Documenting Computational Grammars with the Linguistic Type Database

Francis Bond
Palacký University

2022-07-30

We have constructed a web-based framework for supporting collaborative multilingual grammar and treebank development in which developers are distributed around the world.

It is important for developers of the world-wide collaboration to **i**) grasp and share the big picture of the grammar and treebank of each language and **ii**) understand commonalities of human languages.

Our framework, which we call the Linguistic Type Database (**ltdb**), describes linguistic types of the grammar and gives examples of their usage in a treebank. Currently, we have applied the **ltdb** to grammars and treebanks of Chinese, English, Indonesian, Japanese and Spanish. It is also being integrated into the MATRIX grammar development system. The **ltdb** is based on an earlier system that documented only the linguistic types: the LexType DB (Hashimoto et al., 2007a,b)

Introduction

Treebanks constructed with detailed linguistic information play an important role in various aspects of natural language processing; for example, grammatical knowledge acquisition; world knowledge acquisition (Bond et al., 2004); and statistical language model induction (Toutanova et al., 2005). Such treebanks are typically semi-automatically constructed by a linguistically rich computational grammar. A detailed grammar in turn is a fundamental component for **precise** natural language processing. It provides not only detailed syntactic and morphological information on linguistic expressions but also precise structural semantics, which can be used in, for example, machine translation or computer aided language learning (Bond et al., 2011; Suppes et al., 2014).

The Deep Linguistic Processing with HPSG Initiative (DELPH-IN)[1] has been constructing open-source linguistically precise grammars and treebanks for several languages, including English (Flickinger, 2000), Korean (Kim et al., 2011), Spanish (Marimon, 2012), Chinese (Fan et al., 2015) and Japanese (Siegel et al., 2016). DELPH-IN grammars are compatible in that they are all based on the same formalism, which is Head-driven Phase Structure Grammar (HPSG Sag et al., 2003) and can be used by the same processors. The semantics are based on Minimal Recursion Semantics (Copestake et al., 2005), a shallow semantic representation that allows for underspecification of scope.

[1] `https://www.delph-in.net/`, `https://delph-in.github.io/docs/`

Developers of DELPH-IN are distributed all over the world and are contributing their expertise in linguistics to the DELPH-IN grammar and treebank construction via the Internet. Most grammars are developed along with one or more treebanks of examples. The grammars and treebanks are then available for download, either as snapshots or through version control systems such as github.

One of the aims of DELPH-IN is to capture commonalities of human languages in the course of the development. Capturing commonalities (or universality) of human languages is not only of interest for theoretical linguistics but also an aid to multilingual grammar development since it makes existing grammars more compatible and developing a new grammar much easier. The universal core of the DELPH-IN grammars is codified in the MATRIX (Bender et al., 2002), a bottom-up approach to building a universal grammar.

However, grammars and treebanks are getting more complicated and hard to maintain in the course of development. This is brought about by two factors; one is the linguistically sophisticated nature of DELPH-IN grammars and treebanks, and the other is involved with difficulties in communication during the collaboration The former comes about as a natural result of hand-crafting large-scale HPSG grammars and treebanks for practical NLP purposes, and the latter is a natural consequence of a collaboration in which participants are located away from each other, speak different languages and have different backgrounds and interests.

At this point, the **ltdb** comes on stage. It plays two roles; one is to automatically summarize a grammar and a treebank for each language in terms of lexical types, and the other is to show the summary to developers around the world through the Web. With the **ltdb**, a developer can grasp the big picture of the grammar and the treebank no matter how large they are, and developers distributed over the world can share the big picture.

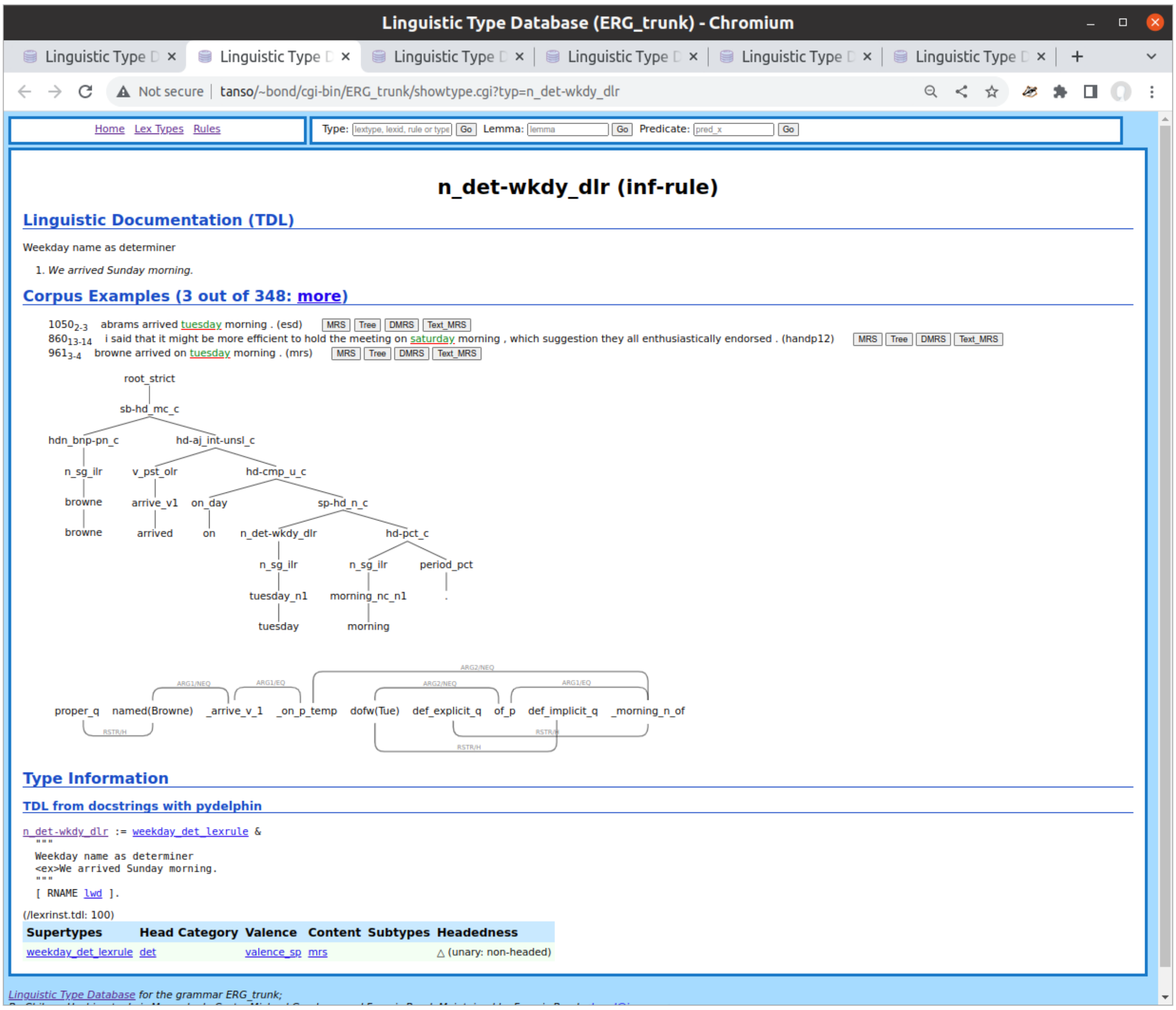
In addition to alleviating the complication of a large-scale grammar and treebank, the **ltdb** helps to facilitate the understanding of commonalities of human languages. This is because the **ltdb** reveals the linguistic essence of a grammar in terms of lexical types for whatever language it deals with. If developers who are in charge of a particular language show the linguistic essence of the grammar by means of the **ltdb**, other developers can compare it with their own grammars easily through the Web.

Structure

Documentation is stored with the grammar itself, as docstrings on types, following Dini and Mazzini (1997), in the style of *literate programming* (Knuth, 1992). This ensures that the grammarian can easily refer to it when they are developing the grammar, and makes it harder to get out of sync.

The web interface shows an easier to interpret subset of information:

- The name of the type
- The docstring (formatted)
- Positive and Negative examples from the docstring
- Examples from the corpus (for instances such as lexical types, rules and roots). We can show derivation trees and MRS for these.



Linguistic Type Database (ERG_trunk) - Chromium
Linguistic Type D ×
Linguistic Type D ×
Linguistic Type D ×
Linguistic Type D ×
Linguistic Type D ×
Linguistic Type D ×
Not secure | tanso/~bond/cgi-bin/ERG_trunk/showtype.cgi?typ=n_det-wkdy_dlr
Home Lex Types Rules
Type: lextype, lexid, rule or type Go Lemma: lemma Go Predicate: pred_x Go

n_det-wkdy_dlr (inf-rule)

Linguistic Documentation (TDL)

Weekday name as determiner

1. *We arrived Sunday morning.*

Corpus Examples (3 out of 348: more)

$1050_{2\text{-}3}$ abrams arrived tuesday morning . (esd) MRS Tree DMRS Text_MRS
$860_{13\text{-}14}$ i said that it might be more efficient to hold the meeting on saturday morning , which suggestion they all enthusiastically endorsed . (handp12) MRS Tree DMRS Text_MRS
$961_{3\text{-}4}$ browne arrived on tuesday morning . (mrs) MRS Tree DMRS Text_MRS

Type Information

TDL from docstrings with pydelphin

```
n_det-wkdy_dlr := weekday_det_lexrule &
  """
  Weekday name as determiner
  <ex>We arrived Sunday morning.
  """
  [ RNAME lwd ].
```

(/lexrinst.tdl: 100)

Supertypes	Head Category	Valence	Content	Subtypes	Headedness
weekday_det_lexrule	det	valence_sp	mrs		△ (unary: non-headed)

Linguistic Type Database for the grammar ERG_trunk;

Figure 1: **ltdb** for `n_det-wkdy_dlr`

- The parent and daughter types
- A few specially linguistically interesting types
  - category
  - valence
  - content

To enable debugging, it also shows the complete source of the type (as TDL: Krieger and Schäfer, 1994), which file it appears in and on what line the definition starts. We give an example of a rule in 1.

Each time the grammar is revised based on treebank annotation feedback, grammar developers consult the database to see the current status of the grammar. After finishing the revision, the grammar and **ltdb** are updated. Each time the treebank is annotated, annotators can consult the database to make sure the chosen parse is correct.

As much as possible, we are integrating the **ltdb** infrastructure into the existing grammar development environment, so that any grammar developer can take advantage of it with almost no additional effort. To this end, we have simplified the software (to python from java and perl) and host the source code on github: `https://github.com/fcbond/ltdb`.

Conclusion

We have constructed a web-based framework for supporting collaborative multilingual grammar and treebank development in which developers are distributed around the world. Our framework, which we call the **ltdb**, tells developers around the world about lexical types of the grammar and treebank they are developing. Lexical types can be seen as very detailed parts-of-speech and are the essence for the two important points just mentioned.

We have applied the **ltdb** to grammars and treebanks of Japanese and English, in the near future we plan to make the framework available to other grammar developers, and allow them to create their own **ltdb**.

References

Emily M. Bender, Dan Flickinger, and Stephan Oepen. The grammar matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, pages 8–14, Taipei, Taiwan, 2002.

Francis Bond, Eric Nichols, Sanae Fujita, and Takaaki Tanaka. Acquiring an ontology for a fundamental vocabulary. In *20th International Conference on Computational Linguistics: COLING-2004*, pages 1319–1325, Geneva, 2004.

Francis Bond, Stephan Oepen, Eric Nichols, Dan Flickinger, Erik Velldal, and Petter Haugereid. Deep open source machine translation. *Machine Translation*, 25(2):87–105, 2011. URL `http://dx.doi.org/10.1007/s10590-011-9099-4`. (Special Issue on Open source Machine Translation).

Ann Copestake, Dan Flickinger, Ivan A. Sag, and Carl Pollard. Minimal Recursion Semantics. An introduction. *Research on Language and Computation*, 3(4):281–332, 2005.

Luca Dini and Giampolo Mazzini. Hypertextual grammar development. In *Computational Environments for Grammar Development and Linguistic Engineering*, pages 24–29, Madrid, 1997. ACL.

Zhenzhen Fan, Francis Bond, and Sanghoun Song. An hpsg-based shared-grammar for the chinese languages: Zhong [|],. In *ACL 2015 Workshop on Grammar Engineering Across Frameworks (GEAF 2015)*, 2015.

Dan Flickinger. On building a more efficient grammar by exploiting types. *Natural Language Engineering*, 6(1):15–28, 2000. (Special Issue on Efficient Processing with HPSG).

Chikara Hashimoto, Francis Bond, and Dan Flickinger. The lextype DB: A web-based framework for collaborative multilingual grammar and treebank development. In

First International Workshop on Intercultural Collaboration (IWIC-2007), pages 44–58, 2007a.

Chikara Hashimoto, Francis Bond, Takaaki Tanaka, and Melanie Siegel. Semi-automatic documentation of an implemented linguistic grammar augmented with a treebank. *Language Resources and Evaluation*, 42(2):117–126, 2007b. URL `http://dx.doi.org/10.1007/s10579-008-9065-9`. (Special issue on Asian language technology).

Jong-Bok Kim, Jaehyung Yang, Sanghoun Song, and Francis Bond. Processing of Korean and the development of the Korean resource grammar. *Linguistic Research*, 28 (3):635–672, 2011.

Donald E. Knuth. *Literate Programming*. CSLI Publications, 1992.

Hans-Ulrich Krieger and Ulrich Schäfer. TDL — A type description language for constraint-based grammars. In *15th International Conference on Computational Linguistics: COLING-94*, pages 893 – 899, Kyoto, Japan, 1994.

Montserrat Marimon. The Spanish DELPH-IN grammar. *Language Resources and Evaluation*, 47(2):371–397, 2012.

Ivan A. Sag, Tom Wasow, and Emily Bender. *Syntactic Theory: A Formal Introduction*. CSLI Publications, Stanford, 2 edition, 2003.

Melanie Siegel, Emily Bender, and Francis Bond. *Jacy — An Implemented Grammar of Japanese*. CSLI Publications, 2016. ISBN 9781684000180.

Pat Suppes, T. Liang, E. E. Macken, and Dan Flickinger. Positive technological and negative pre-test-score effects in a four-year assessment of low socioeconomic status K-8 student learning in computer-based math and language arts courses. *Computers & Education*, 71:23–32, 2014.

Kristina Toutanova, Christopher D. Manning, Dan Flickinger, and Stephan Oepen. Stochastic HPSG parse disambiguation using the Redwoods corpus. *Research on Language and Computation*, 3(1):83–105, 2005.