Unsupervised Lexicon-Based Resolution of Unknown Words for Full Morphological Analysis

Meni Adler and **Yoav Goldberg** and **David Gabay** and **Michael Elhadad**
Ben Gurion University of the Negev
Department of Computer Science*
POB 653 Be'er Sheva, 84105, Israel
{adlerm,goldberg,gabayd,elhadad}@cs.bgu.ac.il

Abstract

Morphological disambiguation proceeds in 2 stages: (1) an analyzer provides all possible analyses for a given token and (2) a stochastic disambiguation module picks the most likely analysis in context. When the analyzer does not recognize a given token, we hit the problem of unknowns. In large scale corpora, unknowns appear at a rate of 5 to 10% (depending on the genre and the maturity of the lexicon).

We address the task of computing the distribution $p(t|w)$ for unknown words for full morphological disambiguation in Hebrew. We introduce a novel algorithm that is language independent: it exploits a maximum entropy letters model trained over the known words observed in the corpus and the distribution of the unknown words in known tag contexts, through iterative approximation. The algorithm achieves 30% error reduction on disambiguation of unknown words over a competitive baseline (to a level of 70% accurate full disambiguation of unknown words). We have also verified that taking advantage of a strong language-specific model of morphological patterns provides the same level of disambiguation. The algorithm we have developed exploits distributional information latent in a wide-coverage lexicon and large quantities of unlabeled data.

*This work is supported in part by the Lynn and William Frankel Center for Computer Science.

1 Introduction

The term *unknowns* denotes tokens in a text that cannot be resolved in a given lexicon. For the task of full morphological analysis, the lexicon must provide all possible morphological analyses for any given token. In this case, unknown tokens can be categorized into two classes of missing information: *unknown tokens* are not recognized at all by the lexicon, and *unknown analyses*, where the set of analyses for a lexeme does not contain the correct analysis for a given token. Despite efforts on improving the underlying lexicon, unknowns typically represent 5% to 10% of the number of tokens in large-scale corpora. The alternative to continuously investing manual effort in improving the lexicon is to design methods to learn possible analyses for unknowns from observable features: their letter structure and their context. In this paper, we investigate the characteristics of Hebrew unknowns for full morphological analysis, and propose a new method for handling such unavoidable lack of information. Our method generates a distribution of possible analyses for unknowns. In our evaluation, these learned distributions include the correct analysis for unknown words in 85% of the cases, contributing an error reduction of over 30% over a competitive baseline for the overall task of full morphological analysis in Hebrew.

The task of a morphological analyzer is to produce all possible analyses for a given token. In Hebrew, the analysis for each token is of the form lexeme-and-features[1]: lemma, affixes, lexical cate-

[1]In contrast to the prefix-stem-suffix analysis format of

Proceedings of ACL-08: HLT, pages 728–736,
Columbus, Ohio, USA, June 2008.

gory (POS), and a set of inflection properties (according to the POS) – gender, number, person, status and tense. In this work, we refer to the morphological analyzer of MILA – the Knowledge Center for Processing Hebrew[2] (hereafter *KC analyzer*). It is a synthetic analyzer, composed of two data resources – a lexicon of about 2,400 lexemes, and a set of generation rules (see (Adler, 2007, Section 4.2)). In addition, we use an unlabeled text corpus, composed of stories taken from three Hebrew daily news papers (Aruts 7, Haaretz, The Marker), of 42M tokens. We observed 3,561 different composite tags (*e.g.*, noun-sing-fem-prepPrefix:be) over this corpus. These 3,561 tags form the large tagset over which we train our learner. On the one hand, this tagset is much larger than the largest tagset used in English (from 17 tags in most unsupervised POS tagging experiments, to the 46 tags of the WSJ corpus and the about 150 tags of the LOB corpus). On the other hand, our tagset is intrinsically factored as a set of dependent sub-features, which we explicitly represent.

The task we address in this paper is morphological disambiguation: given a sentence, obtain the list of all possible analyses for each word from the analyzer, and disambiguate each word in context. On average, each token in the 42M corpus is given 2.7 possible analyses by the analyzer (much higher than the average 1.41 POS tag ambiguity reported in English (Dermatas and Kokkinakis, 1995)). In previous work, we report disambiguation rates of 89% for full morphological disambiguation (using an unsupervised EM-HMM model) and 92.5% for part of speech and segmentation (without assigning all the inflectional features of the words).

In order to estimate the importance of unknowns in Hebrew, we analyze tokens in several aspects: (1) the number of unknown tokens, as observed on the corpus of 42M tokens; (2) a manual classification of a sample of 10K unknown token types out of the 200K unknown types identified in the corpus; (3) the number of unknown analyses, based on an annotated corpus of 200K tokens, and their classification.

About 4.5% of the 42M token instances in the training corpus were unknown tokens (45% of the 450K token types). For less edited text, such as random text sampled from the Web, the percentage is much higher – about 7.5%. In order to classify these unknown tokens, we sampled 10K unknown token types and examined them manually. The classification of these tokens with their distribution is shown in Table 1[3]. As can be seen, there are two main classes of unknown token types: Neologisms (32%) and Proper nouns (48%), which cover about 80% of the unknown token instances. The POS distribution of the unknown tokens of our annotated corpus is shown in Table 2. As expected, most unknowns are open class words: proper names, nouns or adjectives.

Regarding unknown analyses, in our annotated corpus, we found 3% of the 100K token instances were missing the correct analysis in the lexicon (3.65% of the token types). The POS distribution of the unknown analyses is listed in Table 2. The high rate of unknown analyses for prepositions at about 3% is a specific phenomenon in Hebrew, where prepositions are often prefixes agglutinated to the first word of the noun phrase they head. We observe the very low rate of unknown verbs (2%) – which are well marked morphologically in Hebrew, and where the rate of neologism introduction seems quite low.

This evidence illustrates the need for resolution of unknowns: The naive policy of selecting 'proper name' for all unknowns will cover only half of the errors caused by unknown tokens, *i.e.*, 30% of the whole unknown tokens and analyses. The other 70% of the unknowns (5.3% of the words in the text in our experiments) will be assigned a wrong tag.

As a result of this observation, our strategy is to focus on full morphological analysis for unknown tokens and apply a proper name classifier for unknown analyses and unknown tokens. In this paper, we investigate various methods for achieving full morphological analysis distribution for unknown tokens. The methods are not based on an annotated corpus, nor on hand-crafted rules, but instead exploit the distribution of words in an available lexicon and the letter similarity of the unknown words with known words.

Buckwalter's Arabic analyzer (2004), which looks for any legal combination of prefix-stem-suffix, but does not provide full morphological features such as gender, number, case etc.

[2]http://mila.cs.technion.ac.il.html

[3]Transcription according to Ornan (2002)

Category	Examples		Distribution	
			Types	Instances
Proper names	*’asulin* (family name) *’a’udi* (Audi)	אסולין אאודי	40%	48%
Neologisms	*’agabi* (incidental) *tizmur* (orchestration)	אגבי תזמור	30%	32%
Abbreviation	*mz”p* (DIFS) *kb”t* (security officer)	מז"פ קב"ט	2.4%	7.8%
Foreign	*presentacyah* (presentation) *’a’ut* (out) right	פרזנטציה אאוט	3.8%	5.8%
Wrong spelling	*’abibba’aḥronah* (springatlast) *’idiqacyot* (idication) *ryušalaim* (Rejusalem)	אביבבאחרונה אידיקציות ריושלים	1.2%	4%
Alternative spelling	*’opyynim* (typical) *priwwilegyah* (privilege)	אופייניים פריווילגיה	3.5%	3%
Tokenization	*ha”sap* (the”threshold) *‘al/17* (on/17)	ה"סף על/71	8%	2%

Table 1: Unknown Hebrew token categories and distribution.

Part of Speech	Unknown Tokens	Unknown Analyses	Total
Proper name	31.8%	24.4%	56.2%
Noun	12.6%	1.6%	14.2%
Adjective	7.1%	1.7%	8.8%
Junk	3.0%	1.3%	4.3%
Numeral	1.1%	2.3%	3.4%
Preposition	0.3%	2.8%	3.1%
Verb	1.8%	0.4%	2.2%
Adverb	0.9%	0.9%	1.8%
Participle	0.4%	0.8%	1.2%
Copula	/	0.8%	0.8%
Quantifier	0.3%	0.4%	0.7%
Modal	0.3%	0.4%	0.7%
Conjunction	0.1%	0.5%	0.6%
Negation	/	0.6%	0.6%
Foreign	0.2%	0.4%	0.6%
Interrogative	0.1%	0.4%	0.5%
Prefix	0.3%	0.2%	0.5%
Pronoun	/	0.5%	0.5%
Total	60%	40%	100%

Table 2: Unknowns Hebrew POS Distribution.

2 Previous Work

Most of the work that dealt with unknowns in the last decade focused on unknown tokens (OOV). A naive approach would assign all possible analyses for each unknown token with uniform distribution, and continue disambiguation on the basis of a learned model with this initial distribution. The performance of a tagger with such a policy is actually poor: there are dozens of tags in the tagset (3,561 in the case of Hebrew full morphological disambiguation) and only a few of them may match a given token. Several heuristics were developed to reduce the possibility space and to assign a distribution for the remaining analyses.

Weischedel et al. (1993) combine several heuristics in order to estimate the token generation probability according to various types of information – such as the characteristics of particular tags with respect to unknown tokens (basically the distribution shown in Table 2), and simple spelling features: capitalization, presence of hyphens and specific suffixes. An accuracy of 85% in resolving unknown tokens was reported. Dermatas and Kokkinakis (1995) suggested a method for guessing unknown tokens based on the distribution of the hapax legomenon, and reported an accuracy of 66% for English. Mikheev (1997) suggested a guessing-rule technique, based on prefix morphological rules, suffix morphological rules, and ending-guessing rules. These rules are learned automatically from raw text. They reported a tagging accuracy of about 88%. Thede and Harper (1999) extended a second-order HMM model with a $C = c_{k,i}$ matrix, in order to encode the probability of a token with a suffix s_k to be generated by a tag t_i. An accuracy of about 85% was reported.

Nakagawa (2004) combine word-level and character-level information for Chinese and Japanese word segmentation. At the word level, a segmented word is attached to a POS, where the character model is based on the observed characters and their classification: **B**egin of word, **I**n the middle of a word, **E**nd of word, the character is a word itself **S**. They apply Baum-Welch training over a segmented corpus, where the segmentation of each word and its character classification is observed, and the POS tagging is ambiguous. The segmentation (of all words in a given sentence) and the POS tagging (of the known words) is based on a Viterbi search over a lattice composed of all possible word segmentations and the possible classifications of all observed characters. Their experimental results show that the method achieves high accuracy over state-of-the-art methods for Chinese and Japanese word segmentation. Hebrew also suffers from ambiguous segmentation of agglutinated tokens into significant words, but word formation rules seem to be quite different from Chinese and Japanese. We also could not rely on the existence of an annotated corpus of segmented word forms.

Habash and Rambow (2006) used the root+pattern+features representation of Arabic tokens for morphological analysis and generation of Arabic dialects, which have no lexicon. They report high recall (95%–98%) but low precision (37%–63%) for token types and token instances, against gold-standard morphological analysis. We also exploit the morphological patterns characteristic of semitic morphology, but extend the guessing of morphological features by using contextual features. We also propose a method that relies exclusively on learned character-level features and contextual features, and eventually reaches the same performance as the patterns-based approach.

Mansour et al. (2007) combine a lexicon-based tagger (such as MorphTagger (Bar-Haim et al., 2005)), and a character-based tagger (such as the data-driven ArabicSVM (Diab et al., 2004)), which includes character features as part of its classification model, in order to extend the set of analyses suggested by the analyzer. For a given sentence, the lexicon-based tagger is applied, selecting one tag for a token. In case the ranking of the tagged sentence is lower than a threshold, the character-based tagger is applied, in order to produce new possible analyses. They report a very slight improvement on Hebrew and Arabic supervised POS taggers.

Resolution of Hebrew unknown tokens, over a large number of tags in the tagset (3,561) requires a much richer model than the the heuristics used for English (for example, the capitalization feature which is dominant in English does not exist in Hebrew). Unlike Nakagawa, our model does not use any segmented text, and, on the other hand, it aims to select full morphological analysis for each token,

including unknowns.

3 Method

Our objective is: given an unknown word, provide a distribution of possible tags that can serve as the analysis of the unknown word. This unknown analysis step is performed at training and testing time. We do not attempt to disambiguate the word – but only to provide a distribution of tags that will be disambiguated by the regular EM-HMM mechanism.

We examined three models to construct the distribution of tags for unknown words, that is, whenever the KC analyzer does not return any candidate analysis, we apply these models to produce possible tags for the token $p(t|w)$:

Letters A maximum entropy model is built for all unknown tokens in order to estimate their tag distribution. The model is trained on the known tokens that appear in the corpus. For each analysis of a known token, the following features are extracted: (1) unigram, bigram, and trigram letters of the base-word (for each analysis, the base-word is the token without prefixes), together with their index relative to the start and end of the word. For example, the n-gram features extracted for the word `abc` are { `a:1 b:2 c:3 a:-3 b:-2 c:-1 ab:1 bc:2 ab:-2 bc:-1 abc:1 abc:-1` } ; (2) the prefixes of the base-word (as a single feature); (3) the length of the base-word. The class assigned to this set of features, is the analysis of the base-word. The model is trained on all the known tokens of the corpus, each token is observed with its possible POS-tags once for each of its occurrences. When an unknown token is found, the model is applied as follows: all the possible linguistic prefixes are extracted from the token (one of the 76 prefix sequences that can occur in Hebrew); if more than one such prefix is found, the token is analyzed for each possible prefix. For each possible such segmentation, the full feature vector is constructed, and submitted to the Maximum Entropy model. We hypothesize a uniform distribution among the possible segmentations and aggregate a distribution of possible tags for the analysis. If the proposed tag of the base-word is never found in the corpus preceded by the identified prefix, we remove this possible analysis. The eventual outcome of the model application is a set of possible full morphological analyses for the token – in exactly the same format as the morphological analyzer provides.

Patterns Word formation in Hebrew is based on root+pattern and affixation. Patterns can be used to identify the lexical category of unknowns, as well as other inflectional properties. Nir (1993) investigated word-formation in Modern Hebrew with a special focus on neologisms; the most common word-formation patterns he identified are summarized in Table 3. A naive approach for unknown resolution would add all analyses that fit any of these patterns, for any given unknown token. As recently shown by Habash and Rambow (2006), the precision of such a strategy can be pretty low. To address this lack of precision, we learn a maximum entropy model on the basis of the following binary features: one feature for each pattern listed in column **Formation** of Table 3 (40 distinct patterns) and one feature for "no pattern".

Pattern-Letters This maximum entropy model is learned by combining the features of the letters model and the patterns model.

Linear-Context-based $p(t|c)$ approximation The three models above are context free. The linear-context model exploits information about the lexical context of the unknown words: to estimate the probability for a tag t given a context c – $p(t|c)$ – based on all the words in which a context occurs, the algorithm works on the known words in the corpus, by starting with an initial tag-word estimate $p(t|w)$ (such as the morpho-lexical approximation, suggested by Levinger et al. (1995)), and iteratively re-estimating:

$$\hat{p}(t|c) = \frac{\sum_{w \in W} p(t|w)p(w|c)}{Z}$$

$$\hat{p}(t|w) = \frac{\sum_{c \in C} p(t|c)p(c|w)allow(t,w)}{Z}$$

where Z is a normalization factor, W is the set of all words in the corpus, C is the set of contexts. $allow(t,w)$ is a binary function indicating whether t is a valid tag for w. $p(c|w)$ and $p(w|c)$ are estimated via raw corpus counts.

Loosely speaking, the probability of a tag given a context is the average probability of a tag given any

Category	Formation		Example	
Verb	Template	’iCCeC	*’ibḥen* (diagnosed)	אבחן
		miCCeC	*miḥzer* (recycled)	מחזר
		CiCCen	*timren* (manipulated)	תמרן
		CiCCet	*tiknet* (programmed)	תכנת
		tiCCeC	*ti’arek* (dated)	תארך
Participle	Template	meCuCaca	*mšwḥzar* (reconstructed)	משוחזר
		muCCaC	*muqlaṭ* (recorded)	מוקלט
		maCCiC	*malbin* (whitening)	מלבין
Noun	Suffixation	ut	*ḥaluciyut* (pioneership)	חלוציות
		ay	*yomanay* (duty officer)	יומנאי
		an	*’egropan* (boxer)	אגרופן
		on	*paḥon* (shack)	פחון
		iya	*marakiyah* (soup tureen)	מרקייה
		it	*ṭiyulit* (open touring vehicle)	טיולית
		a	*lomdah* (courseware)	לומדה
	Template	maCCeC	*mašneq* (choke)	משנק
		maCCeCa	*madgera* (incubator)	מדגרה
		miCCaC	*mis‘ap* (branching)	מסעף
		miCCaCa	*mignana* (defensive fighting)	מגננה
		CeCeC[a]	*peleṭ* (output)	פלט
		tiCCoCet	*tiproset* (distribution)	תפרוסת
		taCCiC	*taḥriṭ* (engraving)	תחריט
		taCCuCa	*tabru’ah* (sanitation)	תברואה
		miCCeCet	*micrepet* (leotard)	מצרפת
		CCiC	*crir* (dissonance)	צריר
		CaCCan	*balšan* (linguist)	בלשן
		CaCeCet	*šaḥemet* (cirrhosis)	שחמת
		CiCul	*ṭibu‘* (ringing)	טיבוע
		haCCaCa	*hanpaša* (animation)	הנפשה
		heCCeC	*het’em* (agreement)	התאם
Adjective	Suffixation[b]	i	*nora’i* (awful)	נוראי
		ani	*yeḥidani* (individual)	יחידני
		oni	*ṭelewizyoni*[c] (televisional)	טלוויזיוני
		a’i	*yeḍida’i* (unique)	יחידאי
		ali	*sṭudentiali* (student)	סטודנטיאלי
	Template	$C_1C_2aC_3C_2aC_3$[d]	*metaqtaq* (sweetish)	מתקתק
		CaCuC	*rapus* (flaccid)	רפוס
Adverb	Suffixation	ot	*qcarot* (briefly)	קצרות
		it	*miyadit* (immediately)	מידית
	Prefixation	b	*bekeip* (with fun)	בכיף

[a] CoCeC variation: עותק *‘wyeq* (a copy).
[b] The feminine form is made by the *t* and *iya* suffixes: יחידני *yeḥidanit* (individual), נוצריה *nwcriya* (Christian).
[c] In the feminine form, the last *h* of the original noun is omitted.
[d] $C_1C_2aC_3C_2oC_3$ variation: קטנטון *qṭanṭwn* (tiny).

Table 3: Common Hebrew Neologism Formations.

Model	Analysis Set			Morphological Disambiguation
	Coverage	Ambiguity	Probability	
Baseline	50.8%	**1.5**	**0.48**	57.3%
Pattern	82.8%	20.4	0.10	66.8%
Letter	76.7%	5.9	0.32	69.1%
Pattern-Letter	84.1%	10.4	0.25	**69.8%**
WordContext-Pattern	84.4%	21.7	0.12	66.5%
TagContext-Pattern	85.3%	23.5	0.19	64.9%
WordContext-Letter	80.7%	7.94	0.30	**69.7%**
TagContext-Letter	83.1%	7.8	0.22	66.9%
WordContext-Pattern-Letter	85.2%	12.0	0.24	68.8%
TagContext-Pattern-Letter	**86.1%**	14.3	0.18	62.1%

Table 4: Evaluation of unknown token full morphological analysis.

of the words appearing in that context, and similarly the probability of a tag given a word is the averaged probability of that tag in all the (reliable) contexts in which the word appears. We use the function $allow(t, w)$ to control the tags (ambiguity class) allowed for each word, as given by the lexicon.

For a given word w_i in a sentence, we examine two types of contexts: **word context** w_{i-1}, w_{i+1}, and **tag context** t_{i-1}, t_{i+1}. For the case of word context, the estimation of $p(w|c)$ and $p(c|w)$ is simply the relative frequency over all the events $w1, w2, w3$ occurring at least 10 times in the corpus. Since the corpus is not tagged, the relative frequency of the tag contexts is not observed, instead, we use the context-free approximation of each word-tag, in order to determine the frequency weight of each tag context event. For example, given the sequence תגובה לעומתית למדי *tgubah l'umatit lmadai* (a quite oppositional response), and the analyses set produced by the context-free approximation: *tgubah* [NN 1.0] *l'umatit* [] *lmadai* [RB 0.8, P1-NN 0.2]. The frequency weight of the context {NN RB} is $1 * 0.8 = 0.8$ and the frequency weight of the context {NN P1-NN} is $1 * 0.2 = 0.2$.

4 Evaluation

For testing, we manually tagged the text which is used in the Hebrew Treebank (consisting of about 90K tokens), according to our tagging guideline (**?**).

We measured the effectiveness of the three models with respect to the tags that were assigned to the unknown tokens in our test corpus (the 'correct tag'), according to three parameters: (1) The coverage of the model, *i.e.*, we count cases where $p(t|w)$ contains the correct tag with a probability larger than 0.01; (2) the ambiguity level of the model, *i.e.*, the average number of analyses suggested for each token; (3) the average probability of the 'correct tag', according to the predicted $p(t|w)$. In addition, for each experiment, we run the full morphology disambiguation system where unknowns are analyzed according by the model.

Our baseline proposes the most frequent tag (proper name) for all possible segmentations of the token, in a uniform distribution. We compare the following models: the 3 context free models (patterns, letters and the combined patterns and letters) and the same models combined with the word and tag context models. Note that the context models have low coverage (about 40% for the word context and 80% for the tag context models), and therefore, the context models cannot be used on their own. The highest coverage is obtained for the combined model (tag context, pattern, letter) at 86.1%.

We first show the results for full morphological disambiguation, over 3,561 distinct tags in Table 4. The highest coverage is obtained for the model combining the tag context, patterns and letters models. The tag context model is more effective because it covers 80% of the unknown words, whereas the word context model only covers 40%. As expected, our simple baseline has the highest precision, since the most frequent proper name tag covers over 50% of the unknown words. The eventual effectiveness of

Model	Analysis Set			POS Tagging
	Coverage	Ambiguity	Probability	
Baseline	52.9%	**1.5**	**0.52**	60.6%
Pattern	87.4%	8.7	0.19	76.0%
Letter	80%	4.0	0.39	77.6%
Pattern-Letter	86.7%	6.2	0.32	**78.5%**
WordContext-Pattern	88.7%	8.8	0.21	75.8%
TagContext-Pattern	**89.5%**	9.1	0.14	73.8%
WordContext-Letter	83.8%	4.5	0.37	**78.2%**
TagContext-Letter	87.1%	5.7	0.28	75.2%
WordContext-Pattern-Letter	87.8	6.5	0.32	77.5%
TagContext-Pattern-Letter	89.0%	7.2	0.25	74%

Table 5: Evaluation of unknown token POS tagging.

the method is measured by its impact on the eventual disambiguation of the unknown words. For full morphological disambiguation, our method achieves an error reduction of 30% (57% to 70%). Overall, with the level of 4.5% of unknown words observed in our corpus, the algorithm we have developed contributes to an error reduction of 5.5% for full morphological disambiguation.

The best result is obtained for the model combining pattern and letter features. However, the model combining the word context and letter features achieves almost identical results. This is an interesting result, as the pattern features encapsulate significant linguistic knowledge, which apparently can be approximated by a purely distributional approximation.

While the disambiguation level of 70% is lower than the rate of 85% achieved in English, it must be noted that the task of full morphological disambiguation in Hebrew is much harder – we manage to select one tag out of 3,561 for unknown words as opposed to one out of 46 in English. Table 5 shows the result of the disambiguation when we only take into account the POS tag of the unknown tokens. The same models reach the best results in this case as well (Pattern+Letters and WordContext+Letters). The best disambiguation result is 78.5% – still much lower than the 85% achieved in English. The main reason for this lower level is that the task in Hebrew includes segmentation of prefixes and suffixes in addition to POS classification. We are currently investigating models that will take into account the specific nature of prefixes in Hebrew (which encode conjunctions, definite articles and prepositions) to better predict the segmentation of unknown words.

5 Conclusion

We have addressed the task of computing the distribution $p(t|w)$ for unknown words for full morphological disambiguation in Hebrew. The algorithm we have proposed is language independent: it exploits a maximum entropy letters model trained over the known words observed in the corpus and the distribution of the unknown words in known tag contexts, through iterative approximation. The algorithm achieves 30% error reduction on disambiguation of unknown words over a competitive baseline (to a level of 70% accurate full disambiguation of unknown words). We have also verified that taking advantage of a strong language-specific model of morphological patterns provides the same level of disambiguation. The algorithm we have developed exploits distributional information latent in a wide-coverage lexicon and large quantities of unlabeled data.

We observe that the task of analyzing unknown tokens for POS in Hebrew remains challenging when compared with English (78% vs. 85%). We hypothesize this is due to the highly ambiguous pattern of prefixation that occurs widely in Hebrew and are currently investigating syntagmatic models that exploit the specific nature of agglutinated prefixes in Hebrew.

References

Meni Adler. 2007. *Hebrew Morphological Disambiguation: An Unsupervised Stochastic Word-based Approach*. Ph.D. thesis, Ben-Gurion University of the Negev, Beer-Sheva, Israel.

Roy Bar-Haim, Khalil Sima'an, and Yoad Winter. 2005. Choosing an optimal architecture for segmentation and pos-tagging of modern Hebrew. In *Proceedings of ACL-05 Workshop on Computational Approaches to Semitic Languages*.

Tim Buckwalter. 2004. Buckwalter Arabic morphological analyzer, version 2.0.

Evangelos Dermatas and George Kokkinakis. 1995. Automatic stochastic tagging of natural language texts. *Computational Linguistics*, 21(2):137–163.

Mona Diab, Kadri Hacioglu, and Daniel Jurafsky. 2004. Automatic tagging of Arabic text: From raw text to base phrase chunks. In *Proceeding of HLT-NAACL-04*.

Michael Elhadad, Yael Netzer, David Gabay, and Meni Adler. 2005. Hebrew morphological tagging guidelines. Technical report, Ben-Gurion University, Dept. of Computer Science.

Nizar Habash and Owen Rambow. 2006. Magead: A morphological analyzer and generator for the arabic dialects. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 681–688, Sydney, Australia, July. Association for Computational Linguistics.

Moshe Levinger, Uzi Ornan, and Alon Itai. 1995. Learning morpholexical probabilities from an untagged corpus with an application to Hebrew. *Computational Linguistics*, 21:383–404.

Saib Mansour, Khalil Sima'an, and Yoad Winter. 2007. Smoothing a lexicon-based pos tagger for Arabic and Hebrew. In *ACL07 Workshop on Computational Approaches to Semitic Languages*, Prague, Czech Republic.

Andrei Mikheev. 1997. Automatic rule induction for unknown-word guessing. *Computational Linguistics*, 23(3):405–423.

Tetsuji Nakagawa. 2004. Chinese and Japanese word segmentation using word-level and character-level information. In *Proceedings of the 20th international conference on Computational Linguistics*, Geneva.

Raphael Nir. 1993. *Word-Formation in Modern Hebrew*. The Open University of Israel, Tel-Aviv, Israel.

Uzi Ornan. 2002. Hebrew in Latin script. *Lěšonénu*, LXIV:137–151. (in Hebrew).

Scott M. Thede and Mary P. Harper. 1999. A second-order hidden Markov model for part-of-speech tagging. In *Proceeding of ACL-99*.

R. Weischedel, R. Schwartz, J. Palmucci, M. Meteer, and L. Ramshaw. 1993. Coping with ambiguity and unknown words through probabilistic models. *Computational Linguistics*, 19:359–382.