Learning Morpho-Lexical Probabilities from an Untagged Corpus with an Application to Hebrew*

Moshe Levinger
Haifa Research Lab
IBM Science & Technology, Haifa, Israel

Uzzi Ornan
Computer Science Department
Technion, Haifa, Israel

Alon Itai
Computer Science Department
Technion, Haifa, Israel

This paper proposes a new approach for acquiring morpho-lexical probabilities from an untagged corpus. This approach demonstrates a way to extract very useful and non-trivial information from an untagged corpus, which otherwise would require laborious tagging of large corpora. The paper describes the use of these morpho-lexical probabilities as an information source for morphological disambiguation in Hebrew. The suggested method depends primarily on the following property: a lexical entry in Hebrew may have many different word forms, some of which are ambiguous while the others are not. Thus, the disambiguation of a given word can be achieved using other word forms of the same lexical entry. Even though it was originally devised and implemented for dealing with the morphological ambiguity problem in Hebrew, the basic idea can be extended and used to handle similar problems in other languages with rich morphology.

1. Introduction

This paper addresses the problem of morphological disambiguation in Hebrew by extracting statistical information from an untagged corpus. Yet, the primary point is not to propose a method for morphological disambiguation per se, but rather to suggest a method to compute morpho-lexical probabilities to be used as a linguistic source for morphological disambiguation. Let us start with a few definitions and terminology that will be used throughout this paper.

We consider written languages, and for the purpose of this paper, a **word** is a string of letters delimited by spaces or punctuation. Given a language L, and a word $w \in L$, we can find (manually or automatically by a **morphological analyzer** for L) all the possible morphological analyses of the word w. Suppose a word w has k different analyses, $A_1, \ldots, A_k$, will be used to denote these k analyses. A word is **morphologically ambiguous** if $k \geq 2$. The number and character of the analyses depend on the language model. We have used the definitions of the automatic morphological analyzer developed at the IBM Scientific Center, Haifa, Israel (Bentur and Segev, 1992).

Given a text T with n words: $w_1, \ldots, w_n$, for each morphologically ambiguous word $w_i \in T$, with k analyses: $A_1, \ldots, A_k$, there is one analysis[1], $A_r \in \{A_1, \ldots, A_k\}$ which is the **right analysis**, while all the other $k-1$ analyses of w are **wrong analyses**. The same word w_i in a different text, may have, of course, a different right analysis, thus, *right* and *wrong* in this case are meaningful only with respect to the context in which w_i appears.

* This research was partially supported by grant number 1380031 of the Israel Council for Research and Development.
1 We will assume that there is only one right analysis, although, in rare cases, there might be more than one.

Morphological disambiguation of a text T is done by indicating for each ambiguous word in T – which of its different analyses is the right one. At present, this can be done manually by a speaker of the language, and hopefully in the future it will be done automatically by a computer program. When dealing with *automatic* disambiguation of a text it is sometimes useful to **reduce** its ambiguity level. A reduction of the ambiguity level of an ambiguous word w, with k morphological analyses: $A_1, \ldots, A_k$, occurs when it is possible to select from $A_1, \ldots, A_k$, a proper subset of l analyses $1 \leq l < k$, such that the right analysis of w is one of these l analyses. In the case where $l = 1$, we say that the word w is **fully disambiguated**.

Since this paper suggests a method for morphological disambiguation using probabilities, the notion of **morpho-lexical probabilities** is also required. Our model of the language is based on a large fixed Hebrew corpus. For a word w with k analyses, $A_1, \ldots, A_k$, the morpho-lexical probability of A_i is the estimate of the conditional probability $P(A_i|w)$ from the given corpus, i.e.,

$$P_i = P(A_i|w) = \frac{no.\ of\ times\ A_i\ was\ the\ right\ analysis\ of\ w}{no.\ of\ occurrences\ of\ w} \quad .$$

Note that P_i is the probability that A_i is the right analysis of w *independently* of the context in which w appears. Since the word w has exactly k different analyses: $\sum_{i=1}^{k} P_i = \sum_{i=1}^{k} P(A_i|w) = 1$.

For reasons which will be elaborated in Section 2, our problem is most acute in Hebrew and some other languages (e.g., Arabic). Though ambiguity problems of a similar nature occur in other languages. One such problem is sense disambiguation. In the context of Machine Translation, Dagan and Itai (Dagan, Itai, and Schwall, 1991; Dagan and Itai, 1994) used corpora in the target language to resolve ambiguities in the source language. Yarowsky (Yarowsky, 1992) proposed a method for sense disambiguation using wide contexts. Part of speech tagging – deciding the correct part of speech in the current context of the sentence – has received major attention. Most successful methods have followed speech recognition systems (Jelinek, Mercer, and Roukos, 1992) and used large corpora to deduce the probability of each part of speech in the current context (usually the two previous words – trigrams). These methods have reported performance in the range 95–99% "correct" by word (DeRose, 1988; Cutting et al., 1992; Jelinek, Mercer, and Roukos, 1992; Kupiec, 1992). (The difference in performance is due to different evaluation methods, different tag sets and different corpora). See Church (Church, 1992) for a survey.

Our work did not use the trigram model, since due to the relatively free word order in Hebrew it was less promising, and also, in some cases the different choices are among words of the same part-of-speech category. Thus tagging for part of speech alone would not solve our problems. Note that a single morphological analysis may correspond to several senses. Even though each sense may have different behavior patterns, in practice this did not present a problem for our program.

The rest of this paper is organized as follows. Sections 2–4 include a description of the morphological ambiguity problem in Hebrew, followed by the claim that knowing the morpho-lexical probabilities of an ambiguous word can be very effective for automatic morphological disambiguation in Hebrew.

Then, in Sections 5–6, we present the key idea of this paper: How to acquire a good approximation for the morpho-lexical probabilities from an *untagged corpus*. Using this method we can find for each ambiguous word w with k analyses: $A_1, \ldots, A_k$, probabilities $\overline{P}_1, \ldots, \overline{P}_k$ which are an approximation to the morpho-lexical probabilities: $P_1, \ldots, P_k$.

In Section 7 we clarify some subtle aspects of the algorithm presented in Section 6 by looking at its application to several ambiguous words in Hebrew. A description of an experiment which serves in order to evaluate the approximated morpho-lexical probabilities which were calculated using an untagged corpus, will be given in Section 8.

Finally, in Section 9, a simple strategy for morphological disambiguation in Hebrew using morpho-lexical probabilities, will be described. This simple strategy was used in an experiment which was conducted in order to test the significance of the morpho-lexical probabilities as a basis for morphological disambiguation in Hebrew. The experiment shows that using our method we can significantly reduce the level of ambiguity in a Hebrew text.

no. of Analyses	1	2	3	4	5	6
no. of Word-Tokens	17,551	9,876	6,401	2,760	1,309	493
%	45.1	25.4	16.5	7.1	3.37	1.27

no. of Analyses	7	8	9	10	11	12	13
no. of Word-Tokens	337	134	10	18	1	3	5
%	0.87	0.34	0.02	0.05	0.002	0.007	0.01

Table 1: The dimension of morphological ambiguity in Hebrew

2. Morphological Ambiguity in Hebrew

Morphological ambiguity is a severe problem in modern Hebrew due to the dimension of this phenomenon. Thus, finding methods to reduce the morphological ambiguity in the language is a great challenge for researchers in the field, and for people who wish to develop natural language applications for Hebrew.

Table 1 demonstrates the dimension of the morphological ambiguity in Hebrew. The data was obtained by analyzing large texts, randomly chosen from the Hebrew press, consisting of nearly 40,000 word-tokens. According to this table, the average number of possible analyses per word-token was 2.1, while 55% of the word-tokens were morphologically ambiguous. The main reason for this amount of ambiguity is the standard writing system used in modern Hebrew (unpointed script). In this writing system not all the vowels are represented, several letters represent both consonant and different vowels, and gemination is not represented at all (Ornan, 1986; Ornan, 1991). The rich morphology of the language and the fact that many particles are attached to the word, forming a single string, further contribute to the morphological ambiguity.

In order to demonstrate the complexity of the problem, we should take a closer look at Hebrew morphology. A **morphological analysis** of a word in Hebrew should extract the following information:

- lexical entry.
- category.
- tense (for verbs only).
- attached particles (i.e., prepositions, connectives, determiners).
- status – a flag indicating whether a noun is in its construct or absolute form.
- gender, number and person (for nouns, adjectives, verbs etc.).
- gender, number and person of pronoun suffixes.

For example, the morphological analysis of the Hebrew string וכשראיתיו (written in a Latin transliteration[2] WK$R^YTYW) is as follows:

- lexical entry: R^H (ראה) – the verb 'to see'.
- category: verb.
- tense: past.
- attached particles: W + K$ (כש + ו) = 'and when'.
- gender: feminine/masculine, number: singular, person: 1-st person.
- object pronoun: masculine, singular, 3-rd person.

2 See Appendix A for the Hebrew–Latin transliteration.

Thus, WK$R^YTYW should be translated into English as: 'and when I saw him'.

To see the nature of the morphological ambiguity in Hebrew, consider, for example, the following string – HQPH (הקפה), which has three possible analyses:

1. The determiner H + the noun QPH (קפה + ה – 'the coffee').
2. The noun HQPH (הקפה – 'encirclement').
3. The noun HQP + the feminine possessive suffix H (ה + הקף – 'her perimeter').

The use of computers for morphological analysis of Hebrew words is nowadays well studied and understood. Several high-quality morphological analyzers for Hebrew have been developed in the last decade. One such morphological analyzer[3] was used to supply the input for the morphological disambiguation project described in this paper.

3. Former Approaches

Eliminating or reducing the ambiguity at this early stage of automatic processing of Hebrew is very crucial for the efficiency and the success rate of parsers and other natural language applications. It should be noted that the morphological ambiguity in Hebrew makes even "simple" applications – as is often considered when dealing with other languages – complicated.

One good example for this is **full-text retrieval systems** (Choueka, 1980). Such systems must handle the morphological ambiguity problem. To see that, consider, for example, the case where we look for all the texts with the word HQPH ('encirclement'). Without morphological disambiguation, we get many texts which really include the word H+QPH ('the coffee'), or even HQP+H ('her perimeter') (Ornan, 1987). Another application which is more difficult in Hebrew than in other languages is **text to speech systems**, which cannot be implemented in Hebrew without first solving the morphological ambiguity, since in many cases different analyses of a word imply different pronunciations. A much simpler problem occurs in English, where for some words the correct syntactic tag is necessary for pronunciation (Church, 1988).

The notion that this ambiguity problem in Hebrew is very complicated and that it can be dealt with only by using vast syntactic and semantic knowledge, has led researchers to look for solutions involving a considerable amount of human interaction.

Ornan (Ornan, 1986), for instance, developed a new writing system for Hebrew, called 'The Phonemic Script'. This script enables the user to write Hebrew texts which are morphologically unambiguous, in order to use them later as an input for various kinds of natural language applications. However, since regular Hebrew texts are not written in this script, they first must be transcribed to phonemic texts. Choueka and Lusignan (Choueka and Lusignan, 1985) presented a system for the morphological tagging of large texts which is based on the short context of the word but depends also heavily on human interaction.

Methods using the **short context** of a word in order to resolve ambiguity (usually categorical ambiguity) are very common in English and other languages (DeRose, 1988; Church, 1988; Karlsson, 1990). A system using this approach was developed by Levinger and Ornan in order to serve as a component in their project of morphological disambiguation in Hebrew (Levinger, 1992). The main resource, used by this system for disambiguation, is a set of **syntactic constraints** which were defined manually by the authors and which followed two theoretical works which defined short context rules for Hebrew (Pines, 1975; Albeck, 1992). The syntactic constraints approach, which is an extension of the short context approach, was found to be useful and reliable, but its applicability (the proportion of ambiguous words which were fully disambiguated) was very poor. Hence, the overall performance of this system is much less promising than in other languages. These results can be explained by the following properties of the ambiguity problem in Hebrew:

3 The morphological analyzer was developed at the IBM Scientific Center, Haifa, Israel (Bentur and Segev, 1992). We would like to thank the center for letting us use it for research purposes.

1. In many cases two or more alternative analyses share the same category and hence these alternatives satisfy the same syntactic constraints. Moreover, there are cases where two or even more analyses share exactly the same morphological attributes and differ only in their lexical entry. For instance, the word XLW (חלו) has two such morphological analyses:
   (a) The verb XLH (חלה), fem./masc., plural, 3-rd person, past tense ('they became ill').
   (b) The verb XL (חל), fem./masc., plural, 3-rd person, past tense ('they occurred').
2. The short context constraints use unambiguous anchors which are often function words such as determiners and prepositions. In English most such function words are unambiguous. In Hebrew, these words are almost always morphologically ambiguous. Moreover, many of them appear as prefixes of the word to be analyzed, and their identification is part of the morphological analysis. We thus have a circularity problem: In order to perform the morphological analysis, we need the short context, to identify the short context, we have to find anchors, but in order to find such words, we need first to perform the morphological analysis.
3. The word order in Hebrew is rather free.

4. Our Approach

The purpose of this paper is to suggest a new approach to deal with the above-mentioned problem. This approach provides highly useful data which can be used by systems for automatic, unsupervized morphological tagging of Hebrew texts. In order to justify and motivate our approach, we must first make the following conjecture:

> *Although the Hebrew language is highly ambiguous morphologically, it seems that in many cases a native speaker of the language can accurately "guess" the right analysis of a word, without even being exposed to the concrete context in which it appears. The accuracy can even be enhanced if the native speaker is told from which sub-language the ambiguous word was taken.*

If this conjecture is true, we can now suggest a simple strategy for automatic tagging of Hebrew texts:

> *For each ambiguous word – find the morpho-lexical probabilities of each possible analysis. If any of these analyses is substantially more frequent than the others choose it as the right analysis.*

As we have already noted, by saying morpho-lexical probabilities, we mean the probability of a given analysis to be the right analysis of a word, independently of the context in which it appears. It should be emphasized that having these morpho-lexical probabilities enables us not only to use them, rather naively in the above-mentioned strategy, but also to incorporate these probabilities into other systems which exploit higher level knowledge (syntactic, semantic etc.). Such a system which use the morpho-lexical probabilities together with a syntactic knowledge is described in (Levinger, 1992).

5. Acquiring the Probabilities

Adopting this approach leaves us with the problem of finding the morpho-lexical probabilities for the different analyses of every ambiguous word in the language. Since we use a large corpus for this purpose, the morpho-lexical probabilities we acquire must be considered relative to this specific training corpus.

One way to acquire morpho-lexical probabilities from a corpus is to use a large **tagged corpus**. Given a corpus in which every word is tagged with its right analysis, we can find the morpho-lexical probabilities as reflected in the corpus. This is done by simply counting for each analysis the number of times that it was the right analysis, and using these counters to calculate the probability of each analysis being the right one. The main drawback of this solution is the need for a very large tagged corpus. No such corpus exists for modern Hebrew. Moreover, for such a solution a separate **tagged corpus** is required for each domain. The method we are about to present saves us the laborious effort of tagging a large corpus, and enables us to find a good approximation to the morpho-lexical probabilities by learning about them from an *untagged corpus*. Using this method, one can easily move to a new domain by applying the method to a new *untagged* corpus suited to this new domain.

This might seem, at first sight, as an impossible mission. When we see the word HQPH in an untagged corpus we cannot automatically decide which of its possible readings is the right one. The key idea is to *shift* each of the analyses of an ambiguous word in such a way that they all become distinguishable. To be more specific, for each possible analysis (lexical entry + the morphological information), we define a set of words which we call *Similar Words* (*SW*). An element in this set is another word form of the same lexical entry which has similar morphological attributes as the given analysis. These words are assumed *similar* to the analysis in the sense that we expect them to have *relatively* the same frequency in the language as the analysis they belong to. A reasonable assumption of this kind would be, for instance, to say that the *masculine* form of a verb in a certain tense in Hebrew is expected to have relatively the same frequency as the *feminine* form of the same verb, in the same tense. This assumption holds for most of the Hebrew verbs, since all Hebrew nouns (and not only animate ones) have the gender attribute[4]. To see a concrete example, consider the word R^H (ראה) and one of its analyses: the verb 'to see', masculine, singular, 3-rd person, past form. A similar word for this analysis is the following one:

- R^TH (ראתה) – feminine, singular, 3-rd person, past tense.

The choice of which words should be included in the *SW* set of a given analysis is determined by a set of pre-defined rules based on the intuition of a native speaker. Nevertheless, the elements in the *SW* sets are not determined for each analysis separately, but rather are generated automatically, for each analysis, by changing the contents of one or a few morphological attributes in the morphological analysis. In the previous example the elements are generated by changing the contents of the *gender* attribute in the morphological analysis, while keeping all the other attributes unchanged.

The set of rules which are used by the algorithm for automatic generation of *SW* sets for each analysis in the language are of a heuristic nature. For the problem in Hebrew, a set of 10 rules[5] was sufficient for the generation of *SW* sets for *all* the possible morphological analyses in Hebrew. In case we wish to move to some other domain in Hebrew, we should be able to use the *same* set of rules but with a suitable training corpus. Hence, the set of rules are language-dependent but not domain-dependent. To clarify this point, consider the word MCBY& (מצביע) which has the following two morphological analyses:

1. The verb HCBY& (הצביע), masculine, singular, present tense ('indicates' or 'votes').
2. The noun MCBY& (מצביע – 'a pointer').

The set of rules defined for Hebrew would enable us to observe that in the domain of daily newspaper articles, the first analysis probably has a high morpho-lexical probability while the second analysis has a very low probability. Using the same set of rules, we should be able to deduce for a domain of articles dealing with computer languages that the second analysis is probably much more frequent than the first one. Whenever we wish to apply our method to some other language which has a similar ambiguity problem – all we need to do is to define a new set of rules for generation of *SW* sets in that other language.

4 This assumption does not hold for a small number of verbs that take as a subject only animate nouns with a specific gender, such as YLDH (ילדה – 'she gave birth'.)
5 See Appendix B for the list of the rules used for Hebrew.

By choosing the elements in the *SW* set carefully so that they meet the requirement of similarity, we can study the frequency of an analysis from the frequencies of the elements in its *SW* set. Note that we should choose the words for the *SW* sets such that they are morphologically unambiguous. We assume that this is the case in the following examples, and will return to this issue in the next two sections.

To illustrate the whole process, let us reconsider the ambiguous word HQPH (הקפה) and its three different analyses. The *SW* sets for each analysis is as follows:

- HQPH (הקפה – 'encirclement')
  SW_1 = { HHQPH (ההקפה – 'the encirclement') }
- H+QPH (קפה + ה – 'the coffee')
  SW_2 = { QPH (קפה – 'coffee') }
- HQP+H (ה + הקף – 'her perimeter')
  SW_3 = { HQPW (הקפו – 'his perimeter') , HQPM (הקפם – masculine 'their perimeter')
  HQPN (הקפן – feminine 'their perimeter') }.

Given the *SW* set of each analysis we can now find in the corpus how many times each word appears, calculate the expected frequency of each analysis and get the desired probabilities by normalizing the frequency distribution.

Had our similarity assumption been totally correct, namely, that each word in the *SW* set appears *exactly* the same number of times as the related analysis, we would have expected to get a neat situation such as the following (assuming that the ambiguous word HQPH appears 200 times in the corpus)[6]:

- SW_1 = { HHQPH = 18 }
- SW_2 = { QPH = 180 }
- SW_3 = { HQPW = 2, HQPM = 2, HQPN = 2 }.

These counters suggest that if we manually tagged the 200 occurrences of the string HQPH in the corpus, we would find that the first analysis of HQPH is the right one 18 times out of the 200 times that the word appears in the corpus, that the second analysis is the right one 180 times, and that the third analysis is the right analysis only twice.

Using these counters we can relate the following morpho-lexical probabilities to the three analyses of HQPH: 0.09, 0.90, 0.01, respectively. These probabilities must be considered as an *approximation* to the real morpho-lexical probabilities, due to the following reasons:

1. The words in the *SW* set are only expected to appear *approximately* the same number of times as the analysis they represent.
2. The reliability of the probabilities we acquire using our method depends on the number of times the ambiguous word appears in the corpus (which is really the size of the sample which we use to calculate the morpho-lexical probabilities).

In the corpus we worked with the word HQPH appeared 202 times and the number of occurrences of the words in its *SW* sets were as follows:

- SW_1 = { HHQPH = 3 }
- SW_2 = { QPH = 368 }

6 The numbers in this example are fictitious. They were chosen in order to clarify our point.

- $SW_3 = \{$ HQPW = 0, HQPM = 0, HQPN = 0 $\}$.

By applying now the algorithm of the next section on these counters, we can calculate the desired probabilities.

6. The Algorithm

Our algorithm has to handle the frequently occurring case where a certain word appears in more than one *SW* set. In that case, we would like to consider the counter of such a word appropriately. The algorithm takes care of this problem and works as follows:

- Initially we assume that the proportions between the different analyses are equal.
- For each analysis we compute its **average number of occurrences**, by summing up all the counters for each word in the *SW* set, and dividing this sum by the *SW* size. Note that in this stage we also include the ambiguous word in each of the *SW* sets[7].
- If a word appears in several *SW* sets, we calculate its contribution to the total sum according to the proportions between all those sets, using the proportions calculated in the previous iteration.
- Calculate the new proportions between the different analyses by computing the proportions between the **average number of occurrences** of each analysis.
- This process is iterated until the new proportions calculated are sufficiently close to the proportions calculated in the previous iteration.
- Finally, the proportions are normalized to obtain probabilities.

A formal description of the algorithm written in a pseudo-code is given in Figure 1.

Applying this algorithm to the sets and the counters extracted from the corpus (our previous example), yields the following probabilities[8]:

- HQPH = 0.0113
- H + QPH = 0.9870
- HQP + H = 0.0017.

Although this method for acquiring morpho-lexical probabilities gives very good results for many ambiguous words, as will be shown in the Section 8, we detected two types of inherently problematic cases:

1. Due to the high degree of morphological ambiguity in Hebrew, some of the words in the *SW* sets may also be ambiguous. As long as the other possible analyses of such a word are not too frequent, it only slightly affects the final probabilities. Otherwise, we might get wrong results by erroneously crediting the high number of occurrences of such a word[9] to one of the analyses. For this reason, we try to construct the *SW* sets from as many suitable elements as possible, in order to be able to detect "misleading" words of this sort.
2. Occasionally, the *SW* sets defined for two different analyses are actually the same. Thus, a differentiation between those two analyses cannot be done using our method.

---

7 This is done mainly in order to handle cases where a certain analysis has an empty *SW* set, since it does not have naturally similar words. The third example in the next section serves to clarify this point.

8 Due to the finite nature of our algorithm, we assign non-zero probabilities even to events that do not occur in the training corpus. This property agrees with common statistical practice (Agresti, 1990).

9 Due to technical reasons, we cannot decide whether a given word is ambiguous or not when we automatically generate the words for the *SW* sets. See Section 7 for more details.

Input:

w – A word with k analyses: $A_1, \cdots, A_k$.

$SW_1, \cdots, SW_k$ – The similar words sets of analyses $A_1, \cdots, A_k$.

sw – A word in some SW set.

$C(sw)$ – The number of occurrences of sw in the training corpus.

$Inc(sw)$ – A set of indexes representing the analyses for which sw is a member in their SW set, i.e., $Inc(sw) = \{l : 1 \leq l \leq k, sw \in SW_l\}$

ε – A prespecified threshold indicating the convergence of the algorithm.

Internal Variables:

P_j^i – The approximated morpho-lexical probability of A_j in the i-th iteration.

$SumAnal_j$ – The sum over the contribution of all the words in SW_j.

$AvgAnal_j$ – The average contribution of a single word in SW_j to $SumAnal_j$.

The Algorithm:

$P_1^0 := P_2^0 \cdots := P_k^0 := 1/k$;
i := 0;
repeat
    $i := i + 1$;
    for $j := 1$ **to** k **do begin**
        $SumAnal_j = \sum_{sw \in SW_j} C(sw) \times (P_j^{i-1} \;/\; \sum_{l \in Inc(sw)} P_l^{i-1})$;
        $AvgAnal_j := SumAnal_j \;/\; size(SW_j)$
    end;

    for $j := 1$ **to** k **do**
        $P_j^i := AvgAnal_j \;/\; (AvgAnal_1 + \cdots + AvgAnal_k)$

until ($\max_j |P_j^i - P_j^{i-1}| < \varepsilon$) .

Figure 1
Calculating the approximated morpho-lexical probabilities.

Another potentially problematic case is the **coverage problem**, which arises whenever we do not have enough data in the corpus for disambiguation of a certain word (see a discussion on this problem in (Dagan, Itai, and Schwall, 1991)). This problem was found to occur very rarely – for only 3% of the ambiguous words in our test-texts the counters found in the corpus were smaller than 20. We expect this percentage would be even smaller had we used a larger training corpus. For such words, we simply ignored the data and arbitrarily gave a uniform probability to all their analyses.

7. Examples

Several aspects of the algorithm described in the previous section can be better understood by looking at some clarifying examples. To see an example for the convergence of the algorithm, consider the neat situation described in Section 5 for the word HQPH:

- SW_1 = { HQPH = 200, HHQPH = 18 }
- SW_2 = { HQPH = 200, QPH = 180 }
- SW_3 = { HQPH = 200, HQPW = 2, HQPM = 2, HQPN = 2 }.

For these sets and counters and for $\varepsilon = 0.001$, the algorithm converges after 10 iterations. The probabilities for each iteration are given below:

- Iteration no. 1: $P_1 = 0.333$, $P_2 = 0.333$, $P_3 = 0.333$
- Iteration no. 2: $P_1 = 0.230$, $P_2 = 0.671$, $P_3 = 0.099$
- Iteration no. 3: $P_1 = 0.164$, $P_2 = 0.803$, $P_3 = 0.033$
- Iteration no. 4: $P_1 = 0.128$, $P_2 = 0.857$, $P_3 = 0.015$
- Iteration no. 5: $P_1 = 0.110$, $P_2 = 0.880$, $P_3 = 0.010$
- Iteration no. 6: $P_1 = 0.100$, $P_2 = 0.890$, $P_3 = 0.010$
- Iteration no. 7: $P_1 = 0.095$, $P_2 = 0.895$, $P_3 = 0.010$
- Iteration no. 8: $P_1 = 0.092$, $P_2 = 0.898$, $P_3 = 0.010$
- Iteration no. 9: $P_1 = 0.091$, $P_2 = 0.899$, $P_3 = 0.010$
- Iteration no. 10: $P_1 = 0.091$, $P_2 = 0.899$, $P_3 = 0.010$

In this example the similarity assumption holds, and the words in the *SW* sets (excluding the word HQPH itself) are also unambiguous. This need not hold in other situations.

As we have pointed out already, due to technical reasons we have not been able to apply the **morphological analyzer** to the words in the *SW* sets, and thus we have not been able to automatically observe that a given similar word is ambiguous by itself. The problem stems from the fact that we have been able to use the **morphological analyzer** on personal computers only, while both the corpus and the program which automatically generates the *SW* sets for each analysis could have been used only on our mainframe computer. Given this, the **morphological analyzer** was only used in order to obtain the input files for the disambiguation project.

Nonetheless, the fact that ambiguous words in the *SW* sets cannot be automatically identified does not affect the quality of the probabilities obtained by our method for most ambiguous words[10]. To see the reason for this, consider the word XWD$ (חודש) and its two analyses:

1. The noun XWD$ (חודש – 'a month') and its two analyses:
   SW_1 = { XWD$ = 2079, HXWD$ = 970 (החודש – 'the month') }

10 In our test sample of 53 words, the probabilities were significantly affected by this phenomenon in only 3 cases.

2. The verb XWD$, masculine, singular, 3-rd person, past tense ('he/it was resumed').
   SW_2 = { XWD$ = 2079, XWD$H = 41 (חודשה – 'she/it was resumed'),
   XWD$W = 57 (חודשו – 'they were resumed') }

Both XWD$H and XWD$W (SW_2) are ambiguous words. Still, since the counters for these two words are substantially smaller than the counter for the word HXWD$ (SW_1) – the probabilities calculated according to these counters can be considered as a reasonable approximation for the real morpho-lexical probabilities. The algorithm, applied to these sets and counters, yielded the following probabilities: $P_1 = 0.961$, $P_2 = 0.039$.

This kind of situation is not unique for the word XWD$. Similar situations occur in many other ambiguous words in Hebrew. Hence, not having the ability to identify ambiguous words in the *SW* sets has a meaningful effect on the quality of the probabilities only in cases where some similar word is ambiguous and its other analysis is frequent in the language. In such cases the analysis which this word belongs to is assigned a higher probability than its real morpho-lexical probability. We use the term **misleading words** for such ambiguous similar words.

A partial solution for such cases was implemented in the revised algorithm which we used for morpho-lexical probabilities calculation. In this revised version we automatically identified similar words as misleading words by looking at the counters of all the similar words in a given *SW* set. A word was considered misleading if its counter was at least five times greater than that of any other word in the set. This solution was not applicable in cases where *all* the similar words in a given *SW* set were misleading words.

The need to add the original ambiguous word to all the *SW* sets of its analyses can be made clear by the following example. Consider the word ^T (את) and its sets and counters as were found in our training corpus:

1. The direct object particle for definite nouns – ^T.
   SW_1 = { ^T = 197,501 }
2. The feminine, singular, 2-nd person, nominal personal pronoun ^T (feminine 'you').
   SW_2 = { ^T = 197,501, ^TH (אתה) = 1689, ^TM (אתם) = 891, ^TN (אתן) = 105 }
3. The noun ^T ('a spade').
   SW_3 = { ^T = 197,501, H^T (האת) = 0 }

The key point here is that the particle ^T has no natural similar word[11]. Yet, from the above counters we should be able to deduce that the first analysis has a very high morpho-lexical probability. This is since the ambiguous word ^T is very frequent in the corpus, while the counters in the *SW* sets for the second and third analyses indicate that these analyses are not the "reason" for the high frequency of ^T in the corpus.

Adding the ambiguous word to all the *SW* sets allows the algorithm to take this fact into account. Applying the algorithm on the above sets and counters yields the following morpho-lexical probabilities: $P_1 = 0.9954$, $P_2 = 0.0045$, $P_3 = 0.0001$.

8. Evaluating the Probabilities

Before we evaluate the quality of the approximated probabilities which can be acquired using our method, we would like to start with a definition of three terms that will be used in this section:

Morpho-Lexical Probabilities Estimated from a Training Corpus – given a large corpus in Hebrew the morpho-lexical probabilities of a given word are the probabilities of its analyses as calculated by manual tagging of all the occurrences of the given word in the corpus. We will use the abbreviation **morpho-lexical probabilities** to denote this term.

11 In fact, all the prepositions in the language lack natural similar words.

Morpho-Lexical Probabilities Estimated over a Test-Corpus – in order to avoid the laborious effort needed for manual tagging of all the occurrences of an ambiguous word in a large corpus, we estimate the morpho-lexical probabilities by calculating them from a relatively small corpus. The abbreviation **test-corpus probabilities** will be used for this term.

Approximated Probabilities – given an ambiguous word, the approximated probabilities of the word are the probabilities calculated using the method described in this paper.

The approximated probabilities obtained by our method were evaluated by comparing these probabilities with test-corpus probabilities obtained by manual tagging of a relatively small corpus. Since the approximation we acquire depends on the corpus we have been using – texts taken from the Hebrew newspaper *Ha'aretz*[12], we have to calculate the test-corpus probabilities from texts taken from the same source. For this purpose we used a small corpus consisting of more than 500,000 word-tokens taken from the same newspaper.

For our experiment we picked from this small corpus two kinds of test-groups. Test-group1 consisted of 30 ambiguous word-types which were chosen randomly from all the ambiguous word-types which appeared more than 100 times in the corpus. For the second test-group, test-group2, we randomly picked a short text from the corpus from which we extracted all the ambiguous word-tokens which appeared at least 30 times in the small corpus. This test-group consisted of 23 words.

These two test-groups are of a different nature. Test-group1 consists only of very frequent word-types in Hebrew, but the test-corpus probabilities for these word-types can be viewed as a reliable estimate of the morpho-lexical probabilities. The word-tokens in test-group2 better represent the typical ambiguous word in the language but their test-corpus probabilities were calculated from a relatively small sample of tagged words.

For each word in these test-groups we extracted from the small corpus all the sentences in which the ambiguous word appears. We then manually tagged each ambiguous word and found for each one of its analyses how many times it was the right analysis. For example, the word ^WLM (אולם) (taken from test-group1) has the following two morphological analyses:

1. The particle ^WLM ('but').
2. The noun ^WLM ('a hall').

The word ^WLM appeared 236 times in the small corpus. By manually tagging all the relevant sentences we found that the first analysis, 'but', was the right analysis 232 times, and the second analysis, 'a hall', was the right analysis only 4 times. Given these numbers we can calculate the **relative weights** of these two analyses: 232/236, 4/236 and the **test-corpus probabilities**: 0.983, 0.017, respectively. In the same way, using the small corpus we found the test-corpus probability, P_{test}, for each of the analyses in the test-groups.

12 We would like to thank *Ha'aretz* for the permission to use magnetic tapes from its archives.

Ambiguous Word	Approximated Probability	Relative Weight	Test-Corpus Probability
^WLM	0.968	232/236	0.983
(אולם)	0.032	4/236	0.017
^T	0.995	300/300	1.000
(את)	0.001	0/300	0.000
	0.004	0/300	0.000
XWD$	0.976	75/78	0.962
(חודש)	0.024	3/78	0.038
LPNY	0.725	100/100	1.000
(לפני)	0.274	0/100	0.000
	0.001	0/100	0.000
^LH	0.141	112/168	0.667
(אלה)	0.005	0/168	0.000
	0.001	0/168	0.000
	0.849	56/168	0.333
	0.001	0/168	0.000

Table 2: Approximated and test-corpus probabilities for five ambiguous words from the two test-groups

Table 2 shows the test-corpus probabilities and the approximated probabilities for five representative ambiguous words from our test-groups. In this table the approximation for the probabilities of the first three words is very good while the approximation for the fourth word is quantitatively poor, but still succeeds in identifying the first analysis of 'LPNY' (לפני – 'before') as the dominant analysis. As for the fifth word, here the approximation we got is totally incorrect. At the end of this section we shall identify some cases for which our method fails to find a reasonable approximation for the morpho-lexical probabilities of an ambiguous word.

In order to evaluate the quality of the approximation we got by our method, we should compare the approximated probabilities for the words in these test-groups with the test-corpus probabilities we found.

When we tried to make a quantitative comparison using statistical methods we found that for many analyses P_{app} "looks" like a good approximation for P_{test} but from a statistical point of view the approximation is not satisfying. The main reason for this is that the words in the SW set of a given analysis can be considered similar in their frequency to the analysis only from a qualitative point of view, and not from a quantitative one. Thus, the comparison we describe in what follows serves for evaluation of the *quality* of the approximated probabilities.

Motivated by the way we use the morpho-lexical probabilities for morphological disambiguation, we can divide the probability of an analysis into 3 categories:

1. **Very high probability** – an analysis with a probability from this category is the dominant analysis of the ambiguous word and thus, given we cannot use any other source of information to disambiguate the given word, we would like to select the dominant analysis as the right analysis.
2. **Very low probability** – given no other information, an analysis with a very low probability should be treated as a wrong analysis.

	Total	Good Approximation	Reasonable Approximation	Incorrect Approximation
no. of Words	30	29	0	1
%	100	97	0	3

Table 3: The quality of the approximation for test-group1

	Total	Good Approximation	Reasonable Approximation	Incorrect Approximation
no. of Words	23	17	2	4
%	100	74	9	17

Table 4: The quality of the approximation for test-group2

3. **All other probabilities** – an analysis with probability of this sort should not be selected as wrong/right analysis solely according to its morpho-lexical probability.

Formally, the mapping from the probability of an analysis to its category is done using two thresholds – **upper threshold** and **lower threshold**, as follows:

$$CAT(prob) = \begin{cases} 1 & \text{prob} \geq \text{upper threshold} \\ 2 & \text{prob} \leq \text{lower threshold} \\ 3 & \text{otherwise} \end{cases}$$

The quality of the approximated probabilities we acquire using our method is now measured by examining the proportion of words for which the estimated-category for **each** of their analyses agrees with the category defined by the approximated probabilities. The results of this comparison for the 2 test-groups we used is shown in Table 3 and Table 4. In these tables we divide the words into 3 groups according to the quality of the approximation found for them:

1. Words with good approximation – words for which $CAT(P_{test}) = CAT(P_{app})$ holds for all their analyses, using: **lower threshold** = 0.20, and **upper threshold** = 0.80. (The first three words in Table 2 belong to this category).
2. Words with reasonable approximation – words which do not fall into the previous category, but $CAT(P_{test}) = CAT(P_{app})$ holds for all their analyses, using: **lower threshold** = 0.35, and **upper threshold** = 0.65 (The fourth word in Table 2 belongs to this category).
3. Words with incorrect approximation – the words which their approximation is neither good nor reasonable. (The fifth word in Table 2 belongs to this category).

From these tables we can see that our method yielded incorrect approximation for only 5 words out of the 53 words in the test-groups (9.5%). By closely looking at these words, we can identify two reasons for failure:

1. Ambiguity of a word in the *SW* set of a given analysis. This may effect the probabilities calculated for this analysis. To see that, consider the word MWNH (מונה) (test-group2), one of whose analyses is: the noun MWNH ('a counter'). By manually tagging all the occurrences of

MWNH in our small corpus we found that the above-mentioned analysis is extremely rare – its relative weight is 0/44. As for the approximated probability of this analysis, its *SW* set contains a single word: HMWNH (המונה – 'the counter') – the definite form of the same noun. The word HMWNH is very frequent in our corpus and for that reason the approximated probability found for this analysis is very high: 0.894. The mismatch between P_{test} and P_{app} in this case is due to the fact that HMWNH is a misleading word – an ambiguous word where one of its other analyses: H + present form of MNH (מנה – 'numbered') is a frequent idiom in Hebrew ('which numbers').

2. Our method may also yield an incorrect approximation for analyses where the similarity assumption we use between the frequency of an analysis and the frequency of the words in its *SW* set does not hold. An example for this is the word $&H (שעה) (test-group2), and one of its analyses: the noun $&H ('an hour'). The approximated probability for this analysis is calculated by looking at the frequency of the similar word H$&H (השעה – 'the hour'). Unfortunately, the similarity assumption does not hold in this case, since the indefinite form of $&H is much more frequent in Hebrew than the definite form of the word. For this reason[13], the approximated probability for this analysis (0.376) is substantially lower than its test-corpus probability (0.847).

9. Morphological Disambiguation

In the previous section we compared the approximated probabilities obtained by our method to the probabilities found by manually tagging a small corpus. We found that the acquired probabilities are truly a good approximation for the morpho-lexical probabilities. In this section we describe an experiment which was conducted in order to test the effectiveness of the morpho-lexical probabilities for morphological disambiguation in Hebrew.

Following are the main components in our project which were used in order to conduct the experiment:

1. A robust morphological analyzer for Hebrew which gives for each word in the language all its possible analyses. The input for our project is supplied by this module.
2. An interactive program for manually tagging Hebrew texts. It was created in order to rapidly tag large texts and was used to mark the right analysis for each ambiguous word in order to be used later to evaluate the performance of our method.
3. Untagged Hebrew corpus. Due to the fact that Hebrew corpora (untagged and tagged as well) are not available in the public domain, we had to build a Hebrew corpus especially for this project. This corpus consists of 11 million word-tokens taken from the daily newspaper *Ha'aretz*.
4. A hash table which stores all the words in the corpus. Each word is accompanied by a counter indicating how many times it appears in the corpus. Since this is the only information we extract from the corpus, our algorithm needs only this hash table and is therefore very efficient.
5. A morphological generator for Hebrew which was written especially for this project. The *SW* sets for every analysis are generated using this module. Due to technical reasons, we were not able to use the morphological analyzer at this stage, and thus we could not identify ambiguous words in the *SW* sets.
6. An implementation of the iterative algorithm which calculates the probabilities.
7. A simple selection algorithm which reduces the level of morphological ambiguity using the probabilities obtained from the corpus. The algorithm uses two thresholds, an **upper**

13 The indefinite form of $&H appears in many Hebrew idioms, e.g., LPY $&H (לפי שעה – 'for the time being'), B^WTH $&H (באותה שעה – 'at the same time') etc.

threshold and a **lower threshold**, which serve to choose the right analysis or to rule out wrong analyses, respectively.

A set of 21 articles was selected in order to test the performance of the method. Since the morpho-lexical probabilities we use are calculated from a large Hebrew corpus (representing a certain Hebrew sub-language), these 21 texts were randomly selected from texts which belong to the same sub-language. The total number of word-tokens in these test-texts was 3,400 out of which nearly 50% were morphologically ambiguous.

The reason for testing the method only on relatively small set of test-texts is due to the fact that no *tagged* Hebrew corpus is currently available for a more powerful evaluation. The need to manually tag the texts used for evaluation limited the number of words in the test-texts we used. Nevertheless, we believe that the results obtained for this restricted set of texts, gives a fairly good indication for the success of the method on large texts as well.

We tested the performance of the method on the test-texts from two different perspectives. First, by using the probabilities only for ambiguous words which can be fully disambiguated – a single analysis can be selected as the right analysis. The performance of the method for full-disambiguation is measured by the **recall** parameter which is defined as follows:

$$\textbf{Recall} = \frac{no.\ of\ correctly\ assigned\ words}{no.\ of\ ambiguous\ words} \quad .$$

In addition to this parameter we present two additional performance parameters – **applicability** and **precision**. We believe that these parameters are relevant for the particular naive method which is described in the current section. This is due to the fact that the morpho-lexical probabilities are not supposed to be used alone for disambiguation, but rather are meant to serve as one information source in a system which combines several linguistic sources for disambiguation. The above-mentioned parameters are defined as follows:

$$\textbf{Precision} = \frac{no.\ of\ correctly\ assigned\ words}{no\ of\ fully\ disambiguated\ words}$$

$$\textbf{Applicability} = \frac{no.\ of\ fully\ disambiguated\ words}{no.\ of\ ambiguous\ words}$$

The results obtained for full disambiguation are shown in Table 5.

Ambiguous words	Disambiguated words	Correct assignments	Recall	Applicability	Precision
1613	1315	1160	72%	82%	88%

Table 5: The performance for full disambiguation

However, the morpho-lexical probabilities can also be used in order to *reduce* the ambiguity level in the text. The performance of the method in this sense is much more interesting and important since it examines, more accurately, the quality of the probabilities as data for other, more sophisticated, systems which use higher levels of information. In this experiment we test the performance of the morpho-lexical probabilities on the task of **analysis assignment**. Here one or more analyses of an ambiguous word are recognized as wrong and hence are rejected. The right analysis should be one of the **remaining analyses**. The three parameters used for evaluation are as follows:

$$\textbf{Recall} = \frac{no.\ of\ correct\ right\ assignments}{no.\ of\ ambiguous\ words}$$

$$\textbf{Precision} = \frac{no.\ of\ correct\ right\ assignments}{no.\ of\ remaining\ analyses}$$

$$\textbf{Fallout} = \frac{no.\ of\ incorrect\ assignments}{no.\ of\ wrong\ analyses}$$

The results are shown in Table 6.

Ambiguous words	Wrong analyses	Remaining analyses	Correct assignments	Incorrect assignments	Recall	Precision	Fallout
1613	3260	1802	1444	358	90%	80%	11%

Table 6: The performance for analysis assignment

In another experiment we examined 891 words with more than two analyses. Table 7 shows how our algorithm reduced the ambiguity of these words.

number of analyses	total	remaining analyses									
		1	2	3	4	5	6	7	8	9	10
3	487	411	62	14							
4	218	170	36	11	1						
5	90	72	12	4	2	0					
6	52	40	12	0	0	0	0				
7	28	22	6	0	0	0	0	0			
8	11	8	2	1	0	0	0	0	0		
9	1	1	0	0	0	0	0	0	0	0	
10	4	4	0	0	0	0	0	0	0	0	0
total	891	728	130	30	3	0	0	0	0	0	0

Table 7: Reducing the degree of ambiguity

These results demonstrate the effectiveness of morpho-lexical probabilities in reducing the ambiguity level in a Hebrew text, and it seems that by using such information combined with other approaches for morphological disambiguation in Hebrew, we come very close to a practical solution for this problem.

10. Conclusions

A method to acquire morpho-lexical probabilities from an *untagged* corpus has been described. The main idea was to use the rich morphology of the language to learn the frequency of a certain analysis from the frequency of other word forms of the same lexical entry.

The results of the experiment confirm the conjecture we made about the nature of the morphological ambiguity problem in Hebrew. It can be argued, therefore, that the computer with its complete morphological knowledge, is facing a much more complex problem than that of a human reading a Hebrew text who may be ignorant of some rare analyses. This observation is also supported by the fact that humans are very often surprised to see the amount of possible analyses of a given ambiguous word. It may even have a significance from a psycholinguistic point of view, by suggesting that this kind of probabilities are also used by a human reader of Hebrew. However, this conjecture should be tested empirically.

An experiment to test the usefulness of the morpho-lexical probabilities for morphological disambiguation in Hebrew yielded the following results: a recall of 70% for full-disambiguation, and a recall of 90% for analysis assignment.

However, the morpho-lexical probabilities cannot serve as the only source of information for morphological disambiguation since they are *imperfect* by definition – they always choose the same analysis as the right one, regardless of the context in which the ambiguous word appears. Thus, as has been already mentioned, we have incorporated these probabilities into an existing system for morphological disambiguation. The combined system tackles the disambiguation problem by combining two kinds of linguistic information sources: **Morpho-Lexical Probabilities** and **Syntactic Constraints** (a full description of this system can be found in (Levinger, 1992)).

Acknowledgments

We would like to thank Ayal Shiran and Ohad Zeliger for programming support of this project, and Ido Dagan for useful discussions concerning this paper. We would also like to thank IBM which enabled us to complete this paper.

References

Academy of The Hebrew Language. 1957. The rules for Hebrew-Latin transcription. *Memoirs of the Academy of the Hebrew Language*, pages 5–8. (In Hebrew).

Agresti, Alan. 1990. *Categorical Data Analysis*. John Wiley & Sons.

Albeck, Orly. 1992. Formal analysis in a register of Israeli Hebrew using a predicative grammar. In U. Ornan G. Arieli and I. Doron, editors, *Hebrew Computational Linguistics*. Israel Ministry of Science and Technology, pages 88–102. (In Hebrew).

Bentur, Esther, Aviela Angel and D. Segev. 1992. Computerized analysis of Hebrew words. In *Hebrew Computerized Linguistics*. Israel Ministry of Science and Technology. (In Hebrew).

Choueka, Yaacov. 1980. Full-text systems and research in the humanities. *Computers and the Humanities*, 14:153–169.

Choueka, Yaacov and S. Lusignan. 1985. Disambiguation by short contexts. *Computers and the Humanities*, 19:147–157.

Church, Kenneth W. 1988. A stochastic parts program and noun phrase parser for unrestricted text. In *Proc. of ACL Conference on Applied Natural Language Processing*, pages 136–143.

Church, Kenneth W. 1992. Current practice in part of speech tagging and suggestions for the future. In A. W. Mackie, T. K. McAuley, and C. Simmons, editors, *For Henry Kucera*. Michigan Slavic Publications, University of Michigan, Ann Arbor, pages 13–48.

Cutting, Doug, Julian Kupiec, Jan Pedersen, and Penelope Sibun. 1992. A practical part-of-speech tagger. In *Proc. of ACL Conference on Applied Natural Language Processing*, pages 133–140.

Dagan, Ido and Alon Itai. 1994. Word sense disambiguation using a second language monolingual corpus. *Computational Linguistics*. to appear.

Dagan, Ido, Alon Itai, and Ulrike Schwall. 1991. Two languages are more informative than one. In *Proc. of the Annual Meeting of the ACL*, pages 130–137.

DeRose, Steven J. 1988. Grammatical category disambiguation by statistical optimization. *Computational Linguistics*, 14:31–39.

Jelinek, Frederick, Robert L. Mercer, and Salim Roukos. 1992. Principles of lexiacal language modeling for speech recognition. In Sadaoki Furui and M. Mohan Sondhi, editors, *Advances in Speech Signal Processing*. Mercer Dekker, Inc., pages 651–699.

Karlsson, Fred. 1990. Constraint grammar as a framework for parsing running text. In *Proc. of COLING*, pages 168–179.

Kupiec, Julian. 1992. Robust part-of-speech tagging using a hidden Markov model. *Computer Speech and Language*, 6:225–242.

Levinger, Moshe. 1992. Morphological disambiguation. Master's thesis, Computer Science Department, Technion – Israel Institute of Technology. (In Hebrew).

Ornan, Uzzi. 1986. Phonemic script: a central vehicle for processing NL – the case of Hebrew. Technical Report 88–181, IBM Scientific Center, Haifa.

Ornan, Uzzi. 1987. Computerized index to decisions of supreme court by phonemic script. *Mishpatim, Hebrew University, Jerusalem*, 17:15–24. (In Hebrew).

Ornan, Uzzi. 1991. Theoretical gemination in Israeli Hebrew. In Alan S. Kaye, editor, *Semitic Studies in Honor of Wolf Leslau*. Otto Harrassowitz, Wiesbaden, pages 1158–1168.

Ornan, Uzzi. 1994. A proposal of the Standards Institution of Israel to ISO – using the phonemic script for Hebrew transcription. ISO conference – Stockholm.

Pines, Ronny. 1975. Ambiguity on the syntactic level. In Ben-Zion Fischler and Uzzi Ornan, editors, *Rosen Memorial Volume*. Council of the Teaching of Hebrew, pages 74–85. (In Hebrew).

Yarowsky, David. 1992. Word sense disambiguation using statistical models of Roget's categories trained on large corpora. In *Proc. of COLING*, pages 454–460.

Appendix A:

Given below is the Latin–Hebrew transliteration used throughout the paper. Note that accepted transcriptions for Hebrew (Academy of The Hebrew Language, 1957; Ornan, 1994) include indication for the vowels which are missing in the modern Hebrew writing system. For this reason, these transcriptions are not suitable for demonstrating the morphological ambiguity problem in the language. Instead, we use the following transliteration which is based on the phonemic script (Ornan, 1994):

Latin	Hebrew	Latin	Hebrew	Latin	Hebrew
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

Table 7: The Hebrew–Latin transliteration

Appendix B:

Following is the set of rules used for Hebrew in order to automatically generate the *SW* set for every morphological analysis in Hebrew. Note that in case an analysis includes a particular attached particle, this particle is also attached to each of its similar words:

1. A definite form of a noun – the *SW* set includes the indefinite form of the same noun.
2. An indefinite form of a noun – the definite form of the same noun.
3. A noun with a possessive pronoun – the same noun with all the other possessive pronouns with the same **person** attribute.
4. An adjective – the other forms of the same adjective (changing the **gender** and **number** attributes).
5. A verb without an object pronoun – the same verb in the same **tense** and **person** (changing the **gender** and **number** attributes only).
6. A verb with an object pronoun – the same verb form with all the other object pronouns forms (preserving the **person** attribute while changing the **gender** and **number** ones).
7. Nominal personal pronoun – the other nominal personal pronouns of the same **person**.
8. A masculine form of a number – the feminine form of the same number.
9. A feminine form of a number – the masculine form of the same number.
10. A proper noun, a particle (preposition, connective etc.) – the empty *SW* set.