A Support Tool for Tagset Mapping

Simone Teufel
IMS-CL Institut für Maschinelle Sprachverarbeitung – Computerlinguistik
Universität Stuttgart

Workshop SIGDAT (EACL95)

Abstract

Many different tagsets are used in existing corpora; these tagsets vary according to the objectives of specific projects (which may be as far apart as robust parsing vs. spelling correction). In many situations, however, one would like to have uniform access to the linguistic information encoded in corpus annotations without having to know the classification schemes in detail. This paper describes a tool which maps unstructured morphosyntactic tags to a constraint-based, typed, configurable specification language, a "standard tagset". The mapping relies on a manually written set of mapping rules, which is automatically checked for consistency. In certain cases, unsharp mappings are unavoidable, and noise, i.e. groups of word forms *not* conforming to the specification, will appear in the output of the mapping. The system automatically detects such noise and informs the user about it.

The tool has been tested with rules for the UPenn tagset (Marcus et al. 92) and the SUSANNE tagset (Garside, Leech, Sampson 87), in the framework of the EAGLES[1] validation phase for standardised tagsets for European languages.

1 Motivation

Tagsets used in existing corpora have usually been designed to satisfy the needs of specific projects. A tagset used for robust parsing will tend to stress distributional properties, whereas a corpus within a lexical resource specially designed for human interaction (which might include a human oriented dictionary) will most likely distinguish word classes along traditional linguistic lines.

The tool described in this paper performs tagset mapping with manually written rules to introduce a standardised morphosyntactic tagset. Standardisation of tagsets has been a goal of some contemporary projects (e.g. (EAGLES 94) and the Text Encoding Initiative (TEI-AI1W2 91)); at the same time, it has been the object of much controversy because of the obvious advantages of tailoring tagsets to project needs. Looking at the problem from a larger perspective than that of isolated projects, a uniform tagset has the following advantages:

- **Objectivisation and standardisation of similar information**: Millions of words have been analysed in the past, using different annotation schemes. Especially the manually analysed linguistic data is expensive to produce and extremely valuable. With a standardised tagset, linguistic information from different corpora of the same language can be *reused* and thus merged into a large data base. Such data bases improve the performance of statistical methods and are a useful resource for the production of balanced corpora.
- **Shared use of language resources**: Corpus manipulation tools such as retrieval tools can be applied to merged resources in a uniform format without much customisation. As well, users of these tools will find it easier to work with a corpus tagged in a standardised tagset. Now, they have to memorize only *one* scheme of tag classes (class names, class semantics, exceptions), as opposed to several schemes for several corpora before.
- **Comparison of annotation schemes**: A comparison of the granularity and degree of similarity of tagsets can be carried out more objectively, once the mapping results are available. The validation of the suggestions of the LRE-project EAGLES is an application in this field.

We believe that standards are important for the linguistic community, especially from the point of view of reusablility.

Of course, there are limits: proposals for standard tagsets should be regarded as approaches towards a neutral platform between projects and different the-

[1] LRE project EAGLES, cf. (EAGLES 94).

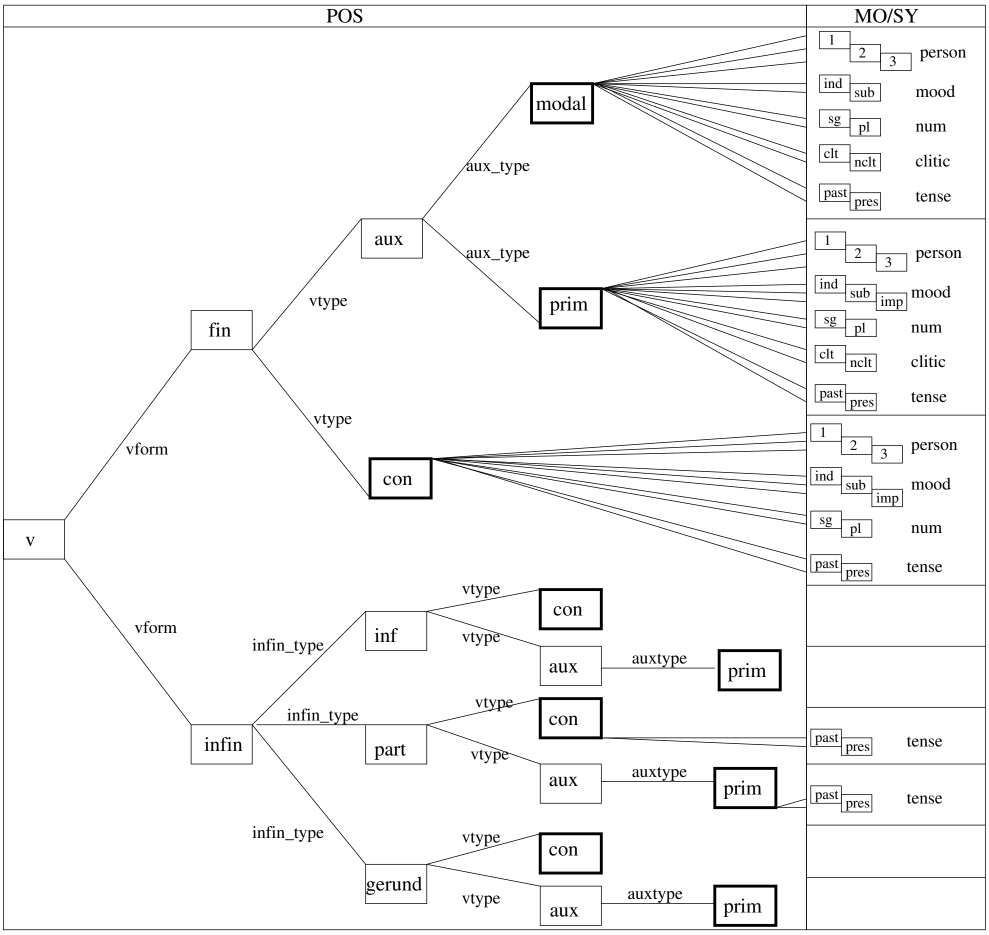

Figure 1: Detail of the type graph (verbs)

ories, rather than as ready-made tagsets that will never be changed. It is important indeed that standards and their support tools be flexible about possible extensions and improvements.

The more general problem of retagging has been approached with tools like ICA (Mamrak O'Connell 92), a public domain retagging tool which uses SGML as interlingua[2]. We also know of current work at Leeds University on mapping tagsets, though this work is concerned with the mapping of syntactic structure encoded in corpora (Atwell et al. 94).

2 A standardised tagset

When designing the architecture of a standardised tagset, we implemented the following constructs as they provide considerable advantages compared to the the traditional flat word labels.

- As the tagset is **constraint-based**, a flexible generalisation is possible over all atomic constraints and combinations of constraints[3]. As a formal grammar[4] is used to define syntactically well-formed specifications of word forms, we can regard our standard tagset as a *specification language*.

  Example: The specification `[pos = v & vtype = aux & pers = 3]` denotes 3rd person auxiliary verbs.

- The tagset is also **typed**, which adds to the naturalness of the specifications of wordforms and helps discover semantic errors in specifications (inconsistent combinations of features, wrong values for features). In our implementation, we follow the closed-world-assumption, which leads to a coherent interpretation for underspecified and/or negated descriptions.

  Example: `[pos = v & (vform = fin | case != gen)]` is a syntactically correct,

[2] Many other retagging tools are available in the SGML world.

[3] Constraints are expressed as attribute-value-pairs.

[4] We used a grammar for Boolean expressions with the usual precedences.

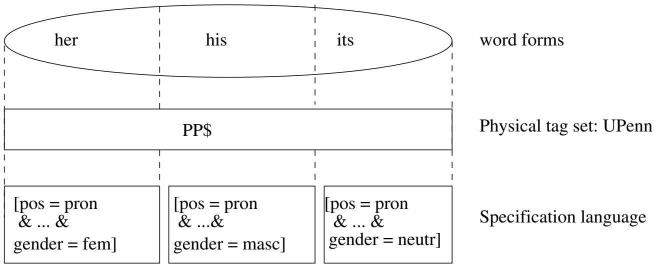
her
his
its
word forms
PP$
Physical tag set: UPenn
[pos = pron & ... & gender = fem]
[pos = pron & ...& gender = masc]
[pos = pron & ... & gender = neutr]
Specification language

Figure 2: $\boxed{\text{1:n}}$ 1 class of physical tagset ↔ n classes of specification language

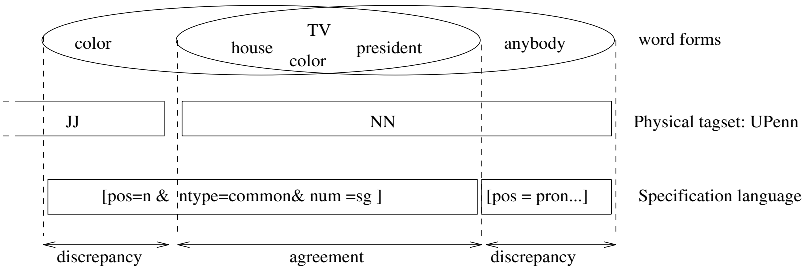

Figure 3: $\boxed{\text{n:m}}$ n classes of physical tagset ↔ m classes of specification language

but ill-typed specification, as the Types `v` (Verb) and `gen` (Genitive) are not type compatible.

- The tagset can be easily **modified** because its manually written definition is compiled into a system internal format.[5] As the design of a tagset involves a cycle with feedback phases, including manual tagging and the writing of guidelines[6], there will be frequent modifications to the tagset, especially in the initial phase.

The EAGLES expert group (cf. (Monachini, Calzolari 93)) suggested an inventory of features and values for a standardised morphosyntactic tagset for European[7] languages; there are different layers, depending on language specificity as well as on application specificity. For the design of a standardised tagset in a specific language, relevant features and values are to be chosen from the inventory. Fig. 1 shows a detail of the tentative English tagset we designed and used for our tests. The type relations are divided into hierarchical (POS) features and nonhierarchical features (MO/SY).

3 Tag mapping: the problems

Mapping tags of an existing, flat-labeled tagset[8] or source annotation scheme to tags of a specification language (target annotation scheme) is an instance of the retagging problem. It is straightforward only in the trivial cases 1:1 (renaming) and n:1. In the latter case, the physical tagset makes finer distinctions than the target annotation scheme. This case introduces no problem for the mapping itself even if not all information contained in the corpus can be accessed. Unfortunately, what we usually find in the mapping business is a mixture of two more problematic cases:

$\boxed{\text{1:n}}$ The physical tagset cannot support a distinction intended by the specification language, e.g. as the distinction `gender` in fig. 2. Therefore, there is a lack of information: the corpus annotation does not provide the wanted distinction.

$\boxed{\text{n:m}}$ There is an overlap between tag classes, as illustrated in fig. 3. In the example case, the source annotation scheme includes special indefinite pronouns like *anybody* into the normal common nouns, whereas some word forms (*color*) are (wrongly!) tagged as adjectives in the source annotation scheme but as common nouns in the target annotation scheme.

[5] The system is implemented in PROLOG, and the definition can be spelled out as a structured PROLOG fact.

[6] The guidelines document is a very important resource for manual taggers as well as for users of the corpus data, as it provides the semantics of the tag classes.

[7] English, French, Greek, German, Dutch, Portugese, Spanish, Italian, Danish.

[8] We call such a tagset *physical tagset* because its tags are actually annotated in an existing corpus, in contrast to the derived tags of the specification language.

4 Mapping Rules

We opted for symbolic mapping rules[9] and designed two kinds of mapping rules to deal with the discrepancies indicated above.

- **Class coverage rules** describe a correspondence of source and target annotation classes[10]. The rule format is as follows: for each physical tag, the equivalent expression in the specification language is named.

  Example: [pos = 'NN'] =>
  [n & (common & sg | mass)].

  The word forms that are annotated with the physical tag `NN` are "common singular nouns or mass nouns" in the terms of the specification language.

- The **exception lexicon** provides a treatment of the individual discrepancy areas of case n:m, in order to deal with noise from unsharp mappings[11]. Specific lexical items can be reclassified, i.e. their standard mapping can be overridden. (Notation: the sign << stands for "out of") They can be reclassified in a different target annotation scheme class instead (sign >> stands for "into").

  Example: The following exception lexicon entry expresses that the target tag for wordforms *anybody, nothing* ... in fig. 3 should not be the standard reading for `NN` (common singular nouns or mass nouns), but should be described as an indefinite pronoun relating to persons.

  [anybody, nothing, something, anything] << [pos = 'NN'] >> [pos=pron & antec=prs & type=indef].

The exception lexicon lookup takes place after the mapping of the class coverages. For more details, see (Teufel 94).

5 Mtree: Internal representation

After the compilation of the mapping rules, the system keeps the information in a data structure called an MTree (mapping tree), see fig. 4, which shows the verb mappings for UPenn. There is an MTree for each physical tagset regarded. MTrees contain a subset of the information contained in the type graph (see fig. ??), namely only those distinctions of the original type graph that are distinguishable in the physical tagset. The new terminals (boxes with thick lines in fig. 4) in this pruned type graph correspond to physical tags (encircled tag names).

Within the rule set, the system keeps track of consistency. Warnings are issued in case of one of the following inconsistencies which might occur during the construction of an MTree:

- **definition holes**: Either target or source annotation schemes are not covered by a mapping rule (classes have been forgotten by the person writing the mapping rules).
- **nondisjunctiveness of classes**: A target annotation class has several source annotation correspondences. Although this might be an instance of case n:1, a warning is issued, because most such cases occur due to a conceptual error.
- **hierarchical inconsistency**: Instead of keeping a clear distinction between terminal classes and nonterminal classes, an odd mapping assigns terminal status to ancestors of classes that are terminals themselves. In fig. 4, the correspondence specified by the dashed arrow introduces a hierarchical inconsistency, as it assigns a physical tag (`VBN`) to a class (`con`) that cannot be terminal because its daughters (`past` and `pres`) already are.

6 System Support

System support includes

- Compilation of the tagset definition: useful for tagsets with many non-hierarchical, i.e. combinatory features (which would have to be multiplied out manually otherwise.)
- Compilation of mapping rules: consistency checks (cf. section 5).
- Interpretation of specifications: Each specification is syntactically and semantically checked, and the corresponding (set of) physical tag(s) is computed, using the MTree information. Due to 1:n and/or n:m cases (unsharp mapping), there can be noise (i.e. groups of word forms which do *not* conform to the specification) in the output. In these cases, the system anticipates the noise to be expected and informs the user. Warnings about noise are essential for a correct interpretation of the output.

  Noisy word classes can be deduced from the MTree: In the MTree given in fig. 4, we can see that target specification `inf` (infinitives) will always induce noise from finite forms, namely subjunctive and imperative forms, because the

[9] We also thought about having a program deduce mapping rules from a corpus. The automatic learning of tag correspondences, at least on a semiautomatic basis, seems possible with standard statistical means (e.g. HMM based learning algorithms).

However, the amount of data needed for such an enterprise (a large training corpus, (manually) annotated in both source and target annotation scheme) made us vote for the symbolic approach.

[10] These rules are used in cases 1:1, n:1, 1:n and in the agreement area of case n:m.

[11] This solution accounts for lexical exceptions only. Contextual discrepancies like the decision to tag a certain wordform like *that* in one class or in several classes (demonstrative pronoun or conjunction or relative pronoun) are not dealt with in this work as this includes a new disambiguation run (pos-tagging)

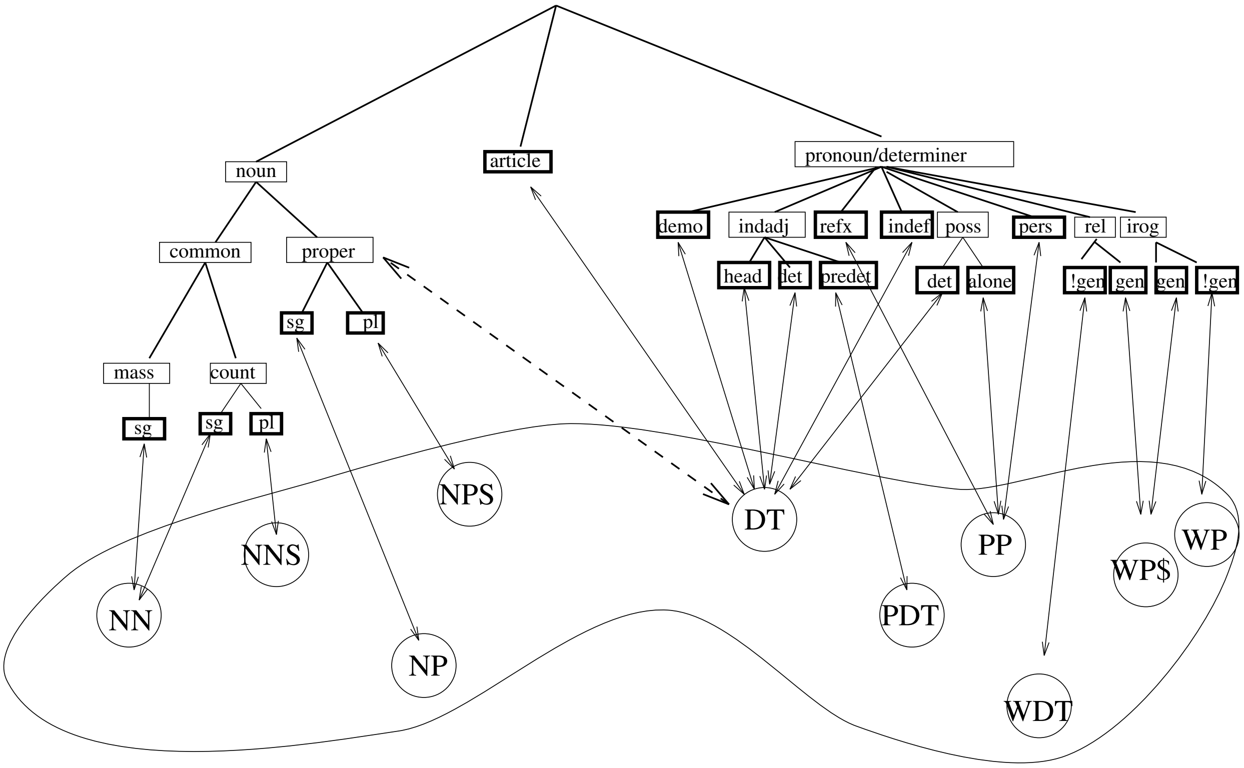

Figure 4: Detail of the MTree for the UPenn annotation scheme

physical class VB does not distinguish between these groups (case 1:n).

7 Results and Outlook

For test purposes, we wrote mapping rules for the UPenn and SUSANNE tagsets. The number of coverage rules is equivalent to the number of physical tags. Rules are easy to formulate, once users have got used to the class semantics of the standard tag set. Information input are tagging guidelines, if the source annotation scheme comes with a comprehensive description of the intended class semantics[12], or corpus queries otherwise, which is more time consuming.

We wrote exemplary exception lexicon entries for auxiliary verbs and some for noun exceptions, but more work can be put into the exception lexicon to improve the accuracy in the lexically determinable cases of discrepancies.

Apart from being used for the validation of the EAGLES standard for English and German, the tool has been integrated into a corpus query system (Christ 94, Schulze 94) to allow for "more abstract" and corpus independent queries. A typical query (content verbs in infinitive or primary auxiliaries in past tense) to a specific corpus (here: UPenn) looks like this:

```
Query> [(vtype=con & vform=inf) |
        (vtype=prim & tense=past)].

%% warning:  Noise from [con & fin & imp]
%%             and from [con & fin & sub]
%%              (Due to tag "VB")!

[((pos = "VB" & word != "be|do|have")|
(pos = "VBD" & word = "was|were|had|did")|
(pos = "VBN" & word = "been|had|done"))]
```

We get the information that the system will query for tags VB, VBD, VBN (with lexical constraints) in the UPenn corpus; however, we must expect to find *finite* content verbs (namely imperative and subjunctive forms) in our output (1:n case).

It would be particularly interesting to explore ways of how to use an MRD to build an exception lexicon automatically, which is especially useful for closed word classes.

Another interesting case are multi-word tags and discrepancies with respect to the assignment of word boundaries (tokenising).[13] Compare the following cases (UPenn tokenising and tagging):

[12] (Santorini 91) provides tagging guidelines for the UPenn corpus, (Garside, Leech, Sampson 87) for the SUSANNE corpus.

[13] For an exhaustive survey of multi-word phenomena, see (Leech, Wilson 93).

- Peter/NP 's/POS house
- he/PP 's/VBZ not at home

In our opinion, *Peter's* should be regarded as one nominal item (with `genitive` as value for the `case` attribute), whereas *he* and *'s* should be kept as two words. We are thinking about designing a rule construct to express this kind of word bundelling with conditional features.

References

Eric Atwell, John Hughes, Clive Souter: *AMALGAM: Automatic Mapping Among Lexico-Grammatical Annotation Models*, Internal Paper, CCALAS, Leeds University, Aug. 1994.

Oliver Christ: A modular and flexible architecture for an integrated corpus query system. In: *Proceedings of COMPLEX'94* (3rd Conference on Computational Lexicography and Text Research). Budapest, Hungary, Jul. 1994.

EAGLES Lexicon Working Group, Interim Report, draft version, ILC Pisa, Oct. 94, to appear Feb. 95.

Roger Garside, Geoffrey Leech, Geoffrey Sampson (eds.): *The Computational Analysis of English – A Corpus-based Approach*, Longman, London, 1987.

Sidney Greenbaum, Randolph Quirk: *A Student's Grammar of the English Language*, Longman, London, 1990.

Geoffrey Leech, Andrew Wilson: *Morphosyntactic Corpus Annotation*, EAGLES, Text Corpora Working Group, Subtask 3.1: Invitation draft. University of Lancester, Dec. 1993.

S.A. Mamrak, C. S. O'Connell: *Technical Documentation for The Integrated Chameleon Architecture*, Ohio State University, Columbus, Mar. 1992.

Mitch Marcus, Beatrice Santorini, Mary Ann Marcinkiewicz: Building a large natural language corpus of English: The Penn Treebank. – *Computational Linguistics 19,* 313–330, 1993.

Monica Monachini, Nicoletta Calzolari: *Synopsis and Comparison of Morphosyntactic Phenomena Encoded in Lexicon and Corpora*, Internal Document, EAGLES Lexicon Group, ILC, Università Pisa, Oct. 1994.

Beatrice Santorini: *Part-of-Speech Tagging Guidelines for the Penn Treebank Project.* Technical Report. Department of Computer and Information Science, University of Pennsylvania, Mar. 1991.

Bruno Maximilian Schulze: *Entwurf und Implementierung eines Anfragesystems für Textcorpora*, Diplomarbeit Nr. 1059, IMS, Universität Stuttgart, Feb. 1994.

Simone Teufel: *Linguistisch motivierte Corpuserschließung: Spezifikationssprache und Anfrageinterpreter*, Diplomarbeit Nr. 1058, Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart, Jun. 1994.

Text Encoding Initiative: *List of Common Morphological Features.* – TEI-AI1W2, Working Paper, Draft Version, Chicago, Jun. 1991.