PanPhon: A Resource for Mapping IPA Segments to Articulatory Feature Vectors

David R. Mortensen, Patrick Littell, Akash Bharadwaj,
Kartik Goyal, Chris Dyer, Lori Levin
Carnegie Mellon University
Language Technologies Institute
5000 Forbes Ave., Pittsburgh PA 15213
United State of America
dmortens@cs.cmu.edu, plittell@cs.cmu.edu, akashb@cs.cmu.edu,
kartikgo@cs.cmu.edu, cdyer@cs.cmu.edu, lsl@cs.cmu.edu

Abstract

This paper contributes to a growing body of evidence that—when coupled with appropriate machine-learning techniques—linguistically motivated, information-rich representations can outperform one-hot encodings of linguistic data. In particular, we show that phonological features outperform character-based models using the PanPhon resource. PanPhon is a database relating over 5,000 IPA segments to 21 subsegmental articulatory features. We show that this database boosts performance in various NER-related tasks. Phonologically aware, neural CRF models built on PanPhon features are able to perform comparably to character-based models on monolingual Spanish and Turkish NER tasks. On transfer models (as between Uzbek and Turkish) they have been shown to perform better. Furthermore, PanPhon features also contribute measurably to Orthography-to-IPA conversion tasks.

1 Introduction

This paper introduces PanPhon[1], a resource consisting of a database that relates over 5,000 IPA segments (simple and complex) to their definitions in terms of 21 articulatory features (see Tab. 1) as well as a Python package for interacting with this database and manipulating the representations that it provides. While our previous publications (summarized in §4) have described experiments using it, this is the first full description of PanPhon. Combined with a sister package, Epitran[2], it allows the conversion of

	syl	son	cons	cont	delrel	lat	nas	strid	voi	sg	cg	ant	cor	distr	lab	hi	lo	back	round	tense	long
/p/	-	-	+	-	-	-	-	0	-	-	-	+	-	0	+	-	-	-	-	0	-
/p^{h}/	-	-	+	-	-	-	-	0	-	+	-	+	-	0	+	-	-	-	-	0	-
/p^{j}/	-	-	+	-	-	-	-	0	-	-	-	+	-	0	+	+	-	-	-	0	-
/p^{hj}/	-	-	+	-	-	-	-	0	-	+	-	+	-	0	+	+	-	-	-	0	-

Table 1: Illustration of IPA segments and feature vectors from PanPhon

orthographic texts to sequences of articulatory feature vectors. The Epitran-Panphon pipeline is illustrated in Fig. 1. The input to Epitran consists of word tokens in orthographic representation. Take, for example, the Spanish word *Madrid*. Epitran converts this string to a phonemic (not phonetic) representation in IPA, in this case /madɾid/. Epitran then calls a PanPhon function to convert this IPA string into a sequence of feature vectors. It then returns this sequence, aligned with the orthographic representation, capitalization, and Unicode character category features.

[1] https://github.com/dmort27/panphon

[2] https://github.com/dmort27/epitran

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 3475–3484, Osaka, Japan, December 11-17 2016.

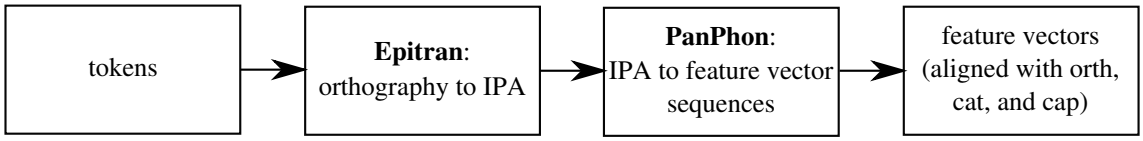

Figure 1: Feature vector pipline

This paper also shows that subsegmental features, as encoded in PanPhon, are useful in NLP tasks. The specific tasks for which PanPhon has been shown to improve performance are named entity recognition and conversion of lossy orthographies[3] to IPA.

Phonologists have long held that articulatory features play a role in three central aspects of phonology: **contrast**, **distribution**, and **alternation**. These constitute the distinctiveness of speech sounds, the restrictions on where they can occur in the speech-stream, and the phonologically-derived differences in the various realizations of the same morpheme. These may be illustrated through examples from Turkish (Lees, 1963) as illustrated in Tab. 2. The difference between /i/ and /e/ is sufficient to distinguish two

NOM.SG	GEN.SG	NOM.PL	GEN.PL	gloss
ip	ip-**in**	ip-**ler**	ip-**ler-in**	'rope'
el	el-**in**	el-**ler**	el-**ler-in**	'hand'
kɪz	kɪz-**ɪn**	kɪz-**lar**	kɪz-**lar-ɪn**	'girl'

Table 2: Turkish vowel harmony (NOM = nominative, GEN = genitive, SG = singular, and PL = plural)

words, evidence that the feature [±high][4] is contrastive (sufficient to make lexical contrasts) in Turkish. The vowels in the suffixes (*-ler*/*-lar* and *-in*/*-ın*) alternate to assimilate in quality to the preceding vowel. This can be easily expressed in terms of the phonological feature [±back]: The vowels ⟨i⟩ and ⟨e⟩ are [−back] while ⟨a⟩ and ⟨ɪ⟩ are [+back][5]. The alternation consists of the smallest change that allows agreement in the specification of this feature between the vowels in the root at the vowel in the suffix. In this way, phonological features allow a degree of generalization, even over orthographic patterns, that purely character-based (and also phoneme-based) models do not. Finally, the static observation that, in all but a minority of "disharmonic" words, each vowel in a word shares the same specification for [±high] and [±back] is a matter of distribution. These linguistic insights behind articulatory features translate into practical benefits for NLP systems.

The usefulness of articulatory features in natural language processing, this paper will show, is most valuable for replacing character-level models where characters bear some predictable relationship to phonemes or phones. In such a scenario, treating characters as atomic entities is rather like treating the chords of a musical score as unanalyzable units. While this approach may be adequate for many purposes, we will argue that it ignores aspects of phonological structure that have demonstrable utility. Articulatory features represent the parts in a layered articulatory score. Operations on this score target these individual features, not the segment as a whole ("the chord"), contrary to the assumption made in strictly character-based approaches. We have found, from a variety of perspectives, that exploiting these components can improve the performance of relevant NLP systems.

2 Past Use of Phonological Features in NLP

Within both formal and empirical linguistics, it is rare to encounter discussion of structural patterns in sounds without some mention of phonological features. Even radically empirical phonologists, rather than denying the existence of such phonological features, tend to simply deny that there is a universal

[3]"Lossy orthographies" are ambiguous writing systems that lose segmental information present in the speech stream.

[4]In the linguistic subfield of phonology, features are often represented in square brackets with the name on the right and the value presented as +, −, or ± (indicating that the feature is binary but that the value is not known).

[5]Note that alternate feature systems use the feature [±front] to account for this contrast.

and innate feature inventory (Mielke, 2008). Likewise, in the speech sciences and speech technology, phonological features have been the subject of widespread inquiry (Bromberg et al., 2007; Metze, 2007). This contrasts with NLP, where phonological features have been subject to less experimentation, perhaps because of the perceived lower relevance of phonology than morphology, syntax, and semantics to NLP tasks. However, some promising NLP results have been achieved using phonological features.

In one of the more widely-cited cases of this kind, Gildea and Jurafsky (1996) added phonological biases—stated in terms of phonological features—to aid the OSTIA (the Onward Subsequential Transducer Inferance Algorithm) in learning phonological rules. Subsequently , Tao et al. (2006) used hand-weighted articulatory feature edit distance (augmented with "pseudofeatures") to facilitate the transliteration of named entities. This feature-based system outperformed a temporal algorithm on a English/Hindi language pair and contributed to the performance of the best model the authors tested. In a successor study, Yoon et al. (2007) employed the model of Tao et al. (2006) but with features weighted by the winnow algorithm rather than by hand; they achieved comparable results without engineered weights. In a related strain of research, Kondrak and Sherif (2006) explored phonological similarity measures based, in some cases, on a kind of phonological feature system (but with a multivalued place feature unlike the binary and ternary features integral to PanPhon).

3 PanPhon and Its Functionality

PanPhon facilitates further experimentation with phonological (specifically, articulatory) features. Our goal in implementing PanPhon was not to implement a state-of-the-art feature system (from the standpoint of linguistic theory) but to develop a methodologically solid resource that would be useful for NLP researchers. Contemporary feature theories posit hierarchical and non-linear feature structures; they cannot be easily expressed in terms of vectors of binary or ternary values like the earlier-vintage system represented in PanPhon. Linear phonological models like that instantiated in PanPhon are arguably less explanatory than non-linear and feature-geometric models, but they can also be said to hew closer to the empirical ground. One limitation of PanPhon involves the representation of tone, one area in which non-linear representations are almost universally conceded to hold the upper hand. PanPhon is segmental by design and tone is suprasegmental by nature.

In constructing PanPhon, our approach was to start with a subset of the International Phonetic Alphabet (or IPA) where every segment represents a sound that is distinct from the others. Contrary to its design principles, the IPA allows multiple transcriptions for the same sound in a variety of cases. An attempt was made, using consensus definitions, to classify each of these segments according to binary (and occasionally, ternary) features. This consensus feature set and the accompanying definitions were based on a survey of the phonological literature.

PanPhon's contribution lies in the following attributes:

Universal. PanPhon will, when queried with a legal, segmental Unicode IPA string, return a sequence of valid vectors of articulatory feature values. Currently, it defines 5,395 simple and complex IPA characters in terms of 21 articulatory features.

Empirically verified. The general feature system used in PanPhon has been widely tested by linguists across a great range of phenomena and languages and found to be effective at modeling the phonological and morphophonological patterns of the world's languages.

Unicode compliant. PanPhon uses the Unicode encoding of the International Phonetic Alphabet both internally and externally. This provides human readability (or, at least, readability to human linguists) in a way that ASCII mappings of all or part of the IPA, like ARPAbet, WorldBet, SAMPA, and X-SAMPA do not. Compare German *müde* 'tired' in ARPAbet /M <no_symbol> D AH/, WordBet /m y: d &/, SAMPA /m y: d @/, X-SAMPA /m y: d @/ with IPA /myːdə/. The development of convenient input methods and appropriate rendering technologies for IPA mitigate much of the past difficulty involved in using it in computing applications.

Open source. Unlike many existing phonological feature resources[6], PanPhon is freely available under a liberal license (MIT).

PanPhon consists of a collection of components:

1. A database relating unmodified IPA segments to vectors of 21 features. These form the core of the database. The features are listed here in the canonical order in which they appear in the database:

   syl [±syllabic]. Is the segment the nucleus of a syllable?
   son [±sonorant]. Is the segment produced with a relatively unobstructed vocal tract?
   cons [±consonantal]. Is the segment consonantal (not a vowel or glide, or laryngeal consonant)?
   cont [±continuant]. Is the segment produced with continuous oral airflow?
   delrel [±delayed release]. Is the segment an affricate?
   lat [±lateral]. Is the segment produced with a lateral constriction?
   nas [±nasal]. Is the segment produced with nasal airflow?
   strid [±strident]. Is the segment produced with noisy friction?
   voi [±voice]. Are the vocal folds vibrating during the production of the segment?
   sg [±spread glottis]. Are the vocal folds abducted during the production of the segment?
   cg [±constricted glottis]. Are the vocal folds adducted during the production of the segment?
   ant [±anterior]. Is a constriction made in the front of the vocal tract?
   cor [±coronal]. Is the tip or blade of the tongue used to make a constriction?
   distr [±distributed]. Is a coronal constriction distributed laterally?
   lab [±labial]. Does the segment involve constrictions with or of the lips?
   hi [±high]. Is the segment produced with the tongue body raised?
   lo [±low]. Is the segment produced with the tongue body lowered?
   back [±back]. Is the segment produced with the tongue body in a posterior position?
   round [±round]. Is the segment produced with the lips rounded?
   tense [±tense]. Is the segment produced with an advanced tongue root.

   Feature vectors from PanPhon for a few example segments (both simple and complex) are shown in Tab. 1.

2. A collection of rules, written in user-editable YAML, that describe diacritics and modifiers—the Unicode codepoint of the modifier, the feature specifications that provide the necessary context for adding the diacritic or modifier, and the feature specifications changes that take place if the diacritic or modifier is added to a segment. An example of one of these YAML rules is shown below:

Listing 1: Diacritic rule

```
- marker: ʷ
  name: Labialized
  position: post
  conditions:
    - syl:  "-"
  exclude:
    - w
    - ʍ
    - ɥ
  content:
    round: "+"
    back:  "+"
    hi:  "+"
```

[6]But see also PHOIBLE's phonological feature set (Moran et al., 2014).

This example shows a rule (named "Labialized") that adds the modifier (marker) "ʷ" after a segment—post(fix), if the segment has the feature [−syllabic] and is not /w/, /ʍ/, or /ɥ/. The new segment has the same features as the input segment except that is it [+round], [+back], and [+hi].

3. A script for applying diacritics and modifiers, as defined in 2, to segments, as defined in 1 and the comprehensive segment database produced by this script.

4. A set of Python convenience functions and classes for accessing, manipulating, and employing the phonological feature vectors associated with any segment, whether simple or complex:

   (a) Greedily parsing IPA strings into segments defined in the database.
   (b) Converting such a sequence of segments into a sequence of articulatory feature vectors.
   (c) Querying sets of segments based upon their features.
   (d) Pattern matching against strings using feature specifications to define character classes.
   (e) Computing phonological distance: weighted and unweighted feature edit distance.
   (f) Computing phonological distance: Levenshtein distance between strings with collapsed, phonologically-based segment equivalence classes.

In some applications, PanPhon is used with a sister library, Epitran, or another—more specialized—package for converting orthography to IPA. Epitran provides a simple interface for quickly implementing grapheme to phoneme mappings for languages with phonemically adequate orthographies. It includes mappings for a variety of languages including Spanish, Dutch, Turkish, and Uyghur.

4 Empirical Evaluation of PanPhon

It is not simply the case that the articulatory features available through PanPhon are well founded in terms of linguistic theory. It is also true that they have been demonstrated to improve the performance of certain machine learning models at certain NLP tasks. This section summarizes the contribution of PanPhon to two classes of tasks: orthography-to-IPA character transduction and named entity recognition (NER). While the second set of experiments (on NER) were prompted by a need to test a particular class of model—phonologically aware LSTM-CRFs—the first set were motivated by the need to solve a particular problem: how best to convert Sorani Kurdish (a Northwestern Iranian language of Iraq and Iran) from orthographic to IPA representation.

4.1 Orthography-IPA Character Transduction

As part of a NER system for the low-resource Sorani Kurdish language, we developed a Sorani-orthography-to-IPA converter Littell et al. (2016). This was challenging because the Sorani orthography, like many Perso-Arabic scripts, badly underdetermines the equivalent phonetic representation. The following steps summarize the workflow behind building the Sorani-to-IPA converter:

1. Human linguists identify the orthographic units (i.e., characters and multigraphs) in the script.

2. Human linguists identify the possible IPA representations of each orthographic unit, using knowledge of the language and the writing system.

3. The system generates all possible hypotheses for a subset of tokens of the language using the mapping developed in the previous step.

4. Human linguists generate training data using a grammar and lexicon (Thackston, 2006) by picking one or more valid pronunciations (or, if unknown, one or more likely hypotheses) for a selection of tokens.

5. Character-level chain conditional random field (CRF) is trained (Lafferty et al., 2001; Dyer et al., 2010) on resulting data.

The hypothesis space within which the CRF operates was determined by the symbol-level IPA map developed in the first step of our work-flow. It is important to note that we allowed many-to-many mapping between orthographic input character sequences and IPA output character sequences in the sense that a single input character can be mapped to multiple IPA symbols and an input multigraph (consisting of multiple orthographic characters) can be mapped to a single IPA symbol.

PanPhon feature vectors were used to create one set of features that were consumed by the CRF. This move was motivated by the insight that phonotactic patterns—patterns in the sequences of speech sounds—tend to be based around the very sorts of classes that phonological features are intended to describe. For the purposes of these experiments, we divided these features into six classes which correspond largely to classes widely used by phonologists:

- **Major Class** features—represent the major classes of sounds: [±syllable], [±sonorant], [±consonantal], [±continuant].
- **Laryngeal** features—specify the glottal states of sounds: [±voice], [±spread glottis], [±constricted glottis].
- **Major place** features—focus on the place of articulation: [±anterior], [±coronal], [±labial] and [±distributed][7].
- **Minor Place** features—related to the position of the dorsum in the tongue: [±high], [±low] and [±back].
- **Manner** features—categorize IPA symbols according to their manner of articulation: [±nasal], [±lateral], [±delayed release] and [±strident].
- **Minor Manner** features—the attributes in this group were [±round] and [±tense].

We then generated the training data with 244 instances (types)[8] and report performance of their IPA predictor on 402 instances. In Table 3 (adapted from Littell et al. (2016)), we compare the accuracy and character error rate (CER) of the predictor, when using PanPhon features compared to, and along with:

- **Basic** features pertaining to the usage of orthography-to-IPA symbol translation rules, and whether the IPA symbols were identified as consonants and vowels[9].
- **Phon** features derived from the PanPhon IPA-to-feature-vector package.
- **Kurmanji** features derived from a 4-gram language model, built using SRILM (Stolcke, 2002), from the IPA-ized Kurmanji corpus. Kurmanji is a "sister" language of Sorani having a phonologically unambiguous orthography.
- **Tajik** features, derived as above, from the IPA-ized Tajik corpus. Tajik is a close cousin of Sorani having a phonologically unambiguous orthography.

While the single most important class of features, all told, were those derived from the Kurmanji language model, the inclusion of PanPhon features gave similar gains to the inclusion of the Tajik features, and the best results overall came from the inclusion of both Kurmanji and PanPhon features. This is a welcome result—while not all languages have a close sister variety with an unambiguous orthography, PanPhon-based features are universally available.

Among the PanPhon features, the most valuable features for this task were the *major class features*, *laryngeal features*, and *place features*. Unsurprisingly, having a language model from a sister language (in this case, Kurmanji) is the most useful single source of features for IPA conversion. However, it is surprising to find that just having the universal features derived from PanPhon vectors were as useful as having a language model from a cousin language (in this case, Tajik).

[7]Phonologists do not generally consider the feature [±distributed] to be a major place feature. This appears to have been an error in our implementation of the feature classes.

[8]It is worth noting that when reducing the training set to a quarter of this, we observed only slightly worse results, suggesting that very little manual effort would be necessary to generate training data for a task like this.

[9]The consonant/vowel feature should have been largely equivalent to the feature [±syllabic].

Features	Accuracy	CER
Basic	0.635	0.237
Basic+phon.	0.669	0.234
Basic+Kurmanji	0.701	0.223
Basic+Kurmanji+phon.	0.721	0.221
Basic+Tajik	0.661	0.231
Basic+Tajik+phon.	0.664	0.228
All features	0.721	0.221

Table 3: IPA prediction for Sorani, trained on 244 tokens, tested on 402 tokens

4.2 NER with Phonologically-Aware Neural Models

We subsequently experimented with PanPhon—and its sister package, Epitran—in performing NER with a character-based LSTM-CRF architecture (Bharadwaj et al., 2016). We made this architecture phonologically-aware by substituting phonological feature vectors from PanPhon for characters. We used the resulting features in a series of NER experiments in both monolingual and transfer scenarios. As a baseline, we employed a character based LSTM-CRF NER system with features from pre-trained word vectors.

In a series of monolingual experiment using CoNLL 2002 data from Spanish (see Tab. 4, adapted from (Bharadwaj et al., 2016)) it was found that substituting PanPhon and phonological attention features for orthographic and orthographic attention features (in a model also incorporating word vector, capitalization, and Unicode character category features) raised F1 from 85.25 to 85.81 and yielded the best-performing model in the series. In a second series of monolingual experiments using the LDC2014E115 BOLT Turk-

Features	F1
WVec	83.61
WVec+Phon	84.08
WVec+Orth	84.52
WVec+Phon+Orth+PhonAttn	84.53
WVec+Orth+OrthAttn	84.64
WVec+Phon+PhonAttn	84.88
WVec+Phon+Cap+Cat	84.89
WVec+Orth+Cap+Cat	84.91
WVec+Phon+Orth+Cap+Cat	84.92
WVec+Orth+OrthAttn+Cap+Cat	85.25
WVec+Phon+PhonAttn+Cap+Cat	**85.81**
WVec+Phon+Orth+OrthAttn+Cap+Cat	85.32
WVec+Phon+PhonAttn+Orth+Cap+Cat	84.84
All features	84.75

Table 4: Ablation tests on Spanish NER; bold indicates best model; factors are pre-trained word vectors (WVec), PanPhon features (Phon), phonological attention features (PhonAttn), orthographic features (Orth), orthographic attention features (OrthAttn), capitalization features (Cap), and Unicode category features (Cat)

ish Language Pack, Bharadwaj et al. (2016) found that the best model incorporated PanPhon as well as orthographic features (see Tab. 5. The second-best scoring model uses only PanPhon features and phonological attention features (in addition to pretrained word vectors) and because it is character-independent, it is more suitable for cross-lingual transfer. Finally, Bharadwaj et al. (2016) conducted a series of Uzbek-to-Turkish cross-lingual transfer experiments using the LDC2015E89 BOLT data pack for Uzbek and the

Features	F1
WVec	49.2
WVec+Orth	65.41
WVec+Orth+OrthAttn	64.76
WVec+Orth+Cap+Cat	60.57
WVec+Orth+Cap+Cat+OrthAttn	60.87
WVec+Phon	63.04
WVec+Phon+PhonAttn	**66.07**
WVec+Phon+Cap+Cat	59.08
WVec+Phon+Cap+Cat+PhonAttn	62.46
WVec+Phon+Orth+PhonAttn	63.43
WVec+Phon+Orth+Cap+Cat+PhonAttn	63.46
All features	**66.47**

Table 5: Ablation tests on Turkish NER; Boldface indicates the best model (66.47); factors are pre-trained word vectors (WVec), PanPhon features (Phon), phonological attention features (PhonAttn), orthographic features (Orth), orthographic attention features (OrthAttn), capitalization features (Cap), and Unicode category features (Cat)

LDC2014E115 BOLT data pack for Turkish. Unsurprisingly, monolingual models with no training data achieve an F1 of zero. In a transfer situation, with no Turkish training data, word vectors alone give an F-score of 2.09. However, using phonological features and phonological attention features with zero training data in the target language (but training in the transfer language) yields an F-score of 11.9. Adding capitalization and Unicode category features allowed the resulting model to achieve an F1 of 26.92 in a zero-shot scenario.

Features	**Source**	**Target 0-shot**	**5% data**	**20% data**	**40% data**	**60% data**	**80% data**	**All data**
WVec	41.87	2.09	23.44	35	42.75	46.32	48.81	50.34
WVec+Phon+PhonAttn	61.24	11.9	34.06	47.84	56.1	53.5	64.72	65.2
WVec+Phon+Cap+Cat	60.92	15.55	39.42	**60.14**	**63.23**	**62.54**	**65.24**	**65.63**
All features	**61.85**	**26.92**	**47.21**	58.58	60.32	60.7	62.84	63.58
Monolingual Models		**Target 0-shot**	**5% data**	**20% data**	**40% data**	**60% data**	**80% data**	**All data**
LSTM-CRF (Lample et al., 2016)		0	33.44	50.61	53.25	57.41	60	61.11
S-LSTM (Lample et al., 2016)		0	15.41	39.33	42.99	51.92	51.55	56.58

Table 6: Model transfer from Uzbek to Turkish at different target data availability thresholds compared to monolingual Turkish baseline, also at different data availability thresholds; factors are pre-trained word vectors (WVec), PanPhon features (Phon), phonological attention features (PhonAttn), capitalization features (Cap), and Unicode category features (Cat)

What do these experiments reveal about the value of PanPhon for NER? The monolingual experiments may not seem compelling, but they drive home an important point: even though conversion from orthography to PanPhon vectors entails a significant loss of information, it does not seem to entail a loss of performance. It is even possible that phonological features facilitate NER in the monolingual case by helping a neural NER system to identify phonologically aberrant tokens which are more likely to reflect borrowed lexical items which, in turn, are more likely to be named entities. The real benefit of using phonological feature representations in NER, however, is manifest in transfer scenarios like the Uzbek-Turkish transfer scenario we explored in Bharadwaj et al. (2016). By projecting both the source and target language into a common phonological space, similarities that would be masked in an orthographic space become accessible to a neural NER system. The value of such an approach is most obvious when it is applied to relatively closely-related languages that are written in different orthographies like Uzbek and

Turkish[10].

5 Conclusion

This paper has reported the creation of a new resource, PanPhon, a database of IPA segment-phonological feature correspondences with a collection of code for exploiting these relationships. Additionally, we briefly documented the PanPhon database and modules. Most significantly, we showed that PanPhon features can improve performance in two NLP tasks: orthography-to-IPA conversion and NER. At this point, the PanPhon feature mapping has still not be tested against other phonological feature implementations (which have typically not been made widely available) in these tasks, so it cannot be claimed that PanPhon boosts performance more than these other databases. However, it can be said conclusively that systems *like* PanPhon are a useful component in some NLP systems.

Acknowledgements

Supported by the U.S. Army Research Office under grant number W911NF-10-1-0533. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the view of the U.S. Army Research Office or the United States Government.

Sponsored by Defense Advanced Research Projects Agency (DARPA) Information Innovation Office (I2O) Program: Low Resource Languages for Emergent Incidents (LORELEI) Issued by DARPA/I2O under Contract No. HR0011-15-C-0114.

References

Akash Bharadwaj, David Mortensen, Chris Dyer, and Jaime G. Carbonell. 2016. Phonologically aware neural model for named entity recognition in low resource transfer settings. Accepted for EMNLP 2016.

Ilana Bromberg, Jeremy Morris, and Eric Fosler-Lussier. 2007. Joint versus indepedent phonological feature models within CRF phone recognition. In *Proceedings of NAACL HLT 2007*, pages 13–16. Association for Computational Linguistics.

Chris Dyer, Adam Lopez, Juri Ganitkevitch, Johnathan Weese, Ferhan Ture, Phil Blunsom, Hendra Setiawan, Vladimir Eidelman, and Philip Resnik. 2010. cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *Procedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*.

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

Grzegorz Kondrak and Tarek Sherif. 2006. Evaluation of several phonetic similarity algorithms on the task of cognate identification. In *Proceedings of the Workshop on Linguistic Distances*, pages 43–50. Association for Computational Linguistics.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. arXiv preprint arXiv:1603.01360.

Robert B. Lees. 1963. *The Phonology of Modern Standard Turkish*. Indiana University Press, Bloomington.

Patrick Littell, David Mortensen, Kartik Goyal, Chris Dyer, and Lori Levin. 2016. Bridge-language capitalization inference in Western Iranian: Sorani, Kurmanji, Zazaki, and Tajik. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 16)*.

Florian Metze. 2007. On using articulatory features for discriminative speaker adaptation. In *Proceedings of NAACL HLT 2007, Companion Volume*. Association for Computational Linguistics.

[10]Note that while Turkish and Uzbek are both written with the Latin alphabet, they use considerably different orthographic conventions that hide similarities in pronunciation behind differences in symbols. The utility of PanPhon is illustrated even more dramatically in the Turkish/Uzbek-Uyghur transfer scenario that is mentioned briefly by Bharadwaj et al. (2016), since Uyghur uses a Perso-Arabic script that differs entirely from the Latin scripts that are now used for Turkish and Uzbek.

Jeff Mielke. 2008. *The Emergence of Distinctive Features*. Oxford University Press, Oxford.

Steven Moran, Daniel McCloy, and Richard Wright, editors. 2014. *PHOIBLE Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Andreas Stolcke. 2002. SRILM — an extensible language modeling toolkit. In J. H. L. Hansen and B. Pellom, editors, *ICSLP*, volume 2, pages 901–904.

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 (EMNLP 2006)*, pages 250–257. Association for Computational Linguistics, July.

W. M. Thackston. 2006. Sorani Kurdish reference grammar with selected readings. Book manuscript.

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 (ACL 2007)*, pages 112–119. Association for Computational Linguistics.