Improving Low-resource RRG Parsing with Structured Gloss Embeddings

Roland Eibers and **Kilian Evang** and **Laura Kallmeyer**
Heinrich Heine University Düsseldorf
Universitätsstr. 1, 40225 Düsseldorf, Germany
firstname.lastname@hhu.de

Abstract

Treebanking for local languages is hampered by the lack of existing parsers to generate pre-annotations. However, it has been shown that reasonably accurate parsers can be bootstrapped with little initial training data when use is made of the information in interlinear glosses and translations that language documentation data for such treebanks typically comes with. In this paper, we improve upon such a bootstrapping model by representing glosses using a combination of morphological feature vectors and pre-trained lemma embeddings. We also contribute a mapping from glosses to Universal Dependencies morphological features.

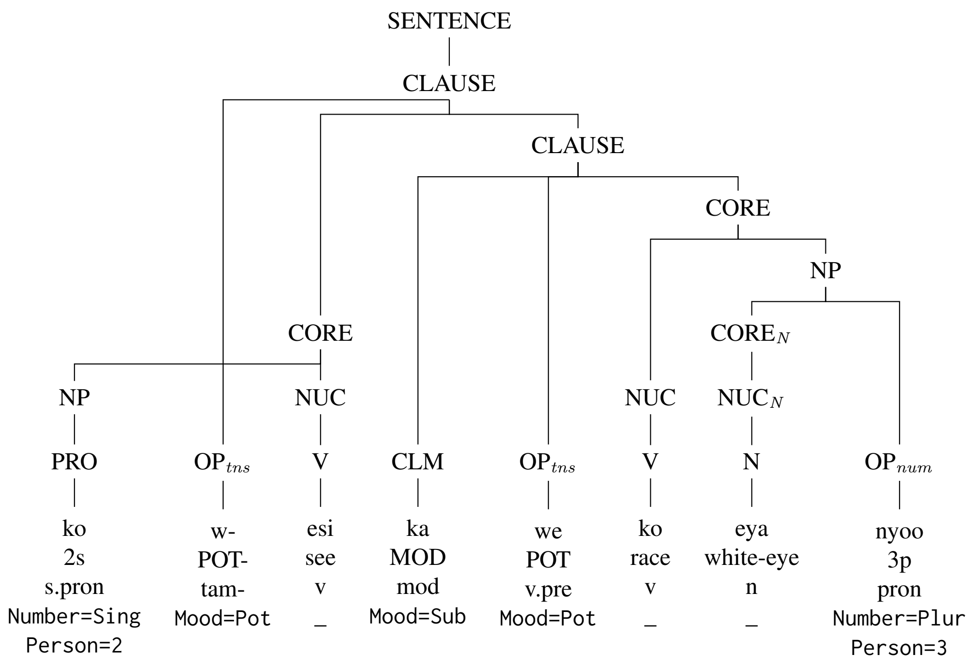
1 Introduction

Treebanking (i.e., annotating large corpora of sentences with syntactic structures) is an important tool for research into the syntax of natural language. Treebanking has long avoided starting from scratch, but used machine-generated pre-annotations that annotators correct (Marcus et al., 1993). For standardized languages, models generating the pre-annotations can nowadays rely on large language models and pre-trained parsers (e.g., Tyers et al., 2018; Jónsdóttir and Ingason, 2020; Bladier et al., 2022). For local languages, the situation looks quite different: usually, no large language models or other models are available. However, if the language is documented, the data usually comes with interlinear glosses and translations to a standardized language such as English (Lehmann, 1982). Evang et al. (2022) show that these annotations can be used to obtain more accurate pre-annotations for local-language treebanks by projecting contextualized word representations from a parser for English onto the target-language sentences, using character-based gloss embeddings, and self-training. In this paper, we show that the accuracy can be further improved by using a more structured representation for glosses. Our contributions are 1) a mapping from interlinear glosses to Universal Dependencies features that can be reused for other language documentation data, 2) based on that, a method for embedding glossed sentences using morphological feature vectors and lemma embeddings, and 3) an evaluation of this embedding method in the context of cross-lingual RRG parsing for treebank pre-annotation.

2 Related work

Low-resource RRG parsing Evang et al. (2022) consider the task of creating pre-annotations for treebanks for the Oceanic local languages Daakaka and Dalkalaen. The annotation scheme is based on that of RRGparbank (Bladier et al., 2022), following Role and Reference Grammar (RRG; Van Valin and Foley, 1980; Van Valin, 2005), a framework designed with diverse languages in mind. The text data for the treebanks comes with interlinear glosses and English translations, but only few have been hand-annotated with RRG trees. Figure 1 shows an annotated example Daakaka sentence. The basic pre-annotation model takes as input Daakaka token embeddings based on character-level LSTMs. It then labels each token with a supertag and a dependency head, which together serve as a derivation tree from which the final tree is constructed under the grammar formalism of Tree Wrapping Grammar (TWG; Kallmeyer et al., 2013). It is then shown that the accuracy of the basic model can be improved by 1) concatenating the token embeddings with similarly character-based gloss embeddings, 2) doing multiple rounds of self-training on unannotated data, and 3) using an English RRG parser (trained on substantially more gold standard data) on the translations and projecting contextualized word representations from the English parser to the Daakaka parser via unsupervised word alignments.

Proceedings of the Second Workshop on NLP Applications to Field Linguistics, pages 46–51
May 6, 2023 ©2023 Association for Computational Linguistics



"and you can see it chase away the white-eye"

Figure 1: RRG annotation of a Daakaka sentence, with its translation. Leaf nodes contain word form, glosses, POS tags and UD features. Glosses: 2s-second person singular, 3p-third person plural, POT-potential mood marker, MOD-complementizer or modal relator. *ka* is a polysemous morpheme with different functions. It can either be a complementizer introducing subjunctive clauses, or a modal relator, which changes a directive speech act into an assertion (von Prince, 2015). Both functions appear similarly glossed in the data and were grouped together as UD feature `Mood=Sub`.

Morphological feature embeddings Adding morphological features explicitly as input on NLP tasks has mixed effects, depending on the task and quality of features. Klemen et al. (2022) show across several languages that the results on (monolingual) dependency parsing and named entity recognition improve on LSTM-based models when UD feature embeddings are added as input, while the performance on comment filtering is not affected. Manually annotated features yield better results than automatically added features. Compared to our work, their approach assumes both a rich data set in the target language and high quality of UD features. An alternative method for encoding glossed words as tensors is described by Schwartz et al. (2022), but does not provide explicit mappings from glosses to feature-value pairs.

Lemma embeddings It is standard in modern NLP systems to represent words as vectors based on word associations in unannotated running text. One such model is FastText (Bojanowski et al., 2017). Less commonly, the same kind of model is trained on lemmatized text, e.g., in Sprugnoli et al. (2019); Ehren et al. (2020).

3 Method

We build on Evang et al.'s (2022) parsing architecture, as shown in Figure 2, with our modification concerning the embedding layer. While they use the same type of character-level LSTM to generate token embeddings, part-of-speech tag embeddings, and gloss embeddings, we seek to improve performance by using a more structured representation. Glosses consist of 1) translations of lemmas to English, and 2) codes representing morphological feature values. The gloss for one token can be seen as a partial function from features to feature values, so order does not matter and different values corresponding to the same feature are mutually exclusive. For example, the gloss `2s` can be represented as $\{(Number, Sing), (Person, 2)\}$. We exploit this by embedding glosses as a concatenation of feature embeddings like Klemen et al. (2022). Besides improving performance, we also aim to create a reusable compatibility layer between the glosses and Universal Dependencies (UD; de Marneffe et al., 2021), an annotation scheme commonly used in many data sets and tools. We therefore create the structured gloss embedding vectors via a mapping

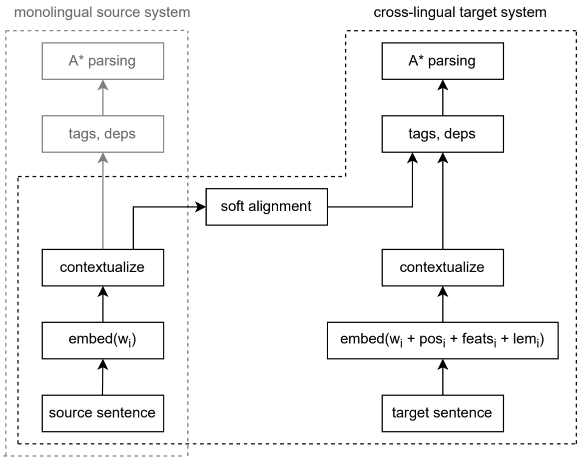


Figure 2: Architecture overview. Input on the target system includes embeddings of words, part-of-speech tags (pos), UD features, (feats) and English lemmas of target words embeddings. Words and pos tag embeddings are character-based, feats and lemmas are detailed below.

to the feature set defined by the UD annotation scheme. For the lemma translations, we exploit the fact that large quantities of text are available for English, and generate rich lemma embeddings. We now turn to the details of both contributions.

Construction of UD feature embeddings The mapping from glosses to UD features was performed with a conversion table, based on descriptions in von Prince (2015) and von Prince (2017) as well as UD guidelines. We focused on the glosses that occur in the Daakaka and Dalkalaen data (von Prince, 2013a,b). The feature `PronType` was added for pronouns, which are not particularly glossed in the data. A number of glosses were not converted to features, such as `EP` for epenthetic consonants /p/ and `ATT` for the morpheme *na*, which derives attributes from lexemes and simple phrases. Daakaka also distinguishes between three possessive classifiers glossed as `CL1`, `CL2` and `CL3` which show agreement with the lexical gender of the head noun or indicate their semantic domain (von Prince, 2015). As their function is mainly semantic and not syntactic, they were all represented as $\{(Poss, Yes)\}$. The gloss sets of both languages largely overlap; two glosses with low occurence appear only in the Dalkalaen data. We gathered 16 distinct features, 7 of which are unary (see Table 1 for an overview of the features). We did not encounter any cases where glosses on the same token mapped to conflicting values for the same feature.

Feature name	**Possible values**
Aspect	Inch*, Prog
Clusivity	In, Ex
Degree	Dim
Deixis*	Med, Prox, Remt
Derivation*	Nml
Mood	Ind, Irr, Pot, Sub
Number	Dual, Pauc, Plur, Sing
NumType	Card
Person	1, 2, 3
Polarity	Neg
Poss	Yes
PronType	Art, Dem, Int, Prs
Redup*	Yes
Tense	Fut, Past
Trans*	Yes
VerbType*	Aux, Cop

Table 1: Overview of UD features and possible values. * indicates that the feature or value is from a language-specific extension and not contained in the universal feature set.

For the UD feature embeddings, we follow the method described in Klemen et al. (2022). Each feature is passed through an individual embedding layer (non-present features receive a special input), yielding 3-dimensional embeddings. The final representation is a 48-dimensional vector, constructed by concatenating all feature embeddings.

Construction of lemma embeddings We use the FastText implementation of Gensim (Řehůřek and Sojka, 2010) to compute 300-dimensional lemma embeddings, trained on the lemma field of the ukWaC corpus (Baroni et al., 2009). The quality of embeddings differs across the data set. For instance, *yaapu* 'big.man' and *eya* 'white-eye' are full translations of Daakaka lemmas, however they do not appear in this form in the source corpus. The same goes for a number of names, e.g. *Simarongrong*, *Tamadu*.

4 Evaluation

We evaluate our UD feature+lemma embedding method by comparing against Evang et al.'s (2022) character-based method. We mirror their experimental setups, performing experiments across different scenarios (how much annotated seed training data is available), different amounts of self-training (adding 500 parses to the training data in each

rounds	0	1	2	3	4	5
mono, chars	67.9	69.5	70.1	70.7	70.9	70.5
mono, struct	67.9	68.5	69.5	69.5	70.2	71.2
mono, struct+lem	69.2*	70.1$_{\dagger}$	70.8$^{*}_{\dagger}$	70.6$_{\dagger}$	71.0$_{\dagger}$	**71.4**$^{*}_{\dagger}$
cross, chars	70.2	70.7	71.7	72.2	72.4	72.2
cross, struct	70.5	71.5*	71.7	72.1	72.2	**72.5**
cross, struct+lem	70.6	71.2*	71.8	72.3	72.4	72.3$_{\dagger}$

Table 2: Daakaka test f-scores in the **very low-resource** scenario (500 training sentences) for different models (monolingual vs. cross-lingual) and different types of gloss embeddings (character-based vs. structured vs. structured + lemma embeddings. The rounds of self-training increase from left to right. The scores are averaged over five runs, except for scores marked with $_{\dagger}$ where only four successful runs were available. Results with character-based embeddings are from Evang et al. (2022). Asterisks denote significant improvement ($p \leq .05$, permutation test) over the corresponding character-based model.

rounds	0	1	2	3	4	5
mono, chars	71.9	71.5	72.4	72.9	73.4	73.3
mono, struct	71.6	71.8	72.8	73.1	72.8	**73.7**
mono, struct+lem	72.2	73.0*	**73.7***	73.3	73.2	73.3
cross, chars	73.1	73.7	74.3	74.2	74.5	74.7
cross, struct	73.3	74.2	74.3	74.6	74.6	75.0$^{*}_{\dagger}$
cross, struct+lem	73.5	73.9	74.0	74.5	74.4	**75.1***

Table 3: Daakaka test f-scores in the **low-resource** scenario (1 000 training sentences).

round), and using the monolingual vs. the cross-lingual model. We compute the overall EVALB f-score (Collins, 1997) of each model on the same test set of 196 trees (Daakaka) resp. 101 trees (Dalkalaen).

In the "very low resource scenario" (500 annotated training sentences; Table 2), we find that structured embeddings tend to improve over character-based embeddings slightly, most significantly in the early stages of self-training. We take this as an indication that structured embeddings provide the information from the start that character-based ones have to learn over mutliple rounds of self-training. We also observe that the structured models seem more stable under self-training than character-based ones: between self-training rounds 4 and 5, the two character-based models lose accuracy whereas three out of four structured models still gain accuracy. Adding lemma embeddings tends to improve over using just morphological feature embeddings.

In the "low resource scenario" (1 000 annotated training sentences; Table 3), the structured models

rounds	0	1	2	3	4	5
cross, chars	69.0	71.8	72.4	73.0	73.6	73.2
cross, struct+lem	68.9	72.1	72.6$_{\dagger}$	73.0$_{\dagger}$	72.6$_{\dagger}$	73.1$_{\dagger}$

Table 4: Dalkalaen test f-scores in the **zero-shot** scenario (no in-language training sentences, but trained on 1 840 Daakaka sentences).

are also better than the corresponding character-based ones in most cases. In the monolingual model, only the model with lemmas gives significant improvement, and only in the early rounds of self-training. In the cross-lingual model, no significant improvement is seen until the fifth round of self-training. The gain from lemma embeddings also fades. We take this as an indication that with 1 000 training trees, the cross-lingual model is already relatively strong, and it gets harder for the structured embeddings to contribute more gains. We still take this as a positive result for the structured models, as they may be able to contribute when few data or no translations are available, or self-training is impossible or impractical.

In the "zero shot" scenario (parser trained on 1 840 Daakaka trees, tested on Dalkalaen; Table 4), the structured model with lemmas is mostly on par with the character-based one, but achieves no significant improvements. We find this surprising as one would think the zero-shot model relies more strongly on feature embeddings, which are more comparable than words between both languages, and would profit more from them being structured. Further research is needed to explain this.

5 Conclusions and Future Work

We have presented an alternative way to embed data from language documentation datasets, based on structured gloss embeddings and translation lemma embeddings. We have shown that (optionally in combination with cross-linguistically projected vectors), in the context of low-resource pre-parsing for RRG treebanking, these structured embeddings can sometimes improve over character-based embeddings, or decrease the model's reliance on self-training.

Perhaps more importantly, by creating structured gloss embeddings via translation rules from interlinear glosses into UD features, we have created the first part of a compatibility layer between both types of morphosyntactic annotation, and opened the way towards morphosyntactically informed model transfer, parameter sharing, etc., between models for documented local languages and models based on existing UD treebanks. We plan to explore this option in future work. We would also like to explore sharing encoders for glossed text between more diverse sets of languages, and study the effect of the translation language on the quality of the cross-lingual word representations.

Acknowledgements

We would like to thank the anonymous reviewers for their insightful comments. This work was carried out as a part of the research project TreeGraSP[1] funded by a Consolidator Grant of the European Research Council (ERC).

References

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43:209–226.

Tatiana Bladier, Kilian Evang, Valeria Generalova, Zahra Ghane, Laura Kallmeyer, Robin Möllemann, Natalia Moors, Rainer Osswald, and Simon Petitjean. 2022. RRGparbank: A parallel role and reference grammar treebank. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4833–4841, Marseille, France. European Language Resources Association.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Michael Collins. 1997. Three generative, lexicalised models for statistical parsing. In *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, pages 16–23, Madrid, Spain. Association for Computational Linguistics.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

Rafael Ehren, Timm Lichte, Laura Kallmeyer, and Jakub Waszczuk. 2020. Supervised disambiguation of German verbal idioms with a BiLSTM architecture. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 211–220, Online. Association for Computational Linguistics.

Kilian Evang, Laura Kallmeyer, Jakub Waszczuk, Kilu von Prince, Tatiana Bladier, and Simon Petitjean. 2022. Improving low-resource RRG parsing with cross-lingual self-training. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4360–4371, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Hildur Jónsdóttir and Anton Karl Ingason. 2020. Creating a parallel Icelandic dependency treebank from raw text to Universal Dependencies. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2924–2931, Marseille, France. European Language Resources Association.

Laura Kallmeyer, Rainer Osswald, and Robert D. Van Valin, Jr. 2013. Tree Wrapping for Role and Reference Grammar. In *Formal Grammar 2012/2013*, volume 8036 of *LNCS*, pages 175–190. Springer.

Matej Klemen, Luka Krsnik, and Marko Robnik-Šikonja. 2022. Enhancing deep neural networks with morphological information. *Natural Language Engineering*, page 1–26.

Christian Lehmann. 1982. Directions for interlinear morphemic translations. *Folia Linguistica*, 16:199–224.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Kilu von Prince. 2013a. *Daakaka, The Language Archive*. MPI for Psycholinguistics, Nijmegen.

Kilu von Prince. 2013b. *Dalkalaen, The Language Archive*. MPI for Psycholinguistics, Nijmegen.

Kilu von Prince. 2015. *A Grammar of Daakaka*. Mouton de Gruyter, Berlin, Boston.

Kilu von Prince. 2017. Daakaka dictionary. *Dictionaria*, (1):1–2167.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA. `http://is.muni.cz/publication/884893/en`.

Lane Schwartz, Coleman Haley, and Francis Tyers. 2022. How to encode arbitrarily complex morphology in word embeddings, no corpus needed. In *Proceedings of the first workshop on NLP applications to field linguistics*, pages 64–76, Gyeongju, Republic of Korea. International Conference on Computational Linguistics.

Rachele Sprugnoli, Marco Passarotti, and Giovanni Moretti. 2019. Vir is to moderatus as mulier is to

[1] `https://treegrasp.phil.hhu.de`

intemperans. Lemma embeddings for Latin. In *Proceedings of the Sixth Italian Conference on Computational Linguistics*, Torino, Italy. Accademia University Press.

Francis Tyers, Mariya Sheyanova, Aleksandra Martynova, Pavel Stepachev, and Konstantin Vinogorodskiy. 2018. Multi-source synthetic treebank creation for improved cross-lingual dependency parsing. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 144–150, Brussels, Belgium. Association for Computational Linguistics.

Robert D. Van Valin, Jr. 2005. *Exploring the Syntax-Semantics Interface*. Cambridge University Press.

Robert D. Van Valin, Jr. and William A. Foley. 1980. Role and reference grammar. In E. A. Moravcsik and J. R. Wirth, editors, *Current approaches to syntax*, volume 13 of *Syntax and semantics*, pages 329–352. Academic Press, New York.