San Diego Linguistic Papers **6** (2016) 122–129

Generating morphological paradigms with a recurrent neural network

Robert Malouf
San Diego State University

Abstract

In traditional word-and-paradigm models of morphology, an inflectional system is represented via a set of exemplary paradigms. Novel wordforms are produced by analogy with previously encountered forms. This paper describes a Long Short-Term Memory (LSTM) network which can use this strategy to learn the paradigms of a morphologically complex language. Results are given which show good performance for a range of typologically diverse languages.

1 Introduction

Word-based theories of morphology take the word rather than the morpheme as the smallest meaningful linguistic element and the basic unit of morphological analysis (Blevins, 2006). Traditionally, in this approach a morphological system is represented via a set of complete exemplary paradigms. Individual lexical items to be stored as a set of diagnostic forms or principal parts which allow inflected wordforms can be produced by analogy from the exemplary paradigms.

While rote memorization certainly plays a large role in lexical learning, it is implausible to imagine that speakers of morphologically complex languages simply memorize all the inflected forms of all the lexemes in the vocabulary. In the Samoyedic language Tundra Nenets, for example, each noun has 210 inflected forms, and this is hardly an extreme case. Furthermore, since both lexemes and wordforms follow a Zipfian frequency distribution, speakers will encounter some forms of a few lexemes very frequently, but many forms of many lexemes will be vanishingly rare. It is likely that speakers will be exposed to complete paradigms for few if any lexemes in any given class, and the sets of wordforms that are learned may vary dramatically from speaker to speaker based on each individual's personal linguistic history. The same observations hold for any system which is to derive morphological patterns from a corpus: learners, whether human or computer, must generalize beyond their direct experience.

Ackerman et al. (2009) highlight this issue by posing the Paradigm Cell Filling Problem: Given exposure to an inflected wordform of a novel lexeme, what licenses reliable inferences about the other wordforms in its inflectional family (Ackerman et al., 2009; Ackerman and Malouf, 2013; Blevins et al., in press)? Stump (2001) formalizes the notion of a paradigm via the paradigm function PF, which maps a lexeme and a morphosyntactic feature set to a wordform. We can take a lexeme to be an abstract identifier for a family of related inflected forms; it is similar to a lemma but has no phonological form. A morphosyntactic feature set is a collection of feature values that identify one cell in a lexeme's inflectional paradigm. For example, in English if the morphosyntactic feature set $\sigma = \{\text{TNS:pres}, \text{PER:3}, \text{NUM:sg}\}$, then $\mathrm{PF}(\textsc{walk}, \sigma) =$ *walks* and $\mathrm{PF}(\textsc{be}, \sigma) =$ *is*. Our goal is to learn the paradigm function by observing the value of PF for some lexeme/feature set pairs, thereby providing a solution to the Paradigm Cell Filling Problem.

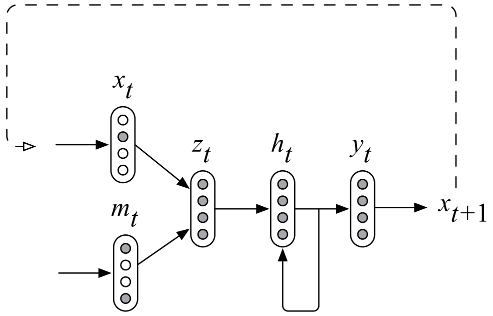

Figure 1: Network architecture

Most of the previous work on learning inflectional morphology has focused on the problem of analysis rather than generation, and many of the systems which can generate novel forms do so by inducing a lexicon of stems and affixes which can be productively combined (Goldsmith, 2006; Kohonen et al., 2010). Approaches which do take paradigms as central have framed the problem as one of generating a set of inflected forms from a single uninflected base form (Dreyer and Eisner, 2011; Durrett and DeNero, 2013; Ahlberg et al., 2014, 2015; Nicolai et al., 2015). This however is founded on an unrealistic assumption: learners often must generalize from a partial set of known forms (which may not happen to include the base form) to a partial set of unknown forms. In many languages, there is no bare'base' form of if there is, it is not especially common. Often, it is not in general possible to uniquely map between base forms and inflected forms in isolation.

These previously published systems make a number of assumptions that are inconsistent with Paradigm Function Morphology, for example that learning is done on the basis of complete paradigms, that words can be exhaustively segmented into meaningful parts (i.e., morphemes), or that all forms of a word can be derived by rule from an underlying base form or lemma. These are useful heuristics for analyzing languages of the familiar European type, but an extensive descriptive and theoretical literature has shown definitively that not all languages are organized along those lines. Since these are not universal properties of morphological systems, they also cannot be the basis for a general theory of morphology.

2 The model

The aim of the model is to simulate a solution to the Paradigm Cell Filling Problem: given knowledge of partial paradigms for a set of lexemes, correctly generate the remaining unobserved forms. Recent work using recurrent neural networks for language modeling (Mikolov et al., 2010, 2012; Mikolov and Zweig, 2012; Sundermeyer et al., 2012, 2015) has shown them to be competitive or superior to standard Markov models. By sampling from the distribution over strings defined by the language model, one can produce plausible-sounding random outputs in a range of domains and modalities (Sutskever et al., 2011; Graves, 2014; Testolin et al., in press). Furthermore, by adding additional inputs, the recurrent neural nets can be made to generate meaningful sequences. For example, Vinyals et al. (2015) use a similar architecture to produce captions from images.

We will use the same basic strategy to generate wordforms for paradigm cells. An overview of the network structure is given in Figure 1. The input x_t is a localist one-hot representation of the previous character and m_t is a 'two-hot' input identifying a lexeme and a paradigm cell: one bit encodes the lexeme (say, WALK) and another encodes the paradigm cell (e.g., {TNS:pres, PER:3, NUM:sg}). These inputs are mapped to a combined projection layer z_t (Bengio et al., 2003):

$$z_t = (W^x x_t + b^x) \oplus (W^m m_t + b^m)$$

where $\oplus$ is vector concatenation. The projection layer z_t in turn is input for the recurrent layer, implemented via Long Short-Term Memory (LSTM) blocks (Hochreiter and Schmidhuber, 1997; Jozefowicz et al., 2015). LSTMs avoid the problems with gradients exhibited by Elman-style simple recurrent networks and allow the model to more easily capture medium and long-distance temporal dependencies in the data (Hochreiter et al., 2001). The output of the recurrent layer h_t is given by:

$$\begin{aligned} i &= \sigma(W^i z_t + U^i h_{t-1} + b^i) \\ f &= \sigma(W^f z_t + U^f h_{t-1} + b^f) \\ o &= \sigma(W^o z_t + U^o h_{t-1} + b^o) \\ c_t &= f \circ c_{t-1} + i \circ \tanh(W^c z_t + U^c h_{t-1} + b^c) \\ h_t &= o \circ \tanh(c_t) \end{aligned}$$

where $\circ$ denotes element-wise multiplication. For implementation purposes, the sigmoid function σ is evaluated using the 'hard sigmoid', a piecewise-linear approximation (Courbariaux et al., 2015):

$$\sigma(x) = \max(0, \min(1, \frac{x}{5} + \frac{1}{2}))$$

Finally, h_t is mapped to a vector with the same dimensionality as the input x_t from which we can induce a probability distribution over output characters: $y_t = W^y h_t$. The probability that the next character in the output x_{t+1} is the jth character in the character set is computed by applying the softmax function on the output layer:

$$p(x_{t+1} = j | x_1 \ldots x_t) = \frac{\exp(y_t^j)}{\sum_k \exp(y_t^k)}$$

The probability of a wordform $p(x_1 \ldots x_n)$ given a lexeme and paradigm cell is the product of the probabilities of each character given the preceding context. During training, the weights W and U and biases b are selected to maximize the log likelihood of the training data.

To produce a candidate wordform, we use the begin-word marker as x_1 and predict a new character x_2, use x_2 as the input to predict x_3, and so on until the end-word marker is generated. The output of the paradigm function PF is the character string that the model assigns the highest probability, found in the current implementation via beam search.

3 Datasets

To evaluate the appropriateness and performance of the proposed model, paradigms were generated based on full-form lexicons for five morphologically complex and typologically diverse languages: Russian, Finnish, Irish, Maltese, and Khaling. The database for each language consists of a set of paradigm function triples: a lexeme, a paradigm cell identifier, and the corresponding wordform. The Russian and Khaling wordforms are given in phonemic transcription. Lexicons for the other languages use the practical orthography. A special word boundary character is added to the beginning and end of each wordform.

The **Russian** lexicon consists of all inflected forms of the 1,500 most frequent noun lexemes as generated in phonemic transcription (including stress) by a DATR implementation (Corbett and Fraser, 1993; Brown and Hippisley, 2012).[1] Russian is a typical fusional Indo-European language: each noun lexeme has 12 wordforms marking for two numbers and six cases via a single fused suffix. Corbett and Fraser (1993) roughly divide Russian nouns into four declensions which determine the endings that are used. For example, the dative singular of Class II KARTA 'map' is *karte*, while the dative singular of Class I ZAKON 'law' is *zakonu*. Russian nominal inflection is made significantly more complex by a cross-cutting system of stress shift patterns. Nouns can fall into one of four stress shift classes (with several subclasses), and there is no direct correspondence between the classes that determine choice of suffixes and the stress shift classes.

The **Finnish** lexicon is based on a sample of 1,000 noun lexemes from the wiktionary-derived paradigms in Durrett and DeNero (2013). Nouns in Finnish have 29 distinct forms, with suffixes marking 15 cases and two numbers (there is no comitative singular form). There is some stem allomorphy (certain lexemes show a change in the final consonant in some paradigm cells) and a high degree of suffix allomorphy: depending on how one counts, Finnish may have as many as 85 nominal declensions (Thymé, 1993).

The **Irish** lexicon is made up of all inflected forms of the 1,216 noun lexemes in Carnie (2008). Irish nouns occur in up to eight different forms, marking two numbers and four cases. Nouns can be classified into two genders, forty singular declensions, and sixty-five cross-cutting plural types. Carnie's lexicon gives examples of 220 distinct gender/declension/plural class combinations. Morphological categories are marked via one or more of an proclitic definite article, a suffix, an initial consonant alternation, and vowel syncope.

The **Maltese**[2] lexicon contains wordforms of 455 verb lexemes taken from the Maltese dictionary for the Apertium translation system (Forcada et al., 2011). Individual verbs have as many as 38 distinct forms, marking subject agreement and tense/aspect/modality. Maltese verbs fall into distinct morphological systems depending on whether they are of Semitic, Romance, or English origin, and the system for Semitic verbs follows a root-and-pattern organization (Hoberman and Aronoff, 2003). For example, the root of the verb meaning 'break' is *k-s-r*, which combines with a vowel pattern and potentially an affix to form an inflected word: *ksirt* 'I broke', *kisret* 'she broke', *niksru* 'we break'.

Khaling is a Sino-Tibetan language with about 15,000 speakers in Eastern Nepal (Jacques et al., 2012). The Khaling lexicon consists of all inflected forms of a sample of 250 verb lexemes. There are up to 331 forms per verb lexeme (depending on the verb type), for a total of approximately 66,000 wordforms (Walther et al., 2013).[3] Khaling verbal morphology is fairly extensive: each verb form potentially consists of a verbal stem plus a prefix and up to seven suffixes indicating negation, subject/object agreement, and tense/aspect/modality marking. The affixal part of the verbal paradigm is straightforwardly agglutinative, and there is little variation in the affixes between verb lexemes. However, there is a complex system of stem alternations, with some verb lexemes occurring with up to ten variant stems depending on the particular set of affixes in the verb form. For example, the verb lexeme ɦod 'to bring' has ten different stem forms: **ɦod**-*u*, **ɦɵts**-*i*, **ɦoɔç**-

[1] http://networkmorphology.as.uky.edu/sites/default/files/ch23_rusnoms.dmp

[2] https://svn.code.sf.net/p/apertium/svn/trunk/apertium-mlt-ara/apertium-mlt-ara.mlt.dix

[3] https://gforge.inria.fr/frs/download.php/file/35119/khalex-0.0.2.mlex.tgz

	x, y	m	z	h	L
Finnish	26	1,030	132	512	17
Irish	34	1,218	260	512	21
Khaling	34	674	132	256	16
Maltese	31	500	136	512	17
Russian	27	732	132	1,024	21

Table 1: Layer sizes: x, y, m are fixed by the input data, z and h were optimized by random search. L is the length of the longest word in the lexicon.

*ki, ʔi-***ɦoɔ̂n***-ni,* **ɦə̄ːd***-ʉ,* **ɦə̂ːt***-nu,* **ɦôː***-tʌ,* **ɦɵs***-ti,* **ɦə̂***-tɛ, ʔi-***ɦoɔ̂j** (Jacques et al., 2012, 1104). Only the initial consonant remains the same. The patterns of alternations are not completely predictable across lexemes, though Jacques et al. argue that the full set of alternate stems can be produced with knowledge of at most four forms of a given lexeme.

4 Results

The architecture in Figure 1 was implemented in Python using Keras and Theano (Bergstra et al., 2010; Bastien et al., 2012; Chollet, 2015). Model parameters were fitted using RMSprop (Hinton, 2012) for 30 epochs with a mini-batch size of 128. The experiments were performed using Nvidia GRID K520 GPUs on Amazon EC2 g2.2xlarge instances. The number of nodes in each layer for each language is given in Table 1. The dimensions of the inputs x and m are fixed by the data and are the number of characters in the character set and the combined number of lexemes and paradigm cells, respectively. The output y is the same dimensionality as the input x. The sizes of the hidden layers were optimized using random search (Bergstra and Bengio, 2012), though the results were not particularly sensitive to the choice of settings. For the most part the performance differences between hyperparameter values were smaller than the variation between runs with the same settings. L is the length of the longest word in the lexicon (including word boundary markers) and is the depth to which the recurrent layer was unrolled for back-propagation through time (Elman, 1990).

Overtraining was not observed to be a problem, and adding regularization did not improve the results. This is consistent with Daelemans et al.'s (1999) observation that regularization actually harms performance on natural language tasks which involve a large amount of rote learning. What in other domains might look like noise in the data that needs to be generalized away from, in this domain is simply irregular. Learning, for example, that the past tense of PAY is *paid* and not *payed* or the past tense of GO is *went* and not *goed* is not 'overtraining'.

Each network was trained on 90% of the paradigm function triples and then evaluated by having it generate the remaining 10%, and then again using 60% of the wordforms for training and evaluating on 40%. The results are given in Table 2.

Performance is quite good for all of the languages with 90% of the wordforms known and remains good for most as the density of training examples is reduced. In general, it seems to perform better for languages which have a large number of wordforms for each lexeme (Finnish, Khaling). Irish, on the other hand, has the smallest paradigms and the worst performance.

It may seem paradoxical that languages with larger systems are more easily learned, since there

	90% train		60% train	
	acc	sd	acc	sd
Finnish	99.6	0.13	99.2	0.04
Irish	90.7	0.84	72.7	1.34
Khaling	99.2	0.15	91.2	0.07
Maltese	95.0	0.90	88.8	2.56
Russian	96.1	0.53	91.9	0.62

Table 2: Ten-fold mean and standard deviation of wordform accuracy, using 90% or 60% of the wordforms in the lexicon for training with the remaining forms used for testing.

are more different forms that need to be produced. However, in larger systems there are also more forms to draw inferences from and it is less likely that an important diagnostic form or principal part will be unknown. This effect is also seen in the high variance for Irish and Maltese: it matters not just how many forms are in the training data but specifically which forms there are.

5 Conclusions

Models of this type have a range of potential applications, both practical and theoretical, and while preliminary, these results are encouraging. In a sense, it is remarkable that an approach like this works at all, given that the input is merely a lexeme code and contains no phonological information. The next steps will be to test the model on a wider range of languages in order to isolate the properties of languages that lead to good and poor performance.

Other extensions that are possible move beyond Stump's conception of a paradigm function. For one, the lexeme and feature set inputs need not be discrete. If we think of the lexeme as a unit of lexicosemantic distinctiveness, then it makes sense to give lexemes a semantic representation. Replacing the 1-of-M encoding with an embedding along the lines of Mikolov et al. (2013) would allow the model to take advantage of any semantic categories that could help predict a lexeme's inflection class.

The probabilistic nature of the model also leads to natural extensions to address situations in which the paradigm function is not strictly functional. Paradigm gaps occur when a lexeme simply lacks a wordform for a particular paradigm cell, and overabundance arises when there is more than one possible wordform for a cell.

Finally, by manipulating the training data, this model could be used to investigate the properties that real languages have that make them learnable (Ackerman and Malouf, 2013; Bonami, 2013; Blevins et al., in press).

References

Ackerman, F., Blevins, J. P., and Malouf, R. (2009). Parts and wholes: Patterns of relatedness in complex morphological systems and why they matter. In Blevins, J. P. and Blevins, J., editors, *Analogy in Grammar: Form and Acquisition*, pages 54–82. Oxford University Press, Oxford.

Ackerman, F. and Malouf, R. (2013). Morphological organization: The low conditional entropy conjecture. *Language*, 89:429–464.

Ahlberg, M., Forsberg, M., and Hulden, M. (2014). Semi-supervised learning of morphological paradigms and lexicons. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 569–578.

Ahlberg, M., Forsberg, M., and Hulden, M. (2015). Paradigm classification in supervised learning of morphology. In *Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL*, pages 1024–1029.

Bastien, F., Lamblin, P., Pascanu, R., Bergstra, J., Goodfellow, I. J., Bergeron, A., Bouchard, N., and Bengio, Y. (2012). Theano: New features and speed improvements. In *NIPS 2012 Workshop on Deep Learning and Unsupervised Feature Learning*.

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimzation. *Journal of Machine Learning Research*, 13:281–305.

Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., and Bengio, Y. (2010). Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*. Oral Presentation.

Blevins, J. P. (2006). Word-based morphology. *Journal of Linguistics*, 42(3):531–573.

Blevins, J. P., Milin, P., and Ramscar, M. (in press). The Zipfian paradigm cell filling problem. In Kiefer, F., Blevins, J. P., and Bartos, H., editors, *Morphological Paradigms and Functions*. Brill.

Bonami, O. (2013). Towards a robust assessment of implicative relations in inflectional systems. Paper presented at Workshop on computational approaches to morphological complexity. Online: `http://www.llf.cnrs.fr/Gens/Bonami/presentations/Bonami-SMG-Paris-2013.pdf`.

Brown, D. and Hippisley, A. (2012). *Network Morphology*. Cambridge University Press.

Carnie, A. (2008). *Irish Nouns: A Reference Guide*. Oxford University Press, Oxford.

Chollet, F. (2015). Keras. `https://github.com/fchollet/keras`.

Corbett, G. G. and Fraser, N. M. (1993). Network morphology: A DATR account of Russian nominal inflection. *Journal of Linguistics*, 29:113–142.

Courbariaux, M., Bengio, Y., and David, J.-P. (2015). BinaryConnect: Training deep neural networks with binary weights during propagations. In *Proceedings of the 29th Annual Conference on Neural Information Processing Systems (NIPS)*.

Daelemans, W., Van Den Bosch, A., and Zavrel, J. (1999). Forgetting exceptions is harmful in language learning. *Machine Learning*, 34:11–41.

Dreyer, M. and Eisner, J. (2011). Discovering morphological paradigms from plain text using a Dirichlet process mixture model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Durrett, G. and DeNero, J. (2013). Supervised learning of complete morphological paradigms. In *NAACL*.

Elman, J. L. (1990). Finding structure in time. *Cognititive Science*, 14:179–211.

Forcada, M. L., Ginestí-Rosell, M., Nordfalk, J., O'Regan, J., Ortiz-Rojas, S., Pérez-Ortiz, J. A., Sánchez-Martínez, F., Ramírez-Sánchez, G., and Tyers, F. M. (2011). Apertium: a free/open-source platform for rule-based machine translation. *Machine Translation*, 25:127–144.

Goldsmith, J. A. (2006). An algorithm for the unsupervised learning of morphology. *Natural Language Engineering*, 12:353–371.

Graves, A. (2014). Generating sequences with recurrent neural networks. `arXiv:1308.0850v5[cs.NE]`.

Hinton, G. (2012). Lecture 6e: rmsprop: Divide the gradient by a running average of its recent magnitude. `http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf`.

Hoberman, R. D. and Aronoff, M. (2003). The verbal morphology of Maltese: From Semitic to Romance. In Shimron, J., editor, *Language processing and acquisition in languages of Semitic, root-based, morphology*, pages 61–78. John Benjamins, Amsterdam.

Hochreiter, S., Bengio, Y., Frasconi, P., and Schmidhuber, J. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In Kremer, S. C. and Kolen, J. F., editors, *A Field Guide to Dynamical Recurrent Neural Networks*.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9:1735–1780.

Jacques, G., Lahaussois, A., Michailovsky, B., and Rai, D. B. (2012). An overview of Khaling verbal morphology. *Language and Linguistics*, 13(6):1095–1170.

Jozefowicz, R., Zaremba, W., and Sutskever, I. (2015). An empirical exploration of recurrent network architectures. In *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)*, pages 2342–2350.

Kohonen, O., Virpioja, S., and Lagus, K. (2010). Semi-supervised learning of concatenative morphology. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, pages 78–86.

Mikolov, T., Karafiát, M., Burget, L., Černocký, J., and Khudanpur, S. (2010). Recurrent neural network based language model. In *Proceedings of Interspeech*, pages 1045–1048.

Mikolov, T., Sutskever, I., Deoras, A., Le, H.-S., Kombrink, S., and Černocký, J. (2012). Subword language modeling with neural networks. `http://www.fit.vutbr.cz/imikolov/rnnlm/char.pdf`.

Mikolov, T. and Zweig, G. (2012). Context dependent recurrent neural network language model. In *Proceedings of Speech Language Technology*, pages 234–239.

Mikolov, T. M., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*.

Nicolai, G., Cherry, C., and Kondark, G. (2015). Inflection generation as discriminative string transduction. In *Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL*.

Stump, G. (2001). *Inflectional Morphology: A Theory of Paradigm Structure*. Cambridge University Press, Cambridge.

Sundermeyer, M., Schlüter, R., and Ney, H. (2012). LSTM neural networks for language modeling. In *Proc. of Interspeech*.

Sundermeyer, M., Schlüter, R., and Ney, H. (2015). From feedforward to recurrent LSTM neural networks for language modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23:517–529.

Sutskever, I., Martens, J., and Hinton, G. (2011). Generating text with recurrent neural networks. In *International Conference on Machine Learning (ICML 2011)*.

Testolin, A., Stoianov, I., Sperduti, A., and Zorzi, M. (in press). Learning orthographic structure with sequential generative neural networks. *Cognitive Science*.

Thymé, A. (1993). *Connectionist Approach to Nominal Inflection: Paradigm Patterning and Analogy in Finnish*. PhD thesis, UC San Diego.

Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2015). Show and tell: A neural image caption generator. In *Computer Vision and Pattern Recognition*.

Walther, G., Jacques, G., and Sagot, B. (2013). Uncovering the inner architecture of Khaling verbal morphology. In *3rd Workshop on Sino-Tibetan Languages of Sichuan*.

Robert Malouf
San Diego State University
rmalouf@mail.sdsu.edu