Abstracting Suffixes: A Morphophonemic Approach to Polish Morphological Analysis[1]

AMIR ZELDES
Institut für deutsche Sprache und Linguistik
Humboldt-Universität zu Berlin
az-omega@013.net

Abstract

This paper presents a morphophonology-based Item-and-Process approach to the finite-state lemmatization and morphological analysis of Polish. Unlike current text-based techniques, which search for all possible orthographic representations of Polish morphological suffixes, the multi-level algorithm presented here extracts morphophoneme arrays from graphemic word forms, allowing the extraction of abstract suffixes, independent of their surface representation. This makes it possible to use a simple mono-lemmatic dictionary, as well as to distinguish between homographic suffixes, and to carry out various phonological and morphological investigations using suffix fields in corpora.

1 Introduction

Lemmatization and morphological analysis are two basic tasks which are essential to a wide variety of applications in computational linguistics, such as machine translation, information retrieval and building electronic corpora. Lemmatization is understood to mean finding the basic dictionary form (or 'lemma') associated with an observed word form, a process which often entails morphological analysis, in which the grammatical categorization of the observed form is determined. The task of morphological analysis and lemmatization in Slavic languages is difficult not only because of their rich morphology, but also because inflection can change word stems, making it difficult to determine what the lemma should look like (e.g. the Polish word for '*hand*' exhibits 3 stem forms, viz. nominative: ręk-a, locative: ręc-e, genitive plural: rąk-Ø).

The basic premise of applications addressing this task in languages with suffixal morphology is that each word is comprised of two parts: a stem at the left of the word (i.e. the first *n* characters which all forms of a lemma have in common) and a suffix at the right of the word (the remaining *m* characters). The most straightforward algorithm is to go over the input string trying to break it up into all possible stem-suffix pairs, and then to look up each possible suffix in a table. For example, <pisze> '*writes*' can be divided into: p-isze, pi-sze, pis-ze, pisz-e or even pisze-, if we allow a Ø ('zero, null') suffix. The Tokarski Index is exactly such a table of suffixes for Polish[2].

However, since Polish has a very high frequency and variety of morphophonemic alternations, this approach results in both a very large list of suffixes (the Tokarski index includes over 18,000 entries), and a possible linguistic misrepresentation of the concept of 'suffix', which will frequently and inconsistently include parts of the stem. For instance <ręce> and <rąk> mentioned above, are analyzed in the Tokarski index with the suffixes '-ęce' and '-ąk', the base form of which has the suffix '-ęka' (essentially usurping part of the stem into the suffix). Furthermore, different variants of what is essentially the same suffix must be recorded separately. For example, the ordinary suffix for a nominative masculine singular adjective is -y, as in <piękny> '*beautiful*', but if the stem ends in a velar consonant it is always -i, as in <ciężki> '*heavy*'. Conversely, different suffixes can

[1] The work described in this paper was supported by DAAD grant number A/05/38058.

[2] Tokarski (1993). For implementations see Bień and Szafran (2001) and the morphological analyzer *Morfeusz* developed by Marcin Woliński and used on the IPI PAN corpus (see Przepiórkowski (2004)).

appear identical, as in the masc. personal plural of the same adjectives, where the forms seem to exhibit the opposite suffixes: <piękni> and <ciężcy>. This means that a text-based index must keep separate entries for -ny, -ni, -ki, -cy etc., which is not only redundant but also potentially error-prone. It also makes it difficult to maintain or expand the index, and possibly even to analyze unexpected loan words or productive word formations.

Partly due to (until recently) prohibitive processing costs, applications trying to deal with this redundancy have adopted lexicon-centered strategies, rather than multi-level Item-and-Process solutions, which have been effective for other languages[3]. Šipka and Končar (1997) use a Word-and-Paradigm model, defining inflection classes for Polish and Serbo-Croatian which point to text-based rules, so that each entry in the lexicon specifies the kind of inflection it undergoes, as well as any irregular forms. While this allows generation of whole paradigms for each entry, it requires substantial lexicographic work. Furthermore, various patterns which may exhibit the same mutation rule must be defined separately (e.g. in Polish an alternation between 'o' and 'ó' occurs in identical phonological environments in the fem. and neut. genitive plural, the masc. singular and the imperative, to name a few). In order to reduce the amount of patterns required, the authors also implement 'string cleanup rules' at the orthographic level to adjust illegal strings (e.g. Polish <ky> > <ki>), which effectively form text-based two-level rules.

Recent formalizations of Czech morphology (Osolsobě (1997), Osolsobě et al. (2002), Sedláček and Smrž (2001)) adopt an Item-and-Arrangement approach, where all variant stems of a lemma are found in the lexicon with instructions as to which stem is used for which grammatical forms. The benefit is a unified mechanism for dealing with irregularities (they are listed under the dictionary entry), but the amount of redundant information and the dictionary's complexity are even greater.

Although these approaches are very effective in analyzing grammatical categories, and ideally suited to generating paradigms, they do not attempt to identify the suffixes used in the analysis. Identifying these suffixes can not only simplify and substantially narrow down the dictionary and suffix list, but also be of substantial linguistic value, which will be discussed below. This paper presents an Item-and-Process approach to extracting the suffix which marks a Polish morphological form, and of representing it independently of its graphemic surface form. In section 2, I describe the phonological analysis of orthographic strings in Polish. Section 3 presents an algorithm for the morphological analysis of the resulting phoneme arrays. The last section discusses benefits and applications of this approach and of the study of the suffixes it identifies.

2 From Orthography to Phonology

Given a tokenized input text, the first step of analysis is extracting a phonological representation. While Polish orthography does represent the phonetics of the language, extracting phonemes from it is nontrivial. This is however necessary in order to create a successful algorithm for morphological analysis based on relatively few rules. In the best case, a Polish orthographic word is composed of a string of characters, each of which represents one phoneme (1). In other cases two letters can stand for one phoneme, i.e. a digraph (2):

(1) <tak> ↔ /t/;/a/;/k/

(2) <**cz**as> ↔ /**cz**/;/a/;/s/

There are however more complicated cases. Most notably, the letter <i> can either stand for a vowel, in which case it represents an allophone of /y/ (the choice between <y> and <i> depends on the preceding phoneme[4]), or it can merely

[3] See e.g. Beesley and Karttunen (2003) for applications to various languages. Item-and-Process models (cf. Hockett, 1954) derive different surface forms from an underlying base form using rules, as opposed to Item-and-Arrangement models, which list all variants of the morphemes comprising a word, and Word-and-Paradigm models, which associate base forms with inflectional types. For a discussion of the different models, see Matthews (1991).

[4] This analysis defines two variants of several consonants as different phonemes, e.g. palatalized and non-palatalized labials to account for otherwise

Code	Chars	Vowel	Voiced	Manner	Place	Softness	R1	R2	R3	R4
ć;	ć	1	1	2	3	2	-t	-t	0	0
t;	t	1	1	1	2	1	+ć	+ć	+c	0

Table 1: Phonemes

mark the previous consonant as palatalized, or it may do both:

(3) <i> ↔ /y/ ↔ [i] (vowel)

(4) <nie> ↔ /ń/;/e/ ↔ [ɲɛ] ('i' marks the 'n' as palatal)

(5) <ci> ↔ /ć/;/y/ ↔ [ʨi] (marks palatality and a vowel)

This means <i> can be part of a digraph, or even a trigraph: <dzie> ↔ /dź/;/e/.

Another complication comes from the fact that certain consonant clusters in Polish behave as distinct units, exhibiting different phonotactic behavior from their constituents. For example, the cluster /sł/ is palatalized in certain environments as one unit into the cluster /śl/, instead of the /ł/ being palatalized alone, without affecting the preceding /s/. Such clusters can mean that a chain of up to five characters will require its own phonemic analysis, e.g.: <ździa> ↔ /źdź/;/a/. Complex strings are therefore stored in a table, and are described in terms of their orthography and the underlying or 'encoded' phonological units[5]:

Chars	Code
cie	ć;e;
ździa	źdź;a;

Once the phonemes underlying a string have been established, the token receives an array of phonemes representing it. Each one of these phonemes is represented through a phoneme data-type, which holds the relevant phonological information, such as voicing, place and manner of articulation, as well as some properties relevant specifically to Polish (and to Slavic languages in general), such as 'softness' of consonants, and 'mutation classes' (labeled R1-R4, using the conventions in Swan (2002:24-26)[6]), that define which consonants can derive from which other consonants through morphophonemic mutation (see section 3). Phonemes are identified by codes independently of the way they are represented orthographically; thus <ci> and <yć> are both comprised of the same two phonemes: /ć/ and /y/, and these are given the codes ć; and y; (all codes end in a semicolon).

The phonological encoding follows the traditional scheme in Swan (2002), which has proven functionally adequate and simpler to implement than SPE-based standard feature analysis (Chomsky and Halle (1968) and developments thereof) or a feature geometry scheme (Clements (1985) and related work). Thus parameters like place and manner of articulation have several possible values, as illustrated in Table 1. The phoneme /ć/, for example, is stored as a non-voiced, non-vocalic, palatal (place=3) affricate (manner=2), with (softness=2) indicating that it is 'soft' (relevant for phonotactic behavior), and the R3-R4 values of 0, that it does not undergo these mutations. The symbol -t in R1-R2 indicates that it may be derived from the phoneme /t/ through R1 and R2 mutations. The phoneme /t/ (in the second row

minimal pairs such as <być> '*to be*' and <bić> '*to hit*'. By analyzing these as /b/;/y/;/ć/ versus /b'/;/y/;/ć/, the different phonemes are /b'/:/b/, while the vowel remains the same phoneme (for this analysis see e.g. Swan (2002:10-12)). The success of the algorithm presented in this paper supports this view's viability.

[5] It seems that less than 300 such strings are required to describe Polish orthography, and each of them describes only 2 units – there are no tri- or more 'phonemographs'.

[6] Diachronically, the mutations labeled R1-4 correspond largely to effects of the second Slavic palatalization (which occurs mostly before Proto-Slavic monophthongized diphthongs), the first Slavic palatalization (which occurs before Proto-Slavic front vowels), palatalization of consonants followed by Proto-Slavic *\/j/, and the Polish softening of velars before /e/ and /y/, respectively.

of the table), conversely, shows a parallel value +ć, indicating that it may produce that phoneme under R1 or R2 mutation. This means that possible mutations are encoded already at the level of phonological analysis[7].

It is important to note that this representation scheme is morphophonological and not phonological. This means, for instance, that the vowel spelled <ó>, which is pronounced [u], is not identical to the vowel spelled <u>, which is pronounced in the same way. This is because the morphophoneme /ó/ exhibits a realization <o> (phonetic [o]) in certain environments, whereas /u/ does not. The result is two distinct phonemes, with identical phonetic features, but different morphophonemic features (i.e. the fields describing mutations)[8].

Beyond the phonemes we have already encountered, there are also some phonemes which have no direct orthographic representation, e.g. the palatalized variants of certain consonants already mentioned above, such as b', w', p', k' etc. These are only represented within longer strings (e.g. <bie> ↔ /b'/;/e/). Another symbol which has no phonetic representation is the token border sign '#', which is added before and after all tokens for analysis, and removed before lemmatization. This makes it possible to define a 'zero-suffix': /#/ = "stem only, no ending at all", and also to condition mutation rules based on word initial or word final position (see next section).

Finally, the mutation operators R1-R4 may or may not be seen as phonemes in the synchronic sense; they represent morphophonemic sound changes which can be motivated by historical processes. For instance, the sequence <ce> can be motivated by the change of an underlying /k/ which sometimes occurs before a vowel /e/. A 'different' vowel /e/ may change /k/ into /cz/ producing <cze>. Swan (2002:23-24) defines 5 vowels /e/ with different symbols for this purpose, as well as several variants of /y/ and some 'null' phonemes. Examples of the two changes above illustrate his notation[9]:

(6) <rę**ce**> (loc. sg. of rę**k**a '*hand'*) ↔ ręk + ě$_1$

(7) <krzy**cz**eć> (imperfective '*to shout*', perfective krzy**k**nąć) ↔ krzyk + ě$_2$ć

It has been found more computationally economical here to define 'pseudo-phonemes' to represent the possible mutations, which repeat regardless of which vowel (if any) is involved:

(8) <ręce> ↔ r;ę;k; + R1;e;

(9) <krzyczeć> ↔ k;rz;y;k; + R2;e;ć;

One may therefore consider /R1e/, /R1y/ etc. to be single, indivisible morphophonemes (as in Swan's notation), or accept /R1/ etc. as separate morphophonemes whose existence is reflected only in the mutations which they cause.

3 Morphophonemic Analysis

Before describing the process of analysis, the definition of a morphological suffix must be discussed. The most straightforward definition would seem to be that the stem contains that part of a word form which is common to all word forms derived from the same lemma, and the suffix contains the remaining characters [10] .

[7] This is however completely equivalent to defining underspecified morphophonemes and rules to determine their realization (cf. Beesley and Karttunen (2003:162-167)).

[8] A similar distinction could be made between German /e/ and /ä/. The form /gäste/, for instance, implies a possible form /gast/, but /feste/ does not imply */fast/. Marking both vowels as /e/ would be discarding information.

[9] Calling these 'different /e/'s' is not untenable, at least from the historical point of view. In these examples the first /e/ derives from an old diphthong, the ending *-āi of the locative singular feminine, while the second /e/ derives from a long 'e' in the infinitive ending *-ēti.

[10] This definition doesn't follow the traditional notion of 'suffix' or 'ending' in Indo-European linguistics. We may consider 'ł' in <mógł>, '*(he) could*', a suffix of the preterit form, although historically it is a derivational suffix of the perfect participle, followed by the case ending, nom. sg. masc. -Ø < -ŭ < *-os. Synchronically it is possible to defend such suffixes, especially considering it is likely many Indo-European suffixes and endings had comparable fusional origins.

Suffix	Case	Number	Gender	Person	Tense	Aspect	Base	Type	Conditions
R1e#	6	1	F				a#	S	
ł#		1	M	3	1		ć#	VFin	vowel=1

Table 2: Suffixes

However, with the adoption of phonemes as the basic unit rather than characters, certain divisions become impossible: e.g. pis-ać '*to write*' and pis-ał '*(he) was writing*' are possible, but pis-ze '*(he) writes*' is impossible, since <sz> represents a single phoneme. But a stem 'pi-', which would also be common to, for instance, 'pi-ć' '*to drink*', and worse a suffix '-sać', need not be resorted to if we use a multi-level generative model and consider the form <pisze> to be derived from an underlying /#;p';y;s;R3;e;#;/, so that the stem could still end in 's-' and the suffix would be /R3e#/. This 'abstracted' suffix[11], independent of its surface form, contains the representation of a mutation which occurs in many similarly conjugated verbs, where it creates a variety of orthographically and phonetically distinct forms. Such an analysis has many advantages: it has morphophonological explanatory power, it unites similarly inflected words with identical suffixes, it can identify productive use of a suffix producing a previously unencountered string, and it also eliminates the need for representing multiple stems within a dictionary entry (barring the few cases of suppletion).

In order to reach this abstract suffix an algorithm must identify and reverse a possible mutation at the stem-suffix border. Once the phonemes have been abstracted from the orthographic string, still possibly in mutated form, every possible border between phonemes is considered for creating a stem-suffix pair. The contact point between the two is then compared to a rule table describing possible phonotactic changes, which lists what kinds of phoneme sequences (in terms of phonological features) result from contact between what kinds of morphophonemes[12].

The following example illustrates how these rules operate: the phoneme array /#ręce#/ contains 6 phonemes, including the start and end of token symbols. One of its segmentations is /#ręc-e#/. The following rule states that a consonant (vowel=1) with a negative (i.e. derived) R1 value followed by a front vowel (softness=6; the softness parameter doubles as a front/mid/back parameter for vowels) and the token end sign (#), may result from contact between its positive (i.e. primary) R1 counterpart on the left, and the morphophoneme R1, followed by the same front vowel on the right (identified by co-indexing):

Left	Right	Result
R1=+, vowel=1, index=1;	R1; softness=6,index=2; #;	R1=-,vowel=1,index=1; softness=6,index=2;#;

A more legible notation for the same rule would be:

$$C_{[+R1]} + R1V_{[+front]}\# > C_{[-R1]}V_{[+front]}\#$$

Since /c/ is the negative R1 counterpart of /k/ and /e/ is a front vowel (this information was retrieved from the phoneme table during phoneme extraction), a possible analysis is created with the stem /#ręk/ and a suffix /R1e#/. This suffix can now be looked up in a suffix table, which contains the entries in Table 2.

The first entry suggests that the form is a locative (case=6) singular feminine substantive (type=S), and that the lemma may be found by adding the base suffix /a#/ to the stem. The resulting lemma /#ręk-a#/ can then be converted into a string using the phoneme table (note this is still a phoneme array) and looked up in the dictionary. With the lemma verified, an analysis can be created with inflectional information from the table, including the suffix and base-suffix used in the analysis.

[11] I avoid the term morpheme, since such a suffix may contain multiple morphemes.

[12] Finite-state rules often describe symbol to symbol correspondences (see e.g. Beesley and Karttunen (2003:133)). However the analogous behavior of many Polish phonemes makes rules defined in terms of phonological features more compact and easier to maintain (cf. Kaplan and Kay (1994:346-351) on feature notation for phonological rewrite rules).

In many cases, it is the reconstruction of the base form which will involve morphophonemic alternations, which means that the phonotactic table must be consulted at this stage too. Thus the form /#gryzł#/ '*(he) bit*' may be analyzed using the suffix /ł#/, with no morphophonemic alternations[13], using the 2nd row in Table 2.

This entry suggests that the suffix marks a 3rd person singular masculine preterit verb form, whose base form may be reached with the suffix /ć#/. Note that the 'Conditions' field specifies limitations on the structure of the stem to which the suffix is attached, in the form of literal phoneme codes or phoneme property arrays, in this case stipulating that it must end with a consonant (consonant stems take the unmediated infinitive suffix /ć#/). Since this is the case here (the stem /#gryz-/ ends with the consonant phoneme /z/), the algorithm consults the phonotactic table and finds the following rule:

Left	Right	Result

manner=3,softness=1,
place=2,R1=+,
index=1;
	ć;#;	manner=3,softness=2,
		place=3,R1=-
		,index=1;ć;#;

On the left side is a hard (softness=1) dental (place=2) sibilant (manner=3), while on the right the literal phoneme /ć/ is followed by the end of token sign. The 'Result' field describes the same elements, with the R1 value of the sibilant changed from + to -, place of articulation from dental to palatal and softness from hard to soft, in this case expressing a change from /z/ to /ź/, which yields the projected lemma 'gryźć' for lookup. Put another way:

$$C_{\begin{bmatrix} +hard \\ +dental \\ +sibilant \\ +R1 \end{bmatrix}} + \acute{c}\,\# > C_{\begin{bmatrix} +soft \\ +palatal \\ +sibilant \\ -R1 \end{bmatrix}} \acute{c}\,\#$$

$$\rightarrow z + \acute{c}\,\# > \acute{z}\acute{c}\,\#$$

Phonemes that are transformed by phonotactic rules must be identified both in the 'Result' field and in the 'Left' or 'Right' field, and both appearances are linked by co-indexing (the 'index' property). Other elements may only appear on one side of the equation, in which case they are not indexed. An example of this are rules describing vocalic syncope, the deletion of a vowel as a result of syllabic structure. The word <dworzec> '*station*', for instance, has the dative plural <dworcom>. The /e/ that causes an R2 mutation in the nominative is absent in the dative. This rule recovers the base form:

Left	Right	Result

vowel=1,index=1,
R2=-;
e;
vowel=1,index=2;
	vowel=2,
	index=3;
		vowel=1,index=1,
		R2=+;
		vowel=1,index=2;
		vowel=2,index=3;

The phoneme /e/ on the left side is absent from the 'Result' field, meaning that adding a vowel to the CeC structure in 'Left' can result in deletion of the /e/, and depalatalization of the first consonant (R2: - > +). Put differently (subscripts mark co-indexing):

$$C_{1[-R2]}eC_2 + V_3 > C_{1[+R2]}C_2V_3$$

Also note that this time the end of token sign is absent, since the vowel isn't necessarily the end of the suffix – indeed here it is followed by /m#/. The part covered by the rule is in brackets here: /#dwo[**r**$_1$**c**$_2$**-** **o**$_3$]m#/. The suffix /om#/ is found in the suffix table with a base suffix /#/ (the 'zero' suffix). The reconstructed stem (containing the 'Left' field, marked in brackets) and base suffix are then: /#dwo[**rz**$_1$**ec**$_2$]- #/. This procedure allows the consistent definition of suffixes, so that /om#/ stands for the dative plural regardless of consequent stem mutations. The text-based alternative would be to define a suffix '-rcom' with a base suffix '-rzec', or even actually ignoring the digraph to define the surreal looking pair '-com' : '-zec'.

4 Applications

The algorithm discussed in this paper has been implemented as part of a tagging program called Polimorph (see figure 1 on the next page). Currently using a basic dictionary of less than 28,000 lemmas, a set of 45 phonotactic rules and some 1,600 suffix entries, the program finds the correct lemma (regardless of disambiguation) for

[13] This is actually realized by the same mechanism, using an 'empty' phonotactic rule, which matches any sequence of two phonemes.

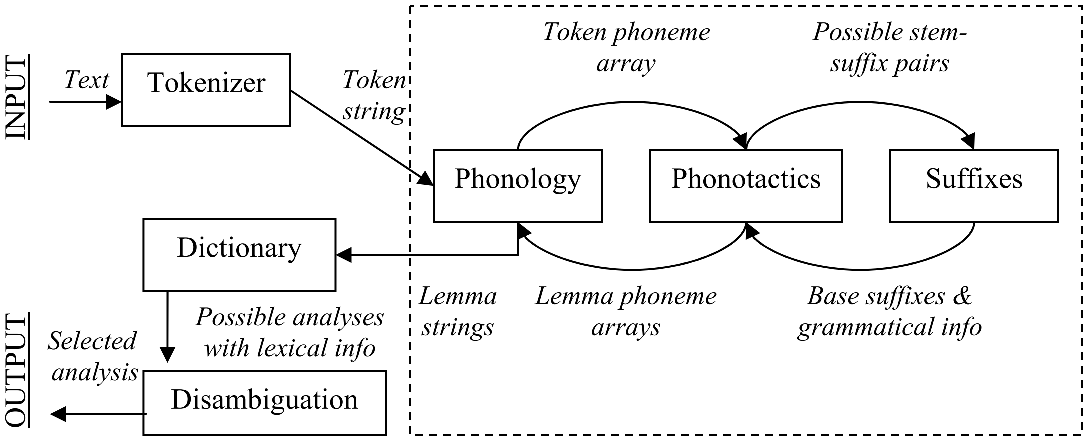

Figure 1: *Application logic of Polimorph. The algorithm discussed here is represented inside the dashed box.*

around 95% of tokens in a running Polish literary text (excluding punctuation). Almost all failures in analysis result from lemmas missing in the dictionary (especially proper names, foreign words), rather than inflectional irregularities, which are handled separately.

The algorithm is a computationally more complex, but lexicographically more compact alternative to text-based morphological analysis techniques currently in use for Polish. Its advantages encompass three domains: recognition power, lexicon structure and morphological informativity. Firstly, by avoiding explicit phonemes where possible, in favor of phonological features, it applies a small set of rules to mutations in all areas of morphology (the same phenomenon occurring in verbal or nominal flexion or derivation is handled by the same rule, which is ignorant of morphological signification). This circumvents problems arising from productive mutations that may not be documented in a suffix list.

Secondly, since the algorithm can test many rules before reaching a lemma, the dictionary doesn't have to include variant stems (genitive forms, 1st and 2nd person singular for verbs, etc.) – most of these can be arrived at through some mutation, the single base form of which the algorithm will compute and verify in the dictionary. This also solves the problem of non-standard analogical use of suffixes other than those listed for a lemma in the dictionary (e.g. both <biolodzy> and <biologowie> are recognized as plural of <biolog> '*biologist*', with different suffixes), and simplifies the structure, maintenance and expandability of the dictionary.

Finally, if suffixes are used as fields in corpora, this analysis makes various morphological investigations possible. Homographic (but morphophonologically distinct) suffixes can be distinguished and searched for in a corpus, e.g.: the suffixes /R1y#/ and /R4y#/, both of which can signify nominative plural masculine, and both of which may be manifested as either <i> or <y>: <chłopi> '*farmers*' and <biolodzy> '*biologists*' both exhibit the former, while <chłopy> '*lads*' and <ptaki> '*birds*' exhibit the latter. Different but homographic derivational types may be distinguished, for example the verb <siać> '*to sow*' has the suffix /R2ać#/, but most verbs exhibiting the same orthographic suffix are imperfective verbs derived from perfective verbs with the suffix /R3ać#/, like <wypuszczać> '*to let out*', derived from the perfective <wypuścić> (using the same stem with the suffix /R2yć#/).

This data is also useful for historical corpora, where changes in the distribution of suffixes can be explored through suffix based queries. For instance, in earlier texts one usually finds the old masculine accusative plural in /R4y#/, but in Middle Polish there are also cases of the modern plural genitive-accusative in /ów#/. It is also easy to define suffixes which are now obsolete for the analysis of older texts, especially as this does not entail creating the entire list of their possible orthographic representations, a resource which is unavailable for older language stages. For example, the suffix /R4em#/ is used for the neuter instrumental and locative pronouns and adjectives in some older texts (e.g. <dobrem> for modern <dobrym>), and there is no need for multiple entries for alternations in stems.

A weakness of the algorithm is that it relies on a division of each token into exactly two parts. This means derivational morphology beneath an inflectional suffix is not covered, which creates some redundancy. For instance, the comparative adjective is derived from an adjective stem plus a comparative formant, followed by adjective endings, e.g.: <długi> '*long*' > <dłuższy> '*longer*' ↔ /#dług/ + /R2sz/ + /R4y#/. To analyze this form the suffix table must contain entries merging these morphemes: nom. /R2szy#/, gen. /R2szego#/… etc. Such repetitions, caused by a compounding of derivational and inflectional suffixes, are a main reason for the still not negligible size of the suffix table. A direction for future study is to define multi-segmental suffixes, which would allow a very significant further reduction in suffix table size, as well as more accurate coverage of derivational morphology. Implementation of multiple segments can already be found in the analysis of Czech morphology in Sedláček and Smrž (2001), where it is however applied on an orthographic level.

Another problem is dealing with non-suffixal morphology, most notably the superlative prefix 'naj-', added to the comparative form, although productive use of the negative prefix 'nie-' offers a similar challenge. At present these elements are explicitly checked for in the event that no lemma can be found (cf. Szafran (1997) for a similar solution, and likewise for the Czech equivalents Sedláček and Smrž (2001)).

References

Beesley K.R. and Karttunen L. (2003) *Finite State Morphology*. CSLI Publications, Stanford, California.

Bień J. and Szafran K. (2001) *Analiza morfologiczna języka polskiego w praktyce*. Bulletin de la société polonaise de linguistique, fasc. LVII, pp. 171-184.

Chomsky N. and Halle M. (1968) *The Sound Pattern of English*. Harper and Row, New York.

Clements G.N. (1985) *The Geometry of Phonological Features*. Phonology Yearbook, 2, pp. 225-252.

Hockett C.F. (1954) *Two Models of Grammatical Description*. Word, 10, pp. 210-231.

Kaplan R.M. and Kay M. (1994) *Regular Models of Phonological Rule Systems*. Computational Linguistics, Computational Linguistics, 20/3, pp. 331-378.

Matthews P.H. (1991) *Morphology, Second Edition*, Cambridge University Press, Cambridge, chapters 6-10.

Osolsobě K. (1997) *Formale Beschreibung der tschechischen Morphologie*. In "Formale Slavistik", U. Junghanns and G. Zybatow, eds., Vervuert Verlag, Frankfurt am Main, pp. 443-451.

Osolsobě K. et al. (2002) *A Procedure for Word Derivational Processes Concerning Lexicon Extension in Highly Inflected Languages*. In "Proceedings of the Third International Conference on Language Resources and Evaluation, LREC", ELRA, Las Palmas de Gran Canaria, pp. 1254-1259.

Przepiórkowski A. (2004) *The IPI PAN Corpus, Preliminary Version*. Institute of Computer Science PAS, Warsaw.

Sedláček R. and Smrž P. (2001) *Automatic Processing of Czech Inflectional and Derivative Morphology*, FI MU Report Series, Brno.

Šipka D. and Končar N. (1997) *Minimal Information Grammar (MIG), Serbo-Croatian and Polish Morphological Paradigms*. In "Formale Slavistik", U. Junghanns and G. Zybatow, eds., Vervuert Verlag, Frankfurt am Main, pp. 427-436.

Swan O.E. (2002) *A Grammar of Contemporary Polish*. Slavica Publishers, Bloomington, Indiana.

Szafran K. (1997) *Automatic Lemmatisation of Texts in Polish – Is it Possibile?* In "Formale Slavistik", U. Junghanns and G. Zybatow, eds., Vervuert Verlag, Frankfurt am Main, pp. 437-441.

Tokarski J. (1993) *Schematyczny indeks* a tergo *polskich form wyrazowych*, Z. Saloni, ed., Wydawnictwo Naukowe PWN, Warszawa.