“Play PRBLMS”: Identifying and Correcting Less Accessible Content in Voice Interfaces

Aaron Springer
Spotify, University of California Santa Cruz
Santa Cruz, CA & Boston, MA, USA
alspring@ucsc.edu

Henriette Cramer
Spotify
San Francisco, CA & Boston, MA, USA
henriette@spotify.com

ABSTRACT
Voice interfaces often struggle with specific types of named content. Domain-specific terminology and naming may push the bounds of standard language, especially in domains like music where artistic creativity extends beyond the music itself. Artists may name themselves with symbols (e.g. M△S‣C△RA) that most standard automatic speech recognition (ASR) systems cannot transcribe. Voice interfaces also experience difficulty surfacing content whose titles include non-standard spellings, symbols or other ASCII characters in place of English letters, or are written using a non-standard dialect. We present a generalizable method to detect content that current voice interfaces underserve by leveraging differences in engagement across input modalities. Using this detection method, we develop a typology of content types and linguistic practices that can make content hard to surface. Finally, we present a process using crowdsourced annotations to make underserved content more accessible.

Author Keywords
Voice, music, natural language processing, findability

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous;

INTRODUCTION
Voice is a rapidly growing modality used to find and access a variety of content. Voice assistants are now used by 46% of United States adults [32]. Despite this rapid growth, voice interfaces may impact accessibility of content both positively and negatively. Content with long but simply pronounced names may be easier to access by voice compared to onerous text input. Other content may become inaccessible to users because of ambiguous pronunciations or automatic speech recognition (ASR) limitations. These changes in accessibility are an example of interface bias [3]; words that are easy to *type* may be less easy to *say* for particular populations. Another voice complication is that people may ask for the same content in different ways. People may not all agree on the same pronunciation, therefore confounding even a voice system trained ‘the right way’. These complications can make it hard for users to find the content that they want, and could disadvantage specific audiences. The accelerating deployment of voice interfaces combined with possible issues accessing specific types of content make it essential that we develop practical ways to examine these issues. We need methods to identify difficult voice commands and inaccessible content for users; furthermore, we need methods to rectify these issues.

Music is one of the primary use cases for voice-enabled devices [48] but music is also associated with challenging and evolving socio-linguistic practices [12]. Music artists bend and extend language in ways that current voice systems do not accommodate. Take, for example, a user familiar with an artist from an on-screen interface, and asking a voice interface to play that artist: MSTRKRFT. A less informed user may assume the intended pronunciation is spelling the name one letter at a time, “M-S-T-R-K-R-F-T”. Other users may have seen similarly titled artists with dropped vowels and choose to pronounce the artist “Mister-craft” or “Mystery-craft”. Each of these pronunciations are reasonable. However, all these may be incorrect if the artist intended their name to be pronounced “Master-craft.” Even when pronounced correctly, a voice system may transcribe the phrase “Master craft”; this transcription has a large edit distance to “MSTRKRFT”, potentially rendering the artist unfound and the user frustrated.

Many more classes of content that are equally hard to surface using voice interfaces. Tracks such as ‘hot in herre’ have intentional alternative spellings. Some tracks use non-word sounds as their titles: OOOUUU by Young M.A. (spoken as “ooo-ooo” like the “oo” in “cool” with a descending tonal inflection starting in the middle) and Skrt (the sound that car tires make when skidding). Other artists use orthographically similar symbols as letter replacements, like 6LACK (pronounced “black”). Content titled using numbers also present a surprising amount of confusion: Quinn XCII is spoken as “Quinn Ninety-Three” and tracks like “Twenty-8” mix numeric representations. Users who want to access such content will face difficulty.

CHI 2018, April 21–26, 2018, Montréal, QC, Canada.

ACM ISBN 978-1-4503-5620-6/18/04...$15.00.
https://doi.org/10.1145/373574.3173870

Language, names [18], music trends, and subcultures' terminology evolve [10]. The changing context of language makes it imperative that we find ways to dynamically assess challenging content classes beyond the examples above. Music services have millions of tracks, any of which could present a problem to voice interfaces. Manually combing through all of this content and testing it on voice interfaces is an infeasible task. Even if this task were feasible, end-users may not pronounce content names as expected and may struggle with names where the tester did not. Another option is re-training a speech model on the basis of the full content catalogue with vetted alternative pronunciations. However, this will also not be possible for many developers using off-the-shelf speech recognition APIs; nor feasible when the content space is extremely large with millions of items. Alternatively, the information retrieval and computational linguistics literature contains a multitude of large-scale approaches to learning aliases [7] from web corpora and learning machine transliterations [22] of terms from one language to another. However, light-weight approaches for voice applications in specific domains are still necessary. This especially applies when a multitude of unknown, varied linguistic issues are present. Each issue class could require dedicated detection and modeling efforts including constructing or acquiring a training corpora. Pragmatic, scalable ways are needed to find potentially problematic content classes so this content can be made more accessible.

Our contributions are three-fold:

- We present a method to automatically recognize content that is problematic for voice interfaces by leveraging engagement differences across input modalities, and apply it in a music case study
- We provide a typology of the content and practices difficult for voice interfaces to correctly surface
- We develop and test a process to make this content accessible, and describe challenges and considerations when applying such a process

Note that we use 'accessibility' here in the information retrieval sense, defined as the ability and likelihood to surface content in a particular system [2,27], in this case voice interfaces. This type of work is essential as toolkits to design voice interactions become more widespread. Few individual developers have the resources to build their own ASR services; thus, many voice system designers will use off-the-shelf solutions. We demonstrate that relying on only off-the-shelf APIs may not suffice for certain content types. However, these APIs allow a much broader audience to build voice interfaces, and thus, methods are necessary to support these efforts. Our case study focuses on music, but our methods generalize to other applications.

BACKGROUND

Voice Interaction Challenges

Voice interfaces rely on ASR systems to enable interaction. While different approaches exist, some recent deep-learning ASR systems for example directly map audio to characters, ASR systems are often made up of three different models [37]. The first, the acoustic model, translates the spoken language into distinct units of sound called phonemes; sounds that make up a language [25]. These phonemes are then mapped to words using the second model, the lexicon. For example, the English word "cat" has three phonemes: a [k], [æ], and [t], transcribed together as /kæt/, the lexicon would associate these sounds back to the word "cat". Finally, these words are evaluated and changed according to the language model which is a probability distribution over word sequences.

Speech recognition has recently improved to the point where claims of human parity--in standard speech evaluation tasks--are beginning to surface [46]. However, this does not mean that ASR is a solved problem. Specific types of commands and words can still be hard to recognize. Each ASR technique, from neural networks to HMMs, comes with specific strengths and weaknesses [9]. Difficulties can be created by factors like disfluencies, repetitions, extreme prosodic values (e.g. pitch), and pairs of similar sounding words (e.g. ask/asked, says/said); regional accents and individual differences in pronunciations present additional problems [4]. Specific domains come with their own problems and potential consequences; see Henton's discussion of ASR problems in recognizing medical terms in patient report dictation [20].

Automatic speech recognition systems may exhibit biases with regards to different voices and use of language. Tatman shows that different English dialects result in significantly different accuracy in automatic captions [41]. Other work shows that current natural language processing systems perform poorly with African-American English compared to Standard American English [5,21]. This may mean that creators who use their dialects may be less accessible than those conforming to current ASR expectations. Recent pushes to create more open and diverse speech datasets for voice recognition models have yet to bear fruit [23]; nor will these efforts cover every domain. Many voice applications also re-use training data from applications in other modalities. For example, voice web search will at least partially rely on text web search data. However, voice queries are longer and closer to natural language than typed queries [17] and named content can contain naming atypical of speech or long-form text.

Solutions to Voice Challenges

Different approaches to overcoming voice interface problems exist, ranging from those focused on the interaction model itself to those dealing with the underlying data and algorithms. One approach is detecting when a user is having speech recognition problems and automatically

adjusting the voice dialogue itself. Other approaches may combine voice recognition with on-screen input. Goto et al (2004) demonstrate this, showing options on-screen in response to uncertain voice commands [15]. However, these approaches do not necessarily solve problems where the training data itself does not suffice and particular content is inaccessible. We focus on identification of inaccessible content and solutions through data collection.

A subset of our inaccessible content identification problem is a common ASR problem: recognizing Out-Of-Vocabulary (OOV) terms. Multiple ways are available to detect and deal with out of vocabulary terms. Parada et al. [34] describes 3 ways to deal with (OOV) terms. The first method, filler models, represents unrecognized terms using fillers, sub-words or generic word models The second method uses confidence estimation scores to find unreliable OOV regions [19]. The third and final method Parada describes uses the local lexical context of a transcription region. Other approaches model the pronunciation for OOV terms, see Can et al. [6]. Alternatively, Parada et al. [35] describe how, after OOV regions have been detected in transcriptions, they use the lexical context to query the web and retrieve related content from which they then derive OOV terms, often names. The above methods for recognizing OOV terms often assume that the developer has built a specialized ASR from the ground up and can modify it however they choose. With the advent of large-scale public ASR APIs, this assumption may no longer be true. In addition, our study finds categories of problems that would exist even with a perfect vocabulary.

Crowdsourcing is a relatively common part of dealing with ASR problems. Data collection through crowdsourcing can be used to learn pronunciations for named entities [38], and similar work exists for the generation of search aliases [7]. Ali et al [1] describe the challenge in evaluating ASR output for languages without standard orthographic representation; where no canonical spelling exists. They use crowdsourced transcriptions to evaluate performance for Dialectal Arabic ASRs. Granell and Martínez-Hinarejos [16] use crowdsourcing to collect spoken utterances to help transcribe handwritten documents, combining speech and text image transcription.

However, before such processes can be applied, we first need to assess potential problems occurring within a domain, and their prevalence. We can no longer assume voice application builders roll their own ASR nor that they have access to the internals of their ASR service; this creates challenges to correcting ASR errors that require pragmatic solutions. We deliberately focus on less accessible content to highlight cultural and linguistic practices that are not well-supported by current speech solutions. Our process paves the way for others to detect and fix similar problems in machine learning APIs without access to the internal workings of the models.

Many ASR systems rely on a language model that prioritizes high-probability word sequences over less likely utterances. These probabilities are trained from frequencies of word n-grams in corpora [37]. Probability of co-occurrence is also a significant predictor of ASR error [14]. For example, a popular track, at time of writing, is "Two High" by Moon Taxi. The name of this song is pronounced [tu haɪ] and can correspond to three possible English strings: "to high", "too high" or "two high". Of these strings, "too high" is the most statistically likely, and so when a user asks for the sound string [tu haɪ], a generalized ASR system's language model is more likely to return the written string "Too high." This can be problematic for named content, creating confusion if the user is asking for a current popular track “Two High” or an older popular track like “Too High” by Stevie Wonder. While some ASR APIs accept custom language models and pronunciation dictionaries, these are usually quite limited. The additional vocabulary words still need to be detected, generated and supplied with a probability. This is especially an issue for domains where creative language usage is valued (e.g. music, art), or systems used by audiences with diverse linguistic backgrounds. These errors may require downstream solutions if the developers use off-the-shelf APIs where language models are not directly modifiable.

Creative Online Language Usage

Non-standard usage of language and symbols is a common practice when communicating. Text messaging doesn’t always follow standard spelling and grammar [43]. Features like emojis are used in variety of functions ranging from adding shared meaning to making an interaction more engaging, complementing or even replacing text [8]. Similarly, online l33tsp34k, replaces letters with digits or other ASCII symbols, and has been around for decades. Even with minimal exposure, people are readily able to translate words in their ‘l33t form’[36]. While the practice in art dates to at least the 1920’s (see e.g. the Dada poem w88888888 [40]), l33tsp34k’s origins aimed to make content harder to automatically process. This allowed to circumvent filtering of ‘forbidden words’ [36].

Language and Music

Language and music have a complex, intertwined relationship. Verbal language is integral in many types of music, music itself can be seen as language, and specific language is used to describe music; each of these constitutes its own whole field of study [12]. People use language to indicate belonging to specific social and cultural groups. Focusing on a Texas country community, Fox [13] discusses music as a identity preservation tool, and the importance of preserving linguistic *forms,* rather than solely meanings. Mastery of a specific language can tie a speaker to a community; Cutler et al. [10] describe the phonological, grammatical, and lexical patterns that together form the linguistic style of American hip-hop. Additionally, Cutler explores the blending of local influences, including code switching with languages and

dialects in Western European hip-hop. Such blending is also described by Dovchin [11] and exists in J-Pop blending English with Japanese lyrics, citing Western influences [31]. These cultural and linguistic practices have consequences for voice interactions. Differences in language use in different genres could cause differences in the accessibility of their content. Unintended biases can arise in what is *not* accessible.

METHODS

In order to ensure that all content can be found via voice we must first understand which content is less accessible through voice interfaces. However, identification and classification of this content is not enough. We must then develop a method to improve the accessibility of the identified content. This results and methods section is structured in two parts:

- **Identification:** We present a method for identification of named content less accessible through our voice interface.
  - We describe the choices and trade-offs that have to be made during this process.
  - We analyze and describe the characteristics, including sociolinguistic practices, of this less accessible content.
- **Correction:** We present a way to correct these issues through a crowdsourcing method. We discuss pragmatic challenges and considerations in the application of this process. We then examine results of implementing this process and its performance improvements.

We apply this process in a music voice case study.

Prototype and Infrastructure

The authors were part of a team that developed an experimental mobile voice prototype to access music streaming service Spotify. This prototype was in use by thousands of end-users during this study. Voice requests for music through this interface were transcribed to text using an off-the-shelf ASR API service. Audio is sent to the API through the internet and then the prototype receives the most likely transcription in response. After the ASR API returns a transcription, the transcription is submitted to a search API connected to an index of track identifiers. This work is not meant as an evaluation of these component services' performance; such evaluation is highly domain dependent and machine learning APIs change over time. This work is an investigation of the *classes* of problems that developers should be ready for when using general purpose speech recognition services in specific domains, in our case music.

The prototype uses a hosted ASR provider; and thus did not have complete control over the ASR language model or lexicon. There are a number of practical challenges in this common type of set-up: the API is a black box to our prototype, we cannot modify the internals, the ASR vocabulary is not available to examine and is ever-changing, and not specialized for specific domains.

The ASR API, as is a common feature, has a mechanism for adding custom vocabulary. Terms can be added to the lexicon with automatically derived pronunciations at runtime, and used to boost n-gram probabilities in the language model. These often have limitations. ASR APIs restrict the number of terms that can be added and/or considered at runtime. For a music application, a user could request any one of millions of artists and tracks. Tens of millions of users request tracks and artists every month that employ linguistic practices problematic for standard ASR systems. The problem is even more pronounced for less popular long-tail content which is less likely to enter ASR API vocabularies. It was not possible to add all these track and artist names, and their multiple pronunciations by different audiences, as vocabulary additions need to stay within API limits. For a catalogue with millions of tracks, each requiring multiple vocabulary variations, this is not feasible. Localization and personalization may help narrow down potential vocabulary additions, but its constraints would still limit and bias the search space. This type of personalization also requires infrastructure that can be costly to build and maintain. Foreshadowing our results, we found that ~7% of the content examined in this study would be affected by ASR limitations and that only 5 of 12 identified problem categories would have been solved by vocabulary additions.

Even if custom vocabulary input would be added, inaccessible content still needs to be detected and vocabulary additions generated. In addition, the entity to ASR output links we create using our method can be used to improve other services, such as textual search performance by accounting for users misspelling names they have only heard.

Identifying Underserved Content

Refinement of Method

Our first priority was to determine if the prototype suffered from differing levels of accessibility for different content. We mimicked a manual editorial process to assess quality for the most popular US-content, as counted by streams in the week of July 28th, 2017. One researcher, a male, US-native, Standard American English speaker, manually attempted to play each of the most popular 200 tracks using the voice interface. This process explicitly focused on ASR misrecognition of named entities and not any other cause for lower voice performance; we ensured that all requests were in a syntax that would result in the correct result as long as the named entities were recognized correctly by the ASR (e.g. 'Play [track name] by [artist]'). This manual editorial process was simply to validate our hypothesis that the prototype had difficulty with specific types of content.

Of the 200 tracks examined, around 7% could not easily be found using the voice interface. Some of these tracks were

still accessible through spelling the entire track title aloud letter by letter or by mispronouncing the title in a way that cued the voice interface correctly. This method of identifying underserved content is informative, but clearly does not scale; manually checking millions of named content entities is not an efficient option. This approach also contributes bias itself, as the editor or researcher asking for the content has a specific accent and displays pronunciation patterns that may not be representative among the target application's population. Even when users may be able to identify the occurrence of problems, assessing the severity and impact of errors in ASR output is hard for human judges [28]. Due to these factors, we developed a method to identify underserved content in a more generalizable and scalable way.

Identification at Scale

To identify underserved content at scale, we leverage the differences in input modalities across platforms the service is presented on. For example, a user searching for the artist 'A$AP Ferg' can easily type in those characters and surface the track using the mobile or desktop client but may encounter issues using voice. They may pronounce A$AP as [ei sæp], spoken as 'a-sap', or spell it aloud as 'A-S-A-P' or 'a-dollar-sign-a-p'. These pronunciations will all result in different ASR transcriptions. A voice interface may not surface the correct artist for many of the possible pronunciations. The variability of pronunciations creates a disconnect between accessibility of content on voice interfaces compared to other mediums. Therefore, if content is very popular on the desktop and mobile interfaces and not popular on the voice interface, this may indicate that the users are not able to surface the content easily.

We create a univariate measure of voice accessibility in order to use anomaly detection techniques. For each track t, we calculate the voice findability $t_{findability,}$ by dividing t_s, the total number of streams that track has experienced by t_v, the total number of voice finds the same track has experienced. The resulting distribution follows a power law distribution that we log transform in order to normalize for better anomaly detection. This equation for voice findability is shown below. For related examples of accessibility metrics,

$$t_{findability} = log\left(\frac{t_s}{t_v + 1}\right)$$

see [2,29]. We define an anomaly as a track with findability that lies over 1.645 standard deviations from mean findability. This threshold corresponds to a one-tailed t-test at p=.05. This threshold is lower than common anomaly detection thresholds at 2 or 3 standard deviations from the mean. We chose this threshold because false positives have a small cost in this context.

Limitations and Considerations in Detecting Tracks

The difference between popularity or finds in a voice versus a non-voice context may be caused by other issues or behavioral differences between platforms. For example, voice users may have different demographics, or situations where voice is used more may be associated with different types of music or playlists. These differences in our data would show up as false positives, anomalous tracks detected by the findability metric that are actually voice accessible. Our procedure was intentionally liberal with the definition of an anomalous track because false-positives are inexpensive (the cost of a limited number of crowdsourced utterances as described in the next section) whereas false-negatives could lead to content being inaccessible. We will later show that this anomaly identification method was accurate in surfacing tracks that are inaccessible by voice.

IDENTIFYING CONTENT RESULTS

Voice Interfaces May Underserve Specific Genres

We applied the anomaly detection procedure to the top 5000 tracks in the 28-day period from July 28th to August 24th, 2017. Before we provide a typology of the less accessible content, and its naming, we first examine the anomalous content through the lens of musical genre in order to gain a clearer view of the content that voice interfaces struggle to surface.

In order to discern which genres are underserved, we examine the proportions of genres in the top 5000 most streamed tracks as compared to the proportions of genres in the English-language titled anomalies from the top 5000 tracks. For example, if ⅕ of the top 5000 tracks are pop tracks but ⅗ of the anomalies are pop tracks, this may indicate that pop tracks are less accessible than other content. If all content were served equally then the proportions between the Top 5000 tracks and the anomalies would be the same. This process and its drawbacks parallel work done in auditing search engines for differences in satisfaction due to demographics [30]. As outlined above, differences in proportions could be caused by other demographic or contextual differences in music consumption through voice interfaces. However, this method provides an indication to developers that it would be worthwhile to further investigate accessibility of content in particular genres.

We use Spotify's metagenres that cluster genres, e.g. trap music and rap belong to the hip-hop metagenre. We use these metagenres to have more reliable and interpretable results. Certain genres are overrepresented in the anomaly set, indicating that these genres may contain a larger amount of content that voice interfaces have difficulty surfacing. Hip hop rises from containing 36% of all tracks in the population to 58% of all anomalous tracks. Country music also experiences a disproportionate increase in the anomalous population, rising from 9% in the full sample to 12% of anomalies. This is in line with prior literature, showing that both hip-hop [10] and country music [13] have their own specific sociolinguistic practices.

Pop music goes in the reverse direction, indicating that pop music does not have as frequent issues with voice

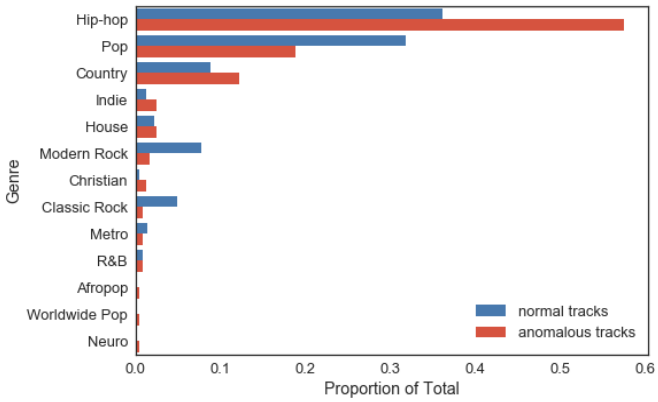

Figure 1. Genre Representation in Full Track Set and Anomalous Track Set

interfaces. In the overall sample, pop contains 32% of the tracks; in the anomalous sample, pop only contains 18% of the tracks. Rock genres experience quite large decreases, suggesting that they may struggle the least with voice interfaces; classic and modern rock combined drop from being 12% of the overall sample to only 2% of the anomalous sample. In order to test for significant differences in major metagenres we eliminated 8 genres that had less than 5 tracks in the anomalous category. This limits us to only making conclusions about the changing distributions of the hip-hop, pop, country, indie, and house genres. Based on these 5 metagenres, the anomalous genres differ significantly in their distribution from the standard genre distribution as indicated by a Chi-Squared test for homogeneity was significant $X^2(4, N=2075) = 1421$, $p < .0001$. We cannot be completely sure that this difference is due to voice user interface problems and not demographic differences between typical users and voice users. However, our later results on accuracy of our classification indicate that much of this variation is likely due to voice interface challenges.

Typology of Underserved Content

We now qualitatively examine the classes of content that suffer from inaccessibility due to their titles or names. This typology was created by coding the anomalies from the top 5000 tracks by number of streams in the 28-day period from July 28th to August 24th, 2017. One researcher went through the anomalies and organized them into prototype categories based upon their characteristics that created problems within the ASR system. This process created 11 different categories of content. Following this prototyping of categories, the two researchers annotated a sample of 100 of the anomalies in order to resolve conflicts and refine the categories until full agreement was reached on all 100. This co-annotation resulted in refining the definitions of 5 categories and the addition of a new category. The final typology consists of 12 categories of titles that are problematic for ASR systems.

English Dialects and Neologisms

English Dialects and Neologisms were defined as track titles that used new words that may contribute to a dialect or track titles that were spelled in a way intended to convey a certain dialect of English speech. Examples include 'You Da Baddest' by Future and 'Any Ol' Barstool' by Jason Aldean. The determiner "da" (pronounced [də]) in 'You Da Baddest' is spoken distinctly from the Standard American English equivalent "the" (pronounced [ðə]) Even though these pronunciation differences are standard in the African American English dialect [42], ASR systems struggle for correctly form this dialect speech and often sanitize it to Standard American English. An example of the relationship between English dialects and Neologisms can be found in the track 'Litty' by Meek Mill and Tory Lanez. 'Lit' has referred to a status of being inebriated since the late 19th century [24]. Recently, in the 21st century, 'lit' has come to mean 'exciting' or 'excellent', pushed in large part by hip hop music [49]. 'Litty' is used as a drop-in replacement for 'lit' but has presented problems for voice interfaces, likely because litty was not in the ASR vocabulary.

Non-English Languages

As discussed earlier, recognizing multiple possible languages in the same system, let alone the same title, is an open problem in speech recognition [45,47]. Current major ASR technology providers require that the implementer specify a single language that will attempt to be recognized. This produces challenges in linguistically heterogeneous regions. We do not attempt to tackle this issue using the method presented in this paper.

Abbreviations and Ambiguous Acronyms

Abbreviations and ambiguous acronyms consist of tracks that include shortened or abbreviated words in their titles or textual cues that imply abbreviation or acronym. Examples of true acronyms include 'E.T.' by Katy Perry and 'She's Mine Pt. 1' by J. Cole. Abbreviations are often ambiguous in their pronunciation. For the above tracks many people would say the first as 'E-T' (pronounced [i ti]) and the second 'She's Mine Part 1' but 'extra-terrestrial' and 'She's Mine P-T 1' would also be valid utterances. An ambiguous acronym can be seen in the track 'LUV' by Tory Lanez, while 'LUV' is intended solely as an alternative spelling, users may interpret the capitalization cues to imply that they should pronounce each letter individually.

Numbers, Dates, and Times

While seemingly simple to represent, numbers, dates, and times also present a challenge for surfacing correct content. For example: 'Twenty 8' by Kodak Black and 'Confessions Part II' by Usher. Similar to the abbreviations class, we have multiple textual representations of the same spoken phrases. 'Confessions Part II' could also be transcribed as 'Confessions Part 2' or 'Confessions Part Two'. This means that properly recognizing and translating between different transcriptions is essential to surfacing the correct content. Similarly, time and date can be represented in different ways; 'seven hundred hours' can be equivalent to 'Seven AM'; '7/11' could be 'Seven eleven', 'July Eleventh', or even 'November Seventh'.

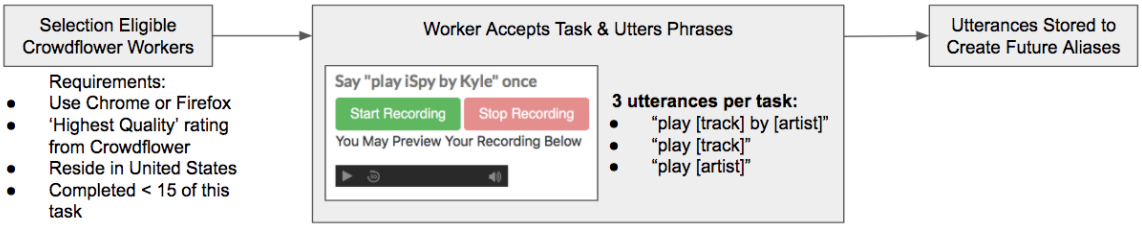

Figure 2. Crowdsourced Utterance Generation for Individual Tracks

Removal of Spaces

Removing spaces in content names can also present challenges. The track title 'DONTTRUSTME' by 3OH!3 is one example of this. Removing spaces can increase the edit distance to the transcription and may result in incorrectly surfaced content.

Vocables

Vocables are modernly defined as utterances that are not words but do contain meaning. Commonly used examples are 'uh-huh' to agree with something and 'ew' to express disgust. Non-lexical vocables, a subclass of vocables that convey no lexical meaning are common in many types of traditional music such as Native American music and Irish music [12]. Today we see vocables in popular music like 'do re mi' by blackbear (or Julie Andrews) and 'OOOUUU' by Young M.A. These are particularly difficult for current ASR technology. Spelling for vocables is not clearly defined and subtle variations in capitalization or spelling may convey prosodic information that is ignored by the ASR. For example, vocalizing 'OOOUUU' like Young M.A. on her track gets transcribed as 'ooh', the exact same transcription as vocalizing the 'Ouu' portion of Lil Pump's track 'Flex Like Ouu'. These two sounds are vocalized quite differently in their respective tracks and current ASR technology does not differentiate.

Non-Replacement Symbols

Artists choose to use symbols in their tracks for many different reasons, a couple include: conveying a specific feeling (***Flawless by Beyoncé) and tagging to contextualize (NAVUZIMETRO#PT2 by NAV). These symbols can also carry implied pronunciation such as Tay-K's track 'I <3 My Choppa'. We cannot simply ignore the symbols when transcribing; if we drop the symbols in 'I <3 My Choppa' we lose an implied word between 'I' and 'My' and will likely not find the correct track.

Orthographical and Semantic Replacement Symbols

Symbols can also be used as replacements to normal letters or words. Common examples of this include the plethora of artists prefixed with 'A$AP'; this is pronounced [ei sæp], spoken as 'a-sap', but many less informed users may try to spell the word. Other artists' names are difficult or completely impossible to form with current voice interfaces such as V▲LH▲LL. Semantically similar replacement symbols include usage of '&' in place of 'and' others like Ed Sheeran's album '÷' (pronounced 'Divide').

Censored Swear Words

Many publishers will censor their own tracks before publishing them by replacing parts of the offensive words with asterisks. This censorship can complicate how easy it is to surface the track using voice. These tracks may be ambiguous, the censored word in 'P**** Print' by Gucci Mane has multiple plausible replacements and only knowledge of the track's lyrics can clarify which is correct.

Expressive and Alternative Spellings

Expressive and alternative spellings are closely related to dialect speech but differ in one key aspect. Alternative spellings are not intended to modify the pronunciation of the word. For example, 'Luving U' by 6LACK is still pronounced /lʌvɪŋ ju/, an identical pronunciation to the more standard spelling 'Loving You'. Alternative spellings may create issues because the actual title can be substantially different than the transcription that the ASR produces. Combinations of alternative spelling and dialects may be particularly challenging for ASR systems, e.g. '100it Racks', pronounced [hənɪr ræks], said 'hunnit racks'.

Wordplay including Homophones, Puns, and Portmanteau

Words with similar pronunciations present issues for ASR systems because they may not be spelled in easily translatable ways. One artist 'Knowmadic' is difficult to surface because ASR will only form 'Nomadic', the name of another artist. Another relatively popular artist, 'Cerebral Ballzy', has an acoustically different name than the disease Cerebral Palsy, but the ASR will only form the name of the disease rather than the band. Presumably the association in the ASR language model between 'cerebral' and 'palsy' is highly probable and varying pronunciations of 'palsy' will not change the transcription.

Names

Proper nouns are a perennial difficulty for ASR systems because of the myriad spelling and pronunciation differences [26,33]. We see evidence of this also. Some artists like SahBabii and NAV have created new names based on shortening their given name (NAV from 'Navraj') or permutations of their given names combined with other words (SahBabii from 'Saheem Baby').

CORRECTING UNDERSERVED CONTENT

Now that we have examined what groups of content may be disadvantaged by current voice interfaces we move to the process needed to fix these accessibility issues. As machine learning technology continues to become commodified, downstream users of these technologies must find ways to adapt these systems to their specific context. Downstream

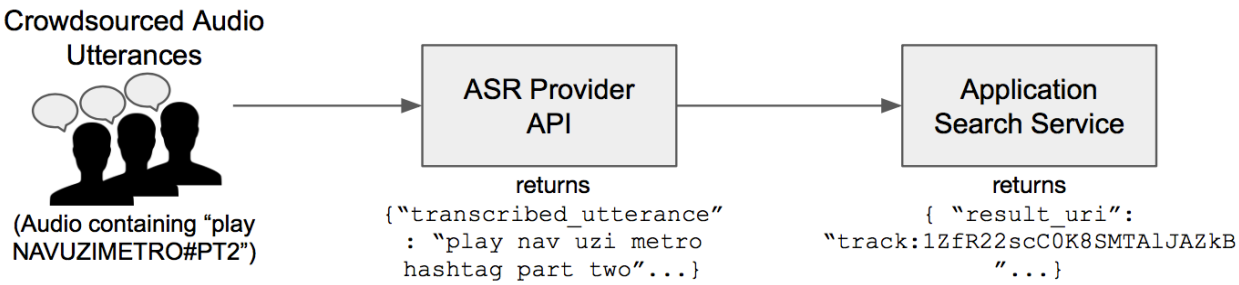

Figure 3. Transcription and Search Process to Resolve URIs

users may not be able to explicitly change the machine learning model or add additional training data. We cannot directly modify our ASR service or add training data, instead we create aliases for ASR mischaracterizations to ensure immediate fixes for underserved content. Each alias serves as a link to the content that the ASR struggles with. For example, the ASR system is unlikely to transcribe the string 'Sk8er Boi' from an utterance and will instead form the more standard English 'skater boy'; this will not surface the right content. To direct the query to the correct content we need to create 'skater boy' as an alias for 'Sk8er Boi'.

A simple way to create content aliases would involve an editor manually saying each of the detected anomalies aloud and recording the transcription from ASR as an alias. The editor may pronounce the track 'LUV' as [ɛl ju vi], said 'ell-you-vee', and then record the ASR's transcription of 'l u v' as an alias for the original track. However, most of the population may actually pronounce 'LUV' as 'Love' and therefore the editor's alias would be ineffective for many users. We need a way to sample the broad pronunciation space for such content that includes many different voices and accents in order to make generalizable aliases [41]. To generate a more diverse set of utterances

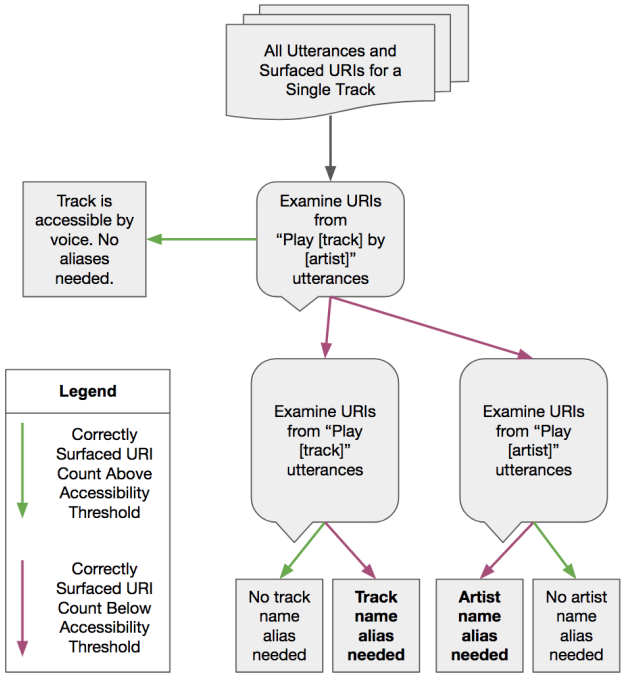

Figure 4. Track Aliasing Decision Process

than a manual editorial process we turn to crowdsourced audio generation. The crowdsourcing process and worker requirements are shown in Figure 2. Workers were paid $0.50 for each completed task; tasks generally took less than 1 minute to complete. We collected utterances until each track had a minimum of 15 utterances for each type. After collection ended, we transcribed all of the utterances using the same ASR and settings that are used by the prototype application in order to ensure that the transcribed content works as an alias in the prototype. This process resulted in a variety of transcriptions for each track and utterance type.

The next step was to verify whether each of the collected utterances resulted in finding the correct content. We used the transcriptions produced by the ASR to calculate which entity would be surfaced if this utterance were made by a user. This process is shown in Figure 3. Each entity is identified by a Universal Resource Indicator (URI). In order to check if an utterance resulted in surfacing the correct content, we compared the original URI, from the track the crowd worker asked for, with the URI that resulted from transcribing and simulating their request. If these URIs matched, then the utterance was considered to be successful in surfacing the correct content.

The previous steps calculated whether any individual crowdsourced utterance resulted in surfacing the correct content. Now we use those steps to make a decision about the performance of the track as a whole. This step in the process verifies that the anomalous tracks our algorithm identified are anomalous due to voice interface difficulties and not due to differences in intent between modalities or failures in another part of the retrieval process. Similar to our anomaly detection threshold, we again set a relatively low accessibility threshold for when a track is accessible. We judged a detected anomalous track to be a false positive anomaly if more than ⅓ of the 'play [track] by [artist]' queries resulted in the correct track URI. In Figure 4, we refer to this URI comparison process as "Examine URIs". This ⅓ accessibility threshold decision indicates that we are looking for tracks that are currently among the least accessible in our prototype. Decisions like these are part and parcel of the practitioner experience, this threshold may need to be set differently for other domains.

We outline the full alias decision process, including false positive decisions, for each track in Figure 4. If the 'play [track] by [artist]' queries do not surface the correct track

URI ⅓ of the time, this indicates there may be an ASR problem with the track or artist name. Examining the track and artist utterances separately allows us to tell whether the track title or artist name is the source of ASR error. We use the same ⅓ correct URI threshold to determine whether or not these utterances are finding the correct content. Once the track or artist name has been identified as causing issues for our interface, we choose an alias from the collected utterances. Aliases are chosen by a simple frequency voting scheme. Aggregating generative work like spoken utterances rather than discriminative crowdsourced work is an open research problem; we use a simple voting scheme to demonstrate that this bias reduction method is robust even with simple aggregation functions.

CORRECTING CONTENT RESULTS

In the next section, we test our accessibility improvement method on the top 1000 most popular tracks from the US, in a 28-day period: July 28th to August 24th, 2017.

Fifty tracks from the anomalous set remained after eliminating anomalous tracks according to our 2 criteria: the track titles were in English and they did not create ethical concerns for our crowdsourcing experiment. In our solution in this paper we focus on English-language tracks. This is a difficult choice because code-mixing and code-switching [11] between languages happens in music applications. However, dealing with code-switching and multiple languages in a single ASR application is an open research problem itself [45,47]. Among the tracks recognized as anomalous were tracks that contained ethnic slurs in the title. These presented an ethical concern for the researchers because we would be paying crowd workers to record themselves saying these slurs aloud. These tracks were discarded for the purposes of this study; note that they would however have to be addressed in live applications as artists can reclaim slurs or use them as social commentary. These types of tracks could be inaccessible if different pronunciations of slurs are not included.

Following the crowdsourcing of aliases for these 50 tracks, ten of the 50 tracks were deemed accessible through voice interfaces and false positives from the anomaly detection process. Recall that we set a low bar for false positives in our methodology, only ⅓ of the crowdsourced 'play [track] by [artist]' utterances had to result in the correct track to be considered a false positive. We wanted to focus in on the most underserved tracks in order to test our method. We did not implement aliases for these 10 false positives.

Parameter	Coefficient	Std. Error	p
(Intercept)	1.771	0.312	< 0.001
Condition	-0.760	0.449	0.090
Time	**0.598**	**0.301**	**0.046**
Condition * Time	**1.445**	**0.428**	**< 0.001**

Table 1. Summary of mixed-effects Poisson Regression

In total, we used the remaining 40 tracks and aliases as input for the crowdsourcing method and alias testing. In order to eliminate potential confounds, we randomly sampled half of the 40 tracks to use as an experimental group to track alias performance. We added the aggregated crowdsourced aliases into a music streaming production environment for the 20 tracks that were in the experimental group. We now examine how the voice finds for these tracks changed after adding the produced aliases.

Aliases Improve Content Accessibility

In order to examine the effect of aliases on the underserved content, we examined the logs of our prototype voice interface. We examine the period directly around the implementation of the aliases in order to control for temporal effects. This time period includes 7 days before and 7 days after the aliases were implemented for the experimental group. As seen in Figure 5, we examine the sum of finds before the alias is implemented compared to after implementation and calculate the percent increase. A majority of tracks experienced explosive growth in their finds through the voice interface.

We test performance of our control and experimental groups using 2 methods. First, we use a pair of Wilcoxon Signed-Ranked Tests to examine control and experimental performance. Due to the fact that we planned two comparisons, we use a Bonferroni correction, thus our α=0.025. A Wilcoxon Signed Rank Test indicates that the tracks in our experimental group were found significantly more using voice after aliases were added, V=1, $p < 0.001$. As expected, our control group did not significantly differ for the same time periods, V=53, $p = 0.093$. In addition, we specified a mixed-effects Poisson regression to better control for between group differences. The model included 2 random effects: track, to control for variation in initial voice finds, and time to control for natural variation in

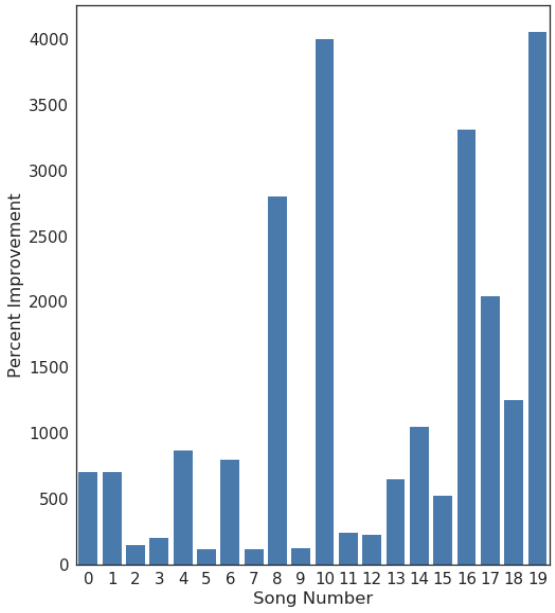

Figure 5. Alias Finds Improvement Over Baseline

voice finds over time. Fixed effects in the model included an interaction between condition and time; we expect an interaction due to the implementation of the aliases in the experimental condition. The coefficient for this model are shown in Table 1. This mixed-effects Poisson regression found a significant interaction between condition and time, $p < 0.001$. Additionally, the time parameter was significant at $p=0.046$, this is likely due to increasing usage of our prototype over the time period. A pair-wise post-hoc test indicated that the experimental group differed significantly before and after aliases were implemented, $p < 0.001$; the control group did not differ on the same time periods, $p = 0.191$. These results indicate that the implementation of aliases increased the accessibility of previously underserved content. A small number of tracks (6) experienced less growth overall, but the data at least illustrates that they were now more accessible than before.

CONCLUSION & DISCUSSION

Voice is a rapidly growing way of interacting with consumer-facing applications. We have presented one approach to identify disadvantaged content which can be generalized to other domains. Voice interfaces are made up of components based on textual, speech, and behavioral data. Groups that are underrepresented in training data, including those with different accents or members of sociolinguistic groups that do not use the majority dialect, will be disadvantaged. Similarly, content less likely to occur in large-scale speech training corpora, may be less likely to be recognized. This makes voice applications particularly prone to biases. Our case study shows that certain genres of content are more affected. We classified 12 linguistic and stylistic practices that present problems in current voice contexts. It is crucial to discover types of content that experience issues in scalable and easy to apply ways. In our evaluation, we showed our method increased accessibility of previously disadvantaged content.

Our method focuses specifically on enabling access to diverse content within the music space but this approach is extensible to many other domains. Developers are increasingly using public ASR APIs similar to what our prototype used. For example, take a developer creating an application containing many local, slang or dialectal terms, or app/company-specific terminology, or profession-specific scientific, medical, legal, industrial terms. While some domain-specialized ASR services are available (e.g. Nuance has medical and legal ASR products), for especially smaller developers with special purpose domains, these may not suffice. Similar issues will arise when automatically making apps voice-accessible; which commands will and will not work may not be clear. Terms may be comparably rare in the data that the general-purpose ASR API was trained on. This rarity in training data could then result in the ASR API transcribing more common similar-sounding phrases or words rather than the specialized terminology needed. Our method could identify these incorrect transcriptions and ensure that they still resolve to the action that the user desired.

Limitations and Trade-Offs for Practitioners

While this method presents a scalable and automated way of addressing accessibility problems, it is important to realize that there are limitations and potential improvements. It is worth considering how each decision in the process may affect the final outcome. Some voice interface problems we identified were related to representation challenges (e.g. multi-lingual content, numbers, dates, and times). Other were related to socio-linguistic practices also identified in music literature (e.g. Hip-hop [10] and country music specific [13]); or in online communication literature, such as l33tsp34k [36]. A tradeoff decision arises: if a particular problematic category becomes large enough, it may be worthwhile to develop a specific solution. However, those can be costly if requiring specific machine learning or domain expertise and datasets. Until then a method such as this one can be applied.

Our evaluation also illustrated the dynamic nature of speech recognition systems. Some problems 'solve themselves'; two tracks in our control group *became* accessible without intervention, potentially through updates in the ASR system. Whether or not a developer can wait for ASR systems to update depends on the domain and expected use cases (e.g. new track releases). The size and variance of the domain-specific named content space will determine the anomaly threshold decisions and annotator decisions (crowdsourced or editorial) necessary. Anomaly detection improvements are possible by ensuring a close match of modality populations. Pronunciations can be provided by a broad population, or one closely matching the target audience, or in-house experts. Cost and access matter here.

Creative Intricacies

Creators are deliberate in the way they name themselves and their content. In some cases, technological considerations are part of this process. We focused on making content accessible. It's worth noting that content creators may have different motivations. Obscurity or findability can both be treasured values. In the genre Witch House, with artists like GL▲SS †33†H, artists may intentionally obfuscate names [44]. In contrast, the electro-pop band Chvrches have claimed to spell their name "using a Roman 'v'" so Google wouldn't confuse the group with actual churches" [39]. Ironically, a general ASR system would have exactly the opposite result for users who pronounce the name correctly: churches would be found, not Chvrches. New interfaces and retrieval techniques may not necessarily align with all communities' practices, nor with content creators' existing technology strategies.

ACKNOWLEDGEMENTS

We foremost thank our listeners, and our crowdworkers. We thank our anonymous reviewers and our colleagues Sravana Reddy, Fernando Diaz, Ben Lambert, Ruth Brillman, and Jenn Thom for their helpful comments.

REFERENCES

1. Ahmed Ali, Walid Magdy, Peter Bell, and Steve Renais. 2015. Multi-reference WER for evaluating ASR for languages with no orthographic rules. In *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*, 576–580.

2. Leif Azzopardi and Vishwa Vinay. 2008. Accessibility in information retrieval. In *European Conference on Information Retrieval*, 482–489. Retrieved September 18, 2017 from http://link.springer.com/chapter/10.1007/978-3-540-78646-7_46

3. Ricardo Baeza-Yates. 2016. Data and algorithmic bias in the web. 1–1. https://doi.org/10.1145/2908131.2908135

4. Catherine T. Best, Jason A. Shaw, and Elizabeth Clancy. 2013. Recognizing words across regional accents: the role of perceptual assimilation in lexical competition. In *INTERSPEECH*, 14th. Retrieved September 18, 2017 from http://www.academia.edu/download/45189070/Recognizing_words_across_regional_accent20160428-24622-16pjop6.pdf

5. Su Lin Blodgett, Lisa Green, and Brendan O'Connor. 2016. Demographic dialectal variation in social media: A case study of African-American English. *arXiv preprint arXiv:1608.08868*. Retrieved September 6, 2017 from https://arxiv.org/abs/1608.08868

6. D. Can, E. Cooper, A. Sethy, C. White, B. Ramabhadran, and M. Saraclar. 2009. Effect of pronounciations on OOV queries in spoken term detection. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, 3957–3960. https://doi.org/10.1109/ICASSP.2009.4960494

7. Pu-Jen Cheng, Jei-Wen Teng, Ruei-Cheng Chen, Jenq-Haur Wang, Wen-Hsiang Lu, and Lee-Feng Chien. 2004. Translating unknown queries with web corpora for cross-language information retrieval. 146. https://doi.org/10.1145/1008992.1009020

8. Henriette Cramer, Paloma de Juan, and Joel Tetreault. 2016. Sender-intended Functions of Emojis in US Messaging. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services* (MobileHCI '16), 504–509. https://doi.org/10.1145/2935334.2935370

9. Michelle Cutajar, Joseph Micallef, Owen Casha, Ivan Grech, and Edward Gatt. 2013. Comparative study of automatic speech recognition techniques. *IET Signal Processing* 7, 1: 25–46. https://doi.org/10.1049/iet-spr.2012.0151

10. Cecelia Cutler. 2007. Hip-Hop Language in Sociolinguistics and Beyond. *Language and Linguistics Compass* 1, 5: 519–538. https://doi.org/10.1111/j.1749-818X.2007.00021.x

11. Sender Dovchin. 2011. Performing identity through language: The local practices of urban youth populations in post-socialist Mongolia. *Inner Asia*: 315–333.

12. Steven Feld and Aaron Fox. 1994. Music and Language. *Annual Review of Anthropology* 23: 25–53. https://doi.org/10.2307/2156005

13. Aaron A. Fox. 2004. *Real Country: Music and Language in Working-Class Culture*. Duke University Press.

14. Sharon Goldwater, Dan Jurafsky, and Christopher D. Manning. 2010. Which words are hard to recognize? Prosodic, lexical, and disfluency factors that increase speech recognition error rates. *Speech Communication* 52, 3: 181–200. https://doi.org/10.1016/j.specom.2009.10.001

15. Masataka Goto, Katunobu Itou, Koji Kitayama, and Tetsunori Kobayashi. 2004. Speech-recognition interfaces for music information retrieval: 'speech completion' and 'speech spotter. In *In Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004*, 403–408.

16. Emilio Granell and Carlos-D. Martínez-Hinarejos. 2016. A Multimodal Crowdsourcing Framework for Transcribing Historical Handwritten Documents. In *Proceedings of the 2016 ACM Symposium on Document Engineering* (DocEng '16), 157–163. https://doi.org/10.1145/2960811.2960815

17. Ido Guy. 2016. Searching by Talking: Analysis of Voice Queries on Mobile Web Search. 35–44. https://doi.org/10.1145/2911451.2911525

18. M. W. Hahn and R. A. Bentley. 2003. Drift as a mechanism for cultural change: an example from baby names. *Proceedings of the Royal Society B: Biological*

Sciences 270, Suppl_1: S120–S123. https://doi.org/10.1098/rsbl.2003.0045

19. Timothy J. Hazan and Issam Bazzi. 2001. A comparison and combination of methods for OOV word detection and word confidence scoring. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, 397–400.

20. Caroline Henton. 2005. Bitter pills to swallow. ASR and TTS have drug problems. *International Journal of Speech Technology* 8, 3: 247–257.

21. Anna Jørgensen, Dirk Hovy, and Anders Søgaard. 2015. Challenges of studying and processing dialects in social media. In *Proceedings of the Workshop on Noisy User-generated Text*, 9–18. Retrieved September 7, 2017 from http://www.aclweb.org/anthology/W15-4302

22. Sarvnaz Karimi, Falk Scholer, and Andrew Turpin. 2011. Machine transliteration survey. *ACM Computing Surveys* 43, 3: 1–46. https://doi.org/10.1145/1922649.1922654

23. Daniel Kessler. Raising Our Common Voice For The Web. *Internet Citizen*. Retrieved September 18, 2017 from https://blog.mozilla.org/internetcitizen/2017/06/19/commonvoice

24. William J. Kountz. 1899. *Billy Baxter's Letters*. Duquesne Distributing Company.

25. Peter Ladefoged and Keith Johnson. 2014. *A Course in Phonetics*. Cengage Learning, Stamford, CT.

26. Antoine Laurent, Sylvain Meignier, and Paul Deléglise. 2014. Improving recognition of proper nouns in ASR through generating and filtering phonetic transcriptions. *Computer Speech & Language* 28, 4: 979–996. https://doi.org/10.1016/j.csl.2014.02.006

27. Steve Lawrence and C. Lee Giles. 2000. Accessibility of information on the web. *intelligence* 11, 1: 32–39.

28. Daniel Luzzati, Cyril Grouin, Ioana Vasilescu, Martine Adda-Decker, Eric Bilinski, Nathalie Camelin, Juliette Kahn, Carole Lailler, Lori Lamel, and Sophie Rosset. 2014. *Human Annotation of ASR Error Regions: is "gravity" a Sharable Concept for Human Annotators?* European Language Resources Association (ELRA). Retrieved from https://hal.archives-ouvertes.fr/hal-01134802

29. Hao Ma, Raman Chandrasekar, Chris Quirk, and Abhishek Gupta. 2009. Page Hunt: Improving Search Engines Using Human Computation Games. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR '09), 746–747. https://doi.org/10.1145/1571941.1572108

30. Rishabh Mehrotra, Ashton Anderson, Fernando Diaz, Amit Sharma, Hanna Wallach, and Emine Yilmaz. 2017. Auditing Search Engines for Differential Satisfaction Across Demographics. *arXiv:1705.10689 [cs]*. https://doi.org/10.1145/3041021.3054197

31. Andrew Moody and Yuko Matsumoto. 2003. "Don't Touch My Moustache": Language Blending and Code Ambiguation by Two J-Pop Artists. *Asian Englishes* 6, 1: 4–33. https://doi.org/10.1080/13488678.2003.10801106

32. Kenneth Olmstead. Voice assistants used by 46% of Americans, mostly on smartphones | Pew Research Center. Retrieved January 5, 2018 from http://www.pewresearch.org/fact-tank/2017/12/12/nearly-half-of-americans-use-digital-voice-assistants-mostly-on-their-smartphones/

33. Aasish Pappu, Teruhisa Misu, and Rakesh Gupta. 2016. Investigating Critical Speech Recognition Errors in Spoken Short Messages. In *Situated Dialog in Speech-Based Human-Computer Interaction*. Springer, Cham, 71–82. https://doi.org/10.1007/978-3-319-21834-2_7

34. Carolina Parada, Mark Dredze, Denis Filimonov, and Frederick Jelinek. 2010. Contextual Information Improves OOV Detection in Speech. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (HLT '10), 216–224. Retrieved January 5, 2018 from http://dl.acm.org/citation.cfm?id=1857999.1858024

35. Carolina Parada, Abhinav Sethy, Mark Dredze, and Frederick Jelinek. 2010. A spoken term detection framework for recovering out-of-vocabulary words using the web. In *Eleventh Annual Conference of the International Speech Communication Association*.

36. Manuel Perea, Jon Andoni Duñabeitia, and Manuel Carreiras. 2008. R34D1NG W0RD5 W1TH NUMB3R5. *Journal of Experimental Psychology: Human Perception and Performance* 34, 1: 237–241. https://doi.org/10.1037/0096-1523.34.1.237

37. Antoine Raux. 2008. Flexible turn-taking for spoken dialog systems. *Language Technologies Institute, CMU Dec* 12. Retrieved September 16, 2017 from http://www.cs.cmu.edu/afs/cs/Web/People/antoine/thesis_antoine.pdf

38. Attapol T. Rutherford, Fuchun Peng, and Françoise Beaufays. 2014. Pronunciation learning for named-entities through crowd-sourcing. In *Fifteenth Annual Conference of the International Speech Communication Association*. Retrieved August 11, 2017 from https://mazsola.iit.uni-miskolc.hu/~czap/letoltes/IS14/IS2014/PDF/AUTHOR/IS140902.PDF

39. Mark Savage. 2012. BBC Sound of 2013: Chvrches. *BBC News*. Retrieved September 18, 2017 from http://www.bbc.com/news/entertainment-arts-20780332

40. Kurt Schwitters. 1923. w88888888. *Drachtster Courant*.

41. Rachael Tatman. 2017. Gender and Dialect Bias in YouTube's Automatic Captions. *EACL 2017*: 53.

42. Erik R. Thomas. 2007. Phonological and Phonetic Characteristics of African American Vernacular English. *Language and Linguistics Compass* 1, 5: 450–475. https://doi.org/10.1111/j.1749-818X.2007.00029.x

43. Crispin Thurlow and Alex Brown. 2003. Generation Txt? The sociolinguistics of young people's text-messaging. *Discourse analysis online* 1, 1: 30.

44. Angela Watercutter. Crazy Characters Help Indie Bands Outsmart Google. *WIRED*. Retrieved January 5, 2018 from https://www.wired.com/2011/01/pl_music_ungoogle/

45. Ewald van der Westhuizen and Thomas Niesler. 2016. Automatic Speech Recognition of English-isiZulu Code-switched Speech from South African Soap Operas. *Procedia Computer Science* 81: 121–127. https://doi.org/10.1016/j.procs.2016.04.039

46. W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig. 2016. Achieving Human Parity in Conversational Speech Recognition. *arXiv:1610.05256 [cs]*. Retrieved from http://arxiv.org/abs/1610.05256

47. Emre Yılmaz, Henk van den Heuvel, and David van Leeuwen. 2016. Investigating Bilingual Deep Neural Networks for Automatic Recognition of Code-switching Frisian Speech. *Procedia Computer Science* 81: 159–166. https://doi.org/10.1016/j.procs.2016.04.044

48. State of the U.S. Online Retail Economy in Q1 2017. *comScore, Inc*. Retrieved September 18, 2017 from http://www.comscore.com/Insights/Presentations-and-Whitepapers/2017/State-of-the-US-Online-Retail-Economy-in-Q1-2017

49. What Does "Lit" Mean? | Merriam-Webster. Retrieved September 18, 2017 from https://www.merriam-webster.com/words-at-play/lit-meaning-origin