Computational Linguistics and Intellectual Technologies:
Proceedings of the International Conference “Dialogue 2023”

June 14–16, 2023

The CoBaLD Annotation Project: the Creation and Application of the full Morpho-Syntactic and Semantic Markup Standard

Maria Petrova
A4 Technology
Moscow
g-fox-ive@mail.ru

Alexandra Ivoylova
RSUH
Moscow
a.m.ivoylova@gmail.com

Ilya Bayuk
A4 Technology
Moscow
ilya.bayuk@yandex.ru

Darya Dyachkova
RSUH
Moscow
d.dyachkova@bk.ru

Mariia Michurina
RSUH
Moscow
marimitchurina@gmail.com

Abstract

The current paper is devoted to the Compreno-Based Linguistic Data (CoBaLD) Annotation Project aimed at creating text corpora annotated with full morphological, syntactic and semantic markup. The first task of the project is to suggest a standard for the full universal markup which would include both morphosyntactic and semantic patterns. To solve this problem, one needs the markup model, which includes all necessary markup levels and presents the markup in a format convenient for users. The latter implies not only the fullness of the markup, but also its structural simplicity and homogeneity. As a base for the markup, we have chosen the simplified version of the Compreno model[1], and as data presentation format, we have taken Universal Dependencies.

At the second stage of the project, the Russian corpus with 400 thousand tokens (CoBaLD-Rus) has been created, which is annotated according to the given standard. The third stage is devoted to the testing of the new format. For this purpose, we have held the SEMarkup Shared Task aimed at creating parsers which would produce full morpho-syntactic and semantic markup. Within this task, we have elaborated neural network-based parser trained on our dataset, which allows one to annotate new texts with the CoBaLD-standard. Our further plans are to create fully annotated corpora for other languages and to carry out the experiments on language transfers of the current markup to other languages.

Keywords: Compreno, semantic markup, Universal Dependencies
DOI: 10.28995/2075-7182-2023-22-421-432

Проект CoBaLD: разработка и применение стандарта полной морфо-синтаксической и семантической разметки текстов

Петрова М.А.
A4 Technology
Москва, Россия
g-fox-ive@mail.ru

Ивойлова А.М.
РГГУ
Москва, Россия
a.m.ivoylova@gmail.com

Баюк И.С.
A4 Technology
Москва, Россия
ilya.bayuk@yandex.ru

Дьячкова Д.С.
РГГУ
Москва, Россия
d.dyachkova@bk.ru

Мичурина М.А.
РГГУ
Москва, Россия
marimitchurina@gmail.com

Аннотация

Данная работа посвящена проекту Compreno-Based Linguistic Data (CoBaLD), целью которого является создание корпусов с полной морфологической, синтаксической и семантической разметкой. Первой задачей проекта является создание стандарта полной универсальной разметки, включающей как морфо-синтаксический, так и семантический уровни. Реализация данной задачи требует, с одной стороны, наличия модели, предлагающей необходимые уровни разметки,

[1]The access to the Compreno data is provided according to the CC BY-NC 4.0 License which allows non-commercial use.

и, с другой стороны, возможности представить разметку в удобном для пользователя формате. Последнее требование предполагает не только полноту разметки, но также ее структурную простоту и однородность описания объектов. В качестве базы для подобной разметки мы выбрали упрощенную модель Compreno, в качестве формата представления данных - формат Universal Dependencies.

Вторым этапом проекта стало создание русскоязычного корпуса объемом 400 тысяч токенов - CoBaLD-Rus, размеченного по предложенному стандарту. Третий этап посвящен тестированию предложенной разметки, в рамках которого было проведено соревнование SEMarkup Shared Task. Задача состояла в создании парсеров, обученных на данном корпусе и позволяющих размечать новые тексты в соответствии с CoBaLD-стандартом. В качестве бейзлайна для соревнования мы также разработали нейросетевой парсер для решения поставленной задачи. В дальнейшем планируется создание аналогичных корпусов для других языков и проведение экспериментов по языковому переносу данной разметки на другие языки.

Ключевые слова: Compreno, семантическая разметка, Universal Dependencies

1 Introduction

In the given paper, we present the Compreno-Based Linguistic Data (CoBaLD) Annotation Project which is aimed at elaborating the general standard of the full text markup, including morphological, syntactic and semantic levels, and the creation of text corpora annotated according to the standard. The current work focuses on the following tasks:

(1) choosing the markup model, which is full enough and at the same time simple enough to be presented in the convenient format;

(2) choosing the format of the full markup presentation;

(3) elaborating the markup standard, including both morphosyntactic and semantic markup;

(4) creating the Russian corpus annotated according to the standard;

(5) conducting a shared task aimed at the creation of the automatic semantic markup in order to investigate the capabilities of the format (SEMarkup-2023 Shared Task);

(6) creating a baseline version of the parser trained on the annotated dataset which allows one to annotate new texts in the CoBaLD-standard.

Since the task of the linguistic markup is an important part of the NLP pipeline, a lot of efforts have been applied to create convenient markup formats.

As far as the formats of the morpho-syntactic markup are concerned, the most popular one is the Universal Dependencies (UD) project (De Marneffe et al., 2006). There are parsers created for the UD standard, such as UDPipe (Straka et al., 2016) (currently, for more than 100 languages including Russian), and, for the Russian language, - the Joint Morpho-Syntactic Parser (Anastasyev, 2020).

Concerning the semantic markup, there are several projects, most of which started with creating a machine translation algorithm. One of the oldest projects is the Universal Networking Language (Uchida and Zhu, 2001), which popularized the idea of using directed graphs for semantic descriptions. Among other well-known projects are Abstract Meaning Representations (AMR) (Banarescu et al., 2013), Universal Conceptual Cognitive Annotation (UCCA) (Abend and Rappoport, 2013), Prague Dependencies (Hajic et al., 2001), Discourse Representation Structures (DRS) (Groenendijk et al., 1984), and Universal Decompositional Semantics (UDS) (White et al., 2016). The Russian National Corpus (ruscorpora.ru) has recently included partial semantic markup, too.

These formats have significant differences with regard to their treatment of morphosyntax and semantics. For instance, UCCA and AMR ignore morphosyntactic data on purpose, while the Prague dependencies represent full three-level linguistic markup. The ETAP system (Boguslavsky, 1999) and the Compreno model (Anisimovich et al., 2012; Petrova, 2014) propose such integral labelling as well. Moreover, UDS, if joined with UD, could represent its semantic part. However, all these formats are rather complicated and difficult to work with. Therefore, there is no generally accepted standard up to now both for the semantic markup and for the full markup, which would include all three markup levels.

Thereby, our first purpose is to develop a standard, which would, on the one hand, include morphological, syntactic and semantic markup, and, on the other hand, be simple and convenient enough for the users to work with.

As for the format, we have chosen the UD model: it is concise enough and uses the CONLL (or CONLL Plus) format, which makes it convenient for scripting and automatic parsing purposes. However, UD lacks a semantical pattern. Therefore, we had to integrate it from some other model.

We have chosen the Compreno one, as the model is very simple from the structural point of view and suggests full semantic markup both for the word meanings and the relations between words.

Further, we will briefly describe the basic principles of the Compreno model and show the conversion process of the Compreno markup into the UD format. Afterwards, we will present our dataset annotated according to the CoBaLD-standard and focus on the SEMarkup-2023 Shared Task together with a baseline parser created for it.

In conclusion, we will sum up the results and discuss further perspectives of the work.

2 The Compreno Markup Format: Simplification and Conversion

2.1 Simplification of the Compreno Format

In Compreno, each word meaning is attributed to a semantic class (SC) - a semantic field denoting the word's meaning. The SCs are organized in a thesaurus-like hierarchy. All semantic links between words are expressed through the semantic roles, or slots (SS) corresponding to actant valencies (Agent, Experiencer, Addressee, etc.), adjuncts (Locative, Distance, Time, Condition, Concession, and so on), characteristics (for instance, evaluation, speed, price, form, or size), specifications and others. It allows one to annotate the semantical meanings of all words and to define all semantic relations of each word, both actant and circumstantial.

However, the model suggests a heavily detailed description: namely, it contains more than 200,000 SCs (which seems too much for a machine learning based parser trained on the dataset of our volume) and more than 330 SSs, which, in turn, does not seem necessary for most application tasks (except the task of building semantic sketches (Detkova et al., 2020)).

Therefore, we decided to reduce the number of categories. First, we have used not the terminal SCs, which denote the exact word meanings, but the hyperonym classes. That is, all words with motion semantics would now belong to the hyperonym class MOTION. Second, we have reduced the number of the SSs. For example, full Compreno markup suggests different roles for different characteristic dependencies, such as form, taste, sound, appearance, importance, genuineness, and so on - more than 60 characteristics in total. In the generalized variant, all such characteristics correspond to one characteristical slot. Or, full model contains several Instrument slots, which differ by the SCs each slot can include (see fig.1) - in the simplified variant, they are joined in one Instrument slot.

Instrument		to write [with a pen]
Instrument_Being	Instrument	to attack [with remaining army]
Instrument_Cognitive		to understand [with one's heart]
Instrument_Time		to be punished [by 30 years in prison]

Figure 1: Instrument slots in full and in reduced Compreno markup

As a result, the number of hyperonym SCs used in the markup was reduced to 1085 classes, and the number of the SSs - to 143 slots.

The semantic hierarchy of the hyperonym classes can be found on the Compreno Semantics Github[2]. The list and the description of the semantic roles are also available on the corpus page[3].

These simplified SCs and SSs are used in the final version of the markup in the UD format.

2.2 Annotation and Conversion

The Compreno markup can be obtained automatically or manually. For the current dataset, the markup includes the boundaries of the constituents, the SCs (their labels are marked with green below) and the

[2]https://github.com/compreno-semantics/semantic-hierarchy/blob/main/hyperonims_hierarchy.csv

[3]https://github.com/compreno-semantics/compreno-corpus/blob/main/semantic_slots.xlsx

SSs (their labels are marked with brown below) - see fig. 2.

Обычно бюджет ко второму чтению готовится непосредственно в Думе: депутаты корректируют правительственные планы. 'Usually the budget is prepared for the second reading directly in the Duma: the deputies update the government plans'.

```
#[[Time: Обычно"обычно:#frequentative_adverbs_adj:FREQUENTATIVE"] [Experiencer_Metaphoric:
бюджет"бюджет:бюджет:BUDGET"] [[ко"к:#preposition:PREPOSITION"] [OrderInTimeAndSpace:
второму"второй:TWO_ORDINAL"] Object_Situation: чтению "чтение:READING_OF_THE_DRAFT_LAW"] Predicate:
готовится"готовить:готовить:PREPAREDNESS" [[DegreeApproximative:
непосредственно"непосредственный:DIRECT_OBLIQUE"] [в"в_Prepositional:#preposition:PREPOSITION"] Locative:
Думе"дума:дума:DUMA"]#: [[Agent: депутаты"депутат:депутат:DEPUTY"] Specification_Clause:
корректируют"корректировать:корректировать:TO_CORRECT" [[Agent:
правительственные"правительство:правительство:GOVERNMENT"] Object_Situation:
планы"план:план:SCHEDULE_FOR_ACTIVITY"]]].
```

Figure 2: An example of the Compreno "bracket" format

The markup can also be provided with surface, or syntactic, roles (marked with $ sign - see Fig. 3 below), coreference and non-tree links, however, the purpose of the given dataset was only the semantic markup. [4]

As one can see, this format of markup representation does not contain morphological and other grammatical information. Nevertheless, after a sentence is annotated, the parser can build its parsing tree (see (Anisimovich et al., 2012)), where each token is provided with full grammatical and semantic data. Fig. 3 is an illustration of the Compreno parsing tree for the above given sentence, and fig. 4 being the fragment of the tree shows the morphological grammemes for the node "готовить:готовить:PREPAREDNESS".

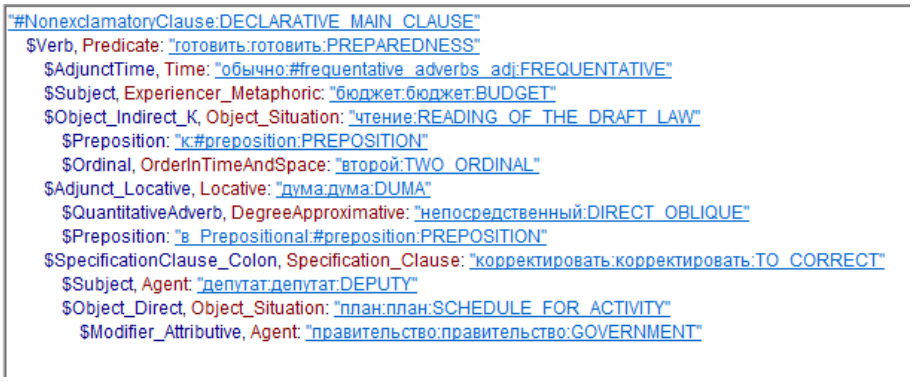

Figure 3: An example of the Compreno parse tree

The "bracket" format presented on fig. 2 is the one that the annotators work with to point out the information necessary for building the correct structure of a sentence, whereas the parsing tree is where full information about the sentence is stored (its syntactic and semantical structure, syntactic and semantic slots, SCs, grammatical features and information about coreference and non-tree links).

Unlike Compreno, UD stores all relevant information in the markup itself, presented in a table-view. Thereby, during the conversion of the Compreno markup into UD, the necessary data is taken from the parsing trees.

UD has its own morphology and syntax, therefore, the corresponding information in Compreno has to be converted into the UD format. Of course, there is a number of differences between the Compreno and the UD formats in this respect. Most significant distinctions concern POS-tagging, tokenization, lemmatization, asymmetry of mapping some grammatical features, ellipsis and copula description, coordination and dealing with punctuation. Besides, the UD format marks the tokens up with so called dependency

[4]The only surface slot mentioned in the markup is the $Dislocation slot – it is the slot for the constituents that syntactically depend on one core, while semantically – on the other core.

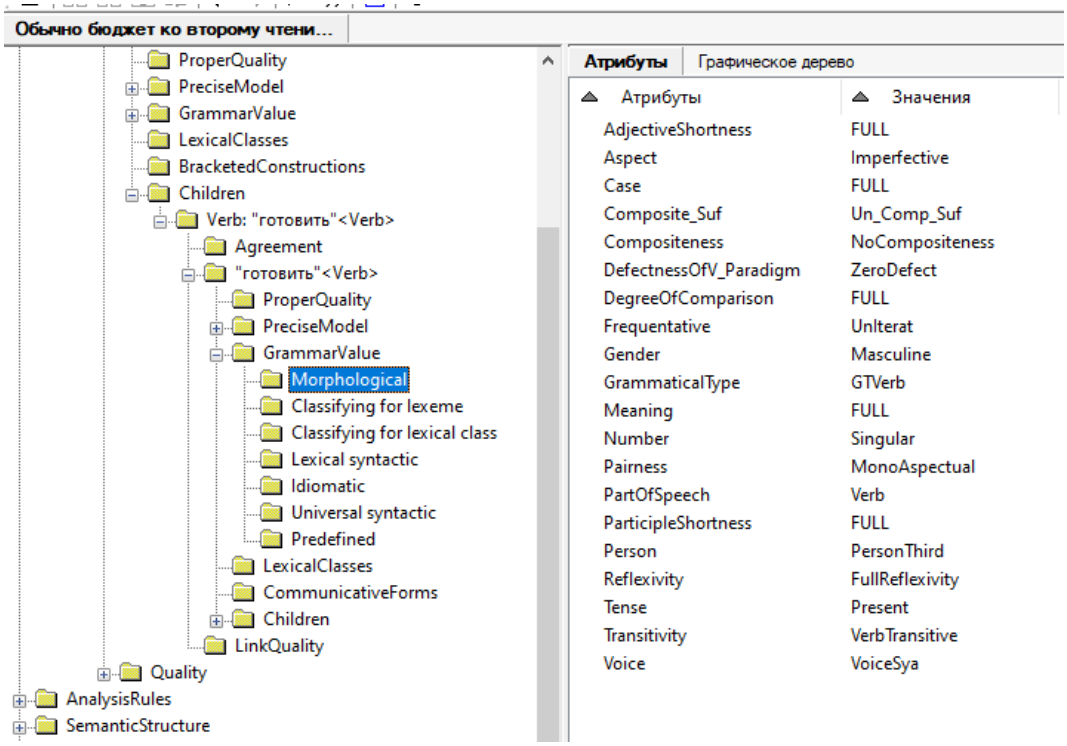

Figure 4: The grammemes for the node "готовить:готовить:PREPAREDNESS"

heads (each token gets the index of its head as a label) whereas the Compreno model operates with the boundaries of the constituents. During the conversion, the labeling of these heads was based on their boundaries. The conversion process is thoroughly discussed in (Ivoylova et al., 2023).

As far as the semantics is concerned, the UD format does not have the semantic level, so the information about the SCs and the SSs can be added to the UD markup in the way it is presented in Compreno (its simplified version).

After the conversion, the markup looks as in fig. 5 and includes morphology, syntax, and semantics.

```
# text = Обычно бюджет ко второму чтению готовится непосредственно в Думе : депутаты корректируют правительственные планы.
1   Обычно  обычно  ADV _   Degree=Pos  6   advmod  Time    CH_REFERENCE_AND_QUANTIFICATION
2   бюджет  бюджет  NOUN  _   Animacy=Inan|Case=Nom|Gender=Masc|Number=Sing   6   nsubj   Experiencer_Metaphoric  BUDGET
3   ко  ко  ADP _   _   5   case    _   PREPOSITION
4   второму второй  ADJ _   Case=Dat|Gender=Neut|Number=Sing    5   amod    OrderInTimeAndSpace
CH_REFERENCE_AND_QUANTIFICATION
5   чтению  чтение  NOUN  _   Animacy=Inan|Case=Dat|Gender=Neut|Number=Sing   6   obl Object_Situation
TO_PROCESS_INFORMATION
6   готовится   готовить    VERB  _   Aspect=Imp|Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin|Voice=Mid  0
root    Predicate   READINESS
7   непосредственно непосредственно ADV _   Degree=Pos  9   advmod  Degree  CH_OF_CONNECTIONS
8   в   в   ADP _   _   9   case    _   PREPOSITION
9   Думе    дума    NOUN  _   Animacy=Inan|Case=Loc|Gender=Fem|Number=Sing    6   obl Locative    STATE_AUTHORITIES
10  :   :   PUNCT   _   _   12  punct   _   _
11  депутаты    депутат NOUN  _   Animacy=Anim|Case=Nom|Gender=Masc|Number=Plur   12  nsubj   Agent
PERSON_BY_SPHERE_OF_ACTIVITY
12  корректируют    корректировать  VERB  _   Aspect=Imp|Mood=Ind|Number=Plur|Person=3|Tense=Pres|VerbForm=Fin|Voice=Act
6   parataxis   Specification   TO_CORRECT
13  правительственные   правительственный   ADJ _   Animacy=Inan|Case=Acc|Degree=Pos|Number=Plur    14  amod    Agent
STATE_AUTHORITIES
14  планы   план    NOUN  _   Animacy=Inan|Case=Acc|Gender=Masc|Number=Plur   12  obj Object_Situation
SCHEDULE_FOR_ACTIVITY
15  .   .   PUNCT   _   _   6   punct   _   _
```

Figure 5: CoBaLD format example

3 Corpus Dataset

Our further goal was to obtain the Russian corpus annotated according to the CoBaLD standard.

For the corpus material, we have chosen news texts from the NewsRu.Com dataset, created during building the RuCoCo corpus (Dobrovolskii et al., 2022). The dataset contains 3 markup levels:

- morphological,
- syntactic,
- semantic.

We have labeled the CoBaLD-Rus dataset - a Compreno-Based Dataset of Russian. It is published on Github[5].

The volume of the corpus is around 400,000 tokens. As our next step is building the parser, the whole sample is divided into two parts:

360 000 training and validation sample,

40 000 test sample for quality evaluation.

The test data does not contain any categories which are not represented in the training data.

4 The Annotation of the Corpus

The annotation process was organized as follows. At the first stage, the corpus was automatically annotated with the Compreno semantic markup with the help of the Compreno parser and included the constituents boundaries, SCs and SSs. Afterwards, the markup and the correctness of the parsing trees were manually checked by a team of professional linguists.

The annotated corpus was converted into the UD format with the help of the Compreno-To-UD Converter presented in (Ivoylova et al., 2023). Finally, the simplification algorithm was applied, which changed the SCs and the SSs to their simplified correlates.

As the morphosyntactic part was converted automatically, about 10% of the conversion results were also human-checked. The percent of modified labels varies from 5 to 10%, which means that the total quality of the conversion is close to 95%.

To measure the ambiguity level of the markup, an experiment on the annotators' agreement has been carried out. 100 sentences have been annotated by two annotators independently. Afterwards, the comparison of the markups has been made, especially as far as the constituents borders, the SCs and the SSs are concerned. The results turned out to be as in the table 1:

	Heads diff.	SemSlots diff.	SemClasses diff.	Overall inter-annotator agreement
Original	0.93%	2.64%	2.72%	**93.71%**
Generalized	0.93%	2.49%	2.41%	**94.17%**

Table 1: Inter-annotator agreement

Most cases of disagreement between the annotators concern polysemy, that is, these are cases, where the sentence can be interpreted differently. For example:

Отметим, контактные линзы для собак и кошек с 2001 года продаются в Японии.
Token: Отметим
SemClass: TO_PERCEIVE / VERBAL_COMMUNICATION

Выявленный дефект во всех машинах будет устранен бесплатно.
Token: машинах
SemClass: APPARATUS / TRANSPORT

As one can see from Table 1, the generalized markup causes less disagreements, because in some cases it does not differentiate between the homonyms with closer semantics.

[5]https://github.com/compreno-semantics/compreno-corpus

5 SEMarkup-2023 Shared Task

To test the created markup format, we suggested the SEMarkup shared task - the task devoted to the creation of the automatic semantic markup. It presupposed creating a solution that would produce a simultaneous morpho-, syntactic and semantic markup. The competition was held on the CodaLab platform[6] and proposed to use the CoBaLD-Rus dataset for learning data. As a baseline, we created a neural networks based parser trained on the CoBaLD-Rus dataset, which allows one to annotate new texts with the CoBaLD standard.

Unfortunately, only one participant succeeded to present the final solution, however, both the baseline and the participant's model demonstrated promising results (see Table 2). Below, we discuss the baseline, the participant's model and our further experiments with the baseline solution.

	Total	Lemma	POS	Features	UAS	LAS	SemSlot	SemClass
baseline	92.2%	96.1%	98.2%	95.3%	90.0%	85.6%	87.8%	92.2%
postoevie	90.2%	94.2%	97.9%	94.5%	86.2%	81.1%	86.9%	90.3%

Table 2: Baseline and participant scores

5.1 Baseline

The baseline model for the competition is a multi-task tagger. It is based on Anastasyev's Joint Morpho-Syntactic Parser (Anastasyev, 2020) (a GramEval2020 winner) extended with semantic tags, and its structure is represented in the fig. 6.

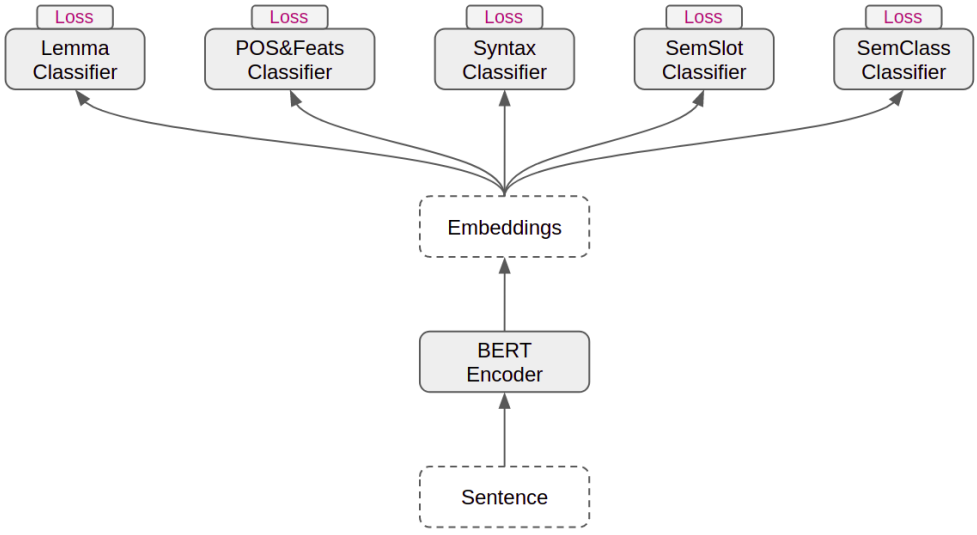

Figure 6: Baseline Architecture

As CoBaLD-Rus consists of multiple tags, the model itself has multiple heads.

Lemma classifier is a nonlinear feed-forward classifier predicting lemmatization rules. Lemmatization rule is a set of modification rules that have to be applied to a word to obtain its lemma. In our case, those are: "cut N symbols from the prefix of the word", "cut N symbols from the suffix" and "append a specific sequence of symbols to the suffix"[7].

POS & Feats classifier is a feed-forward classifier predicting joint POS and grammatical features tags[8].

[6] https://codalab.lisn.upsaclay.fr/competitions/10471
[7] See (Anastasyev, 2020) for details.
[8] See (Anastasyev, 2020) for details.

Syntax classifier is a biaffine dependency classifier (Dozat and Manning, 2016) predicting syntactic head and relation tags.

Semslot classifier and *Semclass classifier* are another two nonlinear feed-forward classifiers predicting SS and SC tags respectively.

The base dataset is split into train and validation parts so that train is 80% and validation is 20% of the base dataset size. The model is trained in a multi-task manner using slanted triangular learning rate scheduler along with gradual unfreezing and discriminative fine-tuning. The configuration is available on GitHub[9].

The model is implemented using the AllenNLP library and publicly available on our GitHub page[10].

For the base version of the parser, we used the pre-trained RuBERT-tiny[11] text encoder, which is 15 times smaller than the well-known DeepPavlov's RuBERT[12]. This exact version was submitted for the competition and set the baseline score, which can be observed in table 2.

We also experimented with the pre-trained Base XLM-RoBERTa[13] text encoder out of competition scope in order to evaluate the importance of embedding quality and the influence of language-specific features. The comparative quality for the variants can be seen in the table below.

	Total	Lemma	POS	Features	UAS	LAS	SemSlot	SemClass
RuBERT-tiny	92.2%	96.1%	98.2%	95.3%	90.0%	85.6%	87.8%	92.2%
XLM-R	**95.1%**	**97.3%**	**98.8%**	**96.8%**	**93.5%**	**89.8%**	**94.3%**	**94.8%**

Table 3: Baseline parser test scores using different encoders

The overall scores have not improved as much as we have expected. Nevertheless, there is a significant growth for SSs and some improvement for SCs scores. As the XLM-R model is multilingual, we can suspect that it could also positively influence the results, as well as its size.

5.2 Participant's model

Apart from the baseline, there is one model proposed for the competition. Generally speaking, it is close to the baseline, but has two new features added.

First, each non-linear feed-forward classifier head is accompanied with Linear Chain Conditional Random Field (CRF) (Huang et al., 2015). Although token embeddings are believed to contain some relevant information about all words in a sentence, feed-forward classifiers predict labels independently, and do not take other heads predictions into account. That is, for example, POS-tag of the last token in a sequence does not depend on the POS-tag of the first one. Chain CRFs are known to overcome this problem by explicitly utilizing tags relationships and modelling joint distribution of the whole sequence of tags throughout timeline, rather than that of a single tag at each timestep.

Second, the Label Attention Layer (Mrini et al., 2019) was introduced into the biaffine dependency classifier. The label attention is a modified version of self-attention, where each head is reasoned by a classification label, and not the other tokens of a sentence, as in the latter. The authors suggest that this mechanism allows the model to learn label-specific views of the sentence, and proves the technique improves the quality of biaffine dependency parser.

Unluckily, due to implementation issues, the proposed model did not manage to beat the baseline score, although, if implemented correctly, it would definitely have.

Now let's consider the evaluation metrics used for the estimation of the parser. Some of them represent the improved variants of the metrics used in GramEval2020 Shared Task (Lyashevskaya et al., 2020), the others had to be introduced specifically for the SSs and SCs.

[9]https://github.com/dialogue-evaluation/SEMarkup-2023/blob/main/parsers/configs/baseline.jsonnet
[10]https://github.com/dialogue-evaluation/SEMarkup-2023/tree/main/parsers
[11]https://huggingface.co/cointegrated/rubert-tiny
[12]https://huggingface.co/DeepPavlov/rubert-base-cased
[13]https://huggingface.co/xlm-roberta-base

5.3 Evaluation Metrics

The evaluation metric is an average of seven scoring functions. The latter can be divided into three categories: morphological, syntactic and semantic scores.

5.3.1 Morphology

Lemmatization score is a weighted true-false classifier, expressed as follows[14]:

$$ScoreLemma(test, gold) = LemmaWeight(gold_{POS}) * [Norm(test_{lemma}) = Norm(gold_{lemma})].$$

The weighting function depends upon a POS tag of a token. If the tag is one of *ADP*, *CCONJ*, *INTJ*, *PART*, *PUNCT*, *SCONJ*, *SYM* or *X*, the weight equals to 0.3. Otherwise, it equals to 0.7. The idea behind this is that we want immutable words to influence score less than mutable ones: normally, a dataset would have many more immutable words and this would make an overall score for lemmatization higher than it should actually be.

Function *Norm* makes input lowercase and replaces letter $ё$ with letter $е$. For instance, the expression $[Norm(\text{Ёж}) = Norm(\text{еж})]$ equals to 1.

POS score is a true-false classifier:

$$ScorePOS(test, gold) = [test_{POS} = gold_{POS}].$$

Grammatical features of a token correspond to a set of pairs $(category,\ grammeme)$ where the category depends on the POS tag of a token and the grammeme depends on the category. Given a grammatical features of a token $token_{feats}$, denote a grammeme of a category cat as $token^{cat}_{feats}$. If features have no cat category, assume the notation equals to empty set.

Now, we can define *grammatical features score*:

$$ScoreFeats(test, gold) = Penalty(test_{feats}, gold_{feats}) * \frac{\sum_{(cat,\ gram)\ \in\ gold_{feats}} CatWeight(cat) * \left[gram = test^{cat}_{feats}\right]}{\sum_{(cat,\ gram)\ \in\ gold_{feats}} CatWeight(cat)}.$$

The left-hand multiplier penalizes test features for excessive length:

$$Penalty(f_{test}, f_{gold}) = \begin{cases} \frac{1}{1+(Size(f_{test})-Size(f_{gold}))} & \text{if } Size(f_{test}) > Size(f_{gold}), \\ 1 & \text{otherwise.} \end{cases}$$

This does not allow test features to contain too many categories. The latter is undesirable, for otherwise the model would gain higher scores by simply labelling a token with all possible categories.

The right-hand side is a weighted mean of true-false grammeme classifiers. The $CatWeight$ function accounts for category size, so that grammemes of a big category (which are harder to guess) are more valuable than those of a small one.

5.3.2 Syntax

We use Unlabeled and Labeled attachment scores as a measure of syntactic match quality:

$$UAS(test, gold) = [test_{head} = gold_{head}],$$
$$LAS(test, gold) = [test_{head} = gold_{head}] * [test_{deprel} = gold_{deprel}].$$

[14] $[x = y]$ is Iverson bracket notation

5.3.3 Semantics

Semantic slot score is a true-false classifier:

$$ScoreSemslot(test, gold) = [test_{semslot} = gold_{semslot}].$$

Semantic class score is calculated based on semantic hierarchy of hyperonym classes:

$$ScoreSemclass(test, gold) = \frac{1}{1 + Distance(test_{semclass}, gold_{semclass})},$$

where

$$Distance(u, v) = \begin{cases} PathLength(u, v) & \text{if } u \text{ and } v \text{ are in same tree,} \\ \infty & \text{otherwise.} \end{cases}$$

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

Averaging

Due to the weighting, some scores are strictly less than one, which means the score of ideal match is also less than one. To account for this issue, we divide the sum of test-gold scores by the sum of gold-gold scores. Now, a perfect match yields an accuracy of one.

Comparative evaluation

It would be interesting to compare the parser's quality with the quality of parsers, based on separate markup levels, namely, UD parsers and parsers aimed at the tasks of semantic labelling (such as UCCA (Hershcovich et al., 2019) or DRS (van Noord et al., 2020)), and to evaluate whether the integral approach makes the parsing process easier or not.

However, at the current stage, such comparison does not seem appropriate. We evaluate data of different corpora. The above mentioned semantic parsers do not suggest Russian parsing. Our metrics differ, as we made them stricter taking the word mutability into account and introducing penalty for excessive grammatical features.

Finally, it would be natural to compare our parser with the solutions for Word Sense Disambiguation task, as it can be solved with the help of the current dataset as well. For Russian, such work was conducted in 2020 (Bolshina and Loukachevitch, 2020). The best score was achieved on a fiction dataset with the use of a bi-LSTM model, and its f1 score is 95%. We have also calculated micro f1 (94%) and macro f1 (71%) scores for our baseline; the authors of the above-mentioned work haven't specified which type of f1 they used, unfortunately. As far as macro f1 is concerned, its lower score deals with SCs and SSc which are more rare and therefore poorly presented in the corpus. After analyzing such cases, we will enrich the corpus with the necessary data. Nevertheless, one should keep in mind that it is just a basic solution which can be seriously improved.

6 Results and Conclusion

First of all, we have simplified the full Compreno markup and made its usage easier. The markup has been converted into the UD format, which has been enriched with the semantic pattern. Therefore, we have elaborated the new standard, CoBaLD, for the full multi-level markup, which is the UD format including both morphosyntax and semantics.

Second, we have obtained the 400K Russian corpus CoBaLD-Rus annotated with the new standard. It is the first Russian corpus annotated in the format of this kind.

Third, we have tested the usage of the CoBaLD format during the SEMarkup-2023 Shared Task and created the integral three-level parser for this format based on neural networks.

Further plans concern several areas.

Currently, we are working on some optimizations of the labeling format, CoBaLD parser and the Compreno-to-UD converter, dealing mostly with ellipsis restoring and possibly adding other semantic information such as coreference. For that matter, we plan to move to the CONLL Plus format for better compatibility with UD.

Other important task is the creation of the English dataset annotated according to the CoBaLD standard. It would allow one to conduct comparative studies which can, inter alia, take semantic sketches into account (Detkova et al., 2020).

We are also considering the ability to hold a shared task on a "Lexical Sample" problem of WSD based on our markup standard.

Besides, we intend to experiment with the Language Transfer task which implies that the model trained on the donor language data can be applied to the data of the recipient language. The analysis of zero-shot transfer results may reveal a number of interesting details concerning the architecture of the parser itself and the qualities of the labelling format.

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.

KV Anisimovich, K Yu Druzhkin, KA Zuev, FR Minlos, MA Petrova, and VP Selegei. 2012. Syntactic and semantic parser based on abbyy compreno linguistic technologies. // *Proc Dialogue, Russian International Conference on Computational Linguistics*, P 91–103.

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.

Igor Boguslavsky. 1999. Translation to and from russian: the etap system. // *EAMT Workshop: EU and the new languages*.

Angelina Bolshina and Natalia Loukachevitch. 2020. All-words word sense disambiguation for russian using automatically generated text collection. *Cybernetics and Information Technologies*, 20(4):90–107.

Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. 2006. Generating typed dependency parses from phrase structure parses. // *Lrec*, volume 6, P 449–454.

J Detkova, V Novitskiy, M Petrova, and V Selegey. 2020. Differential semantic sketches for russian internet-corpora. // *Computational Linguistics and Intellectual Technologies*, P 211–227.

Vladimir Dobrovolskii, Mariia Michurina, and Alexandra Ivoylova. 2022. RuCoCo: a new russian corpus with coreference annotation. // *Computational Linguistics and Intellectual Technologies*. RSUH.

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

Jeroen Groenendijk, Theo MV Janssen, and Martin Stokhof. 1984. *Truth, Interpretation and Information*. Foris Dordrecht.

Jan Hajic, Barbora Vidová-Hladká, and Petr Pajas. 2001. The prague dependency treebank: Annotation structure and support. // *Proceedings of the IRCS Workshop on Linguistic Databases*, P 105–114.

Daniel Hershcovich, Zohar Aizenbud, Leshem Choshen, Elior Sulem, Ari Rappoport, and Omri Abend. 2019. Semeval-2019 task 1: Cross-lingual semantic parsing with ucca. *arXiv preprint arXiv:1903.02953*.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991.

A Ivoylova, D Dyachkova, M Petrova, and M 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»*.

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.

Khalil Mrini, Franck Dernoncourt, Trung Bui, Walter Chang, and Ndapa Nakashole. 2019. Rethinking self-attention: An interpretable self-attentive encoder-decoder parser. *CoRR*, abs/1911.03875.

MA Petrova. 2014. The compreno semantic model: the universality problem. *International Journal of Lexicography*, 27(2):105–129.

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.

Hiroshi Uchida and Meiying Zhu. 2001. The universal networking language beyond machine translation. // *International Symposium on Language in Cyberspace, Seoul*, P 26–27.

Rik van Noord, Antonio Toral, and Johan Bos. 2020. Character-level representations improve drs-based semantic parsing even in the age of bert. *arXiv preprint arXiv:2011.04308*.

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.