POS Tagging and Less Resources Languages Individuated Features in CorpusWiki

Maarten Janssen(✉)

Centro de Lingustica, Universidade de Lisboa, Lisbon, Portugal
maarten.janssen@campus.ul.pt

Abstract. CorpusWiki (http://www.corpuswiki.org) is an online tool for building POS tagged corpora in (almost) any language. The system is primarily aimed at those languages for which no corpus data exist, and for which it would be very difficult to create tagged data by traditional means. This article describes how CorpusWiki uses individuated morphosyntactic features to combine the flexibility required in annotating less-described languages with the requirements of a POS tagger.

Keywords: POS tagging · Less-resourced languages · Morphosyntax

1 Introduction

Part-of-Speech (POS) tags have been a fundamental building block for many Natural Language Processing (NLP) tasks for quite a while. And in that time, POS taggers have been developed for an ever-growing number of languages. With these efforts, the vast majority of texts can now be automatically provided with morphosyntactic labels, since there are POS taggers for all the major languages.

However, when viewed from a different angle, the number of POS taggers is very limited: although it is hard to provide a solid estimate, the number of languages for which there is a working POS tagger is less than a hundred, whereas according to the ISO language codes, there are about 4.000 languages still spoken in the world today, which would mean that less than 2,5 % of the existing languages can be tagged automatically.

CorpusWiki is an initiative to remedy this situation. It is an online environment that allows linguists to develop POS annotated corpora, and automatically train a POS tagger, for almost any language in the world. The system tries to guide the linguists through the process via easy to use graphical interfaces, where the linguist only has to provide linguistic judgement about the language, and the system will automatically take care of the computational management behind the screens.

With the help of CorpusWiki, it becomes easier to develop POS-based resources for languages for which there are no such resources available yet, since it only requires native speakers with sufficient linguistic awareness, and does not require the involvement of computational linguists. This makes it possible to

Z. Vetulani et al. (Eds.): LTC 2013, LNAI 9561, pp. 411–419, 2016.
DOI: 10.1007/978-3-319-43808-5_31

develop resources not only for widely spoken languages with little to no computational resources, such as Runyankore or Mapudungun, but also languages with few speakers, such as Upper Sorbian or Svan, and even dialects that are not considered separate languages, but have sufficiently many distinctive traits to merit a treatment of their own, such as Aranese, a dialect of Occitan spoken in the north of Spain, or Talian, the form of Venetian spoken by the immigrants on the border between Brazil and Argentina.

In order to allow building corpora for as wide a range of languages as possible, CorpusWiki attempts to be as language independent as possible, and the development of a truly language independent framework faces a wide range of problems. Apart from computational challenges, such as getting rid of the need for language-specific computational resources like a tokenization module [5], logistic issues such as the support for right-to-left writing system, and human-computer interface issues such as allowing users to correct structural errors using a pure HTML interface, there is also a more fundamental problem with POS tagging less resourced languages.

The problem that this article deals with is of a more fundamental level: for a significant number of the languages for which no POS tagged resources exist, it is not even that known what the correct morphosyntactic labels are. Part of the motivation for doing corpus-based research in such languages is exactly to find out what the morphosyntax of the language is. And in practical terms, this leads to a vicious circle: before being able to POS tag a corpus, it is first necessary to POS tag a corpus (to find out what the correct labels are).

This article describes the approach used in CorpusWiki which is aimed at overcoming this problem: assigning individuated morphosyntactic labels to words, instead of single morphosyntactic labels. But before turning to the implementation of labelling, the next section will first give a more detailed description of the CorpusWiki project.

2 CorpusWiki

CorpusWiki (http://www.corpuswiki.org) is an online tool for building POS tagged corpora in (almost) any language. The system is primarily aimed at those languages for which no corpus data exist, and for which it would be very difficult to create tagged data by traditional means (although it has been used for large languages like Spanish and English as well). For large but less-resourced languages there are often corpus projects under way, in the case of Georgian there is for instance the corpus project by Paul Meurer [7] as well as corpora without POS tags, such as the dialectal corpus by Beridze and Nadaraia [2]. But for smaller languages such as for instance Ossetian, Urum, or Laz corpus projects of any size are much less likely. Corpora for these languages without POS tags often exist, for these specific languages in the TITUS project (Gippert), but annotating such corpora involves a computational staff that is typically not available for such languages.

CorpusWiki attempts to provide a user-friendly, language-independent interface in which the user only has to make linguistic judgements, and the computational machinery is taken care of automatically behind the screens. The system is designed for the construction of gold-standard style corpora of around 1 millions tokens that are manually verified, although there is no strict upper or lower limit to the corpus size. CorpusWiki intends to make its resources as available as possible, and all corpora, as well as their associated POS tagger, can in principle be downloaded. Corpora are built in a collaborative fashion, in which people from all over the globe can contribute to the various corpora, although the corpus administrator (in principle the user who created the corpus) can determine which users can collaborate on the corpus.

In CorpusWiki, a corpus is not a single object, but a collection of files containing individual texts. Each text is stored in TEI XML format, and each file is individually treated, where the treatment consists of three steps: first, the text is added to the system. Then the text is automatically assigned POS tags using an internal POS tagger, which is trained on all tagged texts already in the system. And finally, the errors made by the automatic tagger have to be corrected manually. Once the verification of the tags is complete, the tagger is retrained automatically. In this fashion, with each new text, the accuracy of the tagger improves and the amount of tagging errors that have to be corrected goes down. The only text that is treated differently in this set-up is the very first text, since for the first text, there are no prior tagged data. The system uses a canonical fable as the first text for each language to make the initial manual tagging of the first text go as smoothly as possible.

The objective of CorpusWiki is to create languages resources that are as available as possible. All corpora and their derived products are available for use online, where the corpora are indexed using the CWB system and can be searched using the CQP query language. Furthermore, from the moment the corpus reaches a minimum critical size, it becomes possible to download the corpus itself, the POS tagger with the parameter files for the language, and other related resources where applicable. Downloading is done via a Java exporter tool that can export the corpora in a number of standardized formats such as TEI and TIGER XML. Each corpus is attributed to the list of its contributors.

The tagging in CorpusWiki is done by the dedicated Neotag tagger, which was designed to be purely data driven: it does not require a language specific tokenization module, but rather tokenization is initially done by simply splitting on white spaces (and punctuation marks), and space-separate unit can be split or merged by the tagger itself. And Neotag does not require an external lexicon, since it uses the lexicon of the training corpus itself as its lexicon. Other than that, it is a relative standard *n*-gram tagger that uses word-endings for tagging out-of-vocabulary items. With the 1 million token target size of the CorpusWiki corpora, the tagger typically provides a 95–98% accuracy, although the actual accuracy of course depends a lot on both the language itself and the tagset it uses.

2.1 Interlinear Glosses Versus POS Tagging

CorpusWiki is built around a POS tagging system. However, its aim of allowing the creation of (computational) resources for less-resourced languages places it more in the domain the class of tools for linguistic fieldwork, and specifically makes it comparable to tools for Interlinear Glossed Texts (IGT), such as Shoebox [3] or Typecraft [1] This section provides a comparison between IGT systems and CorpusWiki.

For the large, mostly western-European languages there is a long tradition of morphosyntactic description. Assigning POS tags to words in a text in those language is not always easy, as anybody who has ever worked with a POS tagged corpus can vouch for, but the labels themselves are clear: even though it might be difficult to decide exactly when to call a past-participle, like *boiled*, an adjective and when to call it a verb-form, it is clear that those are the two choices, wherever the border is placed exactly. And even though there are several different names for the gender in Dutch and Norwegian that is not the neutral gender, including non-neuter, masculine/feminine, and common gender, it is clear that there is such a gender, independently of what it is called.

But for the majority of languages in the world, there is no such extensive grammatical tradition, and it is difficult to list the morphosyntactic features of the language to start with: native speakers are capable of correctly using the morphosyntax, but often not consciously aware of what the exact role of the morphemes is, which morphosyntactic categories can be used with which word classes, or what the possible values for each morphosyntactic feature are. An important task in the creation of corpora for such languages is often exactly to find out the morphosyntax of the language, which makes it difficult if not impossible to define a tagset at the start of the process.

For less-resourced, and less-described languages, the typical tool of choice is therefore not a POS tagging system, but rather an IGT application. In IGT, each word is provided with a variety of labels, most relevantly for the issues at hand with morphosyntactic labels. Words can either be split into morphemes, where each morpheme is provided with a label, or multiple labels can be assigned to the word itself, separated typically by a dot.

In Shoebox, the choice of which labels to use in the morphosyntactic labelling is up to the user, and tagging a texts consists largely of assigning the morphosyntactic label(s) by hand. This makes it very easy to develop the tagset while creating the corpus: you can decide which morphemes there are the moment they first appear, and if in the process of assigning labels it becomes clear that some of the labels were assigned incorrectly, one just has to search though the text for all occurrences of the incorrectly tagged morpheme (or feature), and change the labels.

Despite the ease of use, complete freedom in the assignment of labels makes it unlikely that the labels in one corpus will end up the same as the labels in another corpus. More interactive IGT tools such as Typecraft therefore ask the user to first define a list of labels, where the labels are selected from a list of predefined morphosyntactic features - in the case of Typecraft following the

GOLD ontology [4]. This method keeps the flexibility of creating the tagset on-the-fly, since it is possible to add new labels the moment they are required, while keeping the tagging of various corpora comparable, since the labels are selected from a centralized list.

Although IGT tools are very flexible, they are difficult to scale: IGT tools are not meant for assigning tags automatically, and in principle, each label has to be assigned manually, although several systems like Typecraft can automatically assign a tag to words that had been tagged before. This makes annotation in IGT time consuming: each new word will have to be labelled by hand, and each ambiguous word, such as *hammer* which can be either a noun or a verb, will have to be disambiguated by hand.

POS taggers, on the other hand, are exactly meant for determining the most likely tag for a word given its context, and based on the training corpus. This means that for new sentences, POS taggers will attempt to imitate the decision you made before in that context. To take the (relatively easy) case of past participles in English: in the currently common setting where a PP within a verb cluster is marked as a verb form, whereas a PP within a nominal cluster is marked as an adjectival form, a POS tagger will automatically suggest that a participle next to (auxiliary) verbs, as in *has boiled*, should be a verb form, whereas a participle next to a noun, as in *boiled egg*, should be adjectival. So POS taggers help to tag similar words in similar ways, since they use the context to disambiguate words. As a result, POS taggers help to keeping a consistent tagging within the corpus. Since many taggers can provide confidence scores, it can even alert you to doubtful cases, guiding you where to pay more attention in the correction process.

However, as mentioned before, the traditional design of building a POS tagger is not really meant for discovering the morphosyntax of a language: a traditional (statistical) POS tagger requires that you first define a tagset, then manually annotate a training corpus with that tagset, and inflect a dictionary using that tagset, and only then do you obtain a parameter set for the tagger that you can then use to tag additional tags. This makes it hard to build up the tagset (that is to say, define the morphosyntactic features of the language) during the construction of the corpus, making them only usable for language for which the morphosyntax is well established, and dictionary resources are available, which is often not the case in less-resourced languages. That is why CorpusWiki does not work with a simple tagset, but rather by individuated morphosyntactic features, as will be explained in the next section.

3 Individuated Features in CorpusWiki

In order to allow flexibility similar to that of IGT systems in a POS tagging environment, CorpusWiki uses a simple idea: rather than working with fixed lists of monolithic tags, CorpusWiki treats each morphosyntactic feature separately as individuated attribute/value pairs. Each attribute is stored as an XML attribute on the XML token element.

Like Typecraft, CorpusWiki uses a pre-defined tagset that defines which morphosyntactic features the language has, and which possible values each feature has. Each morphosyntactic feature is associated with a main POS tag, and when annotating a word, this pre-defined tagset is used to let the user select first which main POS the word has, and then select the correct value for each feature associated with that POS. For instance, when (manually) tagging the word *shoes* in English, the user first indicates that it is a (common) noun, and since nouns in English have a number, which is either singular or plural, the user is then asked to select whether *shoes* is a singular or a plural noun.

Because the features are individually stored, it is easy to modify the tagset when the need arises. Say that after a couple of words or texts, we run into the words *mother's*, which shows that English noun actually also have a case, which can be genitive, or non-genitive, which is called *base* in CorpusWiki, but is also called nominative, default or structural case. Like in Typecraft, we can then modify the tagset and add case as a feature for common nouns, with *genitive* and *base* as possible values. For all subsequent nouns, the system will then also ask for the case of the noun, and we can indicate that *base* is the default value.

Although it is easy to insert a new feature, that does not mean that feature is automatically assigned to all words already tagged. After adding case for nouns, all nouns that were already tagged will have to be (manually) marked for case. CorpusWiki attempts to make this easier by allowing the user to search for all nouns, and mark them for case quickly from a list of all nouns in the corpus. Yet even so, it makes adding new features more and more problematic as the corpus grows. In CorpusWiki, users can therefore only modify the tagset as long as the corpus is small. But since for a larger corpus, the tagset should have been largely established, flexibility is also no longer that needed when the corpus reaches a certain critical mass.

The use of individuated features is that it is less efficient as a storage method than position-based representation. For large corpora, this would provide a problem, but CorpusWiki is meant typically for small to medium-sized corpora of up to a couple of million words. With those kinds of sizes, the corpus files are small enough to not be problematic with the current size of hard disks. For extension beyond that, there is a built-in functionality in CorpusWiki to export the corpus to a position-based system, where they can be used in other tools, including the TEITOK system which is a spin-off from the CorpusWiki project and uses the same file structure and architecture.

As should be obvious from the description above, CorpusWiki associates morphosyntactic features with words, and not with morphemes. This has several consequences. Firstly, it gives a similar treatment to languages like Turkish, where each feature can (almost always) be associated to a morpheme, and languages like Spanish, where it is clear that a form like *corrí* is past, perfective, 1st person, and singular, but there is only one single morpheme expressing all these different features. Secondly, it means that it is crucial to correct distinguish different features that can have the same values, as for instance in the case of (female) gender for possessive pronouns, there are different attributes for the possessor gender (as in the English *her*) and object gender (as in the French *sa*). And morphemes below the stem are never marked: when referring to child seats, the Portuguese word *cadeirinhas* is not marked as a diminutive, but only as a plural of *cadeirinha*.

When training and using the Neotag POS tagger, the individuated features are compressed into a single string, which is not a position-based tag, but a monolithic tag nevertheless. Since the tagger is retrained at regular intervals, adding additional features will simply create larger tag strings for the same words when the tagger gets retrained.

3.1 Searching with Individuated Features

The use of individuated features has an additional advantage: searching becomes more transparent. If we want to search for words with specific features, in a traditional, position-based corpus, it is necessary to search in the right position in the tagset. For instance if we want to use CQP to search for singular nouns in the Multext Slovak corpus, the correct expression would be: `[msd="Nc.s.*"]`. With individuated features in CorpusWiki, this type of search query become much more transparent and easy to use: `[pos="N" & number="singular"]`. However, the advantages go beyond merely making searches easier: it allows for searching on agreement in ways that are impossible with position-based or other non-individuated tagsets. In languages with morphological number, the number on the adjective and noun have to agree. If a noun does not have the same number as the adjective following it (or preceding it), that is either a tagging error, or a case in which the noun and adjective do not belong to the same NP. Therefore, it is useful to be able to search for noun that do or do not match the adjacent adjective in number, especially in an environment like CorpusWiki where the corpus is constantly being corrected. In a position based framework, there is no real way to do this, it is only possible to search for specific combinations of tags (using regular expression). With individuated adjectives, on the other hand, it becomes easy to directly compare the number of two adjacent items, and a noun/adjective pair that does not agree in number can be found in the following manner in CQL:

```
a:[pos="N"|pos="A"] b:[pos="N"|pos="A"] :: a.number != b.number
```

4 Conclusion

CorpusWiki attempts to combine the flexibility needed for linguistic fieldwork and the creation of linguistic, POS annotated corpora for less-described languages with the advantages in terms of work-load and consistency provided by a POS tagger. It does this by using individual morphosyntactic feature/value pairs as input, rather than a fixed list of POS tags as traditionally used in POS tagger systems. The use of a flexible tagset is only one of many features implemented in CorpusWiki in an attempt to provide as much as possible an easy-to-use system that is fully language independent, and usable for well-described languages and linguistic fieldwork alike.

The framework has proven to be properly language independent and has been used to create corpora for over 50 different languages of very different language families, for many of which no prior POS taggers existed. Although most of these corpora are very restricted in size for the moment, the tagging and lemmatization process is working well for each and every one of them, meaning that CorpusWiki is well under way to significantly increase the number of languages for which POS taggers are available.

As is not unexpected in a setting like CorpusWiki, the first few text are the most labour intensive since the tagset is still unstable, and the accuracy of the tagger is still low, but the work speeds up considerably after the corpus reaches a critical size. A good part of the existing corpora have been built by students as part of a term project, where the creation of a corpus of 5.000 to 10.000 words (after which the tagger starts tagging with a decent accuracy) from scratch is well feasible for students without any computational background.

Despite the fact that the creation of corpora for new languages is incomparably easier using CorpusWiki than it is using traditional POS methods, practice has shown that the initial effort required provides a large stumbling block for users attempting to create a corpus, and too many external users have abandoned the corpus they started much earlier than we would have hoped. From the limited feedback we managed to obtain from people abandoning their efforts, there are two important reasons for this. Firstly, the creation of a corpus consists of two relatively independent parts: the collection of the actual texts, and the annotation of these texts. And users interested in doing the latter often are not at ease doing the former. And secondly, even with the computational help CorpusWiki provides, creating a corpus is still labour intensive, and people do not feel comfortable investing this time in an online system they do not have under their own control.

To address these issues, we added the option to CorpusWiki to keep a corpus private during its creation, which allows editors to only have access to the corpus for themselves during, say, the writing of their thesis. On top of that, two subsequent projects were developed: the Multilingual Folktale Database (MFTD, http://www.mftd.org) and TEITOK [6] (http://teitok.corpuswiki.org).

MFTD is an online system where people can contribute folktales in any language to be accessible online for the language community at large. These can be originals or translations, which hence includes translations into less resourced languages of well known fairytales by Grimm or Andersen, as well as original folktales from all around the globe, and translations of those traditional folktales in less resources languages into "colonial" languages to make them accessible to a larger audience.

TEITOK is a distributable variant of CorpusWiki, which people can install on their own server. The main thing TEITOK does not include is the system of individuated features, rather in exporting a CorpusWiki to TEITOK, the individuated features are mapped onto a traditional position-based tagset, with a structural description of the tagset that allows translating the position based tagset back into individual attribute/value pairs, allowing for efficient storage once the tagset has been stabilized. Given the advantages described in this article, this means that in order to create a locally installed POS annotated corpus for a new language in TEITOK, the easiest way is to first create a corpus in CorpusWiki, and then export it to TEITOK for further development.

Although it is too early to tell, we hope that with these additions, the number of languages available in CorpusWiki will grow even faster than it has thus far.

References

1. Beerman, D., Mihaylov, P.: TypeCraft collaborative databasing and resource sharing for linguists. In: Proceedings of the 9th Extended Semantic Web Conference, Workshop, Interacting with Linked Data, 27th–31st May 2012 (2012)
2. Beridze, M., Nadaraia, D.: The corpus of Georgian dialects. In: Proceedings of the Fifth International Conference, Slovakia (2009)
3. Drude, S.: Advanced glossing: a language documentation format and its implementation with shoebox. In: Paper presented at the International Workshop on Resources and Tools in Field Linguistics, Las Palmas, Spain, 26–27 May 2002 (2002)
4. Farrar, S., Langendoen, D.T.: A linguistic ontology for the semantic web. GLOT Int. **7**, 97–100 (2003)
5. Janssen, M.: Inline contraction decomposition: language independent POS tagging in the CorpusWiki project. In: Paper presented at the 10th Tbilisi Symposium, Gudauri (2013)
6. Janssen, M.: Multi-level manuscript transcription: TEITOK. In: Paper presented at Congresso de Humanidades Digitais em Portugal, Lisboa (2015)
7. Meurer, P.: Constructing an annotated corpus for Georgian. In: Paper presented at the 9th Tbilisi Symposium, Kutaisi (2011)