Synthetic-Error Augmented Parsing of Swedish as a Second Language: Experiments with Word Order

Arianna Masciolini, Emilie Marie Carreau Francis, Maria Irena Szawerna
Språkbanken Text
Department of Swedish, Multilingualism, Language Technology
University of Gothenburg
{arianna.masciolini, emilie.francis, maria.szawerna}@gu.se

Abstract

Ungrammatical text poses significant challenges for off-the-shelf dependency parsers. In this paper, we explore the effectiveness of using synthetic data to improve performance on essays written by learners of Swedish as a second language. Due to their relevance and ease of annotation, we restrict our initial experiments to word order errors. To do that, we build a corrupted version of the standard Swedish Universal Dependencies (UD) treebank Talbanken, mimicking the error patterns and frequency distributions observed in the Swedish Learner Language (SweLL) corpus. We then use the MaChAmp (Massive Choice, Ample tasks) toolkit to train an array of BERT-based dependency parsers, fine-tuning on different combinations of original and corrupted data. We evaluate the resulting models not only on their respective test sets but also, most importantly, on a smaller collection of sentence-correction pairs derived from SweLL. Results show small but significant performance improvements on the target domain, with minimal decline on normative data.

Keywords: Dependency Parsing, Data Augmentation, Second Language Acquisition, L2 Swedish

1. Introduction and Background

In recent years, off-the-shelf dependency parsers have reached remarkably high performance on standard evaluation sets. This applies to many high and medium-resourced languages, including Swedish. Nonstandard language, however, still poses significant challenges. In a study on dependency parsing of learner English, Huang et al. (2018) showed that the tools available at the time were not robust to grammatical errors, despite misleadingly high overall accuracy scores. In a more recent study on L2 Swedish (Swedish as a second language), Volodina et al. (2022) note that, dependency parsing is especially problematic for standard tools, even when they perform reasonably well on other linguistic annotation tasks such as part-of-speech tagging.

A notable attempt to address this issue is the error-repairing parser introduced by Sakaguchi et al. (2017), specifically meant for ungrammatical texts. This approach combines parsing with Grammatical Error Correction (GEC). In many contexts, such as Second Language Acquisition (SLA) research, it can however be preferable to analyze learner texts as they are and, in some cases, to compare originals with their normalized versions. We therefore test the more straightforward approach of fine-tuning a Bidirectional Encoder Representations from Transformers (BERT, Devlin et al. 2018) model for dependency parsing on data that resembles our target domain, L2 Swedish.

With an approach loosely inspired by Stymne et al. (2023), we use the MaChAmp (Massive Choice, Ample tasks) toolkit (van der Goot et al., 2021) to fine-tune an array of models on different combinations of a treebank of standard Swedish and an artificially corrupted version of the same dataset. Crucially, the evaluation step involves not only normative data and artificial errors, but also authentic L2 Swedish sentences.

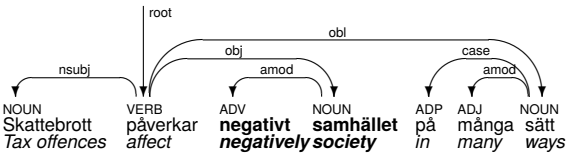


Figure 1: A sentence with incorrect word order parsed with UDPipe 2. Note how the adverb *negativt* is both attached to the wrong token (the noun *samhället*, rather than the main verb *påverkar*) and incorrectly labelled as an adjectival modifier (`amod`) instead of as an adverbial one (`advmod`).

For this first experiment, we restrict ourselves to word order errors. This is out of both principled and practical reasons. On the one hand, as illustrated by the example in Figure 1, it seems reasonable to assume syntax errors to be challenging for a tool that performs syntactic analysis. When it comes to word order errors specifically, this should be especially true for a language with relatively strict word order such as Swedish. At the same time, word order errors appear to be easier to generate and automatically annotate than most other error types: as tokens are swapped without being altered, token-

Proceedings of the Joint Workshop on Multiword Expressions and Universal Dependencies (MWE-UD 2024)
@LREC-COLING-2024, pages 43–49. May 25, 2024.

level linguistic annotation can be easily transferred from a sentence in standard language to its corresponding corrupted version.

2. Data

We utilize three datasets: an L2 Swedish test set, described in Section 2.1, a standard Swedish treebank and an artificially corrupted version of the latter (cf. Section 2.2). Train-dev-test split sizes are outlined in Table 1.

2.1. SweLL

Our target domain data comes from the SweLL Swedish Learner Language corpus (Volodina et al., 2019), a collection of over 500 essays written by learners of L2 Swedish. More specifically, we use SweLL-gold, the manually pseudonymized version of the corpus (Volodina et al., 2022).[1] L1 backgrounds vary, as well proficiency levels, which range from beginner to advanced. Learner texts are paired with correction hypotheses[2] and each error is classified according to the taxonomy discussed in Rudebeck and Sundberg (2021).

For our purposes, the relevant categories are, in decreasing order of frequency, S-Adv (misplaced adverbial), S-FinV (misplaced finite verb), and S-WO, which encompasses all other word order errors. About 15% of SweLL sentences are marked with one of these labels. In the vast majority of the cases, however, word order errors co-occur with other issues, often overlapping in ways that make the former hard to isolate. After filtering out these cases, we were left with a 69-sentence evaluation set. Regrettably, the resulting sentences tend to be shorter than the corpus-wide average.

2.1.1. Linguistic Annotation

While a linguistically annotated version of SweLL is available, it is not manually validated nor does it follow the UD standard. We therefore opted for completely re-annotating our test set. We started by parsing the correction hypotheses with the UDPipe 2 parser (Straka, 2018) using the UD 2.12 model (Straka, 2023) trained on Talbanken (cf. Section 2.2). The first and third authors, both graduate students in Computational Linguistics, manually validated the resulting parse trees with particular attention to the segments that diverged from the corresponding original learner sentences. This manual annotation step only concerned the `DEPREL` and `HEAD` columns of the fully-annotated CoNLL-U files obtained from UDPipe 2, as our models are only trained for UD parsing in its strictest sense.

To annotate L2 originals, we used an *ad-hoc* script which transfers token-level annotations from gold-annotated corrections to L2 originals. Each sentence is first rewritten in the vertical format customary for CoNNL-U files. Then, each token is annotated as follows:

- a token `ID` is assigned sequentially;
- all other fields excepts `HEAD` (syntactic head) are copied from the first unused token of the sentence's correction hypothesis presenting the same word `FORM`. Such token is then immediately marked as used, to deal with cases where the same word occurs multiple times in the same sentence;
- the `HEAD` field is assigned the `ID` of the nearest token in the learner sentence whose `FORM` matches that of the syntactic head of the corresponding corrected token.

Choosing syntactic heads based on the closest homograph is a heuristic that occasionally produces ill-formed trees. For this reason, we also inspected the results of this processing step and made the necessary manual edits.

2.2. Talbanken

For training, we used the UD 2.12 version of Talbanken, a widely used treebank of written and spoken modern Swedish (Einarsson 1976, Nivre and Smith 2023). Due to MaChAmp not supporting the enhanced UD format, the treebank was preprocessed with the cleanup script provided as part of the toolkit itself. Its training portion was then used to fit our baseline model with no further changes. Mimicking the error patterns observed in SweLL, we also built a corrupted version of such a treebank, which we used in conjunction with the original upon training our specialized models (cf. Section 3).

2.2.1. Corruption Process

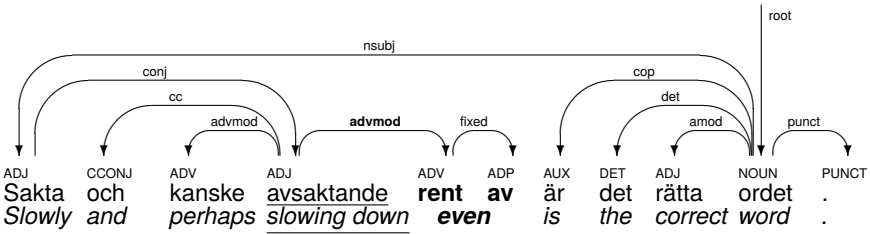
Synthetic error generation is a common task in the field of GEC. Closest to this work is the text corruption method described in Casademont Moner and Volodina (2022), which has been used to build a corpus of Swedish sentences presenting verb

	Train	Dev	Test
SweLL	-	-	69
Talbanken	4303	504	1219
Corrupted	4303	504	1219

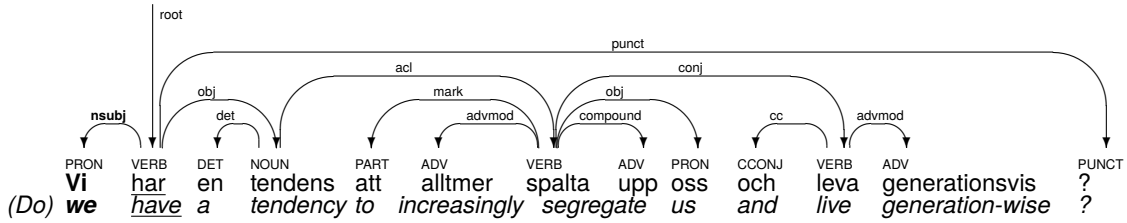
Table 1: Sizes of the training, development and test splits of our datasets in number of sentences.

[1] For conciseness, we refer to SweLL-gold as SweLL.

[2] Annotators often need to guess the learner's communicative intent. For this reason, we refer to normalized sentences as correction hypotheses.



(a) Synthetic S-Adv error. The entire `advmod` subtree is swapped with the pivot (original sentence: "Sakta och kanske **rent av** avsaktande är det rätta ordet").



(b) Synthetic S-FinV error. Note that Swedish features reversed word order in questions (original sentence: "Har **vi** en tendens att alltmer spalta upp oss och leva generationsvis?").

Figure 2: Two corrupted sentences obtained via subtree swapping. The rearranged segments are higlighted in bold; their syntactic heads, acting as pivot elements, are underlined.

order errors using L2 Swedish textbooks as a starting point. We propose a simpler but more general method that covers all three classes of word order errors mentioned in Section 2.1 while preserving UD annotation.

From an operational point of view, such an approach resembles that of Şahin and Steedman (2018), who rely on dependency annotation to "rotate" sentences by swapping subtrees around roots. When it comes to misplaced adverbials (S-Adv), subtrees labelled as adverbial modifiers (`advmod`) or clauses (`advcl`) are swapped with their syntactic heads (see Figure 2a for an example). S-FinV errors are generated by swapping finite verbs with their subjects (a `nsubj`- or `csubj`-labelled subtree[3], cf. Figure 2b). As for S-WO, with a drastic simplification, we always swap two randomly selected adjacent tokens. After each rotation, the `ID`s of the corrupted sentence are reassigned sequentially and dependency `HEAD`s adjusted accordingly, thus ensuring the correctness of the annotation for the resulting corrupted tree.

We tried as much as possible to replicate the error distribution observed in SweLL. For each Talbanken sentence, our corruption script tries to generate three different scrambled sentences (one per error category) and chooses one based on its label's relative frequency in the corpus. Obviously, however, the S-Adv corruption rule cannot be applied to sentences with no adverbials. There are also instances where finite verbs (typically imperatives) lack an explicit subject or, more rarely, where sentences contain no finite verbs at all. In both cases, we revert to one of the other two categories.

[3] If the finite verb in question is an auxiliary, we look for the subject of the head lexical verb.

3. Models

Name		% Normative	% Errors
BASELINE		100	0
MIX15		85	15
MIX50		50	50
SEQ10	(step 1)	100	0
	(step 2)	0	100
SEQ20	(step 1)	100	0
	(step 2)	0	100

Table 2: Our models and the data configurations they were trained on.

We used the MaChAmp toolkit to fine-tune a BERT model for dependency parsing using the original and corrupted Talbanken datasets in different configurations, summarized in Table 2. MaChAmp simplifies the fine-tuning of language models for a variety of NLP tasks including dependency parsing (van der Goot et al., 2021). It is relatively simple to set up with the desired hyperparameters and allows for the fine-tuning of various contextualized word embeddings. While we do not leverage the toolkit's multi-task learning functionalities, we have selected it for its ease of use and sequential fine-tuning. We ran the toolkit with the default hyperparameters, with the exception of changing the default model to the monolingual Swedish BERT (Malmsten et al., 2020) and altering the number of epochs in one

of our sequential models (SEQ10 was only further fine-tuned on 10 epochs of corrupted data, not the default 20).

All in all, we have fine-tuned BERT for Swedish five times, resulting in five final models. The first model we fine-tuned purely on Talbanken, as a baseline (BASELINE), in order to know what results fine-tuning only on normative data yields. Our first specialized model, MIX15, utilized a combination of normative data and synthetic errors that was meant to mimic the relative frequency of this kind of errors in the learner data. In order to see whether increasing that relative frequency would have a detrimental effect on a model, we fine-tuned one with equal parts of normative and corrupted data, MIX50. We also experimented with sequential training to further fine-tune the BASELINE model with 10 and 20 epochs of only corrupted data (SEQ10 and SEQ20, respectively), to investigate whether the performance of an existing dependency parser could be improved by retraining it on non-normative language.

4. Evaluation

Model accuracy was evaluated in terms of Labelled and Unlabelled Attachment Scores, LAS and UAS. To check for statistical significance, these were calculated for each parse tree and compared against a baseline trained on standard Talbanken data to determine if the difference in model performance was significant. A paired t-test with a 95% confidence interval and $\alpha = 0.05$ was used with the Bonferroni correction to compensate for multiple tests against the baseline. Both the UAS score and LAS score were tested against the baseline, so it is possible for only one of the scores to be statistically significant. For nearly all cases, with the exception of Seq20 SweLL (Table 4), either both scores or neither were found to be significant.

Performance on target domain data was assessed on the SweLL-derived test set described in Section 2.1. The models were also evaluated on the original Talbanken test set and its corrupted version (cf. Section 2.2). Talbanken was included to assess whether the addition of ungrammatical examples resulted in a performance decline on normative data, while SweLL allowed for comparison of results on actual learner errors. The expectation was to see a substantial performance increase on corrupted Talbanken instances and a smaller improvement on authentic examples. When it comes to normative data, the ideal outcome would be for the fine-tuning on artificial errors to not have any negative repercussions.

Targeted Evaluation To further analyse how this method affects word order errors, a more targeted evaluation was performed using a modified version of the SweLL test set. Following Berzak et al. (2016), we assumed tokens belonging to erroneous segments to be more likely to be incorrectly parsed, even though annotation errors might cascade to other parts of the sentences. Errors were isolated from learner sentence-correction pairs by removing tokens preceding and following the diverging segment. Attachment scores were then recomputed on the resulting sentence fragments.[4]

4.1. Results and Discussion

Overall average scores are summarized in Table 3. Performance results suggest that exposure to synthetic word order errors in training has a positive effect on the models' ability to handle the (in-domain) corruputed sentences, matching our expectations. Simultaneously, performance decline on normative data is contained. Addressing the central question of whether improvement on synthetic data transfers to actual learner sentences, a slight positive effect on similar errors in out-of-domain texts can be observed. Smaller performance gains on out-of-domain texts may be attributed to synthetic errors not being sufficiently similar to authentic examples, to differences between training and test domains beyond mere grammaticality, or a combination of the two. It must also be taken into account that the margin of improvement on learner sentences is smaller than on artificial errors. On artificially corrupted sentences, the baseline's performance

[4] Postprocessing often result in ill-formed trees, but this does not affect either performance metric.

	Talbanken		Corrupted		SweLL	
	LAS	**UAS**	**LAS**	**UAS**	**LAS**	**UAS**
BASELINE	92.42	94.30	80.20	83.29	88.28	91.16
MIX15	92.23	94.05	87.96	90.50	87.63	90.60
MIX50	91.54	93.58	89.59	92.00	89.86	92.93
SEQ10	92.20	94.06	90.47	92.75	90.05	92.84
SEQ20	92.53	94.32	90.95	93.08	89.02	92.00

Table 3: Overall attachment scores sets for all fine-tuned models. Cells with a grey background indicate that the difference between the scores for the baseline and fine-tuned models is statistically significant.

drops by about 10% for both metrics, while scores stay reasonably high on SweLL. Notably, on the other hand, specialized models perform very similarly on both non-normative datasets. The SEQ10 model performed best across all test sets except Talbanken.

4.1.1. Talbanken

The Talbanken set showed the highest performance overall, with the baseline achieving a LAS of 92.42% and an UAS of 94.3%. This observation is expected, as the models were for the most part trained on the same domain (Talbanken data). Performance with the fine-tuned models generally decreased, but only MIX50 and SEQ10 showed a result that was significantly different compared to the baseline. It appears that exposing the model to atypical word order has little impact on performance for the Talbanken domain.

4.1.2. Corrupted Talbanken

Results for the corrupted Talbanken set showed the largest increase in performance compared to the baseline, about an 8 to 10% increase, and the differences were statistically significant.[5] The SEQ10 and SEQ20 models showed the biggest improvement, with a 10% increase over the baseline. This confirms the viability of the fine-tuning approach for specialized UD parsers, at least when target domain data is available.

4.1.3. SweLL

Most specialized models exhibited small performance improvements against the baseline. However, just the SEQ10 model's improvement was significant. Interestingly, the only model that declined in performance, MIX15, was the one exposed to a percentage of errors corresponding to the one observed in SweLL-gold, which appears not to be enough to produce a positive effect.

A further encouraging signal comes from the targeted evaluation. When we focus on ungrammatical fragments, we see that the performance gap between the baseline and all the specialized models widens (cf. Table 4). Not only does this confirm the baseline's vulnerability to grammatical errors, but it also suggests that the models are learning something about non-normative word order, rather than just exhibiting a general improvement due to exposure to additional training data.

5. Conclusions and Future Work

We generated synthetic word order errors and used them to fine-tune a number of dependency parsers.

[5] p=0.00000000000000022, per paired t-test.

	LAS	UAS
BASELINE	82.80	86.02
MIX15	84.41	89.25
MIX50	87.10	90.32
SEQ10	87.10	89.78
SEQ20	86.02	89.78

Table 4: Attachment scores for the targeted evaluation on the SweLL-based test set. Cells with a grey background indicate that the difference between the scores for the baseline and fine-tuned models is statistically significant.

We evaluated them on (1) normative data, (2) synthetic error data, and (3) authentic L2 sentences containing errors of the same kind. The improvement on the latter was small, but significant. No substantial decrease in performance on normative data was observed, which suggests this is a promising method to increase parser robustness.

Future work aimed at achieving a more significant performance increase on target domain data should revolve around improving the corruption pipeline, especially when it comes to S-WO errors. The choice of material to corrupt is also important. In fact, we believe that applying our method to sentences from a domain closer to learner essays could result in better performance. It would also be beneficial to either have a larger test set or compare models in terms of multi-run averages in the future in order to more confidently assert that the differences between fine-tuning methods are not accidental. Other interesting possibilities are trying to run a hyperparameter search for at least some of the models and seeing how a multilingual model compares to the monolingual one we employed.

To ensure that our method is actually applicable to learner data in a more general sense, a possibility is to add one more test set where word order errors co-occur with other issues. Finally, a central question is to what extent our approach can be generalized to handle other kinds of errors (such as missing or redundant tokens, lack of agreement, etc.), and, most importantly, whether it can be adapted to handle sentences with multiple errors of various kinds.

6. Data and Code

The SweLL-derived test set and code are available at `github.com/spraakbanken/seapass`.

7. Ethical Concerns

While linguistic data can contain personal information, raising privacy concerns, neither of the datasets used in this experiment is likely to leak sen-

sitive information. Aside from its age, Talbanken consists of texts from genres like textbooks and articles, which are unlikely to contain information that should not be shared. As for SweLL-gold, a corpus that is both more recent and more likely to contain sensitive information due to its domain (L2 learner essays), all of the elements considered to be sensitive have been replaced with pseudonyms during corpus creation, and appropriate written consent had been obtained during the data collection step. Therefore, we consider the privacy risks of using these two datasets to be minimal.

8. Acknowledgments

SweLL-gold is part of Språkbanken, the Swedish national research infrastructure. Furthermore, the experiments presented in this paper are preliminary to the release of a UD version of such corpus, which will in turn enrich the existing infrastructure. On this basis, this research is supported by Nationella Språkbanken, funded jointly by the Swedish Research Council (2018–2024, contact 2017-00626) and the ten participating partner institutions.

9. Bibliographical References

Yevgeni Berzak, Jessica Kenney, Carolyn Spadine, Jing Xian Wang, Lucia Lam, Keiko Sophie Mori, Sebastian Garza, and Boris Katz. 2016. Universal Dependencies for learner English. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 737–746, Berlin, Germany. Association for Computational Linguistics.

Judit Casademont Moner and Elena Volodina. 2022. Generation of synthetic error data of verb order errors for Swedish. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 33–38, Seattle, Washington. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Jan Einarsson. 1976. *Talbankens skriftspråkskonkordans*. Institutionen för nordiska språk, Lunds universitet.

Yan Huang, Akira Murakami, Theodora Alexopoulou, and Anna Korhonen. 2018. Dependency parsing of learner English. *International Journal of Corpus Linguistics*, 23(1):28–54.

Lisa Rudebeck and Gunlög Sundberg. 2021. SweLL correction annotation guidelines. In *The SweLL guideline series nr 4*, Gothenburg, Sweden. Institutionen för svenska, Göteborgs Universitet.

Gözde Gül Şahin and Mark Steedman. 2018. Data augmentation via dependency tree morphing for low-resource languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5004–5009, Brussels, Belgium. Association for Computational Linguistics.

Keisuke Sakaguchi, Matt Post, and Benjamin Van Durme. 2017. Error-repair dependency parsing for ungrammatical texts. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 189–195, Vancouver, Canada. Association for Computational Linguistics.

Milan Straka. 2018. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium. Association for Computational Linguistics.

Sara Stymne, Carin Östman, and David Håkansson. 2023. Parser evaluation for analyzing Swedish 19th-20th century literature. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 335–346, Tórshavn, Faroe Islands. University of Tartu Library.

Rob van der Goot, Ahmet Üstün, Alan Ramponi, Ibrahim Sharaf, and Barbara Plank. 2021. Massive choice, ample tasks (MaChAmp): A toolkit for multi-task learning in NLP. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 176–197, Online. Association for Computational Linguistics.

Elena Volodina, David Alfter, Therese Lindström Tiedemann, Maisa Susanna Lauriala, and Daniela Helena Piipponen. 2022. Reliability of automatic linguistic annotation: native vs non-native texts. In *Selected papers from the CLARIN Annual Conference 2021*. Linköping University Electronic Press (LiU E-Press).

Elena Volodina, Lena Granstedt, Arild Matsson, Beáta Megyesi, Ildikó Pilán, Julia Prentice, Dan Rosén, Lisa Rudebeck, Carl-Johan Schenström, Gunlög Sundberg, and Mats Wirén. 2019. The SweLL language learner corpus: From design to annotation. *Northern European Journal of Language Technology*, 6:67–104.

10. Language Resource References

Martin Malmsten and Love Börjeson and Chris Haffenden. 2020. *Swedish BERT models*. National Library of Sweden / KBLab. Distributed via HuggingFace.

Nivre, Joakim and Smith, Aaron. 2023. *Swedish-Talbanken-UD*. Universal Dependencies Consortium. Distributed via LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University as part of Universal Dependencies 2.13. PID http://hdl.handle.net/11234/1-5287.

Straka, Milan. 2023. *Universal Dependencies 2.12 models for UDPipe 2*. Distributed via LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University as part of Universal Dependencies 2.12. PID http://hdl.handle.net/11234/1-5200.

Volodina, Elena and Granstedt, Lena and Matsson, Arild and Megyesi, Beáta and Pilán , Ildikó and Prentice, Julia and Rosén, Dan and Rudebeck, Lisa and Schenström, Carl-Johan and Sundberg, Gunlög and Wirén, Mats. 2022. *SweLL-gold*. Språkbanken Text. Distributed via SBX/CLARIN. PID https://hdl.handle.net/10794/846.