Hard Problems of Tagset Conversion

Abstract

Part-of-speech or morphological tags are important means of annotation in a vast number of corpora. However, different sets of tags are used in different corpora, even for the same language. Tagset conversion is difficult, and solutions tend to be tailored to a particular pair of tagsets. We discuss Interset, a universal approach that makes the conversion tools reusable. While some morphosyntactic categories are clearly defined and easily ported from one tagset to another, there are also phenomena that are difficult to deal with because of overlapping concepts. In the present paper we focus on some of such problems, discuss their coverage in selected tagsets and propose solutions to unify the respective tagsets' approaches.

1 Introduction

Most annotated corpora use various types of tags to encode additional information on words. In some cases this information is merely the part of speech ("noun", "verb" etc.—hence the term *part-of-speech* or *POS tags*). In many cases, however, the string of characters comprising the tag is a compressed representation of a feature-value structure. Most of the features encoded this way are morphosyntactic (e.g. "gender = masculine", "number = singular"), hence the term *morphological tags*.

Unfortunately, it is very rare to see two corpora sharing a common set of tags. Language differences are only partially responsible—it is the corpus designers, their diverse views, theories and intended uses of the corpora, what matters most. Even two corpora of the same language may define two completely incompatible tagsets.

Such diversity proves disadvantageous for both human users and NLP software. A human user (linguist) typically wants to submit queries such as "show me all occurrences of a noun in plural, preceded by a preposition". Tags however rarely contain statements like "number = plural" literally. That would be prohibitively space-consuming. Instead we have to know that e.g. the fourth character of the tag being "P" means "plural". For instance, the tag `NNIS7-----A----`[1] may read as "part of speech = noun, detailed part of speech = common noun, gender = masculine inanimate, number = singular, case = 7th (instrumental), negativeness = affirmative". To work with the corpus efficiently, a linguist either needs to interpret the tags using specialized software, or to memorize the particular tag scheme. Obviously, if the same linguist has to switch to a different corpus, he/she must memorize more schemes or replace the tag interpretation software.

Similarly, various NLP tools may depend on particular tagsets. While some tools indeed treat tags as atomic strings, others could exploit the tag structure to dig more information about the word—no matter whether they use the features in machine learning, or in human-designed rules. If the tagset changes, manual rules become useless and statistical models have to be retrained at least; even that may not be possible in case the training procedure works with selected subsets of the feature pool. Applicability of NLP software to multiple corpora is exactly the reason why one would want to convert tags from one tagset to another.

For many tagset pairs, designing the conversion procedure is not easy. On one hand, there are rare tagsets (e.g. MULTEXT-EAST, Erjavec (2004)) fitting at the same time languages as distant as Czech and Estonian; on the other hand, tagsets of two closely related languages (e.g. Danish and Swedish) or even two tagsets of the same language may differ substantially (for instance,

[1] This example is taken from the Prague Dependency Treebank (Böhmová et al., 2003).

the Mamba tagset of Swedish (Nivre et al., 2006) contains detailed classification of auxiliary verbs and punctuation but lacks features like number, mood, tense etc.; this is in sharp contrast to another Swedish tagset, Parole (Cinková and Pomikálek, 2006), which in turn is not compatible with the Danish Parole (Kromann et al., 2004) tagset (the former classifies participles as verb forms, the latter as adjective forms; the former has separate tags for numerals, the latter classifies both cardinal and ordinal numbers as adjectives; etc.)

From the above said it follows that the typical tag conversion is an information-losing process. Though it is often desirable to perform it anyway and preserve as much information as possible. Creation of a conversion procedure between two tagsets requires hours of tedious work, consisting mostly of reading the tagging guidelines and translating them into a programming language. A universal description, to which all tagsets map, could make this process easier, and its results reusable. One attempt to find such description and deploy it in the conversion task is DZ Interset (XXXXX, 2008). In the present paper we discuss the development of the universal description and focus on selected hard problems that arise when comparing various existing tagsets.

The rest of the paper is organized as follows: In Section 2 we describe Interset and how it works. In Section 2.3 we describe our universal set of features. Then, Section 4 lists decisions that are difficult w.r.t. universality, and proposes solutions. Finally, we demonstrate the implications on real tagsets, and provide illustrative statistics.

2 Interset

Interset is a universal set of features and their values. It shall be able to store all features that are usually encoded in tags. The role of this universal set *("Interset")* is similar to the role of Interlingua in Interlingua-based machine translation (Richens, 1958) or the role of Unicode among character sets. The Interset serves as an intermediate step on the way from tagset A to tagset B. The interaction between the Interset and tagsets A and B, respectively, is described in *tagset drivers*. Once the drivers have been implemented, we can do the two-way conversion A to B and B to A, plus the conversion between one of these tagsets and any other tagset that has been defined so far.

We are not likely to spare much effort during the initial phase, if compared to just writing a targeted A-to-B conversion procedure. Actually, covering two completely new tagsets requires more work and care: we should describe both encoding and decoding of each tagset, we may have to think about features that are present in neither of them, and we will probably want to be more careful about aspects that may not matter to our current application. However, the reusability of the resulting code should compensate for the effort more than adequately. Plus we provide some algorithms to make adding new tagsets easier, and it is also possible that the required tagset has been covered by someone else who is sharing the code on the web.

2.1 A New Standard?

Interset is not a new annotation standard. There have been attempts to standardize morphosyntactic tagging and it is not Interset's mission to compete with them. Instead, the goal is to cover as many existing tagsets as possible whether they conform to a standard or not. The set of Interset features and values could of course be compared to those defined in standards. There have been several European projects concerning tagset standardization. The EAGLES project (EAGLES, 1996; Leech and Wilson, 1999) produced a set of recommendations for tagsets. Output of the LE-PAROLE project (Volz and Lenz, 1996) was a multilingual corpus of 14 European languages, morphosyntactically annotated according to a common core PAROLE tagset, extended with a set of language specific features. Another multilingual corpus with common tagset is MULTEXT (Ide and Véronis, 1994) for six European languages (English, French, Spanish, German, Italian and Dutch), and later its spin-off MULTEXT-EAST (Erjavec, 2004) for 12 languages (English, Bulgarian, Czech, Estonian, Hungarian, Romanian, Slovene, and in later versions also Croatian, Lithuanian, Resian, Russian and Serbian); the tagsets used in MULTEXT corpora comply with EAGLES. Various EAGLES-compliant tagsets can be added to our system and their mutual similarity will probably make adding them all easier. Weakly related is also the Gold Ontology project (Farrar and Langendoen, 2003) that defines various linguistic concepts, some of which serve as feature names and feature values in Interset. Similarly, morphosyntactic and other terms are included in IsoCat.[2]

[2] http://www.isocat.org/

2.2 A New Tagset?

Interset is not primarily meant as a new *physical* tagset for annotation. Although it obviously could be used that way (possibly after compressing the feature values), it is better thought of as a set of concepts that physical tagsets can map to. Design of physical tagsets is often guided by various needs such as conforming to linguistic tradition and terminology of the given language, minimizing errors of automatic disambiguation etc. In contrast, the most important design constraint for Interset is the portability of information from one tagset to the others. If a feature value X, encoded in tagset A, is not defined in tagset B but it is still probable that users of tagset B would tag the same set of words with feature value Y, then it is desirable that the Interset algorithms are able to change X to Y, when converting from A to B. At the same time, converting to tagset C might require changing X to Z, converting to tagset D would keep X (because D has X, too) and converting to other tagsets would result in resetting X to empty value because there is no better choice available.

Conversion via Interset often looses information but never adds new information. Interset may define feature value X but it just won't be set unless the source tagset defines it, too. Specifically, the conversion procedure does not retag words. For instance, the source tagset may define one tag `IN` for both prepositions and subordinating conjunctions. The target tagset may have separate tags for each of those categories. So will Interset sort out the words tagged `IN` into prepositions and conjunctions? No! In fact, the procedure *never* looks at the word the tag is assigned to. It only works with the tag itself.

2.3 Features

- **pos** = noun|adj|num|verb|adv|prep|conj|part|int|punc
- **subpos** = prop|class|pdt|det|art|aux|cop|mod|ex|voc|post|circ|preppron|comprep|coor|sub|comp|emp|res|inf|vbp
- **prontype** = prs|rcp|int|rel|dem|neg|ind|tot
- **numtype** = card|ord|mult|frac|gen
- **numform** = word|digit|roman
- **numvalue** = 1|2|3
- **advtype** = man|loc|tim|deg|cau
- **punctype** = peri|qest|excl|quot|brck|comm|colo|semi|dash|symb|root
- **puncside** = ini|fin
- **synpos** = subst|attr|adv|pred
- **poss** = poss
- **reflex** = reflex
- **negativeness** = pos|neg
- **definiteness** = ind|def|red
- **foreign** = foreign
- **gender** = masc|fem|com|neut
- **possgender** = masc|fem|com|neut
- **animateness** = anim|nhum|inan
- **number** = sing|dual|plu
- **case** = nom|gen|dat|acc|voc|loc|ins|ill|ine|ela|all|ade|abl|par|tmp|ter|tra|ess|abe|com|cau|dis
- **prepcase** = npr|pre
- **degree** = pos|comp|sub|abs
- **person** = 1|2|3
- **politeness** = inf|pol
- **subcat** = intr|tran
- **verbform** = fin|inf|sup|part|trans|ger
- **mood** = ind|imp|cnd|sub|jus
- **tense** = past|pres|fut
- **subtense** = aor|imp|pqp
- **aspect** = imp|perf
- **voice** = act|pass
- **abbr** = abbr
- **hyph** = hyph
- **style** = arch|form|norm|coll
- **typo** = typo
- **variant** = short|long|0|1|2|3|4|5|6|7|8|9
- **other** = *any other info*
- **tagset** = *where does the other info come from?*

The only reason of saving really everything is that converting a tagset to itself should not lose information. For that purpose we use the "other" feature. It contains arbitrary information that does not fit in other features and distinguishes tags. Since the information is not understood by any other tagset, we need to know which tagset the value comes from. Thus the identifier of the tagset should be stored in the "tagset" feature.

Except for "tagset" and "other", there is a predefined list of possible values for each feature. Every feature also allows the empty value. While several feature-based tagsets distinguish between unknown values and irrelevant features, we do not find it wise in Interset. For instance, the fifth character in the PDT Czech tagset identifies grammat-

ical case. Its normal values are 1 to 7. For parts of speech that do not have case (e.g. interjections) the fifth character is - (dash). Adjectives generally do have case, yet there are borrowed words without Czech case suffixes whose case value is unknown (X). An example is the tag `AAIPX----1A----` for "Buenos" in Buenos Aires. The benefit of making this distinction explicit in a tagset is unclear. What is clear, however, is that we must not reflect it in the universal feature set. Who can say that a feature will be irrelevant—given the context of the values of the other features—in any tagset whatsoever? It is quite easy to find features that are relevant in one tagset and not the other: e.g. Czech past participles distinguish gender, English don't.

3 Tagset Drivers

While the Interset is merely an abstract definition, the real implementation lies in the tagset drivers. A driver is a code library responsible for decoding and encoding tags. Decoding is reading a string (tag) into an internal data structure, in accordance with the list of possible features and their values. Encoding works the other way around.

The encoder obviously is the more difficult part. The decoder just reads and sorts the information, ideally not losing a single piece of it. If anything has to be discarded because it does not fit the target tagset, the discarding is encoder's task. There are two main reasons why encoding is not easy:

- The encoder should be prepared to all values of all features, regardless that some of them are unknown in the particular tagset. For instance, if number = dual and the tagset does not know dual, it is probably better to encode plural than just leave number unknown.
- Even if the target tagset knows features A and B, concrete value of A can restrict permitted values of B. Some combinations of feature values are not allowed. For instance, the Swedish Parole tagset allows "pos = noun & gender = common | neuter", and also "pos = pronoun & gender = masculine | feminine | common | neuter". If we are to encode "pos = noun & gender = masculine", we can either honor the part of speech, or the gender, but not both.

Fortunately enough, unknown feature values / combinations can be dealt with automatically if the driver has the list of all possible tags. By decoding all tags on the list, we get feature values for every tag. We thus know all feature values permitted in the given tagset and we know all value combinations. We have defined an ordered list of back-off values for every Interset feature value. The back-off lists contain all other values of the feature, including the empty value, so it is guaranteed that we always find a value that is permitted.[3] Of course, the encoder can override the default back-off list if necessary.

As for unknown feature combinations, there is a predefined total ordering of the features that defines their priority (this can be overridden, too). Since features are ordered, all value combinations can be stored in a trie structure. On selecting value of a higher-priority feature, the structure immediately reveals restricted value space for all lower-priority features.

This back-off technique is implemented in a helper module. Any driver can call it and have the features adjusted to something the driver itself might produce during decoding. The encoder can then concentrate on the driver's native feature combinations. Besides that, the helper module can also check a driver's integrity by looking whether the decoder only sets known features and values, whether encode(decode(x)) = x etc.

The whole thing is implemented in Perl. The drivers are Perl modules whose encode and decode functions can be called from other Perl programs, either to access the feature values, or to convert tagsets. The conversion script is very simple and looks like this:[4]

```
use tagset::cs::pdt;
use tagset::en::penn;
while(<>)
{
    print tagset::en::penn::encode
        tagset::cs::pdt::decode $_, "\n";
}
```

We have implemented and tested drivers for several tagsets of the CoNLL 2006 (Buchholz and Marsi 2006) and 2007 (Nivre et al. 2007) shared task treebanks, for the Penn Treebank (Marcus et al. 1993), the Prague Dependency Treebank (Böhmová et al. 2003) and others, totaling 20 drivers.

[3] The necessary condition is that the decoder only sets known feature values, which is desirable anyway.

[4] Real conversion script would also have to deal with the format in which the tags are mixed with text in the corpus. This example merely assumes a list of tags, without the actual words and other annotation.

Those drivers are freely available on the web.[5] We believe that the reusability will only be truly exploited if the drivers are shared in the community and we encourage everyone to contribute with drivers they need to write for themselves.

4 The Hard Problems

Working with various tagsets, we identified several fields that were difficult to capture and unify.

- Pronouns, determiners, articles, wh-adverbs.
- Numerals.
- Indefinite verb form, especially participles.
- Particles and exotic small word categories.
- Different tokenizations, tags spanning multiple or joint tokens with incompatible categories.

Endemic word classes were one example. Whenever seen fit, we tried to roof them with some more common parts of speech, instead of introducing a new high-level class. We wanted to reduce the necessity of encoders' taking care of parts of speech unknown in their home tagsets. Roofed word classes are usually distinguishable by one of the detailed-part-of-speech features.

Determiners, predeterminers and articles are one group of word classes missing in a substantial number of tagsets. We chose adjectives to serve as the roof class here. To pick another set of examples, here is an overview of various sorts of particles found in our tagsets:

- unclassified particle (Czech `TT`, English `RP`, Swedish `Q`)
- interrogative particle (Arabic `FI` هل *(hal)*, Bulgarian `Tn` *ли (li)*)
- affirmative particle (Bulgarian `Ta` *да (da)*)
- negative particle (Arabic `FN` لا *(lā)*, Bulgarian `Tn` *не (ne)*, German `PTKNEG` *nicht*)
- response particle (German `PTKANT` *ja* = "yes", *nein* = "no", *doch* = "yes", *danke* = "thank you"…)
- auxiliary particle (Bulgarian `Tx` *да (da)* = "to", *ще (šte)* = "will")
- modal particle (Bulgarian `Tm` *май (maj)* = "possibly")
- verbal particle (Bulgarian `Tv` *нека (neka)* = "let")
- emphasis particle (Bulgarian `Te` *даже (daže)* = "even")
- gradable particle (Bulgarian `Tg` *най (naj)* = "most")

[5] `http://xxx.com/`

- unique POS (Danish `U`, covering the words *at* = infinitival "to", *som*, *der*)
- infinitive mark (German `PTKZU` *zu*, Swedish `IM` *att*, English `TO` *to* – includes prepositional occurrences of *to*)
- separated verbal prefix (German `PTKVZ`, *vor* in *stellen Sie sich vor*)
- adjectival particle (German `PTKA`, *am* in *am besten*, *zu* in *zu groß*)
- existential *there* in English (`EX`)
- measure word, quantifier (Chinese `DM`)
- genitive particle de in Chinese (`DE`) 的 and 得
- Chinese particles 了 *(le)* (perfect), 著 *(zhe)*, 起 *(qǐ)*, 過 *(guò)* (`Di`)
- Chinese particles 了 *(le)*, 的 *(de)*, 來 *(lái)* (`Ta`)
- Chinese particles 而已 *('éryǐ)*, 沒有 *(méiyǒu)*, 也罷 *(yěba)*, 沒有 *(méiyǒu)*, 好了 *(hǎole)* (`Tb`)
- Chinese particles 呢 *(ne)*, 吧 *(ba)*, 啊 *('a)*, 囉 *(luō)* (`Tc`)
- Chinese particles 嗎 *(ma)*, 否 *(fǒu)* (`Td`)

As mentioned earlier, some tagsets consider participles forms of verbs, others classify them as adjectives; some tagsets make numerals special cases of adjectives, others have separate POS tags for cardinals, ordinals and various other numeral classes, yet others separate cardinal numbers and put the rest under other POSes. Differences in approaches taken by different tagsets might result in different feature values; for instance, we could decode verbform = "participle" without regard to whether pos = "verb" or pos = "adj". Naturally it is desirable to decode the same thing into the same set of features each time. Although we could ban particular feature-value combinations in Interset, effectively forcing the driver authors to seek the permitted decoding, we prefer to leave it as a recommendation, since we do not want to predict, which feature combinations will never ever be needed to distinguish two different words.

Probably the broadest source of problems is pronouns, determiners and various WH-words. Somewhere pronouns are only personal or possessive; somewhere there is a diversity of interrogative, relative, demonstrative, indefinite and negative pronouns. In the BulTreeBank (Simov et al., 2004), anything interrogative is a pronoun, although it could be considered numeral *(how much?)* or adverb *(where? when? how?)* elsewhere. Some tagsets address the variable syntactic behavior of pronouns (*he* substitutes a noun, *his* functions as an adjective). Some tagsets and lan-

guages do not have determiners but they have pronouns (demonstrative, indefinite) instead. All that lead us to remove pronouns and determiners as independent parts of speech. Instead, nouns, adjectives and adverbs have the feature "prontype" to distinguish the various types (personal, demonstrative, interrogative…) Empty value of this feature signals a normal noun (adjective, adverb).

Note however, that any guidelines are only to ensure unified approach to different presentations of the same information. It does not apply to information that simply is not there. If cardinals were tagged as *normal* adjectives (without sub-classing adjectives to numeral and others) they would remain so in Interset and also in the target tagset. We cannot add information, we only can lose it.

5 Conclusion

We have proposed a method for tagset conversion that is reusable and, to a reasonable extent, universal. Our interlingua-inspired approach enables to interpret part-of-speech and morphological tags in a uniform way, and to convert information that is shared by two tagsets. Besides the obvious advantage of being able to use tools that expect a particular tagset, we also observed improvements in performance of a statistical parser.

Acknowledgements

References

Alena Böhmová, Jan Hajič, Eva Hajičová, and Barbora Hladká. *The Prague Dependency Treebank: A Three-Level Annotation Scenario*, chapter 7, pages 103–128. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2003. 1

Silvie Cinková and Jan Pomikálek. Lempas: A make-do lemmatizer for the swedish parole-corpus. *The Prague Bulletin of Mathematical Linguistics*, 86:47–54, 2006. 1

EAGLES. Recommendations for the morphosyntactic annotation of corpora, 1996. URL `http://www.ilc.cnr.it/EAGLES96/annotate/annotate.html`. 2.1

Tomaž Erjavec. Multext-east version 3: Multilingual morphosyntactic specifications, lexicons and corpora. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004)*, Lisboa, Portugal, 2004. 1, 2.1

Scott Farrar and D. Terence Langendoen. A linguistic ontology for the semantic web. *GLOT International*, 7(3):97–100, 2003. URL `http://www.linguistics-ontology.org/gold.html`. 2.1

Nancy Ide and Jean Véronis. Multext (multilingual tools and corpora). In *Proceedings of the 15th International Conference on Computational Linguistics (COLING-94)*, Kyoto, Japan, 1994. URL `http://www.aclweb.org/anthology/C/C94/C94-1097.pdf`. 2.1

Matthias T. Kromann, Line Mikkelsen, and Stine Kern Lynge. Danish dependency treebank. In *http://www.id.cbs.dk/ mtk/treebank/*, København, Denmark, 2004. 1

Geoffrey Leech and Andrew Wilson. Standards for tagsets. In *Syntactic Wordclass Tagging. Text, Speech and Language Technology*, pages 55–80, Dordrecht, The Netherlands, 1999. Kluwer Academic Publishers. ISBN 0-7923-5896-1. 2.1

Joakim Nivre, Jens Nilsson, and Johan Hall. Talbanken05: A swedish treebank with phrase structure and dependency annotation. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*, Genova, Italy, 2006. 1

Richard Hook Richens. Interlingual machine translation. *The Computer Journal*, 1(3):144–147, 1958. 2

Kiril Simov, Petya Osenova, and Milena Slavcheva. Btb-tr03: Bultreebank morphosyntactic tagset. In *BulTreeBank Project Technical Report No. 03*, Sofija, Bulgaria, 2004. 4

Norbert Volz and Suzanne Lenz. Multilingual corpus tagset specifications. mlap parole 63–386 wp 4.1.4, 1996. URL `http://www.elda.org/catalogue/en/text/doc/parole.html`. 2.1

XXXXX. Xxxxx. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*, Marrakech, Morocco, 2008. European Language Resources Association. ISBN 2-9517408-4-0. 1