

# Unsupervised Induction of Arabic Root and Pattern Lexicons using Machine Learning

**Bilal Khaliq**

Dept. of Informatics  
University of Sussex  
bk54@sussex.ac.uk

**John Carroll**

Dept. of Informatics  
University of Sussex  
johnca@sussex.ac.uk

## Abstract

We describe an approach to building a morphological analyser of Arabic by inducing a lexicon of root and pattern templates from an unannotated corpus. Using maximum entropy modelling, we capture orthographic features from surface words, and cluster the words based on the similarity of their possible roots or patterns. From these clusters, we extract root and pattern lexicons, which allows us to morphologically analyse words. Further enhancements are applied, adjusting for morpheme length and structure. Final root extraction accuracy of 87.2% is achieved. In contrast to previous work on unsupervised learning of Arabic morphology, our approach is applicable to naturally-written, unvowelled Arabic text.

## 1 Introduction

The number and diversity of human languages makes it impractical to manually craft lexicons and morphological processors for more than a very small proportion of them. Further challenges are posed by the need to deal with dialects and colloquial forms of languages. This has motivated recent increased interest in approaches to morphological analysis based on unsupervised learning. Inspired by competitions such as the Morpho Challenge, many techniques have been proposed for unsupervised morphology learning.

Although these techniques are often intended to be language independent, they are often directed to a specific group of languages. Most work has aimed at sequential separation or segmentation of morphemes concatenated together in a surface word form. This type of analysis, outputting stems and appended morphemes aims to identify some kind of border between the different morphemes. However, another type of word formation consists of the interdigitation of a root morpheme with an affix or pattern template; in this case there is no boundary between morphemes, since they are rather intercalated with each other. This type of non-concatenative morphology, which is characteristic of the Semitic group of languages,

has attracted far less interest for unsupervised learning.

In this paper we present an approach to unsupervised learning of non-concatenative morphology, applying it to Arabic. We describe an approach to learning tri-literal roots and affix template of Arabic by first inducing root and affix lexicons. Our approach uses Maximum Entropy modelling to obtain clusters<sup>1</sup> of words based on concatenative and non-concatenative orthographic features, and induces the lexicons from these clusters.

Our data is an undiacritized version of the Quranic Arabic Corpus since we assume a realistic setting of unvowelled text, as most Arabic text is written without vowels; we chose this corpus since correct roots of each word are available, facilitating the evaluation process. The fact that the corpus contains a relatively small vocabulary of around 7000 words also simulates the scenario for most of the world's languages of scarcity of linguistic resources and data.

This paper is structured as follows: Section 2 surveys previous related work. Section 3 provides an introduction to Arabic root and pattern morphology. Our approach to unsupervised lexicon induction based on Maximum Entropy (ME) modelling is explained in section 4. Section 5 describes the procedure for performing morphological analysis of words, followed by evaluation in section 6 and conclusions in section 7.

## 2 Related Work

An active current area of natural language processing research is applying statistical and information-theoretic approaches to unsupervised learning of morphology and grammar. A common starting point is raw (unannotated) text corpora, inducing the target knowledge from word forms and their patterns of usage.

Information theoretic approaches, particularly Minimum Description Length (MDL) as investigated by Goldsmith (2000, 2006) and others

---

<sup>1</sup> Cluster here refers to a collection of words related in terms of morpheme types, without referring to application of any clustering algorithm.

(Cruetz and Lagus, 2005, 2007), have brought a theoretical perspective considering input data to be ‘compressed’ into a morphologically analysed representation. This optimization scheme has achieved good results, and is amongst the most effective approaches for unsupervised morphological analysis.

Most work on unsupervised learning of morphology has focused on concatenative morphology (De Pauw and Wagacha 2007; Hammarström and Borin 2011). Another perspective adopted by Schone and Jurafsky (2001) incorporates orthographic and phonological features, and induces semantic relatedness between word pairs using Latent Semantic Indexing. Their work shows comparable performance to Goldsmith’s (2000) Linguistica system. Yarowsky and Wicentowski (2000) experiment with learning irregular mnaturaorphology using a lightly supervised technique to align irregular words to their lemmas by estimating the distribution of ratios over part-of-speech classes of inflected words to lemmas.

More recently, researchers have addressed non-concatenative morphology, such as for Semitic languages, using a variety of empirical approaches. Daya et al. (2008) learn Semitic roots using supervised learning, building a multi-class classifier for individual root radicals. Clark (2007) uses Arabic as a test-bed to study semi-supervised learning of complex broken plural structure modelled using memory-based algorithms, with the aim of gaining insights into human language acquisition.

Most work on unsupervised learning of morphology has focused on concatenative morphology (Hammarström and Borin 2011). The few studies that have focussed on non-concatenative morphology, such as for Semitic languages, have not used naturally written text. For example, Rodriguez and Cavar (2005) learn roots using a number of orthographic heuristics and then apply constraint-based learning to improve the quality of roots. Xanthos (2008) works on phonetic transcriptions of Arabic text to decipher roots and patterns. The approach is to initially create crude Root and Pattern (RP) transcriptions from words based on vowel-consonant distinctions, and then to apply an MDL approach similar to Goldsmith’s (2006) in order to refine the RP structures.

In contrast to previous work, we learn intercalated morphology, identifying the root and transfixes/ incomplete pattern for words from ‘natural’ text without short vowels or diacritical markers.

### 3 Root and Pattern Morphology

Words in Arabic are formed through three morphological processes. The first (i) is the fusion of a root form and pattern template to derive a base word, which can be a noun, verb or adjective, all of which are semantically related to the root. The second (ii) is affixation, by means of prefixes, suffixes or infixes, including inflectional morphemes marking gender, plurality and/or tense, resulting in a stem. Thirdly (iii) a final layer of clitics may be attached to a word, including a subset of prepositions, conjunctions, determiners and pronouns; these appear at the beginning (proclitics) or end (enclitics) of a word but never in the middle.

Since techniques for concatenative morphology learning are fairly advanced we have focused on using stemmed words, computable through such approaches. We used the QAC stem vocabulary where appended morphemes of type (iii) are mostly absent<sup>2</sup> and hence ignored from analysis. Most of type (ii) are present as part of the stem. In the case of (i), most derived forms consist of short vowels and occasional long vowels or a consonant interdigitated with the root. In unvowelled text the short vowels are ignored, so derived words have at most single letter affixation.

Table 1 shows two example words with their roots and affix pattern templates. The ‘y’ and ‘t’ in the respective words are clitic/inflectional markers, which are part of the affix template. ‘A’ is the derivational infix marker for nouns.

Word	Root	Pattern
ktAby	Ktb	--A-y
tEArf	Erf	t-A--

Table 1: Example words with their roots and affix pattern templates.

For analysis, each word,  $w$ , is decomposed, using a decomposition function, into a set of tuples encoding all  $n$  possible combinations of a root (of at least 3 letters) and associated pattern:

$$d(w) \rightarrow \{(r^x, p^x)\} \quad (\text{Eq. 1})$$

where  $x$  ranges from 1 to  $n$ . For example, the decomposition of the word ‘yErf’, is shown in Figure 1.

<sup>2</sup> Stems in QAC include the attached pronoun clitics

$$yErf \rightarrow \left\{ \begin{array}{l} \langle yEr, \quad --f \rangle, \\ \langle yEf, \quad --r- \rangle, \\ \langle yrf, \quad -E-- \rangle, \\ \langle Erf, \quad y--- \rangle, \\ \langle yErf, \quad ---- \rangle \end{array} \right\}$$

Figure 1: Decomposition of a word into all possible combinations of roots and patterns.

#### 4 Using Maximum Entropy Modelling for Unsupervised Learning

In this study we apply an supervised machine learning technique, Maximum Entropy (ME) modelling, in a completely unsupervised way, taking our inspiration from the work of De Pauw and Wagacha (2007), who applied the approach for extracting prefixes in an African language.

Unlike for supervised learning, no annotated text is used. Instead we simply derive features automatically from the vocabulary words of the dataset. Each word is represented as an output class mapped to by the corresponding features of the words. These word-features are used to train a classifier. Rather than applying the classifier to classify unseen data, we apply the model back to the ‘training data’ to obtain, not the classification but the proximities of each word/class with every other word/class. These proximities are then utilized to derive root and pattern lexicons.

The advantage of this approach to gauge relatedness of words over other approaches, such as minimum edit distance, is the ability to better capture morpheme dependencies between words with common roots which may be orthographically quite different due to substantial affixing.

##### 4.1 Building the Lexicons

We derive two lexicons: a root lexicon and an affix or pattern lexicon. We do this by training ME classifiers on orthographic features computed from each word in the corpus dataset. The classifiers are then applied to the same data to obtain word clusters relating each word to every other word with respect to either common roots or common patterns. Thus, for the root lexicon we obtain neighbours of words that have the same or similar patterns. Conversely, for the pattern lexicon we obtain neighbours of words that have common root radicals.

##### 4.2 Modelling Orthographic Features

We first extract orthographic features for obtaining word clusters with similar roots (i.e. for pattern lexicon acquisition). We then construct the inverse

of these features for obtaining word clusters with similar patterns (i.e. for root lexicon acquisition).

In the former case, feature extraction proceeds as follows: we first enclose each word with beginning and end boundary markers, ‘@’ and ‘#’ respectively. (This is in order to provide context information for the first and last characters of a word). We next compute the power-set of all the character combinations in a word, and then exclude features where the first and last letter of the word appear without the boundary markers (to give emphasis to word boundary features). The final set of these features for the word ‘yErf’ is shown in the first column of Table 2.

In the latter case, pattern features are obtained such that corresponding to each root feature, we replace root radicals with a placeholder; characters between root radicals that are omitted from the root features appear as potential affix characters in the pattern template. These inverse features are shown in the second column of Table 2.

Root Features (for Pattern Lexicon)	Pattern features (for Root Lexicon)
@y, @yE,	@-, @--,
@yEr, @yErf#,	@---, @----#,
@yEr#, @yEf#,	@---f#, @--r#,
@yE#, @yr,	@--rf#, @-E-,
@yrf#, @yr#,	@-E--#, @-E-f#,
@yf#, @y#,	@-Er-#, @-Erf#,
@E, @Er,	@y-, @y--,
@Erf#, Er#,	@y--f#, @y--rf#,
@Ef#, @E#,	@y-r-#, @y-rf#,
@r, @rf#,	@yE-, @yE--#,
@r#, @f#,	@yE-f#, @yEr-#,
E, Er,	-, --,
Erf#, Er#,	--f#, --rf#,
Ef#, E#,	-r-#, -rf#,
r, rf#,	-, --#,
r#, f#,	-f#, -#

Table 2: Features for the word ‘yErf’.

##### 4.3 Word Nearest Neighbors

The classifier is trained using Limited Variable LBFGS optimization method. The number of iterations for training is stopped automatically when 100% accuracy on the training data is achieved. Each trained classifier is reapplied to its respective training data features to get proximity values between each word and every other word. Sorting the list gives us the most related word in terms of root based or pattern based proximity values, with the highest value ( $\approx 1$ ) for the headword,  $h$ , i.e. the word’s own features. Table 3

shows an example of the closest neighbours in a cluster, along with their headword.

Using these words and proximity measures we next apply a strategy to induce the morpheme. Not all words in the list of  $N$  elements for each word are relevant to us since the proximity value starts to drop rapidly towards zero as we go down the ranked list. With each headword we choose a 500 nearest neighbours cluster for each type of morpheme as a sufficient number beyond which we expect no gain in efficiency is expected.

Head-Word, $h$	Proximity for Root Cluster		Proximity for Pattern Cluster	
	$k$	$P(k)$	$k$	$P(k)$
$yErf$	$yErf$	0.9897420	$yErf$	0.99999
	$Erf$	0.0023982	$yHrf$	2.59E-07
	$yEf$	0.0022552	$ysrf$	2.58E-07
	$tErf$	0.0015299	$ySrf$	2.32E-07
	$yErD$	0.0014147	$yEkf$	2.31E-07
	$yEr\$$	0.0011525	$tErf$	1.10E-09
	$yErj$	0.0009722	$yErj$	4.24E-10
	$Ef$	0.0001968	$yErD$	3.29E-10
	$yr$	0.0001052	$yEr\$$	2.36E-10
	$'Etrf$	2.5629E-05	$msrf$	2.14E-12
	$yrd$	8.6797E-06	$zurf$	1.51E-12
	...	...	...	...

Table 3: ME values for the word  $yErf$ .

#### 4.4 Dictionary Induction

Using the respective word clusters we create dictionaries for two types of morphemes, roots and patterns, such that we score the morphemes thus: Higher scoring morphemes are more plausible and ranked higher in the lexical list than lower ones. The procedure for scoring is adapted and amended from the work of De Pauw and Wagacha (2007).

For the pattern lexicon, we score each pattern in the following manner: for each headword,  $h_i$  (having probability value  $\approx 1$ ) in cluster  $c_i$  (with each of the  $i = 1, 2, \dots, N$  words in the vocabulary), we obtain all possible decompositions (equation 1) into template patterns  $p_h^x$  (shown in column 1 of Table 4) and roots,  $r_h^x$  (column 2 of Table 4) with respect to the headword,  $h_i$ . Each pattern is scored with a function  $S(p_h^x)$  (equation 2) which aggregates the Logarithmically Scaled ( $LS$ ) probability value,  $P_{k_j}$  of words  $k_j$  ( $j = 1, 2, \dots, 500$  words in each cluster), such that  $r_h^x$  matches any of the roots in word  $k$ ,  $r_k^y$  ( $y = 1, 2, \dots, m$  root combinations in  $k$ ). This aggregation is not only local to each cluster but covers all occurrences of the pattern in each of the  $N$  clusters.

$$S(p_h^x) = \sum_{i=1}^N \sum_{j=1}^{500} (LS(P_{k_j}) \times LA(|p_h^x|) | r_{h_i}^x = r_{k_j}^y) \quad (\text{Eq. 2})$$

Logarithmic scaling is necessary since the probability drops too rapidly and too low in order to provide a feasible ratio between words. After taking the log of the probability the resulting ratios are negative which are then adjusted by subtracting the log of a base probability value,  $P_0$ , thus linearly inverting the ratios (equation 3).  $P_0$  is hence chosen to be small enough to ensure the resulting logarithmic score is positive. We chose the smallest occurring probability value in our clusters as the value for  $P_0$ .

$$LS(P_{k_j}) = \log P(k_j) - \log P_0 \quad (\text{Eq. 3})$$

The score is also exponentially Length Adjusted ( $LA$ ) for each pattern,  $p$ , according to the length of the pattern,  $|p|$ , in terms of the number of affix characters in  $p$ . This boosts the score for lengthier morphemes which are relatively infrequent. The intuition for adjustment formula comes from the work of (Chung and Gildea, 2009) and (Liang and Klein, 2009), who use an exponential Length Penalty measure to adjust their model for morpheme length.

$$LA(|p|) = e^{|p|} \quad (\text{Eq. 4})$$

Thus the pattern is scored according to the score of words containing plausible roots. Commonly occurring patterns such as 'y---' gather weight and ascend the list of the most frequent (and hence potentially sound) affix templates. Table 4 shows how each pattern for the headword 'yErf' is scored, aggregating the logarithmic score over words (in column 4 of Table 4) containing the roots in column 2 of Table 4.

Pattern	Root	Word, $k$ , with Root	Pattern Weight
$y---$	$Erf$	$Erf, tErf, 'Etrf$	19.97328
$-E--$	$Yrf$	–	0.0
$--r-$	$yEf$	$yEf$	7.353
$---f$	$yEr$	$yErD, yEr$, yErj$	21.200

Table 4: Example pattern candidate scoring.

Similarly, we score the root,  $S(r_h^x)$ , with respect to the pattern occurrence in each word  $k$  of cluster  $c_i$ :

$$S(r_h^x) = \sum_{i=1}^N \sum_{j=1}^{500} (LS(P_{kj}) | p_{h_i}^x = p_{k_j}^y) \quad (\text{Eq. 5})$$

The scoring aggregates over the log scaled probability of words in the affix-based clusters having pattern occurrences in a word in each cluster. There is no need for length adjustment to these ratios since we are considering only three letter roots. Table 5 exemplifies this for scoring roots with words (in column 3 of Table 5) that have corresponding patterns (in column 2 of Table 5).

Root	Pattern	Word, $k$ , with Pattern	Pattern weight
<i>Erf</i>	<i>y---</i>	<i>yHrf, ysrf, ySrf, ...</i>	25.190
<i>Yrf</i>	<i>-E--</i>	<i>yEkf, tErf, yErj, ...</i>	20.032
<i>yEf</i>	<i>--r-</i>	<i>yHrf, ysrf, ySrf, ...</i>	54.259
<i>yEr</i>	<i>---f</i>	<i>yHrf, ysrf, ySrf, ...</i>	46.104

Table 5: Example pattern candidate scoring.

Table 6 shows the top lexicon entries for roots and patterns along with their respective scores. The top entries in the lexicon would plausibly be correct morphemes while lower entries would be not so plausible.

Root Lexicon		Pattern Lexicon	
'mn	49067.2	y---	62987.8
Sdq	44801.4	'---	61905.4
xlf	42768.4	t---	54634.3
\$hd	42607.8	---A	51777.1
xrj	40872.8	n---	44257
nSr	40111.4	--y-	31058.9
k*b	37881.9	---t	30770
HfZ	37784.5	m---	29784.2
Elm	35639.1	--A-	28105.6
kfr	35585.5	-A--	24129.8
...		...	

Table 6: Top Entries in Root and Pattern Lexicons

## 5 Morphological Analysis

A word is analysed into its root and pattern template by considering every possible combination of trilateral root and corresponding pattern pairs,  $(r^x, p^x)$ , as defined in equation 1 for the word,  $w_i$ , in the vocabulary, scoring each analysis with the sum of the scores for the root,  $r^x$ ,

and pattern,  $p^x$ , in the root lexicon and pattern lexicon, respectively. Due to the different ranges of scores for root and pattern, the score for the former is scaled with respect to the latter, as in equation 6, in order to guarantee equal contributions.

$$SS(r) = S(r) \times \frac{\max(S(p))}{\max(S(r))} \quad (\text{Eq. 6})$$

The analysis,  $x$ , with the highest score is selected as the output, as illustrated in equation 7.

$$\max_{x=1..n} (S(r_w^x) + SS(p_w^x)) \quad (\text{Eq. 7})$$

Since we are considering text without diacritics, due to absence of short vowels, we only expect words to contain single letter infixes. Hence we experiment with an alternative configuration of the word decomposition,  $(r^z, p^z)$ : non-contiguous root radicals formed with more than one intervening character are dropped; correspondingly patterns with more than one consecutive character between radical place holder markers are dropped.

## 6 Evaluation

We carry out our evaluation using the Quranic Arabic Corpus (QAC)<sup>3</sup>, since it identifies the root of each word, facilitating the evaluation.

In this section, we first detail some information about our dataset before going onto evaluation of the analyses for correct root extraction.

### 6.1 Data

The QAC consists of approximately 77,900 word tokens, with a total of around 19,000 unique tokens. Since we are interested in investigating learning from undiacritized text, we removed all short vowels and diacritical markers. The size of the resulting vocabulary, after removal of vowels, is approximately 14,850.

We take as input lightly stemmed text, with clitics removed, but with most inflectional markers attached. We assume that stemmed words are obtainable using existing tools for unsupervised concatenative morphology learning. For example, the technique of Poon et al (2009) could be used to obtain accurate stems for each word. The stemmed unvowelled vocabulary size is around 7370.

The original corpus is annotated with roots for all derived and inflected words. More than 95% of words are tagged with their root forms since the

<sup>3</sup> <http://corpus.quran.com/>

Quran consists mostly of words of derivable forms, with very few proper nouns. There are 7192 stemmed words with available roots.

In Arabic, sometimes alterations in root radicals take place; for example, in hollow roots, when moving from a root containing a long vowel to the surface word, the long vowel might change its form to another type or get dropped. Such words with hollow roots or reduplicated radicals, whose characters do not match every radical of the root, were removed from the evaluation as they are beyond the scope of the learning algorithm to identify. Leaving aside these word and root evaluation pairs we evaluated with 5468 stemmed types.

## 6.2 Baseline

As a baseline for evaluation, we derived lexicons in a similar manner to procedure for derivation from clusters (section 5.3). Instead of using clusters we simply scored patterns that matched the largest number of vocabulary words having corresponding roots. Likewise, the root score was obtained by counting the number of words with corresponding patterns.

Comparing our system to the baseline is meant to elucidate the advantage of using the machine learning technique to enhance our lexicons. In the baseline we do not have the ME based word clusters with proximities to the target word; only one cluster exist: the vocabulary set with unit promitiy of 1.

## 6.3 Evaluation of Lexicons

In this section we compare our lexicons, built using maximum entropy modeling approach, (ME), to the baseline(BL).

We evaluated the effect of logarithmic scaling (ME\_LS) comparing it to using raw probability values(ME\_RW). Also we gauged the performance improvement with Length Adjustment (ME\_LS\_LA) for morphemes.

Finally, we evaluated morphological analysis restricted to patterns with single affixes which correspond to roots with single non-contiguous characters from words (ME\_NC1).

We evaluate morphological analysis through correct identification of the root. The accuracy is measured in terms of percentage of the roots that are correctly identified. As stated above, we evaluate on a total of 5468 words. The results for the different configuration evaluations is given in table 7.

Configuration	Total Correct	Percentage Correct
Baseline	4055	74.16
ME_RW	3597	65.78
ME_LS	4415	80.74
ME_LS_LA	4700	85.95
ME_LS_LA_NC1	4768	87.20

Table 7: Evaluation of System Configurations

The accuracy of 74% shows a sound and competitive baseline. The low results for ME\_RW highlights the weakness of considering raw probability values which are too low to provide adequate weightage to morphemes. Hence the dismal performace. The true value for the ME based processing is realized in ME\_LS, where the probabilities have been logarithmically scaled be summing. We see an accuracy gain of 6% over the baseline which is quite significant and encouraging. Further improvements can be seen when the score has been adjusted for morpheme length, ME\_LS\_LA, with performance increase by further 5%. Still more improvement is seen using knowledge of word structure of undiacritized text, ME\_LS\_LS\_NC1, with further accuracy gain of 2.25 %. The final result for ME based analysis with further enhancements gives an promising accuracy result of 87.20%.

## 7 Conclusion and Future directions

In this paper we have presented an approach to solve the problem of learning intercalated morphology in an unsupervised manner with no parameter settings and minimal linguistic knowledge. We applied the machine learning based techniques to learn clusters of words related on basis of either root or pattern morpheme. Thereafter, plausible morphemes are extracted using a scoring method which takes advantage of knowledge of word proximities from clusters built using a maximum entropy classifier. We further apply enhancements to the procedure by accommodating for length and structure of morphemes. The finalized procedure offers significant boost in performance.

The dynamicity of the technique allows its applicability to other types of morphological structures. Also, the system can easily be extended to cater to roots beyond tri-literals by adapting the soring function to accommodate for morpheme length.

## References

- Alexander Clark. 2007. Supervised and unsupervised learning of Arabic morphology. *Arabic Computational Morphology*, volume 38 of *Text, Speech and Language Technology*, pages 181-200.
- Tagyoung Chung and Daniel Gildea. 2009. Unsupervised tokenization for machine translation. In Conference on *Empirical Methods in Natural Language Processing (EMNLP)*.
- Mathias Creutz and Krista Lagus. 2005. Inducing the morphological lexicon of a natural language from unannotated text. In *Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR '05)*, 106-113.
- Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing*, 4(1-3):1-33.
- Ezra Daya, Dan Roth, and Shuly Wintner. 2008. Identifying Semitic roots: Machine learning with linguistic constraints. *Computational Linguistics*, 34:429-448.
- Percy Liang and Dan Klein. 2009. Online EM for unsupervised models. In *North American Association for Computational Linguistics (NAACL)*.
- Guy De Pauw and Peter Wagacha. 2007. Bootstrapping morphological analysis of Gikuyu using unsupervised maximum entropy learning. In *Proceedings of the Eighth Annual Conference of the International Speech Communication Association*. Antwerp, Belgium.
- John Goldsmith. 2000. Linguistica: An automatic morphological analyser. In *Proceedings of the 36th Meeting of the Chicago Linguistic Society*. 125-139.
- John Goldsmith. 2006. An algorithm for the unsupervised learning of morphology. *Natural Language Engineering*, 12(4):353-371.
- Margaret A. Hafer and Stephen F. Weiss. 1974. Word segmentation by letter successor varieties. *Information Storage and Retrieval*, 10(11-12):371-385.
- Harold Hammarström and Lars Borin. 2011. Unsupervised learning of morphology. *Computational Linguistics* 37 (2): 309-350.
- Hoifung Poon, Colin Cherry, and Kristina Toutanova. 2009. Unsupervised morphological segmentation with log-linear models. *Proceedings of NAACL '09: The 2009 Annual Conference of the North American Association for Computational Linguistics*, pages 209-217, Morristown, NJ.
- Paul Rodrigues and Damir Čavar. 2005. Learning Arabic morphology using information theory. In *Proceedings of the Chicago Linguistics Society. Vol 41*. Chicago: University of Chicago. 49-58.
- Patrick Schone and Daniel Jurafsky. 2001. Knowledge-free induction of inflectional morphologies. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, Pittsburgh, PA, 183-191.
- Aris Xanthos. 2008. *Apprentissage automatique de la morphologie: Le cas des structures racine-schème*. Berne, Switzerland: Peter Lang.
- David Yarowsky and Richard Wicentowski. 2000. Minimally supervised morphological analysis by multimodal alignment. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, 207-216.