Generating Feature Vectors from Phonetic Transcriptions in Cross-Linguistic Data Formats

Arne Rubehn[1], **Jessica Nieder**[1], **Robert Forkel**[2] and **Johann-Mattis List**[1]

[1]Chair for Multilingual Computational Linguistics, University of Passau
Passau, Germany

[2]DLCE, MPI-EVA
Leipzig, Germany

Abstract

When comparing speech sounds across languages, scholars often make use of feature representations of individual sounds in order to determine fine-grained sound similarities. Although binary feature systems for large numbers of speech sounds have been proposed, large-scale computational applications often face the challenges that the proposed feature systems – even if they list features for several thousand sounds – only cover a smaller part of the numerous speech sounds reflected in actual cross-linguistic data. In order to address the problem of missing data for attested speech sounds, we propose a new approach that can create binary feature vectors dynamically for all sounds that can be represented in the the standardized version of the International Phonetic Alphabet proposed by the Cross-Linguistic Transcription Systems (CLTS) reference catalog. Since CLTS is actively used in large data collections, covering more than 2,000 distinct language varieties, our procedure for the generation of binary feature vectors provides immediate access to a very large collection of multilingual wordlists. Testing our feature system in different ways on different datasets proves that the system is not only useful to provide a straightforward means to compare the similarity of speech sounds, but also illustrates its potential to be used in future cross-linguistic machine learning applications.

1 Introduction

The past two decades have seen a drastic increase in standardized datasets in historical linguistics and linguistic typology which are available in both human- and machine-readable form (Dellert et al., 2020; Skirgård et al., 2023; Wichmann et al., 2013). With Lexibank (`https://lexibank.clld.org`, List et al. 2022), a large collection of comparative wordlists has been compiled in which word forms from various independently published datasets are standardized along three dimensions, including (1) the languages in which they occur, (2) the concepts which they express, and (3) the sounds that constitute them. Lexibank is a result of the more general Cross-Linguistic Data Formats initiative (CLDF, `https://cldf.clld.org`, Forkel et al. 2018), which aims to unify several kinds of cross-linguistic data (wordlists, typological datasets, interlinearglossed texts) by proposing an exchange format along with guidelines and recommendations for standardization.

Sounds in Lexibank are represented in a unified transcription system, proposed as part of the Cross-Linguistic Transcription Systems reference catalogue (CLTS, `https://clts.clld.org`, List et al. 2024) that can be considered a large standardized subset of the International Phonetic Alphabet (IPA, IPA 1999). The CLTS system tries to handle as much of the variation observed in phonetic transcriptions as possible, using a dynamic method that parses phonetic transcriptions in a given transcription system and derives features from individual symbol combinations (Anderson et al., 2018). The feature system underlying these transcriptions has been designed in a pragmatic way that would allow to capture as much of the graphical variation in using the IPA (and other transcription systems) as possible (Anderson et al., 2023). As a result, the system is powerful in parsing phonetic transcriptions – specifically those represented in IPA – but it is not particularly useful to compare speech sounds with respect to their *similarity* (be it acoustic or articulatory or a combination of both).

Thus, while the CLTS system does its job in helping to standardize phonetic transcriptions in an unprecedented way, as witnessed by the Lexibank collection (and numerous additional CLDF wordlists that have been published in the past years), it falls short in providing a reliable means to compare individual sounds for their similarity.

In this study, we present a very straightforward approach to convert the CLTS feature system to a

Proceedings of the Society for Computation in Linguistics (SCiL) 2024, pages 205-216.
Irvine, California, June 27-29, 2024

vector representation. This approach takes CLTS feature bundles as input and converts them into binary feature vectors that can be used for various downstream tasks in computational phonology, computational historical linguistics, and computational linguistic typology.

2 Background

Modeling speech sounds as bundles of distinctive features can be seen as the most typical and straightforward way in phonology and comparative linguistics to compare the similarity of speech sounds. It is therefore not surprising, that phonological features play a crucial role in a number of different approaches, ranging from historical language comparison (Kondrak, 2000) over dialectology (Nerbonne and Heeringa, 1997; Hoppenbrouwers and Hoppenbrouwers, 2001) and phonological rule induction (Gildea and Jurafsky, 1996) to child language acquisition (Somers, 1998). Representing sounds with the help of features can also enhance the performance of automatic speech recognition (Metze, 2007) and transliteration (Tao et al., 2006; Yoon et al., 2007), as well as automatic phonetic transcriptions from text and named entity recognition (Mortensen et al., 2016). Related studies have additionally demonstrated that meaningful phone(me) embeddings can be learned from distributional properties (Silfverberg et al., 2018; Sofroniev and Çöltekin, 2018).

Beyond these mostly implicit uses of phonological features, there exist several frameworks with the explicit purpose of modeling the interactions between sounds and their respective features. Computational tools for analyzing phoneme inventories in terms of phonological features and natural classes are available in the form of web applications (Steel and Jurgec, 2017) or downloadable programs (van Vugt and Hayes, 2021). However, these tools by default cover rather small inventories of fairly common sounds and are often even only designed for individual languages.

There is a small number of datasets that map a large number of sounds to a feature representation, aiming to cover a substantially large amount of speech sounds in order to be applicable in cross-linguistic studies. PanPhon (Mortensen et al., 2016) defines feature representations for approximately 5,000 sounds, the similar but smaller framework DistFeat (Tresoldi, 2020) spans roughly 500 sounds. Phonotacticon (Joo and Hsu, 2023), a typological resource for phonotactics, extends PanPhon to around 20,000 distinct speech sounds. PHOIBLE (Moran and McCloy, 2019), a database covering various phoneme inventories, is equipped with feature definitions as well, covering all 3,000 distinct sounds attested in the data in its latest version. Finally, the Python package `ipasymbols` (Hamster, 2022) is designed to query IPA symbols by their articulatory properties, but is not equipped with phonological features and is currently (v.0.1.0) limited to only 179 sounds.

While all feature collections are much larger than the earlier feature collections that phonologists proposed for individual languages, reflecting the trend towards cross-linguistic approaches that allow for a comparison across multiple languages, all feature collections are *fixed sets of sounds*, lacking a dynamic component. This limits their potential when applying them to newly compiled datasets, since whenever a sound in a given dataset is not attested in the feature systems, users would have to add it or to label it as missing data.

While this may seem to reflect a minor problem, it has grown into a major obstacle for many concrete applications in computational comparative linguistics, since practical experience in working with concrete language data clearly shows that meeting unobserved sounds when turning to new datasets is rather the rule than the exception (see the observation in Moran 2012, that the overall number of distinct speech sounds seems to increase almost constantly, albeit slowly, when new data are added to the sample). One way to avoid the problem of observing missing sounds is to arbitrarily extend mappings from IPA transcriptions of speech sounds to feature mappings in a systematic way, as exemplified by the extended system proposed by the Phonotacticon, with 20,000 distinct sounds, of which only a couple of hundred sounds occur in the final database.

An alternative, more robust approach, specifically important for data standardized in CLDF, would take the pragmatically oriented non-binary features provided by the CLTS system as a starting point and convert them to a binary vector representation.

3 Materials and Methods

3.1 Materials

The starting point of our approach is the CLTS reference catalogue, which links feature descriptions of speech sounds in the style of the IPA to different transcription systems and datasets. While the CLTS website presents a list of about 8,000 distinct speech sounds that are linked to various datasets, including PHOIBLE and PanPhon, the system that generates the website is dynamic, with only a couple of hundred base sounds being defined explicitly. The rest of the sounds is generated from sound transcriptions mainly provided in the International Phonetic Alphabet. The dynamic system underlying the CLTS reference catalogue can be accessed with the help of a Python API (`https://pypi.org/projects/pyclts`, List et al. 2020, see Anderson et al. 2018 for the details regarding the algorithm used by the API). As a result, IPA strings that are not directly represented in the system can be processed, as long as they conform to IPA standards (broadly defined by CLTS). The CLTS system parses sounds in two ways, taking a phonetic transcription (typically provided in IPA) as starting point, or starting from the typical name of a speech sound, as they are also defined by the IPA. For example, [p] would be described as the 'voiceless bilabial stop consonant', yielding the descriptive feature set ('voiceless', 'bilabial', 'stop', 'consonant'). For the conversion of sound transcriptions accepted by CLTS to binary feature vectors, we use the feature bundle representation rather than the phonetic transcription as our starting point.

In order to evaluate the usefulness of the binary feature vectors derived from CLTS, we use the Lexibank database, since it provides a large collection of wordlists that conform to the standard defined by CLTS.

As of version 1.0 (List et al., 2023) Lexibank is available in an aggregated form in which all datasets that are sufficiently standardized – with all sounds being interpretable by the dynamic CLTS system – are assembled in a single CLDF dataset that can be parsed and processed in various ways, including SQLite (see List and Shcherbakova 2023) or Python (using the CL Toolkit package, see List and Forkel 2021, `https://pypi.org/project/cltoolkit`). In this form, Lexibank covers wordlists of at least 80 distinct words for about 2,000 distinct language varieties.

3.2 Methods

3.2.1 Feature System

We define a classical feature space of 39 binary phonological features that can be present (1) or absent (-1), or non-applicable (0). Strictly speaking, we therefore employ a notion of ternary, rather than binary features, since there are three instead of two possible values. However, this is merely an explicit formalization of the way that binary features are usually treated in phonology: Not all features can apply to all kinds of sounds. It is therefore necessary to distinguish absent from non-applicable features by assigning them different numerical values. To illustrate this, consider the feature `[±strident]` which only applies to fricatives and affricates (Zsiga, 2013) – it is worthwhile to distinguish non-strident fricatives (which *could* be strident) from other sounds that do not have this feature at all. This notion of applicability is frequently found in the literature, and formally makes these systems ternary rather than binary. Keeping this in mind, we will still refer to this feature system as binary, given that it is the commonly used term to describe this kind of feature systems.

The majority of the features we define constitutes a fairly well established standard inventory, where we strictly follow the definitions by Zsiga (2013). Nevertheless, to be able to cover a comprehensive range of sounds, some additions to the feature inventory were required. We incorporate three additional features `[velaric]`, `[hitone]`, and `[hireg]` from Mortensen et al. (2016) to handle clicks and tones. Clicks are assigned `[+velaric]` on top of their other features that are derived from their analogous pulmonic stops. The tonal features `[hireg]` and `[hitone]` refer to the broader register, and the more narrow tone quality within the register – both the high tone [^{5}] and the mid-high tone [^{4}] belong to the high register and are therefore `[+hireg]`, and within that register, [^{5}] is the higher tone and is therefore `[+hitone]` (whereas [^{4}] is `[-hitone]` analogously). Since we do not want to assign the feature value of 0 (non-applicable) to tonal features in tones, however, these two features only yield 4 possible combinations, insufficient for encoding the canonical 5 tones. We therefore introduce the supplementary feature `[loreg]` as a logical counterpart to `[hireg]`, covering the low and the mid-low tones.

Furthermore, we employ three additional features to represent complex tones. Tonal features in CLTS are based on a rather schematic representation of Chao's numeral coding of complex tones in Chinese dialects (Chao, 1930[2006]). Thus, the tone [^{214}] is labeled as `"contour from-mid-low via-low to-mid-high tone"` in the CLTS name space, with the feature value `"from-mid-low"` (from the feature `[start]`) – representing the number ^{2}, `"via-low"` (from the feature `[middle]`) representing ^{1}, and `"to-mid-high"` (from the feature `[end]`) representing ^{4} directly, while the feature value `"contour"` (from the feature `[contour]` adds additional information that tells us that we are dealing with a contour tone. In the same way, CLTS assigns the values `"rising"` and `"falling"` to tones like [^{15}] and [^{51}] respectively. In the vector representation, the contour of a tone is directly translated to the features `[contour, rising, falling]` based on their respective CLTS feature. Contour tones with both rising and falling parts additionally receive `[+rising]` or `[+falling]`, indicating the interval between the initial and the middle segment. All complex tones inherit the features `[hireg, hitone, loreg]` from their initial segment. The example tone [^{214}] is therefore represented as `[-hireg, +hitone, +loreg, +contour, -rising, +falling]`.

Finally, seven more features are defined for representing diphthongs, namely `[backshift, frontshift, opening, closing, centering, longdistance, secondrounded]`. These features are used to model the diphthong's trajectory across the vowel space (for a more detailed description see Rubehn, 2022, 41-43). Additionally, each diphthong is assigned the simple vowel features of its initial segment.

Due to its flexibility, the present system is highly customizable and can be used to generate different feature systems as well. Users can easily define their own feature inventories and mappings, according to their individual needs. The workflow presented in the following section is not dependent on the specific feature system and definitions that we suggest here.

3.2.2 Workflow

Our system generates binary feature vectors for any sound based on its feature set assigned by CLTS. Again, consider the example [p]: The method does not depart from the string `"p"`, but from the feature set ('voiceless', 'bilabial', 'stop', 'consonant') that can easily be obtained from CLTS. The general workflow for generating binary feature vectors is outlined in Figure 1.

Underlying our system is a simple dictionary structure which maps triplets of CLTS feature values, CLTS feature domains, and binary feature representations onto each other. The feature value

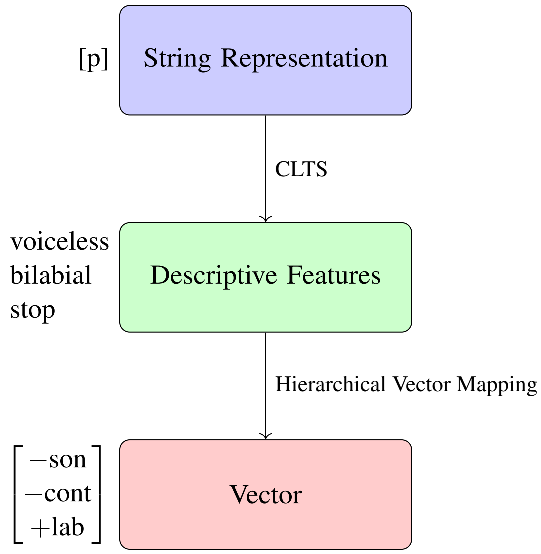

Figure 1: Workflow of vector creation.

'stop' for example would be linked to the domain 'manner' (of articulation in consonants) and to the binary feature representation `[-son,-cont]`.

As a starting point, a zero vector (with the value 0 at every position) of the size of the defined feature inventory is instantiated, with every position of the vector corresponding to exactly one binary feature. This vector then is modified by subsequently processing the features in the CLTS feature set, with the corresponding binary features overwriting the current value in the vector.

A core principle for the successful modification of feature vectors is that we sort the CLTS features by a *hierarchy* of concreteness that determines the order in which those features are processed. This hierarchy states that the least specific features get processed first, and the most specific ones get processed last. In the concrete example of [p], that means that 'consonant', being the least specific feature, is processed first.

This notion of hierarchy is necessary for handling conflicting feature mappings, since we deliberately allow for values in the vector to be overwritten by features that are processed later. To exemplify this, consider the 'devoiced voiced labiodental fricative' [v̥]: The descriptor 'voiced' maps to `[+voice]`, whereas 'devoiced' naturally corresponds to `[-voice]`. However, since diacritics modify the base sound, they should take precedence over it, and the correct feature that should be assigned is `[-voice]`. This is ensured by processing the features according to the hierarchy, which states that the modification 'devoiced' is more concrete and should therefore be applied *after* the regular phonation feature 'voiced', and can thus overwrite the previously assigned `[+voice]` with `[-voice]`.

This notion of hierarchy also conveniently allows for the usage of *default values* that can define which features apply at which representation level. This is important since we distinguish between non-applicable (0) and absent (-1) features, as discussed in Section 3.2.1. We can therefore define which set of binary features always applies to a certain group of sounds by assigning a default value to ensure that applicable features have a non-zero value. For example, the feature `[±lab(ial)]` must be defined for all consonants, which is assured by mapping the CLTS feature 'consonant' to `[-lab]`. If the consonant is actually labial, this feature will be overwritten with `[+lab]`, since the place of articulation is always applied after the sound type. So instead of exhaustively defining `[-lab]` for every non-labial place of articulation, we can just define it as a default value for the CLTS feature 'consonant' instead. This corresponds to the reading that every consonant is `[-lab]` (by default), unless specified otherwise.

The majority of sounds can be handled by this straightforward workflow of hierarchically mapping CLTS features to their binary feature representations. However, there are a few more complex cases that require an extra processing step. For example, the glottal stop [ʔ] has the binary feature `[+cg]` ('constricted glottis') – however, this feature neither corresponds to 'glottal', nor to 'stop'. It is therefore the combination of both 'glottal' and 'stop' that triggers `[+cg]`. The system therefore uses a second dictionary that allows for the definition of joint feature mappings, where a binary feature definition is conditioned by a certain *combination* of CLTS features.

Complex sounds that can alternatively be analyzed as two segments – diphthongs and consonant clusters – pose a similar challenge. For these cases, CLTS provides the means of analyzing its individual constituents: The consonant cluster [kp] can be split into [k] and [p]. The system uses this function to generate separate feature vectors for the two individual sounds, which then are combined by assigning the union of positive features to the joint vector. The resulting feature vector for [kp] therefore contains all positive features that are attributed to either [k] or [p].

In a similar fashion, feature vectors for diphthongs are based on their initial segments. For example, [aɪ] inherits its monophthong vowel features from the feature definitions for [a]. The additional diphthong features, that indicate the trajectory of the diphthong, are assigned based on joint feature definitions: The combination of the CLTS features ('from_open', 'to_near-close') maps (among others) to the binary feature `[+closing]`.

3.3 Implementation

The approach is implemented in the form of a Python package (`soundvectors`) that takes as input the canonical names consisting of feature values that CLTS generates dynamically for speech sounds in standard IPA transcription and can be applied in combination with CLTS and the `pyclts` package, as well as with the `linse` package that offers non-generative access to a larger selection of speech sounds covered by CLTS (`https://pypi.org/project/linse`, List and Forkel 2024), but also independently of these packages, as long as the feature names follow the CLTS standards. The package along with the data on which it was tested is available from the supplementary material accompanying this study.

4 Evaluation

We test the usefulness of our proposed system by (1) investigating the vector similarities for common sounds by calculating cosine similarities and visualizing them with heatmaps, (2) employing techniques for dimensionality reduction to visualize the relationships between sounds, (3) mapping the CLTS sound inventory to binary vectors and analyzing the resulting equivalence classes, and (4) investigating the power of the system to distinguish speech sounds observed in phonetically transcribed wordlists.

4.1 Vector Similarities

To test how well our system analyzes a representative sample of common sounds, we take 25 most common consonants and the 20 most common vowels from Phoible 2.0 (Moran and McCloy, 2019) and use heatmaps to visualize the cosine similarities similarities of their respective feature vectors (Figures 2 and 3). The heatmaps were generated with the Python library Seaborn (Waskom, 2021), with lighter colours representing higher similarities and darker colours representing lower similarity scores. The sounds are ordered by their primary

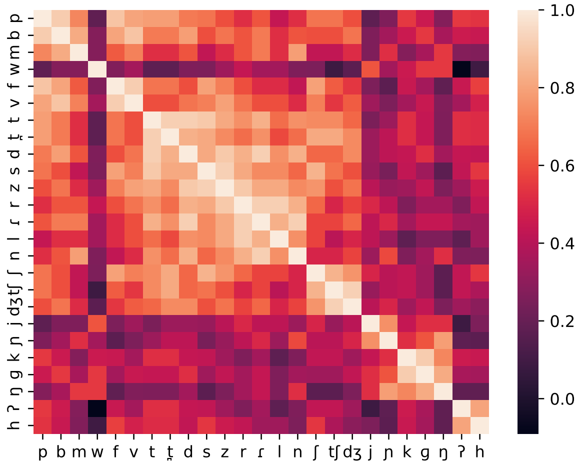

Figure 2: Cosine similarities between consonant vectors generated with our model.

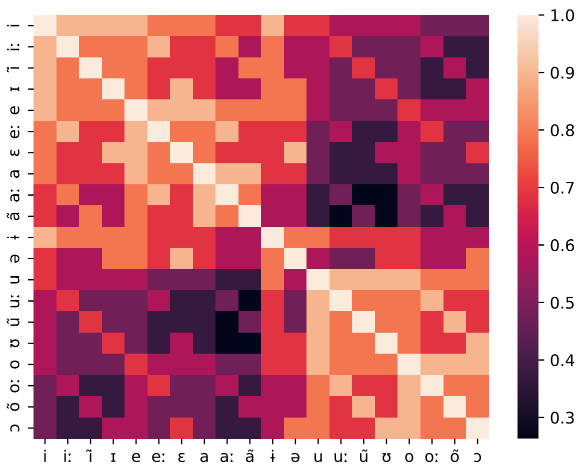

Figure 3: Cosine similarities between vowel vectors generated with our model.

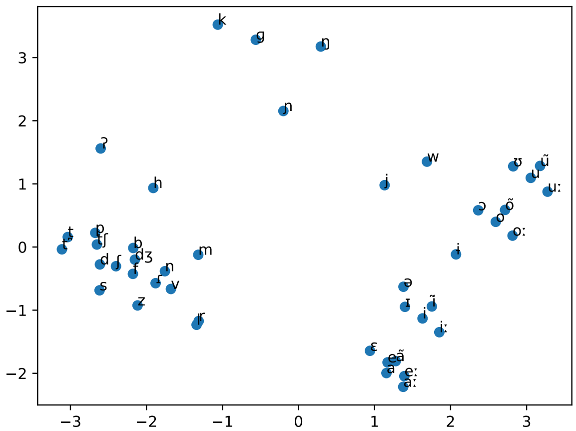

Figure 4: Two-dimensional reduction of feature vectors using PCA.

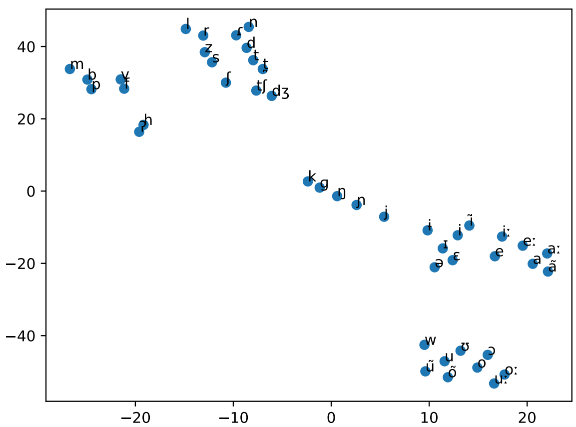

Figure 5: Two-dimensional reduction of feature vectors using t-SNE.

place of articulation, starting from the front and moving to the back of the mouth. As can be observed in Figure 2, the manner of articulation and the phonation have a clear impact on the similarity of consonantal feature vectors: [p] is therefore much more similar to [k] than to [ŋ]. The glides [w] and [j] are strikingly dissimilar to the rest of the consonants, showing their well-known intermediate role in between consonants and vowels.

Figure 3 shows that the vowel space follows a strong division into two clusters which correspond to front and back vowels, with [a] being considered a front vowel according to the IPA nomenclature. This primary partition reflects the fact that typologically unmarked front vowels are unrounded, and back vowels are typically rounded. This naturally translates into a separate feature which drives front and back vowels further apart in terms of their vector similarity. Nonetheless, vowel pairs that share the same height such as [e] and [o] retain a fairly high degree of cosine similarity, indicating that our vectors are able to adequately reflect and apply vowel features to the used vowel data set.

4.2 Dimensionality Reduction

We employed two techniques for dimensionality reduction to project the phonological vector space onto a two-dimensional plane, aiming to gain a comprehensive understanding of the inherent structure of our vectors. The employed techniques are *principal component analysis* (PCA; Figure 4), known for its ability to reveal global linear structures, and *t-distributed stochastic neighbor embedding* (t-SNE; Figure 5; van der Maaten and Hinton 2008), known for capturing local nonlinear patterns in the data. Both dimensionality reductions were computed in Python using the SciKit-Learn package (Pedregosa et al., 2011) and visualized with Matplotlib (Hunter, 2007). By employing PCA and t-SNE at the same time, we sought to ensure a robust and detailed exploration of our vectors, leveraging the complementary strengths of both of these dimensionality reduction techniques (Anowar et al., 2021).

Figure 4 and Figure 5 visualize the results after dimensionality reduction. Both PCA and t-SNE reveal a consistent narrative, grouping similar sounds together. First and foremost, there is a clear distinction between consonants and vowels, with semivowels positioned either between the two clusters in PCA or in close proximity to the corresponding vowel cluster in t-SNE. Once again, this reaffirms the ability of our binary vectors to adequately distinguish between the sounds in a phonologically informed way.

Focusing on the vowel clusters in Figure 4 and Figure 5, it becomes evident once again that vowels are primarily divided into (unrounded) front and (rounded) back vowels, aligning with the well-established phonological classification of vowel sounds. Shifting attention to the consonant clusters in both panels, we once more observe that they are primarily grouped by their place of articulation. In both Figures, the velars [k,g,ŋ] form a rather isolated cluster, loosely associated with the palatal nasal [ɲ]. Both techniques also tend to isolate the glottal sounds [h,ʔ], with t-SNE placing this sound pair much closer to each other. A distinct picture emerges for the remaining consonants, the coronals and labials: While PCA seems to bundle all of them together into a single large cluster, t-SNE forms two distinct clusters based on their place of articulation, however retaining a certain proximity between the two. The t-SNE plot also exhibits a compelling parallelogram symmetry among quadruplets of stops and fricatives in their voiced and voiceless versions: The alveolars [t,d,s,z] form one such parallelogram in the two-dimensional projection; and a similar pattern can be observed for labials [p,b,f,v].

The observed similarities depicted in Figure 2 and Figure 3 as well as the patterns in the vowel and consonant clusters depicted in Figure 4 and Figure 5 align with established phonological classifications, providing a visual representation that echoes the theoretical descriptions and classes in phonological theory to a considerable extent. Our observations confirm the potential utility of our phonological feature vectors in computational models, suggesting that the vectors capture meaningful distinctions and relationships inherent in the sounds of human languages.

4.3 Equivalence Classes

The current version of CLTS (v.2.1.0) provides a collection of 8,684 unique sounds that were observed in its source datasets. Employing our system, these 8,684 sounds map to 5,285 distinct feature vectors. The system is therefore capable of providing a unique representation for 60.9% of this large sound inventory, even though it contains a number of very narrow transcriptions, or aspects that we deliberately chose not to represent in the feature space, such as suprasegmental properties being attributed to a segment (for example, putting tones on vowels).

The largest two equivalence classes contain 18 segments respectively, which are all mid and open-mid vowels. The first class therefore contains sounds that are based on [ə] and [ɜ], including among others [ə́, ə˞, ɜ̟]. All of these modifications are deliberately disregarded by our system: Tones are suprasegmental features that should not be represented as part of a segment, rhotics lack reliable phonetic correlates (Chabot, 2019), and specifying the relative tongue position is an overly narrow transcription style that does not carry distinctive information.

This illustrates the principle of economy, in that we only define as many features as strictly necessary to keep the individual features meaningful and avoid feature inflation. The distinctions that are lost by employing this procedure are extremely narrow in domain, and phonetically not meaningful, as we will argue in the following section.

4.4 Distinctiveness

We investigate the discriminative potential of our system by applying it to all sound inventories observed in phonetic transcriptions of lexical data in Lexibank (List et al., 2022, 2023). The aggregated dataset combines numerous datasets into one unified dataset, spanning over 2,905 language varieties in total. In Table 1, we report the metrics of how well the sound inventories of the languages in Lexibank 1.0 can be described by our system. We report the number of confused sounds per language, that is how many sounds in an inventory share their feature representation with another sound present in this inventory. Formally, this is the difference between the size of the sound inventory, and the number of unique feature representations corresponding

n confused sounds	n varieties	Portion
0	2,376	0.818
≤1	2,567	0.884
≤2	2,648	0.912
≤3	2,689	0.926
≤4	2,841	0.978

Table 1: Number of language varieties in Lexibank 1.0 with at most n confused sounds.

to the inventory.

The sound inventories of 2,376 varieties, amounting to 81.8% of all varieties in the dataset, can be represented with full distinctiveness by our system, meaning that every sound is mapped to a unique feature vector. With 2,841 (97.8%) varieties, the grand share of the dataset's sound inventories can be represented with a maximum of four overlapping feature representations. These overlaps can usually be explained by narrow transcriptions, where the exact realization of the sound is predictable from the context, or can even hint at transcription errors or inconsistencies in the source data.

We investigate such sets of overlapping sounds in their context using concordance lines. This technique is frequently used in corpus linguistics for visualizing in which contexts a certain word appears. Usually, concordance lines are generated by aligning the highlighted target word to the center of a table, and placing the contexts to its left or right respectively (Hunston, 2022, 47).

annoy	e	ɱ	f a ð a ɾ
be annoyed	e	ɱ	f a ð a ɾ s e
be ill	e s t a ɾ + e	ɱ	f e ɾ m o
fall ill	k a e ɾ + e	ɱ	f e ɾ m o
illness	e	ɱ	f e ɾ m e ð a ð
sick, ill	e	ɱ	f e ɾ m o
December	d i θ j e	m	b ɾ e
March		m	a ɾ θ o
May		m	a ʝ o
November	n o β j e	m	b ɾ e
September	s e β t j e	m	b ɾ e
Sunday	d o	m	i ŋ g o

Figure 6: Concordance line for Spanish transcriptions featuring [ɱ] or [m].

Figure 6 exemplifies the usefulness of concordance lines to analyze the contexts in which sounds occur. Here, we investigate the instances of the bilabial and labio-dental nasal consonants [m] and [ɱ] in the Spanish data from the NorthEuraLex database (Dellert et al., 2020). Both sounds are represented by identical feature vector, but the data makes a distinction between them. Analyzing the relevant forms, however, shows that the presence of [ɱ] can be clearly predicted from the context, since it can only occur before labio-dental obstruents. This suggests that there is no actual distinction between these sounds, since the different surface forms can be explained by a fully predictable assimilation process. Similar cases of complementary distribution can be observed within the same dataset: In Nanai, [i̟] is only found preceding bilabial and alveolar consonants, while [i] occurs elsewhere; Estonian [k] is transcribed as [k̥] if preceded by [ŋ]; and Korean voiceless stops are unreleased in word-final position, leading to pairs of distinct transcriptions (e.g. [t] - [t˺]) with a distribution that is completely predictable by context.

5 Discussion

In this study, we introduced a new approach to turn the features for all sounds covered by the CLTS reference catalogue into numerical feature vectors. Given that CLTS not only underlies the Lexibank repository, which offers phonetically transcribed, standardized wordlists for more than 2,000 language varieties, but is also used in many additional applicatations that make use of the standards proposed by the CLDF initiative, this means that the binary feature vectors we propose are directly available for a very large number of language varieties.

To assess the effectiveness of our approach, we conducted a detailed analysis using cosine similarity and dimensionality reduction techniques. The resulting similarity patterns, evident in both PCA and t-SNE plots, align with established phonological classifications. Notably, the model distinguishes between vowels and consonants and groups similar sounds based on their place of articulation. Furthermore, we successfully mapped a substantial inventory of sounds from CLTS to their respective vectors, covering more than half of this extensive sound dataset with unique representations. Finally, we evaluated the distinctiveness of our vector representations by discerning speech sounds from lexical data in the Lexibank repository. Our system accurately represented a significant portion of the data, ensuring full distinctiveness by uniquely mapping each sound to a feature vector.

In conclusion, our approach not only provides a practical solution to address general limitations of the pragmatic feature system underlying the CLTS reference catalogue but also offers a flexible approach for representing phonological features in computational linguistics. By converting CLTS feature bundles into binary feature vectors, the approach enables researchers to integrate phonological insights into various computational tasks, ranging from phonology and historical linguistics to linguistic typology.

For the field of cognitive language modeling, our feature system offers an enhanced, more precise phonological representation. Nieder and List (2024) utilized historical sound class representations in their language processing model (Linear Discriminative Learning, see Nieder and List, 2024) to explore mutual intelligibility among Germanic languages. Expanding such models with phonological vector representations instead, may offer new insights into how speech sounds influence meaning and vice versa, thereby guiding language processing and language learning.

For historical language comparison, feature representations can be used to dynamically extend fixed-size scoring matrices in computational tasks such as phonetic alignment (Kondrak, 2000; List, 2012) or phonological reconstruction (Bouchard-Côté et al., 2013; Jäger, 2019; Meloni et al., 2021). While state-of-the-art approaches to phonetic alignment typically deal with the problem of unseen sounds by resorting to sound class representations that represent sounds in phonetic transcriptions in small classes of similar sounds ranging from 10 to about 40 distinct sound classes in total (see List 2014 for details on sound class systems), feature vectors would offer a much more fine-grained representation of similarities and differences between individual sounds whose impacts on alignment quality have not been fully tested so far (an exception is the feature-based system by Kilani 2020, which requires, however, sound-feature mappings to be set up manually). For the still unsolved task of unsupervised phonological reconstruction (List, 2023), a common problem of those approaches that have been proposed so far is that they cannot propose sounds in ancestral languages that have not been attested in the descendant languages. Here, feature vectors might propose a way of handling the unknown, since the vector representation might well propose feature combinations for ancestral sounds that are not observed in individual languages, thus creating unseen sounds from attested sounds. But further tests would be needed to explore the potential of feature vectors in phonological reconstruction.

In summary, we hope that feature vectors, as they have been introduced here, will prove useful in advancing computational approaches in linguistics and integrating linguistic insights into machine learning approaches.

Supplementary Materials

Data and code of this study are curated on GitHub (`https://github.com/cldf-clts/soundvectors`), the `soundvectors` package is also available via the Python package repository PyPi (`https://pypi.org/project/soundvectors`, Version 1.0).

Limitations

As discussed in Sections 4.3 and 4.4, we want our system to distinguish sounds that sound differently and avoid lumping them together. Our quantitative and qualitative analyses show that the system seems to be capable of maintaining a high degree of distinctiveness, however, it is not guaranteed that all phonemic contrasts in the world's languages are represented truthfully. While our phonological feature vectors are a good approximation to spoken language, we want to point out that they cannot perfectly reflect phonetic similarity – some features are intuitively more meaningful than others, which is not explicitly represented in the vector space; and by extension, similar sounds might differ in a "disproportionately large number of features" (Kondrak, 2000). Heeringa (2004) shows that employing binary features directly as a cost function is not superior to plain edit distance to measure phonetic similarity in dialectal data.

These findings do not undermine the potential of feature systems in analyzing sounds, but rather show that feature vectors should be processed in some way and not be taken at face value. In fact, a number of studies have shown the usefulness of phonological feature vectors for processing sounds in machine learning approaches (e.g. Staib et al. 2020; Lux and Vu 2022). This emphasizes the need for a robust system that reliably generates feature vectors for all IPA segments.

Acknowledgements

The authors would like to thank Johannes Dellert for fruitful discussions on the choice of the feature inventory and several feature mappings, as well as two anonymous reviewers for their helpful comments and suggestions. This project was supported by the ERC Consolidator Grant ProduSemy (PI Johann-Mattis List, Grant No. 101044282, see `https://doi.org/10.3030/101044282`). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency (nor any other funding agencies involved). Neither the European Union nor the granting authority can be held responsible for them.

References

Cormac Anderson, Tiago Tresoldi, Thiago Chacon, Anne-Maria Fehn, Mary Walworth, Robert Forkel, and Johann-Mattis List. 2018. A cross-linguistic database of phonetic transcription systems. *Yearbook of the Poznań Linguistic Meeting*, 4(1):21–53.

Cormac Anderson, Tiago Tresoldi, Simon J. Greenhill, Robert Forkel, Russell D. Gray, and Johann-Mattis List. 2023. Variation in phoneme inventories: quantifying the problem and improving comparability. *Journal of Language Evolution*, 0(0):1–20.

Farzana Anowar, Samira Sadaoui, and Bassant Selim. 2021. Conceptual and empirical comparison of dimensionality reduction algorithms (PCA, KPCA, LDA, MDS, SVD, LLE, ISOMAP, LE, ICA, t-SNE). *Computer Science Review*, 40:100378.

Alexandre Bouchard-Côté, David Hall, Thomas L. Griffiths, and Dan Klein. 2013. Automated reconstruction of ancient languages using probabilistic models of sound change. *Proceedings of the National Academy of Sciences of the United States of America*, 110(11):4224–4229.

Alex Chabot. 2019. What's wrong with being a rhotic? *Glossa: a journal of general linguistics*, 4(1).

Yuenren Chao. 1930[2006]. A system of 'tone letters'. In Z.-j. Wu and X.-n Zhao, editors, *Linguistic Essays by Yuenren Chao*, pages 98–102. Shāngwù, Běijīng.

Johannes Dellert, Thora Daneyko, Alla Münch, Alina Ladygina, Armin Buch, Natalie Clarius, Ilja Grigorjew, Mohamed Balabel, Hizniye Isabella Boga, Zalina Baysarova, et al. 2020. NorthEuraLex: A wide-coverage lexical database of Northern Eurasia. *Language resources and evaluation*, 54(1):273–301.

Robert Forkel, Johann-Mattis List, Simon J. Greenhill, Christoph Rzymski, Sebastian Bank, Michael Cysouw, Harald Hammarström, Martin Haspelmath, Gereon A. Kaiping, and Russell D. Gray. 2018. Cross-Linguistic Data Formats, advancing data sharing and re-use in comparative linguistics. *Scientific data*, 5(1):1–10.

Daniel Gildea and Daniel Jurafsky. 1996. Learning bias and phonological-rule induction. *Computational Linguistics*, 22(4):497–530.

Ulf A. Hamster. 2022. Everybody likes short sentences - a data analysis for the text complexity DE challenge 2022. In *Proceedings of the GermEval 2022 Workshop on Text Complexity Assessment of German Text*, pages 10–14, Potsdam, Germany. Association for Computational Linguistics.

Wilbert Heeringa. 2004. *Measuring dialect pronunciation differences using Levenshtein distance*. Ph.D. thesis, Rijksuniversiteit Groningen.

Cor Hoppenbrouwers and Geer Hoppenbrouwers. 2001. *De indeling van de Nederlandse streektalen: Dialecten van 156 steden en dorpen geklasseerd volgens de FFM (feature frequentie methode)*. Koninklijke Van Gorcum, Assen.

Susan Hunston. 2022. *Corpora in applied linguistics*, 2 edition. Cambridge University Press, Cambridge.

John D. Hunter. 2007. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.

IPA. 1999. *Handbook of the International Phonetic Association. A guide to the use of the international phonetic alphabet*. Cambridge University Press, Cambridge.

Ian Joo and Yu-Yin Hsu. 2023. Phonotacticon: A cross-linguistic phonotactic database. Preprint, Research Square Platform LLC.

Gerhard Jäger. 2019. Computational historical linguistics. *Theoretical Linguistics*, 45(3-4):151–182.

Marwan Kilani. 2020. FAAL: a feature-based aligning ALgorithm. *Language Dynamics and Change*, 11(1):30–76.

Grzegorz Kondrak. 2000. A new algorithm for the alignment of phonetic sequences. In *1st Meeting of the North American Chapter of the Association for Computational Linguistics*.

Johann-Mattis List. 2012. SCA: Phonetic alignment based on sound classes. In Marija Slavkovik and Dan Lassiter, editors, *New directions in logic, language, and computation*, pages 32–51. Springer, Berlin and Heidelberg.

Johann-Mattis List. 2014. *Sequence comparison in historical linguistics*. Düsseldorf University Press, Düsseldorf.

Johann-Mattis List. 2023. Open problems in computational historical linguistics [version 1; peer review: 3 approved, 2 approved with reservations]. *Open Research Europe*, 3(201):1–22.

Johann-Mattis List, Cormac Anderson, Tiago Tresoldi, and Robert Forkel. 2020. *PYCLTS. A Python library for the handling of phonetic transcription systems [Software Library, Version 3.0.0]*. Zenodo, Geneva.

Johann-Mattis List, Cormac Anderson, Tiago Tresoldi, and Robert Forkel. 2024. *Cross-Linguistic Transcription Systems [Dataset, Version 2.3.0]*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Johann-Mattis List and Robert Forkel. 2021. *CL Toolkit. A Python Library for the Processing of Cross-Linguistic Data [Software Library, Version 0.1.1]*. Zenodo, Geneva.

Johann-Mattis List and Robert Forkel. 2024. A new Python library for the manipulation and annotation of linguistics sequences. *Computer-Assisted Language Comparison in Practice*, 7(1):17–23.

Johann-Mattis List, Robert Forkel, Simon J. Greenhill, Christoph Rzymski, Johannes Englisch, and Russell D. Gray. 2022. Lexibank, a public repository of standardized wordlists with computed phonological and lexical features. *Scientific Data*, 9(1):1–16.

Johann-Mattis List, Robert Forkel, Simon J. Greenhill, Christoph Rzymski, Johannes Englisch, and Russell D. Gray. 2023. *Lexibank [Database, Version 1.0]*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Johann-Mattis List and Olena Shcherbakova. 2023. Retrieving and Analyzing Taste Colexifications from Lexibank. *Computer-Assisted Language Comparison in Practice*, 6(2):73–86.

Florian Lux and Thang Vu. 2022. Language-agnostic meta-learning for low-resource text-to-speech with articulatory features. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6858–6868, Dublin, Ireland. Association for Computational Linguistics.

Carlo Meloni, Shauli Ravfogel, and Yoav Goldberg. 2021. Ab antiquo: Neural proto-language reconstruction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4460–4473, Online. Association for Computational Linguistics.

Florian Metze. 2007. On using articulatory features for discriminative speaker adaptation. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 117–120.

Steven Moran. 2012. *Phonetics Information Base and Lexicon*. Phd, University of Washington.

Steven Moran and Daniel McCloy, editors. 2019. *PHOIBLE 2.0*. Max Planck Institute for the Science of Human History, Jena.

David R. Mortensen, Patrick Littell, Akash Bharadwaj, Kartik Goyal, Chris Dyer, and Lori Levin. 2016. Panphon: A resource for mapping IPA segments to articulatory feature vectors. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3475–3484.

John Nerbonne and Wilbert Heeringa. 1997. Measuring dialect distance phonetically. In *Computational phonology: Third Meeting of the ACL Special Interest Group in Computational Phonology*.

Jessica Nieder and Johann-Mattis List. 2024. A computational model for the assessment of mutual intelligibility among closely related languages. In *Proceedings of the 6th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, page 37–43, St. Julian's, Malta. Association for Computational Linguistics.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Arne Rubehn. 2022. A feature-based neural model of sound change informed by global lexicostatistical data. Master's thesis, University of Tübingen.

Miikka Silfverberg, Lingshuang Jack Mao, and Mans Hulden. 2018. Sound analogies with phoneme embeddings. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2018*, pages 136–144, Salt Lake City, Utah.

Hedvig Skirgård, Hannah J. Haynie, Damián E. Blasi, Harald Hammarström, Jeremy Collins, Jay J. Latarche, Jakob Lesage, Tobias Weber, Alena Witzlack-Makarevich, Sam Passmore, Angela Chira, Luke Maurits, Russell Dinnage, Michael Dunn, Ger Reesink, Ruth Singer, Claire Bowern, Patience Epps, Jane Hill, Outi Vesakoski, Martine Robbeets, Noor Karolin Abbas, Daniel Auer, Nancy A. Bakker, Giulia Barbos, Robert D. Borges, Swintha Danielsen, Luise Dorenbusch, Ella Dorn, John Elliott, Giada Falcone, Jana Fischer, Yustinus Ghanggo Ate, Hannah Gibson, Hans-Philipp Göbel, Jemima A. Goodall, Victoria Gruner, Andrew Harvey, Rebekah Hayes, Leonard Heer, Roberto E. Herrera Miranda, Nataliia Hübler, Biu Huntington-Rainey, Jessica K. Ivani, Marilen Johns, Erika Just, Eri Kashima, Carolina Kipf, Janina V. Klingenberg, Nikita König, Aikaterina Koti, Richard G. A. Kowalik, Olga Krasnoukhova, Nora L. M. Lindvall, Mandy Lorenzen, Hannah Lutzenberger, Tânia R. A. Martins, Celia Mata German, Suzanne van der Meer, Jaime Montoya Samamé, Michael Müller, Saliha Muradoglu, Kelsey Neely, Johanna Nickel, Miina Norvik, Cheryl Akinyi Oluoch, Jesse Peacock, India O. C. Pearey, Naomi Peck, Stephanie Petit, Sören Pieper, Mariana Poblete, Daniel Prestipino, Linda Raabe, Amna Raja, Janis Reimringer, Sydney C. Rey, Julia Rizaew, Eloisa Ruppert, Kim K. Salmon, Jill Sammet, Rhiannon Schembri, Lars Schlabbach, Frederick W. P. Schmidt, Amalia Skilton, Wikaliler Daniel Smith, Hilário de Sousa, Kristin Sverredal, Daniel Valle, Javier Vera, Judith Voß, Tim Witte, Henry Wu, Stephanie Yam, Jingting Ye, Maisie Yong, Tessa Yuditha, Roberto Zariquiey, Robert Forkel, Nicholas Evans, Stephen C. Levinson, Martin Haspelmath, Simon J. Greenhill, Quentin D. Atkinson, and Russell D. Gray. 2023. Grambank reveals the importance of genealogical constraints on linguistic diversity and highlights the impact of language loss. *Science Advances*, 9(16).

Pavel Sofroniev and Çağrı Çöltekin. 2018. Phonetic vector representations for sound sequence alignment. In *Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 111–116, Brussels, Belgium.

Harold Somers. 1998. Similarity metrics for aligning children's articulation data. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 2*, pages 1227–1232.

Marlene Staib, Tian Huey Teh, Alexandra Torresquintero, Devang S. Ram Mohan, Lorenzo Foglianti, Raphael Lenain, and Jiameng Gao. 2020. Phonological Features for 0-shot Multilingual Speech Synthesis. In *Proceedings of INTERSPEECH 2020*, Shanghai, China.

George Steel and Peter Jurgec. 2017. *Featurize!: An online tool for natural classes and phonological features*. University of Toronto, Toronto.

Tao Tao, Su-Youn Yoon, Andrew Fister, Richard Sproat, and ChengXiang Zhai. 2006. Unsupervised named entity transliteration using temporal and phonetic correlation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 250–257.

Tiago Tresoldi. 2020. A model of distinctive features for computer-assisted language comparison. *Computer-Assisted Language Comparison in Practice*, 3(6).

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Floris van Vugt and Bruce Hayes. 2021. *Pheatures Spreadsheet: User's Manual*. UCLA, Los Angeles.

Michael L. Waskom. 2021. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021.

Søren Wichmann, André Müller, Annkathrin Wett, Viveka Velupillai, Julia Bischoffberger, Cecil H. Brown, Eric W. Holman, Sebastian Sauppe, Zarina Molochieva, Pamela Brown, Harald Hammarström, Oleg Belyaev, Johann-Mattis List, Dik Bakker, Dmitry Egorov, Matthias Urban, Robert Mailhammer, Agustina Carrizo, Matthew S. Dryer, Evgenia Korovina, David Beck, Helen Geyer, Pattie Epps, Anthony Grant, and Pilar Valenzuela. 2013. *The ASJP Database*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Su-Youn Yoon, Kyoung-Young Kim, and Richard Sproat. 2007. Multilingual transliteration using feature based phonetic method. In *Proceedings of the 45th Annual meeting of the association of computational linguistics*, pages 112–119.

Elizabeth C. Zsiga. 2013. *The Sounds of Language: An Introduction to Phonetics and Phonology*, volume 7. John Wiley & Sons.