kontroll av lexikonförändringens kvalité → Insamling av taldata
↗
Felaktigt uppläst text identifieras
↘
Rättning sker i artikel → Gemenskapsdriven uppmärkning i texten

Process för att lägga till ett språk inkluderar:
Uttryckt intresse för aktivering av nytt språk → Existerande komponenter identifieras → Eventuella API-anpassningar utvecklas → Saknade eller dåliga komponenter utvecklas → Installation → Lokal konfigurering

Egenskaper

Nyckel:

Grön = Walking skeleton
Gul = MVP
Vit = Möjlig vidareutveckling

Process för uppläsning

Denna process är alltså generell för alla språk, och behöver alltså beskrivas på ett sådant sätt så att de språkspecifika delarna istället hamnar i processen för tillägg av språk. (Detta inkluderar även de tre första språken.)


Aktivering
Start- och stopp-funktion för taluppläsning för hela texten från början av artikeln med musklick på spelarknapp
Start- och stopp-funktion för taluppläsning för hela texten från början av artikeln med snabbtangenter
Möjlighet att direkt hoppa från artikeltext till omgivande knappar
Uppläsning av text som markerats
Fungerar i flera webbläsare
Går att aktivera från både mobil- och desktopgränssnitt
Hastigheten och andra inställningar på uppläsningen kan justeras enligt personliga preferenser
Uppläsningen kan gå tillbaka för att lyssna om
Uppläsningen kan hoppa framåt i olika intervaller (ex. ord, mening eller stycke)
På något sätt visa att det är en länk, fotnot eller liknande (ex. med en "pling") som går att ställa in
I redigeringsläget kan artikelförfattare lyssna på och korrigera uttalet för en viss text på direkten
Går att aktivera från Wikipedia-appen
Navigerings av taluppläsning för hela texten från början av artikeln med röststyrning
NLP
Text som ska läsas upp skickas till Wikimedias server (Wikispeech API)
API för textprocessning (ex. https och json/ssml)
Konvertering av text från rå till uppmärkt text (med tillräcklig hög grad av information (minumum en räcka med fonemsekvenser) (textnormalisering/parsning)
API för uttalskomponent
Struktur för uttalslexikon
Automatisk uttalskomponent för ord som inte finns i lexikon
TTS klarar att hantera olika taggar (ex. att det är en bildtext eller tabell som läses upp)
Möjlighet att välja bland befintliga uttal för språket (på flera olika sätt)
Generering av prosodiska taggar (emfas, frasering - antingen från data eller via regler)
Syntes-motor
Syntesmotorhantering (den generella)
Syntesens API för alla motorerna
Tolkning av SSML-uppmärkning (av pausering, satsbetoning, intonation, volym och fart)
Möjliggöra röstval (exempelvis manlig/kvinnlig röst eller brittisk eller amerikansk)
Möjliggöra prosodisk styrning (emfas)
Recording API (för inspelade ljudfiler)
Ljud-spelare
Se vilket ord som uttalas
Möjlighet att välja bland befintliga röster för språket (exempelvis manlig, kvinnlig)
Fungerar i flera webbläsare
Ljudfil går att få ut i olika format för nedladdning
Använda sina egna röster

Steg 1: Navigering
Användaren måste kunna lyssna på syntesen i sin webbläsare. Uppläsningen måste kunna pausas och återupptas på olika sätt. Möjlighet till navigering i texten/sidan är önskvärd. Användaren ska också kunna följa med i texten, där aktuellt ord markeras under uppläsning.

Egenskaper:

1. Start- och stopp-funktion för talupplästning för hela texten från början av artikeln med *musklick* på spelarknapp
2. Start- och stopp-funktion för talupplästning för hela texten från början av artikeln med *snabbtangenter*
3. Möjlighet att direkt hoppa från artikeltext till gränssnittets omgivande knappar
4. Uppläsning av text som markerats
5. Fungerar i flera webbläsare[3]
6. Går att aktivera från både mobil- och desktopgränssnitt
7. Hastigheten och andra inställningar på uppläsningen kan justeras enligt personliga preferenser
8. Uppläsningen kan gå tillbaka för att lyssna om
9. Uppläsningen kan hoppa framåt i olika intervaller (ex. ord, mening eller stycke)
10. På något sätt visa att det är en länk, fotnot eller liknande (ex. med en “pling”) som går att ställa in
11. I redigeringsläget kan artikelförfattare lyssna på och korrigera uttalet för en viss text på direkten
12. Går att aktivera från Wikipedia-appen
13. Navigerings av talupplästning för hela texten från början av artikeln med *röststyrning*

Steg 2: Natural Language Processing (NLP)
NLP-komponenten består i huvudsak av två delar: textprocessning och uttalsgenerering.

Egenskaper:

1. Text som ska läsas upp skickas till Wikimedias server (Wikispeech API)
2. API för textprocessning (ex. https och json/ssml)[4]
3. Konvertering av text från rå till uppmärkt text med tillräcklig hög grad av information (d.v.s. uppmärkning av pausering, satsbetoning m.m.; minumum en räcka med fonemsekvenser och textnormalisering/parsning)
4. API för uttalskomponent
5. Struktur för uttalslexikon[5]
6. Automatisk uttalskomponent för ord som inte finns i lexikon

---

[3] Se här för urval: https://www.mediawiki.org/wiki/Compatibility#Browser_support_matrix
[4] Det kommer att krävas textprocessning (det som händer med texten innan den skickas till talsyntesmotorn) som är anpassad för Wikispeech. Artiklarna innehåller uppmärkning och struktur som är relevant för talsyntesen, och denna måste skickas in i syntesens API i ett fördefinerat format.
[5] Här slår man upp “kända” ord och får ett uttal för dem. Det krävs ett lexikon där varje uppslag har vissa obligatoriska fält (minimum är förmodligen det ortografiska ordet, en fonetisk transkription, samt språk). Förutom de obligatoriska fälten, är det önskvärt med ytterligare information (senast sparat, vem som editerat uppslaget, språk på ordet, språk på uttalet, ordklass, särskiljande benämning för olika uttal av homografer, kod för system för fonetisk uppmärkning (IPA, SAMPA, etc), etc).

7. TTS klarar att hantera olika taggar (ex. att det är en bildtext eller tabell som läses upp)
8. Möjlighet att välja bland befintliga uttal för språket (på flera olika sätt, ex brittisk eller amerikansk)
9. Generering av prosodiska taggar (emfas, frasering - antingen från data eller via regler)

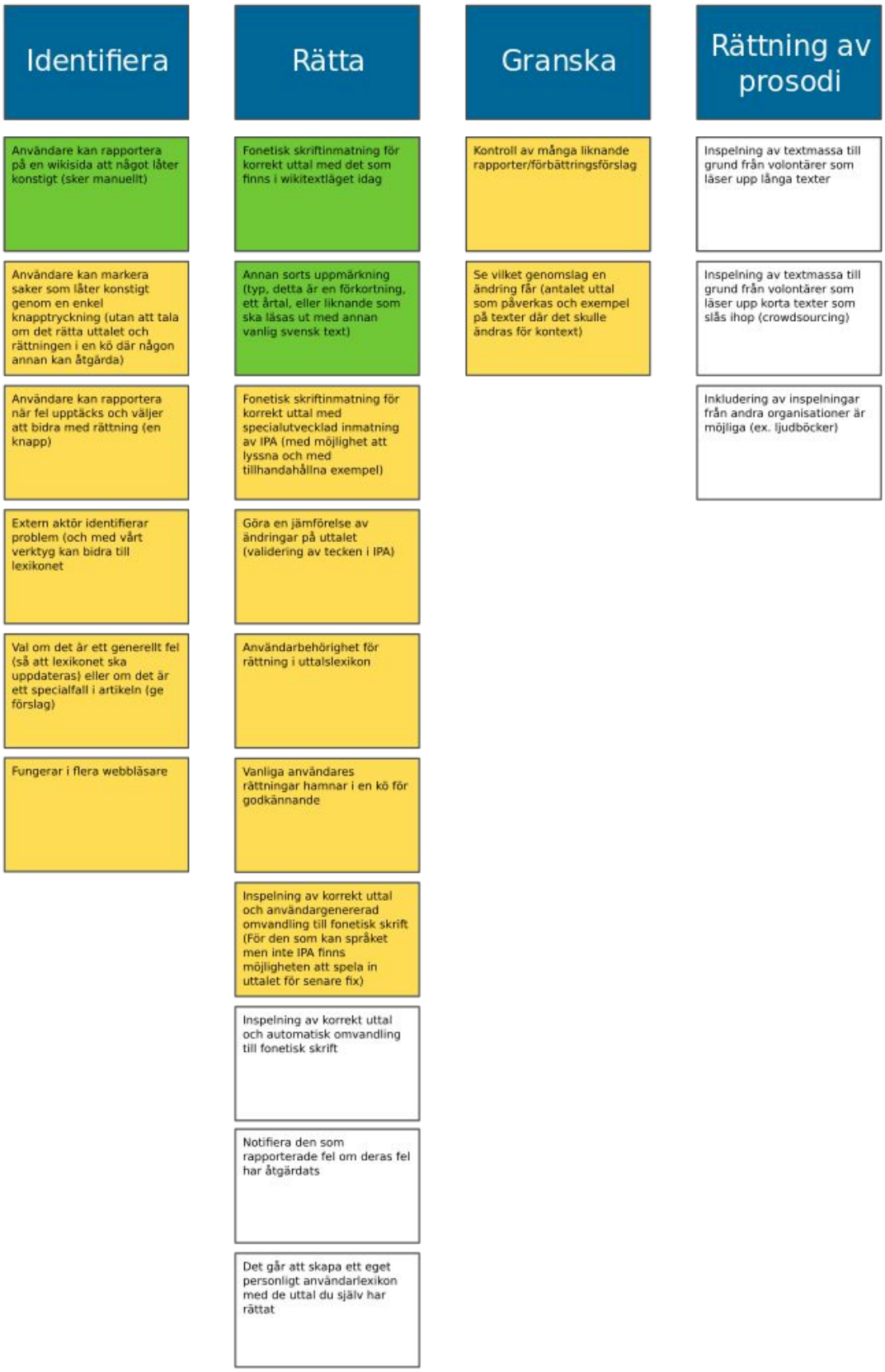
Steg 3: Syntesmotor
Egenskaper:

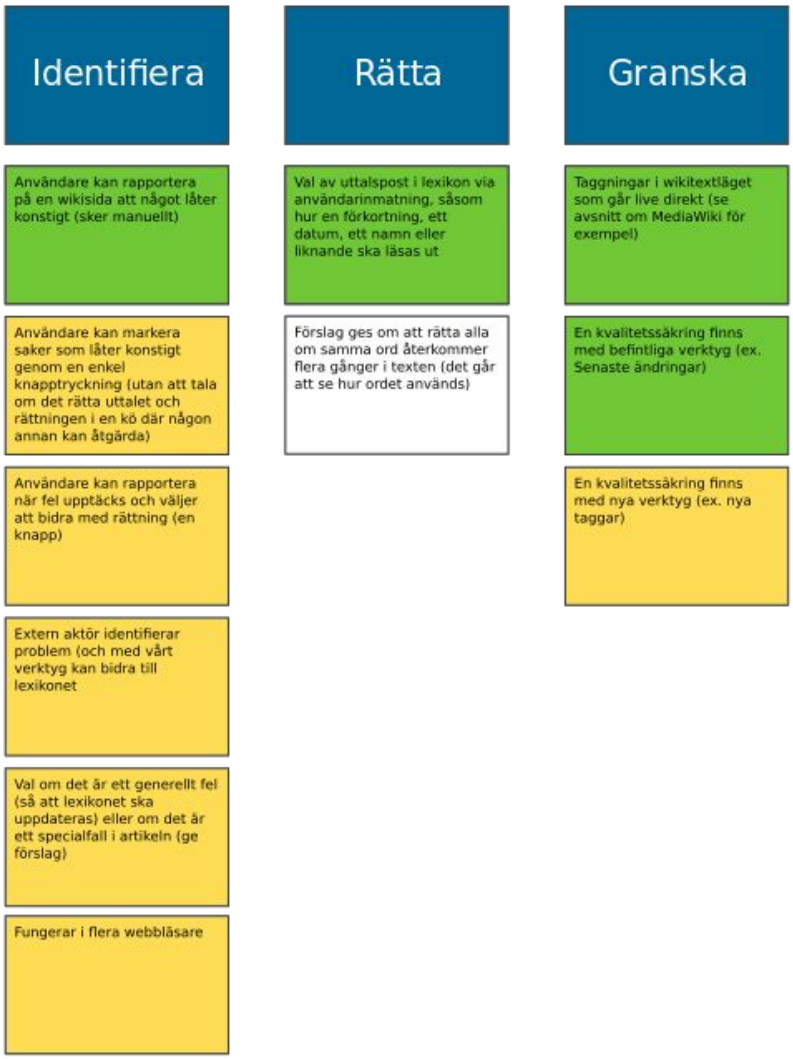
1. Syntesmotorhantering (den generella)
2. Syntesens API för alla motorerna
3. Tolkning av SSML-uppmärkning (av pausering, satsbetoning, intonation, volym och fart)
4. Möjliggöra röstval (exempelvis brittisk eller amerikansk)
5. Möjliggöra prosodisk styrning (emfas)
6. Recording API (för inspelade ljudfiler)

Steg 4: Ljudspelaren
Egenskaper:

1. Se vilket ord det är som uttalas
2. Möjlighet att välja bland befintliga röster för språket (exempelvis manlig, kvinnlig)
3. Fungerar i flera webbläsare
4. Ljudfil går att få ut i olika format för nedladdning
5. Använda sina egna röster

Process för förbättring av talsyntes





Steg 1: Felaktigt uppläst text identifieras
Användaren skall på olika sätt kunna hjälpa till förbättra feluttalade ord m.m. i samband med lyssnandet.

Egenskaper:

1. Användare kan rapportera på en wikisida att något låter konstigt (sker manuellt)
2. Användare kan markera saker som låter konstigt genom en enkel knapptryckning (utan att tala om det rätta uttalet och istället hamnar rättningen i en kö där någon annan kan åtgärda problemet)
3. Användare kan rapportera när fel upptäcks och väljer att bidra med rättning
4. Extern aktör identifierar problem och kan med vårt verktyg bidra till lexikonet (ex. inkluderar Skatteverket Wikispeech och förbättrat uttalet på alla ekonomiska termer)
5. Val om det är ett generellt fel (så att lexikonet ska uppdateras) eller om det ska märkas upp lokalt i artikeln
6. Fungerar i flera webbläsare (detta gäller samtliga delar av editorn, dvs. identifiering, redigering i lexikon samt redigering i artikel)

Steg 2a: Rättning sker i uttalslexikon
Förbättringen läggs till centralt och slår igenom på alla sidor.

Egenskaper:

1. Fonetisk skriftinmatning för korrekt uttal med det verktyg som finns i verktygsraden idag
2. Annan sorts uppmärkning (typ hur en förkortning, ett datum, eller liknande ska läsas ut)
3. Fonetisk skriftinmatning för korrekt uttal med specialutvecklad inmatning av IPA/SAMPA, med möjlighet få din inmatning uppläst och förslag baserat på liknande ord
4. Göra en jämförelse av ändringar på uttalet (validering av tecken i IPA) (ge varning vid saker som skiljer sig alltför mycket)
5. Användarbehörighet för rättning i uttalslexikon
6. Vanliga användares rättningar hamnar i en kö för godkännande
7. Inspelning av korrekt uttal och användargenererad omvandling till fonetisk skrift som även kan föras över till Wikidata och Wiktionary. (För den som kan språket men inte IPA finns möjligheten att spela in uttalet. Andra användare kan då lyssna på den och skriva in IPA baserat på inspelningen.)
8. Inspelning av korrekt uttal och automatisk omvandling till fonetisk skrift
9. Notifiera den som rapporterade fel om deras fel har åtgärdats
10. Det går att skapa ett eget personligt användarlexikon med de uttal du själv vill använda men som inte bör användas globalt

Steg 2b: Rättning sker i artikeln
Förbättringen läggs till lokalt på den specifika sidan.

Egenskaper:

1. Val av uttalspost i lexikon via användarinmatning, såsom hur en förkortning, ett datum, ett namn eller liknande ska läsas ut (genom val av rättelsetyp, språk och existerande uttal som går att lyssna på)
2. Förslag ges om att rätta samtliga förekomster av samma ord i texten (det går att se hur ordet används)

Steg 3a: Kontroll av lexikonförändringens kvalité sker
För att undvika klotter och misstag av nybörjare behövs nya verktyg

Egenskaper:

1. Kontroll av många liknande rapporter/förbättringsförslag
2. Se vilket genomslag en ändring får (antalet uttal som påverkas och utdrag från texter där ordet används för att visa hur det brukas)

Steg 3b: Gemenskapsdriven uppmärkning i texten
För att undvika att klotter och misstag av nybörjare läggs till i artiklarna används MediaWikis verktyg

Egenskaper:

1. Uppmärkningar går live direkt efter redigering
2. En kvalitetssäkring finns med befintliga verktyg (ex. Senaste ändringar)
3. En kvalitetssäkring finns med nya verktyg (ex. nya taggar)

Steg 4a: Insamling av taldata
Huvudsakligen för att förbättra talsyntesen, men kan i framtiden även möjliggöra röstbaserade sökningar på Wikipedia. Detta sker möjligtvis som ett separat utvecklingsprojekt.

Egenskaper:

1. Inspelning av textmassa till grund för talsynes; från enskilda volontärer som läser upp långa texter
2. Inspelning av textmassa till grund för talsynes; från flera volontärer som läser upp kortare texter vilka senare slås samman (crowdsourcing)
3. Inkludering av inspelningar från andra organisationer är möjliga (ex. ljudböcker)

Process för att lägga till nytt språk

Detta sker tre gånger redan under projekttiden, för svenska, engelska och ett höger-till-vänster-språk (arabiska) samt initialt en gång för att göra dokumentation av själva processen.

Steg 1: Uttryckt intresse för aktivering av nytt språk
Egenskaper:

1. Kommunicering om intresse är möjligt (t.ex. wikisida)

Steg 2: Existerande komponenter identifieras
Egenskaper:

1. Analys av vilka komponenter som finns för textprocessning, lexikon, ljudkorpus och syntes samt vilken nivå de ligger på

Steg 3: Eventuella API-anpassningar (delarna förs in i “wrappern”)
Egenskaper:

1. Manuella anpassningar av komponent-API:erna för de nya komponenterna

Steg 4: Saknade eller dåliga komponenter utvecklas
Detta skiljer sig åt från fall till fall.

Egenskaper:

1. Manuell förbättring eller skapande av de dåliga komponenterna (av utvecklare och/eller experter)
2. Utveckling av lexikonet med hjälp av enkla verktyg (import eller mikro-bidrag)
3. Träning av prosodisk modellering med hjälp av befintligt eller nyinspelat material
4. Utveckling av lexikonet med hjälp av gamefierat verktyg (jfr. Wikidata-game)

Steg 5: Installation
Egenskaper:

1. Manuell installation av utvecklare

Steg 6: Lokal konfigurering
Egenskaper:

1. Manuell konfiguration av utvecklare på serversidan
2. Manuell konfiguration av gemenskapen på wikin (exempelvis möjligheten att namnge olika tekniska meddelanden för den specifika wikin samt styling)

Verktyg och resurser som kommer att behövas

I listan nedan återfinns de olika egenskaperna med komponenter som behövs för att uppfylla dem.

Process för uppläsning

Steg 1: Navigering

Egenskap 1:

Komponentnamn: Möjliggör start och stopp med mus
Beskrivning: Start och stoppfunktion för talupläsning för hela texten från början av artikeln med musklick på spelarknapp.
Licens: MIT/GPL
Existerande: Ja, https://github.com/westonruter/html5-audio-read-along
Att göra: Definiera en särskild spelarknapp för användning på Wikipedia. Javascript för att skicka texten till Wikispeech API, infoga den returnerade ljudfilen och tidsmarkörerna i HTML-dom.

Egenskap 2:

Komponentnamn: Möjliggör start och stopp med snabbtangenter
Beskrivning: Start och stoppfunktion för talupläsning för hela texten från början av artikeln med snabbtangenter. tab, return, space.
Licens: MIT/GPL
Existerande: Delvis, https://github.com/westonruter/html5-audio-read-along
Att göra: Definiera snabbkommandon. Användarinställningar.

Egenskap 3:

Komponentnamn: Möjliggör att hoppa mellan omgivande text och artikel
Beskrivning: Det ska finnas möjlighet att hoppa direkt från artikeltext till omgivande knappar/gränssnitt så att man inte behöver tabba sig igenom alla länkar i artikeltexten för att kunna använda de andra tillgängliga funktionerna (som till exempel redigera) i MediaWiki.
Existerande: Nej.
Att göra: Anpassning av navigering.

Egenskap 4:

Komponentnamn: Uppläsning av text som markerats
Beskrivning: Uppläsning av enbart den text som markerats
Existerande: Nej
Att göra: Se till att uppläsningen kan starta och stanna vid markerad text.

Egenskap 5:

Komponentnamn: Flera webbläsare
Beskrivning: Uppspelning fungerar i flera webbläsare
Licens: MIT/GPL
Existerande: Ja, https://github.com/westonruter/html5-audio-read-along
Att göra: Se till så att bakåtkompatibel kod används och att funktioner kan hanteras i äldre läsare. Genomför tester. Observera att den befintliga koden fungerar med HTML5 och alltså inte mycket gamla webbläsare. Inspiration eller delar av koden kan nog däremot återvinnas.

Egenskap 6:

Komponentnamn: Mobilt och desktop
Beskrivning: Går att aktivera från både det mobila och desktopsgränssnittet av Wikipedia.
Licens: MIT/GPL
Existerande: Delvis, https://github.com/westonruter/html5-audio-read-along
Att göra: Se till att uppspelning är tillgängligt från både det mobila och desktop gränssnittet och inte glöms bort.

Egenskap 7:

Komponentnamn: Variera hastigheten
Beskrivning: Hastigheten på uppläsningen ska kunna justeras enligt personliga preferenser
Licens: MIT/GPL
Existerande: Delvis, https://github.com/westonruter/html5-audio-read-along
Att göra: Hastighet kan ändras i webbläsaren mha. HTML5MediaElement.playbackRate. Andra användarinställningar måste definieras och implementeras

Egenskap 8:

Komponentnamn: Lyssna om
Beskrivning: Man kan gå tillbaka i uppläsningen för att lyssna om på ett stycke
Licens: MIT/GPL
Existerande: Delvis, https://github.com/westonruter/html5-audio-read-along
Att göra: Finns delvis, (klicka på första ordet, eller dra tillbaka spelaren). Lägg till knapp och/eller snabbkommando. Definiera vilka funktioner som behövs, utom play/pause.

Egenskap 9:

Komponentnamn: Hoppa framåt
Beskrivning: Uppläsningen kan hoppa framåt i olika intervaller (ex. ord, mening eller stycke)

Licens: MIT/GPL
Existerande: Delvis, https://github.com/westonruter/html5-audio-read-along
Att göra: Finns delvis, i den meningen att det går att klicka på det ord man vill börja ifrån. Men JavaScript behövs för kortkommandon (samma som egenskap 8)

Egenskap 10:

Komponentnamn: Pling på navigation
Beskrivning: Uppläsningen indikerar på något sätt att något är en länk, fotnot eller liknande (ex. med en “pling”), vilket kan ställas in av användaren.
Existerande: Nej
Att göra: Känna igen vissa taggar och spela upp ett ljud.

Egenskap 11:

Komponentnamn: Fungera i redigeringsläge
Beskrivning: Det ska även gå att lyssna medan man är i redigeringsläget.
Existerande: Nej
Att göra: *Detta är inte en del av MVP och eventuell utveckling sker utanför detta projekt.* Se till så att koden som skrivs är tillräckligt generell för att även acceptera den text som visas i redigeringsläget.

Egenskap 12:

Komponentnamn: Fungera i Wikipedia-appen
Beskrivning: Det ska även gå att lyssna i Wikipedia-appen
Existerande: Nej
Att göra: *Detta är inte en del av MVP och eventuell utveckling sker utanför detta projekt.* Anpassa koden så att den kan användas av Wikipedia-appen, alternativt bygg in det direkt i appen.

Egenskap 13:

Komponentnamn: Navigera med röst
Beskrivning: Man ska kunna styra hela navigationen med hjälp av röstkommandon. All funktionalitet i de övriga egenskaperna ska stödjas.
Existerande: Nej.
Att göra: *Detta är inte en del av MVP och eventuell utveckling sker utanför detta projekt.* Bygg en tal-till-text-modul och koppla mot en input-funktion. Dessa behöver kopplas mot kommandon som finns i den övriga navigationen.

Steg 2: Natural Language Processing (NLP)

Egenskap 1:

Komponentnamn: Wikispeech (API)
Beskrivning: Startpunkt för (vanlig) användning av Wikispeech. Rest-API. Ska stödja Https. GET: tar emot text/HTML, skickar ljud+labels. Möjlighet att fråga vilka röster/språk som finns. Hantera (och ev. spara) personliga inställningar och uppmärkning av input.

Existerande: Nej
Att göra: Definiera och implementera API. Skicka data vidare till NLP och syntes. Lista vilka inställningar och vilken uppmärkning som kan behövas för uppspelning, och vilken komponent som ska använda dem. Kan vara sådant som vilken röst som ska användas, hur okända ord ska hanteras, hur text på andra språk ska hanteras, m.m. Implementera hantering av dem.

Egenskap 2 och 7:

Komponentnamn: API för textprocessning
Beskrivning: API för textprocessning (ex. https och Json/SSML)
Input: text med/utan ssml-uppmärkning
Output: text med ssml-uppmärkning
Existerande: Nej
Att göra: Sätta upp tester, skriva kod, ev. avgränsa vilka delar av SSML som ska stödjas. Definiera och implementera API för textprocessning

Egenskap 3:

Komponentnamn: Omvandla till fonetisk uppmärkning
Beskrivning: Konvertering av text från rå till uppmärkt text
Input: text med/utan SSML-uppmärkning. Hantering av SSML-uppmärkning i indata.
Innehåll: En språk/röst-specifik “lista” med olika textprocessningskomponenter (sifferhantering, pausering, med mera)
Output: text med SSML-uppmärkning
Existerande: Nej
Att göra: Defininera uppmärkningsformat och implementera hantering av detta.

Egenskap 3:

Komponentnamn: Språkspecifika komponenter
Beskrivning: I textprocessningen ska man kunna definiera ett valfritt antal språkspecifika komponenter
Existerande: Nej
Att göra: Definiera hur de enskilda komponenterna ska utformas och definieras.

Egenskap 3:

Komponentnamn: Enkel tokenisering
Beskrivning: Enkel tokenisering som splittar på mellanslag
Existerande: Nej
Att göra: Skriv en generell textprocessningskomponent som splittar på mellanslag

Egenskap 4:

Komponentnamn: API för uttalslexikon
Beskrivning: API för uttalslexikon
Existerande: Nej
Att göra: Designa HTTP-API för att anropa databasen utifrån. Implementera HTTP-API i en liten enkel HTTP-server som i sin tur anropar lexikondatabasen

Egenskap 4:

Komponentnamn: Valideringskomponent
Beskrivning: API/komponent för automatisk validering av transkriptioner. Minimum är att fonemen valideras (bara tillåtna symboler får användas). Sedan kan man lägga till regler för annan språk-/röstspecifik validering såsom syntas/format-validering (exempel: hur kan betoningar placeras ut, hur många vokaler/konsonanter kan det finnas i varje stavelse, hur placerar man ut stavelse-/morfemgränser, hur kan konsonanter kombineras i början/slutet på stavelser). Man kan göra “sanity checks” där man jämför ortografi och transkription, och bedömer vad som är rimligt. Man kan också göra mer specifika regler som hanterar vanligt förekommande delsträngar och så vidare. Möjligheterna är i princip oändliga. Regelsviten ska vara självtestande, så att den som utvecklar reglerna kan försäkra sig om att de fungerar som avsett, och inte motsäger varandra.
Existerande: Nej
Att göra: Designa och implementera API. Sätt upp regelsvit.

Egenskap 5:

Komponentnamn: Struktur för uttalslexikon
Beskrivning: Generell (relations)databas för uttalslexikon. Denna centrala komponent skall kunna innehålla alla uppslagsord för alla språk, med fonetiska transkriptioner och annan information. Det krävs ett lexikon där varje uppslag har vissa obligatoriska fält (minimum är förmodligen det ortografiska ordet, en fonetisk transkription, samt språk). Förutom de obligatoriska fälten, är det önskvärt med ytterligare information (senast sparat, onormaliserad/normaliserad ortografi, vem som editerat uppslaget, språk på ordet, språk på uttalet, ordklass, särskiljande benämning för olika uttal av homografer, kod för system för fonetisk uppmärkning (IPA, SAMPA, etc.), etc.).
Databasen skall användas av respektive text-till-talsystem, som antingen anropar lexikondatabasen direkt eller som bygger ett internt lexikon baserat på innehållet i den generella lexikondatabasen. Databasen skall också fungera som ett generellt uttalslexikon, som kan användas i alla upptänkliga sammanhang. Wikipedia-användare skall kunna lägga till och korrigera ord och fonetiska transkriptioner efter behov.
Licens: I det första steget använder vi relationsdatabasen H2 (http://h2database.com/), vars licenser är MPL 2.0 (Mozilla Public License Version 2.0) eller EPL 1.0 (Eclipse Public License). Databasdesignen kan sedan flyttas till något annat databassystem om så skulle önskas (eller översättas till någon annan typ av databas än relationsdatabas).
Existerande: Nej
Att göra: Definiera strukturen för uttalslexikonet -- lista särdrag/fält och deras inbördes hierarki. Design av relationsdatabasen. Implementera relationsdatabasen (tabeller, relationer, index, etc.) Implementera inläsning av textformatet till databasen. Designa generellt textformat för uttalslexikon som lämpar sig för att överföra till databasen. Implementera enklare validering av textformatet, exempelvis av vilka tecken som är tillåtna och vilka fonemsymboler som får användas i databasen för ett visst språk. Validera att indatan till exempel inte innehåller strängar som är längre än motsvarande fält i databasen

kan hantera, etc. Validera att alla fält som krävs för ett lexikonuppslag finns. Testning av funktion och prestanda.

Egenskap 6:

Komponentnamn: API för uttalskomponent
Beskrivning: Input: ett ord med eventuell tillhörande uppmärkning såsom ordklass, morfologi, unikt id (för speciella homografer), med mera
Output: uttal
Ord som finns i lexikondatabasen hämtas därifrån, övriga ord får ett uttal från den automatiska uttalskomponenten
("finns i lexikondatabasen" kan definieras på olika sätt -- detta bör vara konfigurerbart)
Existerande: Nej
Att göra: Implementera uttalskomponentens API, hantera indata: ord eller flerordsuttryck. Definiera och implementera API för uttalskomponent (input-särdrag/output-särdrag).

Egenskap 6:

Komponentnamn: Automatisk uttalskomponent
Beskrivning: Automatisk uttalskomponent för ord som inte finns i lexikon
Input: ett ord med eventuell tillhörande uppmärkning såsom ordklass, morfologi, med mera
Innehåll: en språk/röstspecifik lista med komponenter som behövs för att generera automatiskt uttal (uttalsregler, ev. morfologisk analys, o.s.v.)
Output: uttal
Existerande: Nej
Att göra: Definiera in- och utdata samt hur man specificerar vilka komponenter som ska köras. Det faktiska innehållet (vilken typ av regler eller statistiska modeller som används) är högst språkspecifikt men alla språk behöver någon form av hantering för okända ord.

Egenskap 8:

Komponentnamn: Välja varianter av uttal
Beskrivning: När det finns olika varianter av uttal ska dessa gå att välja mellan.
Existerande: Nej
Att göra: Se till så att valmöjligheten mellan olika uttal inte begränsas. Implementera en koppling mellan uppmärkt text och specifikt uttal i lexikonet.

Egenskap 9:

Komponentnamn: Generera prosodiska taggar
Beskrivning: Utifrån manuell taggning träna en modul som förutsäger var i texten man ska lägga in prosoditaggar (emfas och frasering).
Existerande: Nej.
Att göra: Vi kommer via crowdsourcing tagga upp hur en professionell läsare har läst nyhetstext. Denna taggning kommer i denna komponent användas för att m.h.a. maskininlärning och/eller regler (ny/gammal information, ordfrekvens) för att förutsäga vilka ord man ska betona och var man ska lägga in prosodiska frasgränser. Detta kommer sedan användas som input till Egenskap 5 i syntesmotorn nedan.

Steg 3: Syntesmotor

Egenskap 1 och 4:

Komponentnamn: Syntesmotorhantering
Beskrivning: Indata: SSML med fonetiska tecken. Kontrollera format, hantera fel, skicka vidare till “driver”.
Utdata: ljudfil + tidsuppmärkning. Ta emot från driver. Kontrollera/ändra format, hantera fel
Driver för MaryTTS, flite. Skicka SSML plus parametrar vidare direkt. För att få labelfil kan man behöva göra två anrop. labelfil från flite (-psdur)
Hantering av ytterligare syntesmotorer. hts_engine direkt, utan MaryTTS
Existerande: Nej, men det finns saker att inspireras av. Hos KTH, STTS, även sådant som Speech Dispatcher, MaryTTS, Festival multisyn, Speect, Espeak, Mbrola.
Att göra: Välja någon/några synteser att stödja, skriv kod.

Egenskap 1:

Komponentnamn: Konfigurering
Beskrivning: Konfigurering, inställningar. Default och argument. Ljudformat, labelformat, hastighet, osv. Röster, språk.
Existerande: Nej
Att göra: Se till att inställningar inte är hårdkodade.

Egenskap 2:

Komponentnamn: Syntes API
Beskrivning: API. Rest. GET/POST: tar emot SSML eller liknande, skickar ljuddata och labelfil. Parametrar för ex. hastighet, ljudformat. Lista röster/språk/syntes-drivers
Existerande: Nej
Att göra: Definiera API, skriva kod för server-side hantering.

Egenskap 3:

Komponentnamn: Tolkning av SSML-uppmärkning
Beskrivning: Tolkning av SSML-uppmärkning. Synteskomponenten kontrollerar format o.s.v. Driver gör om till vad syntesmotorn kan använda (ex. labelfil för hts_engine).
Existerande: Nej
Att göra: Definiera vilket SSML som ska hanteras och implementera detta.

Egenskap 4:

Komponentnamn: Möjliggöra röstval
Beskrivning: Gör det möjligt att ha flera olika röster för samma språk.
Existerande: Nej
Att göra: Säkerställ att infrastrukturen är tillgänglig för playern

Egenskap 5:

Komponentnamn: Möjliggöra prosodisk styrning (emfas och frasering)

Beskrivning: En viktig egenskap för att ett text-till-tal-system ska få acceptans för uppläsning av längre texter är att betoningar faller på rätt ställen. Denna komponent gör det möjligt att realisera betoning (prosodisk emfas) i enlighet med SSML-standarden.
Existerande: Nej
Att göra: Förse träningsmaterialet för HMM-syntes med emfasmarkörer och träna om. Alternativt komplettera med regelstyrd modifiering av grundtonskurva och durationsvektor.

Egenskap 6:

Komponentnamn: Recording API
Beskrivning: Existerande inspelningar av artiklar kan ges uppmärkning som gör det möjligt att spela upp dem på samma sätt som med syntes. API för att göra det möjligt att hämta sådana. I framtiden även spela in och spara, både för direkt lyssning och för att bygga/förbättra talsyntes.
Existerande: Nej
Att göra: Detta är inte en del av MVP och utvecklingen sker utanför projektet. Att göra: Alignering av text vs. ljuddata, format för uppspelning där text och ljud är kopplade på samma sätt som vid syntes. Inspelningsgränssnitt.

Steg 4: Ljudspelaren

Egenskap 1-5:

Komponentnamn: Ljudspelaren
Beskrivning: Denna komponent löser alla egenskaper som implementeras via ljudspelaren.
Licens: MIT/GPL
Existerande: Delvis, https://github.com/westonruter/html5-audio-read-along
Att göra: Anpassa till MediaWiki-gränssnitt, utveckla specialfunktioner, användartester.

Egenskap 5:

Komponentnamn: Använda egna röster
Beskrivning: Gör det möjligt för användaren att “plugga in” sin egen talsyntes-röst.
Existerande: Nej
Att göra: *Detta är inte en del av MVP och eventuell utveckling sker utanför detta projekt.* Se till att det finns ett sätt att koppla API:et mot en lokalt liggande talsyntes-röst. Begränsning blir att det måste följa dess format exakt.

Process för förbättring av talsyntes

Steg 1: Felaktigt uppläst text identifieras

Egenskap 1-3 och 5-6:

Komponentnamn: Ljudspelar-editor-koppling
Beskrivning: I ljudspelaren ska det finnas en koppling som gör det möjligt att ta sig in i editorn och hamna i rätt redigeringsläge.

Existerande: Nej
Att göra: Designa och implementera funktion som tar användaren till editorn och skickar med relevant information in i den.

Egenskap 4:

Komponentnamn: Extern aktör bidrar med rättning
Beskrivning: Rättningarna som matas in ska gå via ett väldefinierat API, vilket också möjliggör för externa aktörer att kunna bidra med rättningar..
Existerande: Nej
Att göra: Se till att uttalslexikonet och processen för att godkänna också är åtkomligt från ett API.

Steg 2a: Rättning sker i uttalslexikon

Egenskap 1-4:

Komponentnamn: Editor
Beskrivning: En editor som kan ta emot rättningar för lexikonet från en användare. Innehåller grundläggande kontroller och validering av det som matas in. Ska per steg 1.6 fungera i flera webbläsare.
Existerande: Nej
Att göra: Utvecklas från scratch. Användartester och implementation.

Egenskap 1:

Komponentnamn: Fonetisk skriftinmatning med det verktyg som finns i verktygsraden idag
Beskrivning: Möjlighet att göra fonetisk skriftinmatning för korrekt uttal via de verktyg som idag finns tillgängliga via den befintlig verktygsraden
Existerande: Nej
Att göra: Utveckla lösning för den här typen av annoteringar.

Egenskap 2:

Komponentnamn: Annan sorts uppmärkning (typ hur en förkortning, ett datum, eller liknande ska läsas ut)
Beskrivning: Ibland kan det räcka med att tagga ordet på ett visst sätt för att uttalet ska bli rätt, snarare än att specificera den exakta transkriptionen. Det kan vara så att man behöver tala om att ett ord är en akronym (och ska bokstaveras), eller att ett sifferuttryck är ett årtal eller ordningstal och inte en “vanlig siffra”. Hanteringen av detta är förstås språkspecifik, men vi behöver en mekanism i Wikipedia som möjliggör att skicka med sådan uppmärkning till syntesmotorn.
Existerande: Nej
Att göra: Utveckla lösning för den här typen av annoteringar.

Egenskap 3:

Komponentnamn: Fonetisk skriftinmatning för korrekt uttal med specialutvecklad inmatning av IPA/SAMPA (med möjlighet att lyssna och med tillhandahållna exempel)

Beskrivning: Specialeditor eller liknande för inmatning av uttal med IPA/SAMPA, med återkoppling
Existerande: Nej
Att göra: STTS gör en editor (som Wikimedia senare kan produktifiera/integrera vid behov), och en webblösning som anropar Wikispeech-servern och kan hanterar/validera transkriptioner som input, slå upp i lexikon för att ge användaren exempeltranskriptioner, och även syntetisera transkriptioner för återkoppling till användaren, eller mata inspelade exempel till användaren (se även egenskap 7). Valideringen är språkspecifik och tas fram som en del av processen för “nytt språk”.

Egenskap 5-6 och 9:

Komponentnamn: Editor-backend
Beskrivning: Denna komponent tar hand om det som görs från editorn och ser till att köer och notifieringar hamnar rätt.
Licens: GPLv3
Existerande: Delvis, Echo har ett notifieringssystem som tillägg kan koppla in i.
Att göra: Köer och flöden anpassas och utvecklas. Inkoppling mot MediaWikis notifieringssystem Echo. Tester och implementation. *Notera att egenskap 9 är inte en del av MVP och eventuell utveckling sker utanför detta projekt.*

Egenskap 5:

Komponentnamn: Användarbehörighet för rättning i uttalslexikon
Beskrivning: Endast behöriga användare ska kunna pusha ändringar till uttalslexikonet
Existerande: Delvis, MediaWiki innehåller redan rättigheter av liknande typ
Att göra: Definiera vilken behörighet som krävs (ny användarbehörighet eller kan man använda/bygga ut befintliga funktioner)

Egenskap 6:

Komponentnamn: Vanliga användares rättningar hamnar i en kö för godkännande
Beskrivning: Icke-behöriga användares rättningar ska köas tills behörig användare godkänt ändringen
Existerande: Delvis, MediaWiki innehåller redan rättigheter av liknande typ
Att göra: Definiera när en föreslagen/köad rättning ska kunna godkännas (av vem, hur många o.s.v.). Implementation och testning.

Egenskap 7:

Komponentnamn: Pronuncify

Beskrivning: Webbeditor för inspelning av ord, som sparas i Wiktionary/Wikidata[6], och "skickas vidare" till processning i uttalslexikonet samt sätts i "kö" för att få en fonetisk transkription.
Licens: Public Domain
Existerande: Delvis, en prototyp för att göra det enkelt att läsa in enstaka ord som sedan laddas upp på Commons i bakgrunden med rätt filnamn.
https://github.com/abartov/pronuncify
Att göra: Behöver justeras så att det inmatade ordet kommer från det som har markerats som felaktigt (d.v.s. indata till programmet). Sedan behöver den uppladdade filen "skickas vidare" till uttalslexikonet, Wiktionary och Wikidata. Utveckla ett kösystem för att kunna transkribera inkomna inspelningar.

Egenskap 7:

Komponentnamn: Wikidata- & Wiktionary-portning
Beskrivning: Wikidata- & Wiktionary-portning av IPA och inspelat uttal
Existerande: Nej
Att göra: Bygg vidare på Pronuncify så att kedjan förlängs.

Egenskap 8:

Komponentnamn: Inspelning och automatisk fonetisk skrift
Beskrivning: En komponent som gör det möjligt för användare att rätta uttal genom att spela in egen röst, d.v.s. utan att skriva fonetisk text, med hjälp av övervakad/begränsad taligenkänning. Kan också användas för att stödja transkription av redan inspelade uttal i Wikidata/Wiktionary.
Existerande: Nej, men bygger på flertal existerande taligenkänningsresurser (akustiska modeller)
Att göra: *Detta är inte en del av MVP och eventuell utveckling sker utanför detta projekt.* Hela komponenten behöver byggas från scratch.

Egenskap 10:

Komponentnamn: Skapa personligt lexikon
Beskrivning: Personliga preferenser för uttal. Exempel om man föredrar uttalet "tjex" av "kex" (tje-ljud och inte k-ljud). Då ska man kunna ha ett undantagslexikon som fixar detta uttal varje gång "kex" förekommer.
Existerande: Nej
Att göra: *Detta är inte en del av MVP och eventuell utveckling sker utanför detta projekt.* En komponent behöver byggas som gör detta. Placering/lagring av det personliga lexikonet, redigeringsmöjligheter m.m. Eventuellt möjlighet att "pusha" delar av detta till det generella lexikonet.

---

[6] På Wikidata finns möjlighet att lägga till uttal för ett objekt. Detta skulle eventuellt kunna nyttjas vid förbättring av språket. I dagsläget finns endast ett par tusen objekt med uttal (vissa av dem har dock uttal på väldigt många språk, äpple finns till exempel på 21 olika språk) men potentiellt sett skulle det kunna vara ett lämpligt gränssnitt för volontärer att koppla inläsningar av enstaka ord mot begrepp på ett enkelt sätt. När det blir skala på detta skulle det kunna bli en väldigt stor resurs för språkutveckling.

Steg 2b: Rättning sker i artikeln

Egenskap 1:

Komponentnamn: Uppmärkning
Beskrivning: Denna komponent hanterar hur uppmärkningen av olika ord i artikeltexten ska hanteras. Detta ska göras på ett sätt som inte stör de vanliga Wikipediaredigerarna, och ska alltså kunna vara dold. Detta kommer att utnyttja synergieffekter med Parsoid, den modell som ligger bakom den visuella editorn för Wikipedia. Uppmärkningen är av typen att en siffra är ett ordningstal, att ett ord ska uttalas med ett annat språk än den befintliga språkversionen. Ska per steg 1.6 fungera i flera webbläsare.
Existerande: Nej
Att göra: Bygg den bakomliggande modellen, med eventuell anpassning mot Parsoid. Se till att den kan redigeras från editorn. Användartester.

Egenskap 2:

Komponentnamn: Förslag om lika rättningar.
Beskrivning: Komponenten ska ge användaren förslag på om samma ord förekommer flera gånger i texten och vid behov genomföra ändringar på dessa också. Lite av en "sök-och-ersätt" och "ersätt-alla" funktionalitet
Existerande: Nej
Att göra: *Detta är inte en del av MVP och eventuell utveckling sker utanför detta projekt.* Bygg en enkel modul för detta.

Steg 3a: Kontroll av lexikonförändringens kvalité sker

Egenskap 1 och 2:

Komponentnamn: Lexikonkontroll
Beskrivning: Denna komponent gör kontroller av förslagen som kommer in i kön till lexikonet för att hitta enkla fel och för att hjälpa de som godkänner. Kontroll av många liknande rapporter/förbättringsförslag. Innehåller funktion så att användaren kan se vilket genomslag en ändring får (antalet uttal som påverkas och exempel på texter där det skulle ändras för kontext). Användaren kan också ange typ/tagg på förbättringarna.
Existerande: Nej
Att göra: Utveckla kontroller och gränssnitt för förslagen.

Steg 3b: Gemenskapsdriven uppmärkning i texten

Egenskap 1:

Komponentnamn: Uppmärkningar går live direkt efter redigering
Beskrivning: När något märks upp i artikeln ska det "gå live" direkt på samma sätt som andra redigeringar på Wikipedia. Det ska alltså ögonblickligen vara tillgängligt för alla andra användare.
Licens: GPLv3

Existerande: Delvis i MediaWiki.
Att göra: Säkerställ att inga andra hinder byggs in.

Egenskap 2:

Komponentnamn: Kvalitetssäkring med befintliga verktyg
Beskrivning: Se till att de befintliga kvalitetskontrollsverktygen såsom senaste ändringar och loggar kan användas för att kontrollera de förändringar som görs.
Licens: GPLv3
Existerande: Delvis, i MediaWiki
Att göra: Se till att nödvändiga kopplingar mot loggar, historik, bevakningslista etc. görs. Säkerställ att inga andra hinder byggs in.

Egenskap 3:

Komponentnamn: Kvalitetssäkring med nya verktyg
Beskrivning: Gör det möjligt att kvalitetssäkra med andra verktyg, speciellt via taggar/märken som gör det tydligt vad redigeringens källa är.
Licens: GPLv3
Existerande: Delvis, genom MediaWiki (liknande funktionalitet finns i andra tillägg).
Att göra: Se till att när uppdatering av lexikon eller lokalt i artikeln det skapas relevanta taggar/märken . Säkerställ att inga andra hinder byggs in.

Steg 4: Insamling av taldata

Egenskap 1:

Komponentnamn: Inspelning långa texter för modellering
Beskrivning: Möjliggöra för användare att skapa helt egna talsyntesröster genom att läsa in sin egen röst via ett webbgränssnitt. Gränssnittet promptar texten som ska läsas (som typiskt hämtas från Wikipedia) och ger feedback avseende ljudnivå, talhastighet mm. För att bygga en förståelig HMM-syntesröst krävs ca 20 min tal, men mer tal krävs för god kvalité. Insamlat taldata kan också komma att ligga till grund för en helt fri databas för taligenkänning.
Existerande: Nej
Att göra: *Detta är inte en del av MVP och eventuell utveckling sker utanför detta projekt.* Utveckla webbgränssnitt för inspelning med HTML5 audio samt och backend med initial analys och lagring av ljudfiler.

Egenskap 2:

Komponentnamn: Inspelning korta texter för modellering
Beskrivning: Ett alternativ till att träna en egen röst från grunden är att använda ett kortare material och adaptera en befintlig syntesröst utifrån detta. Gränssnittet fungerar som ovan, och kommer bygga på samma kod, men kan även ge möjlighet att provlyssna den adapterade rösten.
Existerande: Nej

Att göra: *Detta är inte en del av MVP och eventuell utveckling sker utanför detta projekt.* Utveckla webbgränssnitt för inspelning med HTML5 audio samt och backend med initial analys och lagring av ljudfiler.

Egenskap 3:

Komponentnamn: Inkludering av andras inspelningar
Beskrivning: Ett alternativ till att spela in via webgränssnitt är att ladda upp inspelningar gjorda på annat sätt och bygga röster utav dessa. Alla högkvalitativa röstinspelningar med tillhörande text såsom talböcker kan fungera i detta syfte förutsatt att licensen tillåter det.
Existerande: Nej
Att göra: *Detta är inte en del av MVP och eventuell utveckling sker utanför detta projekt.* Utveckla dokumentation och gränssnitt för uppladdning av filer samt backend för verifiering av format och inital analys.

Process för att lägga till nytt språk

Dokumentation och generellt för alla nya språk

Steg 1: Uttryckt intresse för aktivering av nytt språk

Egenskap 1:

Komponentnamn: Kommunicering om intresse är möjligt (dokumentation)
Beskrivning: Wikisida där processen för att lägga till ett nytt språk finns beskriven.
Wikisida där det går att anmäla intresse för nytt språk, med hänvisning till diskussion där konsensus finns.
Inbyggd enkät där det går att anmäla intresse för nytt språk (typ Upload Wizard fast för wikitextsidor), med hänvisning till diskussion där konsensus finns.
Existerande: Nej
Att göra: Skapa sidor med instruktioner och arbetsflöden på mediawiki.org och/eller meta.wikimedia.org.

Steg 2: Existerande komponenter identifieras

Egenskap 1:

Komponentnamn: Identifierar komponenter som finns (dokumentation)
Beskrivning: Här skapas dokumentation om vilka komponenter som behöver finnas för ett nytt språk samt sätter upp checklistor och infrastruktur för communityn.
Existerande: Nej
Att göra: Skapa sidor med instruktioner och arbetsflöden på mediawiki.org och/eller meta.wikimedia.org.

Steg 3: Eventuella API-anpassningar utvecklas

Egenskap 1:

Komponentnamn: Dokumentera API-anpassningar
Beskrivning: Anger hur man ska lägga till och konfigurera språkspecifika komponenter för API:et
Existerande: Nej
Att göra: Definiera och dokumentera de nödvändiga textprocessningskomponenterna för språken i MVP samt dess tester.

Steg 4: Saknade eller dåliga komponenter utvecklas

Egenskap 1:

Komponentnamn: Manuell förbättring av dåliga komponenter (dokumentation)
Beskrivning: Vissa komponenter för vissa språk kan vara sämre än andra och behöva förbättras. Här skapas en prioriteringsordning på vad som är viktigast att göra för att få till en bra uppläsning samt instruktioner på hur man vanligen kan gå till väga.
Existerande: Nej
Att göra: Skapa sidor med instruktioner och arbetsflöden på mediawiki.org.

Egenskap 2:

Komponentnamn: Utveckling av lexikonet med hjälp av enkla verktyg
Beskrivning: För att underlätta en kickstart på lexikonet kan olika slags verktyg användas, till exempel importfunktioner men även enkla gränssnitt för mikrobidrag vilket lämpar sig väl för crowdsourcing. Tillsammans med detta behövs dokumentationer för hur detta används vid uppsättning av nya språk.
Existerande: Nej
Att göra: Skapa importstöd och/eller ett mikrobidragsverktyg. Skapa även sidor med instruktioner och arbetsflöden på mediawiki.org.

Egenskap 3:

Komponentnamn: Träning av prosodisk modellering med nytt och gammalt material
Beskrivning: Med hjälp av gammalt material och nytt material kan den prosodiska modelleringen tränas upp med hjälp av en mjukvara som används. Ju mer material som kan matas in desto större kvalitet kan fås. Tillsammans med detta behövs dokumentationer för hur detta används vid uppsättning av nya språk.
Existerande: Nej
Att göra: *Detta är inte en del av MVP och eventuell utveckling sker utanför detta projekt.* Utveckla process och dokumentation för inmatning av material för träning.

Egenskap 4:

Komponentnamn: Inspelning med gamifierat verktyg

Beskrivning: Här beskrivs hur man gör för att lägga till ett nytt språk i det gamifierade verktyget för röstinspelning.
Existerande: Nej
Att göra: *Detta är inte en del av MVP och eventuell utveckling sker utanför detta projekt.* Skapa sidor med instruktioner och arbetsflöden på mediawiki.org.

Steg 5: Installation

Egenskap 1:

Komponentnamn: Manuell installation av utvecklare (dokumentation)
Beskrivning: Här finns instruktioner riktade mot utvecklare om vad som behöver göras vid installation av ett nytt språk.
Existerande: Nej
Att göra: Skapa sidor med instruktioner och arbetsflöden på mediawiki.org.

Steg 6: Lokal konfigurering

Egenskap 1:

Komponentnamn: Manuell konfiguration på server (dokumentation)
Beskrivning: Här skapas variabler för hur servern kan/bör konfigureras baserat på önskemål från communityn samt vilka komponenter som ingår. Detta ska också dokumenteras.
Existerande: Nej
Att göra: Programmera varieabeluppsättningen. Skapa sidor med instruktioner och arbetsflöden på mediawiki.org.

Egenskap 2:

Komponentnamn: Manuell konfiguration av gemenskap (dokumentation)
Beskrivning: Hör skapas instruktioner för gemenskapen om hur de kan konfigurera tillägget lokalt.
Existerande: Nej
Att göra: Skapa sidor med instruktioner och arbetsflöden på mediawiki.org samt sidor för den lokala plattformen.

Språk i MVP

I Bilaga 5: Nytt språk svenska gås de olika komponenterna som ingår för att lägga till ett språk igenom, med svenska som exempel. I utvecklingsarbetet för svenska ingår även komponenter som kommer att återanvändas i efterkommande språk. En del komponenter kan dock skilja sig åt, och/eller vara mer eller mindre färdigutvecklade för engelska och arabiska.

Identifierade resurser

Nedan listas identifierade resurser för de tre språken som kan komma att användas i den kommande utvecklingen.

Språk: Svenska

Komponentnamn: Norska Språkbanken - Svenskt uttalslexikon
Beskrivning: Svenska, danska och norska språkdata i form av inspelningar av bra kvalitet, vidhängande textmanus till inspelningarna samt omfattande uttalslexikon finns att hämta på norska Nasjonalbibliotekets språkbank (http://www.nb.no/sprakbanken/). Dessa är ursprungligen framtagna för bland annat talsyntes, men behöver processas på olika vis.
Licens: CC0 för Svenskt uttalslexikon
http://www.nb.no/sprakbanken/show?serial=sbr-22&lang=nb (licenserna varierar från resurs till resurs)

Språk: Svenska

Komponentnamn: Norska Språkbanken - Svensk taldatabas för talsyntes
Beskrivning: Svenska, danska och norska språkdata i form av inspelningar av bra kvalitet, vidhängande textmanus till inspelningarna samt omfattande uttalslexikon finns att hämta på norska Nasjonalbibliotekets språkbank (http://www.nb.no/sprakbanken/). Dessa är ursprungligen framtagna för bland annat talsyntes, men behöver processas på olika vis.
Licens: CC0 för Svensk taldatabas för talsyntes
http://www.nb.no/sprakbanken/show?serial=sbr-18&lang=nb (licenserna varierar från resurs till resurs)

Språk: Engelska

Komponentnamn: The CMU Pronouncing Dictionary
Beskrivning: Omfattande, fritt tillgängligt uttalslexikon för amerikansk engelska (http://www.speech.cs.cmu.edu/cgi-bin/cmudict)
Licens: Open source, men oklar licensiering

Språk: Engelska

Komponentnamn: CMU Arctic
Beskrivning: CMU Arctic-databasen innehåller fritt tillgängliga inspelningar inklusive fonetisk uppmärkning för engelska röster med amerikanskt, kanadensiskt, skotskt och indiskt uttal (http://festvox.org/cmu_arctic/)
Licens: Ej tydligt uppmärkt, men fritt licensierat

Språk: Arabiska

Komponentnamn: Arabic speech corpus
Beskrivning: En talsamling (http://en.arabicspeechcorpus.com/) är inspelad på syd-levantin-arabiska (Damascus acccent) i en professionell studio.
Licens: CC BY-NC-SA http://creativecommons.org/licenses/by-nc-sa/4.0/. Licensen är inte

tillräckligt fri och kommer inte att ändras den närmsta tiden. Viss användning kan dock vara möjlig då diskussioner förs med ägaren.

Tidsplan

Arbetspaket	**Månad**	**Kommentar/beskrivning**
Systemspecifikation	Mars 2016-september 2017	Huvuddelen gjord i Etapp 1, men uppdateras med agil metod även efter.
Produktbeskrivning	Mars-april 2016	Läggs på mediawiki.org. Är en grund för kommunikationen med relevanta aktörer.
Informera relevanta aktörer	Mars-maj 2016	Wikimedia-gemenskapen samt andra utvecklare (arbetet påbörjades under förstudien). Strukturera ärandehanteringssystemet, sätt upp wiki-sidor, mailutskick.
“Wrapper” sätts samman	Mars-juni 2016	Få upp en “wrapper” med de olika delarna på Wikimedia Labs. D.v.s. API:er och den grundläggande infrastrukturen.
Testspecifikation skrivs	April-maj 2016	Initialt ett dokument som sätts upp som grund för KTH:s arbete med testerna. Därefter en specifikation av testaktiviteterna vid varje sprint. Inkluderar en Testplan samt Användartestspecifikation och Testmiljöspecifikation.
Etapprapport 1	Maj 2016	Skickas till PTS.
Gränssnittet utvecklas för uppspelning	Mars-september 2016	En del måste utvecklas när wrappern är färdig.
Gränssnittet utvecklas för rättning	Mars-september 2016	Beroende av att wrapper och gränssnitt är klara för att kunna färdigställas.

Release notes	Juni 2016-augusti 2017	Sker kontinuerligt vid varje versionsrelease. I enlighet med hur detta görs av WMF (dvs. branch cuts med alla ingående commits/phabricator tasks).
Rapport Enhetstester	Juni 2016-augusti 2017	Enbart automatiska rapporter. Sker kontinuerligt vid varje kodändring.
Användarmanual & FAQ för slutanvändarna	Juni 2016-augusti 2017	Uppdateras kontinuerligt under projektet efter feedback. Är en översättningsbar sida på mediawiki.org så att instruktionerna enkelt kan översättas och samtligt göra det lätt att hålla informationen uppdaterad.
Testa funktioner	Juli 2016-februari 2017	Ordna testgrupper från gemenskapen och handikapporganisationer som prövar verktyget (wrapper och gränssnitt) och ger kommentarer. Rapporter på detta färdigställs i mars 2017.
Intrimning av de olika delarna	Maj-november 2016	Baserat på kommentarer om språk, buggar, prestanda m.m.
Etapprapport 2	September 2016	Skickas till PTS.
Fortsättningsprojekt	Augusti-oktober 2016	Definiera vad som behövs för fortsättningsprojekt och skriv ansökningar. Detta sker utanför det PTS-finansierade projektet, men bygger på resultatet.
Uppsatsämnen	Juli 2016-augusti 2017	Definiera lämpliga uppsatsämnen som studenter kan arbeta med (ex. moduler).
Etapprapport 3	Februari 2017	Skickas till PTS.
Användar-dokumentation för	Mars-augusti 2017	

installation och drift		
Wikimedia Foundation aktiverar Wikispeech-extensionen	Februari-september 2017	Praktikaliteter om implementationen samt utbyte av code review.
Etapprapport 4	Juni 2017	Skickas till PTS.
Slutrapport	September 2017	Ersätter Etapprapport 5. Projektet slutar 15 september.