Brain and Language **81,** 454–463 (2002)
doi:10.1006/brln.2001.2538, available online at http://www.idealibrary.com on IDEAL®

Whole Word Morphologizer: Expanding the Word-Based Lexicon: A Nonstochastic Computational Approach

Sylvain Neuvel

University of Chicago

Published online December 6, 2001

Whole Word Morphologizer is a small computer implementation of word-based morphology. The program automatically identifies morphological relations in a small word-based lexicon, literally learning its morphology, and uses the knowledge it acquires to generate new words. It is based on a model of the mental lexicon in which all entries are whole, entire, fully fledged words and relies solely on basic cognitive principles (differentiation and generalization) for the automatic acquisition of morphological relations and the population of the lexicon. © 2001 Elsevier Science (USA)

Key Words: computational word-based morphology; automatic acquisition; generation; machine learning.

If we take the enrichment of lexica to be not only the *raison d'être* of morphology but also the central issue of morphological theory, it seems reasonable to evaluate any theory of morphology, not on the means by which it represents recurring partials or lexical relations, but on its ability to generate new words based on a given lexicon. Within the framework of Whole Word Morphology (cf. Ford & Singh, 1991; Ford et al., 1997), I designed a small computer program that identifies morphological relations found in a lexicon and creates new words based on these relations. The purpose of this article is to demonstrate the generative power of this *Whole Word Morphologizer,* to spell out the theory behind it and to discuss some of the theoretical issues that arose during its development. This project is intended as an empirical demonstration that a model of morphology that rejects the notion of morpheme as minimal unit of form and meaning (and/or grammatical properties) is viable from the point of view of acquisition as well as generation. It is based on a model of the mental lexicon in which all entries are whole, entire, fully fledged words and relies solely on basic cognitive principles (differentiation and generalization) for the automatic acquisition of morphological relations and the population of the lexicon.

The Theory

The approach I suggest is characterized, as the long title of this paper implies, by its nonstatistical nature and by its reliance on the word as the basic unit of morphology. Ford and Singh's Whole Word Morphology (cf. Ford & Singh 1991; Ford et

Address correspondence and reprint requests to Sylvain Neuvel, 5125 S. Kenwood Ave. Apt. 408, Chicago, IL, 60615.

0093-934X/01 $35.00

al., 1997; Neuvel & Singh, in press) provides the theoretical backdrop for this application. In Whole-Word Morphology, any and all morphological relations can be represented by a rule of the following form (cf. Singh & Dasgupta, 1999) as follows:

(1) $/ X /_{\alpha} \leftrightarrow / X' /_{\beta}$,

where:

a. $/ X /_{\alpha}$ and $/ X' /_{\beta}$ are words and X and X′ are abbreviations of the forms of classes of words belonging to categories α and β (with which specific words belonging to the right category can be unified or onto which they can be mapped).
b. ′ represents (all the) form-related differences between / X / and / X′ /
c. α and β are categories that may be represented as feature-bundles
d. the ↔ represents a bidirectional implication (if X, then X′ and if X′, then X)
e. X′ is a semantic function of X

The implications of (1) are rather straightforward: (1) There is only one morphology: no distinction, other than a functional one, is made between inflection and derivation; and (2) morphology is relational and not compositional. *Whole Word Morphologizer* (henceforth WWM) thus makes no reference to theoretical constructs such as ''root,'' ''stem,'' and ''morpheme,'' or to devices such as ''levels'' and ''strata'' and relies exclusively on the notion of morphological relatedness. And since its objective is not to assign a probability to a given word or string, it must rely on a strict formal definition of a morphological relation. Unfortunately, it is hard, if not impossible, to find in recent literature any well-defined criterion on the basis of which morphological relations could be identified. The 1940s and 1950s saw many linguists offering what has been referred to as *discovery procedures* (cf. Nida, 1949, for example) but since Chomsky (1965), very few attempts have been made at finding out what is actually part of morphology. What remains constant throughout modern linguistic theories, however, is the idea that one must focus on similarities of form and meaning, or ''recurring partials,'' to identify morphemes or morphological relations, despite the fact that many, if not most words that share form and meaning are not morphologically related (ex: *hand*, *head*, and *heel* are all body parts and begin with /h/).

Neuvel and Singh (in press) argue that morphology may not be a game of similarities between words, as most would have it, but one of differences. Following the Saussurean view that words are defined by the differences among them, they contend that some of these differences are morphologically exploitable and exploited. What makes English words like *completely* and *directly* interesting is not the fact that they look alike (by virtue of sharing *-ly*), but the fact that the difference between $complete_{Adj}$ and $completely_{Adv}$ is exactly the same as that between $direct_{Adj}$ and $directly_{Adv}$; the presence of the substring /*ly*/ at the end of the word $directly_{Adv}$ being no more relevant than the absence of it at the end of $direct_{Adj}$. In this view, the morphology of a speaker consists exclusively of productive differences found in his or her lexicon. In other words:

(2) Two words of a lexicon L are morphologically related if, and only if, they differ in exactly the same way as two other words of L (Neuvel & Singh in press).

The above definition is, it is easy to see, set one degree of abstraction higher than the discovery procedures set forth by the structuralists and shifts the focus from the mere comparison of words to the comparison of comparisons, or differences. The bidirectionality of morphological strategies should also require no further justification given (2): for a word w_1 to differ from word w_2, w_2 must also differ from w_1. While

it may appear somewhat informal, this definition creates very sharp boundaries for morphology and turns the identification of morphological relations from a difficult and subjective endeavor to a precise, well-defined, and almost trivial task. Given four words w_1–w_4, w_1 is said to be morphologically related to w_2 if all the formal, semantic, and grammatical properties that make them distinct also differentiate w_3 and w_4. The definition in (2) clearly adjudicates questions regarding the morphological status of Firthian phonestemes like *gl-* in *gleam* and *glow,* Amritavalli's (1999) ''shapers'' (e.g., *st-* in *stable* and *stability*), or Bender's (1998) ''helpers'' (*-er,* in *spider, water, greater, wander,* etc.). The words *gleam* and *glow,* for example, differ in that one word ends in /ow/ and the other in /ijm/, a contrast which is not exploited anywhere else with the same meaning distinction (cf. Neuvel & Singh, in press).

The Program

Under the assumption that the morphology of a language resides exclusively in differences that are exploited in more than one pair of words within its lexicon, WWM compares every word of a small lexicon and determines the segmental differences found between them. The input to the current version of the program is a small text file that contains anywhere from 1000 to 5000 words. Each word appears in orthographic form and is followed by its syntactic and morphological categories, as in the example below:

(3)

cat,	Ns	(Noun, singular)
catch,	V	
catches,	V3s	(Verb, (pres.) third-person singular)
decided,	Vp	(Verb, past)
etc.		

The word lists used in the development of WWM usually come from texts in electronic format and are labeled manually (The English data used to test WWM consists of the first 2000 forms from *Moby Dick*). For lack of adequate semantic representations, no meaning is assigned to any of the words. In theory, WWM could compare every word with every other word, but to save processing time, only pairs of words that share the first three letters are compared. The necessary number of matching letters is adjustable in the current version of the program.

It should be clear that there is no such thing as the ''right'' comparison algorithm. Searching for prefixes, suffixes, and infixes requires different approaches. For example, if one were to compare the nonexisting word *batu* with the equally nonexisting *bamatu,* one might tentatively conclude that the latter can be formed from the former by adding the sequence *-am-* after the first consonant, or, just as easily, that the sequence *-ma-* is inserted after the first syllable. The correct analysis only becomes apparent when this comparison is in turn compared with others. Things become even more complicated if one wishes to consider reduplication and feature modification. The *batu*/*bamatu* example can in fact be analyzed about a dozen different ways if one considers the possibility that the segment /a/ is reduplicated and that either /b/ or /m/ is both a reduplication and modification of its counterpart. Since WWM was designed to work on European languages first, it limits itself to a very basic algorithm that compares words on a letter-by-letter basis starting from the left or the right. WWM thus easily captures what is usually described as prefixation or suffixation, as well as most cases of modification (those in which the number of segments remains the same) but misses infixes and suppletive morphology altogether. The comparison algorithm WWM uses can easily be expanded to handle templatic morphology like

that of Semitic languages and some cases of partial reduplication. The next version of the program should include these modifications.

The algorithm simply compares each letter from word A to the corresponding one from word B (from the left if the two words share the first segments and from the right if they share the last few) and places them in one of two ''piles'' (differences or similarities) based on whether they are identical. Each comparison also contains the categories of both words and is kept in a large comparison list for further manipulation. The example below shows the comparison produced by the English words *receive* and *reception.*

(4)

Differences		*Similarities*	
Left (First word)	*Right (Second word)*	*Left*	*Right*
####ive$_{V}$	####ption$_{Ns}$	rece###	rece#####

Matching characters in the difference section are replaced with a variable as required by (1). The result is then matched up to comparisons generated by other pairs of words and identical differences are recognized. In the example below, the comparisons produced by the pairs *receive*/*reception, conceive*/*conception* and *deceive*/*deception* are shown.

(5)

Differences		*Similarities*	
Left (First word)	*Right (Second word)*	*Left*	*Right*
X ive$_{V}$	X ption$_{Ns}$	rece###	rece#####
X ive$_{V}$	X ption$_{Ns}$	conce###	conce#####
X ive$_{V}$	X ption$_{Ns}$	dece###	dece#####

The three comparisons in (5) share the same formal and grammatical differences and can be merged into one morphological strategy. Each new morphological strategy is also restricted to apply in as narrow an environment as possible. Neuvel and Singh (*in press*) suggest that any morphological strategy must be maximally restricted at all times; this is accomplished by specifying as constant all the similarities found, not between words, but between the similarities found between words. In (5), all three sets of similarities end with the sequence of letters ''ce'' (if one were to consider phonemes instead of letters, the corresponding sets of similarities would end with an /s/ preceded by exactly one syllable). These similarities between similarities are specified as constant in each strategy and the length of each word is also factored in. The restricted morphological strategy relating the words in (5) is as follows:

(6)

Differences		*Similarities*	
Left (First word)	*Right (Second word)*	*Left*	*Right*
X ive$_{V}$	X ption$_{Ns}$	*##ce###	*##ce#####

For the sake of clarity, we can represent the information contained in (6) in a more familiar fashion using the formalism described in (1). The symbol | is used instead of slashes so as not to confuse orthographic form with phonemic representations.

(7) |*##ceive|$_{V}$ ↔ |*##ception|$_{Ns}$

The # signs in the above representation stand for letters that must be instantiated but are not specified; the * symbol stands for a letter that is not specified and that may or may not be instantiated. Strategy (7) can therefore be interpreted as follows[1]:

(7′) *If there is a verb that ends with the sequence ''ceive'' preceded by no less than two and no more than three characters, there should also be a singular noun that ends with the sequence ''ception'' preceded by the same two or three characters.*

In addition to completely specified or completely unspecified segments, WWM currently generalizes to consonants and vowels and can, for example, restrict a strategy to vowel ending words or to words that end with two consonants. The results are, as one can imagine, less than perfect since orthography often hides the domain of application of strategies and it is unclear if the benefits outweigh the disadvantages. This type of generalization requires the program to be given specific information about which segments are to be called vowels and it seems ideologically preferable, at least to me, to let WWM acquire its knowledge in a completely unsupervised fashion. While the amount of education necessary for WWM to generalize to consonants and vowels is minimal, I believe that WWM can achieve superior results without the benefit of any ''innate'' linguistic knowledge. Rather than specifying the environment of application of a morphological strategy in terms of phonological features or segment classes, the program can just as easily specify the sets of segments that can and do appear in a given position. A given strategy could thus be restricted to words that end or begin with any of the segments {f, (c)h, s, v, z,} without the program knowing anything about the segments in question. This change should be implemented in the next version of the program. Even such a simple modification, however, raises some issues about locality. Since morphological conditioning is usually very local, only one or two segments preceding or following what is usually described as an affix might need to be specified in that fashion but it seems worthwhile to empirically explore the pros and cons of specifying more or even all variable segments as sets of possible characters (or phonemes).

From the list of all performed comparisons, WWM extracts and restricts a series of morphological strategies. Below are a few strategies based on the first few chapters of *Moby Dick.*

(8)

Differences		*Restrictions (similarities)*		
First word	*Second word*	*First word*	*Second word*	*Examples*
Xd_{PP}	X_V	****###Ce#	****###Ce	*baked/bake, charged/charge*
Xed_{PP}	X_V	*C#######	*C#####	*directed/direct*
Xs_{NP}	X_{NS}	******#####	******####	*helmets/helmet, rabbits/rabbit*

[1] As Neuvel and Singh (in press) point out, the |*##| as a variable is doubly grounded: syntagmatically, it ranges over formally similar strings or substrings (''re'' in *receive* and *reception,* ''con'' in *conceive* and *conception,* etc.) and paradigmatically it ranges over formally distinct strings (''re,'' ''con,'' and ''de'').

$Xing_{GER}$	Xed_{PP}	******#######	******######	*walking/walked, talking/talked*
$Xing_{GER}$	Xs_{V3S}	*****C##C###	*****C##C#	*walking/walks, talking/talks*
$Xness_{NS}$	X_{ADJ}	****C##C#####	********C##C#	*short/shortness, kind/kindness*
Xly_{ADV}	X_{ADJ}	******######	******####	*easy/easily, quick/quickly*
$Xest_{ADJ}$	X_{ADJ}	*#V#C###	*#V#C	*hardest/hard, shortest/short*
Xd_{V3S}	X_{V}	***C#####	***C####	*jump/jumps, plays/play*
Xer_{ADJ}	X_{ADJ}	*#V#C##	*#V#C	*harder/hard, louder/loud*
$Xless_{ADJ}$	X_{NS}	*#VC#####	*#VC#	*painless/pain, childless/child*
$Xing_{GER}$	Xy_{ADJ}	*C##C###	*C##C#	*raining/rainy, running/runny*
Xed_{PP}	Xs_{V3S}	**C#V###	**C#V##	*played/plays, cried/cries*
$Xings_{NP}$	X_{V}	***CCV#C####	***CCV#C	*paintings/paint, dealings/deal*
X'_{GER}	Xg_{GER}	*C###in#	*C###in#	*runnin'/running, signin'/signing*
$X's_{POS}$	Xs_{NP}	*****#######	*****######	*neighbor's/neighbors, piano's/piano*

Similarities and differences in a given strategy are merged into a single pattern similar to (7). WWM then goes through the lexicon word by word and attempts to map each word onto either side of this strategy. If it succeeds, WWM replaces all the segments fully specified on the side of the strategy the word is mapped on, with the segments fully specified on the other side. For example, given the word *reception* and strategy (7), WWM will map the word onto the right hand side of (7), take out the sequence ''ception'' from the end, and replace it with the sequence ''ceive.'' The category of the word will also be changed from singular noun to verb. Below are some of the words WWM creates using text from *Le petit prince* as its base lexicon:

(9)

drames;	NP	droitement;	ADV
dressée;	PF	drôles;	AIP
dresser;	INF	drôlement;	ADV
dressa;	VP3	dunes;	NP
dressais;	VI2	durerait;	VC3
dresse;	V3	décidée;	PF

dressent;	V6	décider;	INF
dressez;	V5	décida;	VP3
dressait;	VI3	décide;	V3
droits;	AMP	décoiffé;	AM
droites;	AFP	déconcentrés;	AMP

It should be clear that the method described above will force WWM to create words that were already part of its original lexicon; in fact, each and every word involved in licensing the creation of a morphological strategy will be reduplicated by the program. Contrasting words that were not part of WWM's original lexicon are finally added to a separate word list containing only new words. If needed, this word list can be merged with the original lexicon to create new strategies based on a larger dataset; each new word can also simply be put through another cycle or word creation using the same strategies. WWM will typically be able to create words during five or six cycles by mapping new words onto the strategies created in the first cycle.

Results and Improvements

While some combinatorial restrictions are specified in each strategy, Whole Word Morphology, like most models of word formation, is burdened by overgeneration. The relation between words of two categories is often expressed by two or more competing strategies. For example, when using the text of *Le petit prince* as its base lexicon, WWM produces two strategies relating second-person verb forms to their infinitives. Given the verb *conjugues*, ''conjugate, present tense second singular,'' singular,'' one strategy produces the correct infinitive *conjuguer* while the other creates the word **conjuguere* (based on the relation between words like *fais/faire*, ''do,'' and *vends/vendre*, ''sell''). WWM then creates two, three, or even four formally different ''versions'' of the same word, and since the ''correct'' form is often already found in the original lexicon, only the ''bad'' forms are added to the list, resulting in an error rate of anywhere from 18 to 30% in the first version of the program.[2]

Different solutions to the problem of competing strategies were experimented with and are still part of the current version of WWM. The first and probably most obvious solution to the problem is to implement some version of the so-called Panini's principle, (a.k.a. *Elsewhere condition;* cf. Kiparsky 1973) and to only apply the most restrictive strategy whenever two or more options present themselves. Unfortunately, this solution brings with it a theoretical question which remains to be answered. If a given strategy applies in an environment that is included in the environment of another, it is evidently the more restrictive of the two. But if two strategies apply in mutually exclusive environments or environments that overlap, the question as to which is the most restrictive one is not always so easy to answer. Many theories of word formation make use of some version of Panini's principle but, to my knowledge, no method for evaluating the restrictiveness of a rule or strategy has ever been de-

[2] Since WWM cannot rely on adequate descriptions of word meanings, it could, given the proposed definition of a morphological relation, identify words traditionally described as sharing *pseudoaffixes* as morphologically related. Word pairs like *mot*/*motor* and *rot*/*rotor*, or even pairs like *cat*/*bat* and *car*/*bar*, could give rise to unwanted morphological strategies and even more undesirable words like **tableor* or **bhair*. In a perfect word, the semantic contract between the words above would preclude the creation of a morphological strategy since no other word pair shares the same semantic contrast as *cat* and *bat* or *mot* and *motor*. Fortunately, the datasets WWM used to acquire its morphology are small enough that they rarely include two pairs of morphologically unrelated but similarly contrasting words. Moreover, while Neuvel and Singh's definition requires differences to be shared by two pairs of words, the number of similarly contrasting word pairs required to license a morphological strategy can be adjusted in the program.

vised. For example, in (10) below, it is clear that the environment in a.i. is more restrictive than the environment in a.ii, but it is much harder to rank the environments in (10b) or (10c).

(10) a. i. / *###Ce / ii. / ##XC /
b. i. / **####CC/ ii. / **####Xa /
c. i. / *##CCVC ii. / *##Ce /

During testing, different ''restrictive'' values were assigned to fully specified segments and to consonant or vowel abstractions (C or V). The values settled on are 5 points for a fully specified segment and 2 points for a C or V. This is obviously not an ideal solution. Not only are these numbers not justified in any way, but the resulting ''score'' does not take into account the degree of freedom offered by the environment in terms of the number of segments that can, and must, be instantiated. With this modification, the accuracy rate of WWM increases a mere 2 to 5%

Another method for choosing between competing strategies involves counting the actual amount of word pairs used to license a particular strategy. If, for example, 34 pairs of words in the lexicon share the differential relation $/X/_{Ns} \leftrightarrow /Xs/_{Np}$, a ''strength'' of 34 can be assigned to this strategy and in the event that two strategies compete for the creation of a single word, the stronger (or weaker) strategy can take precedence. There are also a few disadvantages to this option: First, it brings statistics in through the back door and makes WWM more of a hybrid stochastic/analogical model. Perhaps more importantly, the statistics WWM has to use to assign these ''strengths'' or probabilities are unreliable due to the extremely small size of the corpora WWM was designed to work with. Perhaps not surprisingly, the results are even more disappointing than with Panini's principle and it makes no significant difference whether the stronger or the weaker strategy is chosen.

Ironically enough, the most effective solution to the problem of competing strategies comes from turning the problem on its head. Rather than stopping certain strategies from applying, it is possible to let WWM create every possible word, including different versions of the ''same'' one and to let lexical lookup take precedence over productive morphology; in other words, to implement some form of *blocking.* At first glance, the idea may seem dubious: since the words in WWM's lexicon have no meaning assigned to them, there is no way for the program to know if it created a word with the same meaning as one it already knew. This is true, but the knowledge WWM possesses about its lexicon increases considerably during the creation of morphological strategies. As mentioned in the previous section, WWM creates word formation strategies based on morphological relations it recognizes in its lexicon. As a result, the program learns not only which strategies are licensed by a given lexicon, but also which words of its lexicon are related to one another. It is then possible to reinterpret the notion of morphological relatedness to mean something like ''semantically related'' or, more adequately, as ''belonging to the same paradigm'' if one subscribes to a paradigmatic organization of the lexicon. After identifying morphologically related pairs of words, WWM assigns a number to every lexical entry and gives the same number to every related word. Below is a sample from the lexicon created using *Moby Dick* after WWM has assigned paradigm numbers to every word.

(11)

deck;	NS;	487	defaced;	PP;	494
decks;	NP;	487	deformed;	PP;	495
decoction;	NS;	489	degree;	NS;	496
deep;	ADJ;	490	deity;	NS;	497
deeper;	ADJ;	490	delay;	V;	498
deepest;	ADJ;	490	deliberate;	ADJ;	499
deeply;	ADV;	490	deliberately;	ADV;	499

Once a paradigm number has been assigned to every word, WWM creates as many words as it can using the strategies it created. Before a new word is added to the lexicon, the program first looks for a word with the same category and paradigm number in its original lexicon. For example, if WWM maps the word *decoction* from (10) onto a strategy creating plural nouns, it will look for a plural noun belonging to paradigm 489 in its lexicon before it adds *decoctions* to the list of new words. The results are less than perfect but this simple modification cut WWM's error rate in half and brings its accuracy to anywhere from 85 to 92% during testing.[3]

Moreover, some of the errors left over after this modification are quite interesting and somewhat parallel to common speech or, in this case, orthographic mistakes. For example, when using the text of *Le petit prince* as input, WWM will sometimes use the wrong accent or omit a cedilla or double ''l''s or ''p''s when they should not be. Below are a few examples:

(12)

absurdement;	ADV	appeller;	INF
achete;	V3	avanca;	VP3
acheve;	V3	boutone;	V3
acheverez;	VF5	brevetes;	V2
adieus;	NP	commençé;	PM
annoncait;	VI3	commodement;	ADV
aplombe;	V3	crayone;	V3

The reader can draw his or her own conclusions regarding these results, for there is unfortunately, very little if nothing, to weigh them against. Most morphological learners (cf. Goldsmith, 2001; Baroni, 2000, amongst others) work on large corpora and are limited to morphological parsing and/or the identification of morphemes. Applications of two-level morphology like PC-Kimmo (cf. Koskenniemi, 1983; Antworth, 1990) can both parse and generate words but must be provided with a set of rules and a lexicon containing morphemes and morphosyntactic constraints by the user. In other words, it must already know the morphology of the language before it can parse or generate anything. Like stochastic models, WWM literally learns from the lexicon it is given and creates a morphological analysis using nothing but full-fledged words. Like PC-Kimmo, WWM can generate words but it does so using the knowledge it acquired on its own, not a morphological analysis provided by someone else. I am unaware of any other project that goes beyond analyzing a simple word list to create new ones, but the accuracy of WWM in generation is easily comparable with what any of the projects mentioned above achieves in analysis.

Conclusions

The computational approach to morphological learning outlined in this article uses no statistics and no theoretical constructs such as morpheme, stem, or affix. It offers substantial evidence that a radically minimal word-based approach to morphology can be formalized enough to be implemented computationally and shows that analogical methods that are applied to reasonably sized datasets can match or even outperform stochastic methods requiring colossal training corpora. The practical applications of the program are quite obvious. Most projects that require a word list would like to see that list grow to its full potential. While a simple survey of the Internet can supply anyone with as many forms as one could possibly need, WWM provides syntactic and morphological labels for every word it creates. The Internet can then

[3] These results are still preliminary and the program needs to be tested on more corpora from different languages.

be used to verify that a form created by the program is attested. Whole Word Morphologizer suggests that merely cutting words one already knows into pieces is not enough and that a model of morphology, be it a computational one or not, should be judged not only from the point of view of acquisition but also on its ability to enrich the lexicon. From a cognitive point of view, WWM provides evidence that a truly minimalist theory of morphology coupled with a full-entry-theory of the mental lexicon elegantly accounts for both acquisition and generation in a uniform fashion using only basic cognitive principles such as differentiation and generalization. More information on the theory behind the program, as well as a demo version of Whole Word Morphologizer can be found online at http://www.neuvel.net/linguistics.htm.

REFERENCES

Amritavalli, R. (1999). Review of Ford, Singh, and Martohardjono, 1997, Pace Panini. *Indian Linguistics*, **59**.

Antworth, E. L. (1990). *PC-KIMMO: A two-level processor for morphological analysis.* Occasional Publications in Academic Computing No. 16. Dallas, TX: Summer Institute of Linguistics.

Baroni, M. (2000). An automated distribution-driven prefix learner. Paper presented at the 9th International Morphology Meeting, Vienna, Austria, February.

Bender, B. (1998). The sign gravitates to the word. In M. Janse (Ed.), *Productivity and creativity* (pp. 15–24). Berlin: Mouton.

Chomsky, N. (1965). *Aspects of the theory of syntax*. Cambridge, MA: MIT Press.

Ford, A., & Singh, R. (1991). Propedeutique Morphologique. *Folia Linguistica*, 25 (3–4), 549–575.

Ford, A., Singh, R., & Martohardjono, G. (1997). *Pace Panini*. New York: Peter Lang.

Goldsmith, J. (2001). Linguistica: An automatic morphological analyzer. In CLS 36, The Main Session. Proceedings from the 36th Meeting of the Chicago Linguistic Society. Chicago: Chicago Linguistic Society.

Kiparsky, P. (1973) Elsewhere in phonology. In S. Anderson & P. Kiparsky (Eds.), *A Festchrift for Morris Halle*. New York: Holt, Rinehart and Winston.

Koskenniemi, K. (1983). *Two-level morphology: A general computational model for word-form recognition and production.* Publication No. 11. University of Helsinki: Department of General Linguistics.

Neuvel, S., & Singh, R. (in press). *Vive la différence! What morphology is about.* Folia Linguistica.

Nida, E., (1949). *Morphology: The descriptive analysis of words.* Ann Arbor, MI: The Univ. of Michigan Press.

Singh, R., & Dasgupta, P. (1999). On so-called compounds. In R. Singh (Ed.), *Yearbook of South Asian Languages and Linguistics 1999* (pp. 265–275). Thousand Oaks, CA: Sage.