A probabilistic part-of-speech tagger for Swedish

Peter Nilsson
Department of Computer Science
University of Lund
Lund, Sweden
dat00pen@ludat.lth.se

Abstract

This paper presents a project for implementing and evaluating a probabilistic part-of-speech tagger for Swedish. The resulting program accurately tagged 95.9% of text containing only known words, and 89.7% of randomly selected text.

1 Introduction

Part-of-speech tagging is the method to assign each word in a text with a tag describing its proper part of speech in the context.

Two major problems in the process is resolving ambiguous words and classifying unknown words. To solve this task a few different methods has been developed. The most common are tagging based on rules, and probabilistic, or stochastic, tagging based on statistics. This paper presents a project for implementing a probabilistic part-of-speech tagger for Swedish.

2 Probabilistic POS tagging

In probabilistic part-of-speech tagging the tagger program is trained on a corpus annotated with a certain tagset. Statistics for the sequence of words and tags is collected, and is then used for estimating the most probable tagging of untagged text.

The background theory is given in (Charniak et al., 1993). An instructive description of the steps taken and decisions made in implementing an efficient stochastic part-of-speech tagger is found in (Carlberger and Kann, 1999), a paper which has served as a major guide for the initial steps of this project.

2.1 The basic model

This section contains a brief overview of the basic theory for developing the model.

A text can be considered as a sequence of n random variables $W_{1..n} = W_1, W_2, ... W_n$ where each variable can have a value from a fixed set of words $\{w_1, w_2, ... w_n\}$. For each sequence of word variables $W_{1..n}$ there is a corresponding sequence of random variables $T_{1..n}$, that can take their values from a fixed set of tags $\{t_1, t_2, ... t_n\}$. The tagging problem can then be described as finding the most probable sequence of tags, $t_{1,n}$, knowing the sequence of words, $w_{1,n}$. A formal definition of this is:

$$\mathcal{T}\left(w_{1,n}\right) = \arg\max_{t_{1,n}} P\left(t_{1,n} \mid w_{1,n}\right)$$

From Bayes' rule we know that:

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}.$$

Applying this on the right hand side we get:

$$\mathcal{T}\left(w_{1,n}\right)=\arg\max_{t_{1,n}}\frac{P\left(w_{1,n}\mid t_{1,n}\right)P\left(t_{1,n}\right)}{P\left(w_{1,n}\right)}$$

and since the denominator will be constant for a given problem the expression is simplified to:

$$\mathcal{T}\left(w_{1,n}\right)=\arg\max_{t_{1,n}}P\left(w_{1,n}\mid t_{1,n}\right)P\left(t_{1,n}\right) \qquad (1)$$

2.2 Approximations for the implementation

The probability values needed for solving this problem is achieved from hand-annotated corpora. Even a large corpus will not be large enough for collecting the statistics needed in Equation 1. For this reason the problem is approximated using the following two assumptions:

$$P\left(t_i\mid t_{1,i-1},w_{1,i-1}\right)=P\left(t_i\mid t_{i-2},t_{i-1}\right)$$

$$P\left(w_i\mid t_{1,i},w_{1,i-1}\right)=P\left(w_i\mid t_i\right)$$

The first assumption states that the current tag is only dependent on the two previous tags. The second assumption states that the current word is only dependent on the current tag.

The expression for the problem now becomes:

$$\mathcal{T}\left(w_{1,n}\right)=\arg\max_{t_{1,n}}\prod_{i=1}^{n}P\left(t_i\mid t_{i-2},t_{i-1}\right)P\left(w_i\mid t_i\right) \qquad (2)$$

Equation 2 allows us to use statistics from every sequence of three tagged words, called trigrams, in a corpus. Even with this simplification the data will usually be too sparse to produce a reliable result. There are simply too many trigrams that will not appear or will only appear a small number of times in a corpus. We will have to approximate further by using the probabilities for the sequence of two tagged words, bigrams:

$$P\left(t_i\mid t_{i-2},t_{i-1}\right)\approx P\left(t_i\mid t_{i-1}\right)$$

and in the case the bigram data will be too sparse we will also use the unigrams:

$$P\left(t_i\mid t_{i-1}\right)\approx P\left(t_i\right)$$

Now we apply a useful strategy by interpolating between these:

$$P\left(t_i\mid t_{i-2},t_{i-1}\right)\approx$$

$$\lambda_1P\left(t_i\mid t_{i-2},t_{i-1}\right)+\lambda_2P\left(t_i\mid t_{i-1}\right)+\lambda_3P\left(t_i\right)$$

where $\lambda_1+\lambda_2+\lambda_3=1$.

Putting this into Equation 2, we get:

$$\mathcal{T}\left(w_{1,n}\right)=$$

$$\prod_{i=1}^{n}\left[\lambda_1P\left(t_i\mid t_{i-2},t_{i-1}\right)+\lambda_2P\left(t_i\mid t_{i-1}\right)+\lambda_3P\left(t_i\right)\right]P\left(w_i\mid t_i\right) \qquad (3)$$

Finding good values for the three lambdas is part of this project.

2.3 Estimating the probabilities

For estimating the probabilities in Equation 3 the following statistics is needed:

- Cn : the count of all words
- $C(w)$: the count of word w
- $C(t)$: the count of tag t
- $C(t_1,t_2)$: the count of bigrams where tag t_1 is followed by tag t_2
- $C(t_1,t_2,t_3)$: the count of trigrams where tag t_1 is followed by tag t_2 and then by tag t_3
- $C(w,t)$: the count of word w tagged with tag t

$$\mathcal{T}\left(w_{1,n}\right) = \arg\max_{t_{1,n}} \prod_{i=1}^{n} \left[\lambda_1 \frac{C(t_{i-2}, t_{i-1}, t_i)}{C(t_{i-2}, t_{i-1})} + \lambda_2 \frac{C(t_{i-1}, t_i)}{C(t_{i-1})} + \lambda_3 \frac{C(t_i)}{C_n} \right] \frac{C(w_i, t_i)}{C(t_i)}$$

Figure 1: Equation 4

The statistics is then used to estimate the probabilities as follows:

- $P(t_i) = \frac{C(t_i)}{C_n}$
- $P(t_i \mid t_{i-1}) = \frac{C(t_{i-1}, t_i)}{C(t_{i-1})}$
- $P(t_i \mid t_{i-2}, t_{i-1}) = \frac{C(t_{i-2}, t_{i-1}, t_i)}{C(t_{i-2}, t_{i-1})}$
- $P(w_i \mid t_i) = \frac{C(w_i, t_i)}{C(t_i)}$

Substituting this in Equation 3, we get the final equation in Figure 1.

This expression can easily be translated into a computer program. Tables for the various statistics can be built from an annotated corpus, and a suitable algorithm can iterate over the different combinations of probabilities to find the maximum. The naïve implementation, making an exhaustive search, will not suffice for sentences longer than a few words though, since the algorithms expansion is polynomial. The Viterbi algorithm is usually applied in this situation, as it is quite insensitive to any length of a sentence.

3 The SUC corpus

The corpus used is the SUC 1.0 corpus, created between 1989-1996 as part of a research project by the Departments of Linguistics at Stockholm University and Umeå University.

The selection of texts is a collection of modern Swedish prose first published in 1990 or later. Each of the more than one million words is annotated by the SUC tagset, which is listed in appendix A. The corpus is divided into 500 files containing roughly 2000 annotated words each.

4 The implementation

The program was developed in C++ using the Standard Template Library for the tables. It is divided into two major parts. The first part builds the statistics tables by parsing a set of annotated files. The second part implements the algorithm in Figure 1 and produces a suggestion of tagging for a text, one sentence at a time.

4.1 Training and evaluation

The goal for this project was to implement the basic model for probabilistic tagging in a computer program. The program was to be trained on the SUC corpus, and part of the corpus should be set aside for test and evaluation. Unknown words should be handled in a simple manner.

The corpus was divided into 50 randomly selected files each for the optimizing and test sets, and the remaining 400 for the train set.

A complete evaluation of the program was performed as follows:

The training phase consist of collecting the statistics from the train set. This means counting words, bigrams, trigrams etc and store the result in tables.

In the optimizing phase each text in the optimizing set is read and tagged by the program using the equation in Figure 1 with different sets of values for the three lambdas. The values used are between 0 and 1 in steps of 0.1. The tagging is then compared with the correct one, and of course the lambda values giving the best

result are the optimal values for that particular training set. A sample output can be seen in appendix B.

During the test phase the test set is tagged in the same way, using the optimal lambda values, and the result is compared with the correct tags.

In the first evaluation session the program only handled words known from the test set. This was implemented in the optimizing and test phases by simply letting the program skip sentences that contained unknown words.

Two different sets of lambda values produced the same best result for the optimizing set. The values [0.5 0.4 0.1] and [0.7 0.2 0.1] for λ_1, λ_2 and λ_3 respectively each resulted in 95.90% correct tagging of the input. Investigating both these lambda settings for the test set resulted in 95.98% and 95.94% correct.

For the second evaluation session handling of unknown words was implemented as always tagging it as a noun singular. The tag selected from the SUC tagset was "NN UTR SIN IND NOM".

The optimal lambda values were [0.6 0.3 0.1] with a 89.85% correct tagging. The corresponding value for the test set was 89.24%.

To improve the result, the handling of unknown words was modified to use information from the bigram table built during the training. For each of the possible tags in the tagset the bigram table was searched for bigrams starting with that tag. From the collection of bigrams found, the one with the highest count was selected. In this bigram, the second tag was selected as the best candidate for tagging an unknown word following the first tag. This was put into a best-from-bigram table, shown in appendix C.

This time the optimal lambda values were [0.5 0.4 0.1] with 90.25% correct tagging. The corresponding value for the test set vas 89.73%. The output from the latter session is the content of appendix B.

4.2 Interactive session

This section will show a few examples of the program in interactive mode tagging text from the console.

```
> Jag ser en hök flyga.
Jag      PN UTR SIN DEF SUB
ser      VB PRS AKT
en       DT UTR SIN IND
hök      NN UTR SIN IND NOM
flyga    VB INF AKT
.         DL MAD
```

The input line is "Jag ser en hök flyga" ("I see a hawk fly") and the tagger outputs one line for each token with a suggested tag from the SUC tagset. All these words are known from the training, and the tagging is correct.

```
> Jag ser en falk flyga.
Jag      PN UTR SIN DEF SUB
ser      VB PRS AKT
en       DT UTR SIN IND
falk     NN UTR SIN IND NOM (?)
flyga    VB INF AKT
.        DL MAD
```

If we exchange "hök" ("hawk") for "falk" ("falcon") the tagger encounters an unknown word, as denoted by the question mark. In this case the best-from-bigram table is consulted, and a noun is suggested.

```
> Jag ser en glada flyga.
Jag      PN UTR SIN DEF SUB
ser      VB PRS AKT
en       DT UTR SIN IND
glada    JJ POS UTR/NEU PLU …[1]
flyga    VB INF AKT
.        DL MAD
```

Next, we change the bird to "glada" ("kite"). This word is also an adjective in Swedish ("happy/merry"). The training set contains the adjective form, but not the noun form. This results in the tagger finding the sequence Determiner/Singular – Adjective/Plural the most probable, even though it is a rare combination. In the test sentence the sequence is part or two trigrams, [VB DT JJ][2] and [DT JJ VB]. An

[1] This line is truncated. The tag is `JJ POS UTR/NEU PLU IND/DEF NOM`.

[2] These are abbreviations for brevity. The figures refer to the complete tags as shown in the example.

investigation of the tables shows that the count for those trigrams is 0 and 1. The bigram count is 1. If we investigate the corresponding values for the previous example, Determiner/Singular – Noun/Singular, the number of trigrams are 850 for [VB DT NN] and 49 for [DT NN VB]. The bigram count is 8189.

In fact, the word "glada" does appear once as a noun in the SUC corpus. If the tagger is trained on the full corpus it will then tag the test sentence above correctly. This of course a coincident, and the basic problem of incomplete information of the possible tagging for a word remains.

Using the console input it is interesting to test how well the tagger handles ambiguities.

```
> Ser du min min?
Ser     VB PRS AKT
du      PN UTR SIN DEF SUB
min     PS UTR SIN DEF
min     NN UTR SIN IND NOM
?       DL MAD
```

"Ser du min min?" ("Do you see the expression on my face?"). The ambiguous "min" is resolved in context. A sentence that is too difficult is not hard to construct:

```
> Var var var och en en gång?
Var     PN UTR SIN IND SUB/OBJ
var     VB PRT AKT
var     VB PRT AKT
och     KN
en      DT UTR SIN IND
en      DT UTR SIN IND
gång    NN UTR SIN IND NOM
?       DL MAD
```

"Var var var och en en gång?" ("Where were each and everyone once?"). Several words in this sentence have multiple tagging possibilities. Table 1 contains a list of the words, their possible tags and the number of occurrences of the combination word/tag. The tagger fails to catch the multiword expression "var och en" ("each and everyone"). After training the tagger on the full corpus the sentence was tested again:

```
> Var var var och en en gång?
Var     HA
var     VB PRT AKT
var     PN UTR SIN IND SUB/OBJ
och     KN
en      PN UTR SIN IND SUB/OBJ
en      DT UTR SIN IND
gång    NN UTR SIN IND NOM
?       DL MAD
```

This time the expression was noticed. Once again this is a coincidental, and it is easy to find similar cases where the tagger will fail. The difference between training from the train set and full corpus is near a 25% increase of parsed tokens (933629 vs. 1166896), and the resulting improvements are a simple verification of the fact that, in general, the larger the corpus the better.

```
Var
     35 HA
     44 VB PRT AKT
      9 PN UTR SIN IND SUB/OBJ
      3 DT UTR SIN IND
     10 VB IMP AKT
var
   5070 VB PRT AKT
     58 PN UTR SIN IND SUB/OBJ
    132 HA
     13 VB IMP AKT
     49 DT UTR SIN IND
      1 AB
      3 VB INF AKT
      3 NN NEU SIN IND NOM
och
  25562 KN
en
  13139 DT UTR SIN IND
    336 PN UTR SIN IND SUB/OBJ
    315 RG UTR SIN IND NOM
      3 UO
      8 AB
      2 NN UTR SIN IND NOM
gång
    433 NN UTR SIN IND NOM
```

Table 1: Possible tags for words in the example sentence.

5 Conclusion

The goal for this project was to implement a model for probabilistic part-of-speech tagging in a computer program and evaluate it using the SUC corpus. The results of the training are

very encouraging, considering that the tagging algorithm is more or less a straightforward implementation of the equation in Figure 1.

In the introduction the two main problems for part-of-speech tagging were mentioned, ambiguity and unknown words. Resolving ambiguity is part of the design of the probabilistic model, and as has been briefly demonstrated it generally work better the larger the training data is.

Handling of unknown words is a part that has to be added to the model. In the current implementation it is very rudimentary. Judging from the difference in result between the three evaluation sessions there is much to gain in improving this part of the program.

Another problem that appeared during testing is the problem of the program having an incomplete set of tagging options for a token. The tagger will produce a tagging based on available data, of course, but as shown in the example the result would sometimes have been better handled by the unknown word routine. This is a hard problem to solve, and would require some heuristics bridging tagging based on known words and tagging suggested by the handler of unknown words. A different, and complementary, approach is of course trying to avoid the problem by giving the tagger as complete training data as possible.

References

Carlberger, J. and V. Kann. Implementing an efficient part-of-speech tagger. *Software–Practice and Experience*, 29(9):815–832, 1999.

E. Charniak, C. Hendrickson, N. Jacobson, and M. Perkowitz. Equations for part-of-speech tagging. In *11th National Conf. Artificial Intelligence*, :784–789,1993.

A. The SUC tagset.

Category Code	Category
AB	Adverb
DL	Delimiter (Punctuation)
DT	Determiner
HA	Interrogative/Relative Adverb
HD	Interrogative/Relative Determiner
HP	Interrogative/Relative Pronoun
HS	Interrogative/Relative Possessive
IE	Infinitive Marker
IN	Interjection
JJ	Adjective
KN	Conjunction
NN	Noun
PC	Participle
PL	Particle
PM	Proper Noun
PN	Pronoun
PP	Preposition
PS	Possessive
RG	Cardinal Number
RO	Ordinal Number
SN	Subjunction
UO	Foreign Word
VB	Verb

Feature Code	Feature	
UTR	Common (Utrum)	Gender
NEU	Neutre	Gender
MAS	Masculine	Gender
UTR/NEU	Underspecified	Gender
-	Unspecified	Gender
SIN	Singular	Number
PLU	Plural	Number
SIN/PLU	Underspecified	Number
-	Unspecified	Number
IND	Indefinite	Definiteness
DEF	Definite	Definiteness
IND/DEF	Underspecified	Definiteness
-	Unspecified	Definiteness
NOM	Nominative	Case
GEN	Genitive	Case
SMS	Compound	Case
-	Unspecified	Case
POS	Positive	Degree
KOM	Comparative	Degree
SUV	Superlative	Degree
SUB	Subject	Pronoun Form
OBJ	Object	Pronoun Form

SUB/OBJ	Underspecified	Pronoun Form
PRS	Present	Verb Form
PRT	Preterite	Verb Form
INF	Infinitive	Verb Form
SUP	Supinum	Verb Form
IMP	Imperative	Verb Form
AKT	Active	Voice
SFO	S-form	Voice
KON	Subjunctive	Mood
PRF	Perfect	Perfect
AN	Abbreviation	Form

B. Output from test session,

unknown word = best-from-bigram

```
[ 0.0  0.0  1.0 ]    total: 116190, correct:98263, = 84.5710 %
[ 0.0  0.1  0.9 ]    total: 116190, correct:101528, = 87.3810 %
[ 0.0  0.2  0.8 ]    total: 116190, correct:102307, = 88.0515 %
[ 0.0  0.3  0.7 ]    total: 116190, correct:102820, = 88.4930 %
[ 0.0  0.4  0.6 ]    total: 116190, correct:103198, = 88.8183 %
[ 0.0  0.5  0.5 ]    total: 116190, correct:103441, = 89.0275 %
[ 0.0  0.6  0.4 ]    total: 116190, correct:103594, = 89.1591 %
[ 0.0  0.7  0.3 ]    total: 116190, correct:103741, = 89.2857 %
[ 0.0  0.8  0.2 ]    total: 116190, correct:103849, = 89.3786 %
[ 0.0  0.9  0.1 ]    total: 116190, correct:103920, = 89.4397 %
[ 0.0  1.0  0.0 ]    total: 116190, correct:103698, = 89.2486 %
[ 0.1  0.0  0.9 ]    total: 116190, correct:102115, = 87.8862 %
[ 0.1  0.1  0.8 ]    total: 116190, correct:102752, = 88.4345 %
[ 0.1  0.2  0.7 ]    total: 116190, correct:103157, = 88.7830 %
[ 0.1  0.3  0.6 ]    total: 116190, correct:103445, = 89.0309 %
[ 0.1  0.4  0.5 ]    total: 116190, correct:103641, = 89.1996 %
[ 0.1  0.5  0.4 ]    total: 116190, correct:103842, = 89.3726 %
[ 0.1  0.6  0.3 ]    total: 116190, correct:103937, = 89.4543 %
[ 0.1  0.7  0.2 ]    total: 116190, correct:104050, = 89.5516 %
[ 0.1  0.8  0.1 ]    total: 116190, correct:104106, = 89.5998 %
[ 0.1  0.9  0.0 ]    total: 116190, correct:104023, = 89.5284 %
[ 0.2  0.0  0.8 ]    total: 116190, correct:102868, = 88.5343 %
[ 0.2  0.1  0.7 ]    total: 116190, correct:103289, = 88.8966 %
[ 0.2  0.2  0.6 ]    total: 116190, correct:103576, = 89.1436 %
[ 0.2  0.3  0.5 ]    total: 116190, correct:103758, = 89.3003 %
[ 0.2  0.4  0.4 ]    total: 116190, correct:103927, = 89.4457 %
[ 0.2  0.5  0.3 ]    total: 116190, correct:104021, = 89.5266 %
[ 0.2  0.6  0.2 ]    total: 116190, correct:104103, = 89.5972 %
[ 0.2  0.7  0.1 ]    total: 116190, correct:104152, = 89.6394 %
[ 0.2  0.8  0.0 ]    total: 116190, correct:104023, = 89.5284 %
[ 0.3  0.0  0.7 ]    total: 116190, correct:103266, = 88.8768 %
[ 0.3  0.1  0.6 ]    total: 116190, correct:103612, = 89.1746 %
[ 0.3  0.2  0.5 ]    total: 116190, correct:103800, = 89.3364 %
[ 0.3  0.3  0.4 ]    total: 116190, correct:103941, = 89.4578 %
[ 0.3  0.4  0.3 ]    total: 116190, correct:104076, = 89.5740 %
[ 0.3  0.5  0.2 ]    total: 116190, correct:104166, = 89.6514 %
[ 0.3  0.6  0.1 ]    total: 116190, correct:104200, = 89.6807 %
[ 0.3  0.7  0.0 ]    total: 116190, correct:104046, = 89.5482 %
[ 0.4  0.0  0.6 ]    total: 116190, correct:103517, = 89.0929 %
[ 0.4  0.1  0.5 ]    total: 116190, correct:103787, = 89.3252 %
[ 0.4  0.2  0.4 ]    total: 116190, correct:103955, = 89.4698 %
[ 0.4  0.3  0.3 ]    total: 116190, correct:104070, = 89.5688 %
[ 0.4  0.4  0.2 ]    total: 116190, correct:104194, = 89.6755 %
[ 0.4  0.5  0.1 ]    total: 116190, correct:104244, = 89.7186 %
[ 0.4  0.6  0.0 ]    total: 116190, correct:104071, = 89.5697 %
[ 0.5  0.0  0.5 ]    total: 116190, correct:103704, = 89.2538 %
[ 0.5  0.1  0.4 ]    total: 116190, correct:103927, = 89.4457 %
[ 0.5  0.2  0.3 ]    total: 116190, correct:104051, = 89.5525 %
[ 0.5  0.3  0.2 ]    total: 116190, correct:104192, = 89.6738 %
[ 0.5  0.4  0.1 ]    total: 116190, correct:104258, = 89.7306 %
[ 0.5  0.5  0.0 ]    total: 116190, correct:104103, = 89.5972 %
[ 0.6  0.0  0.4 ]    total: 116190, correct:103817, = 89.3511 %
[ 0.6  0.1  0.3 ]    total: 116190, correct:104026, = 89.5309 %
[ 0.6  0.2  0.2 ]    total: 116190, correct:104157, = 89.6437 %
```

```
[ 0.6  0.3  0.1 ]    total: 116190, correct:104247, = 89.7211 %
[ 0.6  0.4  0.0 ]    total: 116190, correct:104089, = 89.5852 %
[ 0.7  0.0  0.3 ]    total: 116190, correct:103917, = 89.4371 %
[ 0.7  0.1  0.2 ]    total: 116190, correct:104118, = 89.6101 %
[ 0.7  0.2  0.1 ]    total: 116190, correct:104207, = 89.6867 %
[ 0.7  0.3  0.0 ]    total: 116190, correct:104086, = 89.5826 %
[ 0.8  0.0  0.2 ]    total: 116190, correct:103957, = 89.4716 %
[ 0.8  0.1  0.1 ]    total: 116190, correct:104148, = 89.6359 %
[ 0.8  0.2  0.0 ]    total: 116190, correct:104065, = 89.5645 %
[ 0.9  0.0  0.1 ]    total: 116190, correct:103966, = 89.4793 %
[ 0.9  0.1  0.0 ]    total: 116190, correct:104023, = 89.5284 %
[ 1.0  0.0  0.0 ]    total: 116190, correct:103242, = 88.8562 %
```

C. best-from-bigram table

Collected from the bigram table, built during training.The known tag points to the most common tag following it. Sorted by the number of occurrences in the train set.

```
 14442: NN UTR SIN IND NOM      -->>    PP
 12567: PP      -->>    NN UTR SIN DEF NOM
  8853: JJ POS UTR SIN IND NOM  -->>    NN UTR SIN IND NOM
  8830: IE      -->>    VB INF AKT
  8189: DT UTR SIN IND  -->>    NN UTR SIN IND NOM
  7524: NN UTR SIN DEF NOM      -->>    PP
  7235: VB PRS AKT      -->>    AB
  6547: <s>     -->>    PP
  6227: NN UTR PLU IND NOM      -->>    PP
  6160: JJ POS UTR/NEU PLU IND/DEF NOM  -->>    NN UTR PLU IND NOM
  5912: AB      -->>    PP
  5876: PM NOM  -->>    PM NOM
  5761: NN NEU SIN IND NOM      -->>    PP
  5052: PN UTR SIN DEF SUB      -->>    VB PRT AKT
  4524: DL MID  -->>    KN
  4377: PN NEU SIN DEF SUB/OBJ  -->>    VB PRS AKT
  3680: JJ POS UTR/NEU SIN DEF NOM      -->>    NN UTR SIN DEF NOM
  3678: HP - - -        -->>    VB PRS AKT
  3601: VB PRT AKT      -->>    AB
  3595: KN      -->>    NN UTR SIN IND NOM
  3546: DT NEU SIN IND  -->>    NN NEU SIN IND NOM
  3378: VB INF AKT      -->>    PP
  3329: DT UTR SIN DEF  -->>    JJ POS UTR/NEU SIN DEF NOM
  3213: JJ POS NEU SIN IND NOM  -->>    NN NEU SIN IND NOM
  3009: NN NEU SIN DEF NOM      -->>    PP
  2857: DL MAD  -->>    DL MID
  2846: PL      -->>    PP
  2563: RG NOM  -->>    NN UTR PLU IND NOM
  2510: NN NEU PLU IND NOM      -->>    PP
  2086: VB PRS SFO      -->>    PP
  2083: SN      -->>    PN UTR SIN DEF SUB
  2067: PS UTR SIN DEF  -->>    NN UTR SIN IND NOM
  2007: NN UTR PLU DEF NOM      -->>    PP
  1988: DT UTR/NEU PLU DEF      -->>    JJ POS UTR/NEU PLU IND/DEF NOM
  1717: DL PAD  -->>    DL MAD
  1683: DT NEU SIN DEF  -->>    JJ POS UTR/NEU SIN DEF NOM
  1513: UO      -->>    UO
  1490: AB POS  -->>    PP
  1456: VB INF SFO      -->>    PP
  1434: PN UTR PLU DEF SUB      -->>    VB PRS AKT
  1393: VB SUP AKT      -->>    PP
  1377: PN UTR SIN IND SUB      -->>    VB PRS AKT
  1339: PN UTR/NEU SIN/PLU DEF OBJ      -->>    PP
  1250: PC PRF UTR SIN IND NOM  -->>    NN UTR SIN IND NOM
  1232: NN UTR SIN DEF GEN      -->>    NN UTR SIN IND NOM
  1176: HA      -->>    PN UTR SIN DEF SUB
  1138: PM GEN  -->>    NN UTR SIN IND NOM
   970: PN UTR/NEU PLU DEF SUB  -->>    VB PRS AKT
   924: PC PRS UTR/NEU SIN/PLU IND/DEF NOM      -->>    NN UTR SIN IND NOM
   923: PS UTR/NEU SIN/PLU DEF  -->>    NN UTR SIN IND NOM
   893: VB PRT SFO      -->>    PP
   871: PS NEU SIN DEF  -->>    NN NEU SIN IND NOM
   849: PC PRF UTR/NEU PLU IND/DEF NOM  -->>    NN UTR PLU IND NOM
   821: VB SUP SFO      -->>    PP
```

```
725: JJ KOM UTR/NEU SIN/PLU IND/DEF NOM      -->>    NN UTR SIN IND NOM
721: NN NEU PLU DEF NOM      -->>    PP
681: IN      -->>    DL MID
673: PS UTR/NEU PLU DEF      -->>    NN UTR PLU IND NOM
619: PN UTR SIN DEF OBJ      -->>    DL MAD
607: PN UTR SIN DEF SUB/OBJ  -->>    VB PRS AKT
574: JJ POS UTR/NEU SIN/PLU IND/DEF NOM      -->>    NN UTR SIN IND NOM
557: PN NEU SIN IND SUB/OBJ  -->>    PP
517: NN NEU SIN DEF GEN      -->>    NN UTR SIN IND NOM
502: HP NEU SIN IND  -->>    VB PRS AKT
481: JJ POS UTR/NEU PLU IND NOM      -->>    NN UTR PLU IND NOM
444: PC PRF NEU SIN IND NOM  -->>    NN NEU SIN IND NOM
444: DT UTR/NEU PLU IND      -->>    NN UTR PLU IND NOM
436: NN AN   -->>    RG NOM
433: PC PRF UTR/NEU SIN DEF NOM      -->>    NN UTR SIN DEF NOM
425: PN UTR SIN IND SUB/OBJ  -->>    PP
417: AB KOM  -->>    KN
385: RO NOM  -->>    NN UTR SIN DEF NOM
359: NN UTR - - SMS  -->>    KN
343: NN UTR PLU DEF GEN      -->>    NN UTR SIN IND NOM
335: DT UTR/NEU SIN IND      -->>    NN UTR SIN IND NOM
312: DT UTR/NEU PLU IND/DEF  -->>    NN UTR PLU IND NOM
305: DT UTR/NEU SIN/PLU IND  -->>    NN UTR SIN IND NOM
301: PN UTR/NEU PLU DEF OBJ  -->>    DL MAD
290: JJ SUV UTR/NEU SIN/PLU DEF NOM  -->>    NN UTR SIN DEF NOM
289: PN UTR/NEU PLU IND SUB/OBJ      -->>    PP
285: JJ POS MAS SIN DEF NOM  -->>    NN UTR SIN DEF NOM
281: JJ POS UTR SIN IND/DEF NOM      -->>    NN UTR SIN IND NOM
243: AB SUV  -->>    PP
200: NN UTR SIN IND GEN      -->>    NN UTR SIN IND NOM
196: AB AN   -->>    RG NOM
175: VB IMP AKT      -->>    AB
156: RG UTR SIN IND NOM      -->>    NN UTR SIN IND NOM
146: PN UTR PLU DEF OBJ      -->>    PP
141: NN NEU - - SMS  -->>    KN
139: PN UTR/NEU PLU DEF SUB/OBJ      -->>    VB PRS AKT
131: NN UTR PLU IND GEN      -->>    NN UTR SIN IND NOM
130: NN NEU SIN IND GEN      -->>    NN UTR SIN IND NOM
126: HD UTR SIN IND  -->>    NN UTR SIN IND NOM
115: JJ POS NEU SIN IND/DEF NOM      -->>    NN NEU SIN IND NOM
108: NN NEU PLU DEF GEN      -->>    NN UTR SIN IND NOM
104: DT UTR SIN IND/DEF      -->>    NN UTR SIN IND NOM
103: RG UTR/NEU SIN DEF NOM  -->>    NN UTR SIN DEF NOM
 95: RG NEU SIN IND NOM      -->>    NN NEU SIN IND NOM
 92: JJ SUV UTR/NEU SIN/PLU IND NOM  -->>    PP
 92: HP UTR SIN IND  -->>    VB PRS AKT
 86: HD UTR/NEU PLU IND      -->>    NN UTR PLU IND NOM
 75: NN NEU PLU IND GEN      -->>    NN UTR SIN IND NOM
 59: KN AN   -->>    PM NOM
 58: RG SMS  -->>    KN
 56: JJ SUV UTR/NEU PLU DEF NOM      -->>    PP
 56: HD NEU SIN IND  -->>    NN NEU SIN IND NOM
 52: HS DEF  -->>    NN UTR SIN IND NOM
 33: NN - - - -      -->>    DL MAD
 33: JJ POS MAS SIN DEF GEN  -->>    NN UTR SIN IND NOM
 32: HP UTR/NEU PLU IND      -->>    VB PRS AKT
 31: PM SMS  -->>    KN
 30: NN UTR - - -    -->>    PP
 30: DT NEU SIN IND/DEF      -->>    NN NEU SIN IND NOM
 27: PC PRF MAS SIN DEF NOM  -->>    NN UTR SIN DEF NOM
```

```
27: NN - - - SMS    -->>    KN
25: JJ POS UTR - - SMS      -->>    KN
23: VB KON PRS AKT  -->>    PN UTR/NEU SIN/PLU DEF OBJ
22: PN MAS SIN DEF SUB/OBJ  -->>    VB PRT AKT
19: VB AN   -->>    PM NOM
14: VB KON PRT AKT  -->>    JJ POS NEU SIN IND NOM
14: JJ SUV MAS SIN DEF NOM  -->>    NN UTR SIN IND NOM
14: JJ POS UTR/NEU PLU IND/DEF GEN  -->>    NN UTR SIN IND NOM
14: AB SMS  -->>    KN
13: DT MAS SIN DEF  -->>    NN UTR SIN IND NOM
12: PL SMS  -->>    KN
11: JJ POS UTR/NEU SIN DEF GEN      -->>    NN UTR SIN IND NOM
 9: JJ AN   -->>    NN UTR SIN IND NOM
 8: RO MAS SIN IND/DEF NOM  -->>    NN UTR SIN IND NOM
 8: RG MAS SIN DEF NOM      -->>    HP - - -
 6: JJ SUV UTR/NEU PLU IND NOM      -->>    NN UTR PLU IND NOM
 5: JJ POS UTR/NEU SIN/PLU IND NOM  -->>    NN UTR PLU IND NOM
 4: RO GEN  -->>    NN UTR SIN IND NOM
 4: PC PRF UTR/NEU PLU IND/DEF GEN  -->>    NN UTR SIN IND NOM
 4: DT UTR/NEU SIN DEF      -->>    NN UTR SIN DEF NOM
 4: DT MAS SIN IND  -->>    NN UTR SIN IND NOM
 3: VB SMS  -->>    KN
 3: RG GEN  -->>    NN UTR SIN IND NOM
 3: NN NEU - - -    -->>    DL MAD
 3: JJ POS UTR/NEU - - SMS  -->>    KN
 2: RO SMS  -->>    KN
 2: PP AN   -->>    NN UTR SIN IND NOM
 2: PC PRF UTR/NEU SIN DEF GEN      -->>    NN UTR SIN IND NOM
 2: PC PRF MAS SIN DEF GEN  -->>    NN UTR SIN IND NOM
 2: JJ KOM UTR/NEU SIN/PLU IND/DEF SMS      -->>    KN
 2: JJ KOM UTR/NEU SIN/PLU IND/DEF GEN      -->>    NN UTR SIN IND NOM
 1: VB KON PRT SFO  -->>    DT NEU SIN IND
 1: VB IMP SFO      -->>    PP
 1: PC PRS UTR/NEU SIN/PLU IND/DEF GEN      -->>    NN UTR SIN DEF NOM
 1: PC PRF UTR SIN IND GEN  -->>    NN UTR PLU IND NOM
 1: PC AN   -->>    NN UTR SIN IND NOM
 1: JJ SUV MAS SIN DEF GEN  -->>    JJ POS UTR/NEU SIN DEF NOM
 1: JJ POS UTR SIN IND GEN  -->>    NN UTR SIN IND NOM
 1: HP NEU SIN IND SMS      -->>    KN
 1: DT AN   -->>    RG NOM
```