Študentská vedecká konferencia FMFI UK, Bratislava, 2010, pp. 295–300
ISBN 978-80-89186-68-6, © 2010 M. Malý

Semi-automatic creation of a stemming dictionary of an inflecting language using grammatical induction

Michal Malý[*]

Department of Applied Informatics, Faculty of Mathematics, Physics, and Informatics
Comenius University, Mlynská dolina, 842 48 Bratislava

Abstract: We propose a novel, language-independent method for stemming. Our method uses grammar induction to create a regular grammar (equivalent to a minimal deterministic final automaton) from the list of words, using Myhill-Nerode equivalence (which, up to our knowledge, has not been used for this purpose). The nonterminals in this grammar represent sets of suffixes. We assume that the most used sets of suffixes are those used to create inflections. The words which differ only in an „inflectional" suffix are supposed to be different variants of the same word and can be grouped together according to the same stem.

This method can be used for any inflecting language. We present the result of this method on Slovak language.

Keywords: stemming, grammatical induction, Slovak language, Myhill-Nerode equivalence

1 Introduction

Stemming is the process of identifying the stem of the word. For example, a word *„mesto"* („a city" in Slovak language) becomes *„mesta"* in genitive case – „(without) a city" – and *„meste"* in prepositional case – „(about) a city". That part of the word, to which affixes are attached in order to form a grammatical variant of the word, is called the stem, e. g. *„mest-"* and usually does not change across different grammar variants.

1.1 Stemming and its use

When a user wants to read an article which mentions the word „city", she can use a search engine to find such articles. However if the article is written in an inflecting language, e.g. in Slovak, searching the text for occurrences of string *mesto* yields only articles which mentions „a city" only in the nominative case, what is unsatisfying.

An advanced search engine can therefore search not only for occurrences of *mesto* but also for occurrences of *mesta*, *meste* and also other inflection variants.

To accomplish this, the search engine has to have a dictionary of variants for every possible word.

2 Previous work

Several methods for automatic stemming were proposed. Some methods are directly based on rules specific to the language, e.g. English ([Lovins, 1968],[Porter, 1997],), Bulgarian ([Nakov, 2003]) and other languages; or require an external rule set ([Paice, 1990],[Paice and Oakes, 1999]).

Existing language-independent approaches are based on n-gram analysis ([Järvelin et al., 2007],[Mcnamee and Mayfield, 2007]) and stochastic learning from training examples ([Goldsmith et al., 2000]).

3 Grammatical induction

Grammatical induction (also called grammar inference) is the process of creating a formal grammar which produces a given (formal) language.

3.1 Formal grammar

A formal grammar is a set of rules which describe how to form strings in a formal language. Usually a formal definition is given, which consist of: a finite set of non-terminal symbols, a finite set of terminal symbols (alphabet), a finite set of production rules, and a starting symbol; in the formal notation a quadruple (N, T, P, S). This formalization is often called a rewriting system.

3.1.1 Regular grammar

Regular grammar is a formal grammar where rules have the form $A \to b$, $A \to bC$ or $A \to \varepsilon$, where A, C

[*] maly@ii.fmph.uniba.sk

are non-terminal symbols, b is a terminal symbol. The ε means empty string.

In the Chomsky hierarchy of grammars, regular grammars are in a lowest position. They are strictly weaker than classes of grammars higher in this position.

4 Method

We propose a method to create a regular grammar from a list of the words belonging to a given natural language. For each non-terminal in this grammar we count the number of its uses when generating words. The terminals with a high count represent suffixes which are often used to form variations for many different stems. One may therefore assume that the suffixes common for many stems are those which characterize the inflection system of the language. Those words which derivation is different only in using those non-terminal symbols can be grouped together because they have a common stem.

4.1 Myhill-Nerode theorem

Myhill-Nerode theorem[Nerode, 1958] states that the relation R defined by

$$u\, R\, v \iff_{def} \forall x \in \Sigma^*(ux \in L \Leftrightarrow vx \in L)$$

is a relation of equivalence. The consequence is that the language L is regular if and only if the relation R is finite, and the states of a minimal deterministic final automaton (DFA) for language L can be constructed according to equivalence classes in R.

The use of Myhill-Nerode theorem for stemming purpose is not known to us. It is surprising, because this method is simple and relatively straightforward.

4.2 Algorithm

We consider the given list of words to be a finite (and therefore regular) language L. Then, the algorithm to implementing the relation from the Myhill-Nerode theorem and creating the grammar is quite straightforward.

The numeric threshold for the count is entered by hand. Then we can mark nonterminals beyond the threshold and group the words which derivations differ only by marked nonterminals. The algorithm is as follows:

1. Create a list of all prefixes of all words.
2. For each prefix, create a set of suffixes that can be attached to it so we get a word in the dictionary.
3. Prefixes with the same set of suffixes belong to the same equivalence class.
4. For each equivalence class, assign a nonterminal, for each equivalence class and a suffix assign a grammatical rule.
5. Each nonterminal, which contains only one rule, can be reduced in every its occurrence by this rule. This will reduce chains of nonterminals to fewer, more readable, string-like rules.
6. Count the number of uses of each nonterminal.
7. Mark the nonterminals with the count greater or equal to the given threshold.
8. Generate the words from the grammar using recursion. If you reach a marked terminal, create a new group and add there all words from all children calls.
9. Output the groups.

An implementation in C++ language is available under the GPL3 license from [Malý, 2010].

5 Results for the Slovak language

The Slovak language belongs to the family of Slavic languages. Similarly to other Slavic languages, it has a rich inflection: endings mark gender, number, case in declension of nouns, adjectives, pronouns and numerals; endings also mark person, number, tense, mood, and aspect in conjugation of verbs.

5.1 Preparation of the dictionary

We have obtained a raw dictionary of all words from the Slovak National Corpus [Jazykovedný ústav L'. Štúra SAV, 2009]. This dictionary lists each word together with the frequency distribution, automatically obtained from texts in corpus. The dictionary is not manually checked, contains misspellings, slang words, words from other languages and other artifacts like *„www"*.

It also does not have to contain every possible form of a word.

We have decided to consider only the words with the frequency greater or equal then 1000. We also filtered out tokens which contained a symbol which is not a letter (e.g. *„2x”*).

5.2 Sample of created grammar

We present an excerpt of resulting grammar related to the words beginning with *„mest”*, see Table 1. Numbers denote nonterminals, each nonterminal (except the starting nonterminal, 0) is assigned a count of uses. We present also a visualisation of this part of the minimal automaton, see Figure 1.

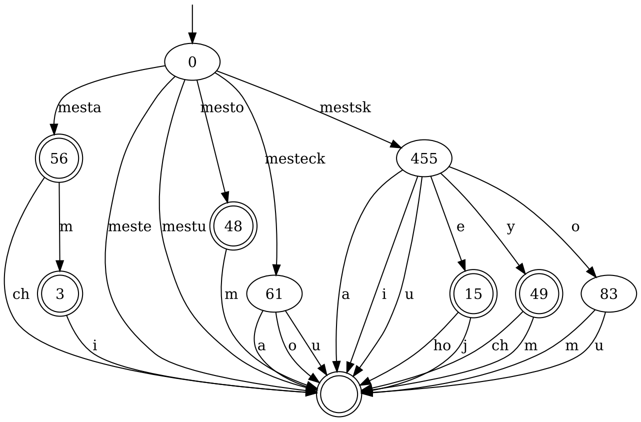

Figure 1: Part of the minimal automaton

5.3 Sample of the stemming dictionary

We present a sample of the stemming dictionary obtained by entering a threshold of 2.

- mestach mestami mestam mesta
- meste
- mestecka mestecko mestecku
- mestom mesto
- mestska mestskeho mestskej mestske mestski mestskom mestskou mestsku mestskych mestskym mestsky

5.4 Summary of the results for Slovak language

Results are of course not perfect. This is partly because inflection suffixes mix here with derivational suffixes: The words *„mesto”* („a city”), *„mestecko”* („little city”) and *„mestsky”* („related to city”) have a slightly different meaning; and each of them occurs in several of its grammatical forms.

Let’s consider an interpretation of the linguistic term „stem” so that one has to distinguish between the derivational and the inflection suffixes (so called light stemming). In this interpretation, the ideal situation would be to create three stem groups:

- mestach mestami mestam mesta meste mestom mesto
- mestecka mestecko mestecku
- mestska mestskeho mestskej mestske mestski mestskom mestskou mestsku mestskych mestskym mestsky

The first group represents different forms of the word *„mesto”*, the second is for *„mestecko”* („little city”) and the third represents *„mestsky”* („related to city”).

Our method failed to identify the first group and broke it up into three subgroups. It correctly identified the second and the third group.

5.5 Estimating the error rate

We have manually estimated the error rate according to the methodology described in [Paice, 1996]. This comprises the understemming index (UI) and the overstemming index (OI).

We have manually processed the first 100 lines from the resulting dictionary. Each line consists of words with the same stem. In the ideal case, one line represents corresponds with one real stem – all words having this particular stem belong in one group. In the understemming case, several lines have to be merged into one group. In the overstemming case, the line has to be split into several stem groups.

The formulas for UI and OI[1] are motivated by looking at every (unordered) pair of words in a group. Each two words can be evaluated as having the same stem, or as not having the same stem. We then evaluate the number of pairs missed (for UI), or incorrectly

[1] $OI(L)$ in Paice’s notation

Table 1: Excerpt of resulting grammar

rules for the nonterminal	#uses
0 $\rightarrow$ … | *mesta* 56 | *meste* | *mesteck* 61 | *mesto* 48 | *mestsk* 455 | *mestu* | …	(start)
3 $\rightarrow$ *i* | ε	111
15 $\rightarrow$ *ho* | *j* | ε	179
48 $\rightarrow$ *m* | ε	101
49 $\rightarrow$ *ch* | *m* | ε	111
56 $\rightarrow$ *ch* | *m* 3 | ε	121
61 $\rightarrow$ *a* | *o* | *u*	7
83 $\rightarrow$ *m* | *u*	35
455 $\rightarrow$ *a* | *e* 15 | *i* | *o* 83 | *u* | *y* 49	2

assigned (for OI) by the algorithm in each group, and sum these numbers for all groups. We also calculate the number of pairs in correct (manual) stemming.

$$UI = \frac{\sum\limits_{\text{stem group } g} \sum\limits_{\text{line } l \text{ in group } g} 0.5 n_l * (N_g - n_l)}{\sum\limits_{\text{stem group } g} 0.5 N_g (N_g - 1)}$$

where g represents a correct group, consisting of some line(s), N_g is the number of words in this group, and n_l is the number of words in the line l. The outer sums run for each stemming group, and inner sum runs for each line which belongs to one group. Only groups consisting of more than one word are considered.

$$OI = \frac{\sum\limits_{\text{line } l} \sum\limits_{\text{stem } s \text{ in line } l} 0.5 n_s * (N_l - n_s)}{\sum\limits_{\text{line } l} 0.5 N_l (N_l - 1)}$$

where l represents a line from the dictionary, consisting of (one or multiple) real stems, N_s is the number of words in this line, and n_s is the number of words for the stem s. The outer sums run for each line, and inner sum runs for each stem, which is present in that particular line. Only lines consisting of more than one word are considered.

5.5.1 Understemming

The UI for our method is 0.57. This is bigger compared to Paice's results for English stemmers (0.11–0.37 for three different stemmers and three different word sources[2]).

5.5.2 Overstemming

In the sample considered, we have seen only one case of overstemming. Therefore the OI is very close to zero (0.007), which is lower than in Paice's results (0.07–0.37).

5.6 Discussion

Our method is usable for light stemming. The understemming ratio is relatively high, the overstemming is almost non-existent. Our method is unable to join cases, where the stem itself changes in different grammatical forms, e.g. *mesto* (a city) – *miest* ((without) cities).

It is also worh noting that the Slovak language is much more complicated than English, and also that our method does not use any specific language rules (contrary to the stemmers used in Paice's test).

6 Expanding the method to prefixes and suffixes

Because our method is based on a regular grammar, it is impossible to use it directly for isolating the morphological root of the word (the part of the word which are prefixes and suffixes attached to). However, it is possible to reverse the direction in which the method reads and processes words. Then only prefixes (instead of suffixes) will be processed. It is also possible to join this information with the information obtained in a direct (non-reversed) run.[3]

[2] for the light stemming level, in [Paice, 1996] so called „tight groupings"

[3] I wish to thank Andrej Lúčny for this idea.

6.1 Prefix (reverse) run

This can be done easily by reversing each word in the original dictionary, running the unmodified algorithm, and reversing each word in the resulting dictionary.

This run produces group of words having a common morphological stem and having the same grammatical ending (because the words were grouped only disregarding prefixes). It should be noted that in Slovak language, adding a prefix can substantially change the semantics of the word[4]. Although the words could be still considered having a common morphological root, this is sometimes undesirable in practical applications.

Another interesting case is the negation, which is created by adding the prefix *ne-*. This run successfully joined the positive and negative forms.

6.2 Joining the information from direct and reverse run

We can use the information contained in the dictionaries created separately in the direct and in the reverse run. The words in first dictionary are grouped disregarding their inflectional suffixes; the words in the second dictionary are grouped according to one grammatical variation of a common morphological root.

Now we can create a graph, each word being a node. We add an edge between those words, which occur in the same group (at the same line) in either of the dictionaries. The connected components in this graph are the resulting stemming groups. One component contains words which share the common morphological root, without regard to the attached prefixes or suffixes.

During this join, the errors from both runs are accumulated: if the direct run failed to merge different grammatical variations, these will remain separate also in the merged dictionary. A similar argument holds also for overstemming.

6.3 Results for bi-directional approach

A good example of the resulting components could be the groups containing the forms with the morphological roots *môž-*, which is common for the words *môže-nemôže* (can-cannot), the root *možn-* for words *možné, nemožné, možnosť, možno* (possible, impossible, possibility, (it is) possible).

- nemôže nemôžeme môžeme môžem môžete nemôžete môžeš nemôžeš môže nemôžem nemôžu
- nemožné možné možného nemožnosť možnosť možnostiach možnostiam možnosti možností možnosťami možnosťou nemožno možno

We have not separately estimated the error rate for this bi-directional run; mainly because it is a laborious task. It is dependent on the error rate of separate runs in both directions, which is relatively high. The bi-directional run is shown only as an illustration for further possibilities – an improvement of one-directional run will result also in improvement of the bi-directional method.

7 Further work

It should be possible to extend our approach using a context-free grammar induction, instead of simple Myhill-Nerode equivalence. An algorithm for context-free grammars is described e.g. in [Crespi-ReghizziStefano, 1971]. Context-free grammar should perform better in identifying morphological roots and also in separating multiple suffixes or prefixes.

The stem-changes are a problematic issue. Assuming that there are regularities in the language in stem-changes, it should be possible to detect them and incorporate this information into the grammar. However, we hypothesize that (at least in Slovak language) it would be necessary to use an context-sensitive grammar, because the rules for stem-changes are mostly context sensitive (deleting or changing a part of the stem depends usually on the part itself, on the grammatical case and the paradigm, and on the preceding and the following characters)[5].

8 Conclusion

Our method is able to create a stemming dictionary using a regular grammar. It can isolate frequent suffixes in an inflecting language and create stemming

[4] e.g. *tvrdenie* (a claim) – *potvrdenie* (an acknowledgment), *činný* (active) – *účinný* (effective), *držať* (to hold) – *vydržať* (to endure) – *zadržať* (to detain).

[5] See, for example, rules for genitive case of plural nouns at page 103 in [Dvonč et al., 1966]

groups accordingly. However it requires a correct and complete dictionary, because every form of a word is taken into account with equal weight (frequencies are not used). When the derivation and the inflection suffixes mix together, our method tends to distinguish between derivation and inflection.

It is also possible to use our method in a reverse manner, for identifying morphological prefixes, and subsequently it is possible to join both directions to obtain morphological (heavy) stemming. This method is only preliminar and is influenced by the errors from the one-directional runs.

The resulting dictionary has to be manually checked. The estimated understemming index for light stemming is approximately 0.57 and the overstemming index is approximately 0.007.

Acknowledgments

This research is supported by grant projects VEGA 1/0688/10 and 1/0361/08.

References

[Crespi-ReghizziStefano, 1971] Crespi-ReghizziStefano (1971). An Effective Model for Grammar Interference. In *IFIP Congress (1)*, pages 524–529.

[Dvonč et al., 1966] Dvonč, L., Horák, G., Miko, F., Mistrík, J., Oravec, J., Ružička, J., and Urbančok, M. (1966). *Morfológia slovenskeho jazyka [Morphology of the Slovak Language]*. Vydavateľstvo Slovenskej Akademie Vied, Bratislava.

[Goldsmith et al., 2000] Goldsmith, J. A., Higgins, D., and Soglasnova, S. (2000). Automatic Language-Specific Stemming in Information Retrieval. *Lecture Notes In Computer Science; Vol. 2069*.

[Järvelin et al., 2007] Järvelin, A., Järvelin, A., and Järvelin, K. (2007). s-grams: Defining generalized n-grams for information retrieval. *Information Processing and Management: an International Journal*, 43(4).

[Jazykovedný ústav Ľ. Štúra SAV, 2009] Jazykovedný ústav Ľ. Štúra SAV (2009). Frequency Statistics for the prim-4.0, Slovak National Corpus. Available at: `http://korpus.juls.savba.sk/stats/prim-4.0/word-lemma/prim-4.0-juls-all-word-frequency.txt.gz`.

[Lovins, 1968] Lovins, J. B. (1968). Developing of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11(1).

[Malý, 2010] Malý, M. (2010). Source code for Myhill-Nerode equivalence stemming. Available at: `http://mmm.ii.fmph.uniba.sk/myhill_nerode.zip`.

[Mcnamee and Mayfield, 2007] Mcnamee, P. and Mayfield, J. (2007). N-Gram Morphemes for Retrieval.

[Nakov, 2003] Nakov, P. (2003). Building an inflectional stemmer for Bulgarian. *International Conference Computer Systems and Technologies*.

[Nerode, 1958] Nerode, A. (1958). Linear automaton transformations. *Proceedings of the American Mathematical Society*, 9(4):541–544.

[Paice and Oakes, 1999] Paice, C. and Oakes, M. (1999). A concept-based method for automatic abstracting. *Library and Information Commission*.

[Paice, 1990] Paice, C. D. (1990). Another stemmer. *ACM SIGIR Forum*, 24(3):56–61.

[Paice, 1996] Paice, C. D. (1996). Method for evaluation of stemming algorithms based on error counting. *Journal of the American Society for Information Science*, 47(8):632–649.

[Porter, 1997] Porter, M. F. (1997). An algorithm for suffix stripping. pages 313–316.