A holistic approach at a morphological inflection task

Rashel Fam **Yves Lepage**

Waseda University
{fam.rashel@fuji., yves.lepage@}waseda.jp

Abstract

Morphological inflection is the task of generating previously unseen words from morphological features. A common approach, the morpheme-based approach, decomposes words into smaller units, such as morphemes or affixes, learned in advance. This paper proposes a different approach. It shows that breaking words into pieces is not necessarily the best option. The proposed approach is holistic and treats whole word forms as basic units in its description of morphological variations among word forms. The approach generates inflected forms by solving analogical equations between whole word forms; morphological features can be used as constraints. Experiment results on the 52 languages of SIGMORPHON 2017 Shared Task show that the proposed approach performs as good as the morpheme-based approach, even slightly better on average. This demonstrates the absence of necessity of explicitly learning how to decompose words.

1. Introduction

Many natural language processing (NLP) tasks, like machine translation, require to analyze and generate morphological word forms, even previously unseen ones. Automatically learning morphological features of word forms is a challenging task, especially for languages with rich morphology. For such languages, data sparsity is a problem which leads to poor generalizations in machine learning (Dreyer and Eisner, 2011). The organizers of the SIGMORPHON[1] Shared Task mentioned that a Polish verb may have almost 100 different word forms (Janecki, 2000).

Existing research can be grouped into three categories. The first one is the hand-engineered rule-based approach. It offers a very high accuracy but needs laborious work to construct. It usually faces the word coverage problem.

The second one is the supervised approach. It automatically induces morphological rules from a given training data set and generates word forms by applying the best rules chosen by some classification methods (Ahlberg et al., 2015). It is practically language independent. It faces the generalization problem.

The third one is the neural network approach. The latest evaluation campaign on morphological reinflection, SIGMORPHON 2016 Shared Task (Cotterell et al., 2016), showed that recurrent neural network (RNN) encoder-decoder models (Kann and Schütze, 2016) give the best results with more than 90 % accuracy. Such models are adaptations of models used in machine translation. However, they suffer from difficulties in designing adequate architectures. Very long training times are also a drawback.

This paper contrasts the use of two different approaches, morpheme-based and holistic, on a morphological inflection task. We consider whole words as basic units of processing instead of breaking them into smaller units identified in advance. Inflected forms are generated by solving analogical equations between word forms. As an illustration, the present participle *taking* of the English lemma *take* can be generated as follows.

$$\textit{release} : \textit{releasing} :: \textit{take} : x \quad \Rightarrow \quad x = \textit{taking}$$

[1] `http://www.sigmorphon.org/`

	Number of word forms		
	Training	Development	Test
low	100	1,000	1,000
medium	1,000	1,000	1,000
high	10,000	1,000	1,000

Table 1: Size of training, development and test set in number of word forms

2. Inflection Task

Word inflection is the task of generating correctly an inflected form (the target form) of a given lemma for some target morphological features. The target features are a sequence of morphological features of the inflected form. This sequence may vary from a single part-of-speech (POS) tag to a rich set of POS, tense, mood, aspect, gender, number, case feature values, etc. which depend on the language. For instance, Arabic verbs have a *gender* feature while English verbs do not.

Below is an example of a word inflection task question for an English verb: produce the present participle form for the lemma *take*. The correct inflected form (target form) is *taking*.

Given lemma: *take*
Target features: V;V.PTCP;PRS
Target form: ***taking***

2.1. Languages and data used

We used the data for morphological reinflection provided during the evaluation campaign SIGMORPHON 2017 in the Shared Task[2]. It contains 52 different languages with different size of training data: low, medium, and high. The size of the training and test sets used in each experiment protocol is presented in Table 1.

These languages are diverse not only by region but also language families. They cover around 20 different language families with various morphological features.

2.2. Evaluation

Two evaluation metrics are used to measure the performance of systems. Accuracy is the ratio of the number

[2] `https://github.com/sigmorphon/conll2017`

Training data

lemma	inflected	morph. features
show	showed	V;PST
show	showing	V;V.PTCP;PRS
release	released	V;PST
release	releasing	V;V.PTCP;PRS
schmear	schmeared	V;PST
⋮	⋮	⋮

Question

Lemma: *take*
Target features: V;V.PTCP;PRS

LEMMA	:	*V;V.PTCP;PRS*	:	*V;PST*
release	:	*releasing*	:	*released*
illustrate	:	*illustrating*	:	*illustrated*
age	:	*aging*	:	*aged*
bake	:	*baking*	:	

release : *releasing* :: *take* : *x*
⇒
$x =$ ***taking***

show	:	*showing*	:	*showed*
enter	:		:	*entered*
reason	:	*reasoning*	:	*reasoned*
schmear	:		:	*schmear*

show : *showing* :: *take* : *x*
⇒
$x =$ ~~*takeing*~~

Figure 1: How to generate target form (present participle) of the given lemma *take* using computational analogy. Different analogical grids may generate different word forms.

of correctly predicted target forms by the total number of questions (see Formula (1)).[3] The range is [0, 100] and the higher the score, the better.

$$\text{Accuracy} = \frac{\sum_{i=1}^{N} \delta(\text{predicted}_i = \text{correct}_i)}{N} \times 100 \quad (1)$$

The other evaluation metric is the average Levenshtein distance between the predicted target forms and the correct target forms (see Formula (2)). In contrast with accuracy, the lower the scores, the better: lower scores mean that the predicted target forms are closer to the correct target forms; a score of 0 means that all predicted target forms were correct.

$$\text{Avg. Levenshtein dist.} = \frac{\sum_{i=1}^{N} lev(\text{predicted}_i, \text{correct}_i)}{N} \quad (2)$$

In the cases where multiple correct target forms exist, like *hung* and *hanged* for the past form of the English lemma *hang*, any of the possible correct target forms is accepted.

3. Baseline system: morpheme-based approach

We use the system provided by the organizer of SIGMORPHON 2017 Shared Task[2] as our baseline. The system uses a morpheme-based approach which first learn how to break words into stems and affixes and memorizes the affixes. The affixes are induced by aligning the lemma and the inflected forms using the Levenshtein distance. Prefixing and suffixing rules are extracted and grouped by their corresponding morphological features. This information is stored as explicit knowledge as a list of triplets. The ones below illustrate suffixes for English present participle (encoded in morphological features as *V;V.PTCP;PRS*). The rule: '-ε replaced with *-ing*' (true suffixing) occurred 1,121 times in the *high* training data. For the same target features, '*-e* replaced with *-ing*' occurred 832 times.

(*substring,*	*replacement,*	*#_of_occurrences*)
'-ε'	'-ing'	1,121
'-e'	'-ing'	832
'-ize'	'-izing'	162
⋮	⋮	⋮
'rank'	'ranking'	1
⋮	⋮	⋮

For the generation phase, the system performs inflection as follows. According to the target features, it first applies the longest suffixing rule with the highest number of occurrences, and then similarly for prefixing rules. This delivers one predicted target form.

4. Proposed method: holistic approach

Based on the fact that word forms are connected with other word forms in a systematic way, we see the morphological inflection task as the task of solving analogical equations. We generate target forms by solving analogical

[3]In Formulae (1) and (2), N is the total number of questions; $\delta(A = B) = 1$ if the two strings A and B are equal, else 0; $lev(A, B)$ is the value of the Levenshtein distance between strings A and B (Levenshtein, 1966; Wagner and Fischer, 1974).

equations built from the systematic evidence observed in the training data.

4.1. Proportional analogy

Derivation between words can be explained with the notion of analogy (Greek and Roman grammatical tradition up to (de Saussure, 1995) or (Hathout, 2008; Hathout, 2009) for recent work in computational morphology). Analogy is a systematic relationship usually noted $A : B :: C : D$. It states that A is to B as C is to D. Various formalisations of analogy have been proposed in (Yvon, 2003; Lepage, 2004; Stroppa and Yvon, 2005). In this work, we select the following definition [4].

$$A : B :: C : D \Rightarrow \begin{cases} d(A,B) &= d(C,D) \\ d(A,C) &= d(B,D) \\ |A|_a + |D|_a &= |B|_a + |C|_a, \forall_a \end{cases} \quad (3)$$

As an example, the following analogy explains the derivation of the word form *taking* mentioned in the Introduction. The four words in this analogy meet the definition in Formula (3).

release : *releasing* :: *take* : *taking*

4.2. Analogical grids

An analogical grid is a matrix of words where any four words from two rows and two columns necessarily constitute an analogy. Formula (4) gives the definition.

$$\begin{array}{l} P_1^1 : P_1^2 : \cdots : P_1^m \\ P_2^1 : P_2^2 : \cdots : P_2^m \\ \vdots \quad \vdots \qquad\quad \vdots \\ P_n^1 : P_n^2 : \cdots : P_n^m \end{array} \quad \stackrel{\Delta}{\Longleftrightarrow} \quad \begin{array}{c} \forall (i,k) \in \{1,\ldots,n\}^2, \\ \forall (j,l) \in \{1,\ldots,m\}^2, \\ P_i^j : P_i^l :: P_k^j : P_k^l \end{array} \quad (4)$$

Analogical grids automatically constructed from a corpus may contain empty cells. Such empty cells are interesting because they can be filled by potential word forms, supposedly unseen inflected forms. E.g., the two empty cells in the analogical grid below can be filled by the two unseen inflected forms *locating* and *baked*.

release	:	*releasing*	:	*released*
age	:	*aging*	:	*aged*
bake	:	*baking*	:	
locate	:		:	*located*

4.3. Holistic approach for inflection task

While the baseline system uses a morpheme-based approach, we adopt a holistic view and do not break words into pieces (Singh, 2000; Singh and Ford, 2000; Neuvel and Singh, 2001). According to the given target features, we first select the relevant analogical grid to generate the target form. If several candidate analogical equations are found, we use heuristic features (longest common subsequences, edit distance, longest matching suffix, longest matching prefix, etc) to select the analogical equation which will generate the (unique) predicted target form. In case several candidate analogical equations are still present (even after using heuristics), we solve all the analogical equations to generate all the possible predicted target forms. The most frequent one is chosen as the (unique) predicted target form.

Figure 1 illustrates how we generate the present participle form (encoded in morphological features as *V;V.PTCP;PRS*) from the given verb lemma *take*. Several candidate analogical equations with the same morphological features are found in the training data. They are grouped into two analogical grids (*top* and *bottom*).

From the first line of the *top* grid, the lemma *release* and the inflected form *releasing*, whose morphological features are the target features, will form an analogical equation with the given lemma. It will produce the target form *taking*, as illustrated in the right part of the *top* grid in Figure 1. The same procedure with the *bottom* grid will generate *takeing* as the inflected form. Heuristic features (longest matching suffix) will select the *top* grid, and discard the *bottom* one. Algorithm 1 sketches the above procedure.

Algorithm 1 Generating target form using analogy

```
function GENERATE_TARGET_FORM(train_data, given_lemma, target_features)
    target_forms ← {}                                  ▷ Empty dictionary
    candidates ← filter train_data by matching target_features)
    candidates ← sort candidates by heuristic scores

                                                       ▷ Generation step
    for all candidate in the candidates do
        (lemma, form) ← candidate
        target_form ← x/ lemma : form :: given_lemma : x
        target_forms[target_form] += 1
    end for

                                                       ▷ Selection step
    if target_forms ≠ {} then
        sort target_forms (decreasing order)
        return pop(target_forms)  ▷ Most frequent form
    else
        return given_lemma
    end if
end function
```

5. Experiments

We carried out experiments on all of the 52 languages with three varying sizes of training data. The performance of the systems are evaluated on two data sets: development set and test set. Following the condition of the SIGMORPHON 2017 Shared Task, we do not use the development set as additional knowledge when performing experiments on test set. This is to our disadvantage.

[4] $d(A,B)$ stands for the value of the LCS edit distance between the two strings A and B. $|A|_c$ notes the number of occurrences of character c in string A.

6. Results

Table 2 shows the average accuracy of the baseline (morpheme-based approach), the proposed method (holistic approach), and the oracle. Table 3 shows the average Levenshtein distance scores. For detailed results on each language with different sizes of training data, see Table 4.

	Training data size					
	low		medium		high	
	dev	test	dev	test	dev	test
Morpheme-based	37.6	38.0	64.1	64.7	77.2	77.8
Holistic	38.2	38.0	65.9	65.7	78.1	78.3
Oracle	41.5	-	77.2	-	91.6	-

Table 2: Average accuracy scores on development set (*dev*) and test set (*test*) for 52 languages for the baseline (morpheme-based), the proposed method (holistic), and the oracle.

On average on the 52 languages, for all training data sizes: *low*, *medium*, and *high*, our proposed method based on a holistic approach performs as well as the baseline; even slightly better on accuracy scores,

6.1. Oracle experiments

To determine the upper-bound performance of our method on this task, we carried out oracle experiments. We stop at the end of the generation part and simply count a success each time a correct target form can be found among the candidate outputs (*target_forms* in Algorithm 1).

The gap between our holistic approach and the morpheme-based baseline shows that the heuristics to rank the candidates perform reasonably. The goal of this paper was to contrast the morpheme-based approach to the holistic approach. In this respect, the results we obtained clearly show that breaking into learned and memorized morphemes or affixes is not necessarily the best option to describe morphological variations of word forms.

The gap between our holistic approach and the oracle experiments shows that the heuristics to rank the candidates can be largely improved by designing a better selection method.

6.2. Possible improvements

One of the main obstacle in the task is when some of the given morphological features for the target word form are missing from the training data. In this case, both the baseline and the proposed method simply return the lemma as the target word form. One may think of learning the contribution of each morphological feature in the given morphological features list. This may introduce flexibility to generate inflected forms from unseen features list.

	Training data size					
	low		medium		high	
	dev	test	dev	test	dev	test
Morpheme-based	2.3	2.2	1.0	0.9	0.6	0.5
Holistic	2.4	2.3	1.0	1.0	0.6	0.6
Oracle	-	-	-	-	-	-

Table 3: Same as Table 2 but for average Levenshtein distance score.

To solve this problem, an approach worth to consider would be to use formal concept analysis (Ganter and Wille, 1999). For instance, (Kuroda, 2016) shows how to automatically acquire inflectional classes in Czech declensional paradigms by using formal concept analysis. The structure of how one word form is related to other word forms is similar to our analogical grids.

As over-generation is a known issue with analogy, one can also think of a way to filter out linguistically incorrect target forms using some word model (n-gram model).

7. Conclusion

We showed how a holistic approach can be used to perform a morphological inflection task, namely SIGMORPHON 2017 Shared Task. We showed how to generate inflected forms without explicitly decomposing words into morphemes, roots, stems or affixes, by first structuring the word forms in the training data into analogical grids and then by solving analogical equations between word forms using the given morphological features as constraint.

We performed all possible experiments with the dataset of SIGMORPHON 2017 Shared Task i.e., in all the 52 available languages for all the sizes of training data: low, medium and high. The results show that our method performs as good as the baseline provided by the organizers, even slightly better on average. We conclude that treating the whole word as a basic unit gives equivalent performance as an approach where words are segmented into learned and stored pieces, like morphemes or affixes.

8. Acknowledgement

This work was supported by a JSPS Grant, Number 15K00317 (Kakenhi C), entitled Language productivity: efficient extraction of productive analogical clusters and their evaluation using statistical machine translation.

9. References

Ahlberg, Malin, Markus Forsberg, and Mans Hulden, 2015. Paradigm classification in supervised learning of morphology. In *NAACL 2015*.

Cotterell, Ryan, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden, 2016. The sigmorphon 2016 shared task—morphological reinflection. In *SIGMORPHON 2016*.

de Saussure, Ferdinand, 1995. *Cours de linguistique générale*. Lausanne et Paris: Payot, 1st edition.

Dreyer, Markus and Jason Eisner, 2011. Discovering morphological paradigms from plain text using a dirichlet process mixture model. In *EMNLP 2011*.

Ganter, Bernhard and Rudolf Wille, 1999. *Formal Concept Analysis: Mathematical Foundations*. Springer-Verlag.

Hathout, Nabil, 2008. Acquistion of the morphological structure of the lexicon based on lexical similarity and formal analogy. In *3rd Textgraphs workshop on Graph-based Algorithms for Natural Language Processing*.

Hathout, Nabil, 2009. Acquisition of morphological families and derivational series from a machine readable dictionary. *CoRR*, abs/0905.1609.

Language	Accuracy									Average Levenshtein distance					
	low			medium			high			low		medium		high	
	B	H	O	B	H	O	B	H	O	B	H	B	H	B	H
Albanian	21.6	22.5	22.0	66.1	73.6	78.6	78.1	88.3	94.8	4.4	4.8	1.2	1.4	0.6	0.4
Arabic	21.5	24.5	24.0	40.0	49.6	60.9	47.7	60.0	83.4	3.1	3.4	1.8	2.1	1.5	1.6
Armenian	37.8	37.3	38.8	76.6	81.5	89.7	89.1	90.4	97.9	2.2	2.4	0.4	0.5	0.2	0.3
Basque	0.0	0.0	0.0	2.0	3.0	4.0	6.0	8.0	15.0	6.5	6.5	5.1	5.7	3.3	4.6
Bengali	44.0	43.0	44.0	75.0	74.0	88.0	84.0	81.0	98.0	1.5	1.7	0.4	0.6	0.3	0.4
Bulgarian	33.1	31.6	31.9	75.0	76.8	84.2	90.0	88.1	97.7	1.6	1.9	0.5	0.6	0.2	0.3
Catalan	55.2	52.8	53.9	83.2	82.7	84.1	94.2	94.5	96.5	1.1	1.3	0.3	0.4	0.1	0.2
Czech	40.8	38.5	37.6	80.7	81.9	88.5	90.4	90.0	96.4	1.9	2.0	0.4	0.5	0.2	0.2
Danish	59.8	67.6	80.8	78.1	79.3	97.1	89.1	88.4	99.3	0.7	0.4	0.3	0.3	0.2	0.2
Dutch	53.7	54.9	56.1	71.7	71.6	86.7	86.8	86.8	97.2	0.7	0.8	0.4	0.5	0.2	0.2
English	76.2	81.3	89.3	90.2	90.4	96.9	95.0	94.8	99.3	0.4	0.3	0.2	0.2	0.1	0.1
Estonian	22.6	21.7	24.3	62.4	60.0	75.3	76.2	76.8	90.6	2.9	3.3	0.8	1.0	0.5	0.5
Faroese	30.7	35.1	44.7	58.7	61.9	81.4	74.7	74.8	92.3	1.6	1.5	0.9	0.8	0.6	0.6
Finnish	16.2	14.4	14.9	42.5	41.1	49.9	78.5	77.6	88.3	4.2	4.8	1.4	2.3	0.4	0.5
French	63.0	61.7	66.4	76.1	75.8	87.3	83.6	83.2	96.0	0.8	1.1	0.5	0.5	0.3	0.4
Georgian	71.2	71.6	75.0	90.0	89.4	92.4	94.0	94.4	97.2	0.6	0.7	0.2	0.3	0.1	0.1
German	53.7	55.2	65.9	71.5	74.4	89.9	81.2	82.6	95.2	1.1	1.0	0.8	0.7	0.6	0.6
Haida	34.0	24.0	24.0	56.0	64.0	75.0	69.0	61.0	88.0	6.0	6.8	1.2	1.8	0.6	1.5
Hebrew	27.9	30.1	31.9	40.0	48.6	70.2	55.8	60.8	91.9	1.3	1.4	0.9	0.8	0.6	0.6
Hindi	31.0	27.7	28.5	86.6	84.2	90.4	94.0	93.2	100.0	3.8	4.1	0.2	0.5	0.1	0.1
Hungarian	17.2	22.0	29.0	41.7	48.7	81.2	71.1	71.1	96.4	2.1	2.2	1.6	1.0	0.6	0.6
Icelandic	34.2	34.2	46.0	61.4	64.1	84.4	76.1	75.2	97.4	1.5	1.6	0.8	0.8	0.5	0.5
Irish	31.8	35.6	43.0	44.7	48.8	70.4	54.3	56.5	90.4	2.7	2.9	1.5	1.9	1.1	1.4
Italian	44.9	47.2	50.1	73.8	86.2	89.6	79.9	94.7	96.9	2.0	2.0	0.7	0.5	0.6	0.2
Khaling	3.9	1.5	1.7	18.4	17.2	20.5	53.8	47.2	69.4	4.3	4.7	1.9	3.0	0.8	1.3
Kurmanji	82.3	86.0	87.1	88.4	88.6	94.0	92.2	91.4	98.4	0.5	0.5	0.2	0.3	0.1	0.2
Latin	16.0	12.6	12.2	36.8	28.3	45.3	45.6	37.4	74.9	2.8	3.2	1.1	1.7	0.9	1.3
Latvian	62.1	59.7	57.9	85.1	86.2	88.3	91.0	91.6	96.7	0.8	1.0	0.3	0.3	0.3	0.2
Lithuanian	23.5	19.3	19.8	53.0	50.1	62.5	64.7	63.3	90.2	1.9	2.2	0.7	1.0	0.5	0.6
Lower-Sorbian	34.3	40.0	40.2	70.5	79.4	86.1	86.0	86.5	96.6	1.3	1.3	0.6	0.4	0.3	0.3
Macedonian	50.0	49.0	47.3	82.3	84.7	91.4	91.9	92.2	97.9	1.0	1.2	0.3	0.3	0.2	0.2
Navajo	0.0	17.7	17.1	0.0	28.8	37.8	0.0	38.2	57.8	9.0	3.8	9.0	3.2	9.0	2.7
Northern-Sami	15.4	10.8	10.8	35.7	32.3	41.4	61.1	59.1	78.1	2.4	3.3	1.5	1.9	0.8	1.0
Norwegian-Bokmal	69.0	72.9	90.6	79.8	80.7	95.0	90.6	91.0	98.8	0.5	0.4	0.3	0.3	0.2	0.2
Norwegian-Nynorsk	50.8	53.5	78.2	63.3	64.7	95.2	78.3	78.4	98.0	0.9	0.8	0.6	0.6	0.4	0.4
Persian	27.3	30.2	29.3	65.4	70.3	77.1	77.6	81.7	90.7	3.4	3.4	1.1	1.2	0.6	0.6
Polish	41.9	44.1	46.4	75.2	76.4	84.6	89.4	89.1	95.8	1.6	1.6	0.5	0.6	0.2	0.3
Portuguese	60.3	58.6	59.7	92.9	92.2	92.8	97.4	97.2	98.2	1.0	1.1	0.1	0.2	0.0	0.0
Quechua	17.2	13.5	12.5	68.1	50.2	57.0	94.7	90.6	95.9	6.7	7.1	1.7	2.9	0.1	0.4
Romanian	44.1	42.3	43.9	70.2	73.3	84.6	80.4	79.1	94.7	1.6	1.9	0.9	0.9	0.7	0.8
Russian	42.8	46.4	43.0	75.0	77.5	85.6	82.0	83.7	95.9	1.3	1.4	0.7	0.7	0.6	0.6
Scottish-Gaelic	48.0	48.0	60.0	52.0	46.0	86.0	-	-	-	0.7	0.9	0.8	0.9	-	-
Serbo-Croatian	21.3	19.5	19.0	65.8	66.7	73.1	84.0	86.3	94.0	2.7	2.9	0.9	0.9	0.4	0.4
Slovak	41.9	47.5	53.7	70.7	73.7	86.7	85.2	83.7	97.8	1.0	1.0	0.5	0.5	0.3	0.3
Slovene	47.4	48.7	52.3	81.9	82.6	89.3	89.8	88.4	95.8	0.9	1.0	0.3	0.3	0.2	0.2
Sorani	20.5	18.8	19.3	52.8	51.6	67.2	64.3	59.0	90.0	3.4	3.7	1.1	1.6	0.7	1.1
Spanish	58.6	53.7	53.3	85.4	84.9	87.0	90.6	92.4	95.6	1.2	1.6	0.3	0.4	0.2	0.2
Swedish	54.3	59.2	69.7	73.7	76.0	93.7	85.4	85.1	98.0	0.9	0.8	0.5	0.4	0.3	0.3
Turkish	14.3	12.1	13.7	33.1	41.7	63.6	72.9	73.2	95.3	4.3	4.7	2.9	2.1	0.8	0.7
Ukrainian	40.7	42.8	52.5	71.5	74.1	85.9	86.3	85.3	96.3	1.0	1.2	0.5	0.5	0.3	0.3
Urdu	30.3	29.7	30.4	86.1	83.4	88.5	95.8	95.1	98.9	4.2	4.3	0.3	0.4	0.1	0.1
Welsh	15.0	13.0	13.0	54.0	51.0	60.0	67.0	67.0	86.0	1.6	2.5	1.0	1.2	0.5	0.5

Table 4: Accuracy and average Levenshtein distance score of the baseline (B), the holistic approach (H), and the oracle (O) for each language on all sizes of training data: low, medium, and high (No high training data for Scottish-Gaelic).

Janecki, Klara, 2000. *300 Polish Verbs*. Barron's Educational Series.

Kann, Katharina and Hinrich Schütze, 2016. MED: The LMU System for the SIGMORPHON 2016 Shared Task on Morphological Reinflection. In *SIGMORPHON 2016*.

Kuroda, Kow, 2016. How are inflectional paradigms represented (in the mind)? Formal Concept Analysis meets Czech declensional paradigms. In *Japanese NLP Assoc. 2016*.

Lepage, Yves, 2004. Analogy and formal languages. *Electronic notes in theoretical computer science*, 53:180–191.

Levenshtein, V.I., 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics-doklady*, 10(8):707–710.

Neuvel, Sylvain and Rajendra Singh, 2001. Vive la différence ! what morphology is about. *Folia Linguistica*, 35(3-4):313–320.

Singh, Rajendra (ed.), 2000. *The Yearbook of South Asian Languages and Linguistics-200*.

Singh, Rajendra and Alan Ford, 2000. In praise of Sakatayana: some remarks on whole word morphology. In *In (Singh, 2000)*.

Stroppa, Nicolas and François Yvon, 2005. An analogical learner for morphological analysis. In *CoNLL 2005*.

Wagner, Robert A. and Michael J. Fischer, 1974. The string-to-string correction problem. *Journal of the Association of Computing Machinery*, 21(1):168–173.

Yvon, François, 2003. Finite-state machines solving analogies on words. Technical report, ENST.