Retrieving Information in Croatian: building a simple and effcient rule-based stemmer

Nikola Ljubešić
Faculty of Philosophy, Department of Information Sciences
Ivana Lučića 3, 10 000 Zagreb
nljubesi@ffzg.hr

Damir Boras
Faculty of Philosophy, Department of Information Sciences
Ivana Lučića 3, 10 000 Zagreb
dboras@ffzg.hr

Ozren Kubelka
Vern' Polytechnic
Trg bana Josipa Jelačića 3
ozren.kubelka@vern.hr

Summary
Since Croatian is a highly flective language there is a need for morphological normalization of natural language information so that information could become retrievable in a more efficient way. Although this topic has been researched for more than two decades in Croatia, the vast majority of information systems that store information written in Croatian still do not have this problem solved. The primary cause for this situation is the high price of existing systems.
The aim of this paper is to analyze the current situation in the industry regarding this problem and to build a rule-based stemmer which would consist of a minimal set of rules for expanding queries to the whole possible paradigm. Such a system could make expensive morphological databases in information retrieval obsolete.
We used a corpus sample, a morphological lexicon of Croatian nouns and a query sample of 1.000 most frequent nouns in base form to build a rule-based stemmer optimized through the steepest ascent hill climbing algorithm. Using this method we built a stemmer which performs almost equally good as the noun lexicon with F1 measures of 97.82% without the rules for adjectives and 97.64% with them.

Key Words: Information retrieval, Croatian language, rule-based stemming, hill climbing optimization, industry awareness

Introduction

The aim of this paper is twofold - first, to analyze the practice in the IT industry in Croatia concerning the problem of Croatian morphology in information retrieval, and second, to construct a simple and efficient stemmer which could be easily used in the industry.

As far as we know, up to this point there were two stemmers built for retrieving information in Croatian. The first one was built by Tomislava Lauc [1]. She wrote rules for noun and adjective paradigms as well as for their morphonological alternations in Croatian. The reported precision of her stemmer was 90,26%. As input, nouns and adjectives in all forms were used. The stemmer was not tested on a corpus, but on a lexicon which completely neglects the frequency of specific forms. The other stemmer was built by Dobrica Pavlinušić [2]. The main purpose of this stemmer was to help in retrieving texts of the official gazette of Republic of Croatia (Narodne novine) [3]. The retrieval system using the stemmer can be found on [4]. His stemmer was not quantitatively tested and is the only one proven to be used in information retrieval of publicly available information. The only other system in public usage which probably uses a stemmer as well as a lexicon is the search engine pogodak.hr [5], but this information is being considered a business secret and couldn't be confirmed.

In the first part of the paper we discuss the results obtained through a online questionnaire taken by 16 web administrators of top web sites in Croatian web space. In the second part we develop a simple and efficient stemmer which could be used widely regardless of the size of the information system. The developed stemmer is rule-based and is optimized by the steepest ascent hill climbing algorithm. Therefore we used a corpus sample, a query sample and a gold standard. As our corpus sample we used a portion of the Vjesnik online newspaper corpus (72M, 4.5M used). We used a morphological lexicon of Croatian nouns for building gold standards - morphological indices (connect tokens that belong to the same paradigm). As our query sample we used 1.000 most frequent nouns in the corpus in their base form.

Current situation in the industry

The aim of the first part of our research was to acquire an insight in retrieval capabilities and the technology used in top 30 Croatian web sites regarding to [6], concerning the problem of morphology in Croatian. For that reason we built an on-line questionnaire and invited the 30 web administrators to fill out the questionnaire. We received 17 answers.

The first and basic question was: "Is your web site search engine sensitive to morphological changes in Croatian language? (e.g. for question "banka" is it capable of _nding documents that contain word forms such as "banaka", "bankama", "banci" etc.):".

Regarding the first answer we divided our questions into two groups:

1. If the answer was "YES", the offered questions were:
   1.1. Which search method do you use?
      - stemmer
      - lexicon
      - both
   1.2. Which engine?
   1.3. Do you find morphological sensitive search techniques enhancing the work of your search engine?
2. If the answer was "NO", the offered questions were:
   2.1. Do you consider morphological sensitive search capable of enhancing the work of your search engine?
   2.2. Are you informed about capabilities of morphological sensitive search?
   2.3. Would you be ready to incorporate morphological sensitive search into your web site in near future?
      - yes - open source engine
      - yes - commercial engine
      - yes - both
      - no - I am not interested

The results showed that at this moment there are hardly any morphologically sensitive search engines used at all. That is to say, just one of the received answers to the first question was positive. Search engines mostly use engines adjusted to English (e.g. Google) or are constructed with simple search techniques like the sameness with the query (=), or similarity with the query (like).

Still, 82% of the users that gave an answer to the question 2.1 believe that introducing a morphologically sensitive search engine would be useful, and 65% of them base their answer on basic knowledge of such search methods (question 2.2). A majority of the users interested in upgrading their search engine didn't care whether the engine would be commercial or open source (76%). Twelve percent was in favor of open source, 6% would prefer a commercial product, and the remaining 6% were not interested.

Building a simple and efficient stemmer

As stated in the introduction, stemmers for Croatian built up to this day were often too complex, mostly built with no regard to specificities of information retrieval and none of them was properly evaluated. In the second part of our research our goal is to build a simple and efficient stemmer which would be empirically tested.

As our dataset, we used two portions of the Vjesnik on-line newspaper corpus [7]. Some specifications regarding that corpus are shown in table 1. Both portions used as the validation set and the test set consist of 10.000 randomly chosen articles that don't overlap. The validation set has 4.515.651 tokens

including punctuation while the test set has 4.459.519 tokens. The validation set was used for finding the most frequent nouns, for building the set of suffix rules and for finding the optimal set of these rules. The test set was used to test the chosen set of suffix rules. In both cases, the gold standard was built with help of the morphological lexicon.
The lexicon we used consists of 21.003 nouns. It was built with the help of the xfst tool [8]. For purposes of this research we transformed the noun part of the lexicon into two hash tables. The first hash table lists all possible forms regarding the lemma {'kava' : ('kava', kave', 'kavi', 'kavu', 'kavo', 'kavom', 'kavama')}.

Table 1: Basic data about the Vjesnik on-line newspaper corpus (PM = puncuation marks)

Number of articles	187.323
Number of tokens (with PM)	82.862.497
Number of tokens (without PM)	71.935.880
Number of punctuation marks	10.926.617
Number of articles with subtitles	76.982
Number of tokens in article bodies (with PM)	78.245.043
Number of tokens in article bodies (without PM)	67.741.145
Number of sentences in article bodies	3.105.495
Average number of tokens per sentence (with PM)	25,20
Average number of tokens per sentence (without PM)	21,81

The second hash table - an inverse index - gives all possible basic forms regarding the form given {'borom' : ('bor', 'bora')}
With the help of those hash tables we formed a frequency list of nouns in the validation set regardless of the homonymy problem. The list consisted of 13.261 nouns. We checked manually the First 1.033 nouns and excluded 33 of them which got such a high ranking because of the homonymy clash with a very frequent lexeme (such examples are bit, bilo, oko, toga, kada, dok etc.). We used the remaining 1.000 nouns in base form as our query sample. This decision is based on the intuition that in most cases queries in information retrieval are nouns and adjectives in their base forms.
As the gold standard we considered morphological indices built with the help of our noun lexicon. As our query sample, we assumed the 1.000 most frequent nouns in base form from the validation set and built a list of hashes of hashes. Every outer hash consists of the key - the query (one of the 1.000 most frequent noun lemmata) and the value - the inner hash - its forms found in the corpus - the keys - with their number of occurrence - the values. An example of a list item would be {'kapacitet': {'kapacitetima': 8, 'kapacitetu': 2, 'kapacitet': 24, 'kapaciteti': 31, 'kapacitetom': 3, 'kapacitete': 38, 'kapaciteta': 112}}. We built a list of 32 su_x rules by hand through the previously described list of 1.000 most frequent nouns in the validation set. A example of a rule would be ('C', 'a', 'e', 'i',

'u', 'om', 'ama') where the first element of the tuple defines whether the last character of the invariable part of a lexeme should be a consonant or a vocal.
We also put together a set of plain suffixes (140 of them) to examine the simplest approach - just a set of suffixes where there are no specific restrictions.
Using the validation set, our goal was to find the set of rules which would maximize the F1 measure regarding our gold standard. Therefore we used the steepest ascent hill climbing algorithm which adds a rule to the rule set that, in combination with rules already in the set, maximizes the F1 measure, iterating the procedure as long as the F1 measure increases. While searching for the optimal rule set we were also interested in the form of the rule that gives highest F1 measure. Therefore, we put together five forms of our rules:

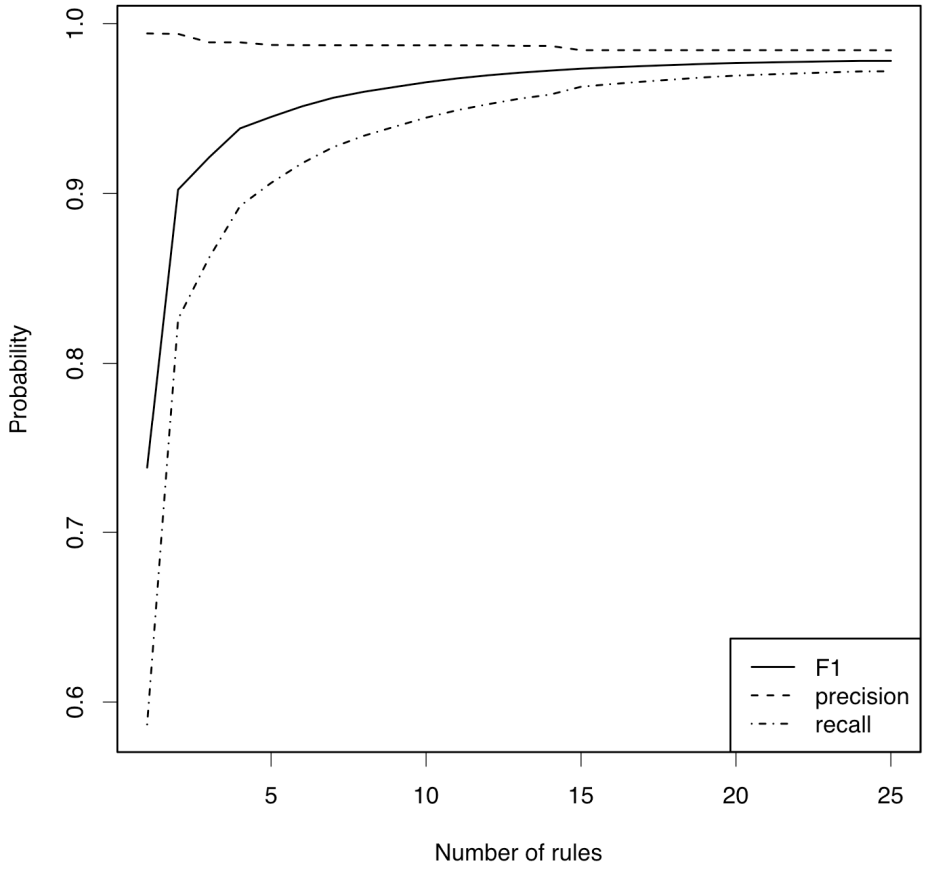
1. there is a constraint regarding the last character of the invariable part of the lexeme (if consonant or vocal) ('C', 'a', 'e', 'i', 'u', 'om', 'ama')
2. there is a constraint regarding the first, entry suffix meaning that the query must _t the entry suffix for the rule to be applied to it (the entry suffix marked with square brackets) (['a'], 'e', 'i', 'u', 'om', 'ama')
3. there are both constraints from the first and the second form of rules ('C', ['a'], 'e', 'i', 'u', 'om', 'ama')
4. there are none of the constraints described above ('a', 'e', 'i', 'u', 'om', 'ama')
5. there is no set of rules, but just a set of suffixes without any entry restrictions ('', 'a', 'u', 'om', 'i', 'ima', 'e', 'ama', 'em', ...)

Regarding the different forms of rules, the final F1 measure, precision, recall and the number of chosen rules are given in table 2. The maximal F1 measure was achieved with the second form (97.81%) with a small advantage towards other forms. The second form on the other hand requires the biggest amount of rules (25). The least success proved to give the fifth form with a F1 measure of only 94.17%. Since the second form, disregarding the fifth one, is the simplest to apply (there is no necessity for checking the character preceding the potential suffix, and only one suffix - namely the first one (the entry suffix) - has to be checked when searching for rules that can be applied to a specific query token) and has a slightly higher F1 than the others, we have decided to use to second form of rules with the 25 chosen rules.

Table 2: F1, precision and recall measures and number of rules on the validation set regarding the five forms of rules

rule	form	F1	precision	recall # of rules
1	0.9730	0.9775	0.9686	23
2	0.9781	0.9843	0.9719	25
3	0.9747	0.9828	0.9667	24
4	0.9754	0.9774	0.9734	23
5	0.9417	0.9698	0.9151	19

The optimal rule set consisting of 25 rules in order how the rules were added to the set is: (('', 'a', 'u', 'om', 'i', 'ima', 'e'), ('a', 'e', 'i', 'u', 'om', 'ama'), ('e', 'a', 'u', 'em', 'ima'), ('o', 'a', 'u', 'om', 'ima'), ('', 'a', 'u', 'om', 'ovi', 'ova', 'ovima', 'ove'), ('ak', 'ka', 'ku', 'kom', 'ci', 'aka', 'cima', 'ke'), ('k', 'ka', 'ku', 'kom', 'ci', 'cima', 'ke'), ('ac', 'ca', 'cu', 'cem', 'ci', 'aca', 'cima', 'ce'), ('anj', 'nja', 'nju', 'njem', 'njom', 'nji', 'anja', 'njima', 'nje'), ('', 'ka', 'ke', 'ci', 'ki', 'ku', 'kom', 'aka', 'kama'), ('ar', 'ra', 'ru', 'rom', 'ri', 'ara', 'rima', 're'), ('ao', 'la', 'lom', 'lu', 'lovi', 'lova', 'lovima', 'love'), ('', 'a', 'u', 'om', 'em', 'evi', 'eva', 'evima', 'eve'), ('an', 'na', 'nu', 'nom', 'ni', 'ana', 'nima', 'ne'), ('in', 'ina', 'inu', 'inom', 'i', 'a', 'ima', 'e'), ('am', 'ma', 'mu', 'mom', 'movi', 'mova', 'movima', 'move'), ('t', 'ta', 'tu', 'tom', 'ti', 'ata', 'tima', 'te'), ('zak', 'ska', 'sku', 'skom', 'sci', 'zaka', 'scima', 'ske'), ('tak', 'tka', 'tku', 'tkom', 'tci', 'ci', 'taka', 'tcima', 'cima', 'tke'), ('dac', 'ca', 'cu', 'cem', 'ci', 'daca', 'cima', 'ce'), ('ga', 'ge', 'zi', 'gi', 'gu', 'gom', 'gama'), ('st', 'sti', '_s_cu', 'stima'), ('g', 'ga', 'gu', 'gom', 'zi', 'zima', 'ge'), ('sao', 'sli', '_slju', 'slima'), ('t', 'ti', '_cu', 'tima')). Every rule can be accessed through the first item in every tuple - the entry suffix. The growth of the F1 measure and the respective recall and precision regarding the number of rules added to the set is shown graphically in figure 1.



Picture 1: Growth of the F1 measure with respective precision and recall on the validation set as new rules in second form are added to the rule set

When applying the rules chosen on the validation set on the test set, we got a F1 measure of 97,82%, a precision of 98,40% and a recall of 97,24%.

Adding rules for adjectives

Although the rules concerning adjectives are not a primary part of this phase of research, we were interested in how much the most general rules for adjectives, which cover most of them, would harm the task of finding all forms of noun queries in our corpus.

We built six most general rules for adjectives and implemented them on top of the 25 previously selected rules for nouns. We designed the rules for adjectives in the same manner as we decided to design those for nouns. The F1, precision

and recall measures when applying these six rules - in order of harmlessness - are shown in table 3.

Table 3: F1 measure with precision and recall on the validation set as rules for adjectives are added to the rule set

# of rules	F1	precision	recall
	0.9781	0.9843	0.9719
1	0.9781	0.9843	0.9719
2	0.9779	0.9836	0.9722
3	0.9776	0.9831	0.9722
4	0.9773	0.9822	0.9725
5	0.9769	0.9814	0.9725
6	0.9763	0.9803	0.9725

The six rules for adjectives, again in order of harmlessness, are as follows: (('an', 'ni', 'nog', 'noga', 'nome', 'nomu', 'nim', 'ni', 'nih', 'nima', 'ne'), ('o', 'og', 'oga', 'om', 'ome', 'omu', 'im', 'a', 'ih', 'ima', 'e'), ('ni', 'an', 'nog', 'noga', 'nome', 'nomu', 'nim', 'ni', 'nih', 'nima', 'ne'), ('a', 'e', 'oj', 'u', 'om', 'ih', 'ima'), ('', 'i', 'og', 'oga', 'om', 'ome', 'omu', 'im', 'ih', 'ima', 'e'), ('i', 'og', 'oga', 'om', 'ome', 'omu', 'im', 'ih', 'ima', 'e')).
On the test set, when applying all the 31 rules, the final F1 measure was 97,64%, precision was 98,00% while recall was 97,29%. Thereby we experienced a rise in the recall measure. We believe it is so because of the similarity of paradigms for nouns and adjectives. On the other hand, we lost on precision since some noun queries hopped over to their related adjective forms. Therefore we experienced a small decline in the overall F1 measure.

Conclusion

In this paper we first analyzed the situation in the industry regarding the problem of morphology in retrieving information in Croatian. Our conclusion was that almost none of the most visited on-line portals have this problem solved (just one), although the majority of web administrators (95%) is interested in using a web service which could deal with such a problem.
In the second part of the paper we developed a simple and efficient rule-based stemmer which we optimized through the steepest ascent hill climbing algorithm on a corpus and query sample regarding the lexicon as our gold standard. By optimizing the stemmer we searched for a set of rules which would maximize the F1 measure. We experimented with five forms of rules and accepted the simplest one in form of a set of rules that also got the highest F1 score. Out of 32 rules we accepted 25 of them. In the last step we also measured the decline of measures while adding general rules for adjectives. On the test set, we got a similar F1 measure to our lexicon-based gold standard - 97.82% without the rules for adjectives and 97.64% with them.

We also built a web service which does query expansion regarding the chosen rules and a web application which uses the web service in searching the 4.5M portion of the Vjesnik on-line newspaper corpus. Instructions how to use the web service as well as the example web application can be found on http://faust.ffzg.hr/stemming/.

Our future research will include analyzing query samples received from different information systems such as [9] to back up our assumption that most queries are nouns and adjectives in their base forms. We plan to use different optimization algorithms, enhance our quality measures regarding the number of over generated forms and use a real-world query sample while optimizing the stemmer.

References

[1] Lauc, Tomislava. Problemi obrade prirodnoga jezika u sustavima za pretraživanje obavijesti putem pretraživanja punoga teksta na hrvatskome književnom jeziku. PhD thesis. Zagreb: Faculty of Philosophy, 2000.

[2] Pavlinušić, Dobrica. NN: pretraživač za hrvatski; Information wants to be free. 04-10-20. http://www.rot13.org/ dpavlin/nn.html (07-07-07)

[3] Narodne novine. 07-07-07. http://www.nn.hr/sluzbeni-list/sluzbeni/pregled.asp (07-07-07)

[4] Pavlinušić, Dobrica. Narodne novine pretraživanje. http://nn.rot13.org/ (07-07-07)

[5] Pogodak! 07-07-07. http://www.pogodak.hr (07-07-07)

[6] geminusAudience, pattern, n=16.531, may 2007, Croatia, VALICON i GEMINUS S.A

[7] Vjesnik on-line. 07-07-01. http://www.vjesnik.hr. (07-07-01)

[8] Beesley, Kenneth R.; Karttunen, Lauri. Finite-state morphology. Stanford: CSLI Publications, 2003.

[9] www.hr; hrvatski homepage od 1994. 07-07-20. http://www.hr (07-07-20)