Learning Transducer Models for Morphological Analysis from Example Inflections

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

Mans Hulden
Department of Linguistics
University of Colorado
mans.hulden@colorado.edu

Abstract

In this paper, we present a method to convert morphological inflection tables into unweighted and weighted finite transducers that perform parsing and generation. These transducers model the inflectional behavior of morphological paradigms induced from examples and can map inflected forms of previously unseen word forms into their lemmas and give morphosyntactic descriptions of them. The system is evaluated on several languages with data collected from the Wiktionary.

1 Introduction

Wide-coverage morphological parsers that return lemmas and morphosyntactic descriptions (MSDs) of arbitrary word forms are fundamental for achieving strong performance of many downstream tasks in NLP (Tseng et al., 2005; Spoustová et al., 2007; Avramidis and Koehn, 2008; Zeman, 2008; Hulden and Francom, 2012). This is particularly true for languages that exhibit rich inflectional and derivational morphology. Finite-state transducers are the standard technology for addressing this issue, but constructing them often requires not only significant commitment of resources but also demands linguistic expertise from the developers (Maxwell, 2015). Access to large numbers of example inflections organized into inflection tables in resources such as the Wiktionary promises to offer a less laborious route to constructing robust large-scale analyzers. Learning morphological generalizations from such example data has been the focus of much recent research, particularly in the domain of morphologically complex languages (Cotterell et al., 2016).

In this paper we present a tool for automatic generation of both probabilistic and non-probabilistic morphological analyzers that can be represented as unweighted and weighted transducers. The assumption is that we have access to a collection of example word forms together with corresponding MSDs. We present two systems: one that is designed to be high-recall and operates with unweighted automata, the purpose of which is to return all linguistically plausible analyses for an unknown word form; the second is an addition to the first in that the word shapes are modeled with a generative probabilistic model that can be implemented as a weighted transducer that produces a ranking of the plausible analyses. The analyzers are constructed with standard finite state tools and are designed to operate similarly to a hand-constructed morphophonological analyzer extended with a 'guesser' module to handle unknown word forms.

The system takes as input sets of lemmatized words annotated with an MSD, all grouped into inflection tables—such as can be found in, for example, the *Wiktionary*. The output is a morphological analyzer either as an unweighted (in the non-probabilistic case) or a weighted model (in the probabilistic case). For the non-probabilistic case we use the Xerox regular expression formalism (Karttunen et al., 1996), which we compile into a transducer with the open-source finite-state toolkit *foma* (Hulden, 2009) and for the weighted case we have used the *Kleene* toolkit (Beesley, 2012).[1]

2 Paradigm Learning

The starting point for the research in this paper is the notion that inflections and derivations of related word forms can be expressed as functions—this idea is often filed under the rubric of 'functional morphology' and is strongly related to word-and-paradigm models of morphology (Hockett, 1954;

[1] Our code and data are available at: https://github.com/marfors/paradigmextract

Proceedings of the ACL Workshop on Statistical NLP and Weighted Automata, pages 42–50,
Berlin, Germany, August 12, 2016.

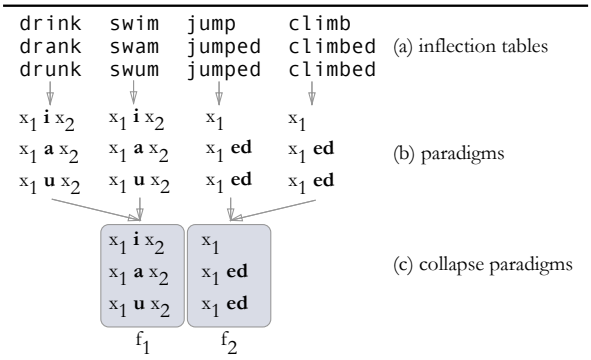


Figure 1: Generalizing inflection tables into paradigm functions: (1) a number of complete inflection tables are given; (2) the aligned Longest Common Subsequence is extracted; (3) resulting identical paradigms are merged. If the resulting paradigm f_1 is interpreted as a function, *f_1(shr, nk)* produces **shrink, shrank, shrunk**.

Robins, 1959; Matthews, 1972; Stump, 2001). In particular, we assume a model where a single function generates all the possible inflected forms of a group of lemmas that behave alike. This approach has earlier been seen as an alternative to finite-state morphology, and the functions that model inflectional behavior have been hand-built in much previous work (Forsberg and Ranta, 2004; Forsberg et al., 2006; Détrez and Ranta, 2012). Here, we assume the recent model of Ahlberg et al. (2014) and Ahlberg et al. (2015), which work with a system that automatically learns these functions that model inflection tables from labeled data.

The purpose of modeling inflection types as functions is to be able to generalize concrete manifestations of word inflection for specific lemmas, and to apply those generalizations to unseen word forms. The generalization in question is performed by extracting the Longest Common Subsequence (LCS) from all word forms related to some specific lemma and then expressing each word form in terms of the LCS (Hulden, 2014). The LCS in turn is broken down into possibly discontiguous sequences that express parts of word forms that are variable in nature. Figure 1 shows a toy example of four inflection tables generalized into variable- and non-variable parts by first extracting the LCS, expressing the original word forms in terms of this LCS, and then collapsing the resulting functions that are identical. The resulting representation, which is essentially a set of strings which have variable parts (x_1, ..., x_n), and fixed parts (such as **i**, **a**, **u**) that can be used to generate an unbounded number of new inflection tables by instantiating the variable parts in new ways and concatenating the variables and the fixed parts.

This learning method often produces a very small number of functions compared with the number of complete inflection tables that have been input—obviously, because many lemmas behave alike and result in identical functions. We note that the output of this procedure is human-readable, i.e. it can be inspected (even in real-world scenarios) for correctness and also hand-corrected in case of noise in the learning data. In the current work, we use these functions as the backbone of a generative model and implement them as transducers that can be run in the inverse direction to map fully inflected forms into their lemmas and morphosyntactic descriptions.

2.1 Paradigm functions

The variables $x_1, \ldots, x_n$ that are used in the paradigm function representation capture possible inter-word variation. This means that each lemma that gives rise to an inflection table can be directly represented as simply an instantiation of the variables, together with the inflection function. As seen in Figure 1, the function f_1 learned from the inflection tables *swim* and *drink* can be used to represent some other word, e.g. *sing* by instantiating x_1 as **s** and x_2 as **ng**.

As we collect a large number of inflection tables, many of which result in identical paradigms, we can also collect statistics about the variables involved and how they were assumed to be instantiated in the original table. For example, from the truncated tables in Figure 1, we can gather that f_1 has witnessed x_1 as both **dr** and **sw**, and x_2 as **nk** and **m**. These statistics can be used to turn the learned functions into a restricted generative model that produces entire inflection tables, but also taking advantage of how variables tend to be instantiated in that paradigm function.

Additionally, since each possible inflected form consists of the same variables, we can also define a string-to-string mapping between any two related forms, where the content of the variable parts stay fixed, and the non-variable parts change. For example, in Figure 1, we know that we can, for some verbs, go from the *past participle* (e.g. **drunk**) to the *past* (e.g. **drank**) by a string transformation $\texttt{x}_1\ \texttt{u}\ \texttt{x}_2 \rightarrow \texttt{x}_1\ \texttt{a}\ \texttt{x}_2$, with some constraints

on the nature of x_1 and x_2. This information can then be encoded in transducer form where the variable parts can be modeled as a probabilistic language model (for weighted transducers) or a non-probabilistic, constrained model (for unweighted transducers).

Figure 2 illustrates this idea. We have learned a paradigm in Spanish—we call the paradigm **avenir** (arbitrarily), since that is one of the verbs out of 12 that behaved the same way and gave rise to the same function. A natural mapping to learn from the data is how to go from any inflected form to the dictionary or 'citation' form. For example, going from the *present participle* to the *infinitive* would involve changing the fixed **i** occurring between the two variables x_1 and x_2 into **e** and then changing the fixed suffix after x_2 from **iendo** to **ir**. The figure also shows how the different variables were instantiated in the training data: x_1 showed up in variable shapes (but always ending in **v**), while x_2 was always **n**.

Although the learning model in principle states nothing about the nature of the variables, morphophonological restrictions will constrain their appearance and the key to producing a transducer that can inflect unseen words without undue overgeneration is to take these restrictions into account. We do so in two ways: (1) for the unweighted case, we collect statistics on the seen variables and constrain their possible shapes in an absolute manner, and (2) for the weighted case, we induce a language model over the shapes of the variables, which can later be used to rank parses produced by the system.

3 The unweighted case: constraining variables

Different generalized inflection tables naturally give rise to different variable instantiations for $x_1, \ldots, x_n$. However, many of the seen variables will not differ arbitrarily in a paradigm. This is something we can take advantage of when designing a parsing mechanism; in particular, we can express preferences to the effect that such parses where variables resemble already seen instantiations should be preferred.

Figure 3 is a case in point. Here, we show the implicit string-to-string rule in the paradigm which derives the lemma form from the present participle and the first person singular present forms in Spanish. In both the paradigms learned, the variables x_1 and x_2 show a somewhat repetitive pattern. In the paradigm **avenir**, x_1 ends in the letter **v** for all the inflection tables seen that produced that paradigm, while x_2 always consists of the single letter **n**. Likewise, in the other paradigm (**negar**), x_2 is consistently the string **eg** across all forms seen (the inflected forms of **cegar**, **denegar**, etc.). The only variable that does not show such a regular pattern is the x_1 variable for the paradigm called **negar**.

3.1 Estimating probabilities of new variable instantiations

That the parts of paradigms that vary from lemma to lemma, i.e. the 'variables', are not subject to arbitrary variation can be used to constrain their shape. To model the unweighted transducers, we begin by formalizing our belief in not seeing novel variable shapes in the future. To quantify this, we assume we have seen n concrete instantiations of t different types of variables, and subsequently ask: if there were in fact $t+1$ types, all of which are drawn from a uniform distribution, how likely are we to have witnessed only the t types we did? This quantity can be expressed as

$$p_{\text{unseen}} = (1 - \frac{1}{t+1})^n \qquad (1)$$

For example, the measure for the x_2 variable in Figure 3 (**avenir**) becomes $(1 - \frac{1}{2})^{12} \approx 0.0002$. We can use this as a cutoff parameter that defines how much evidence we require to declare a variable not subject to further variation apart from the types we have already seen. With this, we assume that if $p_{\text{unseen}} \leq 0.05$ for some variable, that variable in the paradigm will not exhibit new types.[2]

3.2 Expressing constraints through regular expressions

We also expand this measure to cover variables that show variation only in non-edge positions. For example, x_2 in the **avenir**-paradigm in Table 3 is always **n** and can be assumed to not be subject to variation by the calculation above. The paradigm's x_1-variable, however, cannot. That variable seems to vary much more, with the exception of the last letter, which is always **v**. To capture this, we extend the method to apply not only to the whole

[2] Estimating the probability of the existence of unseen types is a classical problem (Good, 1953); see Ogino (1999) and Kageura and Sekine (1999) for linguistics-related discussions and Chen and Goodman (1996) for the relationship to smoothing in language models.

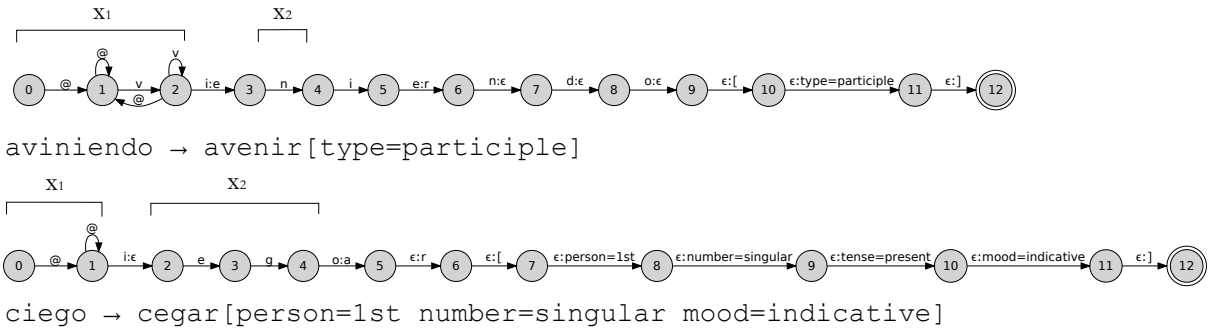


Figure 2: Examples of two single generalized word forms mapped to lemmas followed by morphosyntactic description. The parts that correspond to constraints of the variables x_1 and x_2 are marked. Transitions marked @ are identity transduction 'elsewhere' cases, matching any symbol not explicitly mentioned in the state.

Inflection table	Paradigm	MSD	Inflection table	Paradigm	MSD
avenir	x_1+e+x_2+ir	infinitive	**neg**ar	x_1+x_2+ar	infinitive
aviniendo	x_1+i+x_2+iendo	pres part	**neg**ando	x_1+x_2+ando	pres part
avenido	x_1+e+x_2+ido	past part	**neg**ado	x_1+x_2+ado	past part
avengo	x_1+e+x_2+go	1sg pres ind	**nieg**o	x_1+i+x_2+o	1sg pres ind
avienes	x_1+ie+x_2+es	2sg pres ind	**nieg**as	x_1+i+x_2+as	2sg pres ind

Table 1: Two partial Spanish verb inflection tables generalized into paradigm functions. The segments that are part of the longest common subsequence, which are cast as variables in the generalization, are shown in boldface in the inflection tables.

string, but also edge positions of the string. First, we examine the whole string, and if that fails to yield the conclusion that the variable is 'fixed', we find the longest prefix and suffix which can be assumed to be fixed by the same measure. With this we construct a regular expression that models the variables as follows:

1. $(w_1 \cup w_2 \cup \ldots \cup w_n)$ if the variable is assumed to be fixed, where the w_is are the complete strings seen as instantiations.

2. $(p_1 \cup \ldots \cup p_n)\Sigma^* \cap \Sigma^*(s_1 \cup \ldots \cup s_n)$, if both prefixes and suffixes can be constrained; here the p_is correspond to the prefixes of the maximal length that can be assumed to be drawn from a fixed set of types, and the s_is the suffixes.

3. $(p_1 \cup \ldots \cup p_n)\Sigma^*$ if only prefixes appear fixed.

4. $\Sigma^*(s_1 \cup \ldots \cup s_n)$ if only suffixes appear fixed.

5. Σ^+ otherwise.

In the above, Σ represents all the symbols seen in the training data. Under this formulation, the variables in the **avenir** paradigm in Figure 3 yield the following regular expressions:

$$x_1 = (\Sigma^* v) \;\; x_2 = n \tag{2}$$

4 Deriving morphological analyzers

Once we have the constraints in place, they can be used to construct larger regular expressions that reflect mappings from a specific word form to a lemma together with the MSD. We convert each inflection form in a paradigm to a regular expression that permits the above variable values in place of $x_1, \ldots, x_n$, and that maps the remaining fixed strings to other fixed strings, depending on what kind of application is needed.

For example, to create a regular expression for mapping the *1p pres ind*-form (exemplified by **niego**) to the lemma form in Table 1, we proceed as follows: we construct a transducer that repeats the x_1 and x_2-variables, possibly subject to the constraints on their shape, and maps an **i** to the empty string and **o** to **ar**. Since x_1 is not constrained in the paradigm, while x_2 is constrained to always be the string **eg**, this produces the following regular expression:

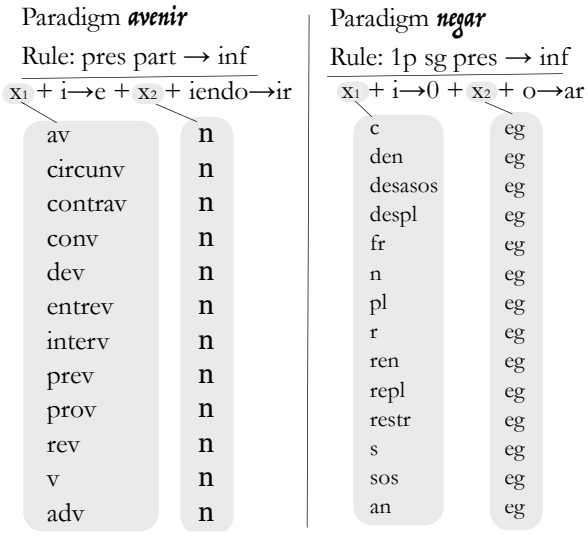


Figure 3: Paradigm functions generalized from inflection tables provide a mechanism for mapping an inflected form to any other inflected form. Illustrated here are two rules extracted from different Spanish verb paradigms showing a string-to-string mapping from the participle to the infinitive, and from the first person singular present form to the infinitive. Also shown are the variable parts of the paradigms x_1 and x_2 and how they have been instantiated in the training data.

$$\underbrace{(\Sigma^+)}_{x_1}\ (i{:}\epsilon)\ \underbrace{e\ g}_{x_2}\ (o{:}ar[\text{1sg pres ind}]) \quad (3)$$

The transducer corresponding to the expression is seen in Figure 2, and will generalize to words that fit the variable pattern, e.g. **ciego** → **cegar**.

Each inflection form of every paradigm is converted in such a manner to a transducer that maps that single inflection to its lemma and morphosyntactic description. All such individual transducers can then be unioned together for every form in every paradigm:

$$f_1 \cup f_2 \cup \ldots \cup f_1 \cup \ldots \cup f_m \quad (4)$$

5 Prioritizing analyses

The above formulation, though it already produces a working transducer that generalizes to unseen forms, can be refined further. First, if a word form matches the original variables seen exactly, it may be superfluous to return extra analyses from other paradigms that the word form might also fit. Secondly, it may be the case that we have overconstrained some variable with the heuristic described earlier, and so return no analyses at all, motivating a potential relaxation of the constraints on variable shapes.

	Analysis: **peleaste**	
O	**pelear**	**[pers=2 number=sg tense=past mood=ind]**
C	peleastar	[pers=1 num=sg tense=pres mood=subj]
	peleastar	[pers=3 num=sg tense=pres mood=subj]
	peleastir	[pers=3 num=sg tense=pres mood=ind]
	pelear	**[pers=2 num=sg tense=past mood=ind]**
U	peleaster	[pers=3 num=sg tense=pres mood=ind]
	peleastar	[pers=1 num=sg tense=pres mood=subj]
	peleastar	[pers=3 num=sg tense=pres mood=subj]
	peleastir	[pers=3 num=sg tense=pres mood=ind]
	pelear	**[pers=2 num=sg tense=past mood=ind]**
	pelastir	[pers=3 num=sg tense=pres mood=ind]
	pleastir	[pers=3 num=sg tense=pres mood=ind]
	Analysis: **aceleran**	
O	???	???
C	**acelerar**	**[pers=3 num=pl tense=pres mood=ind]**
	acelerir	[pers=3 num=pl tense=pres mood=subj]
U	**acelerar**	**[pers=3 num=pl tense=pres mood=ind]**
	aceler	[pers=3 num=pl tense=imp-ra mood=subj]
	acelerir	[pers=3 num=pl tense=pres mood=subj]
	acelerer	[pers=3 num=pl tense=pres mood=subj]
	acelrir	[pers=3 num=pl tense=pres mood=subj]
	acelir	[pers=3 num=pl tense=imp-ra mood=subj]
	aclerir	[pers=3 num=pl tense=pres mood=subj]

Table 2: Example of the tri-level analyses produced by the unweighted system: here the three sub-grammars (**Original = O**, **Constrained = C**, **Unconstrained = U**) each allow for successively more analyses. The word **peleaste** 'quarrel' has been seen in the training data and thus receives an analysis from the constrained analyzer, whereas **aceleran** 'accelerate' has not and only receives parses from **C** and **U**.

To provide a ranking of the analyses in the unweighted analyzer, we actually generate a layered approach with three different models:

- **Original**: an analyzer where each x_i must match exactly some shape seen in training data.
- **Constrained**: an analyzer where variables are constrained as described above.
- **Unconstrained**: an analyzer where there are no constraints on variables, except that they must be at least one symbol long, i.e. match Σ^+.

The three analyzers can be joined by a "priority union" operation (Kaplan, 1987), in effect producing a single analyzer that prioritizes more constrained analyses, if such are possible: **Original** $\cup_P$ **Constrained** $\cup_P$ **Unconstrained**.

This in effect leads to an analyzer that can be thought of as first consulting **Original**, and that failing to produce an analysis, consults **Constrained**, and if that also fails, consults **Unconstrained**. The same effect can also be modeled in runtime code by keeping the three transducers separate for potential savings of space. Table 2 illustrates this priority effect with two Spanish words being analyzed.

6 The weighted case: language models over variables

The above unweighted model provides a hierarchical system by which to return plausible analyses, while curbing implausible ones. However, it lacks the power to provide a ranking of analyses within each layer of ever laxer constraints on the variables. An alternative to that model is to directly use the statistics over the variable parts to generate a weighted transducer that performs the same type of parsing, but with a (hopefully) strict ranking of candidate parses. We address this by inducing an n-gram model over each variable in each paradigm. We calculate these individual n-gram models in the usual way for a single variable **v**, consisting of the letters $v_1, \ldots, v_n$:

$$P(v_1, \ldots, v_n) = \prod_{i=1}^{n} P(v_i | v_{i-(n-1)}, \ldots, v_{i-1}) \quad (5)$$

For each variable and function, we perform a standard maximum likelihood estimate of the n-grams by

$$P(v_i | v_{i-(n-1)}, \ldots, v_{i-1}) = \frac{\#(v_{i-(n-1)}, \ldots, v_{i-1}, v_i)}{\#(v_{i-(n-1)}, \ldots, v_{i-1})} \quad (6)$$

with some additional add-δ smoothing to prevent zero counts. The resulting variable models can then (after taking negative logs) replace the variable portions of each individual transducer that maps a word form to its citation form. The fixed parts of the inflection mappings retain the weight 0.

These language models are then concatenated with the same model as used for the unweighted case in place of the variables. This is illustrated in Figure 4.

We tune the model for each language evaluated by doing a grid search on (1) the order of the n-gram (1–5), (2) the prior on the n-grams (0.01–3.0), (3) the prior of picking a paradigm (we include a paradigm weight for each individual paradigm).

Similarly to the unweighted case, the final model is a union of all the individual inflection models for each paradigm and word, with the language models for the variables interleaved.

7 Evaluation

To evaluate the systems, we used the data set published by Durrett and DeNero (2013) (D&DN13), which includes full inflection tables for a large number of lemmas in German (nouns and verbs), Spanish (verbs), and Finnish (nouns+adjectives and verbs). That source also provides a division into train/dev/test splits, with 200 tables in dev and test, respectively. We then evaluated the ability of our systems to provide a correct lemmatization and MSD of each word form in the held-out tables, testing separately on each part of speech. For the unweighted analyzer, we use the three-part setup as described above. For the weighted case, we produce a single highest scoring analysis. The train/dev/test sets are entirely disjoint and share no tables.

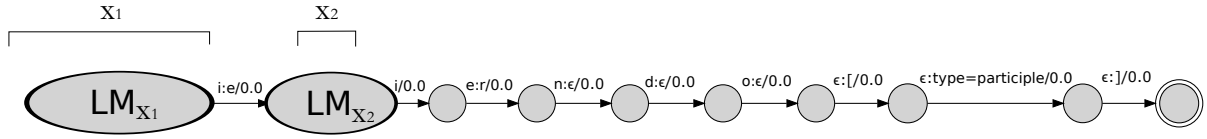
We trained the models by inspecting all the word forms and corresponding MSDs, organizing them into tables, learning the paradigms, and the generating weighted and unweighted transducers as described above. These transducers were then run on the test data to provide lemmatization and analyses of the unseen word forms. Table 4 summarizes the number of inflection tables seen during training, together with the final number of paradigms learned. Table 5 shows the statistics in the held-out data.

Because we focus on the recall figures of the analyzers, we also calculated an "inherent ambiguity" measure of the test data. This is the average number of different MSDs that are given for each word form. This ambiguity may arise as follows: the Spanish verb **tenga**, for example, can be either the first person singular present subjunctive of **tener** 'to have' or the third person singular present subjunctive. Such ambiguity shows that there exist cases where returning multiple analyses is warranted, given that we do not have any sentence context to determine the correct choice.

For the weighted case, sometimes the system returns multiple equally scoring parses. This is due to the fact that the language model only operates over the variables, and, in many languages multi-

Language		L-recall	L+M-recall	L/W	L+M/W
German	nouns	95.30	95.06	2.08	9.52
	verbs	91.18	92.44	4.16	9.57
	nouns+verbs	92.11	93.04	4.91	14.10
Spanish	verbs	98.06	97.98	1.93	2.20
Finnish	nounadj	88.69	88.48	4.10	5.30
	verbs	94.52	94.47	3.77	4.60
	nounadj+verbs	92.63	92.43	12.56	16.40

Table 3: The result of the unweighted evaluation, where we report separately on the recall of just the lemma (L-recall), and the recall of the lemma and corresponding MSD (L+M-recall). Also shown are the average number of unique lemmas returned per word form to be analyzed (L/W), and the average number of lemmas and MSDs returned (L+M/W).



`aviniendo → avenir[type=participle]`

Figure 4: Illustration of the coupling of language models for variables x_1 and x_2 to create the weighted analyzer. Here, LM_{x_1} and LM_{x_2} illustrate a collection of states representing the language models for the variables, inferred from variable instantiations seen in the training data.

Language		Tables	Paradigms
German	nouns	2564	70
	verbs	1827	139
	nouns+verbs	4391	209
Spanish	verbs	3855	96
Finnish	nounadj	6200	259
	verbs	7049	276
	nounadj+verbs	13249	535

Table 4: Statistics on the D&DN13 train+dev sets. **Paradigms** is the corresponding number of induced paradigm functions.

Language		Tables	Unique wf's	Amb.
German	nouns	200	553	2.89
	verbs	200	2324	2.32
	nouns+verbs	400	2877	2.43
Spanish	verbs	200	10003	1.14
Finnish	nounadj	200	5198	1.08
	verbs	200	10466	1.03
	nounadj+verbs	400	15664	1.05

Table 5: Statistics on the D&DN13 test set. **Amb.** is the average number of lemma-MSD pairs per unique word form (wf).

Language		Lemma	L+MSD	MSD
German	nouns	77.06	69.44	79.50
	verbs	90.02	89.76	92.78
Spanish	verbs	96.92	96.92	97.43
Finnish	nounadj	70.29	69.68	91.59
	verbs	90.44	90.44	98.02

Table 6: Evaluation of the weighted model (all figures represent the recall).

ple MSDs often have the same surface form. For example, Spanish **compraba** 'bought 1P/3P' (and **-aba** suffix-bearing verbs in general) are always ambiguous between 1st/3rd past tense. For this reason, we calculate the recall (as opposed to accuracy) of all the top scoring parses. The weighted system always returns a single lemma in the evaluation. It can, of course, produce a number of ranked analyses if needed—an example of extracting the top-10 ranked analyses of a word form is given in Table 7.

7.1 Results

Table 3 shows the main results of the evaluation of the unweighted model and Table 6 the results of the weighted model. For the unweighted case,

rank	w	paradigm	vars	lemma	analyses
1	14.10	p1_abadernar	(1=compr)	**comprar**	[**pers=2 num=sg tense=past mood=ind**]
2	18.22	p1_abadernar	(1=comprast)	comprastar	[pers=1 **num=sg** tense=pres mood=sub]
				comprastar	[pers=3 **num=sg** tense=pres mood=sub]
3	23.57	p5_abogar	(1=compr)	**comprar**	[**pers=2 num=sg tense=past mood=ind**]
4	24.58	p4_abolir	(1=comprast)	comprastir	[pers=3 **num=sg** tense=pres **mood=ind**]
5	24.58	p8_acrecentar	(1=com,2=pr)	**comprar**	[**pers=2 num=sg tense=past mood=ind**]
6	25.51	p37_colgar	(1=c,2=mpr)	**comprar**	[**pers=2 num=sg tense=past mood=ind**]
7	26.20	p10_acostar	(1=c,2=mpr)	**comprar**	[**pers=2 num=sg tense=past mood=ind**]
8	26.61	p7_acceder	(1=comprast)	compraster	[pers=3 **num=sg** tense=pres **mood=ind**]
9	26.87	p8_acrecentar	(1=comp,2=r)	**comprar**	[**pers=2 num=sg tense=past mood=ind**]
10	29.98	p20_cegar	(1=c,2=ompr)	**comprar**	[**pers=2 num=sg tense=past mood=ind**]

Table 7: Weighted parsing example: top-10 ranked parses for the word form **compraste** 'buy PAST' in Spanish with weights (in effect the negative log probability), the inferred variable division, the lemmatization, and MSDs. Lemmas and parts of the analysis that are correct are given in boldface. Note that several paradigms can produce an entirely correct parse for a single form such as this one, even though the paradigms would differ in other forms.

we consider the lemma-recall and lemma+MSD recall, and also document the average number of unique parses returned (lemma or lemma+MSD). For the weighted model, we give the recall for all combinations of lemma+MSD.

The weighted recall is—for obvious reasons—consistently below the unweighted version as the unweighted case uses the hierarchical model to potentially return a much larger number of analyses. The weighted version always returns a single lemma, and possibly several equally ranked MSDs, as discussed above. Still, for some languages (Spanish and Finnish verbs in particular), despite returning only a single analysis, performance is on par with the unweighted model, which returns 1.93 analyses on average (Spanish) and 3.77 (Finnish). We emphasize that the test set for our experiments is entirely disjoint from the training set, and that the figures therefore reflect potential performance on unseen word forms, not standard per-token performance in running text, which is presumably much higher. The reported figures can thus be interpreted to correspond to a per-type performance for OOV items.

8 Conclusion and future work

We have described two supervised methods for producing finite-state models morphological analyzers and guessers from labeled word forms, organized into inflection tables. The method can be used to quickly produce high-recall morphological analysis from labeled data with little or no linguistic development effort.

These tools can be used as is and can also be modified to exploit unlabeled data in the form of raw text corpora in a semi-supervised lexicon expansion setting. Some potential extensions could be of immediate value: the generative weighted model could be combined and evaluated on a task of tagging/disambiguating running text where contextual features could be used and seamlessly combined with the morphological language model. The weighted model also offers paths for further experimentation—for example, it is not immediately obvious that an n-gram model is the best choice. It seems reasonable to assume that those parts of the variables modeled that stand closer to the fixed parts, i.e. at the edges, would be more important in judging similarity to previously seen inflected forms. Table 2 hints at this being the case since, for example, the Spanish variables seem far more constrained at edge positions than in the middle of the variable string. Which parts to weight as more important in judging similarity could also be inferred from data. Another potential extension is to also constrain the analysis form by integrating a word-level language model instead of only a variable-level one, either replacing the variable-level model or working in conjunction with it.

Acknowledgements

This work has been partly funded by the Swedish Research Council under grant number 2012-5738, *Towards a knowledge-based culturomics* and the University of Gothenburg through its support of the Centre for Language Technology and its support of Språkbanken.

References

Malin Ahlberg, Markus Forsberg, and Mans Hulden. 2014. Semi-supervised learning of morphological paradigms and lexicons. In *Proceedings of the 14th EACL*, pages 569–578, Gothenburg, Sweden. Association for Computational Linguistics.

Malin Ahlberg, Markus Forsberg, and Mans Hulden. 2015. Paradigm classification in supervised learning of morphology. In *Proceedings of the NAACL-HLT 2015*, pages 1024–1029, Denver, Colorado, May–June. Association for Computational Linguistics.

Eleftherios Avramidis and Philipp Koehn. 2008. Enriching morphologically poor languages for statistical machine translation. *NAACL-HLT 2008*, pages 763–770.

Kenneth R. Beesley. 2012. Kleene, a free and open-source language for finite-state programming. In *10th International Workshop on Finite State Methods and Natural Language Processing*, page 50.

Stanley F. Chen and Joshua Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th ACL*, pages 310–318. Association for Computational Linguistics.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. 2016. The SIGMORPHON 2016 shared task—morphological reinflection. In *Proceedings of the 2016 Meeting of SIGMORPHON*, Berlin, Germany, August. Association for Computational Linguistics.

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.

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

Markus Forsberg and Aarne Ranta. 2004. Functional morphology. *ACM SIGPLAN Notices*, 39(9):213–223.

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.

Irving J. Good. 1953. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40(3-4):237–264.

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

Mans Hulden and Jerid Francom. 2012. Boosting statistical tagger accuracy with simple rule-based grammars. In *Proceedings of the Language Resources and Evaluation Conference (LREC 2012)*, pages 2114–2117.

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

Mans Hulden. 2014. Generalizing inflection tables into paradigms with finite state operations. In *Proceedings of the 2014 Joint Meeting of SIGMORPHON and SIGFSM*, pages 29–36. Association for Computational Linguistics.

Kyo Kageura and Satoshi Sekine. 1999. A note on Ogino's "method to estimate probability of new appearance". *Journal of Mathematical Linguistics*, 22(3).

Ronald M. Kaplan. 1987. Three seductions of computational psycholinguistics. In P. Whitelock, M. M. Wood, H. L. Somers, R. Johnson, and P. Bennett, editors, *Linguistic Theory and Computer Applications*, London. Academic Press.

Lauri Karttunen, Jean-Pierre Chanod, Gregory Grefenstette, and Anne Schiller. 1996. Regular expressions for language engineering. *Natural Language Engineering*, 2(4):305–328.

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

Michael Maxwell. 2015. Grammar debugging. In *Systems and Frameworks for Computational Morphology*, pages 166–183. Springer.

T. Ogino. 1999. How many examples are required in language research—a proposal of a method to estimate probability of new appearance. *Mathematical Linguistics*, 22(1):11–17.

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

Drahomíra Spoustová, Jan Hajič, Jan Votrubec, Pavel Krbec, and Pavel Květoň. 2007. The best of two worlds: Cooperation of statistical and rule-based taggers for Czech. In *Proceedings of the Workshop on Balto-Slavonic Natural Language Processing*, pages 67–74.

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

Huihsin Tseng, Daniel Jurafsky, and Christopher Manning. 2005. Morphological features help POS tagging of unknown words across language varieties. In *Proceedings of the fourth SIGHAN workshop on Chinese language processing*, pages 32–39.

Daniel Zeman. 2008. Reusable tagset conversion using tagset drivers. In *Proceedings of the Language Resources and Evaluation Conference (LREC 2008)*.