A Framework for Unsupervised Natural Language Morphology Induction

Christian Monson
Language Technologies Institute
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA, USA 15213
cmonson@cs.cmu.edu

Abstract

This paper presents a framework for unsupervised natural language morphology induction wherein candidate suffixes are grouped into candidate inflection classes, which are then arranged in a lattice structure. With similar candidate inflection classes placed near one another in the lattice, I propose this structure is an ideal search space in which to isolate the true inflection classes of a language. This paper discusses and motivates possible search strategies over the inflection class lattice structure.

1 Introduction

Many natural language processing tasks, including parsing and machine translation, frequently require a morphological analysis of the language(s) at hand. The task of a morphological analyzer is to identify the lexeme, citation form, or inflection class of surface word forms in a language. Striving to bypass the time consuming, labor intensive task of constructing a morphological analyzer by hand, unsupervised morphology induction techniques seek to automatically discover the morphological structure of a natural language through the analysis of corpora.

This paper presents a framework for automatic natural language morphology induction inspired by the traditional and linguistic concept of inflection classes. Monson et al. (2004) uses the framework discussed in this paper and presents results using an intuitive baseline search strategy. This paper presents a discussion of the candidate inflection class framework as a generalization of corpus tries used in early work (Harris, 1955; Harris, 1967; Hafer and Weiss, 1974) and discusses an as yet unimplemented statistically motivated search strategy. This paper employs English to illustrate its main conjectures and a Spanish newswire corpus of 40,011 tokens and 6,975 types for concrete examples.

2 Previous Work

It is possible to organize much of the recent work on unsupervised morphology induction by considering the bias each approach has toward discovering morphologically related words that are also orthographically similar. Yarowsky et al. (2001), who acquire a morphological analyzer for a language by projecting the morphological analysis of a second language onto the first through a clever application of statistical machine translation style word alignment probabilities, place no constraints on the orthographic shape of related word forms.

Next along the spectrum of orthographic similarity bias is the work of Schone and Jurafsky (2000; 2001), who first acquire a list of potential morphological variants using an orthographic similarity technique due to Gaussier (1999) in which pairs of words with the same initial string are identified. They then apply latent semantic analysis (LSA) to score the potential morphological variants with a semantic distance. Word forms with small semantic distance are proposed as morphological variants of one anther.

Goldsmith (2001), by searching over a space of morphology models limited to substitution of suffixes, ties morphology yet closer to orthography. Segmenting word forms in a corpus, Goldsmith creates an inventory of stems and suffixes. Suffixes which can interchangeably concatenate onto a set of stems form a signature. After defining the space of signatures, Goldsmith searches for that choice of word segmentations resulting in a minimum description length local optimum.

Finally, the work of Harris (1955; 1967), and later Hafer and Weiss (1974), has direct bearing on the approach taken in this paper. Couched in modern terms, their work involves first building tries over a corpus vocabulary and then selecting, as morpheme boundaries, those character boundaries with corresponding high branching count in the tries.

The work in this paper also has a strong bias toward discovering morphologically related words that share a similar orthography. In particular, the

morphology model I use is, akin to Goldsmith, limited to suffix substitution. The novel proposal I bring to the table, however, is a formalization of the full search space of all candidate inflection classes. With this framework in place, defining search strategies for morpheme discovery becomes a natural and straightforward activity.

3 Inflection Classes as Motivation

When learning the morphology of a foreign language, it is common for a student to study tables of inflection classes. Carstairs-McCarthy formalizes the concept of an inflection class in chapter 16 of The Handbook of Morphology (1998). In his terminology, a language with inflectional morphology contains lexemes which occur in a variety of word forms. Each word form carries two pieces of information:

1) Lexical content and
2) Morphosyntactic properties.

For example, the English word form *gave* expresses the lexeme GIVE plus the morphosyntactic property *Past*, while *gives* expresses GIVE plus the properties *3rd Person*, *Singular*, and *Non-Past*.

A set of morphosyntactic properties realized with a single word form is defined to be a *cell*, while a *paradigm* is a set of cells exactly filled by the word forms of some lexeme. A particular natural language may have many paradigms. In English, a language with very little inflectional morphology, there are at least two paradigms, a noun paradigm consisting of two cells, *Singular* and *Plural*, and a paradigm for verbs, consisting of the five cells given (with one choice of naming convention) as the first column of Table 1.

Lexemes that belong to the same paradigm may still differ in their morphophonemic realizations of various cells in that paradigm—each paradigm may have several associated *inflection classes* which specify, for the lexemes belonging to that inflection class, the surface instantiation for each cell of the paradigm. Three of the many inflection classes within the English verb paradigm are found in Table 1 under the columns labeled A through C.

The task the morphology induction system presented in this paper engages is exactly the discovery of the inflection classes of a natural language. Unlike the analysis in Table 1, however, the rest of this paper treats word forms as simply strings of characters as opposed to strings of phonemes.

4 Empirical Inflection Classes

There are two stages in the approach to unsupervised morphology induction proposed in this paper. First, a search space over a set of candidate inflection classes is defined, and second, this space is searched for those candidates most likely to be part of a true inflection class in the language. I have written a program to create the search space but the search strategies described in this paper have yet to be implemented.

Verb Paradigm	Inflection Classes		
	A	**B**	**C**
Basic	blame roam solve	show sow saw	sing ring
3rd Person Singular Non-past	-/z/ blames roams solves	-/z/ shows sows saws	-/z/ sings rings
Past	-/d/ blamed roamed solved	-/d/ showed sowed sawed	V→ /eɪ/ sang rang
Perfective or *Passive*	-/d/ blamed roamed solved	-/n/ shown sown sawn	V→ /ʌ/ sung rung
Progressive	-/iŋ/ blaming roaming solving	-/iŋ/ showing sowing sawing	-/iŋ/ singing ringing

Table 1: A few inflection classes of the English verb paradigm

4.1 Candidate Inflection Class Search Space

To define a search space wherein inflection classes of a natural language can be identified, my algorithm accepts as input a monolingual corpus for the language and proposes candidate morpheme boundaries at every character boundary in every word form in the corpus vocabulary. I call each string before a candidate morpheme boundary a *candidate stem* or *c-stem*, and each string after a boundary a *c-suffix*. I define a *candidate inflection class* (CIC) to be a set of c-suffixes for which there exists at least one c-stem, t, such that each c-suffix in the CIC concatenated to t produces a word form in the vocabulary. I let the set of c-stems which generate a CIC, C, be called the *adherent c-stems* of C; the size of the set of adherent c-stems of C be C's *adherent size*; and the size of the set of c-suffixes in C be the *level* of C.

I then define a lattice of relations between CIC's. In particular, two types of relations are defined:

1) C-suffix set inclusion relations relate pairs of CIC's when the c-suffixes of one CIC are a superset of the c-suffixes of the other, and
2) Morpheme boundary relations occur between CIC's which propose different mor-

pheme boundaries within the same word forms.

Figure 1 diagrams a portion of a CIC lattice over a toy vocabulary consisting of a subset of the word forms found under inflection class A from Table 1. The c-suffix set inclusion relations, represented vertically by solid lines, connect such CIC's as **e.es.ed** and **e.ed**, both of which originate from the c-stem *blam*, since the first is a superset of the second. Morpheme boundary relations, drawn horizontally with dashed lines, connect such CIC's as **me.mes.med** and **e.es.ed**, each derived from exactly the triple of word forms *blame*, *blames*, and *blamed*, but differing in the placement of the hypothesized morpheme boundary

Hierarchical links, connect any given CIC to often more than one parent and more than one child. The empty CIC (not pictured in Figure 1) can be considered the child of all level one CIC's (including the **Ø** CIC), but there is no universal parent of all top level CIC's.

Horizontal morpheme boundary links, dashed lines, connect a CIC, C, with a neighbor to the right if each c-suffix in C begins with the same character. This entails that there is at most one morpheme boundary link leading to the right of each CIC. There may be, however, as many links leading to the left as there are characters in the orthography. The only CIC with depicted multiple left links in Figure 1 is **Ø**, which has left links to the CIC's **e**, **s**, and **d**. A number of left links emanating from the CIC's in Figure 1 are not shown; among others absent from the figure is the left link from the CIC **e.es** leading to the CIC **ve.ves** with the adherent *sol*.

While many ridiculous CIC's are found in Figure 1, such as **ame.ames.amed** from the vocabulary items *blame*, *blames*, and *blamed* and the c-stem *bl*, there are also CIC's that seem very reasonable, such as **Ø.s** from the c-stems *blame* and *tease*. The key task in automatic morphology induction is to autonomously separate the nonsense CIC's from the useful ones, thus identifying linguistically plausible inflection classes.

To better visualize what a CIC lattice looks like when derived from real data, Figure 2 contains a portion of a hierarchical lattice automatically generated from the Spanish newswire corpus. Each entry in Figure 2 contains the c-suffixes comprising the CIC, the adherent size of the CIC, and a sample of adherent c-stems. The lattice in Figure 2 covers:

1) The productive Spanish inflection class for adjectives, **a.as.o.os**, covering the four cells feminine singular, feminine plural, masculine singular, and masculine plural, respectively;

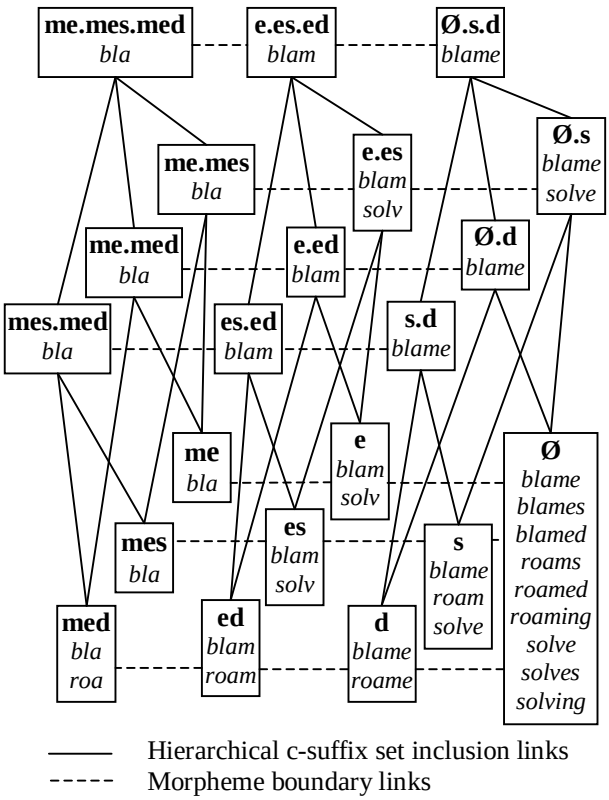

Figure 1: Portion of a CIC lattice from the toy vocabulary: blame, blames, blamed, roams, roamed, roaming, solve, solves, solving

2) All possible CIC subsets of the adjective CIC, e.g. **a.as.o**, **a.os**, etc.; and
3) The imposter CIC **a.as.o.os.tro**, together with its rogue descendents, **a.tro** and **tro**.

Other CIC's that are descendents of **a.as.o.os.tro** and that contain the c-suffix *tro* do not supply additional adherents and hence are not present either in Figure 2 or in my program's representation of the CIC lattice. The CIC's **a.as.tro** and **os.tro**, for example, both have only the one adherent, *cas*, already possessed by their common ancestor **a.as.o.os.tro**.

4.2 Search

With the space of candidate inflection classes defined, it seems natural to treat this lattice of CIC's as a hypothesis space of valid inflection classes and to search this space for CIC's most likely to be true inflection classes in a language. There are many possible search strategies applicable to the CIC lattice. Monson et al. (2004) investigate a series of heuristic search algorithms. Using the same Spanish newswire corpus as this paper, the implemented algorithms have achieved F_1 measures above 0.5 when identifying CIC's belonging to true inflection classes in Spanish. In

this paper I discuss some theoretical motivations underlying CIC lattice search.

Since there are two types of relations in the CIC lattices I construct, search can be broken into two phases. One phase searches the c-suffix set inclusion relations, and the other phase searches the morpheme boundary relations. The search algorithms discussed in Monson et al. (2004) focus on searching the c-suffix set inclusion relations and only utilize morpheme boundary links as a constraint.

In previous related work, morpheme boundary relations and c-suffix set inclusion relations are implicitly present but not explicitly referred to. For example, Goldsmith (2001) does not separate these two types of search. Goldsmith's triage search strategies, which make small changes in the segmentation positions in words, primarily search the morpheme boundary relations, while the vertical search is primarily performed by heuristics that suggest initial word segmentations. To illustrate, if, using the Spanish newswire corpus from this paper, Goldsmith's algorithm decided to segment the word form *castro* as *cas-tro*, then there is an implicit vote for the CIC **a.as.o.os.tro** in Figure 2. If, on the other hand, his algorithm decided not to segment *castro* then there is a vote for the lower level CIC **a.as.o.os**.

The next two subsections motivate search over the morpheme boundary relations and the c-suffix set inclusion relations respectively.

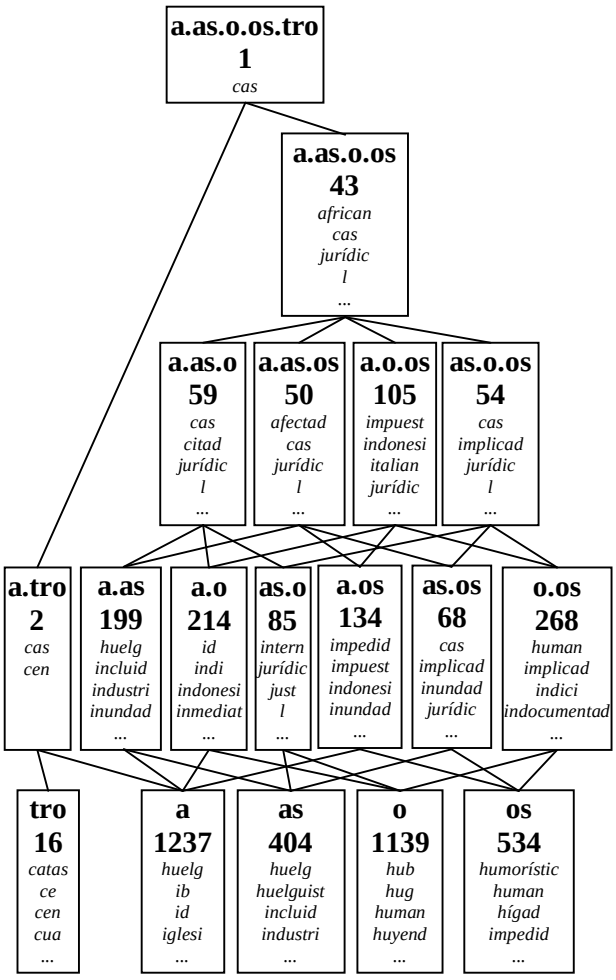

Figure 2: Hierarchical CIC lattice automatically derived from Spanish

4.2.1 Searching Morpheme Boundary Relations

Harris (1955; 1967) and Hafer and Weiss (1974) obtain intriguing results at segmenting word forms into morphemes by first placing the word forms from a vocabulary in a trie, such as the trie pictured in the top half of Figure 3, and then proposing morpheme boundaries after trie nodes that have a large branching factor. The rationale behind their procedure is that the phoneme, or grapheme, sequence within a morpheme is completely restricted, while at a morpheme boundary any number of new morphemes (many with different initial phonemes) could occur. To assess the flavor of Harris' algorithms, the bottom branch of the trie in Figure 3 begins with *roam* and subsequently encounters a branching factor of three, leading to the trie nodes *Ø*, *i*, and *s*. Such a high branching factor suggests there may be a morpheme boundary after *roam*.

One way to view the horizontal morpheme boundary links in a CIC lattice is as a character trie generalization where identical sub-tries within the full vocabulary trie are conflated. Figure 3 illustrates the correspondences between a trie and a portion of a CIC lattice for a small vocabulary consisting of the word forms: *rest*, *rests*, *resting*, *retreat*, *retreats*, *retreating*, *retry*, *retries*, *retrying*, *roam*, *roams*, and *roaming*. Each circled sub-trie of the trie in the top portion of the figure corresponds to one of the four CIC's in the bottom portion of the figure. For example, the right-branching children of the *y* node in *retry* form a sub-trie consisting of *Ø* and *ing*, but this same sub-trie is also found following the *t* node in *rest*, the *t* node in *retreat*, and the *m* node in *roam*. The CIC lattice conflates all these sub-tries into the single CIC **Ø.ing** with the four adherents *rest, retreat, retry,* and *roam*.

Taking this congruency further, branching factor in the trie corresponds roughly to the level of a CIC. A level 3 CIC such as **Ø.ing.s** corresponds to sub-tries with initial branching factor of 3. If separate c-suffixes in a CIC happen to begin with the same character, then a lower branching factor may correspond to a higher level CIC. Similarly, the number of sub-tries which conflate to form a CIC corresponds to the number of adherents belonging to the CIC.

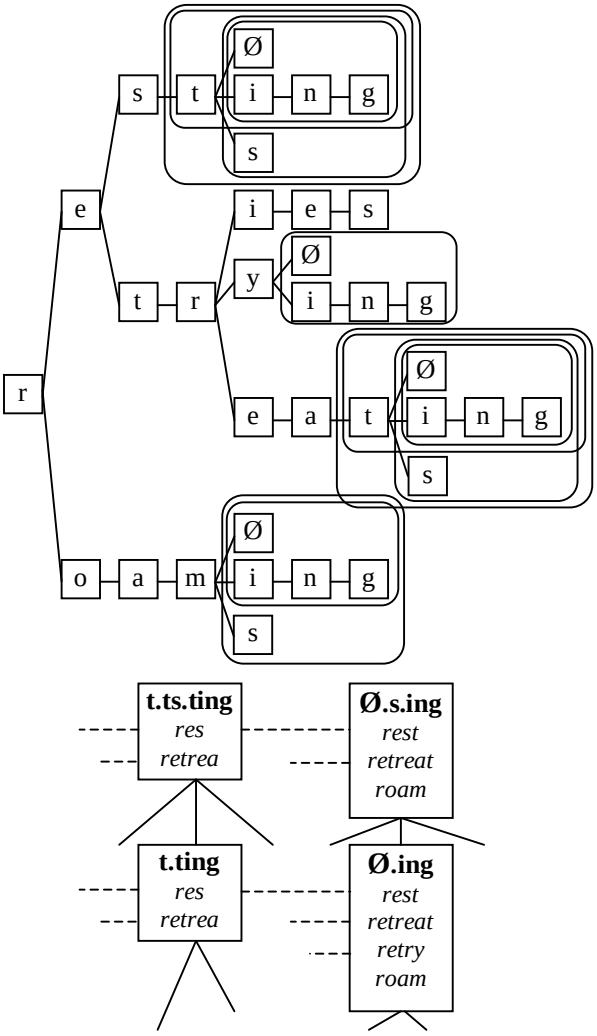

Figure 3: A trie (top) with some repeated sub-tries circled. These sub-tries are then conflated into the corresponding CIC lattice (bottom).

It is interesting to note that while Harris' style phoneme successor criteria do often correctly identify morpheme boundaries, they posses one inherent class of errors. Because Harris treats all word forms with the same initial string as identical, any morpheme boundary decision is global for all words that happen to begin with the same string. For example, Harris cannot differentiate between the forms *casa* and *castro*. If a morpheme boundary is (correctly) placed after the *cas* in *casa*, then a morpheme boundary must be placed (incorrectly) after the *cas* in *castro*. Using a CIC lattice, however, allows an algorithm to first choose which branches of a trie are relevant and then select morpheme boundaries given the relevant sub-trie. Exploring the vertical CIC lattice in Figure 2, a search algorithm might hope to discover that the *tro* trie branch is irrelevant and search for a morpheme boundary along the sub-tries ending in *a.as.o.os*. Perhaps the morpheme boundary search would use the branching factor of this restricted trie as a discriminative criterion.

4.2.2 Searching C-suffix Set Inclusion Relations

Since trie branches correspond to CIC level, I turn now to outline a search method over the vertical c-suffix set inclusion relations. This search method makes particular use of CIC adherent counts through the application of statistical independence tests. The goal of a vertical search algorithm is to avoid c-suffixes which occur not as true suffixes that are part of an inflection class, but instead as random strings that happen to be able to attach to a given initial string.

To formalize the idea of randomness I treat each c-suffix, F, as a Boolean random variable which is true when F attaches to a given c-stem and false when F does not attach to that c-stem. I then make the simplifying assumption that c-stems are independent identically distributed draws from the population of all possible c-stems. Since my algorithm identifies all possible initial substrings of a vocabulary as c-stems, the c-stems are clearly not truly independent—some c-stems are actually substrings of other c-stems.

Nevertheless, natural language inflection classes, in the model of this paper, consist of c-suffixes which interchangeably attach to the same c-stems. Hence, given the assumption of c-suffixes as random variables, the true inflection classes of a language are most likely those groups of c-suffixes which are positively correlated. That is, if knowing that c-suffix F_1 concatenates onto c-stem T increases the probability that the suffix F_2 also concatenates onto T, then F_1 and F_2 are likely from the same inflection class. On the other hand, if F_1 and F_2 are statistically independent, or knowing that F_1 concatenates to T does not change the probability that F_2 can attach to T, then it is likely that F_1 or F_2 (or both) is a c-suffix that just randomly happens to be able to concatenate onto a T. And finally, if F_1 and F_2 are negatively correlated, i.e. they occur interchangeably on the same c-stem less frequently than random chance, then it may be that F_1 and F_2 come from different inflection classes within the same paradigm or are even associated with completely separate paradigms.

There are a number of statistical tests designed to assess the probability that two discrete random variables are independent. Here I will look at the χ^2 independence test, which computes the probability that two random variables are independent by calculating a statistic Q distributed as χ^2 by comparing the expected distributions of the two random variables, assuming their independence with their actual distribution. The larger the values of Q, the lower the probability that the random variables are independent.

Summing the results of each c-stem independent trial of the c-suffix Boolean random variables, re-

sults in Bernoulli distributed random variables whose joint distributions can be described as two by two contingency tables. Table 2 gives such contingency tables for the pairs of random variable c-suffixes (*a, as*) and (*a, tro*). These tables can be calculated by examining specific CIC's in the lattices. To fill the contingency table for (*a, as*) I proceed as follows: The number of times *a* occurs jointly with *as* is exactly the adherent size of the **a.as** CIC, 199. The marginal number of occurrences of *a*, 1237, can be read from the CIC **a,** and similarly the marginal number of occurrences of *as*, 404, can be read from the CIC **as**. The bottom right-hand cell in the tables in Table 2 is the total number of trials, or in this case, the number of unique c-stems. This quantity is easily calculated by summing the adherent sizes of all level one CIC's together. In the Spanish newswire corpus there are 22950 unique c-stems. The remaining cells in the contingency table can be calculated by assuring the rows and columns sum up to their marginals. Using these numbers we can calculate the Q statistic: Q(**a**, **as**) = 1552 and Q(**a**, **tro**) = 1.587. These values suggest that **a** and **as** are not independent while **a** and **tro** are.

	a	~a	marginal
as	199	205	404
~as	1038	21508	22546
marginal	1237	21713	22950

	a	~a	marginal
tro	2	14	16
~tro	1235	21699	22934
marginal	1237	21713	22950

Table 2: Contingency tables for a few c-suffixes

5 Future Work

There is clearly considerable work left to do within the CIC framework presented in this paper. I intend to implement the search strategies outlined in this paper. I also plan to apply these techniques to describe the morphologies of a variety of languages beyond English and Spanish.

Acknowledgements

The research presented in this paper was funded in part by NSF grant number IIS-0121631.

References

Andrew Carstairs-McCarthy. 1998. "Paradigmatic Structure: Inflectional Paradigms and Morphological Classes." *The Handbook of Morphology*. Eds. Andrew Spencer and Arnold M. Zwicky. Blackwell Publishers Inc., Massachusetts, USA, 322-334.

Éric Gaussier. 1999. Unsupervised learning of derivational morphology from inflectional lexicons. In *Proceedings of ACL '99 Workshop: Unsupervised Learning in Natural Language Processing*.

John Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27(2): 153-198.

Margaret A. Hafer and Stephen F. Weiss. 1974. Word segmentation by letter successor varieties. *Information Storage and Retrieval*, 10:371-385.

Zellig Harris. 1955. From phoneme to morpheme. *Language*, 31:190-222. Reprinted in Harris 1970.

Zellig Harris. 1967. Morpheme boundaries within words: Report on a computer test. *Transformation and Discourse Analysis Papers* 73, Department of Linguistics, University of Pennsylvania. Reprinted in Harris 1970.

Zellig Harris. 1970. *Papers in Structural and Transformational Linguistics*. D. Reidel, Dordrecht, Holland.

Christian Monson, Alon Lavie, Jaime Carbonell, and Lori Levin. 2004. Unsupervised Induction of Natural Language Morphology Inflection Classes. In *Proceedings of the Seventh Meeting of the ACL Special Interest Group in Computational Phonology (SIGPHON'04)*.

Patrick Schone and Daniel Jurafsky. 2000. Knowledge-free Induction of Morphology Using Latent Semantic Analysis. In *Proceedings of the Fourth Conference on Computational Natural Language Learning and of the Second Learning Language in Logic Workshop*, 67-72.

Patrick Schone and Daniel Jurafsky. 2001. Knowledge-free Induction of Inflectional Morphologies. In *Proceedings of the North American Chapter of the Association of Computational Linguistics*. 183-191.

David Yarowsky, Grace Ngai, and Richard Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the Human Language Technology Conference*, 161-168.