Computational Linguistics and Intellectual Technologies:
Proceedings of the International Conference "Dialogue 2025"

April 23–25, 2025

CoBaLD Parser: Joint Morphosyntactic and Semantic Annotation

Ilia Baiuk
MIPT
Russia, Moscow
ilia.baiuk.hu@gmail.com

Alexandra Baiuk
RSUH
Russia, Moscow
alexandra.m.baiuk@gmail.com

Maria Petrova
RSUH, Smart Data Expert
Russia, Moscow
maria.ap.world@gmail.com

Abstract

Dependency parsing is a common task for modern NLP, and Universal Dependencies (De Marneffe et al., 2021) is widely acknowledged nowadays as a morphosyntactic annotation standard. Yet, its dependency relations are rather generalized, therefore, in order to take more syntactic details into account, the Enhanced UD standard was proposed. A newly developed CoBaLD annotation standard elaborates the E-UD principles by enriching it with the semantic level. It is aimed at structural simplicity and the compatibility with UD in all possible issues. Currently, there are several datasets annotated in CoBaLD standard, but until now, there has been no appropriate tool for automatic data parsing in CoBaLD format. In this paper, we present a neural-based joint parser capable of automatic annotation both in E-UD and in CoBaLD, including ellipsis restoration which is supposed by these standards. Additionally, we provide a qualitative analysis of automatic annotation errors.

Keywords: morphosyntactic parsing, semantic parsing, joint parsing, Enhanced Universal Dependencies, CoBaLD annotation standard

DOI: 10.28995/2075-7182-2025-23-XX-XX

Парсер CoBaLD: интегральная морфосинтаксическая и семантическая разметка

Илья Баюк
МФТИ
Россия, Москва
ilia.baiuk.hu@gmail.com

Александра Баюк
РГГУ
Россия, Москва
alexandra.m.baiuk@gmail.com

Мария Петрова
РГГУ, Smart Data Expert
Россия, Москва
maria.ap.world@gmail.com

Аннотация

Задача лингвистической разметки – одна из центральных задач в современной обработке естественного языка (NLP), а стандарт Universal Dependencies (De Marneffe et al., 2021) является наиболее распространненным стандартом морфосинтаксической разметки. Данный стандарт, однако, не содержит семантического уровня, а его синтаксические зависимости носят несколько обобщенный характер. В качестве продолжения UD был предложен стандарт Enhanced UD, позволяющий больше детализировать синтаксические отношения. Новый стандарт разметки CoBaLD развивает принципы E-UD, добавляя к ним уровень семантической аннотации. Его основные цели – простота использования, максимальная совместимость с UD и обогащение UD и Enhanced UD семантической разметкой. Существует ряд датасетов, размеченных в стандарте CoBaLD, однако до настоящего момента не существовало подходящего инструмента для автоматической разметки данных в данном формате. В настоящей работе мы представляем интегральный парсер для CoBaLD, выполняющий автоматическую разметку как в формате E-UD, так и в формате CoBaLD. Важным достижением является также способность восстанавливать эллипсис, предусмотренный этими стандартами. Кроме того, мы проводим качественный анализ ошибок автоматической аннотации.

Ключевые слова: морфосинтаксическая разметка, семантическая разметка, лингвистическая разметка, Enhanced Universal Dependencies, CoBaLD

1 Introduction

Linguistic annotation is an important NLP task. In addition to theoretical purposes, annotated text corpora can also be used for solving different downstream tasks, such as the creation of chat-bots or question-answering systems, for instance. Most annotation schemes provide morphological and syntactic markup, nevertheless, there are standards which suppose semantic or discourse markup as well.

One of the most popular standards nowadays is Universal Dependencies (UD, (De Marneffe et al., 2021)) – a project aimed to capture the so-called dependencies in a language. It includes morphological and syntactic levels, although the relations UD annotates are not purely syntactic due to its principles concerning linguistic universality. Moreover, the UD project offers its enhanced syntax version, Enhanced UD (E-UD, (Schuster and Manning, 2016)), which suggests a more detailed syntax annotation and ellipsis restoration.

On the other hand, there are a few standards of semantic annotation, though none of them is universally accepted as of now. Some models are purely semantic (such as Abstract Meaning Representations (Banarescu et al., 2013) or Universal Cognitive Conceptual Annotation (Abend and Rappoport, 2013)), others include several language levels (Prague Tectogrammatical Graphs (Mikulová et al., 2006) or the ETAP model (Apresian et al., 2003)). There are projects as well developed to be compatible with UD, namely, Universal Decompositional Semantics, or UDS (White et al., 2016) and CoBaLD Annotation Project (Petrova et al., 2023; Petrova et al., 2024).

UDS suggests only the semantic annotation for the UD morphosyntax, presented in the form of different semantic attributes rather than labels for definite semantic meanings, which may make its usage inconvenient for solving practical tasks and parser learning.

CoBaLD includes both morphosyntax and semantics, striving to be as compatible with UD/E-UD as possible in the description of morphosyntax and enriching it with the semantical pattern. Its semantics provides both lexical meanings and the relations between words. The former are presented in the form of semantic classes (SCs, or semantic fields) organized in a thesaurus-like structure, the latter – in the form of deep slots (DSs, or semantic roles). Their main feature is that the DSs in CoBaLD include not only arguments, but adjuncts, attributes and all other dependencies as well. The descriptions and lists of the DSs and the SCs can be found on our Github page[1].

In UD, the annotation is presented in the CONLL-U Plus file standard[2], which looks like a table, where each column corresponds to some annotated meaning and each row – to the annotated token. As the key purpose of the CoBaLD standard is to add the semantic domain to the UD/E-UD morphosyntax, it just adds two new columns with the semantic information in the CONLL-U Plus, that is, a column for the DSs and a column for the SCs.

The only difference in the syntactic level annotation is that CoBaLD restores more ellipsis cases than E-UD, namely, the elided subjects. As in E-UD, some referential connections are annotated here, too. Currently, there are two CoBaLD-annotated datasets[3] that have an E-UD level as well. Therefore, the parser developed for CoBaLD can also be used to parse E-UD.

Our contributions in this paper are as follows:

- we propose a joint morphosyntactic and semantic parser which can be used to parse both CoBaLD and E-UD, and publish our code and pre-trained models;
- we investigate syntax and semantics mutual influence that affects the quality of automatic annotation;
- we provide a qualitative analysis of semantic annotation errors.

[1] https://github.com/CobaldAnnotation
[2] https://universaldependencies.org/ext-format.html
[3] https://github.com/CobaldAnnotation/CobaldEng CoBaLD Eng for English
https://github.com/CobaldAnnotation/CobaldRus CoBaLD Rus for Russian

2 Experimental Setup

2.1 Architecture

The suggested model is a multi-task sequence tagger based on Joint Morpho-Syntactic Parser proposed in (Anastasyev, 2020). It consists of a Masked Language Model encoder paired with a set of tagging heads. The encoder transforms the sequence of tokens into contextual embeddings, then each head selects the features relevant to its tier and predicts the tags upon the embeddings (Fig. 1).

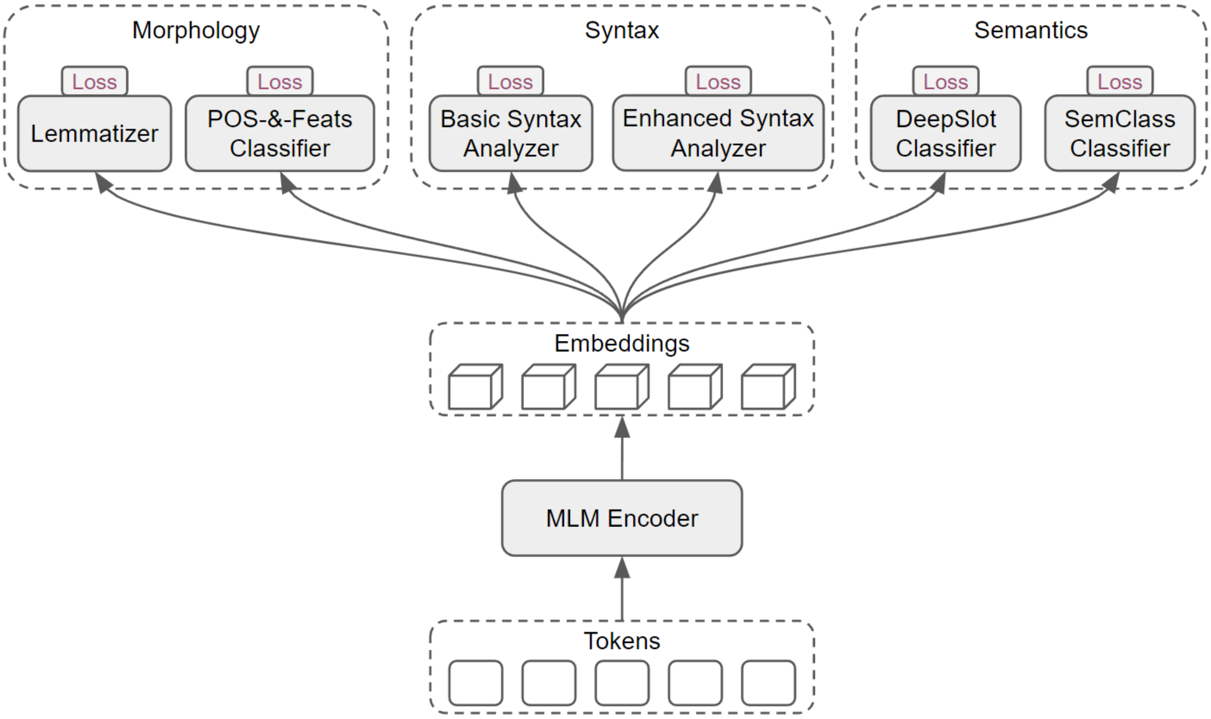


Figure 1: Parser architecture

There are six heads: a lemmatizer, a joint part-of-speech and grammatical features classifier, two syntax analyzers and two semantic classifiers.

Lemmatizer predicts *lemmatization rules* – a sequence of modifications that transform a word into a lemma. The three modifications are available:

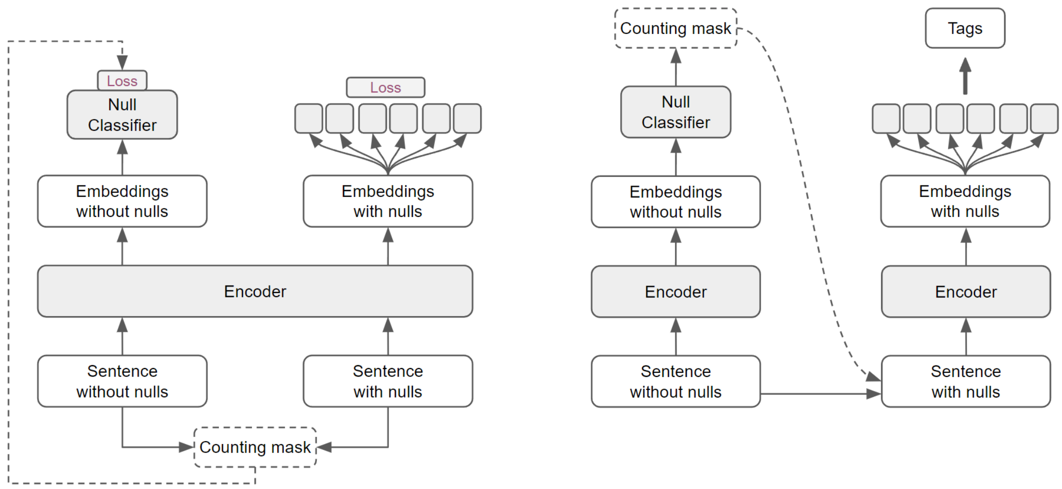
- cut N symbols at the prefix,
- cut N symbols at the suffix,
- append a specific suffix.

For example, in order to obtain *elf* from *elves*, one needs to leave prefix intact, cut suffix of length three and append “f”, so the lemmatization rule for this case is represented as a `cut_pref=0|cut_suff=3|append_suff=f`.

POS-&-Feats head predicts joint part of speech and grammatical features tags. It ensures POS and features consistency, so that nouns never acquire a category of tense and cases never relate to verbs. Since the morphological tags are heavily correlated, the number of combinations remains small and tractable for the classification approach.

Basic syntax analyzer is a biaffine dependency classifier (Dozat and Manning, 2016) that predicts basic syntactic connections between tokens in a sentence. During the training, the arcs are chosen greedily; at the inference, we use Chu-Liu-Edmonds algorithm to find a spanning arborescence of a maximum probability.

As for the **enhanced syntax**, our model is based on the dependency graph parser (He and Choi, 2020). In contrast to the basic analyzer that builds a single distribution over a token’s arcs by minimizing a multi-class cross-entropy loss, the graph parser estimates each arc probability independently using a



(a) Training pipeline. The null classifier is optimized alongside the tagging heads.

(b) Inference pipeline. The parser restores the nulls first (left), then tags the modified sentence (right).

Figure 2: Parser workflow

multi-label approach with a binary cross-entropy loss. This distributional relaxation allows the enhanced analyzer to predict multiple edges per token, accounting for the non-acyclic nature of enhanced syntactic graphs.

However, the dependency graph parser does not address the ellipsis, as the latter involves a more complex task of elided (null) tokens restoration. Up to this date, no attempt has been made to approach this problem in full — for example, at IWPT Shared Task (Bouma et al., 2020), the arcs involving null nodes are collapsed into multi-edges, which allows one to identify the presence of an elided node, but not its position. In the current version of CoBaLD, as opposed to E-UD, elided functional words such as auxiliary verbs or copulas may be restored, so the positions of the elided elements are important. For this reason, we target the original task and actually restore the elided tokens.

To achieve this, we introduce a *counting mask* that shows how many nulls follow a token. Formally, given a gold sentence $(s_1, \dots, s_n)$ with nulls, denote $I = (i \in \{1, \dots, n\} \mid s_i \neq \varnothing)$ as indices of non-null tokens, i.e. the subsequence $(s_i)_{i \in I}$ is a test sentence with null tokens removed. The mask is defined as $L = (I^+_{i+1} - I^+_i - 1)_{i=1}^m$, where $I^+ = I \cup \{n\}$ and $m = |I|$ is a length of the test sentence.

For example, suppose the gold sentence is `[Quick, brown, #NULL, #NULL, fox, #NULL]`. Then the test sentence without nulls would be `[Quick, brown, fox]`, $I = [1, 2, 5]$ and the counting mask $L = [0, 2, 1]$, for there is no null after `Quick`, two nulls after `brown` and one after `fox`.

Given the counting mask, ellipsis restoration becomes as simple as sequence labeling; we add another head that predicts how many nulls should be inserted after a token. It splits the pipeline into two stages: first, the null classifier retrieves the elided tokens and adds them to an input sentence, then the refined sentence is parsed as usual (Fig. 2b). During the training, sentences with and without nulls are passed through the encoder independently, so the tagging heads are always trained upon the well-formed sentences, and not the restored ones (Fig. 2a). It is similar to a teacher forcing approach used in sequence generation, where a ground truth token is fed to a model instead of the previously predicted word to avoid error accumulation and speed up the training.

We jointly optimize the null classifier with the tagging heads by adding up the losses. The encoder aggregates the gradients from all classifiers and updates the weights once per training step.

Note that the counting mask ignores the leading nulls, e.g. `[#NULL, On, that, date]`, as there is no token to follow. To account for this issue, we prepend an auxiliary token to the beginning of a test sentence prior to mask construction.

The **semantic classifiers** predict deep slots and semantic classes. As the semantic categories of CoBaLD are very simple from a technical point of view, the task of semantic parsing represents basic multinomial classification for both DSs and SCs which are predicted independently.

Pretrain dataset	Train dataset	Test quality, %			
		UAS	LAS	EUAS	ELAS
-	CoBaLD Eng	92.3	91.2	86.4	84.6
-	CoBaLD Eng + GUM + EWT	92.5	91.3	87.1	85.1
GUM + EWT	CoBaLD Eng	**93.3**	**91.6**	**89.3**	**87.6**

Table 1: Different training strategies (scores for the joint model)

All the heads (except for syntactic ones) are two linear layers separated by a ReLU activation.

2.2 Experiments

We have experimented with several setups for the parser, changing the data we trained it on, the encoder and the amount of tagging heads in order to analyze how joint morphosyntax and semantics parsing would affect overall quality.

The two datasets we experimented on are referred to as EWT and BBC. The former is a portion of English EWT dataset (Silveira et al., 2014) available on UD official Github[4], with manually annotated semantics level, containing around 80,000 tokens. The latter is the aforementioned CoBaLD Eng dataset which contains BBC news texts and is around 190,000 tokens. We also used a portion of English PUD dataset available on UD Github[5] for testing.

Since the syntax level, especially the enhanced one, is known to be a weak spot of the parsers (judging by the scores, e.g. in (Straka et al., 2016)), we experimented on enriching the training data with syntactic tags of public E-UD treebanks, namely GUM (Zeldes, 2017) and EWT[6], around 200,000 tokens each. We tried two strategies: enlarge the training set with the new data, or pre-train the model on the public data first and then finetune it on the original dataset. The Table 1 shows that the pre-training approach significantly improves the syntactic scores, so we adhere to this strategy in the subsequent experiments.

We have used two pre-trained MLMs as encoders: distilbert-uncased[7] and XLM-RoBERTa Base[8].

The experiments concerning syntax and semantics mutual influence included training of the following models: 1) a joint model; 2) a model with morphosyntax parsing only; 3) a model with semantics parsing only.

3 Results

The results for our experiments are presented in Table 2. The metrics we adopted for the morphology and base UD syntax are commonly used metrics which are described, for instance, in (Lyashevskaya et al., 2020) with slight improvements for lemmatization and grammatical features described in (Petrova et al., 2023).

Since the F_1-based enhanced attachment scores used in IWPT Shared Tasks are described quite loosely, we define enhanced labeled attachment score in a manner similar to a Jaccard coefficient, but use max instead of union to soften the penalty:

$$ELAS(test,\ gold) = \frac{|test_{deps} \cap gold_{deps}|}{max(|test_{deps}|, |gold_{deps}|)},$$

where $token_{deps}$ is a dictionary with keys representing heads and values representing relations, e.g. `{(24, nsubj), (26, nsubj:xsubj)}`. The EUAS is defined similarly, but the relations are ignored. These metrics equal to LAS and UAS in case of tree syntax, which makes base and enhanced tiers easy to compare to each other.

[4]https://github.com/UniversalDependencies/UD_English-EWT
[5]https://github.com/UniversalDependencies/UD_English-PUD
[6]The part of EWT annotated with semantics was excluded
[7]https://huggingface.co/distilbert/distilbert-base-uncased
[8]https://huggingface.co/FacebookAI/xlm-roberta-base

As for the semantics metrics, they are calculated according to the next formulas.

The *deep slot* (DS) score is a simple accuracy:

$$DeepSlot(test,\ gold) = [test_{DS} = gold_{DS}].$$

Semantic classes (SC) are scored based on the semantic hierarchy of CoBaLD standard:

$$SemClass(test,\ gold) = \frac{1}{1 + Distance(test_{SC},\ gold_{SC})}.$$

That is, the closer test and gold semantic classes are in hierarchy, the higher the score is.

Considering the fact that elided tokens are to be predicted, the number of tokens in test and gold sentences may differ. It causes certain problems, because all metrics operate on the token level and expect the latter to be aligned. To account for this issue, we add a special *empty token*, which fills the gaps caused by misplaced nulls. If a gold null is missed, the empty token is inserted into the test sentence; if an extra null is predicted, the empty token is added to the gold one. Thus, the empty token always faces a null in the opposite sentence. If one of the tokens in a test-gold pair is empty, the scores of such pair are zeroed.

Dataset	Model	Type	Lemma	POS	Feats	UAS	LAS	EUAS	ELAS	DeepSlot	SemClass
EWT	Distil	Joint	**93.21**	91.97	**93.79**	88.23	81.06	83.14	75.84	84.38	85.77
		NoSem	92.83	**92.05**	93.70	**88.36**	**81.31**	**83.89**	**76.62**	-	-
		Sem	-	-	-	-	-	-	-	**84.92**	**87.78**
	XLM	Joint	**93.73**	**93.12**	**93.98**	**89.21**	**82.20**	80.13	73.37	**84.85**	86.95
		NoSem	93.38	93.02	93.95	89.12	82.11	**81.39**	**74.67**	-	-
		Sem	-	-	-	-	-	-	-	83.80	**88.30**
BBC	Distil	Joint	**97.10**	**97.90**	**98.52**	93.41	91.84	89.77	88.07	89.63	92.06
		NoSem	97.06	97.76	98.47	**93.47**	**91.88**	**90.41**	**88.74**	-	-
		Sem	-	-	-	-	-	-	-	**90.00**	**93.04**
	XLM	Joint	97.23	**98.51**	**98.55**	92.93	91.10	84.84	82.83	88.71	91.93
		NoSem	97.23	98.45	98.47	**93.16**	**91.28**	**85.08**	**83.03**	-	-
		Sem	-	-	-	-	-	-	-	**88.79**	**92.83**
BBC + EWT	Distil	Joint	**97.15**	**97.75**	**98.41**	93.02	91.28	89.18	87.31	89.59	92.32
		NoSem	97.12	97.60	98.29	**93.30**	**91.54**	**90.05**	**88.16**	-	-
		Sem	-	-	-	-	-	-	-	**90.20**	**93.02**
	XLM	Joint	**97.25**	98.13	98.19	**92.83**	90.59	84.89	82.46	88.92	92.20
		NoSem	97.08	**98.22**	**98.23**	92.72	**90.64**	**84.94**	**82.62**	-	-
		Sem	-	-	-	-	-	-	-	**89.23**	**92.95**

Table 2: Model scores for various setups with different training data, backbone LM and tuned tagging heads. Joint stands for a joint morphosyntactic and semantic model, NoSem – a model with morphosyntax only, Sem – a model with semantics only

3.1 Morphosyntax and Semantics

Our results appeared quite stable and a little bit surprising as, first, a more complex XLM-RoBERTa Base (XLM in Table 2) steadily showed worse scores than distilbert-uncased, second, the quality of Enhanced UD syntax and semantics decreased a little for joint versions. Slight drops in quality for the models trained on combined datasets may be caused by different data distribution (EWT contains colloquial texts while BBC is comprised of news only). Also intriguing is the fact that base UD syntax prediction quality didn't suffer much (differences in scores are quite insubstantial).

A qualitative error analysis showed that there is no obvious connection between semantic and syntax levels concerning errors, but a joint model has a tendency of predicting higher probabilities to plausible arcs which sometimes leads to better results compared with the syntax-only model, whose probabilities are too low to overcome the threshold causing it to predict no arcs at all. As for the joint model, it ends up predicting more arcs than it should (so generally there are about 1% more arcs predicted by a joint model than by a syntax-only model). It must be noted that incorrectly predicted syntax heads are quite reasonable and usually do have both semantic and syntactic relationship, e.g., for a sentence 'Based on the Hellblazer comics, the film took $ 11.8 m (£ 6.1 m) on its second week of release' token *comics* is governed by *based*, which in turn has *took* as its head. XLM-R joint model predicts both *based* and *took* as heads with the same dependency type (which is correctly predicted as *obl:on*).

Unfortunately, we also found out that the private test dataset we used for validation purposes contained errors in syntactic annotation, but this fact does not seem to influence the quality drops much, as we also tested syntax prediction quality on English PUD (see Table 3), and it showed the same tendency, though overall quality worsened, most probably due to differences in data distribution and annotation principles (note that E-UD annotation level may be quite different for various E-UD datasets as there are no detailed manuals for it).

Model	Type	EUAS	ELAS
Distil	Joint	83.68	78.40
	NoSem	**83.87**	**78.66**
XLM	Joint	81.02	75.50
	NoSem	**81.33**	**76.06**

Table 3: English PUD testing scores for E-UD annotation level

Another interesting fact is that although the XLM-R-based models consistently show worse results than the distilbert-based ones, their performance drops for E-UD syntax are less, moreover, they are diminished when the training data amount (and diversity) increases. This tendency for XLM-R is less pronounced with semantics, though. Nevertheless, there may be another factor we should take into account, namely, there are some discrepancies in semantics and syntax level we are aware of. Most notable is the difference in copula annotation: as the semantics part of CoBaLD standard was inherited from another annotation model (for details see (Ivoylova et al., 2023)), copula is considered the head of its complement, while in UD, it is the dependent node. However, deep slots in CoBaLD are annotated according to the former principle, so that it may be confusing for the parser. There are also some minor discrepancies of this kind which we plan to obliterate, so the drops in semantics parsing quality may disappear.

In any case, it should be said that overall quality of the parser is quite high and the performance drops for joint versions that we have discussed are tolerable. Therefore, morphosyntax and semantics can be parsed simultaneously at almost no cost in quality, and there is a chance that if we increase the amount of training data and improve the compatibility of the syntactic and semantic annotation tiers, the quality of joint parsing will even grow compared to separate syntax and semantics prediction.

3.2 Semantic errors of CoBaLD parser

The semantic errors of the parser concern the definition of the word's SC, the DS which binds the dependent node with its parent node, or the combination of the items.

The scores for DSs and SCs are given in Table 2.

The mistakes in defining the SCs deal mostly with the following cases:

- the parser suggests a SC of a homonym instead of the proper class: for instance, the SCs BEING vs CH_REFERENCE_AND_QUANTIFICATION for the word 'couple' in *The couple now have four children together* ('couple' meaning 'a pair of people being together' vs 'an indefinite small number' as in *just a couple more questions*);

- the parser suggests a hyperonym with a more general meaning - e.g. ORGANIZATION instead of MILITARY_FORCES_AS_ORGANIZATION;
- the parser defines a SC of an unknown word (usually, proper name) or a pronoun other than the human annotator: as the SCs ORGANIZATION vs ARRANGEMENTS for ‘CES’ in *CES showcases 50,000 new gadgets that will be hitting the shelves in 2005*.

The mistakes dealing with the DSs can be divided in two groups: first, when the SC of the head token is correct but the DS of its dependent is improper (≈56% of all DS errors), and, second, when both the SC of the head (or the head itself) and the DS of its dependent are improper (≈44% respectively).

The first case can be illustrated by the example: *In my book this is cheating on the club, the supporters, the manager and his own team-mates.* Here the human annotator labelled ‘my’ as Possessor, while the parser marked it as Agent (as if ‘I am the author of the book’).

An example of the second case can be the sentence: *A number of fans questioned Gerrard’s commitment and sarcastically branded his own goal in Liverpool’s 3-2 defeat as his first goal for Chelsea.* The parser defines ‘number’ as Agent DS of the verb ‘question’, while the human annotator suggests the Experiencer DS here. The reason is that the parser and the annotator chose different SCs for ‘question’ as well: ‘question’ as VERBAL_COMMUNICATION (‘to ask a question’) which demands Agent for its subject and ‘question’ as STATE_OF_MIND (‘feel or express doubt about’) which demands the Experiencer subject.

Such errors seem natural as they mainly deal not with the parser’s work itself, but with the semantic ambiguity.

Nevertheless, there are errors as well, where the parser suggests the annotation which is semantically impossible in terms of the given model, for instance, marks the subject of some active verb as Experiencer instead of Agent.

This problem mainly concerns actant slots (Agent, Object, Experiencer, and alike) – that is, slots with similar filling and syntactic realizations which differentiate through their semantic relations with their cores only. Circumstantial adjuncts (like locative, temporal, conditional, or concession adjuncts) are usually defined properly as their surface realizations, semantic filling and sense does not depend on the cores they are governed by. The number of such errors should reduce with enlarging the volume of the annotated data.

4 Related Work

There have been two shared tasks dedicated to the parsing of E-UD held on the International Conference on Parsing Technologies in 2020 (Bouma et al., 2020) and 2021 (Oepen et al., 2021). The winner for the former task is TurkuNLP (Kanerva et al., 2020), and for the latter one – TGIF (Shi and Lee, 2021).

There exists a hypothesis that joint parsing of several language levels should help to improve the quality for all of them. As for morphology and syntax, it has become a standard to parse them jointly. Concerning joint syntax and semantics parsing, some of the earliest attempts to build the complete formal semantic representations along with the syntactic ones automatically were made almost 20 years ago (Ge and Mooney, 2005), and the attempts to solve the task of Semantic Role Labeling simultaneously with syntactic dependency parsing even earlier than that, e.g., (Gildea and Jurafsky, 2002).

Normally, the task of syntactic parsing is regarded as a way to improve the quality of semantic parsing (Li et al., 2020), although modern research shows that semantic parsing can achieve impressive results without the support of syntax. Thus, syntax and semantic parsing can be now viewed as separate tasks for multi-task learning. There have been researches dedicated to such simultaneous parsing in recent years as well, e.g. (Zhou et al., 2019), where the authors attempt parsing constituents and syntax dependencies together with the Semantic Role Labeling. As shown in the mentioned paper, syntax dependencies don’t improve semantics, while the constituents affect everything positively. The other notable work (Stengel-Eskin et al., 2021) concerns the aforementioned Universal Decompositional Semantics; the authors report that syntax dependencies slightly improve the semantic level parsing.

5 Conclusion

In this paper, we present a joint morphosyntactic and semantic parser for CoBaLD annotation standard which can be applied to Enhanced UD parsing as well and is capable of ellipsis restoration. Its scores across all linguistic levels show the quality comparable with that of SOTA solutions for E-UD and UD parsing. The code for the parser is available on Github[9] and the trained models are published on Huggingface.co[10]. We also provide a qualitative analysis of the parser's semantic errors and analyze the possible reasons of the slight performance drops for the joint syntax and semantics parsing. Our future goals include further investigation of the syntax and semantics mutual influence, as well as the improvements to the architecture and the annotation standard itself.

References

Omri Abend and Ari Rappoport. 2013. Universal conceptual cognitive annotation (ucca). // *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, P 228–238.

DG Anastasyev. 2020. Exploring pretrained models for joint morpho-syntactic parsing of russian. // *Computational Linguistics and Intellectual Technologies*, P 1–12.

Jurij D Apresian, Igor M Boguslavsky, Leonid L Iomdin, Alexander V Lazursky, Vladimir Z Sannikov, Victor G Sizov, and Leonid L Tsinman. 2003. Etap-3 linguistic processor: a full-fledged nlp implementation of the meaning ótext theory1. *Institute for Information Transmission Problems, Russian Academy of Sciences*.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. // *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, P 178–186.

Gosse Bouma, Yuji Matsumoto, Stephan Oepen, Kenji Sagae, Djamé Seddah, Weiwei Sun, Anders Søgaard, Reut Tsarfaty, and Daniel Zeman. 2020. Proceedings of the 16th international conference on parsing technologies and the iwpt 2020 shared task on parsing into enhanced universal dependencies. // *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*.

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

Timothy Dozat and Christopher D. Manning. 2016. Deep biaffine attention for neural dependency parsing. *CoRR*, abs/1611.01734.

Ruifang Ge and Raymond Mooney. 2005. A statistical semantic parser that integrates syntax and semantics. // *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, P 9–16.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational linguistics*, 28(3):245–288.

Han He and Jinho D. Choi. 2020. Adaptation of multilingual transformer encoder for robust enhanced Universal Dependency parsing. // Gosse Bouma, Yuji Matsumoto, Stephan Oepen, Kenji Sagae, Djamé Seddah, Weiwei Sun, Anders Søgaard, Reut Tsarfaty, and Dan Zeman, *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, P 181–191, Online, July. Association for Computational Linguistics.

Alexandra Ivoylova, Darya Dyachkova, Maria Petrova, and Mariia Michurina. 2023. The problem of linguistic markup conversion: the transformation of the compreno markup into the ud format. // *International Conference on Computational Linguistics and Intellectual Technologies «Dialog*.

Jenna Kanerva, Filip Ginter, and Sampo Pyysalo. 2020. Turku enhanced parser pipeline: From raw text to enhanced graphs in the iwpt 2020 shared task. // *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, P 162–173.

Zuchao Li, Hai Zhao, Shexia He, and Jiaxun Cai. 2020. Syntax role for neural semantic role labeling. *arXiv preprint arXiv:2009.05737*.

[9]https://github.com/CobaldAnnotation/CobaldParser
[10]https://huggingface.co/CoBaLD

ON Lyashevskaya, TO Shavrina, IV Trofimov, NA Vlasova, et al. 2020. Grameval 2020 shared task: Russian full morphology and universal dependencies parsing. // *Proc. of the International Conference Dialogue*, volume 2020, P 553–569.

Marie Mikulová, Alevtina Bémová, Jan Hajič, Eva Hajičová, Jiří Havelka, Veronika Kolářová, Lucie Kučová, Markéta Lopatková, Petr Pajas, Jarmila Panevová, et al. 2006. Annotation on the tectogrammatical level in the prague dependency treebank. *annotation manual. Technical Report*, 30:5–11.

Stephan Oepen, Kenji Sagae, Reut Tsarfaty, Gosse Bouma, Djamé Seddah, and Daniel Zeman. 2021. Proceedings of the 17th international conference on parsing technologies and the iwpt 2021 shared task on parsing into enhanced universal dependencies (iwpt 2021). // *Proceedings of the 17th International Conference on Parsing Technologies and the IWPT 2021 Shared Task on Parsing into Enhanced Universal Dependencies (IWPT 2021)*.

Maria Petrova, Alexandra Ivoylova, Ilya Bayuk, Darya Dyachkova, and Mariia Michurina. 2023. The cobald annotation project: the creation and application of the full morpho-syntactic and semantic markup standard. // *Proceedings of the International Conference "Dialogue*, volume 2023.

Maria Andreevna Petrova, Alexandra M Ivoylova, and Anastasia Tishchenkova. 2024. Cobald annotation: The enrichment of the enhanced universal dependencies with the semantical pattern. // *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, P 3422–3432.

Sebastian Schuster and Christopher D Manning. 2016. Enhanced english universal dependencies: An improved representation for natural language understanding tasks. // *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, P 2371–2378.

Tianze Shi and Lillian Lee. 2021. Tgif: Tree-graph integrated-format parser for enhanced ud with two-stage generic-to individual-language finetuning. *arXiv preprint arXiv:2107.06907*.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Christopher D. Manning. 2014. A gold standard dependency corpus for English. // *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*.

Elias Stengel-Eskin, Kenton Murray, Sheng Zhang, Aaron Steven White, and Benjamin Van Durme. 2021. Joint universal syntactic and semantic parsing. *Transactions of the Association for Computational Linguistics*, 9:756–773.

Milan Straka, Jan Hajic, and Jana Straková. 2016. Udpipe: trainable pipeline for processing conll-u files performing tokenization, morphological analysis, pos tagging and parsing. // *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, P 4290–4297.

Aaron Steven White, Drew Reisinger, Keisuke Sakaguchi, Tim Vieira, Sheng Zhang, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. 2016. Universal decompositional semantics on universal dependencies. // *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, P 1713–1723.

Amir Zeldes. 2017. The GUM corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.

Junru Zhou, Zuchao Li, and Hai Zhao. 2019. Parsing all: Syntax and semantics, dependencies and spans. *arXiv preprint arXiv:1908.11522*.