A NEW PROGRAM FOR HEBREW INDEX BASED ON THE PHONEMIC SCRIPT

Uzzi Ornan
Dept. of Hebrew Language
Hebrew University
Jerusalem, Israel 91905

Michael Katz
Computer Science Department,
Technion
Haifa, Israel 32000

ornan@cs.technion.ac.il

1 Introduction

[1] Preparing an index or concordance for an analytic language in which each word has only one semantic factor is a simple task easily automated in a computer system.

However, most natural languages are rather synthetic, i.e., they have other words which contain more than one semantic factor. The English word *'DOGS'* for example has two factors. We may refer to these words as complex. Usually one of the factors is lexical while the rest are grammatical.

A linguist may use lists of 'complex words' according to their grammatical factor. For example, according to the plural indication. He or she may even be interested in derivative words that have a common constituent, such as *ject* in *subject*, *project* or *inject*, where each word has an independent lexical entry.

However, in most cases we would like to have lists of words, including 'complex words', which are gathered in groups according to their common lexical factor. This process is known as "Lemmatization".

Hebrew is clearly a synthetic language. The noun, besides having a suffix for the plural, may also have a suffix to indicate the possessor. Different suffixes are used for the singular and the plural. Thus, there are 24 different forms for each noun, while a verb has 29 regular forms.

[1] This research is partially supported by the Ministry of Science and Technology, via the Technion Research and Development Foundation, Contract No. 120-831.

Furthermore, a special form of an inflected verb may sometimes include another suffix which refers to the object of the action. This suffix changes according to number, person and gender. The sum of all forms related to the same lexical entry of a verb may thus reach 170!

However, the real obstacle in obtaining an efficient index program for Hebrew does not lie in these big numbers. Many languages have similar, or even more difficult problems of morphology. The main problem is the Hebrew writing system.

2 Deficiencies of Hebrew Writing System

Four shortcomings of the Hebrew writing system cause it to resist the conventional methods of compiling programs to prepare a complete and accurate index (or concordance) of Hebrew texts.

2.1 Many Vowels are not Written

The first deficiency is that not all vowels are written. For example, an *a* or an *e* inside a word are not indicated by any letter. Thus

פסל - may indicate both /pasal/‘disqualified’ and /pesl/‘statue’ [2]

חברה - /xevra/‘company’, ‘society’, /xvera/ ‘friend’ (f), or /xabbrah/ ‘connect her’.

In many cases *i* inside a word is not written, e.g.,

גזרים - /gzarim/ ‘carrots’, /gizrim/ ‘cut (f) them’.

פרסים - /parsim/ ‘Persians’ , /prasim/ ‘prizes’ , /pirsim/ ‘slice (f) them’

2.2 Attached Particles

A second deficiency is that short particles, such as the definite article *ha-*, the connective *w-* (= ‘and’), the subordinator *$e* - (= ‘that’, ‘which’) and many short prepositions (*l* - = ‘to’, *b* = ‘in’, ‘with’, *mi* - = ‘from’, etc.), are all written with no space before the following word. The result is that strings of letters may represent one word or a combination of a particle and a word or even two, three or more particles and a word. Thus,

מכתב - may be either /miktab/ ‘letter’, or /mi-ktab/ ‘from handwriting’.

ברכות - may be either /brakot/ ‘blessings’, or /b-rakkut/ ‘with softness’.

מהמר - may be either /mhammer/ ‘gambler’, or /mi-ha-marr/ ‘from the bitter’.

[2] for the phonemic signs, see appendix.

2.3 Double Consonant - One Letter

A double consonant is written using one letter only exactly as one consonant (Ornan-91). Thus

פסל - may indicate /pasal/ ‘disqualified’, or /passal/ ‘sculptor’.

גמל - may be either /gamal/ ‘rewarded’, or /gamall/ ‘camel’, or /gammal/ ‘camel driver’.

2.4 Ambiguous Characters

Several Hebrew letters are ambiguous. For example, ה may represent either a consonant /h/ or a vowel (at the end of a word). The same applies to י and ו .

Furthermore, all of them, i.e., ו, י, ה may indicate several vowels,

ה - at the end of a word may be /h/, /a/ or /e/;

ו - may be /w/, /o/ or /u/

י - may be /y/, /i/, /e/ or even /a/!

3 Existing Index-Programs for Hebrew

3.1 Difficulties prevent Progress

These features of written Hebrew cause great difficulties when compiling efficient programs for indexing a Hebrew text. Projects such as compiling dictionaries, building centers for information retrieval, working on machine translation or developing a mechanism for text-to-speech generation, can hardly develop under the regular linguistic conditions.

3.2 An Example

Consider the string אל . It may be read either as *âl* (= ‘not’, imperative construct) or *êl* (= ‘to’, ‘towards’). If we prepare a concordance of a Hebrew text which contains both words in several occurrences, all of them will be collected by the computer into the same entry. In *any* case, the user will receive many redundant citations as in the following:

אל תגידו בגת - II Sam, 1,20 - tell it *not* in Gath [3]

ואל תבשרו בחוצות אשקלון A proclaim it *not* in the streets of Ashkelon [4]

אל אשר תלכי אלך - Ruth 1,16 - *where* you go, I will go

[3] The examples are taken from the Bible. English translations are from a modern American Bible, except the last one, which is of King James’ version.

[4] ו, ‘and’, is written with the following word אל , without blank between them.

אל דמי לכם - Jes, 62,6 - take *no* rest

וישע יהוה אל הבל - Gen 4,4 - Jehovah had respect *unto* Abel

ואל מנחתו - and *to* his offering.

3.3 Manual Intervention

The list apparently includes all cases. Therefore, the *completeness test* has been passed. What about the *accuracy test?* One solution is a manual intervention as follows:

The list is copied and each copy is devoted to a separate entry: One list should contain only references of *âl*, while the other will have all references of *êl*. A person manually checks each citation of the first record in order to delete any *êl* and then, from the other record he or she deletes every citation in which אל should be read as *âl*. At the end of this work, admittedly, two accurate records are achieved. However, you cannot allow free search in the document itself. The index or concordance should be prepared in advance. Furthermore, why do we need manual intervention when we use a computer.

3.4 Non-Consecutive Sequence of Letters

A special strategy, introduced by a commercial company (Pinkas-85), improves the results. They developed a procedure which enables the user to search for a sequence of letters which are not necessarily consecutive. When the query is e.g. *ש*מ*ר* , the program will give all cases of words like שמורה, משמרת, שמירה, שימר, שמר , and many others.

This strategy seems to overcome the need to use a command such as 'PICK HEADWORD' of the OCP (Hockey-88), since most cases of the inflection of a word will be collected. However, it is not a *complete* and *accurate* solution.

1. In many cases some forms of the inflection do not include every consonant. For example, *נ*פ*ל* will collect all forms of the verb in the past and present tenses but will fail to collect those forms in the future tense. In other words, the results are not *complete*.

2. On the other hand, many irrelevant cases will be collected. Thus, if you ask for פרש with no asterisks between the letters, i.e., פרש , you will receive the following words (among many others) -

פרש	‘horseman, knight’	(/parra$/)
פרש	‘excrement’	(/per$/)
פרשן	‘commentator’	(/par$an/)
פרשנות	‘commenting’	(/par$anut/)
הפרש	‘difference’	(/hepre$/)
הפרשה	‘setting aside’, ‘the affair’	(/hapra$a/, /ha-parra$a/).

If you put the asterisks between the letters you will receive not only the above cases (among others), but also the following:

פרישה	‘retirement’	(/pri$a/)
פירוש	‘explanation’	(/pirru$/)

It is easily seen that *accuracy* is not achieved.

3.5 Intersection of Words

In order to overcome the problem of accuracy the company suggests that an intersection of words will suffice. The user should always make a point of combining several words and never ask for only one word. In this way, it is assumed, the tendency of words to appear only in connection with other specific words will help eliminate redundant citations. However, if we take this advice, we may confront the other danger: *Incompleteness*. We might receive only part of the material in which we are interested, as some important material may still not be revealed when asking for intersections of words (Ornan-87).

Even long experience and clever handling of related words and collocations will not suffice to achieve both accuracy and completeness. It seems that both features can never be achieved in the regular Hebrew writing system.

In order to achieve both *accurate* and *complete* results without manual intervention, we must think of a strategy in which the input of each word contains all relevant information about its structure. Once a text is written in this way each word of the input is distinct from all other words, and processing of Hebrew becomes a completely different story (the same no doubt applies to other languages with a similar writing system, such as Arabic).

4 "Vocalized Hebrew"

Both accuracy and completeness could be achieved by using "vocalized (or "pointed") Hebrew" script as an input. However, most educated Hebrew speakers do not have sufficient knowledge in order to write it correctly. In fact, it is impractical since most printed material of post biblical Hebrew, and of course of Modern Hebrew, is in regular script which is not "pointed". Furthermore, pointed script includes much superfluous material. It is a kind of phonetic script which reflects pronunciation of the eighth century C.E. but has no value for the modern speaker. Another less crucial issue is that computing prefers linear data as input, while the dots and points of the pointed system are diacritic signs which are spread beneath the letters, above them and inside them. If we are ready to prepare a special input by inserting all these dots and small signs in the same line of the letters, the question is why do we not instead look for an easier solution. The method we propose is discussed in the following chapters.

5 Phonemic Script (PhS)

5.1 Phonemic Versus Phonetic

Our suggestion is that for the duration of the computing process, Hebrew texts should first be converted into a writing system which will reveal the structure of each word. This writing system is a *phonemic script*. Phonemic script aims at presenting the structure of the word. It relies on relevant features of each phoneme and does not take into account any accidental change due to its position in the word, or to the influences by other phonemes. Phonemic script should be prepared for each language separately. It should be distinguished from a phonetic script. The latter tries to reflect the succession of sounds of speech and pays no attention to structure. The most famous phonetic script is that of IPA, which is suggested for use as it is for all languages.

5.2 Need for Romanization

In phonemic script, every character should represent one and the same phoneme, and every phoneme should be represented by one and the same character, although a fixed combination of two or more letters, or a letter and a diacritic sign, may be used as "one character" to represent one phoneme.

The idea of using Roman characters for languages which use another alphabet is not new. Many efforts have been made in recent years by various Agencies of Standards to establish a common script to be used for preparing bibliographies, catalogues, historical or geographical texts, etc. of non-Latin scripts [5]. This activity is not intended to replace the national systems, but rather to provide a modern means for international communication for both men and machines. It is agreed that such a script should mainly consist of Roman characters.

5.3 Methods for Romanization of Scripts

Many documenis by the International Standard Organization, such as ISO-259-1984(E), which deals with Hebrew, describe principles of Romanization and explains the two main ways to achieve it - transliteration and transcription. The following question arises: Can we use them for our aim?

In *transliteration*, every character of the orignal script is converted into another character. This approach does not help us when we have to convert unpointed Hebrew, since no vowel will appear, the attached particles will still come with no blank following them, a gemination will be written with one character, and some letters will still be ambiguous.

In *transcription*, one converts the sounds of the word into signs or letters which have phonetic value for the reader, but which cannot reveal the structure of the original word and therefore the transcribed text is not reversible.

Instead, we suggest a method which reflects the theoretical structure of the word. Figure 1 should illuminate the relations of the three methods.

5.4 Phonemic Transcription

The sounds (phones) of a language are observational entities. One can describe them according to articulation or acoustic features. However, the phonemes are theoretical entities. Like every theoretical concept it is possible to perform each phoneme in several ways. The result of the performance is a phone - an observational entity.

When we write phonemes rather than phones, we theoretically include in them all sounds, without being dependent on the phonetic habits of one user or another, nor on the phonetic values of one Hebrew dialect or another, since our phonemic transcription, as every written language, is followed by *rules of reading*. Thus, having a phonemic text, we achieve all dialects of the language and in order to realize them it is enough to use slight variations of the reading rules.

[5] see bibliography.

6 Man versus Machine

Even though the regular Hebrew script has the above-mentioned deficiencies (section 2), a Hebrew speaker easily reads every Hebrew text: He or she can correctly "decipher" each string of letters, e.g. he/she will read פסל or חברה as the context demands. How do Hebrew speakers overcome the enormous difficulties?

They do it on the basis of their control of the grammar, on their acquaintance with the lexicon, on their knowledge of semantic connections, and by understanding the real world. When we enumerate their linguistic abilities in Hebrew we find that they read a word as if all its vowels are inserted in it, as if particles are separated from the following word, as if words with a gemination of a consonant are distinguished from a similar word without gemination, as if each ambiguous letter is correctly interpreted and also as if changes of the base do not disturb identification.

We want the computer to be able to do the same. Practically, we want the computer to be able to convert regular Hebrew script into Phonemic Script. Once we manage to achieve this, we shall achieve, among other results, a powerful device that enables us to prepare an index or a concordance.

7 Description of our System

7.1 A Modular Approach

Performance of this idea had to be done using a modular approach. We did not try to solve the many phases according to their logical order, but treated each of them independently, assuming that all previous phases had been solved.

7.2 Steps for Realization

In order to establish a working system we compiled the following:

1. A formulation of a Phonemic Script for Hebrew.

2. A program for conversion of PhS into regular Hebrew script.

3. A program for converting PhS into "pointed" Hebrew script.

   These two steps are needed in order to be sure that the PhS correctly reflects both regular (unvocalized) and vocalized script.

4. Establishing a complete computerized morphological system based on the phonemic script, which both analyzes and generates each Hebrew form.

5. Preparing a "phonemic lexicon" which accepts a Hebrew entry written in the regular script as an input and outputs a list of all possible phonemic words that can be interpreted from it.

6. A program which converts every regular Hebrew form (including inflections and attached particles) into all possible phonemic interpretations.

7. Pre-syntax devices which rule out improper phonemic interpretations.

8. A syntactic-semantic system which can analyze as well as generate every legal Hebrew sentence. The system heavily relies on a rich lexicon which includes subcategories, thematic roles and selectional restrictions. In many cases we added a preferred choice of words. This part includes verbal, nominal, adjectival and prepositional components.

8 An Index Based on PhS

8.1 A Simple Index

Equipped with these devices we may now use it for building a program for preparing an index for a text given in regular Hebrew script. Internally we work entirely with material written in phonemic script but both the input and output are written in the regular script.

Suppose for a moment that both our query and the text are written in Phonemic script. The task of indexing is thus very simple: each string of letters relates to one entry only and the main efforts are directed towards a good morphological analyzer. Indeed, we have already prepared a complete morphological analyzer for a Phonemic text. However, we want to treat regular Hebrew texts. An example is called for.

8.2 Stages of Simple Index

Let us assume that we have some regular Hebrew documents and we want to collect all occurrences of the word כלל. The following are the steps:

1. We type this word on the regular keyboard.

2. The program translates our word into all possible phonemic interpretations, i.e. *klal* (noun), *kalal* (verb). We are asked which interpretaion is our choice.

3. When we choose no.1, the phonemic word *klal* is transferred to the morphological program which generates the full inflection of the word, and which presents each form in a separate line followed by all necessary grammatical details. Figure 2 shows the product in the left column with the grammatical details in the right.

4. This list of forms is then converted into regular written Hebrew (section 7.2(2)), as shown in Figure 3.

5. Now each word of the Hebrew text is compared consecutively with each of the forms contained in the inflection list of Fig. 3. If we ask for several words, they are compared in turn. The procedure includes devices that admit cases where a particle or string of particles appear before the word.

8.3 Programs as Filters

The above results are still similar to other conventional programs which use regular Hebrew script since the index may include not only all occurrences of the searched word, but also many irrelevant cases. Suppose we have a Hebrew text in which the verb *kalal* (which is written also as כלל) appears. It will be included in the results! It means that the index is *complete* but not *accurate.* However, this is not the end of the story. We now show how we rule out irrelevant readings.
Let us take the following Hebrew short text as an example:
חזרה כללית עם תזמורת נעימה חזרה שם.
A Hebrew speaker will easily read it as -
xzara klalit &imm tizmort n&ima xazra $am.
Literally:
Rehearsal general with orchestra pleasant repeated there.
Which means:
'A general rehearsal with pleasant orchestra was conducted there for the second time'.

But these words, i.e. the written sequences of letters, may be read in some other ways giving other real words in Hebrew. Here we give a list of them.

חזרה - xzara, xazra

כללית - klalit

עם - &imm, &amm

תזמורת - tizmort

נעימה - n&ima

חזרה - xzara, xazra

שם - $am, $em, $am

some of the phonemic words that are written on the right side have each more than one grammatical status, e.g. /n&ima/ is both a noun ('melody' or 'tune') and an adjective in feminine ('pleaseant'); *$am* is both an adverb ('there') and a verb ('put'), while this verb is both present and past of the same person (3rd, singular).

Since we rely on the lexicon, we obtain only phonemic words which do exist in Hebrew. In the Hebrew language most words can be considered as the output of morphological processes. Thus, if we analyse the string כללית , i.e., if we take into consideration grammar only (without checking lists of existing words), we may think that it is the preposition *k-* which is followed by a certain adjective *lalit*.

But since we rely on the lexicon, non-existant "words" do not appear at all. We withhold a lot of irrelevant material, but to omit other irrelevant readings, we find that the lexicon is not sufficient. Many other readings of the same string of letters also need to be erased. One can realize it by looking at the several readings of the words included in the example above. See full list of them with the grammatical analysis in Fig. 4.

We have built a second filter. It checks the close neighborhood of each word. Thus, we check whether a particle may be used with the suggested word that follows. Let us look at the analysis of another word, e.g., הספר . It may be either -

הספר $ sepr ,a(bsolute),... masculine, singular,

הספר $ sepr ,c(onstruct),.. masculine, singular,

both cases preceded by the definite article *ha-*.

The second line analyses it as a construct state ("nismak"). However, a construct state cannot be preceded by a definite article *ha-*. This means that the analysis of this line is false, and therefore the line should be deleted. The same applies to cases of verbs which are preceded by the definite article. This second filter takes care of other cases where there is a contradiction between a possible reading and a preceding or a following word. The close neighborhood is the first step of a powerful "short context" devices, which we have developed, and which are included in this filter. It erases most of improper readings, as we can see by the results of the analyzed text given in figure 5.

After this filter, the number of suggested words in the example is reduced from 17 to 11.

8.4 A Third Filter

We are now able to activate the syntactic-semantic analyzer. As mentioned above, it is heavily based on a rich dictionary, which includes elements of the well-known theories of the thematic features. By activating this analyzer we select the proper reading from the several ”suggested words” [6].

This analyzer is indeed a third filter, since wherever certain morphological or contextual analysis does not match the rest of the sentence, it is abandoned by the syntax and this reading is ruled out.

This means that irrelevant phonemic readings of a Hebrew written word are not included in the output of the syntactic-semantic analyzer. Therefore, when the analysis is complete, we collect only the intended proper phonemic readings from the many phonemic interpretations of the original word. On the other hand, if the sentence is indeed ambiguous, all relevant readings appear in the output. Let us come back to the same example.

In Fig. 6 we give a full syntactic analysis of the sentence. The input for the analysis is the table of Fig.5. Since there are two possible syntactic interpretations for one word, the syntactic analyser must give *two* analyses of the sentence. Our main point here is the cases of the string חזרה: The noun /xzara/ חזרה (‘rehearsal’) which is written exactly as the verb /xazra/ חזרה in the regular script. Even when we want to retrieve the noun only - according to the regular indexing programs - including ours as it has been described before - the string חזרה will be retrieved twice, from both occurrences of the sentence of the example.

The first occurrence of this string should indeed be interpreted as /xzara/, a noun. But it has little to do with the second occurrence, /xazra/ - a verb, past tense, singular, feminine, the entry of which is /xazar/ ‘came back’, of the second occurrence. However, if we first activate the series of filters, the form /xzara/, which was included in the morphological results of the second occurrence, and managed to survive the previous filter, is now ruled out by the syntax. The analyzer has selected the reading /xzara/ for the first occurrence, while for the second occurrence it selected the reading *xazra* ! [7]

[6] It should be emphasized that in some parts our dictionary is still small, so we should choose our examples very carefully.

[7] The final result may now be the same sting written in phonemic script, i.e.:`xzara klalit &imm tizmort n&ima xazra $am`. How we may use this new version of the text is another story.

As mentioned above, the user may type the query in regular Hebrew script. The system will respond in printing all possible readings of the word, asking the user to indicate which of them is meant [8]. From now on the system will look for this phonemic reading only and will give these instances only where the Hebrew written word corresponds to the phonemic query.

8.5 Summary

To sum up, when a Hebrew text is to be indexed, each of its words (strings of letters) goes through the program that converts every regular Hebrew written form into possible legal phonemic readings. These readings are achieved after using two filters. Then the sentence is analyzed. The analysis is the third filter.

We began our investigation by looking at the simple possibility that both the query and the text are given in phonemic script (Section 8.1). This is not so straightforward when we begin the line of procedures not from the phonemic script but rather from the regular Hebrew script. The number of lines which are now delivered to the syntactic analyzer is higher. Therefore, the analyzer should rule out some readings. However, we achieve the accurate results since only lines containing the proper forms are chosen.

9 Conclusion

Our approach seems to succeed in producing an index which is both accurate and complete. It was achieved since we refused to accept the deficient Hebrew writing system as the basis for work. By the modular approach we managed to build the system as a collection of independent units which later became one system.

When we complete several details, especially when we have the complete syntactic-semantic lexicon, our system will be ready for other applications, such as Hebrew interface, machine translation, text-to-speach devices, and similar projects.

10 Bibliography

- ANSI-Z39.25 - 1975, American National Standard Romanization of Hebrew, Secretariat Council of National Library Associations, American National Standards Institute, Inc.

[8] Ornan-87, p. 24.

- Choueka, Yaacov (ed.), Computers in Literary and Linguistic Research (ALLC 1988 Conference), Champion-Slatkine, 1990.
- Draft International Standard ISO/DIS 8957, Information and Documentation - Hebrew alphabet coded character sets for bibliographic information interchange, ISO 1992.
- Hockey, Susan, Micro-OCP (Version 2.0), OUP 1988.
- Ornan, Uzzi, “Phonemic Script: A Central Vehicle for Processing Natural Languages - The Case of Hebrew”, Technical Report 88.180, 1986, IBM Center for Science and Technology, Haifa, Israel.
- Ornan, Uzzi, “Processing of Legal Texts in Hebrew”, Mishpatim Vol. 17 (1987) [in Hebrew].
- Ornan, Uzzi, “Program-Generator for OPC in a Language with Heavy Morphology”, Literary & Linguistic Computing, Journal of the ALLC, Vol. 4 (1989), 159-162.
- Ornan, Uzzi, “Machinery for Hebrew Word Formation”, Advances in Artificial Intelligence, Natural Language and Knowledge-Based Systems, ed. Martin C. Golumbic, 1990, 75-93.
- Ornan, Uzzi, “Theoretical Gemination in Israeli Hebrew”, Semitic Studies in Honor of Wolf Leslau, ed. Alan S. Kaye, Otto Harrassowitz, Wiesbaden, Vol. II, 1991, 1158-1168.
- Ornan, Uzzi, “Basic Concepts in ‘Romanization’ of Scripts”, Technical Report # 5, Computational Linguistics, Computer Science Dept., Technion, Haifa, Israel, March 1994.
- Pinkas, Gaddi, “A Linguistic System for Information Retrieval”, ma’ase choshev, Vol. 12 (1985) [in Hebrew].

11 Appendix

Each of the 22 Hebrew *consonantal phonemes* is transcribed in phonemic script by a single special sign. See the table that follows. For three phonemes which have been introduced to modern Hebrew, ג', ז', צ' , we use for the time being a combination of a letter followed by an apostrophe. ש' is also transcribed with an apostrophe, i.e. g’, z’, c’ and $’. See below for their pronunciation.

Phonemic	Regular	Phonemic	Regular	Phonemic	Regular
p	פ,ף	@	ט	^,‘	א
c	צ,ץ	y	י	b	ב
q	ק	k	כ,ך	g	ג
r	ר	l	ל	d	ד
$	ש	m	מ,ם	h	ה
t	ת	n	נ,ן	w	ו
		S	ס	z	ז
		&	ע	x	ח

Later we hope to achieve fonts which include other characters, especially with *hachek* for the g’,z’, c’, $’ and the ”left $in”.

The couples [b-v], [k-j], [p-f], as well as [g-g], [d-d], t-t] in certain traditional dialects, are each considered a single phoneme. Both members of each couple are written with the same letter but for foreign proper names one may use: *v* for *b* when it is pronounced as [v], *j* for *k* when it is pronounced as [x], *f* for *p* when it is pronounced as [f].

The following is the list of special signs introduced for Hebrew phonemes which are self-understood.

‘ or ˆ - is a glottal stop (can be omitted in speech)
x - is pronounced as a Scottish [ch]
@ - is another [t]
& - is another, fricative, glottal voice
c - is pronounced as a cluster of [ts] (e.g., cats)
q - is another [k]

$ - is sh
$’ - is another s
z’ - is the French [j]
g’ - is [g] in giant
c’ - is [tsh] (checkoslovakia)

(All other consonants are approximately equivalent in realization to the English ones.)

There are five vowels: *a,e,i,o,u.* A compound sign of *ei* is used to represent cases when the vowel *e* is written with a י after the letter which represents the consonant before the *e*. *ei* is considered as the sixth written vowel in phonemic script. Vowels are pronounced as in Italian or in Spanish. Length is not phonemic in Hebrew [9].

[9] See Ornan-94. This suggestion has been approved by a special committee nominated by The Standards Institute of Israel, and has been accepted by the International Standards Organization as a proposal in the framework of ”conversion of written languages into Roman script“ (ISO/TC 46 /SC2 N356).