Tiered Tagging Revisited

Dan Tufiş[1,2], Liviu Dragomirescu[1]

[1]Research Institute for Artificial Intelligence of the Romanian Academy, [2]University „A.I. Cuza” of Iaşi
Calea „13 Septembrie”, no. 13, PO 050711, Bucharest
tufis@racai.ro, liviud@racai.ro

Abstract

In this paper we describe a new baseline tagset induction algorithm, which unlike the one described in previous work is fully automatic and produces tagsets with better performance than before. The algorithm is an information lossless transformation of the MULTEXT-EAST compliant lexical tags into a reduced tagset that can be mapped back on the lexicon tagset fully deterministic. From the baseline tagsets, a corpus linguist, expert in the language in case, may further reduce the tagsets taking into account language distributional properties. As any further reduction of the baseline tagsets assumes losing information, adequate recovering rules should be designed for ensuring the final tagging in terms of lexicon encoding.

Introduction

Tiered tagging (TT) is a very effective technique (Tufiş, 1999) which allows accurate morpho-syntactic tagging with large lexicon tagsets and requires reasonable-sized training data. The basic idea consists of using for proper tagging a hidden tagset, for which training data is sufficient, and a post-processing phase transforms the tags from the hidden tagset into the more informative tags from the lexicon tagset. The hidden tagset is obtained in two steps, first eliminating encoding redundancy (this is called the baseline tagset; we will elaborate on this below) and then further eliminating some attributes which could be recovered by a post-tagging processing. A major problem in TT is designing the hidden tagset so that the recovery of the left-out information from the tags of the lexicon tagset could be deterministically recovered. In (Tufiş, 2000) we largely discuss the experiments and their evaluation for Romanian, where the initial lexicon tagset contained almost 1000 tags, and the hidden tagset only 92 (plus 10 punctuation tags). In (Tufiş et al., 2000; Varadi, 2002, Oravecz and Dienes 2002) there are presented the results of TT applied to Hungarian, a very different language, with even more spectacular results. Hinrichs and Trushkina (2003) described the use of TT with very promising results for German. The main limitation of the previous design algorithm is that the procedure generating the baseline tagset relies only on lexicon information and does not take into account frequency of the lexical items in running texts. It also requires intensive interaction with the human expert in evaluation of the proposed tagsets.

Lexical Encoding Normalisation

The morpho-syntactic descriptions (MSD) defined by the MULTEXT-East lexical encoding are provided as strings, using a linear encoding. In this notation, the position in a string of characters corresponds to an attribute, and specific characters in each position indicate the value for the corresponding attribute. That is, the positions in a string of characters are numbered 0, 1, 2, etc., and are used in the following way:

- the character at position 0 encodes part-of-speech;
- each character at position 1, 2,...,n, encodes the value of one attribute (person, gender, number, etc.), using the one-character code;
- if an attribute does not apply, the corresponding position in the string contains the special marker '-' (hyphen); the trailing hyphens are omitted.

Each word-form in a lexicon compliant with the MULTEXT-East lexical specifications is associated with its lemma form and the appropriate MSD. However, in most cases, the word-form and the associated MSD are informationally redundant. This means that the word-form and a few attribute-value pairs from the corresponding MSD (we call them *the determinant*) uniquely determine the rest of the attribute-value pairs (*the dependant*). By dropping the dependant attributes, provided this does not reduce the cardinal of *ambiguity classes* (see Tufis (1999)), several initial tags are merged into fewer and more general tags. This way the cardinality of the tagset is reduced, with benefic results on the tagging accuracy even with limited training data. If we consider an attribute *orth* the value of which is a given word-form, then the *orth* attribute is one element of the determinant. Since the attributes and their values depend on the grammar category of the word-forms, we will have different determinants and dependants for each part of speech. Thus a natural option is to include the part of speech (the attribute at position 0 in the MSD encoding) in the respective determinant. Unfortunately, finding the rest of the attributes in the determinants of an MSD encoding has not a unique solution. Although for any part of speech one can identify the smallest set of determinant attributes, it is not the case that the smallest determinant (and implicitly the smallest baseline tagset) necessarily ensures the best tagging accuracy. Thus, any refinement (therefore, enlargement) of the smallest baseline tagset which is still deterministically transformable into the MSD tagset, will be called also a baseline tagset.

The Data and the Validation Method

For our experiments we used the CONCEDE edition (Erjavec, 2001) of the parallel corpus "1984" and the associated word-forms lexicons. These resources were produced during the Multext-East and Concede European projects. The tagset-design algorithm takes as input a word-form lexicon and a corpus encoded according to CES-specifications used by the Multext-East consortium. We ran the algorithm and generated tagsets for English and five East-European languages: Czech, Estonian, Hungarian, Romanian and Slovene. In order to find the baseline tagset with the best properties (that is ensuring the best tagging results) each generated tagset (see next section) is used for building a language model and tagging unseen data. We used a ten-fold validation procedure (using for training 9/10 of the corpus and the remaining

1/10 of the corpus for evaluation and averaging the accuracy results).

The Algorithm

Due to space limitations we direct the interested reader to our previous paper (Tufis, 2000) where we define and motivate the basic algorithm. For the sake of clarity we resume part of the definitions:

MSD- a morpho-syntactic descriptor as discussed before; the set of all MSDs is called *MSD-set*

Ambiguity class(AMB)-the set of all possible MSDs attached in the lexicon to an ambiguous word-form; several words have the same ambiguity class;

CTAG-a reduced morpho-lexical code, generalizing 2 or more MSDs; the set of all CTAGs is called *CTAG-set;* we should note that to any MSD it corresponds a unique CTAG; in (Tufis, 2000) we showed that when MSDs are replaced by their corresponding CTAGs in all AMBs, only a limited number of words should have their ambiguity classes smaller than initially. We called this property of the CTAG-set the *CTAG-set to MSD-set recoverability*, formally described by expression (1) with the following notations:

W_i represents a word from the lexicon (Lex),

T_i represents a CTAG assigned to W_i,

MSD_k represents a tag from the MSD-tagset,

$AMB(W_k)$ represents the ambiguity class of the word W_k in terms of MSDs (as encoded in Lex),

MAP is an application that maps each T_i onto a subset of MSD-set and

|X| represents the number of elements of the set X.

(1)$\forall$ Ti $\in$ CTAG-set, MAP(Ti) = {MSD1…MSDk} $\subset$ MSD-tagset,

$\forall W_k \in$ Lex & $AMB(W_k) = \{MSD_{k1} \ldots MSD_{kn}\}$ $\subset$MSD-tagset $\Rightarrow$

$$\left|\mathrm{MAP(Ti)} \bigcap \mathrm{AMB(Wk)}\right| = \begin{cases} 1 & \text{for more than 90\% cases} \\ >1 & \text{for less than 10\% cases} \end{cases}$$

When the set intersection above results always in a unique MSD, the recoverability is fully deterministic and the CTAG-set with this property is called a baseline tagset. Otherwise, the recoverability is partial, the CTAG-set is called a hidden tagset. In a text tagged with a hidden tagset, most of the tags may be uniquely turned into the appropriate MSDs. For those tagged words were the recoverability is partial, a post-tagging processing is required to chose the relevant MSD from the intersection set described in (1). Allowing a small percentage of indeterminism in the MSD-tagset recoverability procedure above, one can dramatically reduce the size of the CTAG-set. In (Tufis, 1999, 2000) we showed that allowing 10% of the words in the lexicon to remain ambiguous after the recoverability procedure (1) was applied, required only 18 local grammar rules (regular expressions) to disambiguate the partially recovered tags. Yet, the size of the hidden tagset was reduced to half as compared to the corresponding baseline tagset. Previously, the decision on which were the permissible ambiguities left in the CTAG-set relied exclusively on the MSD lexicon thus, not taking into account the frequency of the words that might remain ambiguous after the computation described in (1). In the present algorithm the frequency of words in the corporus is a significant design parameter. More exactly, instead of counting how many words of the dictionary would be partially disambiguated when using a hidden tagset we compute a score for the ambiguity classes characterizing these words based on their frequency in the corpus. If further reducing a baseline tagset creates ambiguity in the recovering process for a number of AMBs and these AMBs are characteristic to very rare words, then the licensing the reduction should be almost harmless even without recovering rules.

The definitions below used in describing the algorithm:

$\mathbf{S_{AC}(AMB_i)} = \Sigma_{w \in AMB_i} RF(w) \leq$ *threshold*: the frequency score of an ambiguity class AMB_i where:

RF(w) is the relative frequency in a training corpus of the word *w* characterized by the AMB_i ambiguity class and *threshold* is a designer parameter (a null value would corespond to the baseline tagset); we compute these score only for AMBs characterizing the words the CTAGs of which (members of a hidden tagset) might not be fully recovered by the procedure (1);

$\mathbf{f_{AC}(T_i)} = \{(AMB_{ik}, S_{AC}(AMB_{ik}) | AMB_{ik} \cap MAP(T_i) \neq \emptyset\}$: the set of pairs of ambiguity classes and their scores so that each AMB containing at least one MSD in $MAP(T_i)$;

$\mathbf{pen(T_i, AMB_j)} = S_{AC}(AMB_j)$ if card $|AMB_j \cap MAP(T_i)| > 1$ and 0 otherwise; this is a penalty for a ctag labeling any words characterized by AMB_i and which cannot be deterministically turned into an unique MSD. We should note that the same ctag labeling a word characterized by a different AMB_j might be deterministically recovered to the appropriate MSD.

$\mathbf{PEN(T_i)} = \Sigma(pen(T_i, AMB_j) | AMB_j \in f_{AC}(T_i))$

DTR = $\{A_{Pi}\}$ = a determinant set of attributes: P is a part of speech and the indexes i represent the attribute at position i in the MULTEXT-East encoding of P; for instance A_{V4} represents the *PERSON* attribute of the verb. The attributes in DTR are not subject to elimination in the baseline tagsets generation. Because the search space of the algorithm is structured according to the determinant attributes for each part of speech, the running time significantly decreases as DTRs are larger.

POS(code)=the part of speech in a MSD or a ctag code.

The input data for the algorithm is the word-form lexicon (MSD encoded) and the corpus. The output is a baseline CTAGset. The CTAGSET-DESIGN algorithm is a trial and error procedure that generates all possible baseline tagsets and with each of them constructs language models which are used in the tagging of unseen texts. The central part of the algorithm is the procedure CORE, which will be briefly commented.

```
procedure CTAGSET-DESIGN (LEX, corpus;CTAG-set) is:
 MSD-set = GET-MSD-SET (Lex)
 AMB = GET-AMB-CLASSES (Lex)
 DTR = {POS($MSD_i$)}, i=1..|MSD-set|
 MATR = GET-ALL-ATTRIBUTES (MSD-set)
 T= {} ; a temporary CTAG-set
 for each $AMB_i$ in AMB
   execute COMPUTE-$S_{AC}$(corpus, $AMB_i$)
 end for
 while DTR $\neq$ MATR
   for each attribute $A_i$ in MATR \ DTR
     D=DTR $\cup$ {$A_i$} ; temporary DTR
     T=T $\cup$ execute CORE ({$(AMB_i, S_{AC}(AMB_i))^+$})
   end for
   $A_k$ = execute FIIND-THE-BEST(T)
   DTR= DTR $\cup$ {$A_k$} & T={}
 end while
 CTAG-set=KEEP-ONLY-ATT-IN-DTR (MSD-set, DTR)
 ; values of attributes which are not in DTR are turned into
 ;'+' (redundant) in all MSDs and duplicates are removed.
end procedure
procedure FIND-THE-BEST ({(ctagset, DTR)$^+$}; Attr) is:
rez = {}
  for each ctagset in {(ctagset$_i$, DTR$_i$)$^+$}
   tmp-corpus = execute MSD2CTAG(corpus, ctagset$_i$)
   train = 9/10*tmp-corpus & test = tmp-corpus \ train
   LM = execute BUILD-LANGUAGE-MODEL(train)
   Prec = execute EVAL (tagger, LM, test)
   rez = rez $\cup$ {(|ctagset$_i$|, Prec$_i$, DTR$_i$)}
  end for
  Attr = LAST-ATTRIB-OF-DTR$_I$-WITH-MAX-PREC$_I$-IN(rez)
end procedure
procedure CORE ({(AMB$_i$ , S$_{AC}$(AMB$_i$))$^+$}, DTR;
                              ({(T$_i$, MAP(T$_i$))$^+$}, DTR))
  T$_i$ = MSD$_i$  i=1..|MSD-set|
  MAP(T$_i$)={MSD$_i$} & AMB(T$_i$)=f$_{AC}$(T$_i$)
  TH = threshold & CTAGSET={T$_i$}
  {repeat until no attribute can be eliminated
      for each T$_i$ in CTAGSET
       {START:
        for each attribute A$_{jk}$ of T$_i$ so that A$_{jk}$$\notin$DTR
          if newT$_i$ is obtained from T$_i$ by deleting A$_{jk}$
          1) if newT$_i$ $\notin$ CTAGSET then
               CTAGSET=(CTAGSET\{T$_i$})$\cup${newT$_i$}
               continue from START
          2) else if newT$_i$ =T$_n$$\in$ CTAGSET then
                MAP(newT$_i$) = MAP(T$_n$) $\cup$ MAP(T$_i$)
               AMB (newT$_i$) = AMB(T$_n$) $\cup$ AMB(T$_i$)
                if PEN(newT$_i$) = 0 then
                 CTAGSET=(CTAGSET\{T$_n$,T$_i$})$\cup${newT$_i$}
                 continue from START
               else
               3) if PEN(newT$_i$) $\leq$ THR then
                   mctag=T$_i$ & matrib=A$_{ik}$ & TH=PEN(newT$_i$)
                   continue from START
        end for}
      end for}
     { 4) eliminate matrib from mctag and obtain newT$_i$
       for each T$_n$ în CTAGSET so that T$_n$ = newT$_i$
         MAP(newT$_i$) = MAP(T$_n$) $\cup$ MAP(mctag)
         AMB (newT$_i$) = AMB(T$_n$) $\cup$ AMB(mctab)
         CTAGSET=(CTAGSET\{mctag,T$_n$})$\cup${newT$_i$}
         TH=threshold   } ;  closing 4)
  end repeat }
end procedure
```

The procedures BUILD-LANGUAGE-MODEL and EVAL were not detailed as they are standard procedures ensured by any tagging platform. All the other procedures (COMPUTE-S_{AC}, KEEP-ONLY-ATT-IN-DTR, MSD2TAG, and LAST-ATTRIB-OF-DTR$_I$-WITH-MAX-PREC$_I$-IN) not shown are simple transformation scripts.

The way the MAP and AMB sets are computed in step **2)** of the procedure CORE could generate non-determinism in MSD recovery process (i.e. PEN(newT$_i$) $\neq$ 0). Step **3)** recognizes the introduced non-determinism and provided the generated ambiguity is acceptable, it stores the dispensable attribute and the current ctag which are eliminated in the step **4)**.

In order to obtain the optimal CTAGSET one should be able to use a large training corpus (where all the MSDs defined in the lexicon are present) and to run the algorithm on all the possible DTRs. Unfortunately this was not the case of our multilingual data. The MSDs used in the "1984" corpus represent only a fraction of the MSDs present in the word-form lexicons of each language. Most of the ambiguous words in the corpus occur only with a subset of their ambiguity classes. It is not clear whether some of the morpho-syntactic codes would be seen in a larger corpus or they are theoretically potential interpretations, hard to be found in any reasonably large corpus. Heuristically, we assumed that the unseen MSDs of un ambiguity class were rare events, so they were given a happax legomenon status in the computation of the scores $S_{AC}(AMB_j)$. Various other heuristics were used in order to speed up this heavy computation algorithm (e.g. generating of the baseline tagset for Slovene or Czech required more than 80 hours).

Evaluation results

We performed experiments with six languages out of seven[1] represented in the parallel corpus "1984": Romanian (RO), Slovene (SI), Hungarian (HU), English (EN), Czech (CZ) and Estonian (ET). For each language we computed three baseline tagsets: the minimal one (smallest sized DTR), the best performing one (the one which produced the best precision in tagging) and the CTAGSET with the precision comparable to the MSD tagset.

Lang.	**MSD**		**Minimal CTAGSET**		**Best prec. CTAGSET**		**CTAGSET with prec. close to MSD**	
	No.	**Prec.**	**No.**	**Prec.**	**No.**	**Prec.**	**No.**	**Prec.**
RO^{sc1}	615	95.8	56	95.1	174	96.0	81	95.8
RO^{sc2}	615	97.5	56	96.9	205	97.8	78	97.6
SI ^{sc1}	2083	90.3	385	89.7	691	90.9	585	90.4
SI ^{sc2}	2083	92.3	404	91.6	774	93.0	688	92.5
HU ^{sc1}	618	94.4	44	94.7	84	95.0	44	94.7
HU ^{sc2}	618	96.6	128	96.6	428	96.7	112	96.6
EN ^{sc1}	133	95.5	45	95.5	95	95.8	52	95.6
EN ^{sc2}	133	95.9	45	95.9	61	96.3	45	95.9
CZ ^{sc1}	1428	89.0	291	88.9	735	90.2	319	89.2
CZ ^{sc2}	1428	91.8	301	91.0	761	92.5	333	91.8
ET ^{sc1}	639	93.0	208	92.8	355	93.5	246	93.1
ET ^{sc2}	639	93.4	111	92.8	467	93.8	276	93.5

Table 1: Baseline tagsets for 6 languages

[1] As Jan Hajic (2000) mentions, the Bulgarian tagging was not hand-validated and is unreliable. Although he used the same data as us, the quantitative data seem to differ. An explanation might be that the CONCEDE edition of "1984" has been significantly modified (presumably improved); yet, we found in the English part of the corpus several mistakes;

We considered two scenarios, differing in whether the tagger had to deal or not with unknown words; in both scenarios, the ambiguity classes were computed from the large word-form lexicons.

1) the tagger lexicon was generated from the training corpus; in this scenario, words that appeared only in the test part of the corpus were unknown for the tagger;
2) the unigram lexicon was computed from the entire corpus AND the word-form lexicon (with the entries not appearing in the corpus been given a lexical probability corresponding to a single occurrence); in this scenario, no unknown words were faced by the tagger in tagging the test data.

The results are summarized in Table 1. We do agree with (Hajic, 2000) that "it is not unreasonable to assume that a larger dictionary[2] exists, which can help to obtain a list of possible tags for each word-form in the text data". Therefore we consider the scenario no. 2 more relevant than the first one.

Implementation and Conclusions

The algorithm is implemented in PERL and for the evaluation of the generated baseline tagsets we used Brant's TnT trigram HMM tagger (Brants, 1998). However, the algorithm is independent on the tagger or tagging method (HMM, ME, rule-based, etc), provided the input/output format is the same. The programs and the baseline tagsets can be freely obtained with an e-mail request sent to the first author.

There are some interesting observations concerning the results in Table 1:

- the tagging accuracy with the "Best precision CTAGSET" for Romanian was only 0.68% inferior to the tagging precision reported in (Tufis, 2000) where the hidden tagset went with 18 recovering rules;
- for all languages the "Best precision CTAGSET" (scenario 2) is much smaller than the MSD tagset, it is fully recoverable to the MSD annotation and it is always better performing than the MSD tagset; it seems unreasonable to use the MSD tagset when significantly smaller tagsets in a tiered tagging approach would ensure the same information content in the final results;
- using the baseline tagsets instead of MSD-sets in language modeling should result in more reliable language models since the data sparseness effect is significantly diminished; the small differences in precision shown in Table 1 between tagging with the MSD-set and any baseline tagset should not be misleading: it is very likely that on different register texts, the performance difference will be much larger.
- currently the algorithm is very time consuming but if can be improved in various ways.
- as mentioned several times, the tagsets produced by the algorithm represent a baseline. To take full advantage of the tiered tagging approach power, one should go further with the reduction of the baseline tagset towards the hidden tagset. The way our algorithm is implemented, suggests that the best approach in designing the hidden tagset is use as DTRs the attributes preserved in the "Best Precision CTAGSET" out of which there will be removed a few more attributes. The threshold parameter (procedure CORE) which controls the frequency of not fully disambiguated words in the tagged text should be empirically determined. To obtain the hidden tagset mentioned in (Tufis, 2000) we used a threshold of 0.027.

Acknowledgements

The work reported here was carried with within the European project Balkanet, no. IST-2000 29388 and support from the Romanian Ministry of Education and Research under the CORINT programme.

References

Brants, T. (1998). TnT - A Statistical Part-of-Speech Tagger. Instalation and User Guide. University of Saarland, Computational Linguistics, 1998.

Erjavec, T. (ed.) (2001). Specifications and Notations for MULTEXT-East Lexicon Encoding. Edition Multext-East/Concede, http://nl.ijs.si/ME/V2/msd/html/, 210 p.

Erjavec, T. (2001). Harmonised Morphosyntactic Tagging for Seven Languages and Orwell's 1984. In Proceedings of the 6th Natural Language Processing Pacific Rim Symposium, (pp. 487--492), Tokyo.

Hajic, J. (2000). Morphological Tagging: Data vs. Dictionaries. In Proceedings of the ANLP/NAACL 2000, Seatle.

Hinrichs, E. and Trushkina, J. (2002). Forging Agreement: Morphological Disambiguation of Noun Phrases. In Proceedings of the Workshop Treebanks and Linguistic Theories, (pp. 78--95), Sozopol.

Oravecz, C. and Dienes, P. (2002). Efficient Stochastic Part-of-Speech tagging for Hungarian. In Proceedings of the Third International Conference on Language Resources and Evaluation, (pp. 710—717), Las Palmas.

Tufiş, D. (1999). Tiered Tagging and Combined Classifiers. In F. Jelinek, E. Nöth (eds) Text, Speech and Dialogue, Lecture Notes in Artificial Intelligence 1692, Springer, (pp. 28--33).

Tufiş, D.(2000). Using a Large Set of Eagles-compliant Morpho-Syntactic Descriptors as a Tagset for Probabilistic Tagging. In Proceedings of the Second International Conference on Language Resources and Evaluation, (pp. 1105—1112), Athens.

Tufiş, D., Dienes, P., Oravecz, C., Váradi T.(2000). Principled Hidden Tagset Design for Tiered Tagging of Hungarian. In Proceedings of the Second International Conference on Language Resources and Evaluation, (pp. 1421--1426), Athens.

Varadi, T. (2002). The Hungarian National Corpus. In Proceedings of the Third International Conference on Language Resources and Evaluation, (pp. 385--396), Las Palmas.

[2] Larger than the one derived from the training corpus.