How to Pronounce Hebrew Names

Alon Itai and Gai Shaked
Computer Science Department
Technion, Haifa, Israel

June 2, 2010

1 Introduction

This paper addresses the problem of determining the correct pronunciation of people's names written in Hebrew, by extracting clues from the way the same name is written in other languages, and by using a database of names whose pronunciation is known to guess the correct pronunciation of a given name.

Names differs from other words in a language because they do not follow the language's fixed set of rules. Names are not necessarily composed of the morphemes of the language and its inflections. Names may have different origins, while some names are legitimate word forms of the language, others originate in different languages.

A naive approach to this problem might be

> *Collect a list of all the names in the language, and manually attach each of them with a description of the correct pronunciation.*

However, names seem to obey the long tail concept – in each population there is a set of common and widespread names, but it seems the majority of the names are pretty unique and have few occurrences. Thus this approach might work, but it requires exhausting laborious work and has no significant contribution to similar problems. Moreover, new names are constantly being introduced, and the naive approach will not help for such names.

Two features of the modern Hebrew makes the problem of determining the correct pronunciation of names written in Hebrew especially interesting. First, the consonants are written in the form of letters while the vowels written as diatritics (dots–niqqud, ניקוד). The common modern Hebrew script, "ktiv male", omits the niqqud so a word is actually a sequence of consonants and one must be familiar with the word and the context in order to add the right vowels and pronounce it correctly. Second, a large portion of the population in Israel, are immigrants, or descendants of immigrants. This fact creates a great variety of names, from a large variety of cultures and languages.

However, the problem is not unique to Hebrew, nor is it restricted to languages written without vowels (e.g., Arabic, Farsi, Urdu). Even for languages written in a Latin script, names originating in one language retain the spelling in another language, but the pronunciation depends strongly on the original language. For example - notice the difference in pronouncing ***Charles****, Prince of Wales* and ***Charles*** *de Gaulle.* The methods and results described in this paper were tested and fitted to the specific problem of pronouncing Hebrew names, but by with minor changes they can be used for other languages as well.

2 Related Work

There is a large body of work on English names pronunciation, mainly for text to speech purposes [1, 2, 3]. Most of the work is based on large scale dictionaries of names as a training set and use various algorithms in order to guess the pronouncing of other names. Estimations [1] about the order of magnitude of unique names in English reach up to about two million different surnames, with much more than 100,000 first names.

The distribution of names is also different than the distribution of common words, the 141 most common words cover 50% of the non-names tokens in an English corpus [4] while for names in the United States 2300 of most common surnames names are needed for the same cover. While there is no large scale data for Hebrew, there have been some work on covering English names [4], the 50,000 most common names in English covers only about 70% of name tokens.

There is some work on automatically identifying pairs of transliterations [5, 6]. In section 5 we will show how we use such pairs in order to determine the correct pronunciation.

3 Pronunciation of Hebrew

Modern Hebrew text has several features which makes it hard to determine the correct pronunciation:

- The common script omits the niqqud.
- Some letters may indicate a missing vowel (e.g. Vav indicates O or U, Yod indicates I), but these letters may also indicate consonants (Vav - V or W, Yod - Y).
- Some letters may represent two different consonants - both B and V can be represented by Bet (ב), both P and F by Pe (פ) etc.

In this paper we will regard pronunciation as adding the niqqud signs, or dots, that resolve the above ambiguities. Namely, we will add letters to indicate A, E, I, O or U vowels, and for each of the letters - Bet (ב), Kaf (כ), Pe (פ) and Shin (ש) we will denote which of the two possible consonants it represents. The task of adding the niqqud signs which represents the correct pronunciation will be referred to as *dotting*, and a name with these niqqud signs will be called as *dotted.* We will use the following tables to represent Hebrew letters and vowels using Latin letters and few special characters, consonants are rendered by uppercase letters or special characters while vowels and consonants disambiguation (niqqud) marked by lowercase, boldface, letters.

4 The Agglutinating Algorithm

Although there is great number of different names form different origins, it seems that many of them were created by adding prefixes or suffixes to a common name (e.g. Avrahami - אברהמי - ^BRHMY is created by adding the suffix Y to the common name Avraham). While the prefix or suffix depends on the origin of the name, the method of agglutinating two common words, or a parts of them, holds in many different cultures and languages, and therefore for different names.

We would like to take advantage of this feature by using a relatively small list of manually dotted names in order to find the correct dotting for common prefixes and suffixes, and use them to dot names that do not appear in the original list. In order to do so we created a trie containing all the

Letters

א	^	ו	W	כ	K	ע	&	ש	Š
ב	B	ז	Z	ל	L	פ	P	ת	T
ג	G	ח	X	מ	M	צ	C		
ד	D	ט	Ṭ	נ	N	ק	Q		
ה	H	י	Y	ס	S	ר	R		

Vowels

A	**a**
E	**e**
I	**i**
O	**o**
U	**u**

Disambiguation

ב	b	B**b**	ב	v	B**v**
פ	p	P**p**	פ	f	P**f**
כ	k	K**k**	כ	x	K**x**
ש	s	Š**s**	ש	sh	Š**h**

Table 1: The transcription of Hebrew letters and pronunciation

Hebrew	Undotted	Dotted
אבי	^BY	^**a**B**vi**Y
אברהם	^BRHM	^**a**B**v**R**a**H**a**M
אבנר	^BNR	^**a**B**v**N**e**R
עמיחי	&MYXY	&**a**M**i**YX**a**Y

Table 2: An example of 4 Hebrew first names

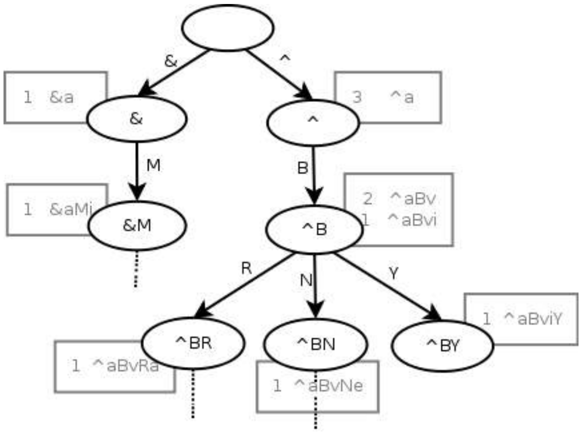

Figure 1: The trie resulting from the names of table 2.

dotted names. Each node in the trie holds a list of dotting variations for the prefix it represents, and a counter for the occurrence of each dotting.

For example, a list of manually dotted names as depicted in table 2 will result in the trie portrayed on figure 1.

We then reverse each dotted name and insert it to another trie, to create a similar trie of suffixes. When we get a new, undotted name, we search for the longest matching dotted prefix and the longest matching suffix. Given these tries and an undotted name we perform the following algorithm:

- The parts of prefix and suffix which do not overlap are dotted according to the most common dotting in the trie.

- If the prefix and suffix overlap – a new list of possible dottings for the overlap only is created from a union of the relevant parts of the prefix and suffix lists, each possible dotting in this list has a counter which is the sum of the counters in both lists. The dotting with the highest counter in this list is chosen for the overlapped section.
- If the prefix and suffix do not overlap, but leave a few uncovered letters in the center of the name - the algorithm is applied recursively to the middle part of the name together with one letter from the prefix and one from the suffix[1]. The dotting of the intersecting letters is determined as mentioned above.

Finally, we apply a few Hebrew specific fixes to the chosen dotting:

- If an O vowel appears twice in a row – once before a Vav and once after a Vav, we omit the first O. This situation happens because the letter Vav usually stands for a part of the vowel O or U and not the consonant V or W. However, the vowel O can also appear without Vav. Some names (e.g. Jacob - יעקב - Y&KB) are frequently written without Vav, and rarely with Vav (יעקוב - Y&KWB), in these cases the most common prefix would likely be "Y**a**&**a**K**o**" while the longest suffix would be "W**o**B" so the output would be "Y**a**&**a**K**o**W**o**B"- the vowel O appears twice for the same consonant, and that should be fixed.
- If an O vowel appears before a Vav followed by the vowel U - the vowel O is omitted. The vowel U is also commonly represented by a Vav, and therefore this situation is similar to the previous one, however less common.
- If the last letter in the name is Yod, and the previous letter has no vowel attached to it – the vowel I is added to the previous letter. The vowel I is usually followed by the letter Yod, and the letter Yod at the end of a name is a strong evidence that the name should end with the vowel I. The above algorithm sometimes misses that vowel because of very common names (e.g. Avraham - אברהם - ^BRHM) that appear in the dotted list while a small disruption creates a new name (אברהמי - ^BRHMY). The prefix "^BRHM" is likely to appear in the dotted list more than the suffix "MY" or "HMY" - thus the intersection will be dotted according to it - "^**a**B**v**R**a**H**a**M" and not "M**i**Y" as it should. The letter Y with no I vowel at the end of a name is extremely rare - so this is a good change in most of the cases.
- If the last letter was dotted by the vowels I, A or E – this vowel is omitted, since the vowel I at the end of a word is followed by a Yod, and A or E vowel at the end of a word is followed by a He (ה).

This simple algorithm achieves pretty good results, as will be discussed in section 6, but it is still strongly based on a list of manually dotted names. Next, we will present a method to automatically determine the correct dotting of a Hebrew name based on the transliteration of the same name to English.

[1] In case both the longest prefix and suffix are one letter long - the algorithm fails.

5 The Transliteration Algorithm

When translating a text from one language to another most of the words change completely. Names, on the other hand, are transliterated rather than translated, therefore the pronunciation of the name in both languages is similar or even identical. Given the same name both in Hebrew and in English, we take advantage of this fact (together with the fact that in English, as opposed to Hebrew, vowels are letters) to conclude the correct pronunciation of a Hebrew name.

We use a rule-based algorithm to match Hebrew and English consonants, and than use the vowels in the English name to add the appropriate dotting, representing the pronunciation, to the Hebrew name. We match each Hebrew letter with the English letters which denote similar phoneme. This allows us to align the English vowels to the Hebrew consonants, decide the right pronunciation for ambiguous Hebrew letters (e.g. Hebrew ב - Bet can be regarded as "B**b**" if aligned to an English "b"or a "B**v**" if aligned to a "v"). The consonants' alignment also helps us to make sure that the English and Hebrew words indeed represent the same name, or the same sequence of phonemes.

Once the consonants are aligned we use the English vowels to deduce the correct dotting for the Hebrew name. Once again we use a rather simple rule-based algorithm, which matches a Hebrew vowel for each sequence of English vowels.

For example, consider the name *Even* . The Hebrew form is ^BN (אבן). This example demonstrates both the lack of vowels and the ambiguity of letters in Hebrew. We first align the consonants in Hebrew and in English:

$$\begin{array}{cccc} E & v & e & n \\ \updownarrow & \updownarrow & & \updownarrow \\ \hat{} & B & & N \end{array}$$

The letters א (^) and ע (&) do not have an equivalent in English, they are usually represented only by a vowel, thus the E on "*Even*" is both aligned to the letter ^ and is used to determine its correct dotting. Finally, the algorithm adds the dotting:

$$\begin{array}{cccccc} & E & & v & e & n \\ & \updownarrow & & \updownarrow & \updownarrow & \\ \hat{} & \mathbf{e} & B & \mathbf{v} & \mathbf{e} & N \end{array}$$

Given a list of pairs of English and Hebrew names the above algorithm can automatically create a list of dotted Hebrew names, which can be used for the agglutinated names algorithm to dot even larger lists. However, the algorithm fails on names that, for some reason, are pronounced differently in Hebrew and in English. Such names include biblical names, which have gone through many changes throughout the centuries, one example is *Abraham* which in Hebrew pronounced *Avraham* (and written ^BRHM (אברהם)), the algorithm will therefore disambiguate the letter Bet incorrectly. As we will seen in section 6, this problem decreases the accuracy of the algorithm only slightly.

The last problem is how to create the list of such pairs. Luckily, some databases which include records of people names in Hebrew include the English form of the name as well. As an example we can look at a university's students list or almost any commercial client list. Since such databases usually contain private and confidential, they are not freely available. Another, available, source for such list is Wikipedia. The English version of Wikipedia contains[2] more than 3 Million articles, while the Hebrew version contains over 100,000. There is a link between the English and Hebrew versions of each article. We also take advantage of the fact that each article is labelled by categories

[2] As for March 2010

Training set	Records	First Names	Last Names
Dotted First Names	525	83.8%	27.6%
Dotted Last Names	1,046	62.7%	54.2%

Table 4: The success rate of the Agglutinating algorithm with manually dotted names.

Training set	Records	Transliteration algorithm First Names	Transliteration algorithm Last Names	Combined algorithm First Names	Combined algorithm Last Names
Names from Wikipedia	5,358[3]	75.9%	38.8%	86.1%	56.9%
Technion students	75,997[4]	62.2%	62.8%	76.0%	84.0%

Table 5: The success rate of both algorithms when using bilingual lists of names.

to search all the articles in the meta category "People" on the Hebrew wikipedia, and list any article name that has an English equivalent.

6 Evaluation

In order to test our algorithms we took 1000 random records from the Israeli phone book, the names in the records were manually dotted and divided into two test corpora – one for first names and one for surnames. There is a total of 1338 first names (476 unique), and 1002 surnames (770 unique). We used three sources of dotted names to test the performance of the algorithms:

- Another set of random names from the phonebook, manually dotted and divided into first and last names.
- A list of article names from Wikipedia, all the articles in the category *people* in the Hebrew Wikipedia which have an equivalent article in English. This list was dotted using the English-Hebrew algorithm described above.
- A list of last names of the Technion students, this list contains both Hebrew and English last names of the students in the Technion–Israel Institute of Technology, throughout the years. This list was also dotted by the English-Hebrew algorithm.

Table 4 summarizes the results of the first algorithm, applied with the manually dotted lists, on both test corpora. For each corpora and each list we present the percent of correct dotting.

Table 5 summarizes the results of both the algorithms, applied with the lists of English and Hebrew names. First we present the results obtained using the second algorithm alone. We also tested a combined algorithm: we used the dotted names generated by the transliteration algorithm as a training set for the agglutinated algorithm.

Finally, we have used the combined algorithm on all of the source lists together, and obtained a success rate of 93.4% for first names, and 87.1% for surnames. First names are clearly easier to dot, they tend to be shorter and they are less diverse than surnames. The results of the algorithm are better with the size of the lists of dotted names supplied to it. For each of the dotted lists, the percent of accurate dottings as a function of the number of dotted names in the lists roughly fits a log curve, while the parameters of the curve depends on the source of the dotted names and the test corpus (first names or surnames).

7 Conclusions and Further Research

The problem of determining the right pronunciation of a name is especially interesting on languages written without vowels (Hebrew, Arabic) but exists also in many other languages. The agglutinating algorithm can be applied for many other languages, requiring only little adjustments for the specific exceptional in each language. Using transliterations of a name into a third language can also be used in any other pair of languages, if a set of rules for aligning consonants and concluding vowels can be supplied. Russian-English version of the algorithm, for example, could tell the difference between the pronunciation of ***Charles*** (*Шарль*) *de Gaulle and* ***Charles*** (*Чарльз*), *Prince of Wales.*

One may further improve the results by using Machine Learning techniques to determine which of the dottings to choose when addressed with conflicting overlap data.

Finally, we have not discussed the problem of stress: determining which syllable is stressed. First there is a discrepancy between the location of the stress in normative and colloquial Hebrew (IlAn vs. Ilan, ShlomO vs. ShlOmo). Native Hebrew speakers agree on the stress of most names which are derived from other word by adding a suffix (ShmuEli, ShkEdi). They even agree on the stress of names with non Hebrew suffixes (e.g. ShmuelOvich, RabinOvich). We did not apply our methods since we were not able to obtain a database of names that includes the stress. We hope to obtain such a database and test our methods.

References

[1] Murray F. Spiegel "Proper Name Pronunciations for Speech Technology Applications", *International Journal of Speech Technology 6,* 419-427, 2003.

[2] Bechet, F., de Mori, R., and Subsol, G. "Dynamic generation of proper name pronunciations for directory assistance", *ICASSP. Orlando, FL, vol. I,* pp 745–748, 2002.

[3] Ngan, J., Ganapathiraju, A., and Picone, J. "Improved surname pronunciations using decision trees", *ICSLP. Sydney, Australia,* paper 653, 1998.

[4] M.Y. Liberman and K.W. Church. "Text analysis and word pronunciation in text-to-speech synthesis", *S. Furui and M. Sondhi, editors, Advances in Speech Signal Processing*, 1991.

[5] Yoon, S.Y., Kim, K.Y., Sproat, R. "Multilingual transliteration using feature based phonetic method", *Proc. of ACL*, 2007.

[6] Klementiev, A., Roth, D. "Named entity transliteration and discovery from multilingual comparable corpora", *Proc. of NAACL*, 2006.