Bridge-Language Capitalization Inference in Western Iranian: Sorani, Kurmanji, Zazaki, and Tajik

Patrick Littell, David Mortensen, Kartik Goyal, Chris Dyer, Lori Levin
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
plittell@cs.cmu.edu, dmortens@cs.cmu.edu, kartikgo@cs.cmu.edu, cdyer@cs.cmu.edu, lsl@cs.cmu.edu

Abstract

In Sorani Kurdish, one of the most useful orthographic features in named-entity recognition – capitalization – is absent, as the language's Perso-Arabic script does not make a distinction between uppercase and lowercase letters. We describe a system for deriving an inferred capitalization value from closely related languages by phonological similarity, and illustrate the system using several related Western Iranian languages.

Keywords: Kurdish, named-entity recognition, phonology

1. Introduction

In constructing a named-entity recognition system for Sorani Kurdish (Gautier, 1998; Thackston, 2006; Walther and Sagot, 2010; Esmaili and Salavati, 2013), a low-resource Western Iranian language written in a Perso-Arabic script, we were faced with a dilemma: one of the most useful orthographic features in named-entity recognition—capitalization—is absent in Perso-Arabic writing.

However, within the Western Iranian family there are several languages, including Kurmanji Kurdish, Zazaki, and Tajik, that are written in Latin or Cyrillic scripts and therefore *do* feature capitalization. This article details the process we developed and the challenges we faced in attempting to infer a "surrogate" capitalization feature for Sorani named entity recognition based on Kurmaji, Zazaki, and Tajik sources.

The question we are attempting to answer, and the process that we developed for answering it, is more general than just capitalization inference. Broadly, it is the question "How, for a language that is low-resourced with respect to feature F, can we infer values for F from material in a closely-related language?" This question, of *bridge-language feature inference*, could equally be asked of various other features: the presence of vowels in scripts that only partially distinguish them, word class, tonal features, and, potentially, any lexical feature F.

2. Background

2.1. The larger project

This research took place as part of a pilot project on *linguistic rapid response* for emergency situations. When given data in a low-resource language on which they have not worked, how much can a small, interdisciplinary team process in a very short (24- or 48-hour) timeframe? In particular, what NLP milestones are possible within this timeframe when conventional textual and lexical resources are unavailable?

This project is valuable not just because of its practical applications, but because it spurs investigation of potential types of language resources that, in ordinary circumstances, might be overlooked. We had little in the way of gold-standard annotated data, English-Sorani parallel text, or Sorani-language lexica and gazetteers[1]; as mentioned above, even the familiar feature of capitalization was absent. This spurred us to consider what resources we might be able to adapt from "bridge languages": closely related languages that are better-resourced.[2]

2.2. Languages

2.2.1. Kurdish

Sorani (or "Central") Kurdish is a language in the Iranian branch of the Indo-European family, spoken by 6.7 million people in Iraqi Kurdistan and the Kurdistan Province of Iran.[3] It is written in a Perso-Arabic script, with modifications that allow writers to indicate all but one of its vowels, short [i].

Sorani is closely related to Kurmanji (or "Northern") Kurdish, spoken by about 20 million speakers primarily in Eastern Turkey; Kurmanji and Sorani are sometimes described as dialects of the same language, but exhibit significant morphological differences. Kurmanji is usually written in a Latin script called *Bedirxan*.

2.2.2. Zazaki

Another language of Eastern Turkey, Zazaki (also known as Zaza, Kirmanjki, Kirdki, and Dimli) is usually not considered a part of the Kurdish language group in the narrow sense, although the majority of Zaza people identify ethnically as Kurds. It is likewise written in a Latin script; the Zazaki sources we utilized were written in Kurmanji *Bedirxan*, which, although it does not perfectly correspond to the Zazaki phonemic inventory, is sometimes chosen by writers to

[1] A Sorani lexicon is in development (Walther and Sagot, 2010), but was not yet accessible in its entirety.

[2] To be precise, Kurmanji is not a higher-resourced language than Sorani *in general* (Esmaili et al., 2013), but Kurmanji texts are an abundant source of the feature "capitalized" where Sorani texts are not, a clearer representation of the phonetic forms of words, etc.

[3] All population figures are from Lewis et al. (2015).

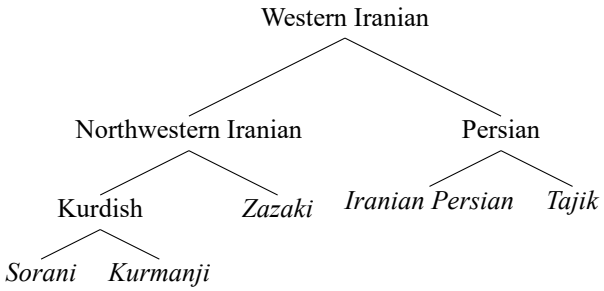

Figure 1: Kurdish and selected relatives

Language	Script family	Utilizes capitalization
Sorani Kurdish	Perso-Arabic	No
Kurmanji Kurdish	Latin	**Yes**
Zazaki	Latin	**Yes**
Iranian Persian	Perso-Arabic	No
Tajik	Cyrillic	**Yes**

Table 1: Selected Western Iranian languages and their writing systems

express their linguistic and ethnic solidarity with the Kurmanji.

2.2.3. Persian

The largest (and best-resourced) language of the Iranian family is Persian (or Farsi), with about 47 million speakers. The Iranian dialect of Persian is written in a Perso-Arabic script, and therefore not directly useful to the capitalization task, but the Tajik dialect, spoken by about 8 million speakers primarily in Tajikistan, is primarily written in a Cyrillic script. Iranian Persian and Tajik are sometimes considered dialects of the same language, and sometimes considered different languages.

2.2.4. Choosing a bridge language

Of these languages, the best "capitalization" surrogate for Sorani Kurdish is clearly Kurmanji Kurdish: they are closely related and textual material in Kurmanji is relatively plentiful compared to other closely-related languages. Zazaki is reasonably closely related, but the amount of textual material available is considerably less. Iranian Persian is the most extensively resourced language among the Western Iranian languages, but more distantly related to Sorani, and in any case written material lacks the feature we are interested in here. Tajik, while adequately-resourced for our purposes, is sufficiently distantly related that matching Sorani and Tajik roots can be a difficult task.

We include Zazaki and Tajik in this study, however, because it allows us, during evaluation, a means of getting error rates (since all of these languages capitalize, we can determine to what extent we predict the correct ones) and interpreting whether those error rates are reasonable for the languages involved (i.e., we would predict that using values from more distantly related Tajik should be more error-prone than values from the more closely-related Zazaki).

3. IPA conversion

We began with monolingual text in Sorani, Kurmanji, Zazaki, and Tajik; the texts were not parallel but were largely comparable, drawn from the Pewan corpus of Kurdish (Esmaili et al., 2013), news articles written in the last ten years and, in the case of Tajik, Wikipedia articles.

To render Perso-Arabic Sorani text, Latinate Kurmanji and Zazaki text, and Cyrillic Tajik text into comparable forms, we converted all text into IPA transliterations. The IPA, and the feature space (e.g. [±syllabic], [±coronal]) that conceptually underlies it, can be used to provide a common "space" by which distances between words in different writing systems can be measured in a uniform manner.

For the Latinate and Cyrillic texts (i.e., the Kurmanji, Zazaki, and Tajik texts), IPA conversion was straightforward, since these orthographies represent the respective languages' phonemes with little ambiguity.[4] For the Sorani texts, this process took greater attention, both because of complications in Unicode rendering, and because Sorani's Perso-Arabic script is more ambiguous.

3.1. Unicode normalization

An entirely digital complication in processing Sorani text regards the Unicode expression of the very frequent short vowel /ɛ/ (also transcribed as /æ/), which is expressed by a special variant of the Arabic letter *heh* (Esmaili et al., 2013). This complication is straightforward to fix, but we mention it here because it is also very easy to overlook, and thereby produce inaccurate phonetic renderings of Sorani written text.

Arabic is a cursive script, with letter forms that differ according to whether they join with the following letter ("initial"), the preceding letter ("final"), both ("medial"), or neither ("independent"). Most letters have distinct forms for all

[4] More precisely, while the orthography used in the Zazaki texts was not a 100% unambiguous rendition of the Zazaki phonemes, this ambiguity was not an issue for our task. The Zazaki texts ignored some Zazaki-specific distinctions in favor of writing in a more "pan-Kurdish" manner. Since our end goal was, in any case, the minimization of Kurdish-Zazaki distinctions in order to identify Kurdish-Zazaki shared lexical items, text that *already* attempts to minimize these distinctions is a benefit rather than a hindrance.

four positions, but six letters (specifically, ا د ذ ر ز و) never join with the following letter and therefore only have two distinct forms.

The Arabic letter *heh*, representing /h/, has all four forms, initial (ﻫ), medial (ﻬ), final (ﻪ), and independent (ه). In Sorani, however, these four forms express two different letters: the initial (ﻫ) and medial (ﻬ) forms (those that join with the following letter) express /h/, whereas the final (ﻪ) and independent (ه) forms (those that do not join with the following letter) express /ɛ/.[5] Put another way, Sorani has developed a seventh non-joining letter, representing /ɛ/, from the non-joining forms of the letter *heh*.

There exist separate Unicode points to distinguish these in Sorani (ARABIC LETTER HEH U+0647 and ARABIC LETTER AE U+06D5 respectively), but most Sorani writers are using Arabic-targeted software and fonts. Instead of expressing [ɛ] with ARABIC LETTER AE, they create the appropriate on-screen letter form by typing ARABIC LETTER HEH and then, when necessary, preventing it from joining with the following letter using the special invisible character ZERO WIDTH NON-JOINER U+200C. This suffices for typesetting purposes, but for text processing it can require downstream components to understand the rules of when ARABIC LETTER HEH should or should not be interpreted as if it represents /ɛ/. Instead, we simply normalize all instances of /ɛ/ to ARABIC LETTER AE at the outset, along with a few other normalizations for special letter forms that one can encounter (like the occasional occurrence of Persian-style ک for /k/ instead of the Arabic-style ك).

3.2. Vowel prediction

A more substantial problem involves the rendering of Sorani vowels. Arabic script does not, when writing Arabic, represent short vowels or make a distinction between long high vowels and glide consonants. However, this can be a source of greater ambiguity when Arabic scripts are used to write languages in which vowels carry a higher functional load – in particular, Indo-European languages like Iranian Persian and Sorani. While both Iranian Persian and Sorani have innovated new vowel letters and developed other disambiguatory strategies, ambiguities nonetheless remain. In particular, Sorani does not represent the short vowel [i], expresses [w] and [u] identically, and expresses [y] and [iː] identically, and has a few additional context-dependent ambiguities.

3.3. Implementation: CRF

We implemented and trained a character level linear chain conditional random field (CRF) based system for converting the Perso-Arabic script for Sorani to IPA. This system relied on the following components:

- A lookup table including all of the possible mappings from Sorani orthography to IPA (initially based on information from Wikipedia, Omniglot, the Unicode standard, etc)[6].
- A human linguist who interactively selected the "best" (phonotactically best-formed) outputs from the CRF-based component.
- The CRF implementation from `cdec` (Dyer et al., 2010).

An important additional set of features used were the distinctive articulatory (phonological) features corresponding to each IPA symbol. Due to extensive linguistic literature on these features, their characteristics are fairly well understood and superordinate groupings are well-established. (Proposals differ in details but are broadly similar.)

3.4. PanPhon

We developed a resource combining a database of articulatory features for IPA segments, a library of Python classes and functions for manipulating IPA representations and articulatory feature vectors, and a pair of utilities for manipulating IPA-feature databases. This resource is distributed as PanPhon.[7]

The core of PanPhon is a database of mappings between IPA single-letter bases and values for a widely used set of phonological features (all of which are defined in articulatory terms). Members of this feature set include [±coronal] (+ for sounds produced with the tip or blade of the tongue), [±nasal] (+ for sounds produced with nasal airflow), and [±voice] (+ for sounds accompanied by vibration of the vocal folds). The features are technically three-valued: plus (+), minus (−) and unspecified (0). However, unspecification is used sparingly in the database and 0 values can be safely recoded as + values. This base component of the database is stored as a CSV table. In contrast, the definitions of diacritics and modifiers are written as rules which are represented in a human-readable YAML file format. We took this approach for the following reasons:

- The semantics of diacritics and modifiers, in terms of features, is predictable.
- The bases to which a particular diacritics/modifiers can attach are predictable in terms of the bases' features.
- More than one diacritic/modifier may be affixed to a single base.

A Python script takes the database of IPA base letters and the collection of rules for diacritics and modifiers as inputs. It uses them to generate a comprehensive table of IPA segments—both simple (consisting of a single letter) and complex (consisting of a letter and one or more diacritics/modifiers—with their corresponding definitions in terms of articulatory features. A second (very simple) script validates Unicode IPA files (UTF-8 only) against this comprehensive table.

[5] Word-initially, /ɛ/ is expressed with a preceding *hamza*, which eliminates the ambiguity between word-initial /ɛ/ and word-initial /h/ (Thackston, 2006); we also encountered this form when /ɛ/ follows another vowel.

[6] During the course of the project, we developed an improved version of Unitran (Qian et al., 2010) which can be used to generate a lookup table of this type rapidly.

[7] This resource (PanPhon) will be made available through the ELRA Catalogue of LRs.

Additionally, PanPhon includes a Python library with numerous utility functions for manipulating articulatory feature vectors and a Python class for interacting with the comprehensive IPA feature database. This class includes methods for querying the database in various ways, for querying IPA segment inventories, for fixed width pattern matching based on articulatory features, for calculating the sonority of an IPA segment, and for implementing feature-based edit distance (both unweighted and weighted) between IPA strings.

The comprehensive database of feature definitions for IPA segments in PanPhon served as the source for the source of the universal segment source used in the IPA prediction process.

3.5. Experiments

For Kurmanji, Zazaki, and Tajik, conversion to IPA presents only a trivial challenge, so experimental conditions and results are reported for Sorani alone. Training data were generated for 244 instances (types) and the IPA predictor was tested on 402 instances. Additional factors were character-level language models derived from Kurmanji Kurdish and Tajik. Both LM languages have transparent orthographies that can be converted unambiguously to IPA. They differ, however, in their phylogenetic proximity to Sorani: Kurmanji is very close while Tajik is somewhat more distantly related. The Kurmanji and Tajik LMs were 4-gram language models and were implemented using SRILM (Stolcke, 2002). It was predicted that inclusion of the Kurmanji factor would increase performance more than the more remotely-related Tajik.

3.6. Results

The effects of these factors on accuracy and total character error rate (CER) are given in Table 3.

Our predictor, using only basic features, predicts IPA forms with better than chance accuracy. Adding the articulatory features improves the performance significantly. The most important articulatory features were major class, laryngeal, and place features. However, even without the articulatory features, a reliable IPA predictor can be built. Kurmanji fluency features have a significant impact on the performance of the predictor. However, even adding the somewhat distantly-related Tajik boosts performance significantly.[9]

4. Lexical matching and capitalization inference

The next step in the capitalization inference process involved taking the lexicon derived from these corpora and inferring initial, hypothesized matches between lexical items. (In this paper, we will illustrate this using Sorani and Kurmanji, but the same process was performed for each pair of languages.) This task is complicated, however, in that we do not know, *a priori*, the appropriate distance metric that would best match Sorani and Kurmanji forms. That is to say, there exists, abstractly, some (potentially very complex) distance metric that would match a Sorani word to the Kurmanji word to which it corresponds (either by cognatehood or common borrowing). If we knew this "perfect" metric (call it m^*), we could know which words correspond, and if we knew which words corresponded, we could approximate that perfect metric to some degree. In the absence of knowledge of either of these, however, we can at least start with a naïve metric (call it m_0) and iterate from that.

So, we began by taking an unweighted Levenshtein distance as our initial approximation of the distance function, and from this getting an initial hypothesis regarding Sorani-to-Kurmanji word correspondences. For example, by Levenshtein distance, the nearest Kurmanji neighbor to the Sorani word *bri:tanja* ("Britain") is the Kurmanji word *bri:tanija* (also "Britain").

In practice, calculating the distance between every Sorani and Kurmanji word is computationally prohibitive. In order to restrict the task to a reasonable run-time for an emergency situation, we made a simplifying assumption, that every viable Kurmanji correspondence to a Sorani word *w* will be within t edits of *w*.[10]

The reason for choosing some value t, higher than which we will not consider Kurmanji words to be viable candidates for correspondence, is because a Levenshtein automaton can find all word pairs in two lexica that are within t edits[11] of each other (Schulz and Mihov, 2002), in *O(m+n)* rather than *O(mn)* time. We constructed from our Sorani lexicon a Levenshtein trie automaton for distance t, in which a trie that recognizes each known word in Sorani exactly is augmented into a Levenshtein automaton by adding an additional t layers of nodes to represent paths that include up to t errors. Meanwhile, we constructed a Kurmanji trie as well, that recognizes each known Kurmanji word exactly. By traversing the intersection of these trees, we can efficiently find the list of $\langle w_{Sor}, w_{Kur} \rangle$ word pairs within t edits of each other.

From this, we take the nearest neighbor w_{Kur} for each w_{Sor}, and calculate the edits for this pair. For example, for the pair ⟨*bri:tanja*, *bri:tanija*⟩, the edits would be ⟨b,b⟩,

[8] For example, the features [±syllabic] and [±sonorant] are considered major class features here, while [±continuant] is a manner feature.

[9] Manual investigation of errors suggests, in addition, that some of the cases where the predictor generates the "wrong" output, that output is actually a linguistically acceptable pronunciation of the orthographic token, although not necessarily the same pronunciation or the same phonetic rendering as the one in the test set.

[10] That is to say, if we consider the "perfect" metric that we are seeking m^*, and our naïve initial metric m_0 (in this case, an unweighted Levenshtein distance), there exists some threshold constant t such that every nearest neighbor of a word w according to m^* is within t distance of w according to m_0. Since we do not know m^*, we cannot know t, and in any case, any practical value chosen for t in a time-critical situation will probably be well below the actual t. However, for any chosen value of t, there is at least some percentage of actual nearest neighbors of w that will be within t edits.

[11] For the purposes of this algorithm, "edits" are counted between IPA segments – e.g. *r* or *i:* or *d͡ʒ* – rather than between Unicode characters, so a change of *r* to *d͡ʒ* counts as a single edit rather than three.

Features	Description
Basic	Simple features relevant to IPA symbol translation rules; whether IPA symbols are consonants or vowels.
PanPhon	Phonological features (articulatory), grouped according to major[8] classes.
Kurmanji	Fluency features from Kurmanji derived from character-level 4-gram language models.
Tajik	Fluency features from Tajik derived from character-level 4-gram language models.

Table 2: Features used for IPA prediction

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

Table 3: IPA prediction results

⟨ɾ,ɾ⟩, ⟨i:,i:⟩, ⟨t,t⟩, ⟨a,a⟩, ⟨n,n⟩, ⟨∅,i⟩, ⟨j,j⟩, ⟨a,a⟩.[12]

From the aggregate edits from *all* best $\langle w_{Sor}, w_{Kur} \rangle$ pairs, we can derive a further metric m_1, by treating the cost of an edit between the Sorani character c_S and the Kurmanji character c_K as the positive log frequency which with that correspondence occurs when using our unweighted metric m_0:

$$m_1(\langle c_S, c_K \rangle) = \begin{cases} 0, \text{ if } c_S = c_K \\ -\log P(\langle c_S, c_K \rangle), \\ \quad \text{if } c_S \neq c_K \end{cases}$$

Using this new metric to discourage unlikely correspondences – that is, to encourage pairs with reasonable correspondences like ⟨e,ɛ⟩ and discourage pairs with unreasonable correspondences like ⟨p,ɛ⟩, we attempted to find Sorani-Kurmanji word pairs again, this time using a weighted Levenshtein algorithm with the cost function m_1.[13]

The final correspondent chosen is then taken as that word's surrogate for capitalization frequency. For example, if the Kurmanji word *amɾi:kaji:jɛkan* ("Americans") has a capitalization frequency of 1.0 (that is, it is capitalized 100% of the time in the Kurmanji text), the hypothesized corresponding Sorani word *ɛmɾi:ki:jɛkan* (also "Americans") is assigned an inferred capitalization frequency of 1.0. Since this assignment is the trivial case of a k-nearest-neighbors regression, we also performed experiments using higher values of k.

5. Results

The narrow question – how well can we predict the capitalization rates of Language X words given text in Language Y? – cannot be answered directly for Sorani, of course, because written Sorani does not utilize capitalization. This, as mentioned above, is among the reasons we included other Western Iranian languages in this sample, because all of these languages utilize capitalization.

There remains, however, the broader question as well. How well do these inferred resources support named-entity recognition or other tasks in which capitalization is a relevant feature, and how best to utilize them within a larger system? We do not yet know how the resources above correspond (if at all) to performance increases in named entity recognition or other tasks, and in particular what ranges of the error rates above would result in performance increases; this is a subject we are pursuing as ongoing research.

In Table 4, we can see the coverage (the percentage of L1 words for which correspondents were found) and the accuracy (the mean squared error between the L1 words' inferred capitalization rates and their actual capitalization rate in the original L1 text) of inferred capitalization rates between the different languages.

It is difficult to interpret these error rates in isolation, although we can observe that, as predicted, the error rates are better between the more closely-related languages than between the more distantly-related languages and, unsurprisingly, that inferring capitalization from a smaller corpus (i.e., from Zazaki) gets worse results. It is unsurprising, therefore, that we get the best coverage and error rate when inferring Zazaki capitalization from a Kurmanji text – that

[12] Not all such edits would be genuine Sorani-Kurmanji correspondences, of course, since not all word pairs found in the previous step are genuine Sorani-Kurmanji correspondences, but manual inspection of the collected edits showed that reasonable cross-linguistic correspondences like ⟨e,ɛ⟩ and ⟨a,ɛ⟩ overwhelmingly outnumber unlikely correspondences like ⟨p,ɛ⟩ and ⟨q,ɛ⟩.

[13] It is worth noting that performing further iterations on this process, in which we generate further metrics m_2, m_3, etc. based on the pairs generated by the previous step, did not result in overall lower error rates in the end.

L1	Word types in L1 text	L2	Word types in L2 text	t	Coverage	Error
Sorani	13,240	Kurmanji	124,089	3	83.47%	N/A
Sorani	13,240	Zazaki	26,305	3	61.72%	N/A
Sorani	13,240	Tajik	108,814	3	34.90%	N/A
Kurmanji	124,089	Zazaki	26,305	3	63.62%	0.294
Kurmanji	124,089	Tajik	108,814	3	31.80%	0.470
Zazaki	26,305	Kurmanji	124,089	3	**85.66**%	**0.255**
Zazaki	26,305	Tajik	108,814	3	46.09%	0.534
Tajik	108,814	Kurmanji	124,089	3	37.41%	0.459
Tajik	108,814	Zazaki	26,305	3	31.99%	0.467

Table 4: Results of inferring L1 capitalization rate from L2 text, for $t = 3$

is, when inferring capitalization from a much larger corpus in a closely-related language.

The Zazaki-from-Kurmanji error rates can probably be used as a rough surrogate for what the Sorani-from-Kurmanji error rates would be, were Sorani to use capitalization. The Zazaki text was, like the Sorani text, fairly short compared to the Kurmanji text. Meanwhile, although Zazaki is not quite as closely related to Kurmanji as Sorani is, the Zazaki texts we used were, as noted in §2., written using Kurmanji-style spelling conventions, making the Zazaki text more similar to the Kurmanji text than it would otherwise be.

The choice of threshold $t = 3$ for the experiments above depended largely on practical constraints; as mentioned above, these experiments are intended to simulate a component in a pipeline that must execute, in its entirety, within a limited timeframe. A choice of $t = 2$ gave similar overall error rates (as illustrated for Zazaki-to-Kurmanji experiments in Table 5), but simply did not provide correspondents for a third of Zazaki words – that is to say, it happened that a third of Zazaki words did not have any correspondents within two edits. Meanwhile, however, $t > 4$ led to significant slowdown without a gain in accuracy; there are so many Kurmanji forms within five or six edits of each Zazaki form that the subsequent derivation of the m_1 metric was hindered. While the first step of the correspondence calculator – the construction of the Levenshtein automaton and its cotraversal with the bridge language trie – runs in *O(m+n)* time, the subsequent step runs in *O(mn)* time where *n* is the mean number of bridge language words generated by the previous step. $t > 4$ leads to a sufficiently high n that the efficiency of using a Levenshtein automaton was rendered moot by the subsequent step.

As noted above, the capitalization inference procedure described above is the trivial case of a k-nearest-neighbors regression, leading to a further experiment: how does considering higher values for k effect the accuracy? As illustrated in Table 6, including additional neighbors leads to a significant improvement in the error rate. (Considering neighbors beyond the tenth-closest neighbor resulted in only marginal differences in results.)

6. Discussion

The capitalization error rates between different Western Iranian languages are as expected; more closely-related languages have correspondingly more coverage and (when knowable) less error, while more distantly-related languages have relatively poor coverage and error. Moreover, we see that relative corpus size is unsurprisingly a major factor in coverage and error as well.

However, it may seem counterintuitive that expanding the number of neighbors k would increase the accuracy so much in this particular task; unlike many regression tasks, this is a task in which we expect there to be one (or at most a few) genuine correspondent words, and that further neighbors beyond that will not be as viable surrogates for capitalization, any more than a random word might be. That is to say, once the system has chosen Kurmanji <loksemburg> as the closest corresponding word to Zazaki <luksembu:rg> (both "Luxembourg"), we would not expect including ten further neighbors, more distant from <luksembu:rg>, to serve as a better capitalization surrogate than <loksemburg> itself does.

Manual inspection of the produced gazetteers suggests a possible explanation for why increasing the number of neighbors k does not increase error by including additional spurious correspondents. Many of the capitalized words in both corpora are renderings of foreign names and places, and are not, in their inventory or phonotactics, similar to the native Western Iranian vocabulary. When the Zazaki rendering of a foreign name has any Kurmanji correspondents within the threshold t at all, they are often renderings of the same foreign name. Increasing the number of allowed neighbors thus does not tend towards introducing error.

On the other hand, more non-capitalized words are Western Iranian vocabulary and tend to have many neighbors within t. Since native, non-capitalized words are comparatively frequent, one of the main sources of overall error is Zazaki native, non-capitalized words being in spurious correspondence with capitalized Kurmanji words. Introducing additional neighbors (spurious or not) will on average move the result towards zero, which in these cases is the correct result.

That is to say, increasing the number of neighbors does not greatly increase false positives (because many capitalized words do not have additional neighbors within the threshold, and there are fewer capitalized words in any case), while greatly reducing false negatives (because uncapitalized words are more common, and many uncapitalized words tend to have many neighbors within the threshold, so adding more neighbors brings down the average capitalization rate).

The accuracy increase from the inclusion of more neighbors, therefore, is probably not due to collecting additional

L1	L2	t	Coverage	Error
Zazaki	Kurmanji	1	45.05%	0.227
Zazaki	Kurmanji	2	67.58%	0.253
Zazaki	Kurmanji	3	85.66%	0.255
Zazaki	Kurmanji	4	91.27%	0.257
Zazaki	Kurmanji	5	95.64%	0.259
Zazaki	Kurmanji	6	97.83%	0.258

Table 5: Results of inferring Zazaki capitalization from Kurmanji text, for different thresholds t

L1	L2	t	k	Error
Zazaki	Kurmanji	3	1	0.256
Zazaki	Kurmanji	3	2	0.207
Zazaki	Kurmanji	3	3	0.188
Zazaki	Kurmanji	3	4	0.180
Zazaki	Kurmanji	3	5	0.175
Zazaki	Kurmanji	3	**10**	**0.169**

Table 6: Results of inferring Zazaki capitalization from Kurmanji text, using k neighbors

good capitalization correspondences. However, number of neighboring words within a threshold may end up being a surrogate for phonological foreignness, and phonological foreignness a surrogate for capitalization. This suggests that investigating phonological foreignness directly – rather than indirectly, as here – may be a promising avenue for future experiments in capitalization prediction and low-resource NER feature engineering.

7. Acknowledgments

This work was sponsored by a grant from DARPA.

8. Bibliographical References

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 *Proc. ACL*.

Kyumars Sheykh Esmaili and Shahin Salavati. 2013. Sorani Kurdish versus Kurmanji Kurdish: An empirical comparison. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 300–305.

Kyumars Sheykh Esmaili, Shahin Salavati, Somayeh Yosefi, Donya Eliassi, Purya Aliabadi, Shownem Hakimi, and Asrin Mohammadi. 2013. Building a test collection for Sorani Kurdish. In *2013 ACS International Conference on Computer Systems and Applications (AICCSA)*, pages 1–7.

Gérard Gautier. 1998. Building a Kurdish language corpus: An overview of the technical problems. In *Proceedings of ICEMCO*.

M. Paul Lewis, Gary F. Simons, and Charles D. Fennig. 2015. Ethnologue: Languages of the world, Eighteenth edition.

Ting Qian, Kristy Hollingshead, Su-youn Yoon, Kyoungyoung Kim, and Richard Sproat. 2010. A Python toolkit for universal transliteration. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. European Language Resources Association (ELRA).

Klaus U. Schulz and Stoyan Mihov. 2002. Fast string correction with Levenshtein-automata. *International Journal of Document Analysis and Recognition*, 5(1):67–85.

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

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

Géraldine Walther and Benoît Sagot. 2010. Developing a large-scale lexicon for a less-resourced language. In *SaLTMiL's Workshop on Less-resourced Languages (LREC)*.