Cost-Effectiveness in Building a Low-Resource Morphological Analyzer for Learner Language

Scott Ledbetter
Indiana University
Bloomington, IN, USA
saledbet@indiana.edu

Markus Dickinson
Indiana University
Bloomington, IN, USA
md7@indiana.edu

Abstract

In this paper, we describe the development of a morphological analyzer for learner Hungarian, outlining extensions to a resource-light system that can be developed by different types of experts. Specifically, we discuss linguistic rule writing, resource creation, and different system settings, and our evaluation showcases the amount of improvement one gets for differing levels and kinds of effort, enabling other researchers to spend their time and energy as effectively as possible.

1 Introduction

There is much work on developing technology and corpora for lesser-resourced languages (LRLs), involving varying assumptions about the amount of available data (e.g., Feldman and Hana, 2010; Duong et al., 2015; McDonald et al., 2011; Garrette et al., 2013; Garrette and Baldridge, 2013; Lynn et al., 2013; Smith and Dickinson, 2014). To our knowledge, though, there has been very little work focusing on tools for automatically analyzing learner language. Yet LRLs present many challenges and opportunities, not least of which is the chance, through language learning, to increase awareness and usage of the language. In that light, we build from our work in Ledbetter and Dickinson (2015) to develop a morphological analyzer for learner Hungarian in a resource-light way, extending the framework to handle a wider range of phenomena and to increase accuracy, addressing the trade-offs between effort and performance.

Part of the purpose in Ledbetter and Dickinson (2015) is to push for the automatic analysis of learner language beyond English, incorporating a wider range of linguistic and error patterns. But there is little indication about how much effort is required to build a system of sufficient accuracy. Garrette and Baldridge (2013) point out exactly the time spent in annotation, and we follow in that vein of estimating time costs, here for analyzing learner language, to foster resource-light development (cf. Smith and Dickinson, 2014).

Indeed, there is a convenient connection between LRLs and the automatic analysis of learner language, one enabling a rule-based approach. In a low-resourced setting, there is a lack of data for data-driven modeling, and one can write rules in a short amount of time; because such rules are linguistically motivated, they are relatively easy to connect to tasks such as providing feedback on learner input or modeling learner behavior, i.e., tasks requiring linguistic analysis. Systems for learner language are also amenable to low-resource development in that the vocabulary is often restricted and the requirement for high precision dovetails nicely with using linguistic rules. Such an approach allows for quick, transparent development, helping identify parts of the linguistic system that the tool gets (in)correct.

Given the need to interact with NLP researchers, educators, and learners, this linguistic transparency is a key property (cf. Loukina et al., 2015). In this work, we propose a number of different kinds of extensions to a system, appropriate for different aspects of analysis. The system is rule-based, which—in addition to being appropriate for Hungarian morphology in a resource-light setting—makes it feasible to plug in different components and thus allows one to test the effect of different extensions. We envision different kinds of experts (linguists, teachers, NLP researchers, etc.) being able to contribute within such a paradigm, thus broadening the range of who can contribute to successful analyzer design.

Corresponding to different types of knowledge, we propose four kinds of system improvements: 1) writing linguistic rules (e.g., exception marking); 2) creating resources (e.g., name dictionary); 3) optimizing system settings (e.g., automatic ranking); and 4) analyzing system output. Seeing the effects of these improvements can help contribute to a broader discussion of which components are most in need of development for learner data (cf. constraint relaxation (Reuer, 2003; Schwind, 1995)). The main contribution of this work is thus to outline and identify the benefits of different kinds of resources for building a transparent analyzer, highlighting the best gains for the amount of time one has available.

In section 2 we cover the basics of Hungarian, turning to learner Hungarian and the general framework in section 3. In section 4 we outline the types of improvement, detailing specific modifications to the system and the amount of effort involved in each one. Section 5 then shows the results for tagging accuracy, and we provide general advice on prioritizing effort in section 6. Although our current results are below the state-of-the-art, we have spent little time in getting a workable system. We do not address in this paper how accurate an analyzer needs to be in order to be deployed for providing learner feedback, but it is important to note: a) we highlight starting points for development, not a finished product; and b) for some purposes—such as assisting in the semi-automatic annotation of a learner corpus or targeting subparts of the grammar (for which the system is more accurate) for learner modeling—even the current system can be of benefit.

2 Hungarian

Hungarian is a Finno-Ugric language known for its rich agglutinative morphology, in which words are formed by joining morphemes together, as shown in (1). Morphemes convey information such as number and case (e.g., INESSive) for nouns (1a), number, person, tense, and definiteness for verbs (1b), or changes to part-of-speech (1c). Hungarian also exhibits vowel harmony, which requires that the backness of vowels (represented by the feature [+/-BK]) matches during affixation (Törkenczy, 2008); e.g., the front vowel variant *-ben* is inappropriate for (1a).

(1)	a.	fá	-k	-ban
		tree[+BK]	-PL	-INESS[+BK]
		'in trees'		
	b.	felejt	-ett	-em
		forget[-BK]	-PST[-BK]	-1SG.INDEF[-BK]
		'I forgot'		
	c.	álm	-atlan	-ság
		sleep[+BK]	-NEG.ADJ[+BK]	-NOM[+BK]
		'sleeplessness/insomnia'		

3 Framework

We build from the analyzer in Ledbetter and Dickinson (2015). Given the nature of Hungarian, the morphological analyzer follows in the tradition of rule-based approaches (Prószéky and Kis, 1999; Megyesi, 1999; Trón et al., 2005, 2006), as a statistical approach relying on large amounts of data for training would be more challenging in a resource-light setting. Data sparsity, common with agglutinative languages in which a given wordform appears few times in training data, compounds this problem. The amount and type of resources necessary for our work is comparable to other research on morphologically-rich languages in low-resource settings, e.g. for some Indian languages (Singh et al., 2006; Shrivastava and Bhattacharyya, 2008; Alfter, 2014). Details of the system are in section 3.2.

3.1 Data

To develop and evaluate a system, we rely on both native language and learner data. The corpus of learner data was collected from L1 English students of Hungarian, divided into three levels of proficiency (Beginner, Intermediate, Advanced) as determined by course placement in one of three two-semester sequences. The corpus consists of journal entries, each a minimum of ten sentences in length on a topic selected by the student. The data and error annotation are described in Dickinson and Ledbetter

(2012). For native data, we use the Szeged Corpus of Hungarian (Csendes et al., 2004).

For development of all our improvements, we use the same 1000 words of native data and 1024 words of learner data that were used in Ledbetter and Dickinson (2015). The native data is taken from compositions, so as to be relatively comparable to the journal entries found in the learner corpus. For evaluation (section 5), we use new approximately same-sized (1000, 1032) sections of data, i.e., the test sets are comprised of new compositions never before seen by the analyzer. Note that there is no training data, as we are assuming a small set of resources.

3.2 Morphological analyzer

The morphological analyzer is affix-driven and works simply by using a small set of known affixes to derive a final analysis. For example, with a suffix *-at* in the grammar that takes a noun stem (N) to the left to derive a cased noun phrase (KP)—notated KP\N—the word *házat* ('house+ACC') obtains a KP analysis, segmented as *ház+at*.

The analyzer supports using features, too, e.g., KP\N {vh:bk} to indicate that vowel harmony (*vh*) requires back (*bk*) vowels here. Error detection can thus be performed on top of morphological analysis simply by allowing for and storing feature clashes (instead of requiring feature unification). In this paper, we focus on providing correct morphological tags, as this is the tool's most primary function.

The baseline performance of the morphological analyzer assumes the minimum of resources required to function: a knowledge base of grammatical affixes and the analyzer (i.e., rule compiler) itself. In section 5, we report two baseline scores, with 10 and then with 50 affixes. Although Ledbetter and Dickinson (2015) also start with a dictionary, we assume no other resources for our baseline; with only affixes, the resulting scores are accordingly low. Each improvement to the analyzer (section 4) is then evaluated separately with respect to this baseline.

Since we are concerned about time, we should note that the rule compiler is around 3000-4000 lines of Python code (depending on how one counts comments, data handling, printing of intermediate output, system-internal checks, etc.). We estimate it would take 1–2 months for someone to build, but as it relies on a CYK parser (see details in Ledbetter and Dickinson, 2015) and as one may choose different kinds of rule compilers for the same effect (e.g., a constraint grammar compiler (Didriksen, 2016) or an FST-based system (Hulden, 2009)), this time could be significantly less.

4 Improvements

The morphological analyzer is designed with low-resource scenarios in mind, starting with as little as a rule compiler and however much grammatical information one has time to specify in rules (i.e., affixes). We want to improve beyond this simple design, but we need to determine the best ways to improve. Ideal improvements: a) require little time to implement; b) contribute (significantly) to better performance; and c) are as transparent as possible.

As mentioned in section 1, improvements to the analyzer are divided into four categories. For ease of implementation and development, each category can represent a different person or team, based on the resources and expertise available.

In each subsection below, we discuss a series of related improvements we have completed and estimate the time required to implement them to the same degree we have. We assume 1 day = 8 hours of work for one person. For these estimates, we do not strictly follow our own experience, as we did not precisely log the time and, more importantly, we expect future users to require less time to choose and implement changes. We also assume someone of moderate expertise in the area for which the improvements need to be made—e.g., someone familiar with Hungarian linguistics enriching the knowledge base (section 4.1) vs. someone familiar with corpus linguistics extracting a most frequent word list (section 4.2). The improvements in section 4.4 are more likely than any other to involve discussion and cooperation among the different people, as it involves incremental changes in the other areas.

4.1 Linguistic rule writing

The first set of improvements involves writing rules that correspond to linguistic generalizations. These come in different levels of granularity, e.g., general rules (#1) vs. exceptions (#3).

1. Enrich the knowledge base (KB): The morphological analyzer relies primarily on grammatical rules encoded in the knowledge base (KB) of affixes, all of which can be obtained from a traditional grammar reference or language textbook. The size of the KB can vary, and the system's performance increases with richer and more complete grammatical information. The baseline systems detailed in section 3.2 assume a modest base of either 10 or 50 affixes.

Although we experiment with small KBs, it is possible to obtain over 200 affixes for Hungarian using a grammar reference (Törkenczy, 2008), such as the derivational suffix in (2). Here, the affix *-ság* creates a noun (N) when combined with an adjective (A) to its left. Furthermore, the adjective stem must contain back vowels to match the harmonizing suffix (indicated by the feature *vh* and its required value *bk* in braces). An example word can be found in (3). The adjective *boldog* ('happy') can combine with the nominalizing suffix *ság* because it contains back vowels, as specified in the knowledge base. Thus, the completed derivation is the noun *boldogság* ('happiness'). **Time estimate:** 2 days.

(2) *-ság*: N\A{vh:bk}

(3)

boldog	*-ság*
A{vh:bk}	N\A{vh:bk}

"happiness" (N)

2. Modify knowledge base features: The KB can be augmented with grammatical features, used to constrain, i.e., eliminate certain combinations of morphemes. (In the case of learner language, features may only disfavor analyses—see the *Free* setting in section 4.3.) As seen in (2), the feature *vh* constrains a derivation based on vowel harmony. Features also supply key information during derivation to increase the accuracy of morphological tags. In (4), the feature *k* indicates that the noun stem (N) suffixed with *-t* will become a cased noun phrase (KP) in the accusative case (*acc*).

(4) *-t*: KP{k:acc}\N

Note that modifications to grammatical features are included in all results presented in section 5, as removing a single feature from each entry in the KB to measure its effectiveness would take a long time and provide little in the way of guidance for future researchers. **Time estimate:** 1-2 days.

3. Encode exceptions (Except): Any target language will have exceptions to grammatical rules. A list of the most frequent grammatical exceptions (even as few as 5–10), including handling of irregulars, may provide a boon to performance. In Hungarian, irregular stem changes during verb conjugation, for example, can mask the relatedness of different forms in the paradigm, and only one stem will be found in a dictionary (when a dictionary is used). Forms like those in (5) can be linked to a single stem—the dictionary form of 'to be,' *van*—to facilitate a complete paradigm. The analyzer makes use of only five such cases, targeting the most frequent irregular verb stem changes. **Time estimate:** 1 day.

(5)

a.	*vagy*	*-ok*	b.	*van*	*-nak*
	be	-1SG		be	-3PL
	'[I] am.'			'[they] are.'	

4.2 Resource creation

The next improvements involve building resources, generally corresponding to finding data that fits the context under which the language is being produced (e.g., learners with a first language (L1) of English).

1. Obtain a language dictionary (Dict): A dictionary of attested target language words is a straightforward improvement. If available digitally, a target language dictionary is simple to add; otherwise, one must be built by hand or via transcription. In the latter case, a minimal list of words can be obtained from a grammar reference or language textbook and then digitized. Our analyzer accesses an English-Hungarian dictionary of 161,000 entries (Vonyó, 1998) with words only (i.e., no part-of-speech or other grammatical information). **Time estimate:** 1 day (electronic resource).

2. Obtain a list of common names (Names): A common problem for morphological analysis is the treatment of unknown words. A list of names common to the source (when known) and target languages is often easy to acquire or create and could alleviate a frequent cause of unknown words. The analyzer has access to a list of 400 common first names, 200 from English (L1 in our data) and 200 from Hungarian (L2), obtained from *behindthename.com*. **Time estimate:** 1 day.

3. Obtain a list of the most frequent words (Freq): A small number of words (e.g., function words) appear very frequently in written data, independent of domain, while the vast majority of words appear only a few times. With access to corpus data, a list of the most frequent words in the language with the proper morphological tags (added by hand) can ensure that the analyzer is accurate for a large amount of written data. Of the 100 most frequent words in the Hungarian National Corpus (Oravecz et al., 2014), 50 were selected and placed in a database for the analyzer along with the appropriate morphological tags. Selection proceeded one word at a time, beginning with the most frequent, using several criteria, including familiarity to learners, opaque rather than transparent morphology, and appropriateness to genre. **Time estimate:** 1 day.

4.3 System settings

The third set of improvements focus on implementation, deriving ways to obtain the best analysis from among many possible ones or in cases where little-to-no information is known.

1. Devise a method to rank analyses (Rank): Morphological analysis for learner data must be more flexible than similar methods for native language data. Misspelled words, clashes of grammatical features, and a non-targetlike distribution of morphological affixes can all cause failure for rules designed to analyze the standard target language. Overgeneration is accordingly a significant problem for any system that attempts to account for such differences. A method for prioritizing one possible analysis over another is essential for reducing ambiguity, which in turn has a direct effect on precision.

We created a simple method to rank analyses in the output of the analyzer. First, any analysis that makes use of a stem found in the dictionary is ranked higher than one that doesn't. Second, an analysis is ranked higher than another if it exhibits fewer clashes of grammatical features during derivation (section 3.2). If neither of the above rules applies, analyses are ranked in the order they were created. Future work can explore more sophisticated methods of ranking. **Time estimate:** 1 week.

2. Determine a default tag for unknown words (Def): As indicated earlier, unknown words can reduce accuracy during analysis, and learners are prone to innovate new forms. This problem can be minimized by assigning a default analysis to unknown words. Our analyzer proposes a single tag *Nc-sn* (noun: common, singular, nominative case) when it fails to produce a derivation. Following standard practice in POS tagging, this default tag specifically targets the most likely open class of words. **Time estimate:** 1 day.

3. Hypothesize roots (Hyp): The design of the analyzer emphasizes flexibility to account for the differences between learner data and standard target language data. In order to make the system more robust, a derivation can hypothesize a potential stem after an affix from the knowledge base has been recognized. This allows for non-standard roots (e.g., those that have been misspelled) during derivation, even if the system is relying on dictionary lookup.

(6)	**haz*	*-ban*
	house	-INESSIVE
	'in [a] house.'	

In (6), for example, a system built to analyze standard Hungarian may not recognize *haz* as part of a valid word, as *ház* ('house') is the likely intended form. By completing a derivation when possible, on the basis of the suffix (e.g., N_{hyp} + KP\N), the analyzer can be more robust—though, it can also overposit analyses. **Time estimate:** 2 days.

4. Relax constraints (Free): Another method for introducing flexibility into the analyzer involves the features associated with the affixes in the KB. When analyzing target language data, the features of stems and affixes should unify, resulting in a complete derivation. In learner data, on the other hand, we expect feature clashes as learners have not yet mastered the rules of the target language grammar. By relaxing the unification of features to allow clashes, we can more often obtain complete derivations.

(7) **ház* -*ben*{vh:fr}
house -INESSIVE
'in [a] house.'

Consider the example in (7). The suffix *-ben* requires a stem with front vowels (the value of the feature *vh* must be *fr*), but the stem *ház* contains back vowels. With the appropriate setting, our analyzer permits the derivation and provides the correct morphological tag. **Time estimate:** 2 days.

The last two modifications (Hyp, Free) are proposed in Ledbetter and Dickinson (2015), but need to be evaluated within the space of other settings.

4.4 System analysis

The improvement in this section is relevant to the whole system, relying on other settings and not easily isolated to a single area of expertise. No evaluation of this improvement is provided, as it was undertaken at several stages during development.

1. Perform a system/error analysis: Time can be spent on an analysis of system performance, using a small set of (annotated) development data. At any stage of development, it is useful to inspect the output of the analyzer and investigate potential mistakes in rules, affixes, features, improvements, or the underlying code itself. Thus, creating a small set of annotated data is important for this stage.

Indeed, some of the time for the other improvements is "hidden" in this task. In our experience, we were able to identify a number of ways in which the system was not functioning as expected. We identified affixes omitted from the KB, discovered new features to restrict spurious analyses, and developed the initial ideas for ranking analyses—in addition, of course, to debugging the underlying code. Resolving each of these led to an increase in accuracy. **Time estimate**: 1 month (or more).

5 Evaluation

After devising the improvements on the development data, we evaluate the morphological analyzer first on native Hungarian data and then on learner data, recording precision, recall, and accuracy measures. Each improvement to the system is first considered individually, assessing its value apart from all others, and then combined with other improvements to determine the overall effectiveness.

The test sets, as mentioned in section 3, are new excerpts from the target (1000 tokens) and learner corpora (1032 tokens). The analyzer produces as output one or more morphological tags for each word in the input. During evaluation, this list of output tags is compared to a list of gold standard tags. For native data, the gold tag list from the target language corpus contains the correct context-specific tag as well as a list of possible tags based on morphology. The annotation of the target language corpus uses corrected output from the *magyarlanc* (Zsibrita et al., 2013) tool. For learner data, we annotate a single gold standard tag for each word.

We evaluate how many tags from the analyzer match possible tags from the gold standard and whether the correct context-specific tag appears in the output. Specifically, we report:

- **Precision (P)**: the number of tags in the output that appear in the gold standard, divided by the number of tags in the output.
- **Recall (R)**: the number of tags in the output that appear in the gold standard, divided by the number of tags in the gold standard.
- **Accuracy (A)**: the number of correctly identified context-specific tags divided by the total number of words in the input.

5.1 Individual improvements

The precision, recall, and accuracy measures for each individual improvement are given in Table 1 for native data and Table 2 for learner data. The top half of the table, labeled *KB-10*, indicates that the analyzer used a knowledge base of ten affixes, and the lower half of *KB-50* indicates 50 affixes. All other improvements are identified by columns with

KB-10	ML	Base	Except	Dict	Names	Freq	Rank	Def	Hyp	Free
P	0.970	0.018	0.023	0.107	0.018	0.385	0.018	0.221	0.145	0.018
R	0.413	0.008	0.010	0.051	0.008	0.182	0.008	0.094	0.070	0.008
A	0.930	0.018	0.023	0.089	0.018	0.303	0.018	0.058	0.125	0.018
KB-50	**ML**	**Base**	**Except**	**Dict**	**Names**	**Freq**	**Rank**	**Def**	**Hyp**	**Free**
P	0.970	0.037	0.041	0.213	0.037	**0.401**	0.037	0.239	0.221	0.037
R	0.413	0.016	0.018	0.115	0.016	**0.190**	0.016	0.103	0.146	0.016
A	0.930	0.036	0.041	0.194	0.036	**0.321**	0.036	0.076	0.245	0.036

Table 1: Evaluation for each improvement (native data)

KB-10	ML	Base	Except	Dict	Names	Freq	Rank	Def	Hyp	Free
P	0.850	0.003	0.016	0.062	0.003	0.290	0.003	0.104	0.079	0.003
R	0.850	0.003	0.016	0.068	0.003	0.311	0.003	0.105	0.091	0.003
A	0.850	0.003	0.016	0.068	0.003	0.311	0.003	0.105	0.091	0.003
KB-50	**ML**	**Base**	**Except**	**Dict**	**Names**	**Freq**	**Rank**	**Def**	**Hyp**	**Free**
P	0.850	0.011	0.024	0.147	0.011	**0.297**	0.011	0.112	0.162	0.011
R	0.850	0.011	0.024	0.184	0.011	**0.319**	0.011	0.112	0.240	0.011
A	0.850	0.011	0.024	0.184	0.011	**0.319**	0.011	0.112	0.240	0.011

Table 2: Evaluation for each improvement (learner data). Recall (R) = Accuracy (A) since only one gold standard tag is annotated.

labels matching their descriptions in section 4. The *ML* column provides a comparison with a native language tool, *magyarlanc* (Zsibrita et al., 2013).

For native data (Table 1), *magyarlanc* attains nearly perfect precision and very high accuracy. This is to be expected, given that the tool was used in creating the gold standard annotation of the test corpus. Recall, naturally, is less, as *magyarlanc* disambiguates among all the context-independent tags. These figures provide a rough estimate of the difficulty of morphological analysis for Hungarian. As expected, the baseline performance of our analyzer is extremely low, at 2% precision and less than 1% recall with a meager knowledge base, and about double that with a slightly richer one.

For every improvement, the size of the knowledge base increases performance. Three different improvements (Names, Rank, and Free), however, make no difference in performance as compared to Base, irrespective of the difference in size of the KB. By far, the most effective improvement is Freq (most frequent word list), attaining the highest performance (e.g., precision above 40%) with 50 affixes in the KB. The Dict, Def, and Hyp improvements also exhibit noteworthy gains for all three metrics. There is a slight gain from the addition of Except, but no more than 1%—unsurprising, given the small number of encoded cases.

For learner data (Table 2), performance is lower across the board. The *magyarlanc* tool loses about 13% in precision and 8% in accuracy, illustrating the increased difficulty of the task of analyzing learner language. Our analyzer begins with baseline scores at 1% or lower for both KB-10 and KB-50. The results for learner data parallel those for native data: the Names, Rank, and Free improvements provide no increase in performance, while Freq provides the greatest gains, reaching nearly 30% precision and 32% recall and accuracy. Likewise, the Dict, Def, and Hyp improvements provide noticeable gains, with Except providing small gains.

While no single improvement on its own results in very high scores, the evaluation highlights the most effective improvements. The Freq improvement alone, for example, contributes nearly 40% to precision for native language data.

5.2 Improvements in combination

We now turn to an evaluation of the system in successive iterations, beginning with the baseline system and adding one new improvement each time. Table 3 shows the results for native data, while Table 4 shows the results for learner data. The last column in each table gives the results of a final test with all improvements added. In each series of tests, a knowledge base of 50 affixes is used.

	Base	+Dict	+Freq	+Def	+Hyp	All
P	0.037	0.213	0.497	**0.579**	0.472	0.451
R	0.016	0.115	0.281	0.323	0.350	**0.363**
A	0.036	0.194	0.479	0.517	0.567	**0.594**

Table 3: Evaluation for stepwise improvements in combination, moving left to right (KB 50, native data)

	Base	+Dict	+Freq	+Def	+Hyp	All
P	0.011	0.147	0.378	**0.457**	0.384	0.370
R	0.011	0.184	0.492	0.592	0.631	**0.661**
A	0.011	0.184	0.492	0.592	0.631	**0.661**

Table 4: Evaluation for stepwise improvements in combination, moving left to right (KB 50, learner data)

Note that the order of added improvements is based on both practical concerns and effectiveness in the individual evaluations discussed above. The first improvement added to the system is Dict (a target language dictionary): though this did not provide the greatest gains in the evaluation metrics, it is the most central resource and one of the easiest to obtain. The next improvements (Freq, Def, and Hyp) were selected for their contributions during individual evaluations, and each is added in an order corresponding to its impact on performance. Finally, the other modules (Except, Names, Rank, Hyp) are added for the *All* model.

For native data (Table 3), the system displays a general trend of increasing precision, recall, and accuracy as improvements are added. Precision reaches its peak at 58% with the addition of the default tag (Def) improvement, while recall and accuracy continue to increase. Both recall and accuracy reach their maximum with all improvements added. Precision at its maximum remains well below that of *magyarlanc*'s 97%, but recall reaches 36% with all improvements, just 5% short of *magyarlanc*. The *All* model, despite making use of every improvement, exhibits a drop in precision; an inspection of the data reveals that this is due to the increased number of proposed analyses. The system is outputting more tags that are correct (thus, the increase in recall), but the number of proposed analyses increases by a larger margin, and thus precision falls.

For learner data (Table 4), the trends are nearly identical for the successive improvements. The Dict, Freq, and Def improvements obtain a precision of 46%, compared to *magyarlanc*'s 85%, while recall and accuracy rise to 66% with all improvements added, about 20% from *magyarlanc*'s 85%. As with native data, precision falls with the *All* model, even though the number of correct tags is at its highest. The figures for maximum recall and accuracy are encouraging, considering that *magyarlanc* makes use of contextual information for disambiguating possible morphological tags. Our analyzer currently uses no contextual information, analyzing one word at a time. With the incorporation of additional information such as syntactic frames, the results could be more competitive with those for native data. It is also important to note that the *magyarlanc* tool is based on the *morphdb* database (Trón et al., 2006), which represents years of work, while many of our system improvements require less than a month.

It may seem surprising that recall and accuracy for learner data surpass the results for native data. This stems from the fact that the gold standard annotation for learner data has only one correct tag. Similarly, for the calculation of precision, multiple analyses proposed by the system adversely affect the score, even if alternatives may be correct for a word.

Although the evaluation data is different, we can roughly compare accuracies of 0.594 and 0.661 to corresponding accuracies of 0.505 and 0.478 in Ledbetter and Dickinson (2015). This gain is in spite of using 205 affixes in the KB in Ledbetter and Dickinson (2015) vs. 50 here, highlighting the point that some improvements (e.g., Freq) are more time-effective than others (e.g., expanding the KB).

6 Discussion and Outlook

Concerning ourselves with the balance between improving analyzer accuracy and/or coverage and spending time to do so, the most valuable improvement for the analysis of L2 Hungarian morphology is a list of the most frequent words and the corresponding tags. Other notable improvements include incorporating a method for hypothesizing stems, a dictionary of Hungarian words, and a default tag for analysis. Finally, some rich knowledge base of grammatical information is essential, and even small KB sizes can have moderate performance with other improvements in place.

With the exceptions of enriching the knowledge base and creating a method for hypothesizing stems, each improvement is estimated to need only a single day to implement (to the level we have). Together, all these improvements would require just over a single work week. Additional time, when available, may be spent on further improvements or system analysis, as discussed in section 4.4.

The recommended improvements are spread across several areas of expertise, and many require at least some existing resources. In some settings, it may not be possible to implement even these few. With these restrictions in mind, we propose the following prioritization of improvements:

1. Target language dictionary. While not the largest improvement, it is hard to envision adequate analysis without a dictionary. For many languages, the ease of acqusition and widespread availability of this resource make it quick and effective to obtain. In the absence of an electronic dictionary, it is possible to extract some lexicon from a grammar resource or language textbook; much more time will then be spend in digitization.
2. Frequent word list. This improvement provides the greatest benefit for performance, requiring a moderate amount of target language data to ascertain the most common words. It may also be necessary to add morphological information if the corpus isn't annotated. If no corpus data is available, one could also use intuition to derive something akin to the most common, or most salient, words in a language.
3. Default tag. One of the simplest methods to improve the analyzer, a default tag requires very little knowledge of coding to implement.
4. Knowledge base. The size of the knowledge base directly affects performance, though it takes longer to implement. We have shown that as few as 10 affixes may be used to begin analysis, and 50 shows great promise.
5. Hypothesized stems. This requires knowledge of the underlying code of the analyzer and may need fine-tuning for different target languages.
6. System analysis. System analysis should be a part of every stage of development, but it also requires the most time of any improvement.

Performance of the system with these improvements does not reach the level of a tool designed for native language morphological analysis. However, our work illustrates that significant gains can be made with fewer resources, most of which are easily accessible, assuming only a grammar reference or language textbook to begin. With the prioritized list of improvements above, time and energy can be used efficiently to streamline the development of a low-resource morphological analyzer.

There is still much to do to get a better handle on the impact of different improvements to obtain a better analyzer. We envision exploring larger knowledge bases and their effect on increasing ambiguity, developing more methods for ranking analyses—perhaps incorporating trends from any possible available learner data—and experimenting with resources (dictionaries, name lists) which are targeted towards particular domains or learning contexts. Additionally, one question has still not been addressed by our evaluation: how good is good enough? Utilizing other evaluation measures that probe into real-world usage, e.g., for error detection and grammar extraction (Ledbetter and Dickinson, 2015), will be crucial in that respect.

Acknowledgments

We would like to thank the three anonymous reviewers for their helpful feedback.

References

David Alfter. 2014. *Morphological analyzer and generator for Pali*. Ph.D. thesis, University of Trier, Trier, Germany.

Dóra Csendes, János Csirik, and Tibor Gyimóthy. 2004. The szeged corpus: A pos tagged and syntactically annotated hungarian natural language corpus. In *Text,* *Speech and Dialogue: 7th International Conference, TSD*, pages 41–47.

Markus Dickinson and Scott Ledbetter. 2012. Annotating errors in a hungarian learner corpus. In *Proceedings of the 8th Language Resources and Evaluation Conference (LREC 2012)*.

Tino Didriksen. 2016. *Constraint Grammar Manual: 3rd version of the CG formalism variant*. GrammarSoft ApS.

Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 845–850. Beijing, China.

Anna Feldman and Jirka Hana. 2010. *A resource-light approach to morpho-syntactic tagging*. Rodopi, Amsterdam/New York, NY.

Dan Garrette and Jason Baldridge. 2013. Learning a part-of-speech tagger from two hours of annotation. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 138–147. Association for Computational Linguistics, Atlanta, Georgia.

Dan Garrette, Jason Mielens, and Jason Baldridge. 2013. Real-world semi-supervised learning of pos-taggers for low-resource languages. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 583–592. Association for Computational Linguistics, Sofia, Bulgaria.

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

Scott Ledbetter and Markus Dickinson. 2015. Automatic morphological analysis of learner hungarian. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 31–41. Denver, CO.

Anastassia Loukina, Klaus Zechner, Lei Chen, and Michael Heilman. 2015. Feature selection for automated speech scoring. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 12–19. Denver, CO.

Teresa Lynn, Jennifer Foster, Mark Dras, and Josef van Genabith. 2013. Working with a small dataset semi-supervised dependency parsing for irish. In *Proceedings of SPMRL 2013*. Seattle.

Ryan McDonald, Slav Petrov, and Keith Hall. 2011. Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 62–72. Association for Computational Linguistics, Edinburgh, Scotland, UK.

Beata Megyesi. 1999. Improving Brill's POS tagger for an agglutinative language. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 275–284.

Csaba Oravecz, Tamás Váradi, and Bálint Sass. 2014. The hungarian gigaword corpus. In *Proceedings of LREC 2014*.

Gabor Prószéky and Balazs Kis. 1999. A unification-based approach to morpho-syntactic parsing agglutinative and other (highly) inflectional languages. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 261–268.

Veit Reuer. 2003. Error recognition and feedback with lexical functional grammar. *CALICO Journal*, 20(3):497–512.

Camilla B. Schwind. 1995. Error analysis and explanation in knowledge based language tutoring. *Computer Assisted Language Learning*, 8(4):295–324.

Manish Shrivastava and Pushpak Bhattacharyya. 2008. Hindi pos tagger using naive stemming: Harnessing morphological information without extensive linguistic knowledge. In *Proceedings of ICON-2008: 6th International Conference on Natural Language Processing*, pages 1–8. ICON.

Smriti Singh, Kuhoo Gupta, Manish Shrivastava, and Pushpak Bhattacharyya. 2006. Morphological richness offsets resource demand: Experiences in constructing a pos tagger for hindi. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 779–786. Association for Computational Linguistics.

Amber Smith and Markus Dickinson. 2014. Evaluating parse error detection across varied conditions. In *Proceedings of the 13th International Workshop on Treebanks and Linguistic Theories (TLT13)*. Tübingen, Germany.

Miklós Törkenczy. 2008. *Hungarian Verbs and Essentials of Grammar, 2nd ed.* McGraw-Hill, New York.

V. Trón, Gy. Gyepesi, P. Halácsy, A. Kornai, L. Németh, and D. Varga. 2005. Hunmorph: Open source word analysis. In *Proceedings of the Workshop on Software*, pages 77–85. Association for Computational Linguistics.

V. Trón, P. Halácsy, P. Rebrus, A. Rung, P. Vajda, and E. Simon. 2006. Morphdb. hu: Hungarian lexical database and morphological grammar. In *Proceedings of 5th International Conference on Language Resources and Evaluation (LREC)*, pages 1670–1673.

Attila Vonyó. 1998. A magyar elektronikus könyvtár.

János Zsibrita, Veronika Vincze, and Richárd Farkas. 2013. Magyarlanc: A toolkit for morphological and dependency parsing of hungarian. In *Proceedings of RANLP*, pages 763–771.