Low-Resource Autodiacritization of Abjads for Speech Keyword Search

Patrick Schone

U.S. Department of Defense
Ft. George G. Meade, Maryland, USA
pjs500@afterlife.ncsc.mil

10.21437/Interspeech.2006-254

Abstract

Keyword search in speech requires retrieval systems to know the pronunciation of keywords. Many languages of the world are either largely alphabetic or have pronouncing dictionaries so that deducing pronunciations at run-time is manageable. There are many under-resourced languages, though, with writing systems where only some of the vowels are represented in the orthography (i.e., "abjads"). The absence of vowels makes direct mapping of abjads to pronunciation non-trivial. We describe an automatic system for inferring pronunciations from abjadic languages which seamlessly integrates into an existing context-sensitive pronunciation generator that serves a language-universal keyword search system. We also identify Web resources and system performance for each of five abjadic languages: Arabic, Farsi, Hebrew, Pashto, and Urdu. We show that almost effortlessly, the system can learn new rules which increase pronunciation accuracies by as much as 31.2% relative.
Index Terms: phonetics, IPA, vowelization, autodiacritization

1. Introduction

In various types of speech technology, such as text-to-speech conversion, automatic speech recognition, and speech retrieval, there is a necessity for converting orthographic representations of words into phonetic equivalents. The process of converting words into phonetic equivalents is often referred to as *rule-based transliteration* (RBT). Many languages of the world are largely phonetic in their word spellings which means that the process of converting from orthographic to phonetic representations can be done with limited amounts of error using context-sensitive rules. Spanish, for example, can be almost completely pronounced using a table of rules. There are also other languages where pronunciation dictionaries are required in support of an RBT in order to ensure that the correct pronunciation is given. For example, Mandarin Chinese would be extremely hard to pronounce automatically without the use of a pronunciation dictionary; but fortunately, dictionaries do exist for Chinese and other highly-resourced languages (eg., see [1]).

Languages that use orthographies of Semitic origin present unique challenges to RBTs. These languages tend to pronounce the consonants as written (though some consonants may get doubled), but vowels are either omitted completely from the orthography or they share symbols with semi-vowels (such as /w/ and /j/). These languages, referred to as *abjads*, typically have the ability to annotate vowels and doubled consonants through the use of diacritics, but this is not standard practice for native users of the language. Some abjadic languages have significant resources available (such as in some dialects of Arabic) and various research efforts have been commited by others to automatically diacriticizing them (see [2], [3], [4], [5]). Nonetheless, the prospect of developing reasonable quality RBTs for the remaining under-resourced abjadic languages is dubious. It is conceivable to merely disregard vowels altogether, but the utility of such a practice depends on the application.

We have developed a keyword searching system which can query speech in potentially any language [6]. It uses universal phonetic recognition to convert speech streams into phonetic symbols. It does this for any language with varying accuracies depending on the information it knows about the language [7].

The system also employs a RBT which can convert many languages from their native orthographies into pronunciations. Our RBT uses context-sensitive rules (to be described later) coupled with pronunciation dictionaries. This RBT is useful not only for identifying the likely pronunciations of keywords, but it can also help enhance the phonetic predictions of the UPR. Since vowels are the phonetic units best recognized by phonetic recognizers like ours, it is essential that the vowelization of abjadic languages be either known or predicted well to ensure the highest accuracies in keyword search. Moreover, for our purposes, the vowelization process has to be made transparent to the RBT so that languages are handled uniformly.

We have developed a process whereby with a small amount of training material available from the Web, the system can start with a basic transliteration capability and can generate and learn those context-sensitive rules which would allow it to optimize itself. This process performs a one-pass transformation from orthography to phonetic representation. The notion of learning rules automatically has strong similarities to transformation-based learning (TBL) [8] of pronunciations, but TBL applies rules in sequence and performs many passes of transformation.

We here describe our existing RBT and the rule-learning process. We also identify the existence and processing techniques of limited amounts of Web materials in five separate languages (namely Arabic [9], Farsi [10], Hebrew [11], Pashto [12], and Urdu [13]) which can be used to deduce the vowels and doubled consonants of the these languages. Lastly, we illustrate our system's performance under various conditions.

2. The RBT and the Rule-Learning Process

Before we can describe the process for automatic rule-learning, we first provide a description of our existing RBT as well as the mechanism for applying the rules. We then illustrate the rule-learning paradigm and later, how it performs on various data.

2.1. Description of the Existing RBT

As was mentioned before, our existing rule-based transliterator is designed to automatically transliterate potentially any

language. It maps the original orthographic information into International Phonetic Association (IPA) representation (possibly to include allophones as well as phonemes). Much of the phonetic information of the RBT is extracted from such websites as RosettaProject.org [14] and Omniglot.com [15]. This system couples both pronunciation dictionaries and context-sensitive rules for handling various kinds of languages. Yet for the processes in this paper, we exclude any available pronunciation dictionaries from the system so that it can focus exclusively on its ability to pronounce unseen words.

Our RBT is a greedy, left-to-right algorithm which is based on context-sensitive string matching. For a given IPA symbol and language, the RBT stores the collection of rules, if any, that can produce that given IPA symbol in the specified language. These strings can employ any lower-case character of the native orthography along with left and right context information (set off by curly braces "{" and "}") and any of three special anchor symbols representing word beginnings (^), word endings (\$), and single-phoneme insertion (\~). Other symbols are not relevant to the current discussion.

For example, in English, the symbol "s" typically makes the sound /s/. However, the larger rule "sh" tends to produce the sound /ʃ/. Note, though, that a "s" as the pluralizer at the end of a word ending in "es" can actually make the sound /z/. These three rules could appear in the RBT respectively as

IPA=**s** english[s]
IPA=ʃ english[sh]
IPA=**z** english[{e}s{\$}]

where the last case means an "s" which is preceded by the character "e" and which is at the end of the word. The intercharacter "\~" does not have much application in English unless a user wanted to account for the insertion of some sound such as the /p/ which in sometimes inserted in the word "something," /s-^-m-**p**-Θ-I-ŋ/. In such a case, the rule would be

IPA=**p** english[{m\~e,m}\~{t\~h}]

which indicates that "…eth…" or "…th…" may insert a /p/.

When the system applies its rules, it starts at the leading character of the word (the start symbol, ^) and applies the longest contextual rule that incorporates that symbol. It continues to advance until it has transliterated the entire word.

2.2. Learning Potential New Rules for the RBT

Given the constraints of the RBT, the next issue is to determine how to generate rules which can serve to provide missing vowel and doubled consonant information in abjad languages. These rules must be able to integrate seamlessly into the existing RBT structure based upon the RBT's transliteration paradigm.

We begin rule-learning by first seeding the RBT table with the base pronunciations for each character if the language (e.g., ب→/b/, ن→/n/, etc.). This helps enormously in constraining the rules that can be produced. These base pronunciations are used by the rule generator as a means of estimating pronunciations for substrings of the script form of the word.

The rule-learning process first encapsulates the script form with a preceding "^" and a following "\$" to represent word bounds. The system then performs alignment between the encapsulated string and the true pronunciation using minimum edit distance (MED). This alignment treats as perfect those places where putative pronunciations of substrings in question match the true pronunciation in comparison. MED also gives partial weight to vowels and semivowels being misaligned and to vowels being deleted. All other misalignments are counted as errors. Table 1 illustrates alignment for "دبي" (normally written from right to left) with a pronunciation, /a-b-a-d-i:-j-u:/.

Table 1: *Alignment between script and a pronunciation*.

	Ø	^	a	b	a	d	i:	j	u:	\$
Ø	**0**	1	1.25	2.25	2.5	3.5	3.75	4.75	5	6
^	1	**0**	**0.25**	1.25	1.5	2.5	2.75	3.75	4	5
ب	2	1	1.25	**0.25**	**0.5**	1.5	1.75	2.75	4	4
د	3	2	2.25	1.25	1.5	**0.5**	0.75	1.75	2	3
ي	4	3	3.25	2.25	2.5	1.5	**0.6**	**0.75**	**1**	2
\$	5	4	4.25	3.25	3.5	2.5	1.6	1.75	2	**1**

The MED alignments are retained in string form. Phonetic insertions are marked by inserting an intercharacter symbol "\~" into the script form, and deletions are denoted by inserting a null phone /Ø/ into the phonetic stream. For the above, the alignment was ^\~ب\~ي\~\~\$ matching ^a-b-a-d-i:-j-u:-\$. We null pad the alignment in those places where script characters are not continuguous to an intercharacter symbol; this gives us the strings ^\~ب\~د\~ي\~\~\$ matching ^a-b-a-d-Ø-i:-j-u:-\$. The presence of nulls in the phonetic stream suggests that the /i:/ is either derived from the multicharacter unit "د\~" or from "ي\~". (Note that this is an imperfect alignment, but we expect rule frequencies will help overcome faulty rules.)

Next, for every script-to-pronunciation alignment, the system retains counts for the mapping between every character (and known multicharacters as described) and the phonetic representations derived from that particular alignment. It also identifies nine contexts for each of those alignments (three characters each to the right and to the left of the unit). For example, from the alignment presented above, the system would retain counts for matches such as ب→b, {\~}ب→b, {^\~}ب→b, ب{\~}→b, {\~}ب{\~}→b, {^\~}ب{\~}→b, ب{\~ي}→b, {\~}ب{\~ي}→b, and {^\~}ب{\~ي}→b and so forth. The system purges rules that are not allowed within the context of the RBT (such as a ^ followed by \~) and it eliminates rules that are merely more complicated forms of simpler rules (such as ب{\~}→b which simplifies in the RBT to ب→b).

Finally, all rules are counted and prioritized first by frequency (higher frequencies first) and then by size (smaller rules first). Later rules are purged if they convey information that would never be used since earlier rules take precedence. The resulting list forms the set of rules which the system will try to test, one rule after another. Those rules that increase prediction performance are retained and all others are discarded.

3. Finding Data for Low-resource Languages

In order to evaluate these rules, we must first obtain data. As was mentioned previously, various organizations [1] have collected or acquired numerous speech and text corpora in various dialects of colloquial and formal Arabic and to a lesser extent in other languages. For the sake of this effort, we seek to demonstrate that Web resources alone, even though scant, can provide definite boosts to RBT performance in languages with abjads. We have identified sources of Web data for each of the five languages of this study, namely Arabic, Persian/Farsi, modern Hebrew, Pashto, and Urdu. We also indicate how we

could exploit these scant resources to build an appropriate corpus for the learning of diacritization rules.

3.1. Modern Standard Arabic (MSA)

The Arabic language has many dialects, some of which are not mutually intelligible. There are roughly 220 million speakers world wide according to Ethnologue.com [16]. MSA is the language used in the news media and is a standard form of communication which is largely intellible across most forms of Arabic. According to Wikipedia.org, there is little difference between the diacriticization process of MSA and classical Arabic [17]. Thus, as other researchers have done (such as [5]), we chose to use portions of the Arabic version of the Bible as a source of diacriticized text. Our RBT was given, as a starting point, the basic pronunciations of MSA along with the interpretations of diacritics. Automatically-generated pronunciations based on those diacritics (with some fixes by hand) were treated as truth data, and the rule-learning process was given a fractional lexicon of 2659 words (22086 sounds) with an even mix of diacriticized and non-diacriticized words. Already-diacriticized words were included in the learning process so that the RBT did not begin to override its capability to handle diacritics, but would attempt to jointly optimize.

3.2. Farsi (Persian)

Farsi (Persian) is a primary language of Iran and is spoken by roughly 22 million people worldwide. Farsi data exists at places like the LDC [1], but it is typically not diacriticized. In order to be able to learn appropriate Persian vowelization, it is important to be able to know how the native script is presented.

Fortunately, UniPers.com [10] presented a solution to this problem. UniPers.com is commited to the notion of overcoming reading difficulties of Persian by introducing a romanized version of the language. The romanization, interestingly, corresponds very highly to that of LDC. UniPers.com provides several pages of information illustrating their goals which they represent in both English, Persian script, and their UniPers romanization. We were able to align three of the four represented Web pages to provide the mapping between undiacriticed Persian and the phonetic-like romanization. This provided us with a word list of 8918 non-diacriticized Persian words (56294 phones) and their corresponding pronunciations.

3.3. (Modern) Hebrew

Modern Hebrew is spoken worldwide by approximately 8 million people [15]. The Web has a wealth of material describing ancient Hebrew and the vowelization processes thereof. However unlike with Arabic, the classic language of Hebrew differs substantially from the modern form; so Biblical representations of the language are not adequate for representing current Hebrew. Most other websites in Hebrew or about Hebrew omit vowelization. However, the National Center for the Hebrew Language has a website devoted to the Hebrew word of the day [11]. The definitions for the daily words are diacriticized. For instance, one daily word was אַבָטָלָה which can be transliterated directly into /a-v-t-a-l-a-h/ using the existing RBT. We provide a subset of these words and their corresponding non-diacriticized forms when unique to our learning system. This constitutes 3051 words (19519 phones).

3.4. Pashto

Pashto is an extremely under-resourced language, but it is a primary language of Afghanistan and is spoken by 25-30 million people [15]. Despite limited resources, that does not mean there is no available information regarding Pashto. In 1955, Penzl developed a small dictionary of approximately 1250 words of Pashto which he created using his own romanization. The dictionary is currently available of the Web [12], as are the phonetic interpretations for Penzl's romanized symbols. This means we can obtain pronunciations for the words, but unfortunately, this does not provide us with the actual script representation. In order to actually have script-to-pronunciation truth data for training, we must try to transliterate from the pronunciation back to original script.

The reverse mapping from pronunciations to script is not a one-to-one process. For some sounds such as /t/, there can be as many as four separate characters. We generate all potential script forms for each pronounciation. Although most of these forms are bogus, we can obtain some arbitrary, undiacriticized Pashto text from the Web as a means of trying to isolate the original script form of the language. When one of the putative scripts matches the exact form of a word identified from the arbitrary Pashto text, we determine this to be the original form. Using this process, we were able to identify what we believe to be the appropriate script form for 638 of the unique words. Although this corpus is on the small side, it can be extremely useful for learning the vowelization process because it does constitute 2674 phonetic symbols.

3.5. Urdu

Urdu is a language spoken in such places as the Indian subcontinent, and it spoken by over 60 million people. Like Pashto, resources for Urdu are scarce. Yet, as with Hebrew, websites such as UrduWord.com [13] exist which are devoted to presenting readers with words of the day. The Urdu words for the day illustrate translations into English and romanized forms of the Urdu, and several years worth of words are available (with some word reuse). We used the romanizations to generate pronunciations, and from those, distilled out 385 unique words from this data representing 2650 phonetic symbols.

4. System Evaluation

We now illustrate the performance of the RBT rule-learner by applying the system to each of the data sets previously mentioned. For each language, we compute the phonetic (vice word) accuracy as it applies its rules to the data. In order to provide specific detail to evaluation, we also chart (see Figure 1) its performance improvements over time in one of the languages, namely Modern Standard Arabic. We also provide Table 2 which shows, for each of the five languages, their starting accuracies, final accuracies after several thousand iterations are performed, and relative improvements (differences in accuracy divided by the starting accuracies). Bear in mind that the data sets are not all "created equal." In Pashto, Urdu, and Farsi, we did not have available the diacriticized forms of words when we began the learning process. Therefore, the performance illustrated there represents performance only on non-diacriticized forms. In the other languages, the system had various amounts of pre-diacriticized materials which will yield a

higher baseline, but will tend to have a lesser amount of relative growth. It is also important to comment that when romanization is provided as opposed to diacritization, the odds are better that the truth data generated thereby will be more accurate. Even so, the key values of interest across each of these sets is the relative improvement in phonetic accuracy between the start and end of evaluation, and the final accuracy obtained by the system.

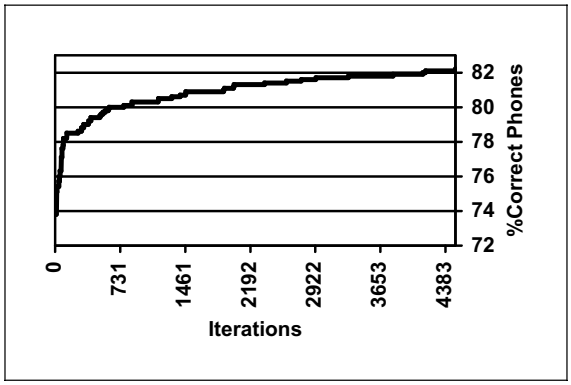

Figure 1 *MSA Phonetic Accuracy Improvements*

From Figure 1, we see that by having the RBT select only those rules that improve its accuracy when applied to MSA, we get a steady, logarithmic gain. When the system began executing, the RBT could only tag the half-diacriticized/half-not data with 73.8% phonetic accuracy. At the conclusion of the process, which took approximately 18-20 hours, the system had improved 11.4% relative to reach an end accuracy of 82.2%. The beauty of this automatic process is that these rules constitute permanent improvements to the existing RBT which were obtained over night with no human intervention other than the initial seed.

When we apply the rule-learner to the other abjadic languages of our evaluation, we observe comparable performance to what we had seen with MSA (which is included in the table as well). Those languages which had a mix of diacriticized data at the onset are identified by asterisks.

Table 2. *Start-to-finish accuracy improvements per language*

Language	Start Accuracy	End Accuracy	Relative Improvement
*MSA	73.8%	82.2%	11.4%
Farsi	71.5%	79.3%	10.9%
*Hebrew	69.8%	82.6%	18.3%
Pashto	63.5%	78.9%	24.3%
Urdu	56.8%	74.2%	31.3%

Diacriticized data sets were able to attain higher accuracies than those without diacritics, but even the non-diacriticized corpora were able to reach accuracies that were meaningful. State-of-the-art in abjad vowelization yields accuracies that are somewhat higher than these, but many of those processes make use of large, fully-annotated corpora; part of speech and syntactic information; probabilities; and multiple transformation passes. We on the other hand were able to make use of scant Web resources to obtain demonstrable improvements that operate well within the scope of our context-sensitive RBT.

5. Summary

We have demonstrated a system which can automatically learn pronouncing rules for abjadic languages whose orthography usually exclude annotation of vowels and doubled consonants. Furthermore, we described how these rules could be automatically developed so as to integrate perfectly into an existing context-sensitive rule-based transliterator. We were able to demonstrate the existence of small amounts of Web data from each of five abjadic languages: Arabic, Farsi, Hebrew, Pashto, and Urdu. Most importantly, we were able to illustrate that our rule-learner was able to take fairly poor RBT in these languages and was able, using even the scant Web resources, to provide significant relative improvements (as much as 31.3%) to the task of identifying the correct phonetic representations.

6. References

[1] Linguistic Data Consortium, http://ldc.upenn.edu
[2] K. Kirchhoff, J. Bilmes, S. Das, N. Duta, M. Egan, G. Ji, F. He, J. Henderson, D. Liu, M. Noamany, P. Schone, R. Schwartz and D. Vergyri, "Novel Approaches to Arabic Speech Recognition: Report from the 2002 Johns-Hopkins Workshop", ICASSP-2003, Hong Kong, April 2003
[3] D. Vergyri, K. Kirchhoff, "Automatic Diacritization of Arabic for Acoustic Modeling in Speech Recognition," Comp. Approaches to Arabic Script-based Languages Wkshp., COLING, Geneva, 2004.
[4] S. Hussain, "Letter-to-sound Conversion for Urdu Test-to-speech System", Computational Approaches to Arabic Script-based Languages Wkshp., COLING, Geneva, 2004.
[5] Y. Gal, "An HMM Approach to Vowel Restoration in Arabic and Hebrew", Wkshp. On Comp. Approaches to Semitic Languages, ACL, Philadelphia, 2002, pp. 27-33
[6] Schone, P., McNamee, P., Morris, G., Ciany, G., Lewis, S. "Searching Conversational Telephone Speech in Any of the World's Languages", International Conference on Intelligence Analysis. McLean, VA., 2005
[7] Walker, B., Lackey, B., Muller, J., Schone, P., "Language-reconfigurable universal phone recognition", EUROSPEECH-2003, Geneva, Swtz., 2003, pp. 153-156.
[8] Brill, E. "Some advances in transformation-based part-of-speech tagging." Proc. of the Twelfth National Conference on Artificial Intelligence, 1994, pp.722-727.
[9] The Arabic Bible
[10] UniPers.com, "UniPers: A New Alphabet for Persian," URL: "http://www.unipers.com/"
[11] National Center for the Hebrew Language, "Daily Dictionary", URL: "http://www.ivrit.org"
[12] Penzl, H. A grammar of Pashto: A descriptive study of the dialect of Kandahar, Afghanistan, American Council of Learned Societies, Washington, 1955, pp. 154-165, URL: http://www.yorku.ca/twainweb/troberts/pashto/lexicon.html by T. Roberts.
[13] UrduWord.com, "English-Urdu Dictionary: Word of the Day", http://www.urduword.com/wotd.php
[14] The Rosetta Project, http://www.rosettaproject.com
[15] Omniglot.com, http://www.omniglot.com
[16] Ethnologue.com, http://www.ethnologue.com.
[17] Wikipedia.org, "Modern Standard arabic: Pronunciation" en.wikipedia.org/wiki/Modern_Standard_Arabic, 4/7/2006.