A Probabilistic Tagging Module Based on Surface Pattern Matching

Robert Eklund

Book Chapter

N.B.: When citing this work, cite the original article.


Copyright: Stockholm University

Available at: Linköping University Electronic Press
http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-135280
A Probabilistic Word Class Tagging Module Based On Surface Pattern Matching

Robert Eklund
Stockholm

Abstract

This paper treats automatic, probabilistic tagging. First, residual, untagged, output from the lexical analyser SWETWOL is described and discussed. A method of tagging residual output is proposed and implemented: the left-stripping method. This algorithm, employed by the module ENDTAG, recursively strips a word of its leftmost letter, and looks up the remaining ‘ending’ in a dictionary. If the ending is found, ENDTAG tags it according to the information found in the dictionary. If the ending is not found in the dictionary, a match is searched in ending lexica containing statistical information about word classes associated with the ending and the relative frequency of each word class. If a match is found in the ending lexica, the word is given graded tagging according to the statistical information in the ending lexica. If no match is found, the ending is stripped of what is now its left-most letter and is recursively searched in dictionary and ending lexica (in that order). The ending lexica – containing the statistical information – employed in this paper are obtained from a reversed version of Nusvensk Frekvensordbok (Allén 1970), and contain endings of one to seven letters. Success rates for ENDTAG as a stand-alone module are presented.

1 Introduction

One problem with automatic tagging and lexical analysis is that they are never (as yet) 100% accurate. Varying tagging algorithms, using different methods, arrive at success rates in the area of 94–99%. After machine analysis there remains an untagged residue, and the complete output may – somewhat roughly – be divided into three subgroups:

1 A group of unambiguously tagged words.
2 A group of homographs given alternative tags.
3 A residual group lacking tags.

1 This paper is an abbreviated version of my diploma paper in computational linguistics with the same title, presented in April 1993 at the Department of Linguistics, Computational Linguistics, Stockholm University.
4 There is a bulk of words which is never found in this group, preponderantly those belonging to the closed words classes, since these normally are found in the lexicon.
Whereas the second of these groups is treated in Eriksson (1992), the task undertaken in this paper is to develop an algorithm for tagging material which has come through lexical analysis untagged.

The paper falls into the following areas:

First, untagged, residual output from the lexical analyser SWETWOL (Karlsson 1990; Koskenniemi 1983a,b; Pitkänen 1992) is described and analysed. This is done in order to pin down what input is, in one way or another, problematic to an automatic tagger. This is covered in section 3.

This paper presents a probabilistic tagger – henceforth ENDTAG – which tags according to statistics on the relations between final-letter combinations and word classes. The statistical information was obtained from the listings in NFO (Allén 1970) and collected in special ending lexica. This is described in section 4.

The ENDTAG module is presented in section 5. ENDTAG is based on what is here called the left-stripping algorithm, which recursively strips a word from its leftmost letter and compares the remaining ending1 with the statistical information in ending lexica described in section 4.

The results of ENDTAG are evaluated in section 6.

2 Method

The untagged material used in this paper consists of residual files from the lexical analyser SWETWOL in Helsinki. SWETWOL was run on 831.289 words, whereof 10.988 came out untagged. Since SWETWOL yields output files of words on a word-for-word basis – thus ignoring (more or less) things like lexicalised phrases, particle verbs (ubiquitous in Swedish) and the like, words were only analysed one-by-one. A conjectural supposition is that a higher rate of accuracy is to be expected if context is also considered, as attempts with purely heuristic parsers show (cf. Källgren 1991b,c, Brodda 1983). On the other hand, it can be argued that there is palpable explanatory value in trying to find out how much information can be extracted from the words alone, neglecting their immediately adjacent ‘text-mates’.

The success rate of any automatic tagger or analyser, per se and in comparison with other automatic taggers, is of course dependent on what tagset is being employed. The more general it is, i.e., the fewer the tags,

---

1 The word ‘ending’ will throughout this paper denote any word final letter cluster, be this a grammatical suffix or not.
the more ‘accurate’ the output will be, due to the lack of more subtle subcategories. Since it was judged important that the tagset easily harmonise with already existing tagsets employed in other systems, the ending statistics were obtained from *Nusvensk Frekvensordbok*, NFO hereinafter (Allén 1970). I opted to adhere to the tagset employed therein, thus, the tags employed in this paper constitute a proper subset of the NFO tags. It should be pointed out that NFO also contains tags for subcategories. The tagset employed by ENDTAG is shown in Table 1.

<table>
<thead>
<tr>
<th>ABBREVIATION</th>
<th>WORD CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ab</td>
<td>adverb</td>
</tr>
<tr>
<td>al</td>
<td>article</td>
</tr>
<tr>
<td>an</td>
<td>abbreviation</td>
</tr>
<tr>
<td>av</td>
<td>adjective</td>
</tr>
<tr>
<td>ie</td>
<td>infinitival marker</td>
</tr>
<tr>
<td>in</td>
<td>interjection</td>
</tr>
<tr>
<td>kn</td>
<td>conjunction</td>
</tr>
<tr>
<td>nl</td>
<td>numeral</td>
</tr>
<tr>
<td>nn</td>
<td>noun</td>
</tr>
<tr>
<td>pm</td>
<td>proper noun (proprium)</td>
</tr>
<tr>
<td>pn</td>
<td>pronoun</td>
</tr>
<tr>
<td>pp</td>
<td>preposition</td>
</tr>
<tr>
<td>vb</td>
<td>verb</td>
</tr>
<tr>
<td>**</td>
<td>non-Swedish unit</td>
</tr>
<tr>
<td>NT1</td>
<td><em>Not tagged in NFO</em></td>
</tr>
</tbody>
</table>

The ENDTAG module was implemented in COMMON LISP.

3 A Curt Description of the Untagged Output

In order to pin down what needs to be accounted for in tagging algorithms for arriving at better figures, one naturally has to scrutinise, with as great a punctilio as possible, the contents of that residual group of untagged words. I will here briefly list just a few observations made.²

In the untagged material, *proper* and *place nouns* abound! This is not really surprising, since they do not to any greater extent exhibit consistent morphological patterns.³ It is also hard to list them all in the lexicon. Liberman and Church (1992) mention that a listtv from the Donnelly marketing organisation 1987 contains 1.5 million proper nouns (covering

---

¹ Since it was found that not all words in the computer readable version of NFO were tagged, an additional tag was created to render the format consistent. Hence, the tag ‘NT’ was added.

² For a more detailed account, the reader is referred to Eklund 1993.

³ Of course *some* consistent patterns can be found. Thus the suffix -(s)son in Swedish typically denotes a surname, as in *Eriksson, Svensson* etc.
72 million American households). Since these have any number of origins, it is not feasible to cover them either with morphological rules or with a lexicon.\textsuperscript{1}

**Abbreviations** were also common, which is more surprising, since all these should, it is assumed, have been expanded/normalised in the pre-processing. A related – and harder – problem concerns lexical abbreviations and **acronyms**.

**Compounds** constitute a notorious problem in all automatic processing of Swedish. Because they are legion, compounds constitute a very dire problem for any tagging module working on Swedish text. It might even be hard to decide where the compound border is located.

A related problem is encountered in what I call **complex compounds**. By that I mean compound words created in ways diverging from the ‘normal’ compounding of two ordinary words. One example of this is when more than two words are compounded. Instances of such compounds are:

*Djurholms-Bromma-Lidingö-gången*  
‘The Djurholm-Bromma-Lidingö gangs’

*karpatisk-balkansk-bysantiska*  
‘Carpathian-Balcanian-Bysanthinian’

*du-och-jag-ensamma-i-världen*  
‘you-and-I-alone-in-the-world’

These clearly exhibit a **word-hyphen-word** pattern which could be formalized thus:

\[ X-(Y-)*Z \]

These, I assume, would normally obtain the correct tag if one just looked at \( Z \) alone, and tagged accordingly. Compounds like these, I have found, were rather common in psychological terminology, where it also typically was used rather freely as to word class. A ‘word’ such as *du-och-jag-ensamma-i-världen* may be used as an adjective or a noun, for example.

\textsuperscript{1} Something that could be considered here is majuscule heuristics, but this is not done without problems since upper case letters appearing in texts might indicate a wide variety of different phenomena. For example, the first letter of each sentence in a typical text, Roman figures, initials, titles and headings etc. Because of these problems, I chose to let the algorithm exempt majuscules altogether. For further discussion on majuscules, cf. e.g. Libermann & Church (1992), Eeg-Olofsson (1991:iv et passim), Källgren (1991b) and Sampson (1991).
A similar problem concerns what I call slash compounds like:

Dannemora/Österby
Hornstein/Voristan

... where the slash (/) separates two words according to the formalised pattern:

X/Y

Other phenomena occurring amongst the untagged residue were professional/special terms, diacritica, archaisms and numbers in various forms.

Another rather amusing, problem is posed by a word like

aaaaaahh!

This word is of a recursive disposition which could be formalised thus:

a+h+!+

... where the plus sign denotes any number, equal to or greater than one, and not necessarily the same number in all three instances.

A large part of the untagged output was made up of foreign words, expressions and quotations et cetera. Interestingly enough, some of the suffixes used in certain languages are sufficiently unambiguous to permit a graded tagging in Swedish. Thus, some endings of Latin origin, -ium, -ukt or -tion, and some endings of Greek origin, -graf, -lit, -ark, -skop or -logi are highly unambiguous as to word class.

A problem harder to solve is that of new words being continuously created, old words given new interpretations, and then being used as members of other word classes. Thus, even a word like the conjunction but can not be considered a sure-fire case. In a phrase like

‘But me no buts!!’

... ‘but’ first occurs as a verb in its imperative form, and then as a noun in the plural.¹ One must also point out that all words, irrespective of word class, might be used as nouns in a meta-linguistic way, for instance:

---

¹ A Swedish, idiomatic, counterpart would perhaps be Menna mig hit och menna mig dit!, the story being a speaker annoyed with a listener who interrupts by saying but all the time!
A ‘green’ would suit this phrase better!
Thou employest too many a ‘lest’ in thy prolegomenon, young esquire!

Lexicalized phrases typically receive the wrong parses, especially if they allow other constituents to be included ‘inside’ them. Since, as mentioned before, the module works with but a one-word window, lexicalized phrases cannot be properly accounted for by the module.

4 Obtaining the Ending Lexica

If we are to tag on a probabilistic basis, we need statistical information on the ending/word class relation. Hence, the first task was to create a number of ending lexica containing information as to word classes associated with particular endings. As mentioned earlier, the ending lexica were obtained from the lists in NFO (Allén 1970). NFO is a listing based on one million running words obtained from the material PRESS-65 and exists in computer-readable format. It might be pointed out that NFO is based exclusively on newspaper texts, and that other types of texts would perchance result in different ending lists. (Then again, results always depend on the input material used.)

Ending lexica were created with endings of 1–7 letters1 – one lexicon per ending length – and word classes and their relative frequencies were calculated. Thus, the final format is as follows:

("ENDING" ("WORD-CLASS1 PERCENTAGE1) "WORD-CLASS2 PERCENTAGE2") "WORD-CLASSn PERCENTAGEn")

Word class frequencies are given with four decimals, and the word classes appear in falling order according to frequency. Thus, an authentic typical lexicon entry (from the three-letter ending lexicon):

("ari" ("nn" 0.7802) ("ab" 0.1209) ("pm" 0.0934) ("**" 0.0055))

In other words, if the three final letters of a Swedish words are -ari, then there is a 78% probability that the word is a noun, a 12% probability that it is an adverb, a 9% probability that is a proper noun, and finally, a 0.5% probability that it is a foreign word.

The output files of the ENDTAG module look exactly the same apart from the first member of the list which will be the entire word, instead of as above, a final letter cluster.

---

1 The number seven was chosen without any reason in particular.
The number of entries for each of the ending lexica is shown in Table 2.

**TABLE 2 : Numbers of entries in the ending lexica obtained from NFO.**

<table>
<thead>
<tr>
<th>Number of letters in each ending lexicon</th>
<th>one</th>
<th>two</th>
<th>three</th>
<th>four</th>
<th>five</th>
<th>six</th>
<th>seven</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of entries in lexica</td>
<td>43</td>
<td>669</td>
<td>3 936</td>
<td>13 176</td>
<td>26 494</td>
<td>38 464</td>
<td>46 179</td>
</tr>
</tbody>
</table>

One thing which cannot be bypassed is the extent to which the number of word classes associated with an ending decreases with the number of letters in the ending, i.e., the longer the final letter cluster, the fewer word classes associated with that ending. Statistics showing these relationships are illustrated in Table 3.

**TABLE 3 : Number of word classes associated with number of letters in endings (percentages). Zero percent area is marked with bold line.**

<table>
<thead>
<tr>
<th>Number of word classes</th>
<th>one letter</th>
<th>two letters</th>
<th>three letters</th>
<th>four letters</th>
<th>five letters</th>
<th>six letters</th>
<th>seven letters</th>
</tr>
</thead>
<tbody>
<tr>
<td>one</td>
<td>30.2</td>
<td>39.5</td>
<td>52.9</td>
<td>71.4</td>
<td>85.4</td>
<td>92.1</td>
<td>94.9</td>
</tr>
<tr>
<td>two</td>
<td>23.0</td>
<td>13.8</td>
<td>20.9</td>
<td>19.4</td>
<td>12.3</td>
<td>7.2</td>
<td>4.7</td>
</tr>
<tr>
<td>three</td>
<td>–</td>
<td>12.1</td>
<td>12.9</td>
<td>6.2</td>
<td>1.8</td>
<td>0.6</td>
<td>0.3</td>
</tr>
<tr>
<td>four</td>
<td>23.0</td>
<td>9.6</td>
<td>6.2</td>
<td>2.0</td>
<td>0.3</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>five</td>
<td>4.7</td>
<td>7.3</td>
<td>3.3</td>
<td>0.7</td>
<td>0.1</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>six</td>
<td>7.0</td>
<td>5.2</td>
<td>2.1</td>
<td>0.2</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>seven</td>
<td>–</td>
<td>4.0</td>
<td>1.0</td>
<td>0.1</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>eight</td>
<td>9.3</td>
<td>3.7</td>
<td>0.5</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>nine</td>
<td>2.3</td>
<td>1.6</td>
<td>0.2</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>ten</td>
<td>4.7</td>
<td>1.5</td>
<td>0.1</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>eleven</td>
<td>11.6</td>
<td>0.7</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>twelve</td>
<td>9.3</td>
<td>0.1</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>thirteen</td>
<td>9.3</td>
<td>0.7</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>fourteen</td>
<td>4.7</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>fifteen</td>
<td>2.3</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>
A detailed description of the contents of the ending lexica will not be given here, but one example will perhaps serve as an indicator as to how the module works. Table 4 shows that probability rises as a function of increasing length for three noun declensions in Swedish.

<table>
<thead>
<tr>
<th>Decisions</th>
<th>$-r$ (plural)</th>
<th>$-r,na$ (plural+definite)</th>
<th>$-r,nas$ (plural+definite+genitive)</th>
</tr>
</thead>
<tbody>
<tr>
<td>First declension</td>
<td>74.8</td>
<td>96.9</td>
<td>100.0</td>
</tr>
<tr>
<td>Second declension</td>
<td>26.5</td>
<td>97.8</td>
<td>98.4</td>
</tr>
<tr>
<td>Third declension</td>
<td>41.5</td>
<td>94.9</td>
<td>97.6</td>
</tr>
</tbody>
</table>

5 A Description of the Left-Stripping Algorithm

The tagging problem has been approached by many a linguist in many a way. Morphological models of Swedish have been provided by Hammarberg (1966), Kiefer (1970), Linell (1972, 1976), Cedwall (1977), Hellberg (1978),1 Brodda (1979), Blåberg (1984), Eeg-Olofsson (1991:III), Ejerhed and Bromley (1986) and others. These works however, predominantly treat either very specific areas of Swedish morphology with varying degrees of minuta, or are generative models for Swedish word formation.


1 Implemented by Ivan Rankin (1986).
The algorithm presented in this paper – the left-stripping algorithm – works by simple surface structure pattern matching. The concept is to strip a word of its leftmost letter, look for the resulting ‘word’ – i.e., the previous word sans its first letter – in a dictionary (e.g. SWETWOL for Swedish). If it is found, the word is tagged according to the dictionary, and the procedure is repeated with the next word. If it is not found, and the number of letters in the word is small enough to have a corresponding ending lexicon, i.e., the same number of letters, the word is looked for in that ending lexicon. If it is found in the ending lexicon, it is tagged, and the whole procedure is repeated with the next word. If it is not found, the word is stripped of what is now its leftmost letter, searched for in the dictionary et cetera. If no match is found even at the final (one) letter stage, the word is tagged thus:

(“ENDING” ((NONE 0.0)))

The rationale behind this somewhat pleonastic design of the word class list is a desire to keep the format consistent. The flow chart in Figure 1 describes the module.

\[\text{\textbf{FIGURE 1 – Flow chart of the ENDTAG module.}}\]

As mentioned earlier, ‘ending’ here denotes the \(n\) final letters of a word, irrespective of whether these be grammatical suffixes, common combinations of any kind or unique word-final clusters. The dictionary lookup is likely to succeed before the ending lexicon, since the length of a complete word (normally) perforce exceeds the length of its ending.\(^1\)

The module iterates over the untagged output list and strips the words recursively until a match is found in either the dictionary or the ending lexica. In the test run carried out here, no dictionary was employed, and the sub-routine intended to perform the dictionary lookup was foregone.

\(^1\) In some instances, however, it might be hard to tell the difference between a word and its ending. Thus, in quoting John Lennon’s *Give Peace A Chance*: …Ragism, Tagism, / This-ism, that-ism, ism, ism. …it might be hard to tell the difference between the word and the ending in ism.
6 Results and Discussion

Since the output files of the module provide graded tagging, it is somewhat hard to discuss the results in terms of ‘hits’ or ‘misses’. What could be discussed is how often the word class with the highest percentage is also the ‘correct’, word class. Although the module was not conceived as being used as a stand-alone module, it is of a certain interest to check its capabilities a such. Thus, a test run was carried out on 316,599 already tagged – and manually checked – words in the Stockholm-Umeå Corpus (Källgren 1991a). The leftmost member in the resulting output lists of ENDTAG were compared to the tags in SUC. The percentages are given in Table 5.

<table>
<thead>
<tr>
<th>WORD CLASS</th>
<th>PERCENTAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infinitival marker</td>
<td>100</td>
</tr>
<tr>
<td>Nouns</td>
<td>93</td>
</tr>
<tr>
<td>Verbs</td>
<td>93</td>
</tr>
<tr>
<td>Prepositions</td>
<td>82</td>
</tr>
<tr>
<td>Adjectives</td>
<td>78</td>
</tr>
<tr>
<td>Adverbs</td>
<td>78</td>
</tr>
<tr>
<td>Conjunctions</td>
<td>69</td>
</tr>
<tr>
<td>Proper nouns</td>
<td>66</td>
</tr>
<tr>
<td>Pronouns</td>
<td>63</td>
</tr>
<tr>
<td>Interjections</td>
<td>37</td>
</tr>
<tr>
<td>Numerals</td>
<td>26</td>
</tr>
<tr>
<td>Abbreviations</td>
<td>16</td>
</tr>
</tbody>
</table>

One interesting feature of ending-list based tagging is the method’s inherent capabilities regarding the tagging of new words (cf. Greene & Rubin 1971). Since word formation obeys morphological rules, one may predict that neologisms and inflected loan words should be given rather accurate tags by the module.

One could also point out that one of the contributions of this work is the actual ending lexica per se. These have not been scrutinised in detail, but could presumably provide interesting information if studied.
Another point worth making is the module’s limitations. Primo, it works on a brute force basis, rather than with linguistic finesse. The fact that it is not based on grammatical or morphological descriptions or models of Swedish, precludes generation, whence it follows that the module is not bi-directional, a lack we will have to make do with if we want to be able to handle foreign entries. Secundo, as already pointed out, the ending information in the ending lexica is perforce dependent upon the material on which they are based (in this case NFO). Tertio, tagging is graded. If an unambiguous tagging is desired, the module must succeed at lengths greater than (in most cases) three to four letters.

As a final remark, it could be said that no one tagging strategy, hitherto, has been able to solve this task fully. A combination of several different methods might increase success rates. A combination of a lexically based method (SWETWOL) with a statistically based method (ENDTAG), disambiguated by a module like the one described by Eriksson (1992) could enhance success rates in automatic word class recognition.

References


