Multiple alignments of inflectional paradigms

Sacha Beniamine
University of Surrey,
Guildford, England*

Matías Guzmán Naranjo
Université de Paris,
Paris, France

Abstract

Most models of inflectional morphology rely at their core on the identification of recurrent and diverging material across inflected forms. Across theoretical frameworks, this can be expressed in terms of morpheme segmentation, rules, processes, patterns or analogies.

Finding these recurrences in large structured lexicons is an important step in empirical computational morphology, where analyses are induced bottom-up from inflected forms. This can be done by aligning all the forms in each paradigm, a task of Multiple Sequence Alignments which is well known in other fields such as evolutionary biology and historical linguistics.

In this paper, we present the specific problems which arise when aligning inflected forms, provide a simple alignment format, define evaluation measures and compare two implemented methods on 13 inflectional lexicons. Our intent is to provide the conditions for the interoperability of future systems, and for incremental improvements in this fundamental step for quantitative morphology.

1 Introduction

When analyzing inflectional systems, linguists draw on complex intuitions informed by segmentation conventions, diachronic evidence and their own sense of grammatical elegance. As a result, these analyses are not meant to produce commensurate units across languages. This is an obstacle in comparative research such as linguistic typology and evolutionary linguistics. The alternative is to induce inflectional generalizations bottom-up, starting from paradigms of word forms, using the same procedure across languages.

*This work was done in part while working at the Max Planck Institute for the Science of Human History, Jena, and at the Max Planck Institute for Evolutionary Anthropology, Leipzig.

PRS.IND.1SG	l	i	b	ɛ	ɾ	t	-	-	u	-
PRS.IND.2SG	l	i	b	ɛ	ɾ	t	ɐ	-	-	ʃ
PRS.IND.3SG	l	i	b	ɛ	ɾ	t	ɐ	-	-	-
PRS.IND.1PL	l	i	b	ə	ɾ	t	ɐ	m	u	ʃ
PRS.IND.2PL	l	i	b	ə	ɾ	t	a	-	i	ʃ
PRS.IND.3PL	l	i	b	ɛ	ɾ	t	ɐ̃	-	ũ	-
indexes	0	1	2	3	4	5	6	7	8	9

Table 1: Alignment for a sub-paradigm of the European Portuguese verb LIBERTAR, 'to free'

This program resembles that of the unsupervised learning of morphology (Goldsmith, 2001; Creutz and Lagus, 2002; Kurimo et al., 2010), but differs in its goals: rather than reproducing morphemic segmentations or scaling them up for NLP, we seek to generate new inflectional analyses for empirical linguistics. This leads to crucial divergences in data, tasks and evaluation.

Hockett (1954) classifies inflectional models according to the units they manipulate (morphemes, roots, markers, words) and the way these are related to each other (rules, processes, analogies). Despite this considerable variation, most models of inflection rely at their core on the identification of recurrent and diverging material across inflected forms.

We propose that this fundamental step of grammatical description can be performed by aligning all the surface forms of a same lexeme, as shown in Table 1[1].

Most systems which attempt to induce inflectional rules from paradigms do start by aligning forms (Albright and Hayes, 2002; Durrett and DeNero, 2013; Ahlberg et al., 2014; Bonami and Boyé, 2014; Beniamine, 2017; Guzmán Naranjo, 2020; Guzmán Naranjo and Becker, in press). But so far,

[1]For the sake of brevity we show only the present subparadigm, though the full alignment has 69 rows.

Proceedings of the Society for Computation in Linguistics (SCiL) 2021, pages 216-227.
Held on-line February 14-19, 2021

contributions have been devised only in the context of specific applications. As a result, there is no existing generic solution, and the alignments themselves are not evaluated, even though they impact downstream analyses. Instead, we suggest to make multiple alignments a common modular step for the induction of various types of inflectional analyses.

The goal of this paper is to introduce the task of multiple alignments in inflectional paradigms. In order to facilitate incremental improvements, we define a simple format and provide evaluation measures of alignment quality. Moreover, we describe and evaluate two systems on 13 inflectional lexicons: the nouns of Hungarian, Kasem, Latin, Latvian, and Russian; and the verbs of Modern Standard Arabic, English, French, Latin, Navajo, European Portuguese, Yaitepec Chatino and Zenzontepec Chatino.

2 Related work

The two questions which underpin this work are general enough to have seen contributions from multiple disciplines. On one hand is the issue of how to align sequences together. Several methods were elaborated in biology in order to compare DNA or protein sequences (Durbin, 1998) and then adapted to align strings of phonemes in historical linguistics and phonology. The second is the *Segmentation Problem* (Spencer, 2012): how can we segment inflected forms algorithmically? This is relevant in NLP in the unsupervised learning of morphology (Goldsmith, 2001) as well as for the reinflection task (Cotterell et al., 2016), and in theoretical and quantitative morphology, where it figures as a nearly unavoidable step.

As in the task of unsupervised morphological segmentation (Goldsmith, 2001; Creutz and Lagus, 2002; Kurimo et al., 2010), we are concerned with identifying morphologically motivated sub-word recurrences. However, our task differs in several important respects. First, we take as input structured paradigms rather than lists of forms. Second, we take inflected forms to be strings of phonemes, not orthographic strings. Third, we seek to identify only inflectional morphology. Fourth, alignments do not entirely determine segmentations, and the segmentations which can be inferred from alignments do not need to coincide with traditional morphemic analyses. Finally, it would not make sense for us to evaluate alignment algorithms against a gold standard, as our intent is to induce new descriptions: it is then impossible to select in advance a single preferred segmentation.

Aligning two sequences together is a task of PAIRWISE ALIGNMENT, which can be solved optimally for a given scoring scheme using the dynamic Needleman-Wunsch algorithm (Needleman and Wunsch, 1970), also called Wagner-Fischer (Wagner and Fischer, 1974) algorithm in the context of Levenshtein distances. Aligning more than two sequences is a problem of MULTIPLE SEQUENCE ALIGNMENT, which is NP-complete. Various heuristic algorithms have been devised to find good multiple alignments of DNA sequences (for a review, see Durbin, 1998). Two methods which make use of repeated pairwise alignments are relevant to the current work. In ITERATIVE approaches (Barton and Sternberg, 1987), a first solution is obtained by repeated pairwise alignments, then it is refined by repeatedly taking a sequence out of the alignment and re-aligning it to the others. PROGRESSIVE approaches use hierarchical clustering to calculate a similarity tree between sequences, then align (clusters of) sequences two by two following this guide tree (Feng and Doolittle, 1987).

Both pairwise and multiple alignments have been used in historical linguistics for cognate detection, aligning sequences of phonemes across languages. Covington (1996) and Kondrak (2000, 2003) use pairwise alignments with phonetically weighted scores. Kondrak (2003) implements several improvements from the biology literature. In particular, he retrieves a set of alignments rather than a single best alignment. List (2010, 2014) adapted multiple progressive alignment methods to the alignment of cognate words, relying on the simplification of sequences into classes of sounds which are likely to be historically related. This method, called Sound Class Alignment (SCA) simultaneously reduces the number of sequences to be aligned and the overall vocabulary size, while introducing a simple sensitivity to phonetic similarity. While this is linguistically motivated for cognate detection, it does not fit the needs of language-internal comparisons.

The need to align inflected forms is a leitmotiv in symbolic approaches to reinflection and paradigm completion. Durrett and DeNero (2013) align all forms pairwise to a selected 'base form', then perform iterative alignments. Nicolai et al. (2015) uses a modified version of the EM-driven, many-to-

many aligner from Jiampojamarn et al. (2007). The baselines for Cotterell et al. (2016, 2017) rely on pairwise alignments. Rather than directly aligning sequences, Ahlberg et al. (2014) find the longest common subsequence (LCS) in paradigms. The most successful solutions to reinflection are neural, and do not provide explicit alignments.

In quantitative morphology, most approaches so far have used pairwise alignments obtained through heuristics such as justifying all forms left or right (Albright and Hayes, 1999, 2002, 2003; Bonami and Boyé, 2014). Beniamine et al. (2017a)[2] use left justification for both pairwise and multiple alignments. Albright and Hayes (2006); Beniamine (2017), find pairwise alignments using the dynamic algorithm and a phonologically weighted edit distance based on Frisch (1997). The use of pairwise alignments fits with the emphasis on implicative relations and the Paradigme Cell Filling Problem in Word of Paradigm morphology. However, they can not support other types of generalizations such as affixal segmentations. Patterns result in very large analyses which are difficult to evaluate by hand. Moreover, strictly pairwise alignments are sometimes counter-intuitive, as they neglect information available in the rest of the paradigm.

3 Inflectional alignments

Input lexicons consist of triples of ⟨ *lexeme identifier, paradigm cell, inflected form* ⟩, and can be written in tabular format as shown in the first three columns of Table 2. Each inflected form of a lexeme is a sequence of phonological segments. Since a segment can be written by more than one character, we separate phonemes by spaces in the table.

An alignment of n sequences of maximal length m, is a matrix with n rows and at least m columns (see Table 1). Rows of the matrix represent sequences, and columns represent matches across sequences. Gaps, noted '-', pad empty cells and represent the absence of a match.

Alignment matrices of entire paradigms can have a large number of columns, which makes them impractical to write. We submit instead a simple sparse format: each row of the matrix is encoded as the space separated list of non-gap indexes, as in Table 2, which describes the matrix from Table 1. Given a set of sequences and these indexes, it is trivial to reconstitute the matrix.

The semantics of a match varies according to the purpose of alignments. In evolutionary biology as in historical linguistic, they express common ancestry. In the case of inflectional paradigms, matches relate recurrent material, that is phonological segments which can be identified by speakers as being "the same" across sequences, whether they are identical or not. Identical matches can be part of invariant lexical material (the initial /l/ in Table 1), or be common to only some cells (for PRS.IND.2SG, PRS.IND.3SG and PRS.IND.1PL share /ɐ/ in column 6). Non identical matches relate distinct segments when a clear correspondence can be identified, e.g /ɐ/, /a/ and /ɐ̃/ in column 6 or /ɛ/ and /ə/ in column 3.

3.1 Downstream applications

Alignments of all the paradigms in a lexicon can support the computational induction of entire grammars, both in word-based or morpheme-based approaches, in the form of processes, analogies, segmentations, or other descriptive devices.

For example Table 1 supports a segmentation into suffixes /-u/, /-ɐʃ/, /-ɐ/, /-ɐmuʃ/, /-ɐiʃ/, /-ɐ̃ũ/ and two stem allomorphs /libɛrt-/ and /libərt-/. The same information could serve as a basis to compute alternation patterns such as [ɛ_ ⥋ ə_ɐm_ʃ / lib_ɾt_u_] relating the PRS.IND.1SG and PRS.IND.1PL cells.

Providing procedures to extract such analyses is beyond the goals of the current paper. For the purposes of discussing and evaluating our systems, we provide however two simple definitions for units which can be readily deduced from alignments and bear some resemblance to more usual linguistic abstractions:

- CONSTANT STEM: the discontinuous sequence of phonemes which expresses no inflectional meaning. It is extracted from identical match columns in the alignment matrix.
- MARKERS: the discontinuous substrings remaining in each form after removing the constant stem.

For example, in Table 1, the constant stem *lib-ɾt-* is highlighted in gray columns, and the marker for the PRS.IND.1SG is /-ɛ-u/. Constant stems can not, by definition, have allomorphs. As a result, suppletion is either analyzed as an empty stem, transferring entire words to markers, or as a short

[2] They use the terms "local" and "global" for resp. pairwise and multiple alignments. This choice is unfortunate, as these terms already have an established meaning (Durbin, 1998; List, 2014), and refer resp. to partial alignments of sequences and complete alignments of entire sequences.

lexeme	cell	form	alignment
LIBERTAR	PRS.IND.1SG	l i b ɛ ɾ t u	0 1 2 3 4 5 8
LIBERTAR	PRS.IND.2SG	l i b ɛ ɾ t ɐ ʃ	0 1 2 3 4 5 6 9
LIBERTAR	PRS.IND.3SG	l i b ɛ ɾ t ɐ	0 1 2 3 4 5 6
LIBERTAR	PRS.IND.1PL	l i b ə ɾ t ɐ m u ʃ	0 1 2 3 4 5 6 7 8 9
LIBERTAR	PRS.IND.2PL	l i b ə ɾ t a i ʃ	0 1 2 3 4 5 6 8 9
LIBERTAR	PRS.IND.3PL	l i b ɛ ɾ t ɐ̃ ũ	0 1 2 3 4 5 6 8

Table 2: Tabular format for aligned paradigms

common substring. To find suppletive and allomorphic stems, further processing would be necessary to determine stem alternants. To find morphemes or exponential sub-strings, it would be necessary to contrast systematically sets of forms according to shared grammatical features in their cells. Phonological similarity could be leveraged in order to identify allomorphy in both stems and exponents.

3.2 Challenges

In some suffixal systems, a left alignment of all sequences is enough to identify the constant stem, and conversely a right alignment may work for some prefixal systems. However, this is inadequate in the presence of allomorphy (which may result in variable stem lengths) and will always fail to identify recurrences across markers. Good solutions to the problem should be able to identify various types of alignments, including non-concatenative morphology, without knowing in advance the type of exponence to expect.

lexeme	cell	form
DESESPERAR	INF	d ə z ə ʃ p ə ɾ a ɾ
DESESPERAR	PST.IND.1SG	d ə z ə ʃ p ɛ ɾ u
DESESPERAR	COND.1PL	d ə z ə ʃ p ə ɾ a ɾ i ɐ i ʃ

Table 3: Three forms of the European Portuguese verb DESESPERAR

1.	d	ə	z	ə	ʃ	p	ə	ɾ	a	ɾ	-	-	-	-	-
	d	ə	z	ə	ʃ	p	ɛ	ɾ	-	-	u	-	-	-	-
	d	ə	z	ə	ʃ	p	ə	ɾ	ɐ	ɾ	-	i	ɐ	i	ʃ
2.	d	ə	z	ə	ʃ	p	ə	ɾ	a	ɾ	-	-	-	-	-
	d	ə	z	ə	ʃ	p	ɛ	-	-	ɾ	u	-	-	-	-
	d	ə	z	ə	ʃ	p	ə	ɾ	ɐ	ɾ	-	i	ɐ	i	ʃ

Table 4: Two alignments for the forms of Table 3

Repeated material between stems and markers can make it impossible to choose between conflicting alignments without looking at other paradigms. As an example, Table 3 presents three forms of the Portuguese lexeme DESESPERAR. Table 4 shows two alignments of these forms which only differ by the position of the phoneme /ɾ/ shown in shaded cells. These alignments could be scored identically by an alignment algorithm, yet only the first one appears linguistically motivated: no lexemes have infinitives in /-ra-/, but many do have an infinitive in /-ar/. This problem justifies a strategy in two steps (following Beniamine, 2017): (i) generate competing hypotheses, and (ii) compare across lexemes to select the most general hypotheses.

Non-identical matches in alignment can support the identification of morpho-phonological changes and suppletion. As an example, Table 5 illustrates palatalization in Russian nouns. To recognize the palatalization in the dative and locative singular, the relevant consonants need to be aligned together. Doing so may require a scoring scheme sensitive to phonology, in order to prefer a match of the palatalized /mʲ/ with the first /m/ of /zʲˈimamʲi/ rather than with the identical phoneme /mʲ/.

lexeme	CELL	form
ZʲIMˈA	NOM.SG	zʲi m ˈa
ZʲIMˈA	ACC.SG	zʲi m ˈu
ZʲIMˈA	GEN.SG	zʲi m ˈi
ZʲIMˈA	DAT.SG	zʲi mʲ ˈe
ZʲIMˈA	INS.SG	zʲi m ˈo j
ZʲIMˈA	LOC.SG	zʲi mʲ ˈe
ZʲIMˈA	NOM.PL	zʲˈi m i
ZʲIMˈA	ACC.PL	zʲˈi m i
ZʲIMˈA	GEN.PL	zʲˈi m
ZʲIMˈA	DAT.PL	zʲˈi m a m
ZʲIMˈA	INS.PL	zʲˈi m a mʲ i
ZʲIMˈA	LOC.PL	zʲˈi m a x

Table 5: Paradigm of the Russian noun ZʲIMˈA, ‘winter’

The detection of some phenomena require extensions to the definition of alignments provided above. Metathesis rearranges the order of segments. Table 6 describes an example of morphological metathesis from Saanich, a Central Salishan language (Kurisu, 2001) and how it can be coded using our alignment format. Metathesis is more challenging to code in a typical alignment matrix (see List, 2014). Consonant gemination and reduplication can not be written in simple alignment matrices either but our format would allow the mapping of more than a single phoneme to a column (*one-to-many* alignments). Since these phenomena could all be identified by post-processing steps, we do not attempt to recognize them in this paper.

lexeme	cell	form	alignment
Qʔpʔ	non-actual	qʔpʔˈə t	0 1 2 3
Qʔpʔ	actual	qʔˈə pʔt	0 2 1 3
ʃtʃ	non-actual	ʃ tʃ ˈə t	0 1 2 3
ʃtʃ	actual	ʃ ˈə tʃ t	0 2 1 3

Table 6: Metathesis in Saanich Qʔpʔ 'patch' and ʃtʃ, 'whip'.

Finally, the ordering assumption of alignments is particularly challenged by phenomena where position is involved in exponence. The example (1) illustrates parallel exponence in Swahili (Stump 1993, via Crysmann and Bonami 2017). Aligning the sequence */-penda/* to find the stem is straightforward, but aligning */-ni-/*, */-ta-/* and */-wa-/* requires a more complex mapping:

(1) a. ni-ta-wa-penda
1SG-FUT-3PL-*like*
'I will like them'

b. wa-ta-ni-penda
3PL-FUT-1SG-*like*
'They will like me.'

4 Algorithms

This section presents two methods for the alignment of inflectional paradigms, designed specifically for the purpose of aligning the inflected forms of a paradigm in a given language. The first adapts a heuristic algorithm for multiple alignment from the biological literature. The second proceeds by searching for the Longest Common Subsequence. Both proceed in two steps: first, they generate sets of competing hypotheses, then, they pick the best alignments based on comparisons across lexemes.

4.1 Multiple alignments

Our first system relies on progressive alignments.[3] It differs from existing multiple alignment software and algorithms in three main ways: the scoring schemes are tailored for inflection; it finds a set of best alignments rather than a single best; the implementation is multi-threaded in order to process large inflected lexicons in reasonable time.

4.1.1 Scoring models

We present three scoring models to assign scores to edit operations. We write $s(a, b)$ the substitution of distinct phonemes and $s(a, a)$ the substitution of identical phonemes. Both insertion and deletion are given the same constant score $s(a, -) = s(-, a) = \gamma$. Our schemes rely on *similarity* (not distances), and scores are maximized by the algorithms. Affixes are common in inflection, and their presence in some, but not all paradigm forms, leads to frequent insertions or deletions in aligned paradigms. For this reason, we do not penalize insertion/deletions, and always set γ to zero. Each scoring model associates positive scores to good matches, and negative scores to bad matches.

Simple model: In the absence of any knowledge on a language's phonology, we can only favor identical matches, and penalize all substitutions: we set $s(a, b) = -1$ and $s(a, a) = 1$.

Phonological similarity models: In order to recognize non-affixal morphological alternations, we need a scoring function which is sensitive to phonological similarity. We define phonological similarity as the Jaccard index over either sets of features (sim_{nc}) or sets of natural classes (sim_{feat}), following Frisch (1997); Albright and Hayes (2006). The feature sets are read from a table of language specific distinctive features, provided in input.[4] Natural classes are computed from these by using Formal Concept Analysis (Ganter and Wille, 1998; Bank, 2016). We define a similarity threshold t as the median of all possible similarities. We then set $\gamma = 0$, $s(a, b) = \text{sim}(a, b) - t$ and $s(a, a) = \text{sim}(a, a) - t$.

[3] The code for the progressive method and the evaluation can be found at https://gitlab.com/sbeniamine/morphalign/-/tree/v0.1.1-scil

[4] While it is possible to conceive universal feature systems, these can not capture language specific phonology. They could however be used as fallback, and are useful as a basis for writing language specific distinctive features (several of the tables used here derive from Hayes (2012), see A).

4.1.2 Progressive alignments

Progressive alignments proceed in three steps, repeated for each paradigm:

First, we compute all pairwise alignments for unique sequences in the paradigm, using the Needleman-Wunsch algorithm. Second, we compute a similarity tree between sequences using hierarchical clustering and the scores from the pairwise alignments. Third, we align sequences and alignments pairwise by following the tree bottom-up. The simplest case is that of aligning a pair of sequences. However, since alignments can be seen as sequences of columns, we can also align sequences to alignments, or alignments together (Durbin, 1998; List, 2014). The score of aligning the columns $M_{\cdot,j}$ and $M'_{\cdot,l}$ from two alignments matrices M and M' with resp. m and m' rows is (Durbin, 1998, p.147): $S(M_{\cdot,j}, M'_{\cdot,l}) = \sum_{i=0}^{m} \sum_{k=0}^{m'} S(M_{ij}, M'_{kl})$.

Usually, this process produces a single best alignment at each step. Instead, we backtrack through multiple paths in the alignment matrix, and retrieve the k optimal scoring alignments which differ by at least one substitution. As a result, each pairwise step yields a set of alignments. When joining two sets A and B, we compute an alignment for each (a, b) from $A \times B$ and keep only k of the alignments with the same best score.

4.2 Longest common subsequence

The second method is based on the fact that our definition of CONSTANT STEM corresponds to the longest common subsequence (LCS).[5] There are multiple algorithms (Hirschberg, 1977) to find the LCS of two sequences, and it is known that the complexity of the problem increases with the size of the sequences and the size of the vocabulary (Ullman et al., 1976). The problem becomes even more complex when we have k sequences instead of just 2. While approximations exist for the $k = 2$ case (Hajiaghayi et al., 2020), here we use a simple combinatorial approach which works well when k is large and the length of the sequences is small (< 100 segments). This implementation follows (Guzmán Naranjo and Becker, in press).

We solve this problem by generating all possible subsequences for all sequences, but to save time we do this sequentially. We start with all subsequences of size p, where p is the length of the shortest sequence. If one of the generated subsequences appears in all sets of subsequences, we are done. If no subsequence satisfies this condition, we calculate all subsequences of size $p - 1$ and repeat this process until we find a subsequence common to all sets of subsequences, or until we determine that there is no stem.

The process can be made fast by using two preprocessing steps. Since the LCS can only be comprised of segments which appear in all sequences, we first remove segments which do not appear in all sequences. After this first step, we find and remove a common prefix and suffix from all sequences. By taking these two first steps, the computation of the LCS is fast in the average case.

After finding the stem, we calculate all optimal alignments of each cell to the stem using Levenshtein distance. Segments which do not correspond to the stem are kept aligned to gaps. If there are multiple possible alignments of the cell to the stem or multiple possible stems, we calculate all the resulting hypotheses and disambiguate as explained in the next section.

4.3 Disambiguation

Both the progressive alignments and the LCS system generate a set of hypotheses for each paradigm. From each hypothesis, we compute the markers (see section 3.1). Then we select the alignment with the highest sum of marker frequency. For the progressive alignments, in case of a tie, we fall back on the frequency of the continuous sub-strings in the markers. This step introduces comparisons across lexemes, selecting the alignments which lead to the most general analysis of the overall system.

5 Evaluation

Most NLP tasks, whether supervised or non supervised, rely on evaluation by comparison to a gold standard. This is the case, in particular, of non supervised approaches to morphological segmentation. In the present task, comparison to a gold segmentation does not make sense, for two main reasons.

First, alignments under-determine segmentations, such that various segmentations could be generated based on the same alignments. The simple segmentation described in section 3.1 is not given as a final linguistic analysis. As a result, it is unclear how to compare alignments to a given

[5] The code to the R package for finding the LCS can be found at: https://gitlab.com/mguzmann89/paradigma.

segmentation. Second, and more importantly, our goal is not to reproduce known morphemic segmentations, but to find new analyses through a methodology which can be reproduced identically across systems and languages. The goal of this process is to obtain comparative units for both typological and evolutionary linguistics. In this perspective, even if it was possible, the comparison to known segmentations would not inform us on the success of our methods.

Evaluation against gold alignments would be possible using dedicated datasets for inflection, similar to existing benchmarks for historical linguistics (List and Prokić, 2014). Unfortunately, these still have to be created. Instead, we define two simple measures based on the simple segmentation described in section 3.1.

All other things being equal, we prefer alignments which recognize more constant stem material. We measure this by computing the **Stem length**: $\text{Sl} = \frac{\text{constant stem length}}{\text{length of the shortest sequence}}$, which we give as a percentage. This value is averaged over the entire lexicon.

However, maximizing stem length is not enough, as we saw in section 3.2: alignments must lead to sound linguistic generalizations. In order to measure this, we remove all stems from the paradigms, so that each lexeme is characterized by a set of markers. We then observe the number of **distinct marker sets**, and their sizes. Any unfortunate alignment will drive the number of sets up.

Both of these measures are only comparative: they can not tell us whether an alignment is perfect, but given several solutions, they can point us to the best one. They target the two main difficulties in inflectional alignments: finding a longest common subsequence, and aligning it in a way which leads to good morphological generalizations.

6 Data

We evaluate our systems on 13 inflectional systems from 12 languages, as summarized in Table 7. The development set is composed of the entire lexicon for French and of around 18% of lexemes for English, Modern Standard Arabic, Latin verbs and European Portuguese. All other paradigms were held out until evaluation. Because of overabundance and defectivity, the total number of forms is not always the product of the number of lexemes and paradigm cells. The detailed source of each lexicon is given in Appendix A. [6]

Language	POS	Cells	Lexemes		Total
			Dev	Eval	Forms
Zenzontepec-Chatino	V	4	-	392	1567
Yaitepec-Chatino	V	12	-	324	3916
Kasem	N	2	-	1909	3936
Latin	N	12	-	1038	12355
Latvian	N	14	-	3706	43359
English	V	8	1092	4972	48732
Mod. Std. Arabic	V	109	183	835	95440
Navajo	V	70	-	2153	122756
Eur. Portuguese	V	69	366	1630	137724
French	V	51	5249	-	266490
Hungarian	N	34	-	12729	410391
Russian	N	12	-	45183	555654
Latin	V	254	599	2749	752154

Table 7: Overview of the lexicons

7 Results and discussion

We ran and evaluated alignments for each lexicon and each method. Progressive alignments ran with each of the possible scoring schemes. One version of the progressive algorithm used a random selection in step 2 in order to evaluate the contribution of this step. Evaluation results are given in Table 8.

The LCS algorithm is designed to optimize the average stem length, and it should be no surprise that it always finds the longest constant stems, across all datasets. The progressive algorithm often agrees in finding the LCS, but not always. Qualitative analysis of the result shows that the progressive algorithm sometimes narrows down the set of hypotheses too soon, and due to its greedy nature, can not recover from early mistakes. Variations in scoring scheme all lead to quasi-identical stems lengths, with sim_{feat} most often leading to shorter stems than the other two schemes.

It should be noted, however, that the right stem is not simply the longest one, but also that which leads to the most general alignments, at the level of the lexicon. This generality is precisely what can be evaluated by measuring the number of marker sets.[7]

[6] Those of these lexicons which can be shared freely can be found, together with our dev/eval splits, at https://gitlab.com/sbeniamine/inflectionallexicons/-/tree/SCiL-2021

[7] We would like to stress that marker sets are not intended to be interpreted directly as inflection classes, as is obvious from the fact that all methods count more than two thousands sets in both Navajo and Russian. The sets differ for any variation in surface realization, and collapse together exponent types which are usually analyzed as orthogonal dimensions, such as the two stems of Navajo verbs (McDonough, 1999) or the stress patterns and affixes of Russian nouns (Brown, 1998).

		step 1.	LCS		Progressive							
		step 2.	markers		markers							
		scores			simple		sim_{feat}		sim_{nc}		random sim_{nc}	
			SL	MS	SL	MS	SL	MS	SL	MS	SL	MS
Language	POS	set										
Modern std. Arabic	V	dev	60.1	67	59.9	66	58.9	69	59.7	65	59.7	65
		eval	60.7	144	60.6	137	59.4	160	60.3	140	60.3	140
English	V	dev	98.3	47	98.3	45	98.3	45	98.3	45	98.3	46
		eval	98.1	118	98.1	113	98.1	113	98.1	113	98.1	114
French	V	dev	97.6	111	97.6	99	97.6	99	97.6	99	97.6	99
Hungarian	N	eval	96.8	279	96.8	302	96.3	281	96.8	281	96.8	294
Kasem	N	eval	78.7	326	78.7	318	78.7	318	78.7	318	78.7	331
Latin	N	eval	81.4	81	81.4	73	81.4	73	81.4	73	81.4	73
	V	dev	76.1	190	76.1	186	76.0	184	76.1	184	76.1	185
		eval	75.0	434	75.0	426	74.9	424	75.0	424	75.0	429
Latvian	N	eval	85.2	143	85.2	128	85.2	128	85.2	128	85.2	128
Navajo	V	eval	40.0	2041	38.9	2044	36.4	2040	37.1	2041	37.1	2041
European Portuguese	V	dev	76.6	34	76.0	31	76.5	31	76.5	31	76.5	31
		eval	76.2	59	75.3	61	76.0	55	75.9	57	75.9	57
Russian	N	eval	85.1	2230	85.1	2116	85.1	2115	85.1	2115	85.1	2118
Yaitepec Chatino	V	eval	38.3	294	38.3	294	38.2	294	38.3	294	38.3	294
Zenzontepec Chatino	V	eval	72.4	109	72.4	111	72.2	112	72.4	111	72.4	112

Table 8: Average stem length percentage (SL) and number of market sets (MS) for each algorithm and dataset. Best values are given in light gray for SL and dark gray for MS.

Despite its lesser performances in finding the LCS, the progressive method leads to less marker sets in all but two datasets. A qualitative analysis of the results confirms that this is due to the quality of the alignments themselves. The LCS method usually fails not at identifying the correct stem, but rather at picking the optimal alignment of the stem to the individual forms. This leads to frequent mistakes which impact overall regularity.

The comparison between marker disambiguation and a random choice shows without doubt that cross-lexeme comparisons are useful. The similarity schemes score overall better than the simple scheme, with one exception for Modern Standard Arabic verbs. The cause is again that the similarity schemes narrow the set of hypotheses too early, and miss some alignments which are most general across lexemes. Neither of the similarity schemes seems much superior to the other.

In two cases, the progressive method is outperformed by the LCS method. In Zenzontepec Chatino, this is due to lexemes such as TU^{2}KWA^1, which forms are *nku^{0}tu^{2}kwa^1* (CPL), *tyu^{0}kwa^0* (POT), *ntyu ^{0}kwa^0* (HAB) and *nte^{0}tu^{2}kwa^1* (PROG). Two LCS can be found, either /u-kwa/, or /^{0}-kwa/. Only the first one makes sense in the context of other lexemes, but the second one is preferred by all variants of the progressive alignments, as it allows for better partial matches. In this case, the LCS method finds the correct stem because it generates hypotheses for all competing LCS. A better strategy overall would be to consider separate tiers for tones and segments. In Hungarian, the median of all similarity scores, used as a threshold to calculate the score matrix, is very low. As a result, the system prefers most substitutions to insertions, and fails to find the best alignments. This case calls into question the use of the median of all scores as a threshold.

As discussed in Section 3.1, the simple units we extract for the purposes of evaluation do not capture suppletive stems. For example, the French verb ALLER, 'to go' is suppletive. Table 9 shows a few suppletive forms. Because the suppletive stems share nothing in common, the constant stem is empty and entire words are pushed into the markers. Markers are parts of words with inflectional information, and indeed each suppletive stem is informative: for example, the forms of ALLER which start in /v-/ are never futures. However, the alignments themselves provide enough information to extract more detailed generalizations. For example,

PRS.3SG	-	v	-	-	-	a
PRS.2PL	a	l	-	-	-	E
PRS.3PL	-	v	-	-	-	ɔ̃
FUT.2SG	i	ʁ	-	-	-	a
FUT.3SG	i	ʁ	-	-	-	a
PST.SBJV.1PL	a	l	a	s	j	ɔ̃
indexes	0	1	2	3	4	5

Table 9: Progressive alignment (sim_{nc}) for a few forms of the French verb ALLER, 'to go'.

in Table 9, the observation of identical matches in columns 0 and 1 can support the identification of the suppletive stems, and their respective stem spaces.

In Standard Modern Arabic, the LCS method scores better for stem length, but worse for marker sets compared to the best progressive method (sim_{nc}). A qualitative analysis of the results shows that the LCS method more reliably finds the longest constant stems, as the progressive method tends to be derailed by the repetitions of a sound in the stem (this difficulty was illustrated in Table 4). For example, in forms of the verb 'AMADDA ('to grant a delay, to assist'), the LCS method successfully identifies a stem /m-d-d/, but the progressive method sometimes misaligns one /d/ with the other, resulting in a shorter stem /m-d/. On the other hand, the progressive method chose better generalization among competing stems of identical length. For example, for the forms of the verb BA'ISA 'to be sad', there are two possible LCS, /b-ʔ-s/ or /b-a-s/, as the indicative past forms all start in /baʔis-/, while all other forms start with a consonant, followed by /-abʔas-/. The LCS method finds /b-a-s/, which results in a unique marker set. The progressive method correctly finds /b-ʔ-s/, resulting in a marker set common to four other verbs (FARIḤA 'to be happy', 'ARIQA 'to sweat', ḠALIBA 'to be thick necked', and ḠALIMA 'to be sexually aroused'). For all of these verbs, the LCS method finds a stem in /C-a-C/ rather than the correct /C-C-C/.

Overall, the performances of both methods are close, and we believe that both constitute reasonable, though imperfect solutions. Since they stumble over different difficulties, combining them seems promising. For example, a system could compute all possible LCS, then use them to customize the scoring matrix for each paradigm, before using progressive alignments.

8 Conclusion

Multiple alignments are a crucial task for quantitative morphology, as they constitute the first step in extracting analytical units from inflectional lexicons. Alignments have important impact on inflectional analyses, therefore they should be carefully designed and evaluated. Only then can they constitute a first step in computational morphology, from which to induce comparable inflectional analyses for typological and evolutionary quantitative work.

The intent of this paper is to bring attention to this new task, and to provide the conditions for incremental improvements. Towards this goal, we described data formats and evaluation measures, and compared two implemented systems. Since the evaluations we propose are comparative, these systems can constitute strong baselines to which future systems may be compared.

Overall, we found that there are two main difficulties in multiple alignments of inflectional paradigms: first, finding the right longest common subsequence in a given paradigm, and second, aligning it in a linguistically motivated way. Both difficulties require the alignment to be optimized both at the level of a single paradigm, and across lexemes at the level of the lexicon.

Further research could focus on further improving scoring schemes and on leaving more ambiguity until the disambiguation step. Variants such as many-to-many alignments (Jiampojamarn et al., 2007) could also be useful in identifying some inflectional phenomena such as gemination or reduplication. Moreover, supra-segmental material, such as tones, may need to be aligned independently, and a tiered approach to inflectional alignments should be considered. The manual elaboration of gold alignment sets would be beneficial for the evaluation of new alignment methods. Finally, more work is necessary in order to extract useful linguistic units from alignments, whether in the form of segmentations, processes or analogies.

Acknowledgments

This work was partially funded by a British Academy Newton International Fellowship. We thank Olivier Bonami for his helpful comments and input, the participants of the workshop "How to fill a cell: Computational approaches to inflectional morphology" (online, 09/16/2020) and the CONLL 2020 and SCiL 2021 reviewers, whose comments were instrumental in improving our paper.

References

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

Adam Albright and Bruce Hayes. 1999. An automated learner for phonology and morphology. Unpublished manuscript.

Adam Albright and Bruce Hayes. 2002. Modeling english past tense intuitions with minimal generalization. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning - Volume 6*, pages 58–69, Stroudsburg, PA, USA. Association for Computational Linguistics.

Adam Albright and Bruce Hayes. 2006. Modeling productivity with the gradual learning algorithm: the problem of accidentally exceptionless generalizations. In Gisbert Fanselow, Caroline Féry, Matthias Schlesewsky, and Ralf Vogel, editors, *Gradience in Grammar: Generative Perspectives*, pages 185–204. Oxford University Press, Oxford.

Adam C. Albright and Bruce P. Hayes. 2003. Rules vs. analogy in english past tenses: A computational/experimental study. *Cognition*, 90:119–161.

R Baayen, R Piepenbrock, and L Gulikers. 1995. Celex2 ldc96l14.

Sebastian Bank. 2016. Concepts: Formal concept analysis with python. V0.7.10.

Geoffrey J. Barton and Michael J.E. Sternberg. 1987. A strategy for the rapid multiple alignment of protein sequences: Confidence levels from tertiary structure comparisons. *Journal of Molecular Biology*, 198(2):327 – 337.

Sacha Beniamine. 2017. Un algorithme universel pour l'abstraction automatique d'alternances morphophonologiques. In *24e Conférence sur le Traitement Automatique des Langues Naturelles (TALN)*, volume 2, pages 77–85, Orléans, France.

Sacha Beniamine. 2018. *Classifications flexionnelles: Étude quantitative des structures de paradigmes*. Ph.D. thesis, Université Sorbonne Paris Cité - Université Paris Diderot.

Sacha Beniamine, Olivier Bonami, and Benoît Sagot. 2017a. Inferring inflection classes with description length. *Journal of Language Modelling*, 5(3):465–525.

Sacha Beniamine, Joyce McDonough, and Olivier Bonami. 2017b. When segmentation helps: Implicative structure and morph boundaries in the navajo verb. In *ISMO*, pages 11–15, Lille, France.

Olivier Bonami and Gilles Boyé. 2014. De formes en thèmes. In Florence Villoing, Sarah Leroy, and Sophie David, editors, *Foisonnements morphologiques. Etudes en hommage à Françoise Kerleroux*, pages 17–45. Presses Universitaires de Paris Ouest.

Olivier Bonami, Gauthier Caron, and Clément Plancq. 2014. Construction d'un lexique flexionnel phonétisé libre du français. In *Actes du quatrième Congrès Mondial de Linguistique Française*, pages 2583–2596.

Dunstan Brown. 1998. *From the general to the exceptional*. Ph.D. thesis, University of Surrey.

Noam Chomsky and Morris Halle. 1968. *The sound pattern of English*. Harper and Row.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. 2017. CoNLL-SIGMORPHON 2017 shared task: Universal morphological reinflection in 52 languages. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 1–30, Vancouver. 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 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 10–22. Association for Computational Linguistics.

Michael A. Covington. 1996. An algorithm to align words for historical comparison. *Comput. Linguist.*, 22(4):481–496.

Mathias Creutz and Krista Lagus. 2002. Unsupervised discovery of morphemes. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*, pages 21–30. Association for Computational Linguistics.

Berthold Crysmann and Olivier Bonami. 2017. Atomistic and holistic exponence in underspecified realisational morphology. In *Proceedings of the 24th International Conference on Head-Driven Phrase Structure Grammar*, pages 141–161, Stanford. CSLI Publications.

François Dell. 1973. *Les règles et les sons: Introduction à la phonologie générative*. Collection Savoir. Hermann, Paris.

Richard Durbin. 1998. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press.

Greg Durrett and John DeNero. 2013. Supervised learning of complete morphological paradigms. In *Proceedings of the 2013 Conference of the North*

American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1185–1195. Association for Computational Linguistics.

Timothy Feist and Enrique L. Palancar. 2015. Otomanguean inflectional class database. University of Surrey.

Da-Fei Feng and Russell F. Doolittle. 1987. Progressive sequence alignment as a prerequisitetto correct phylogenetic trees. *Journal of Molecular Evolution*, 25(4):351–360.

Stefan Frisch. 1997. *Similarity and frequency in phonology*. Ph.D. thesis, Northwestern University.

Bernhard Ganter and Rudolf Wille. 1998. *Formal concept analysis: Mathematical foundations*. Springer.

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

Matías Guzmán Naranjo. 2019. *Analogical Classification in Formal Grammar*. Empirically Oriented Theoretical Morphology and Syntax. Language Science Press.

Matías Guzmán Naranjo. 2020. Analogy, complexity and predictability in the russian nominal inflection system. *Morphology*, 30.

Matías Guzmán Naranjo and Laura Becker. in press. Coding efficiency in nominal inflection: Expectedness and type frequency effects. *Linguistics Vanguard*.

MohammadTaghi Hajiaghayi, Masoud Seddighin, Saeed Seddighin, and Xiaorui Sun. 2020. Approximating lcs in linear time: Beating the $\sqrt{n}$ barrier. *arXiv preprint arXiv:2003.07285*.

M. Halle and George N. Clements. 1983. *Problem Book in Phonology: A Workbook for Introductory Courses in Linguistics and in Modern Phonology*. Bradford Books. MIT Press.

Bruce Hayes. 2012. Spreadsheet with segments and their feature values. Distributed as part of course material for Linguistics 120A: Phonology I at UCLA.

Daniel S Hirschberg. 1977. Algorithms for the longest common subsequence problem. *Journal of the ACM (JACM)*, 24(4):664–675.

Charles F. Hockett. 1954. Two models of grammatical description. *Word*, 10:210–234.

Sittichai Jiampojamarn, Grzegorz Kondrak, and Tarek Sherif. 2007. Applying many-to-many alignments and hidden Markov models to letter-to-phoneme conversion. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 372–379, Rochester, New York. Association for Computational Linguistics.

Christo Kirov, John Sylak-Glassman, Roger Que, and David Yarowsky. 2016. Very-large scale parsing and normalization of wiktionary morphological paradigms. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

Grzegorz Kondrak. 2000. A new algorithm for the alignment of phonetic sequences. In *1st Meeting of the North American Chapter of the Association for Computational Linguistics*.

Grzegorz Kondrak. 2003. Phonetic alignment and similarity. *Computers and the Humanities*, 37(3):273–291.

Mikko Kurimo, Sami Virpioja, Ville Turunen, and Krista Lagus. 2010. Morpho challenge 2005-2010: Evaluations and results. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, pages 87–95, Uppsala, Sweden. Association for Computational Linguistics.

kazutaka Kurisu. 2001. *The Phonology of Morpheme Realization*. Ph.D. thesis, University of California Santa Cruz.

Johann-Mattis List. 2010. Phonetic alignment based on sound classes. In *Proceedings of the 15th Student Session of the European Summer School for Logic, Language and Information*, pages 192–202, Kopenhagen.

Johann-Mattis List. 2014. *Sequence comparison in historical linguistics*. Düsseldorf University Press, Düsseldorf.

Johann-Mattis List and Jelena Prokić. 2014. A benchmark database of phonetic alignments in historical linguistics and dialectology. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages 288–294. European Language Resources Association (ELRA).

Joyce McDonough. 1999. On a bipartite model of the athabaskan verb. In T. B. Fernald and P. R. Platero, editors, *The Athabaskan Languages: Perspectives on a Native American Language Family*, Oxford Studies in Anthropological Linguistics, pages 139–166. Oxford University Press.

Saul B. Needleman and Christian D. Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443 – 453.

Garrett Nicolai, Colin Cherry, and Grzegorz Kondrak. 2015. Inflection generation as discriminative string transduction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 922–931. Association for Computational Linguistics.

Idda Niggli and Urs Niggli. 2007. *Dictionaire Bilingue Kasɩ m - Français Français - Kassem*. Société Internationale de Linguistique.

Matteo Pellegrini and Marco Passarotti. 2018. Latinflexi: an inflected lexicon of latin verbs. In *Proceedings of the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018)*, volume 2253 of *CEUR Workshop Proceedings*, page December, Aachen.

Andrew Spencer. 2012. Identifying stems. *Word Structure*, 5(1):88–108.

Gregory T. Stump. 1993. Position classes and morphological theory. In Geert Booij and Jaap van Marle, editors, *Yearbook of Morphology 1992*, pages 129–180. Springer Netherlands, Dordrecht.

JD Ullman, AV Aho, and DS Hirschberg. 1976. Bounds on the complexity of the longest common subsequence problem. *Journal of the ACM (JACM)*, 23(1):1–12.

Arlindo Veiga, Sara Candeias, and Fernando Perdigão. 2013. Generating a pronunciation dictionary for european portuguese using a joint-sequence model with embedded stress assignment. *Journal of the Brazilian Computer Society*, 19(2):127–134.

Robert A. Wagner and Michael J. Fischer. 1974. The string-to-string correction problem. *J. ACM*, 21(1):168–173.

Robert W. Young and William Morgan. 1987. *The Navajo Language: A Grammar and Colloquial Dictionary*, revised edition edition. University of New Mexico Press, Albuquerque.

Andrey Anatolyevich Zaliznyak. 1977. *Grammatical Dictionary of the Russian Language*. Russkij Jazyk.

A Appendix: Sources of the lexicons

The Kasem lexicon was compiled by Guzmán Naranjo (2019) from the Kasem dictionary by Niggli and Niggli (2007). The two Latin lexicons come from LatinFlexi (Pellegrini and Passarotti, 2018), the distinctive features are adapted from a table provided by the authors. The Latvian lexicon is adapted from Wiktionary with manual corrections. The English lexicon is from the CELEX database (Baayen et al., 1995), with distinctive features adapted from Chomsky and Halle (1968); Halle and Clements (1983). The French lexicon is based on Flexique (Bonami et al., 2014), with distinctive features based on Dell (1973). The Hungarian data was compiled from Wiktionary with added manual corrections. The Russian lexicon was compiled by Guzmán Naranjo (2020) from the Zaliznyak Russian dictionary (Zaliznyak, 1977). The Chatino lexicons come from the Oto-Manguean Inflectional Class Database (Feist and Palancar, 2015). The European Portuguese lexicon is derived from Veiga et al. (2013). The Modern Standard Arabic lexicon is derived from Unimorph (Kirov et al., 2016). The Navajo lexicon was compiled and phonemised from Young and Morgan (1987), and based on data used by Beniamine et al. (2017b). The phonemisation for Modern Standard Arabic and Chatino, and the phonological features for Portuguese are derived from Beniamine (2018). For Chatino, Modern Standard Arabic and Navajo, the distinctive features are adapted from Hayes (2012).