Memory-Based Learning of Morphology with Stochastic Transducers

Alexander Clark
ISSCO / TIM
University of Geneva
UNI-MAIL, Boulevard du Pont-d'Arve,
CH-1211 Genève 4,
Switzerland
Alex.Clark@issco.unige.ch

Abstract

This paper discusses the supervised learning of morphology using stochastic transducers, trained using the Expectation-Maximization (EM) algorithm. Two approaches are presented: first, using the transducers directly to model the process, and secondly using them to define a similarity measure, related to the Fisher kernel method (Jaakkola and Haussler, 1998), and then using a Memory-Based Learning (MBL) technique. These are evaluated and compared on data sets from English, German, Slovene and Arabic.

1 Introduction

Finite-state methods are in large part adequate to model morphological processes in many languages. A standard methodology is that of two-level morphology (Koskenniemi, 1983) which is capable of handling the complexity of Finnish, though it needs substantial extensions to handle non-concatenative languages such as Arabic (Kiraz, 1994). These models are primarily concerned with the mapping from *deep* lexical strings to surface strings, and within this framework learning is in general difficult (Itai, 1994). In this paper I present algorithms for learning the finite-state transduction between pairs of uninflected and inflected words. – supervised learning of morphology. The techniques presented here are, however, applicable to learning other types of string transductions.

Memory-based techniques, based on principles of non-parametric density estimation, are a powerful form of machine learning well-suited to natural language tasks. A particular strength is their ability to model both general rules and specific exceptions in a single framework (van den Bosch and Daelemans, 1999).

However they have generally only been used in supervised learning techniques where a class label or tag has been associated to each feature vector. Given these manual or semi-automatic class labels, a set of features and a pre-defined distance function new instances are classified according to the class label of the closest instance. However these approaches are not a complete solution to the problem of learning morphology, since they do not directly produce the transduction. The problem must first be converted into an appropriate feature-based representation and classified in some way. The techniques presented here operate directly on sequences of atomic symbols, using a much less articulated representation, and much less input information.

2 Stochastic Transducers

It is possible to apply the EM algorithm to learn the parameters of stochastic transducers, (Ristad, 1997; Casacuberta, 1995; Clark, 2001a). (Clark, 2001a) showed how this approach could be used to learn morphology by starting with a randomly initialized model and using the EM algorithm to find a local maximum of the joint probabilities over the pairs of inflected and uninflected words. In addition rather than using the EM algorithm to optimize the joint probability it would be possible to use a gradient descent algorithm to maximize the conditional probability.

The models used here are Stochastic Non-Deterministic Finite-State Transducers (FST), or Pair Hidden Markov Models (Durbin et al., 1998), a name that emphasizes the similarity of the training algorithm to the well-known Forward-Backward training algorithm for Hidden Markov Models.

Instead of outputting symbols in a single stream, however, as in normal Hidden Markov Models they output them on two separate streams, the *left* and *right* streams. In general we could have different left and right alphabets; here we assume they are the same. At each transition the FST may output the same symbol on both streams, a symbol on the left stream only, or a symbol on the right stream only. I call these q_{11}, q_{10} and q_{01} outputs respectively. For each state s the sum of all these output parameters over the alphabet A must be one.

$$\sum_{c \in A} q_{11}(c|s) + q_{10}(c|s) + q_{01}(c|s) = 1$$

Since we are concerned with finite strings rather than indefinite streams of symbols, we have in addition to the normal initial state s_0, an explicit end state s_1, such that the FST terminates when it enters this state. The FST then defines a joint probability distribution on pairs of strings from the alphabet. Though we are more interested in stochastic transductions, which are best represented by the conditional probability of one string given the other, it is more convenient to operate with models of the joint probability, and then to derive the conditional probability as needed later on.

It is possible to modify the normal dynamic-programming training algorithm for HMMs, the Baum-Welch algorithm (Baum and Petrie, 1966) to work with FSTs as well. This algorithm will maximize the joint probability of the training data.

We define the *forward* and *backward* probabilities as follows. Given two strings $u_1, \ldots u_l$ and $v_1, \ldots v_m$ we define the forward probabilities $\alpha_s(i,j)$ as the probability that it will start from s_0 and output $u_1, \ldots, u_i$ on the left stream, and $v_1, \ldots, v_j$ on the right stream and be in state s, and the backward probabilities $\beta_s(i,j)$ as the probability that starting from state s it will output $u_{i+1}, \ldots, u_l$, on the right and $v_{j+1}, \ldots, v_m$ on the left and then terminate, ie end in state s_1.

We can calculate these using the following recurrence relations:

$$\begin{aligned} \alpha_s(i,j) = & \sum_{s'} \alpha_{s'}(i, j-1) p(s|s') q_{01}(v_j|s) \\ & + \sum_{s'} \alpha_{s'}(i-1, j) p(s|s') q_{10}(u_i|s) + \\ & \sum_{s'} \alpha_{s'}(i-1, j-1) p(s|s') q_{11}(u_i, v_j|s) \end{aligned}$$

$$\begin{aligned} \beta_s(i,j) = & \sum_{s'} \beta_{s'}(i, j+1) p(s'|s) q_{01}(v_{j+1}|s') \\ & + \sum_{s'} \beta_{s'}(i+1, j) p(s'|s) q_{10}(u_{i+1}|s') + \\ & \sum_{s'} \beta_{s'}(i+1, j+1) p(s'|s) q_{11}(u_{i+1}, v_{j+1}|s') \end{aligned}$$

where, in these models, $q_{11}(u_i, v_j)$ is zero unless u_i is equal to v_j. Instead of the normal two-dimensional trellis discussed in standard works on HMMs, which has one dimension corresponding to the current state and one corresponding to the position, we have a three-dimensional trellis, with a dimension for the position in each string. With these modifications, we can use all of the standard HMM algorithms. In particular, we can use this as the basis of a parameter estimation algorithm using the expectation-maximization theorem. We use the forward and backward probabilities to calculate the expected number of times each transition will be taken; at each iteration we set the new values of the parameters to be the appropriately normalized sums of these expectations.

Given a FST, and a string u, we often need to find the string v that maximizes $p(u, v)$. This is equivalent to the task of finding the most likely string generated by a HMM, which is NP-hard (Casacuberta and de la Higuera, 2000), but it is possible to sample from the conditional distribution $p(v|u)$, which allows an efficient stochastic computation. If we consider only what is output on the left stream, the FST is equivalent to a HMM with null transitions corresponding to the q_{01} transitions of the FST. We can remove these using standard techniques and then use this to calculate the *left backward* probabilities for a particular string u: $\beta_s^L(i)$ defined as the probability that starting from state s the FST generates $u_{i+1}, \ldots, u_l$ on the left and terminates. Then if one samples from the FST, but weights each transition by the appropriate left backward probability, it will be equivalent to sampling from the conditional distribution of $P(v|u)$. We can then find the string v that is most likely given u, by generating randomly from $p(v|u)$. After we have generated a number of strings, we can sum $p(v|u)$ for all the observed strings; if the difference between this sum and 1 is less than the maximum value of $p(v|u)$ we know we have found the most likely v. In practice, the distributions we are interested in often have a v with $p(v|u) > 0.5$; in this case we immediately know that we have found the maximum.

We then model the morphological process as a transduction from the lemma form to the inflected form, and assume that the model outputs for each input, the output with highest conditional or joint probability with respect to the model. There are a number of reasons why this simple approach will not work: first, for many languages the inflected form is lexically not phonologically specified and thus the model will not be able to identify the correct form; secondly, modelling all of the irregular exceptions in a single transduction is computationally intractable at the moment. One way to improve the efficiency is to use a mixture of models as discussed in (Clark, 2001a), each corresponding to a morphological paradigm. The productivity of each paradigm can be directly modelled, and the class of each lexical item can again be memorized.

There are a number of criticisms that can be made of this approach.

- Many of the models produced merely memorize a pair of strings – this is extremely inefficient.

- Though the model correctly models the productivity of some morphological classes, it models this directly. A more satisfactory approach would be to have this arise naturally as an emergent property of other aspects of the model.

- These models may not be able to account for some psycho-linguistic evidence that appears to require some form of *proximity* or similarity.

In the next section I shall present a technique that addresses these problems.

3 Fisher Kernels and Information Geometry

The method used is a simple application of the information geometry approach introduced by (Jaakkola and Haussler, 1998) in the field of bio-informatics. The central idea is to use a generative model to extract finite-dimensional features from a symbol sequence. Given a generative model for a string, one can use the sufficient statistics of those generative models as features. The vector of sufficient statistics can be thought of as a finite-dimensional representation of the sequence in terms of the model. This transformation from an unbounded sequence of atomic symbols to a finite-dimensional real vector is very powerful and allows the use of Support Vector Machine techniques for classification. (Jaakkola and Haussler, 1998) recommend that instead of using the sufficient statistics, that the Fisher scores are used, together with an inner product derived from the Fisher information matrix of the model. The Fisher scores are defined for a data point x and a particular model as

$$U_x^i = \frac{\partial \log p(x;\theta)}{\partial \theta_i} \tag{1}$$

The partial derivative of the log likelihood is easy to calculate as a byproduct of the E-step of the EM algorithm, and has the value for HMMs (Jaakkola et al., 2000) of

$$U_x^i = \frac{E[z_i|x]}{\theta_i} - E[s_j|x] \tag{2}$$

where z_i is the indicator variable for the parameter i, and s_j is the indicator value for the state j where z_i leaves state j; the last term reflects the constraint that the sum of the parameters must be one.

The kernel function is defined as

$$K(x,y) = U_x I_\theta^{-1} U_y \tag{3}$$

where I_θ is the Fisher information matrix.

This kernel function thus defines a distance between elements,

$$d(x,y) = (K(x,x) - 2K(x,y) + K(y,y))^{1/2} \tag{4}$$

This distance in the feature space then defines a pseudo-distance in the example space.

The name information geometry which is sometimes used to describe this approach derives from a geometrical interpretation of this kernel. For a parametric model with k free parameters, the set of all these models will form a smooth k-dimensional manifold in the space of all distributions. The curvature of this manifold can be described by a Riemannian tensor – this tensor is just the expected Fisher information for that model. It is a tensor because it transforms properly when the parametrization is changed.

In spite of this compelling geometric explanation, there are difficulties with using this approach directly. First, the Fisher information matrix cannot be calculated directly, and secondly in natural language applications, unlike in bio-informatic applications we have the perennial problem of data sparsity, which means that unlikely events occur frequently. This means that the scaling in the Fisher scores gives extremely high weights to these rare events, which can skew the results. Accordingly this work uses the unscaled sufficient statistics. This is demonstrated below.

4 Details

Given a transducer that models the transduction from uninflected to inflected words, we can extract the sufficient statistics from the model in two ways. We can consider the statistics of the joint model $p(u, v|\Theta)$ or the statistics of the conditional model $p(v|u, \Theta)$. Here we have used the conditional model, since we are interested primarily in the change of the stem, and not the parts of the stem that remain unchanged. It is thus possible to use either the features of the joint model or of the conditional model, and it is also possible to either scale the features or not, by dividing by the parameter value as in Equation 2. The second term in Equation 2 corresponding to the normalization can be neglected. We thus have four possible features that are compared on one of the data sets in Table 4. Based on the performance here we have chosen the unscaled conditional sufficient statistics for the rest of the experiments presented here, which are calculated thus:

$$C_i(< u, v >) = E[z_i| < u, v >] - E[z_i|u] \quad (5)$$

v	$p(v\|u)$	d	Closest
`6pl3Id`	0.313	1.46	`pl3 pl3d`
`6pl3d`	0.223	0.678	`s6pl3 s6pl3d`
`6pld`	0.0907	1.36	`s6pl3 s6pl3d`
`6pl3It`	0.0884	1.67	`p6f p6ft`
`6pl3t`	0.0632	1.33	`p6f p6ft`

Table 1: Example of the MBL technique for the past tense of *apply (6pl3)*. This example shows that the most likely transduction is the suffix *Id*, which is incorrect, but the MBL approach gives the correct result in line 2.

Given an input string u we want to find the string v such that the pair u, v is very close to some element of the training data. We can do this in a number of different ways. Clearly if u is already in the training set then the distance will be minimized by choosing v to be one of the outputs that is stored for input v; the distance in this case will be zero. Otherwise we sample repeatedly (here we have taken 100 samples) from the conditional distribution of each of the submodels. This in practice seems to give good results, though there are more principled criteria that could be applied.

We give a concrete example using the LING English past tense data set described below. Given an unseen verb in its base form, for example *apply*, in phonetic transcription `6pl3`, we generate 100 samples from the conditional distribution. The five most likely of these are shown in Table 1, together with the conditional probability, the distance to the closest example and the closest example.

We are using a k-nearest-neighbor rule with $k = 1$, since there are irregular words that have completely idionsyncratic inflected forms. It would be possible to use a larger value of k, which might help with robustness, particularly if the token frequency was also used, since irregular words tend to be more common.

In summary the algorithm proceeds as follows:

- We train a small Stochastic Transducer on the pairs of strings using the EM algorithm.

- We derive from this model a distance function between two pairs of strings that is sensitive to the properties of this transduction.

- We store all of the observed pairs of strings.
- Given a new word, we sample repeatedly from the conditional distribution to get a set of possible outputs.
- We select the output such that the input/output pair is closest to one of the oberved pairs.

5 Experiments

5.1 Data Sets

The data sets used in the experiments are summarized in Table 2. A few additional comments follow.

LING These are in UNIBET phonetic transcription.

EPT In SAMPA transcription. The training data consists of all of the verbs with a non-zero lemma spoken frequency in the 1.3 million word CO-BUILD corpus. The test data consists of all the remaining verbs. This is intended to more accurately reflect the situation of an infant learner.

GP This is a data set of pairs of German nouns in singular and plural form prepared from the CELEX lexical database.

NAKISA This is a data set prepared for (Plunkett and Nakisa, 1997). Its consists of pairs of singular and plural nouns, in Modern Standard Arabic, randomly selected from the standard Wehr dictionary in a fully vocalized ASCII transcription. It has a mixture of broken and sound plurals, and has been simplified in the sense that rare forms of the broken plural have been removed.

5.2 Evaluation

Table 4 shows a comparison of the four possible feature sets on the Ling data. We used 10-fold cross validation on all of these data sets apart from the EPT data set, and the SLOVENE data set; in these cases we averaged over 10 runs with different random seeds. We compared the performance of the models evaluated using them directly to model the transduction using the conditional likelihood (CL) and using the MBL approach with the unscaled conditional features. Based on these results, we used the unscaled conditional features; subsequent experiments confirmed that these performed best.

	Unscaled	Scaled
Joint	75.3 (3.5)	78.2 (3.6)
Conditional	85.8 (2.4)	23.8 (3.6)

Table 4: Comparison of different metrics on the LING data set with 10 fold cross validation, 1 10-state model trained with 10 iterations. Mean in % with standard deviation in brackets.

The results are summarized in Table 3. Run-times for these experiments were from about 1 hour to 1 week on a current workstation. There are a few results to which these can be directly compared; on the LING data set, (Mooney and Califf, 1995) report figures of approximately 90% using a logic program that learns decision lists for suffixes. For the Arabic data sets, (Plunkett and Nakisa, 1997) do not present results on modelling the transduction on words not in the training set; however they report scores of 63.8% (0.64%) using a neural network classifier. The data is classified according to the type of the plural, and is mapped onto a syllabic skeleton, with each phoneme represented as a bundle of phonological features. for the data set SLOVENE, (Manandhar et al., 1998) report scores of 97.4% for FOIDL and 96.2% for CLOG. This uses a logic programming methodology that specifically codes for suffixation and prefixation alone. On the very large and complex German data set, we score 70.6%; note however that there is substantial disagreement between native speakers about the correct plural of nonce words (Köpcke, 1988). We observe that the MBL approach significantly outperforms the conditional likelihood method over a wide range of experiments; the performance on the training data is a further difference, the MBL approach scoring close to 100%, whereas the CL approach scores only a little better than it does on the test data. It is certainly possible to make the conditional likelihood method work rather better than it does in this paper by paying careful attention to convergence criteria of the models to avoid overfitting, and by smoothing the models carefully. In addition some sort of model size selection must be used. A major advantage of the MBL approach is that it works well without re-

Label	Language	Source	Description	Total Size	Train	Test
LING	English	(Ling, 1994)	Past tense	1394	1251	140
EPT	English	CELEX	Past tense	5324	1957	3367
GP	German	CELEX	noun plural	16970	15282	1706
NAKISA	Arabic	(Plunkett and Nakisa, 1997)	plural	859	773	86
MCCARTHY	Arabic	(McCarthy and Prince, 1990)	broken plural	3261	2633	293
SLOVENE	Slovene	(Manandhar et al., 1998)	genitive nouns	921	608	313

Table 2: Summary of the data sets.

Data Set	CV	Models	States	Iterations	CL	MBLSS
LING	10	1	10	10	61.3 (4.0)	85.8 (2.4)
	10	2	10	10	72.1 (2.0)	79.3 (3.3)
EPT	No	1	10	10	59.5 (9.4)	93.1 (2.1)
NAKISA	10	1	10	10	0.6 (0.8)	15.4 (3.8)
	10	5	10	10	9.2 (2.9)	31.0 (6.1)
	10	5	10	50	11.3 (3.3)	35.0 (5.3)
GP1	10	1	10	10	42.5 (0.8)	70.6 (0.8)
MCCARTHY	10	5	10	10	1.6 (0.6)	16.7 (1.8)
SLOVENE	No	1	10	10	63.6 (28.6)	98.9 (0.8)

Table 3: Results. CV is the degree of cross-validation, Models determines how many components there are in the mixture, CL gives the percentage correct using the conditional likelihood evaluation and MBLSS, using the Memory-based learning with sufficient statistics, with the standard deviation in brackets.

quiring extensive tuning of the parameters.

In terms of the absolute quality of the results, this depends to a great extent on how phonologically predictable the process is. When it is completely predictable, as in SLOVENE the performance approaches 100%; similarly a large majority of the less frequent words in English are completely regular, and accordingly the performance on EPT is very good. However in other cases, where the morphology is very irregular the performance will be poor. In particular with the Arabic data sets, the NAKISA data set is very small compared to the complexity of the process being learned, and the MCCARTHY data set is rather noisy, with a large number of erroneous transcriptions. With the German data set, though it is quite irregular, and the data set is not frequency-weighted, so the frequent irregular words are not more likely to be in the training data, there is a lot of data, so the algorithm performs quite well.

5.3 Cognitive Modelling

In addition to these formal evaluations we examined the extent to which this approach can account for some psycho-linguistic data, in particular the data collected by (Prasada and Pinker, 1993) on the mild productivity of irregular forms in the English past tense. Space does not permit more than a rather crude summary. They prepared six data sets of 10 pairs of nonce words together with regular and irregular plurals of them: a sequence of three data sets that were similar to, but progressively further away from sets of irregular verbs (prototypical- intermediate- and distant- pseudoirregular – PPI IPI and DPI), and another set that were similar to sets of regular verbs (prototypical-, intermediate- and distant- pseudoregular PPR, IPR and DPR). Thus the first data sets contained words like *spling* which would have a vowel change form of *splung* and a regular suffixed form of *splinged*, and the second data sets contained words like *smeeb* with regular *smeebed* and irregular *smeb*. They asked subjects for their opinions on the acceptabilities of the stems, and of the regular (suffixed) and irregular (vowel change) forms. A surprising result of this was that subtracting the rating of the past tense form from the rating of the stem form (in order to control for the varying acceptability of the stem) gave different results for the two data sets. With the pseudo-irregular forms the irregular form got less acceptable as the stems became less like the most similar irregular stems, but with the pseudo-regulars the regular form got more acceptable. This was taken as evidence for the presence of two qualitatively distinct modules in human morphological processing.

In an attempt to see whether the models presented here could account for these effects, we transcribed the data into UNIBET transcription and tested it with the models prepared for the LING data set. We calculated the average negative log probability for each of the six data sets in 3 ways: first we calculated the probability of the stem alone to model the acceptability of the stem; secondly we calculated the conditional probability of the regular (suffixed form), and thirdly we calculated the conditional probability of the irregular (vowel change) form of the word. Then we calculated the difference between the figures for the appropriate past tense form from the stem form. This is unjustifiable in terms of probabilities but seems the most natural way of modelling the effects reported in (Prasada and Pinker, 1993). These results are presented in Table 5. Interestingly we observed the same effect: a decrease in "acceptability" for irregulars, as they became more distant, and the opposite effect for regulars. In our case though it is clear why this happens – the probability of the stem decreases rapidly, and this overwhelms the mild decrease in the conditional probability.

6 Discussion

The productivity of the regular forms is an emergent property of the system. This is an advantage over previous work using the EM algorithm with SFST, which directly specified the productivity as a parameter.

6.1 Related work

Using the EM algorithm to learn stochastic transducers has been known for a while in the biocomputing field as a generalization of edit distance (Allison et al., 1992). The Fisher kernel method has not been used in NLP to our knowledge before though we have noted two recent papers that have some points of similarity. First, (Kazama et al., 2001) derive a Maximum Entropy tagger, by training a HMM and using the most likely state sequence of the HMM as features for the Maximum Entropy tagging model. Secondly, (van den Bosch, 2000) presents an approach that is again similar since it uses rules, induced using a symbolic learning approach as features in a nearest-neighbour approach.

7 Conclusion

We have presented some algorithms for the supervised learning of morphology using the EM algorithm applied to non-deterministic finite-state transducers.

We have shown that a novel Memory-based learning technique inspired by the Fisher kernel method produces high performance in a wide range of languages without the need for fine-tuning of parameters or language specific representations, and that it can account for some psycho-linguistic data. These techniques can also be applied to the unsupervised learning of morphology, as described in (Clark, 2001b).

Acknowledgements

I am grateful to Prof. McCarthy, Ramin Nakisa and Tomaz Erjavec for providing me with the data sets used. Part of this work was done as part of the TMR network *Learning Computational Grammars*. Thanks also to Bill Keller, Gerald Gazdar, Chris Manning, and the anonymous reviewers for helpful comments.

References

L. Allison, C. S. Wallace, and C. N. Yee. 1992. Finite-state models in the alignment of macro-molecules. *Journal of Molecular Evolution*, 35:77–89.

L. E. Baum and T. Petrie. 1966. Statistical inference for probabilistic functions of finite state markov chains. *Annals of Mathematical Statistics*, 37:1559–1663.

Francisco Casacuberta and Colin de la Higuera. 2000. Computational complexity of problems on probabilistic grammars and transducers. In Arlindo L. Oliveira, editor, *Grammatical Inference: Algorithms and Applications*, pages 15–24. Springer Verlag.

F. Casacuberta. 1995. Probabilistic estimation of stochastic regular syntax-directed translation schemes.

Data set	Stem	Suffix	Vowel Change	Past Tense - Stem
PPI	14.8 (0.08)	1.34 (0.04)	8.70 (0.30)	-6.1
IPI	13.9 (0.12)	1.50 (0.13)	10.4 (0.31)	-3.5
DPI	14.2 (0.34)	1.40 (0.07)	17.9 (2.12)	3.7
PPR	13.4 (0.34)	0.58 (0.08)	16.5 (2.18)	-12.8
IPR	19.0 (0.22)	1.02 (0.13)	19.5 (2.22)	-18.0
DPR	21.3 (0.14)	1.14 (0.17)	19.3 (0.94)	-20.2

Table 5: Average negative log-likelihood in nats for the six data sets in (Prasada and Pinker, 1993). Larger figures mean less likely. Standard deviations in brackets.

In *Proceedings of the VIth Spanish Symposium on Pattern Recognition and Image Analysis*, pages 201–207.

Alexander Clark. 2001a. Learning morphology with Pair Hidden Markov Models. In *Proc. of the Student Workshop at the 39th Annual Meeting of the Association for Computational Linguistics*, pages 55–60, Toulouse, France, July.

Alexander Clark. 2001b. Partially supervised learning of morphology with stochastic transducers. In *Proc. of Natural Language Processing Pacific Rim Symposium, NLPRS 2001*, pages 341–348, Tokyo, Japan, November.

R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. 1998. *Biological Sequence Analysis: Probabilistic Models of proteins and nucleic acids*. Cambridge University Press.

Alon Itai. 1994. Learning morphology – practice makes good. In R. C. Carrasco and J. Oncina, editors, *Grammatical Inference and Applications: ICGI-94*, pages 5–15. Springer-Verlag.

T. S. Jaakkola and D. Haussler. 1998. Exploiting generative models in discriminative classifiers. In *Proc. of Tenth Conference on Advances in Neural Information Processing Systems*.

T. S. Jaakkola, M. Diekhans, and D. Haussler. 2000. A discriminative framework for detecting remote protein homologies. *Journal of Computational Biology*, 7(1,2):95–114.

Jun'ichi Kazama, Yusuke Miyao, and Jun'ichi Tsujii. 2001. A maximum entropy tagger with unsupervised hidden markov models. In *Proc. of Natural Language Processing Pacific Rim Symposium (NLPRS 2001)*, pages 333–340, Tokyo, Japan.

George Kiraz. 1994. Multi-tape two-level morphology. In *COLING-94*, pages 180–186.

Klaus-Michael Köpcke. 1988. Schemas in German plural formation. *Lingua*, 74:303–335.

Kimmo Koskenniemi. 1983. *A Two-level Morphological Processor*. Ph.D. thesis, University of Helsinki.

Charles X. Ling. 1994. Learning the past tense of English verbs: The symbolic pattern associator vs. connectionist models. *Journal of Artifical Intelligence Research*, 1:209–229.

S. Manandhar, S. Dzeroski, and T. Erjavec. 1998. Learning multi-lingual morphology with CLOG. In C. D. Page, editor, *Proc. of the 8th International Workshop on Inductive Logic Programming (ILP-98)*. Springer Verlag.

J. McCarthy and A. Prince. 1990. Foot and word in prosodic morphology: The Arabic broken plural. *Natural Language and Linguistic Theory*, 8:209–284.

Raymond J. Mooney and Mary Elaine Califf. 1995. Induction of first-order decision lists: Results on learning the past tense of English verbs. *Journal of Artificial Intelligence Research*, 3:1–24.

Kim Plunkett and Ramin Charles Nakisa. 1997. A connectionist model of the Arabic plural system. *Language and Cognitive Processes*, 12(5/6):807–836.

Sandeep Prasada and Steven Pinker. 1993. Generalisation of regular and irregular morphological patterns. *Language and Cognitive Processes*, 8(1):1–56.

Eric Sven Ristad. 1997. Finite growth models. Technical Report CS-TR-533-96, Department of Computer Science, Princeton University. revised in 1997.

Antal van den Bosch and Walter Daelemans. 1999. Memory-based morphological analysis. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 285–292.

Antal van den Bosch. 2000. Using induced rules as complex features in memory-based language learning. In *Proceedings of CoNLL 2000*, pages 73–78.