Semi-supervised learning of morphological paradigms and lexicons

Malin Ahlberg
Språkbanken
University of Gothenburg
malin.ahlberg@gu.se

Markus Forsberg
Språkbanken
University of Gothenburg
markus.forsberg@gu.se

Mans Hulden
University of Helsinki
mans.hulden@helsinki.fi

Abstract

We present a semi-supervised approach to the problem of paradigm induction from inflection tables. Our system extracts generalizations from inflection tables, representing the resulting paradigms in an abstract form. The process is intended to be language-independent, and to provide human-readable generalizations of paradigms. The tools we provide can be used by linguists for the rapid creation of lexical resources. We evaluate the system through an inflection table reconstruction task using Wiktionary data for German, Spanish, and Finnish. With no additional corpus information available, the evaluation yields per word form accuracy scores on inflecting unseen base forms in different languages ranging from 87.81% (German nouns) to 99.52% (Spanish verbs); with additional unlabeled text corpora available for training the scores range from 91.81% (German nouns) to 99.58% (Spanish verbs). We separately evaluate the system in a simulated task of Swedish lexicon creation, and show that on the basis of a small number of inflection tables, the system can accurately collect from a list of noun forms a lexicon with inflection information ranging from 100.0% correct (collect 100 words), to 96.4% correct (collect 1000 words).

1 Introduction

Large scale morphologically accurate lexicon construction for natural language is a very time-consuming task, if done manually. Usually, the construction of large-scale lexical resources presupposes a linguist who constructs a detailed morphological grammar that models inflection, compounding, and other morphological and phonological phenomena, and additionally performs a manual classification of lemmas in the language according to their paradigmatic behavior.

In this paper we address the problem of lexicon construction by constructing a semi-supervised system that accepts concrete inflection tables as input, generalizes inflection paradigms from the tables provided, and subsequently allows the use of unannotated corpora to expand the inflection tables and the automatically generated paradigms.[1]

In contrast to many machine learning approaches that address the problem of paradigm extraction, the current method is intended to produce human-readable output of its generalizations. That is, the paradigms provided by the system can be inspected for errors by a linguist, and if necessary, corrected and improved. Decisions made by the extraction algorithms are intended to be transparent, permitting morphological system development in tandem with linguist-provided knowledge.

Some of the practical tasks tackled by the system include the following:

- Given a small number of known inflection tables, extract from a corpus a lexicon of those lemmas that behave like the examples provided by the linguist.
- Given a large number of inflection tables—such as those provided by the crowdsourced lexical resource, Wiktionary—generalize the tables into a smaller number of abstract paradigms.

2 Previous work

Automatic learning of morphology has long been a prominent research goal in computational linguistics. Recent studies have focused on unsupervised methods in particular—learning morphology from

[1] Our programs and the datasets used, including the evaluation procedure for this paper, are freely available at https://svn.spraakbanken.gu.se/clt/eacl/2014/extract

Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 569–578,
Gothenburg, Sweden, April 26-30 2014. ©2014 Association for Computational Linguistics

unlabeled data (Goldsmith, 2001; Schone and Jurafsky, 2001; Chan, 2006; Creutz and Lagus, 2007; Monson et al., 2008). Hammarström and Borin (2011) provides a current overview of unsupervised learning.

Previous work with similar semi-supervised goals as the ones in this paper include Yarowsky and Wicentowski (2000), Neuvel and Fulop (2002), Clément et al. (2004). Recent machine learning oriented work includes Dreyer and Eisner (2011) and Durrett and DeNero (2013), which documents a method to learn orthographic transformation rules to capture patterns across inflection tables. Part of our evaluation uses the same dataset as Durrett and DeNero (2013). Eskander et al. (2013) shares many of the goals in this paper, but is more supervised in that it focuses on learning inflectional classes from richer annotation.

A major departure from much previous work is that we do not attempt to encode variation as string-changing operations, say by string edits (Dreyer and Eisner, 2011) or transformation rules (Lindén, 2008; Durrett and DeNero, 2013) that perform mappings between forms. Rather, our goal is to encode all variation within paradigms by presenting them in a sufficiently generic fashion so as to allow affixation processes, phonological alternations as well as orthographic changes to naturally fall out of the paradigm specification itself. Also, we perform no explicit alignment of the various forms in an inflection table, as in e.g. Tchoukalov et al. (2010). Rather, we base our algorithm on extracting the longest common subsequence (LCS) shared by all forms in an inflection table, from which alignment of segments falls out naturally. Although our paradigm representation is similar to and inspired by that of Forsberg et al. (2006) and Détrez and Ranta (2012), our method of generalizing from inflection tables to paradigms is novel.

3 Paradigm learning

In what follows, we adopt the view that words and their inflection patterns can be organized into paradigms (Hockett, 1954; Robins, 1959; Matthews, 1972; Stump, 2001). We essentially treat a paradigm as an ordered set of functions $(f_1, \ldots, f_n)$, where $f_i\colon x_1, \ldots, x_n \mapsto \Sigma^*$, that is, where each entry in a paradigm is a function from variables to strings, and each function in a particular paradigm shares the same variables.

3.1 Paradigm representation

We represent the functions in what we call *abstract paradigm*. In our representation, an *abstract paradigm* is an ordered collection of strings, where each string may additionally contain interspersed variables denoted $x_1, x_2, \ldots, x_n$. The strings represent fixed, obligatory parts of a paradigm, while the variables represent mutable parts. These variables, when instantiated, must contain at least one segment, but may otherwise vary from word to word. A complete *abstract paradigm* captures some generalization where the mutable parts represented by variables are instantiated the same way for all forms in one particular inflection table. For example, the fairly simple paradigm

x_1 x_1+**s** x_1+**ed** x_1+**ing**

could represent a set of English verb forms, where x_1 in this case would coincide with the infinitive form of the verb—**walk**, **climb**, **look**, etc.

For more complex patterns, several variable parts may be invoked, some of them discontinuous. For example, part of an inflection paradigm for German verbs of the type **schreiben** (to write) verbs may be described as:

x_1+**e**+x_2+x_3+**en**	Infinitive
x_1+**e**+x_2+x_3+**end**	Present participle
ge+x_1+x_2+**e**+x_3+**en**	Past participle
x_1+**e**+x_2+x_3+**e**	Present 1P Sg
x_1+**e**+x_2+x_3+**st**	Present 2P Sg
x_1+**e**+x_2+x_3+**t**	Present 3P Sg

If the variables are instantiated as x_1=**schr**, x_2=**i**, and x_3=**b**, the paradigm corresponds to the forms (**schreiben, schreibend, geschrieben, schreibe, schreibst, schreibt**). If, on the other hand, x_1=**l**, x_2=**i**, and x_3=**h**, the same paradigm reflects the conjugation of **leihen** (to lend/borrow)—(**leihen, leihend, geliehen, leihe, leihst, leiht**).

It is worth noting that in this representation, no particular form is privileged in the sense that all other forms can only be generated from some special form, say the infinitive. Rather, in the current representation, all forms can be derived from knowing the variable instantiations. Also, given only a particular word form and a hypothetical paradigm to fit it in, the variable instantiations can often be logically deduced unambiguously. For example, let us say we have a hypothetical form **steigend** and need to fit it in the above paradigm, without knowing which slot it should occupy. We may deduce that it must represent the present participle, and that x_1=**st**, x_2=**i**, and x_3=**g**. From this knowledge, all other forms can subsequently be derived.

Although we have provided grammatical information in the above table for illustrative purposes, our primary concern in the current work is the generalization from inflection tables—which for our purposes are simply an ordered set of word forms—to paradigms of the format discussed above.

3.2 Paradigm induction from inflection tables

The core component of our method consists of finding, given an inflection table, the *maximally general paradigm* that reflects the information in that table. To this end, we make the assumption that string subsequences that are shared by different forms in an inflection table are incidental and can be generalized over. For example, given the English verb **swim**, and a simple inflection table **swim#swam#swum**,[2] we make the assumption that the common sequences **sw** and **m** are irrelevant to the inflection, and that by disregarding these strings, we can focus on the segments that vary within the table—in this case the variation **i**$\sim$**a**$\sim$**u**. In other words, we can assume **sw** and **m** to be *variables* that vary from word to word and describe the table **swim#swam#swum** as x_1**+i+**x_2**#**x_1**+a+**x_2**#**x_1**+u+**x_2, where x_1=**sw** and x_2=**m** in the specific table.

3.2.1 Maximally general paradigms

In order to generalize as much as possible from an inflection table, we extract from it what we call the *maximally general paradigm* by:

1. Finding the longest common subsequence (LCS) to all the entries in the inflection table.

2. Finding the segmentation into variables of the LCS(s) (there may be several) in the inflection table that results in

   (a) The smallest number of variables. Two segments xy in the LCS must be part of the same variable if they always occur together in every form in the inflection table, otherwise they must be assigned separate variables.

   (b) The smallest total number of infixed non-variable segments in the inflection table (segments that occur between variables).

3. Replacing the discontinuous sequences that are part of the LCS with variables (every form in a paradigm will contain the same number of variables).

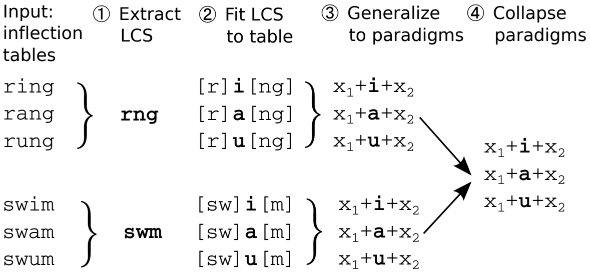


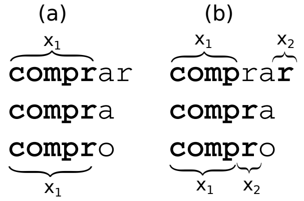
Figure 1: Illustration of our paradigm generalization algorithm. In step ① we extract the LCS separately for each inflection table, attempt to find a consistent fit between the LCS and the forms present in the table (step ②), and assign the segments that participate in the LCS variables (step ③). Finally, resulting paradigms that turn out to be identical may be collapsed (step ④) (section 3.3).

These steps are illustrated in figure 1. The first step, extracting the LCS from a collection of strings, is the well-known multiple longest common subsequence problem (MLCS). It is known to be NP-hard (Maier, 1978). Although the number of strings to find the LCS from may be rather large in real-world data, we find that a few sensible heuristic techniques allow us to solve this problem efficiently for practical linguistic material, i.e., inflection tables. We calculate the LCS by calculating intersections of finite-state machines that encode all subsequences of all words, using the *foma* finite-state toolkit (Hulden, 2009).[3]

While for most tables there is only one way to segment the LCS in the various forms, some ambiguous corner cases need to be resolved by imposing additional criteria for the segmentation, given in steps 2(a) and 2(b). As an example, consider a snippet of a small conjugation table for the Spanish verb **comprar** (to buy), **comprar#compra#compro**. Obviously the LCS is **compr**—however, this can be distributed in two different ways across the strings, as seen below.

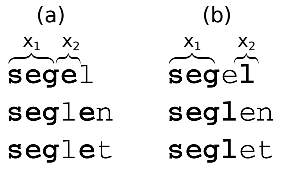
[2] To save space, we will henceforth use the #-symbol as a delimiter between entries in an inflection table or paradigm.

[3] Steps 2 and 3 are implemented using more involved finite-state techniques that we plan to describe elsewhere.



The obvious difference here is that in the first assignment, we only need to declare one variable x_1=**compr**, while in the second, we need two, x_1=**comp**, x_2=**r**. Such cases are resolved by choosing the segmentation with the smallest number of variables by step 2(a).

Remaining ambiguities are resolved by minimizing the total number of infixed segments. As an illustration of where this is necessary, consider a small extract from the Swedish noun table **segel** (sail): **segel#seglen#seglet**. Here, the LCS, of which there are two of equal length (**sege/segl**) must be assigned to two variables where either x_1=**seg** and x_2=**e**, or x_1=**seg** and x_2=**l**:



However, in case (a), the number of infixed segments—the **l**'s in the second and third form—total one more than in the distribution in (b), where only one **e** needs to be infixed in one form. Hence, the representation in (b) is chosen in step 2(b).

The need for this type of disambiguation strategy surfaces very rarely and the choice to minimize infix length is largely arbitrary—although it may be argued that some linguistic plausibility is encoded in the minimization of infixes. However, choosing a consistent strategy is important for the subsequent collapsing of paradigms.

3.3 Collapsing paradigms

If several tables are given as input, and we extract the *maximally general paradigm* from each, we may collapse resulting paradigms that are identical. This is also illustrated in figure 1.

As paradigms are collapsed, we record the information about how the various variables were interpreted prior to collapsing. That is, for the example in figure 1, we not only store the resulting single paradigm, but also the information that x_1=**r**, x_2=**ng** in one table and that x_1=**sw**, x_2=**m** in another. This allows us to potentially reconstruct all the inflection tables seen during learning. Storing this information is also crucial for paradigm table collection from text, fitting unseen word forms into paradigms, and reasoning about unseen paradigms, as will be discussed below.

Form	Input	Generalization
[Inf]	kaufen	x_1+**en**
[PresPart]	kaufend	x_1+**end**
[PastPart]	gekauft	**ge**+x_1+**t**
[Pres1pSg]	kaufe	x_1+**e**
[Pres1pPl]	kaufen	x_1+**en**
[Pres2pSg]	kaufst	x_1+**st**
[Pres2pPl]	kauft	x_1+**t**
[Pres3pSg]	kauft	x_1+**t**
[Pres3pPl]	kaufen	x_1+**en**
...	...	...
x_1 = **kauf**		

Table 1: Generalization from a German example verb **kaufen** (to buy) exemplifying typical rendering of paradigms.

3.4 MLCS as a language-independent generalization strategy

There is very little language-specific information encoded in the strategy of paradigm generalization that focuses on the LCS in an inflection table. That is, we do not explicitly prioritize processes like prefixation, suffixation, or left-to-right writing systems. The resulting algorithm thus generalizes tables that reflect concatenative and non-concatenative morphological processes equally well. Tables 1 and 2 show the outputs of the method for German and Arabic verb conjugation reflecting the generalization of concatenative and non-concatenative patterns.

3.5 Instantiating paradigms

As mentioned above, given that the variable instantiations of a paradigm are known, we may generate the full inflection table. The variable instantiations are retrieved by matching a word form to one of the patterns in the paradigms. For example, the German word form **bücken** (to bend down) may be matched to three patterns in the paradigm exemplified in table 1, and all three matches yield the same variable instantiation, i.e., x_1=**bück**.

Paradigms with more than one variable may be sensitive to the matching strategy of the variables. To see this, consider the pattern x_1**+a+**x_2 and the word **banana**. Here, two matches are possible x_1=**b** and x_2=**nana** and x_1=**ban** and x_2 =**na**. In other words, there are three possible matching

Form	Input	Generalization
[Past1SG]	katabtu (كَتَبْتُ)	x_1+**a**+x_2+**a**+x_3+**tu**
[Past2SGM]	katabta (كَتَبْتَ)	x_1+**a**+x_2+**a**+x_3+**ta**
[Past2SGF]	katabti (كَتَبْتِ)	x_1+**a**+x_2+**a**+x_3+**ti**
[Past3SGM]	kataba (كَتَبَ)	x_1+**a**+x_2+**a**+x_3+**a**
[Past3SGF]	katabat (كَتَبَتْ)	x_1+**a**+x_2+**a**+x_3+**at**
…	…	…
[Pres1SG]	aktubu (اَكْتُبُ)	**a**+x_1+x_2+**u**+x_3+**u**
[Pres2SGM]	taktubu (تَكْتُبُ)	**ta**+x_1+x_2+**u**+x_3+**u**
[Pres2SGF]	taktubīna (تَكْتُبِينَ)	**ta**+x_1+x_2+**u**+x_3+**īna**
[Pres3SGM]	yaktubu (يَكْتُبُ)	**ya**+x_1+x_2+**u**+x_3+**u**
[Pres3SGF]	taktubu (تَكْتُبُ)	**ta**+x_1+x_2+**u**+x_3+**u**
…	…	…
x_1 = **k** (ك), x_2 = **t** (ت), x_3 = **b** (ب)		

Table 2: Generalization from an Arabic conjugation table involving the root **/k-t-b/** from which the stems **katab** (to write/past) and **ktub** (present/non-past) are formed, conjugated in Form I, past and present tenses. Extracting the longest common subsequence yields a paradigm where variables correspond to root radicals.

strategies:[4]

1. shortest match (x_1 =**b** and x_2 =**nana**)
2. longest match (x_1 =**ban** and x_2 =**na**)
3. try all matching combinations

The matching strategy that tends to be successful is somewhat language-dependent: for a language with a preference for suffixation, longest match is typically preferred, while for others shortest match or trying all combinations may be the best choice. All languages evaluated in this article have a preference for suffixation, so in our experiments we have opted for using the longest match for the sake of convenience. Our implementation allows for exploring all matches, however. Even though all matches were to be tried, 'bad' matches will likely result in implausible inflections that can be discarded using other cues.

4 Assigning paradigms automatically

The next problem we consider is assigning the correct paradigms to candidate words automatically. As a first step, we match the current word to a pattern. In the general case, all patterns are tried for a given candidate word. However, we usually have access to additional information about the candidate words—e.g., that they are in the base form of a certain part of speech—which we use to improve the results by only matching the relevant patterns.

From a candidate word, all possible inflection tables are generated. Following this, a decision procedure is applied that calculates a confidence score to determine which paradigm is the most probable. The score is a weighted combination of the following calculations:

1. Compute the longest common suffix for the generated base form (which may be the input form) with previously seen base forms. If of equal length, select the paradigm where the suffix occurs with higher frequency.

2. Compute frequency spread over the set of unique word forms according to the following formula: $\sum_{w \in set(W)} log(freq(w) + 1)$

3. Use the most frequent paradigm as a tie-breaker.

Step 1 is a simple memory-based approach, much in the same spirit as van den Bosch and Daelemans (1999), where we compare the current base form with what we have seen before.

For step 2, let us elaborate further why the frequency spread is computed on unique word forms. We do this to avoid favoring paradigms that have the same word forms for many or all inflected forms. For example, the German noun **Ananas** (pineapple) has a syncretic inflection with one repeated word form across all slots, **Ananas**. When trying to assign a paradigm to an unknown word form that matches x_1, it will surely fit the paradigm that **Ananas** has generated perfectly since we have encountered every word form in that paradigm, of which there is only one, namely x_1. Hence, we want to penalize low variation of word forms when assigning paradigms.

The confidence score calculated is not only applicable for selecting the most probable paradigm for a given word-form; it may also be used to rank a list of words so that the highest ranked paradigm is the most likely to be correct. Examples of such rankings are found in section 5.3.

[4] The number of matches may increase quickly for longer words and many variables in the worst case: e.g. **caravan** matches x_1+**a**+x_2 in three different ways.

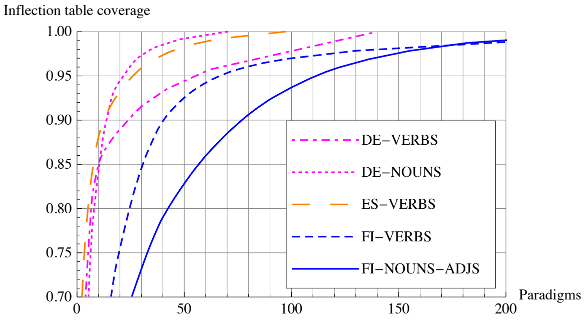


Figure 2: Degree of coverage with varying numbers of paradigms.

Data	Input: inflection tables	Output: abstract paradigms
DE-VERBS	1827	140
DE-NOUNS	2564	70
ES-VERBS	3855	97
FI-VERBS	7049	282
FI-NOUNS-ADJS	6200	258

Table 3: Generalization of paradigms. The number of paradigms produced from Wiktionary inflection tables by generalization and collapsing of abstract paradigms.

5 Evaluation

To evaluate the method, we have conducted three experiments. First we repeat an experiment presented in Durrett and DeNero (2013) using the same data and experiment setup, but with our generalization method. In this experiment, we are given a number of complete inflection tables scraped from Wiktionary. The task is to reconstruct complete inflection tables from 200 held-out base forms. For this task, we evaluate per form accuracy as well as per table accuracy for reconstruction. The second experiment is the same as the first, but with additional access to an unlabeled text dump for the language from Wikipedia.

In the last experiment we try to mimic the situation of a linguist starting out to describe a new language. The experiment uses a large-scale Swedish morphology as reference and evaluates how reliably a lexicon can be gathered from a word list using only a few manually specified inflection tables generalized into abstract paradigms by our system.

5.1 Experiment 1: Wiktionary

In our first experiment we start from the inflection tables in the development and test set from Durrett and DeNero (2013), henceforth D&DN13. Table 3 shows the number of input tables as well as the number of paradigms that they result in after generalization and collapsing. For all cases, the number of output paradigms are below 10% of the number of input inflection tables. Figure 2 shows the generalization rate achieved with the paradigms. For instance, the 20 most common resulting German noun paradigms are sufficient to model almost 95% of the 2,564 separate inflection tables given as input.

As described earlier, in the reconstruction task, the input base forms are compared to the abstract paradigms by measuring the longest common suffix length for each input base form compared to the ones seen during training. This approach is memory-based: it simply measures the similarity of a given lemma to the lemmas encountered during the learning phase. Table 4 presents our results juxtaposed with the ones reported by D&DN13. While scoring slightly below D&DN13 for the majority of the languages when measuring form accuracy, our method shows an advantage when measuring the accuracy of complete tables. Interestingly, the only case where we improve upon the form accuracy of D&DN13 is German verbs, where we get our lowest table accuracy.

Table 4 further shows an oracle score, giving an upper bound for our method that would be achieved if we were always able to pick the best fitting paradigm available. This upper bound ranges from 99% (Finnish verbs) to 100% (three out of five tests).

5.2 Experiment 2: Wiktionary and Wikipedia

In our second experiment, we extend the previous experiment by adding access to a corpus. Apart from measuring the longest common suffix length, we now also compute the frequency of the hypothetical candidate forms in every generated table and use this to favor paradigms that generate a large number of attested forms. For this, we use a Wikipedia dump, from which we have extracted word-form frequencies.[5] In total, the number of word types in the Wikipedia corpus was 8.9M (German), 3.4M (Spanish), 0.7M (Finnish), and 2.7M (Swedish). Table 5 presents the results,

[5] The corpora were downloaded and extracted as described at http://medialab.di.unipi.it/wiki/Wikipedia_Extractor

Data	Per table	D&DN13	Per form	D&DN13	Oracle accuracy per form (per table)
DE-VERBS	68.0	**85.0**	**97.04**	96.19	99.70 (198/200)
DE-NOUNS	76.5	**79.5**	87.81	**88.94**	100.00 (200/200)
ES-VERBS	**96.0**	95.0	99.52	**99.67**	100.00 (200/200)
FI-VERBS	**92.5**	87.5	96.36	**96.43**	99.00 (195/200)
FI-NOUNS-ADJS	**85.0**	83.5	91.91	**93.41**	100.00 (200/200)

Table 4: Experiment 1: Accuracy of reconstructing 200 inflection tables given only base forms from held-out data when paradigms are learned from the Wiktionary dataset. For comparison, figures from Durrett and DeNero (2013) are included (shown as D&DN13).

Data	Per table	Per form	Oracle acc. per form (table)
DE-VERBS	76.50	**97.87**	99.70 (198/200)
DE-NOUNS	**82.00**	**91.81**	100.00 (200/200)
ES-VERBS	**98.00**	99.58	100.00 (200/200)
FI-VERBS	**92.50**	**96.63**	99.00 (195/200)
FI-NOUNS-ADJS	**88.00**	**93.82**	100.00 (200/200)

Table 5: Experiment 2: Reconstructing 200 held-out inflection tables with paradigms induced from Wiktionary and further access to raw text from Wikipedia.

where an increased accuracy is noted for all languages, as is to be expected since we have added more knowledge to the system. The bold numbers mark the cases where we outperform the result in Durrett and DeNero (2013), which is now the case in four out of five tests for table accuracy, scoring between 76.50% for German verbs and 98.00% for Spanish verbs.

Measuring form accuracy, we achieve scores between 91.81% and 99.58%. The smallest improvement is noted for Finnish verbs, which has the largest number of paradigms, but also the smallest corpus.

5.3 Experiment 3: Ranking candidates

In this experiment we consider a task where we only have a small number of inflection tables, mimicking the situation where a linguist has manually entered a few inflection tables, allowed the system to generalize these into paradigms, and now faces the task of culling from a corpus—in this case labeled with basic POS information—the candidate words/lemmas that best fit the induced paradigms. This would be a typical task during lexicon creation.

Top-1000 rank	Correct/Incorrect
Top 10%	100/0 (100.0%)
Top 50%	489/11 (97.8%)
Top 100%	964/36 (96.4%)

Table 6: Top-1000 rank for all nouns in SALDO

We selected the 20 most frequent noun paradigms (from a total of 346), with one inflection table each, from our gold standard, the Swedish lexical resource SALDO (Borin et al., 2013). From this set, we discarded paradigms that lack plural forms.[6] We also removed from the paradigms special compounding forms that Swedish nouns have, since compound information is not taken into account in this experiment. The compounding forms are part of the original paradigm specification, and after a collapsing procedure after compound-form removal, we were left with a total of 11 paradigms.

In the next step we ranked all nouns in SALDO (79.6k lemmas) according to our confidence score, which indicates how well a noun fits a given paradigm. We then evaluated the paradigm assignment for the top-1000 lemmas. Among these top-1000 words, we found 44 that were outside the 20 most frequent noun paradigms. These words were not necessarily incorrectly assigned, since they may only differ in their compound forms; as a heuristic, we considered them correct if they had the same declension and gender as the paradigm, and incorrect otherwise.

Table 6 displays the results, including a total accuracy of 96.4%.

Next, we investigated the top-1000 distribution for individual paradigms. This corresponds to the situation where a linguist has just entered a new inflection table and is looking for words that fit the resulting paradigm. The result is presented in two

[6] The paradigms that lack plural forms are subsets of other paradigms. In other words: when no plural forms are attested, we would need a procedure to decide if plural forms are even possible, which is currently beyond the scope of our method.

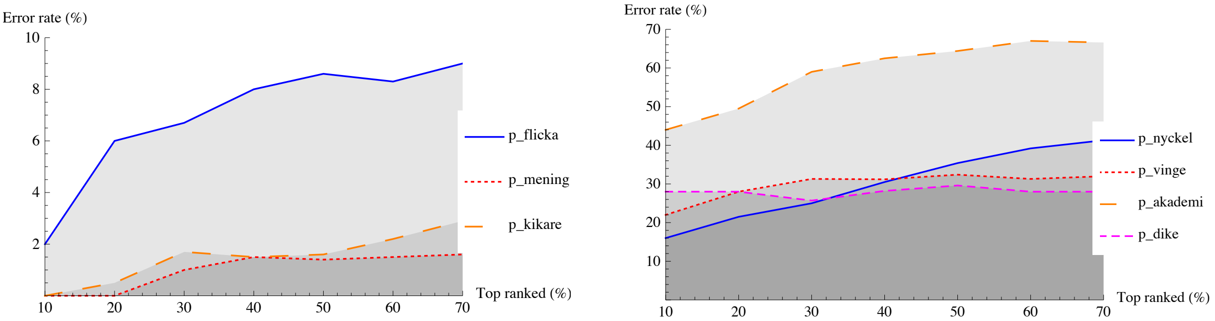


Figure 3: Top-1000: high and low precision paradigms.

error rate plots: figure 3 shows the low precision and high precision paradigms in two plots, where error rates range from 0-2% and 16-44% for the top 100 words.

We further investigated the worst-performing paradigm, **p_akademi** (academy), to determine the reason for the high error rate for this particular item. The main source of error (334 out of 1000) is confusion with **p_akribi** (accuracy), which has no plural. However, it is on semantic grounds that the paradigm has no plural; a native Swedish speaker would pluralize **akribi** like **akademi** (disregarding the fact that **akribi** is defective). The second main type of error (210 out of 1000) is confusion with the unseen paradigm of **parti** (party), which inflects similarly to **akademi**, but with a difference in gender—difficult to predict from surface forms—that manifests itself in two out of eight word forms.

6 Future work

The core method of abstract paradigm representation presented in this paper can readily be extended in various directions. One obvious topic of interest is to investigate the use of machine learning techniques to expand the method to completely unsupervised learning by first clustering similar words in raw text into hypothetical inflection tables. The plausibility of these tables could then be evaluated using similar techniques as in our experiment 2.

We also plan to explore ways to improve the techniques for paradigm selection and ranking. In our experiments we have, for the sake of transparency, used a fairly simple strategy of suffix matching to reconstruct tables from base forms. A more involved classifier may be trained for this purpose. An obvious extension is to use a classifier based on n-gram, capitalization, and other standard features to ascertain that word forms in hypothetical reconstructed inflection tables maintain similar shapes to ones seen during training.

One can also investigate ways to collapse paradigms further by generalizing over phonological alternations and by learning alternation rules from the induced paradigms (Koskenniemi, 1991; Theron and Cloete, 1997; Koskenniemi, 2013).

Finally, we are working on a separate interactive graphical morphological tool in which we plan to integrate the methods presented in this paper.

7 Conclusion

We have presented a language-independent method for extracting paradigms from inflection tables and for representing and generalizing the resulting paradigms.[7] Central to the process of paradigm extraction is the notion of *maximally general paradigm*, which we define as the inflection table, with all of the common string subsequences forms represented by variables.

The method is quite uncomplicated and outputs human-readable generalizations. Despite the relative simplicity, we obtain state-of-the art results in inflection table reconstruction tasks from base forms.

Because of the plain paradigm representation format, we believe the model can be used profitably in creating large-scale lexicons from a few linguist-provided inflection tables.

[7] The research presented here was supported by the Swedish Research Council (the projects *Towards a knowledge-based culturomics*, dnr 2012-5738, and *Swedish Framenet++*, dnr 2010-6013), the University of Gothenburg through its support of the Centre for Language Technology and its support of Språkbanken, and the Academy of Finland under the grant agreement 258373, *Machine learning of rules in natural language morphology and phonology*.

References

Lars Borin, Markus Forsberg, and Lennart Lönngren. 2013. SALDO: a touch of yin to WordNet's yang. *Language Resources and Evaluation*, May. Online first publication; DOI 10.1007/s10579-013-9233-4.

Erwin Chan. 2006. Learning probabilistic paradigms for morphology in a latent class model. In *Proceedings of the Eighth Meeting of the ACL Special Interest Group on Computational Phonology and Morphology*, pages 69–78. Association for Computational Linguistics.

Lionel Clément, Bernard Lang, Benoît Sagot, et al. 2004. Morphology based automatic acquisition of large-coverage lexica. In *LREC 04*, pages 1841–1844.

Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing (TSLP)*, 4(1):3.

Grégoire Détrez and Aarne Ranta. 2012. Smart paradigms and the predictability and complexity of inflectional morphology. In *Proceedings of the 13th EACL*, pages 645–653. Association for Computational Linguistics.

Markus Dreyer and Jason Eisner. 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*, pages 616–627. Association for Computational Linguistics.

Greg Durrett and John DeNero. 2013. Supervised learning of complete morphological paradigms. In *Proceedings of NAACL-HLT*, pages 1185–1195.

Ramy Eskander, Nizar Habash, and Owen Rambow. 2013. Automatic extraction of morphological lexicons from morphologically annotated corpora. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1032–1043. Association for Computational Linguistics.

Markus Forsberg, Harald Hammarström, and Aarne Ranta. 2006. Morphological lexicon extraction from raw text data. In *Advances in Natural Language Processing*, pages 488–499. Springer.

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

Harald Hammarström and Lars Borin. 2011. Unsupervised learning of morphology. *Computational Linguistics*, 37(2):309–350.

Charles F Hockett. 1954. Two models of grammatical description. *Morphology: Critical Concepts in Linguistics*, 1:110–138.

Mans Hulden. 2009. Foma: a finite-state compiler and library. In *Proceedings of the 12th Conference of the European Chapter of the European Chapter of the Association for Computational Linguistics: Demonstrations Session*, pages 29–32, Athens, Greece. Association for Computational Linguistics.

Kimmo Koskenniemi. 1991. A discovery procedure for two-level phonology. *Computational Lexicology and Lexicography: A Special Issue Dedicated to Bernard Quemada*, 1:451–46.

Kimmo Koskenniemi. 2013. An informal discovery procedure for two-level rules. *Journal of Language Modelling*, 1(1):155–188.

Krister Lindén. 2008. A probabilistic model for guessing base forms of new words by analogy. In *Computational Linguistics and Intelligent Text Processing*, pages 106–116. Springer.

David Maier. 1978. The complexity of some problems on subsequences and supersequences. *Journal of the ACM (JACM)*, 25(2):322–336.

Peter H. Matthews. 1972. *Inflectional morphology: A theoretical study based on aspects of Latin verb conjugation*. Cambridge University Press.

Christian Monson, Jaime Carbonell, Alon Lavie, and Lori Levin. 2008. Paramor: finding paradigms across morphology. In *Advances in Multilingual and Multimodal Information Retrieval*, pages 900–907. Springer.

Sylvain Neuvel and Sean A Fulop. 2002. Unsupervised learning of morphology without morphemes. In *Proceedings of the ACL-02 workshop on Morphological and phonological learning-Volume 6*, pages 31–40. Association for Computational Linguistics.

Robert H Robins. 1959. In defence of WP. *Transactions of the Philological Society*, 58(1):116–144.

Patrick Schone and Daniel Jurafsky. 2001. Knowledge-free induction of inflectional morphologies. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–9. Association for Computational Linguistics.

Gregory T. Stump. 2001. *A theory of paradigm structure*. Cambridge University Press.

Tzvetan Tchoukalov, Christian Monson, and Brian Roark. 2010. Morphological analysis by multiple sequence alignment. In *Multilingual Information Access Evaluation I. Text Retrieval Experiments*, pages 666–673. Springer.

Pieter Theron and Ian Cloete. 1997. Automatic acquisition of two-level morphological rules. In *Proceedings of the fifth conference on Applied natural language processing*, pages 103–110. Association for Computational Linguistics.

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. Association for Computational Linguistics.

David Yarowsky and Richard Wicentowski. 2000. Minimally supervised morphological analysis by multimodal alignment. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 207–216. Association for Computational Linguistics.