A Dictionary based Stemming Mechanism for Polish

Michał Korzycki

AGH University of Science and Technology, Department of Computer Science,
Al. Mickiewicza 30, Kraków, Poland

Abstract. In this paper we present and evaluate a robust stemming mechanism for Polish. We use the Polish Inflection Dictionary to build a Rule Based Stemmer and a Generative Reversed Rule Stemmer. The combination of both stemmers in the shape of the described Hybrid Stemmer provides us with a high precision stemming mechanism that is able to match human performance. This assumption is supported by a conducted experiment, the results of which are presented.

1 Introduction

Human linguistic skills are clearly composed of potential abilities ie. the ability to deal with words, forms and expressions that have not been encountered before. That is directly related to the nature of the language - an ever changing entity, conveying new information, often using new, unknown means - such as new words and expressions. Computer systems that are dealing with language should try to replicate this behavior in order to be of use in any serious application. This paper describes the process and results of building a *stemmer* (a mechanism for generating a base form for a word form found in text) that also is able to assign some grammatical categories to the found form. The process of creating the stemmer is automatic - its rules are automatically extracted from the Polish Inflection Dictionary, thus are a direct result of analyzing the language itself and are not biased by some prior grammatical preconceptions. That permits us to postulate that the resulting mechanism is able to recreate closely something that can be called a natural grammar - linguistic knowledge coming directly from observation and not from formal grammar definitions.

2 Related Work

The presented work bases on the notion of generative grammar introduced by N.Chomsky [1] that have been expanded into the two-level morphology formalism of K.Koskenniemi [2]. All those formal approaches have been crucial in the creation of the Polish Inflection Dictionary [3] on which we build upon. The detailed description of this dictionary and its creation approach can be found in Lubaszewski et al. [4].

There are already some stemming solutions available for Polish - such as the commercial solutions Gram from Neurosoft or PoMor from MorphoLogic. Among the free and open stemmers, one must count Stempel created by A.Białecki and L. Galambos, the Lametyzator and Stempelator by D. Weiss [6] and the classic SAM [7]. A stemmer

Korzycki M..
A Dictionary based Stemming Mechanism for Polish.
DOI: 10.5220/0004100301430150
In *Proceedings of the 9th International Workshop on Natural Language Processing and Cognitive Science* (NLPCS-2012), pages 143-150
ISBN: 978-989-8565-16-7

acts often as the replacement of a dictionary, so a stemmer comparison is often a comparison of the quality of the underlying dictionaries. Creating a comparison metric that would be able to evaluate the quality of the stemming mechanism in isolation from the underlying dictionary has been suggested in [8] and is beyond the scope of this paper.

3 Requirements for a Robust Stemmer

The stemmer that will be presented should be able to mimic human skills as closely as possible. If we were to list those skills, we would point to the following issues:

- Being able to discern exceptions from general rules. The language is an entity that has been created through an evolutionary process, resulting in many different competing layers and grammars. This often results in some vestigial grammars that describe behaviors different to the rest of the language.
- Correct behavior on known words. We use the Polish Language Dictionary as the base for our linguistic knowledge. We need the stemmer to be fully compliant with that dictionary.
- For words that are not found in the mentioned dictionary, we need the stemmer to be able to correctly stem and identify their part of speech.

4 Polish Word Representation

Polish is a highly inflected language. Each primary word has a number of inflectional forms: verbs have 47 (if we exclude participles), adjectives 44, numerals up to 49, nouns and pronouns 14, and adverbs 3. These figures, and the fact that many words have irregular stem alternations, show that Polish inflection presents real problems for the computational linguist [5]. If we ask how to inflect properly the Polish word, eg. personal masculine noun *aktor* ('actor'):

	Singular	**Plural**
Nom.	aktor-0	aktorz-y
Gen.	aktor-a	aktor-ów
Dat.	aktor-em	aktor-ami
Acc.	aktor-a	aktor-ów
Instr.	aktor-em	aktor-ami
Loc.	aktorz-e	aktor-ach
Voc.	aktorz-e	aktorz-y

The grammarian's answer is that one must first learn Polish lexical grammar and then apply that grammar to particular lexical items But, in fact, if one wants to inflect a particular word properly, one must first select the proper inflection ending, eg. *-0, -a, -o, -e, -e, -i* or *-y* to form Masc. Pers. Nom. Sing., *-a, -e,* or *-ego* to form Gen. Sing., *-owi, -u,* or *-emu* to form Dat. Sing., etc. and then must apply the proper stem alternation rule. As we can see, the inflectional stem of the word *aktor* changes from *aktor-* to *aktorz-* before the ending *-e* and *-y*. which is the result of the palatalization process. Let's

compare behaviour of the final stem consonant before ending *-y*, which can occur in Nom. Sing. and Nom. Plur. of nouns, eg. *aktor-0 : aktorz-y*, *senior-0 : seniorz-y* ('older person') *amor -0: amor-y* ('cupid'), *gbur-0 : gbur-y* ('bumpkin'), *traktor-0 : traktor-y* and adjectives, eg. *któr-y : którz-y* ('which') and *stary : starz-y* ('old'). It is clear that the global phonological rule which says that a front vowel causes consonant palatalization is not appropriate; here, as in *aktorz-y, seniorz-y* the palatalization takes place before *-y* in Nom. Plur., but cf. the co-existence of *amor-y, gbur-y, traktor-y* and *któr-y*, *star-y* alongside *którz-y, starz-y* respectively Nom.Sing. and Plur. of adjective. This shows that there is a need for a new approach to the stem alternation process. The data show that the belief that it is possible to develop efficient stemming algorithm for Polish seems to be naïve one. We argue, that if one wants to create algorithm, which recognize a particular word properly, one must store all inflection forms in the dictionary - word by word.

5 The Polish Inflection Dictionary; its Lexical Grammar and Generative Mechanism

The Polish Inflection Dictionary [3] is the base that has been used to create the stemmer described in this article. The construction of the dictionary bases on over 420 identified lexical categories. Each is defined by its inflection patterns used to generate it. The first element of the dictionary is the set of rules that are used to assign a lexical category to a word basing on its ending.

Each inflection category pattern is represented by its specific local grammar, which consists of two elements: a vector of inflection endings associated with the category, and the proper local grammar rules, mainly related to stem alternation rules.

There are words in Polish, that in general behave according to a specific inflection pattern, but some of their forms do not match strictly the pattern (such as the words *handel* 'commerce' and *hotel* 'hotel' will have their corresponding genetive cases, respectively, *handlu* but *hot**e**lu*). Such cases are described by additional exception rules that describe over 11.000 such cases as mentioned above.

Although, the generative approach to build the Inflection Dictionary comes directly from the concept of two-level morphology [2], it cannot be used directly for word form recognition. The reason for that is that the dictionary generating mechanism has been augmented by additional filters - the last building block used for generating the Inflection Dictionary. Those mechanisms are rejecting forms that, formally, are correct, but the language itself has rejected them. Those rejected forms range from illegal adjectives comparative form (*bardziej chory* but not *chorszy* 'more ill') to plurale tantum forms (*spodnie* - 'trousers') that for some pragmatic reasons do not possess singular forms. Morphological relations are another problem that cannot be described in rule form. Those relations join different words, which share the same lexical meaning, eg. the imperfective, perfective and iterative form of verbs, *pisać* : *napisać* : *pisywać* 'to write', where one cannot specify the prefix to build the proper perfective form c.f. *od-pisać* 'to answer' *prze-pisać* 'to copy' *nad-pisać* 'to overwrite' and so on. In addition the presence of iterative forms depend on the meaning of a specific verb. It seems impossible

to determine the rules which guide those language selection mechanisms, so the filters had to be provided manually.

All this results in a dictionary of very high quality, but at the cost of not being able to reverse its generative mechanism for recognition. The dictionary consists of more then 120.000 lexical entries excluding proper names, with more than 3.300.000 inflection forms.

6 Building a Stemmer by Extracting Rules from the Dictionary

The stemmer described in this article is composed of two elements. The first part has been automatically generated from the observation of the forms occurring in the inflection dictionary. That approach gives it a large amount of flexibility, as it can be used on any observable linguistic data. We will refer to it further in the text as the rule-based stemmer. The second is based on reversing the generation rules of the Inflection Dictionary. As mentioned above, such reversion is imperfect and can leads to erroneous results, but as it is applied only after the rule-based stemmer has failed to provide an answer, that imperfections (mainly multiple potential results) can be accepted.

As the first stage to create the rule-based stemmer, we need to identify homographs. Homographs are a major issue to consider while trying to identify words in a text. Their occurrence makes it difficult to discern between words and their forms. In a highly inflected language, such as polish, homographs can be an incidental result of the rich inflection (as the word mamy that can be both the form of a verb *mieć* 'to have' or a noun *mama* 'mom/mother'). But they can also be the result of a much deeper phenomenon, such as the inability to distinguish in Polish between the genitive and accusative case of personal nouns. It can also be the result of a common etymology of different words-such as *palący* from *palić* 'to smoke' can be a form of a participle, an adjective and a noun.

The mechanism described in this paper should be able to assign proper lexical values to words found in texts based on the the values available in the Inflection Dictionary, its content is transformed, in order to cope with the problem of homographs as described above. We introduce the notion of meta-tags that describe homographs. So, the meta-tag `ADCAAA-2-3-6-7-9-BDC-26` designates the set of the 2nd, 3rd, 6th, 7th and 9th forms of the lexical group `ADCAAA` and the 26th form of the lexical group `BDC`. This is an example of a proper meta-tag that is a container for over 3.000 participles, and as such represents a real linguistic phenomenon and not only an random event that should be discarded as an exception.

In order to extract grammar rules from the Inflection Dictionary, we build a prefix tree (*trie*) out of its entries but represented in a reversed order from right to left. The leaves of that tree are the meta-tags of the represented forms.

This trie is searched from top to bottom, in order to find the nodes, such that all leaves below that node that share the same meta-tag (grammatical description). With such a node we can identify a key - that is the string that can be constructed by going from the root of the trie to this node and its value - the unique meta-tag of the leaves below that node. Such key,value pair can represented in the form of `-onoma => AAAAAA-2-4` - something that will be described further as a *rule*.

The rule has a straightforward interpretation - it signifies that in the Inflection Dictionary all forms ending with `-onoma` were of the lexical group described as `AAAAAA` and on the 2nd or 4th position if its form vector. Unknown words with such an ending will also be identified as belonging to this lexical group.

It is important to be able to discern exceptions from general rules, as some words, often representing a vestigial grammar, should not influence our ability to recognize new words. That decision comes from the observation that new words appearing tend to be have a much more regular inflection in general. The distinction between exceptions and rules comes directly from a set of simple observations made on the trie described above. First we identify as 0-level exceptions the words in the dictionary that belong to categories that are not inflected (all their forms are identical). 21.331 such words have been found. Next, we identify as 1st level exceptions those words that belong to categories that have rules containing on their keys only full words - their rules are just word dictionaries - they have no discerning power for unknown words. 3.882 such words have been found. After that, the trie is rebuilt after rejecting level 0 and 1 exceptions. After extracting the rules from the trie, some of the rules again contain full words as keys. And some categories have only full word keys. Those keys (words) are identified as 2nd level exceptions and listed separately. There are 9.947 such rules (exceptions) identified. They contain usually categories of rare uninflected nouns such *husky*, *collie* etc. Finally, we list as exceptions (not rules) those keys that are identical to the forms. These are usually very short words, so short, that they are not much different in length from typical lexical endings. In this category we have words like *efeb* 'ephebe', that found themselves in the same category group as keys *-ozof* or *-ligraf* that lead all to the lexical label `AAAAAA-1`. That group of 3rd level exceptions is the largest one at 156.970 elements, as it contain short words that are quite numerous in the language itself. These operations lead to the following example rules after discarding the exceptions from the trie:

```
-achówki ADAB-2-8-11-14 -achówka
-achtem ACAAAAA-5 -acht
```

Where the first column corresponds to the inflected ending of the stemmed form to be matched, the second describes the identified lexical value (the inflection category/ies and the corresponding indexes on their form vectors), the third depicts to the ending of the base which has to replace the inflected ending in column 1 in order to obtain the searched base form.

The process of creation of the stemmer guarantees us one important property - if a rule matches the analyzed form, it will be the only rule matching and the result will be unambiguous.

7 The Rule-based Stemming Mechanism

The application of the rule-based stemmer to a word is a two step process:

– first we check if that word is an exception. If it is found on the exception list - we return the lexical value associated with that exception.

– If it is not an exception, we find the key of a rule that matches the ending of that word. As the keys are extracted from the original dictionary trie as to determine unambiguously the lexical category - the keys are mutually exclusive and do not include each other.

8 The Reversed Generating Rule-based Stemmer

As described above, the Inflection Dictionary has been generated by two set of rules. The first set is the set of category recognition based on word ending (as in `-iwiec => AAACBA`). The second one is a set of stem alternation rules. After applying those two rules, a vector of inflected endings is applied to generate the form vector for the presented word.

The Inflection Dictionary generation rules are used to create the mechanism of the reversed generating rules stemmer. The set of lexical endings from each specific category is prepended with the alternation rules for that category (or taken as such without them). That leads to a set rules that can create many false positives, as in the generative case, the alternation mechanism was optional, ie. was used only if it was applicable to a specific stem. Here we must consider the potential applicability of alternations in each cases - leading to superfluous answers.

The generated recognition rules coming from the generative rules described in the part regarding the generation of the Inflection Dictionary lead to the following example rules:

```
ACACBC awiec awca
ACACBC ec cowi
```

Where the first column corresponds to the lexical category, the second to the ending of the base from, the third column to the inflected ending that has to be removed and replaced by the 2nd column value in order to obtain the searched base form.

9 The Hybrid Stemmer

Combining the two stemmers (the rule-based and the reversed generating rule) by using them sequentially gives us a solution that combines the power of both approaches:

– by using the rule based stemmer first we obtain a mechanism that works perfectly on all known words, and generates high precision unambiguous results for recognized unknown words.
– if the former approach fails (does not provide a result - the form has not been matched by any rule), a more "fuzzy" mechanism based on the reversed generative rules is applied. As the analyzed word has not been matched in the first step, it can be seen as "hard" and can be interpreted in an ambiguous way. Additional tools working on a wider scope that a single form (such as a contextual morphosynctatic disambiguator) can be required for further processing. But issues dealing with more than one form are beyond the scope of this paper.

10 Benchmarking the Hybrid Stemmer

The goal set for the created mechanism is to be able to mimic as closely as possible human behavior in word recognition, so the proper benchmark for its efficiency should be based on human evaluation. The task of stemming words has been presented aThe survey was composed of words to be stemmed in 5 categories:

- Category A - 10 "hard" word forms from the dictionary - those having homographs both incidental (*damy* from *dać* 'to give' and *dama* 'damme') and systemic (participles versus deverbative nouns such as *palący* 'smoking' and 'smoker')
- Category B - 11 word forms not in the Inflection Dictionary, but recognized by the rule based stemmer - such as *edynburskim* from *edynburski* ('from Edinburgh')
- Category C - 10 word forms not in the Inflection Dictionary, but correctly recognized by the hybrid based stemmer, but not by the rule-based stemmer - words like *handlu* from *handel* ('commerce')
- Category D - 6 word forms not found in the Inflection Dictionary, but correctly recognized by the hybrid based stemmer, but not by the rule-based stemmer. Those words have been recognized but not unambiguously. Those words included such forms as *krasnoarmiejców* from *krasnoarmiejec* (adopted from russian: 'soldier of the Red Army').
- Category E - 3 words unrecognized by the stemmer like *jazzmenów* from *jazzman*

The score has been evaluated in each category separately, with a value of 50% given for an partially (ambiguous or incomplete) correct answer.

Table 1. Survey results overview and comparison with the Hybrid Stemmer performance.

Category	Hybrid Stemmer	Survey Results			
		Average	Median	4^{th} Quartile	95^{th} Percentile
A	100.00%	50.80%	46.60%	54.10%	83.30%
B	100.00%	60.41%	63.64%	70.45%	78.64%
C	100.00%	90.00%	90.00%	100.00%	100.00%
D	50.00%	75.56%	83.33%	83.33%	85.00%
E	0.00%	38.60%	33.33%	66.66%	66.66%
Average	85.00%	65.89%	65.63%	74.15%	83.91%

To give an estimate on the efficiency of the described mechanism, we will provide a general performance comparison to A.Weiss Stempelator - a very popular stemmer solution that is probably the closest in terms of general architecture to the Hybrid Stemmer described above. The Stempelator [6] is composed similarly of two parts - the Lematyzator that extracts base forms from the ispell dictionary for Polish and Stempel - a rule based stemmer for forms not found by the Lematyzator. As the comparison should be between the stemming mechanisms, we will compare the results obtained by Stempel on the survey above, but restricting it only to categories B to E - to compare elements that do not belong to the dictionary from the point of view of both stemmers. The results presented below should be regarded more as a comparison of the performance of both stemmers in reference to human performance rather than a relative comparison between them.

Table 2. Survey results with Hybrid Stemmer and *Stempelator* performance.

Category	Hybrid Stemmer	Stempelator Performance	Survey Median
B	100.00%	63.64%	63.64%
C	100.00%	50.00%	90.00%
D	50.00%	33.33%	83.33%
E	0.00%	33.33%	33.33%
Average	80.00%	46.67%	66.66%

11 Conclusions

As can be seen, especially in difficult cases, the described stemmer performance excels typical human skills (actually, only one participant of the review scored more than the described mechanism). Such result proves that we were able to create a mechanism that mimics human behavior in word recognition, extracts its data from the language itself and does not overgeneralize its rules, thanks to its ability to discern between exceptions and generic rules.

References

1. Chomsky, N.: Aspects of the Theory of Syntax, MIT Press, (1965)
2. Koskenniemi, K.: Two-level Morphology - A general Computational Model for Word-Form Recognition and Production, University of Helsinki Publication No. 11 (1983)
3. Lubaszewski, W., Wróbel, H., Gajęcki, M., Moskal, B., Orzechowska, A., Pietras, P., Pisarek, P., Rokicka, T.: Słownik Fleksyjny języka polskiego, Lexis Nexis, Kraków (2001)
4. Lubaszewski, W. (ed.): Słowniki komputerowe i automatyczna ekstrakcja informacji z tekstu, Kraków, AGH Press, (2009), original text in Polish
5. Lubaszewski, W.: A Grammar for the Polish Inflection Lexicon TASK Quarterly : scientific bulletin of Academic Computer Centre in Gdansk ; ISSN 1428-6394 - (2000) vol. 4 no. 2 s.291-300. - Abstr.
6. Weiss, D.: Stempelator: A Hybrid Stemmer for the Polish Language. Technical Report RA-002/05, Institute of Computing Science, Poznań University of Technology, Poland, (2005).
7. Weiss, D.: A survey of freely available polish stemmers and evaluation of their applicability in information retrieval. In: Human Language Technologies as a Challenge for Computer Science and Linguistics, Proceedings of the 2nd Language and Technology Conference, pages 216-221, Poznań, Poland, (2005).
8. Korzycki, M.: Transducer skończenie stanowy jako narzędzie rozpoznawania form tekstowych wyrazów [The Finite-State Transducer as a Tool for Polish Inflection Form Recognition], PhD Thesis, AGH (2008)