# Bulgarian Event Extraction with LLMs

**Kiril Simov, Nikolay Paev, Petya Osenova, Stefan Marinov**
Artificial Intelligence and Language Technology
Institute of Information and Communication Technologies
Bulgarian Academy of Sciences
Bulgaria
kivs@bultreebank.org, nikolay.paev@iict.bas.bg,
petya@bultreebank.org, stefanl.marinov99@gmail.com

## Abstract

The paper presents the results from the experiments with two large language models (LLMs) — T5 and Llama — for extracting events from a Bulgarian event corpus. The two models were pretrained by us on 35 Billion Token Bulgarian Corpus. The extraction was performed within the context of one sentence. Our approach aims at balancing the ACE-oriented approach that uses triggers in event detection, and the MUC-oriented one that uses more general event types. The evaluation relies on the IoU (Intersection over Union) of token spans and is twofold. The first one refers to the predicted event token span. Here if the span is correct, the semantic roles within the event are further checked. The second one refers to the triple of an event type, its semantic roles and participants. The results are promising. A qualitative evaluation is provided as well.

## 1 Introduction

We consider the event modeling as a main mechanism for the knowledge representation in text data. Within this perspective, our goal is to provide some common knowledge for the integration of resources that are necessary for supporting the research in the Social Sciences and Humanities (SS&H). This common knowledge is organized as a Bulgarian-centric Knowledge Graph (BGKG). In this paper we present some specific *Event Extraction* (EE) models based on Large Language Models (LLMs) for Bulgarian. The resource used for training and evaluation is the Bulgarian event corpus (BEC)[1]. The corpus was annotated in accordance with the Bulgarian specific guidelines based on CIDOC-CRM (Doerr, 2021) and further extended with adapted frames from FrameNet (Baker et al., 1998). More information about the corpus specifics can be found in (Osenova et al., 2022). Here we use a consequent version of the corpus developed on the basis of the publicly available one. It should be noted that the newest version v2.0, used here, reflects changes of some previous annotation principles and decisions.

The event extraction (EE) from texts is an active area of research in NLP. The ACE (Automatic Content Extraction — English Annotation Guidelines for Events)[2] guide defines an Event as a specific occurrence involving participants, which happens, and frequently denotes a change of state — (LDC, 2005). Thus, each *Event* is modeled by its *Type* and *Participants*. Participants are named by their *Roles* in the event. The types of events are organized in an event ontology where a hierarchy of events is defined. The participant roles can be inherited along the subclass relation over the event types. For each role a restriction over the appropriate *Role values* is stated. Within BEC 2.0 each occurrence of the event was annotated by its linear span. In most cases the span contains the so-called *Trigger* — a word or a phrase that comprises the core semantics of the event, predominantly verbs.[3] In many works Event Extraction is defined as consisting of two subtasks: (1) *Event Detection*(ED), and (2) *Event Argument Extraction* (EAE) — see (Simon et al., 2024), (Lai, 2022), and citations within them. In the first task the system is expected to identify the span and the type of the event. Usually, the event detection starts with a trigger recognition since trig-

[1] An earlier version of BEC – v.1.0 – is publicly available through ELRA https://catalog.elda.org/en-us/repository/browse/ELRA-W0329/

[2] https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-events-guidelines-v5.4.3.pdf

[3] In BEC also event occurrences that are not expressed by verbal triggers are considered, but these are defined by certain syntactic constructions in the text. For instance, *"Aleksandar Stoimenov Stamboliyski (1 March 1879 – 14 June 1923) . . ."* refers to birth and death of "Aleksandar Stoimenov Stamboliyski" without any explicit trigger.



*Proceedings of Recent Advances in Natural Language Processing*,pages 1163–1171
Varna, Sep 8–10, 2025
https://doi.org/10.26615/978-954-452-098-4-134

gers anchor the events. The second task identifies the arguments (participants) of the event in the text and relates them to their roles. When the two tasks are solved separately by different models, the result is called a *pipeline* approach. In cases when the model solves both tasks together, the result is called a *joint* approach.

The LLM-based Event Extraction typically uses both — classification approaches (in most cases pipelines) and generative approaches. In the *classification approaches*, the systems determine the spans of the trigger and the role values in the text, then classifies them with respect to some event ontology. The *generative systems* exploit the LLMs to generate the event representation as a text (encoder-decoder architectures or decoder-only ones). Simon et al. (2024) classify generative systems in two groups: (1) systems generating event representation in terms of formal structure templates — each template contains elements for the events type, trigger, roles and slots for their values, and (2) systems generating event representation in terms of natural language templates — a text in natural language with empty places for the element values of the recognized event. In this paper we present generative systems of the first kind — producing a formally structured output. We rely on two different output structures: a JSON format which reflects the whole structure of the events, and a triple-related representation [event instance, role, text span]. We first perform experiments with different LLMs for Bulgarian. At the same time, we pay special attention to the evaluation of the results in order to avoid some of the well-known pitfalls of EE evaluation (Peng et al., 2023).

For the experiments the event corpus was split like follows: for training — 13 233 sentences, for development — 1 471 sentences, and for testing — 1 634 sentences. The results are reported on both levels – Named Entities (NE) and events - since NEs enhance the retrieval of the event participants. It should be noted that the Macro F1 of NEs outperforms the one reported in (Osenova et al., 2022).

The paper is structured as follows: in the next section some related works are discussed. In section 3 the Bulgarian event corpus is presented from two points of view - the initial already publicly available version 1.0, and the current 2.0 version which will be made publicly available as well. Section 4 outlines the experimental setting. Section 5 reports on the experiments and evaluation. Section 6 concludes the paper.

## 2 Related Work

As it was mentioned above, we implemented a joint generative event extraction system that produces a formal textual representation of the extracted event information. More specifically, we follow the `Text2Event` approach to EE (Lu et al., 2021). `Text2Event` defines an end-to-end generative model that transforms an input of tokens into a linearized event structure. The authors impose the following requirements for the linearized event representations: (1) to be able to express multiple event records in a text as one expression; (2) to be easy to reversibly convert the text to event records in a deterministic way; and (3) to be similar to the token sequence of general text generation tasks so that text generation models can be leveraged and transferred easily. The paper defines the linearized representation as an S-expression which for each event occurrence contains an expression starting with a pair *the type of the event* and *the corresponding text span* followed by a list of *role* and *span* pairs representing the arguments of the event mentioned in the text. Then the transformed dataset (for example, ACE, mentioned above) has been used for training a `T5` encoder-decoder language model (Raffel et al., 2019). In order to restrict the output to the required event representation, the authors explore the *constrained decoding* that provides a mechanism for exploiting the knowledge from an event schema to form the output.

In our work we implemented the `Text2Event` approach by defining different linearized event representations. This implementation relies on our own pre-trained LLMs — `T5` (Raffel et al., 2019) and `Llama` (Touvron et al., 2023a). These models were pre-trained on Bulgarian data only, and for some basic NLP tasks (sentence splitting, tokenization, lemmatization, POS, UD parsing, WSD) they produced better results in comparison to other available models (of comparable size) that also cover Bulgarian.

Peng et al. (2023) discuss EE evaluation for different approaches to the EE task such as *classification*, *sequence labeling*, *span predication*, and *conditional generation*. The authors identify three major hidden pitfalls. The first one — **Data preprocessing discrepancy**, is related to the fact that two EE evaluations using different pre-processing methods are not directly comparable. The next

one — **Output space discrepancy**, reflects the fact that the outputs of different models are also not directly comparable. For example, the same span could be interpreted as different numbers of event arguments and/or triggers. The last one — **Absence of pipeline evaluation**, uses different models that could exploit different inputs for the EAE task evaluation, and thus the results are not directly comparable again. The authors developed a consistent evaluation framework by standardizing the data processing and selection of event information to compare. During the initial experiments on BEC we observed similar problems. Thus, we considered developing an appropriate evaluation set of metrics for BEC. We also identified a new hidden pitfall related to the peculiarities of the annotation scheme which could result in subtle discrepancy during the annotation schema. Such an example is the representation of the pro-drop subject which is co-referent to some named entity in the text. The identification of such discrepancy could provide insights for further improvements of the BEC. Thus, we paid special attention to the evaluation of EE models trained on BEC.

## 3 The Bulgarian Event Corpus (BEC)

As it was mentioned in the Introduction, the aim of the annotation within BEC is to facilitate the inclusion of knowledge derived from texts into the Bulgarian-centric Knowledge Graph. In version 1.0 it has been done mainly on the basis of knowledge extraction from scientific publications, encyclopedic sources (like Wikipedia) and existing structured data. Thus, we consider text as a main source of information for the represented objects. In this work we follow the initial principles of the corpus, and focus mainly on the following entities: (1) **People** – their biographies – their characteristics, motivations, opinions, events in their lives, roles they played; (2) **Organizations** – their establishment, life cycle, activities, etc.; (3) **Objects** – geographical, artifacts, etc. and their features; (4) **Events** – place, time, participants (People, Objects), relations to other events; (5) **Time and Periods** – ordering of events in time; and (6) **Documents** – authors, content, opinions, mentions of people, events, entities, etc.

Our annotation scheme also reflects the rationale behind the CIDOC-CRM ontology since this ontology has been widely used in the areas of GLAM and Humanities. The annotation scheme keeps the envisaged two main layers: the first one is the Named Entity (NE) layer, and the second one is the event layer where each event is related to its participants. The annotation process followed this differentiation. At the first stage, only the NEs were annotated. The events and their participants were annotated at the second stage only. Thus, the latter process relied on the already annotated NEs. We also added a coreference mechanism and annotated the coreferential links among the participants within the events across the texts. This is necessary to support the extraction of facts from texts even when the participants are not explicitly mentioned within the text span of the event. Thus, in our version we do not add more texts but we attempted at a more consistent annotation scheme.

As it was mentioned above, our event annotation approach allows for some events not to have triggers. Thus, the event annotation depends on the annotated span, the genre of the text and its typical strategies of text organization. For example, in biographies the dates of birth and death are given in brackets without explicit predicates. For that reason we decided to exploit the so-called *span based annotation* within which the participants are annotated. For more details see (Laskova et al., 2020).

The set of named entity categories covers the main ontological types such as persons, locations, organizations, and time. The types in version 1.0 were more specialized. Some of these were specializations of the main ones, and others were specific to a certain domain/genre. There were specializations for Person and Location. The specific ones include **JUR**(idical) for legal documents, **REF**(erence) for bibliographical sources, among others. It is worth mentioning that already at this NE level the events are annotated as **EVT** (for example, wars, sports events, etc.). Three very similar classes for people are **PER-GPE**, **PER-LOC**, and **PER-GRP**. They reflect some subtle differences in categorization of groups of people. The presence of these variations would be valuable as part of the extracted knowledge, because we would like to include in the BGKG also statements about groups of people like *"Blacksmiths were given special rights to trade their wares during the Second Bulgarian Kingdom."*. Thus, we kept the NEs distinctions as a variety, but we performed more focused checks on the annotations and made these more consistent.

Since the event types coming from CIDOC-

| Event | Roles |
|---|---|
| **Birth** | **brought–into–life** (the new born person)<br>**parents** (the mother and father expressed together, for example "his parents")<br>**mother**<br>**father**<br>**place** (the birth place — usually the name of a city, country or hospital)<br>**time** (the time of birth — usually it's a date, etc) |
| **Occupation** | **agent** (person or organization)<br>**position** (doctor, teacher, volunteer, . . . )<br>**domain** (politics, education, . . . )<br>**skill** (knowledge, education, . . . )<br>**assignor** (person or organization playing role of (informal) employer)<br>**goal** (the purpose of taking the position)<br>**payment** (what is the actor received for doing the job — usually money)<br>**result** (the purpose of taking the position)<br>**in-event** (occupation related to/during an event: doctor during the war)<br>**time/beginning/end/period**<br>**place** |

Table 1: Some of the events defined in the annotation scheme.

CRM (like End of Existence, Death, Activity, Modification) are too general to cover all specificities of the domains, the annotation scheme was extended on the basis of the English Framenet with the necessary localizations with respect to the data. For each event a set of participants were defined and mapped to the appropriate places in the CIDOC-CRM hierarchy of events.

Please note that some of the events were left general while other were detailed according to the text needs. Examples for general events (i.e. higher in the hierarchy) are causation, has-parts, start, end, possession, change, existence, destruction. Examples for specific events are teaching, occupation, publication, making a copy, donation. Each event is augmented with its typical roles. Table 1 presents two events with their appropriate participant roles.

In the examples within the table we can see different levels of granularity of the related participants. In the case of the event **Birth** the specific roles of *mother* and *father* are presented for each parent, on the one hand, and the more general and aggregated role of *parents* is outlined, on the other. In the case of the event **Occupation** - it describes the situation when an agent works at a position in a domain. The person needs to have some skills in order to perform a given job. Some position might depend on an assignor (or employer). Usually, the actor receives a payment for their job in money. But it is also possible for the payments to be in goods. The goal and the result of some occupation could be something in addition to the goals and results of the job itself. The occupation is taken for some period defined by different means. It is important to note that not all roles in an event are obligatory for a given instance of the event. Needless to say, one occurrence of an event in the text might mention only some of its possible roles.

The corpus comprises a wide variety of domain texts: texts from different periods of Bulgarian history; cultural artifacts like icons; scientific publications; archival documents; encyclopedic articles from the Bulgarian Wikipedia.

We measured the interannotator agreement between our 2 annotators on a token level trigger tagging. The test was conducted on the sentences in a separate document which was annotated by both annotators. We measured the trigger tagging since the trigger is the key marker for the existence of an event. It resulted in a Cohen's Kappa (Cohen, 1960) score of 0.7317. According to Landis and Koch (1977), a score between 0.61 and 0.80 is considered substantial, indicating a strong agreement among the annotators.

## 4 The Experimental Set Up

As previously mentioned, we used our own pre-trained general purpose LLMs as base models for the event extraction fine-tuning . For the pure classification tasks (e.g. Named entity recognition (NER task)) we rely on BERT models (Devlin et al., 2019) and for the conditional generation task (e.g. event extraction) we consider T5 and Llama models. [4] The `BERT` models are of two different sizes: BERT-base (125M parameters) and BERT-large (BERT-L) (355M parameters). The pre-training was performed for 3 epochs on a corpus of size of 20B

[4]The models are uploaded to Huggingface. `https://huggingface.co/AIaLT-IICT`.

Bulgarian tokens. The corpus consists of Bulgarian Web scraped data (CulturaX) from various topics and sources as well as Literature. The smaller `Llama` model has a hidden dimension of 768 and 12 layers, while the larger one has a dimension of 1024 and 48 layers. We pretrained `T5` (Raffel et al., 2019) and `Llama` (Touvron et al., 2023b) models for 3 epochs on a corpus of size 35B tokens. The `T5` model has 12 encoder and decoder layers, hidden dimension of 1024 and 403M parameter size. The pretraining objective is span denoising as proposed in the original paper with noise density of 0.25 and mean noise span length of 3 tokens. The `Llama` model has 16 layers, a hidden dimension of 2048 and 1.2B parameters and is pretrained on a language modelling task.

For the NER task, the classic approach with fine-tuning of a BERT model was used. For this, we rely on the BOI format of the data representation. There, the text is represented as a table in which the first column contains the tokens in the corpus — one token per line. Each token is marked up with `B-Type` if the token is the first one in a name entity of category `Type`, with `I-Type` if the token is an internal part in a named entity of category `Type`, and with `O` when the token is not related to a named entity.

For the event extraction task we follow the `Text2Event` approach to EE (Lu et al., 2021) and thus the task was modeled as a conditional generation using the `T5` model and `Llama` model. Before performing the experiments, we defined the formal textual representation for the event annotation. As it was mentioned above, we considered two output representations: a more general JSON format, representing the structure of the event, and a list of triples. Examples of both are given in the next section.

## 5 Experiments and Evaluation

As presented in the previous section, the BEC corpus comprises several interrelated annotations: a NE annotation layer; an event annotation layer; an event structure (a set of roles) annotation; a linking annotation; a co-reference layer. In the current experiment settings we worked with the first three layers: Events, Event Roles and NEs. Thus, in this section we report our results for the NER task and then – for the events and roles extraction tasks.

### 5.1 Event Extraction Training

We model the event extraction as a sequence-to-sequence task. The input sentence passes through the `T5` model, and the model is trained to generate a structured output that contains its predictions for the events and the roles.

#### 5.1.1 The Model Output Structure

For the output, we consider a structural representation in a JSON compatible scheme (Listing 1). This makes parsing the output trivial. The model was trained to produce a list of dictionary data structures, each representing a single event. Each event has three fields:

1. a type of the event
2. text which represents the span of the event
3. roles — list of pairs of a type and a text span

We do not use any constraint decoding because the model successfully learns to adhere to the desired output pattern. The invalid outputs are only 0.2%. Thus, in our view, it is not necessary to guide the generation process with additional information about the formal event format.

```
[{
    "type": "occupation",
    "text": "He is a prime-minister of
Bulgaria in the government of BZNS
(1919 - 1923).",
   "roles": [
      ["agent", "He"],
      ["position", "prime-minister"],
      ["assignor", "the government
           of BZNS"],
      ["beginning", "1919"],
      ["end", "1923"],
      ["assignor", "Bulgaria"],
      ["trigger", "is a prime-minister"]
   ]
}]
```

Listing 1: Example event representation of the sentence: He is a prime-minister of Bulgaria in the government of BZNS (1919 – 1923).

```
[
  ["occupation_1", "agent", "He"],
  ["occupation_1", "position",
        "prime-minister"],
  ["occupation_1", "assignor",
        "the government of BZNS"],
  ["occupation_1", "beginning", "1919"],
  ["occupation_1", "end", 1923"],
  ["occupation_1", "assignor",
        "Bulgaria"],
  ["occupation_1", "trigger",
        "is a prime-minister"]
```

```
]
```

Listing 2: Example event representations as triples of the sentence: He is a prime-minister of Bulgaria in the government of BZNS (1919 - 1923).

The above output representation can be easily translated into a list of triples similar to the relations [event type, role, text span]. In order to disambiguate between the different events occurrences in the text with the same event type, an index was appended to the event after its type (Listing 2). Furthermore, we also performed experiments with training the models to directly produce a list of triples of this kind.

#### 5.1.2 Fine-tuning

The models were trained for 10 epochs using the sentences of the event corpus. The batch size is 256 examples. We experimented with different learning rates and eventually selected 3e-4 linearly decaying to 0, which gives the best generalization performance. We chose the best model in terms of the validation loss over the epochs.

### 5.2 Evaluation of the Event Predictions

Similarly to the observations made in (Peng et al., 2023), the evaluation of the predictions of the model proved to be non-trivial, the most significant challenge being the alignment of the predicted events with the references in the test data set. Since the type and the roles cannot uniquely disambiguate the events, we consider the token span corresponding to the event to be the most representative characteristic for its evaluation. Following the evaluation approach used in the image object detection task, we calculated a similarity score between the predicted and the reference events, using the Hungarian matching algorithm (Kuhn, 1955) to perform the optimal 1:1 matching.

Since the token spans of the events are also not completely unique (there are events in the same sentence with equal spans), for the calculation of the similarity score, we considered a combination between the *Intersection over Union* (IoU) of the predicted and reference event spans, and a *BLEU* score of the string representation of the whole event structure [5]. The BLEU score is used only to match the predicted and reference events and thus it is not reported in the evaluation of the span prediction accuracy.

[5]To calculate the IoU, we first do a fuzzy aligning between the output text span tokens, the reference text span tokens and the input sentence tokens.

After performing the 1:1 matching, the precision, recall, and F1 metrics are calculated. The true positives can be subsequently evaluated in terms of a mean IoU of the spans, type prediction accuracy, and role extraction. The role extraction task uses the same prediction-reference matching and is followed by an evaluation of the type prediction accuracy and the IoU of spans.

The second explored option for the output — the representation as triples — can be evaluated more easily. Again, a 1:1 matching is performed, but this time the event type and its roles are strictly matched, and only the span is matched using both strategies — IoU and BLEU. When performing the evaluation, the event index was ignored, because one cannot be sure that the model will predict the events in the same order as in the reference dataset.

| Model | Precision | Recall | F1 |
|---|---|---|---|
| T5 | 0.8673 | 0.8199 | 0.8429 |
| Llama | 0.6748 | 0.7688 | 0.7187 |

Table 2: Event extraction metrics.

| Model | $\overline{\text{IoU}}$ | Type accuracy | Role F1 |
|---|---|---|---|
| T5 | 0.9701 | 0.9025 | 0.9026 |
| Llama | 0.8931 | 0.7534 | 0.8075 |

Table 3: Metrics over the true positives. The column $\overline{\text{IoU}}$ shows the mean Intersection over union of the span predictions. The **Role F1** column shows the F1 metric over the extraction of the role spans.

| Model | Output | Precision | Recall | F1 |
|---|---|---|---|---|
| T5 | Struct | 0.7200 | 0.6481 | 0.6822 |
| Llama | Struct | 0.4255 | 0.4908 | 0.4558 |
| T5 | Triples | 0.4984 | 0.4207 | 0.4563 |
| Llama | Triples | 0.3796 | 0.4243 | 0.4007 |

Table 4: Event triples extraction metrics. The table compares the results of models fine-tuned for structural JSON output which is later mapped to triples to models fine-tuned to directly produce triples.

Table 2 shows the metrics calculated during the 1:1 matching of the predicted and reference event spans. Table 3 shows the metrics for the events with correctly matched spans — the true positives. We could also calculate metrics on the true positive roles of the true positive events. The role type accuracy of the T5 model is 0.9102, and of the Llama model is 0.8138.

In general, the model with the T5 architecture performs better than the Llama model despite having fewer parameters. We assume that this is because the bidirectional encoder of T5 can produce better embeddings of the input tokens.

Table 4 shows the extraction metrics for the alternative representation as triples. The models producing structural output that is later mapped to triples are compared to models trained to directly predict the triples.

Models trained directly to produce triples perform worse than those with the structural output. An intuition behind this is that the more extended output pattern in the structural output (through the prediction of the event span) enables the model to better focus on the particular event. One can draw parallels between this and the instruction fine-tuning task, where the more verbose output of the models (the tokens in thinking stage) helps in the reasoning. Thus, in future, we plan to focus on the triple generation approach but already combined with extended output schemes in order to provide better context information.

### 5.3 NER Experiments

Lu et al. (2021) use the Constrained Decoding mechanism to support the generation of a valid output. In our work, we do not use this mechanism for the generation of our structural representation because the models already learn it effectively. But we think that some additional information outside of the event annotation might be useful. It might constrain the decoding thus producing a semantically better output. Such additional constraints could be derived also from the other layers of annotation such as the NEs annotation, especially when the NER task has been performed with a high accuracy.

Here we present our fine-tuned BERT models for NER. We trained them for 10 epochs on the named entities and other annotations from the event corpus. We used a linearly decaying learning rate with a peak value of 2e-05 and selected the best checkpoint by validation loss.

| Model | Accuracy | Macro F1 |
|---|---|---|
| BERT-Base | 0.9279 | 0.8026 |
| BERT-Large | 0.9305 | 0.8123 |

Table 5: NER metrics.

The results are shown in Table 5. It is evident that the larger model performs better, as expected.

The top three classes of NEs are PER(son), TIME, and LOC(ation). These classes are the most frequent arguments of the events in our domain. The F1 score for them is as follows: PER : 0.9388, TIME : 0.9214, LOC-GPE : 0.9118. We plan to use the predictions of these classes to guide the mapping between the roles in the annotation schema and the candidate arguments within the text.

### 5.4 Manual Evaluation and Discussion

An initial manual evaluation of the event extraction task was performed on the results of the best-performing T5 model over the test set. The evaluation of a random 10 % sample containing extracted events with false positives suggests that all predictions can be considered correct. 53.85 % of the false positive examples have the correct type, the other 46.15 % have a wrong predicted type. The main reason for the discrepancy is that very often the trigger is ambiguous and thus was incorrectly recognized. In cases when the type of the extracted event is correct, the erroneous evaluation is due to the problematic extraction of the corresponding roles.

Also note that 29.41 % of the false negatives contain event-bearing embedded participles to a noun within a bigger event span. The inner event usually shares an argument with the larger event and this causes the problem. For example, in the following sentence the passive participle ‘narichana’ (called) should trigger the Naming event but it failed: ”Bregovata batareya, narichana ”Sveta troitsa” e premestena ot Varna” (Coastal-the battery, called ”Holy-the Trinity”, is moved from Varna”, The coastal battery, called the ”Holy Trinity”, has been moved from Varna). In the process of the re-annotation of BEC, these events are being discarded from the annotation process, but some of the previous annotations remained and thus caused the noise. Actually, the model correctly skipped these cases, because they are not frequent in the training set. However, in the evaluation phase, the matching between the test set and the gold corpus was problematic. This means that 30 % of the reported false negatives by the model are true negatives in the annotations.

Another type of a mismatch between the predictions and the gold annotations occurs within the true positive events. Most of the time, the model seems to predict a more general or a more specific

type that is not necessarily a mistake. The most common among these are the events of **Characterization**, **Occupation** and **Appointment**. Errors can be caused also by the mixing related events such as **Residence** and **Moving**. In general, it can be pointed out that half of the mismatches are errors on the model side, 30% are ambiguous, thus noisy, and 20% are annotation errors (where the model prediction was the correct one).

The evaluation of the errors in the role predictions leads to similar conclusions.

The false positives are usually found in sentences with a pro-drop subject, which remained unannotated, but predicted by the model as a missing agent. In a half of the examples, the model correctly predicts a role that is present in the sentence.

The evaluation of a sample of false negatives shows that they contain predominately roles that are mainly adjuncts, not arguments. Sometimes also a misalignment in the predicted span is present.

The mismatches in the types of the roles appear to be similar to the case of the event types — with the frequent prediction of a more general or a more specific role. For example, the role **Beginning** is more specific while the role **Time** is more general. In the manual annotation process these roles are considered relatively interchangeable in some cases. The model evaluation considers them wrong but they might be interpreted by annotators as acceptable. In cases when the event type is wrongly predicted, the role spans still can be correct, despite being with wrong types (For example, when an **Agent** becomes a **Creator**).

As it became obvious from the manual evaluation, the errors are of two types: (1) errors in the annotation; (2) errors caused by events that are difficult to model, or rarely present. In future we plan to handle better the second type.

## 6 Conclusion and Future Work

The event extraction task is far from trivial since it is a complex endeavor from both points of view – world knowledge and its language representation. The narrower the domain, the better. On the other hand, the LLM-based applications — even in the broad domain of Humanities and Social Sciences — become promising.

In this paper we showed such promising results for Bulgarian event extraction in the H&SS domain. The results show that the important factors for the success of this task are the following: the annotation scheme and the underlying approach, the quality of the event annotation and the preceding NER annotation, the aspects to be evaluated, the (combination of) evaluation metrics, the training and fine-tuning of the used models, the aim of the application.

Thus, our future tasks are related to improving the interrelation among all these factors. We also plan to expand the event corpus on the basis of the LLM output with the necessary checks. We also plan to integrate more knowledge from the other layers of annotation in the event corpus.

Another direction of investigation is to define natural language templates for representation of the extracted event information as discussed within (Simon et al., 2024). This will require definitions of the appropriate templates. We believe that it will be interesting to learn such templates from larger domain specific corpora along the lines of (Yuan et al., 2018). In this way we hope that the templates will be better tuned to the various types of events and thus improve the result.

## Acknowledgments

The reported work has been supported by CLaDA-BG, *the Bulgarian National Interdisciplinary Research e-Infrastructure for Resources and Technologies in favor of the Bulgarian Language and Cultural Heritage, part of the EU infrastructures CLARIN and DARIAH*. We also acknowledge the provided access to *the e-infrastructure of the Centre for Advanced Computing and Data Processing* (the Grant No BG05M2OP001-1.001-0003).

## References

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 86–90, Montreal, Quebec, Canada. Association for Computational Linguistics.

J. Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages

4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Martin Doerr, editor. 2021. *Definition of the CIDOC-CRM Conceptual Reference Model : Version 7.1*. CIDOC.

Harold W. Kuhn. 1955. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, 2(1–2):83–97.

Viet Dac Lai. 2022. Event extraction: A survey.

J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33 1:159–74.

Laska Laskova, Petya Osenova, and Kiril Simov. 2020. Towards an interdisciplinary annotation framework: Combining nlp and expertise in humanities. In *Proceedings of CLARIN Annual Conference 2020*.

LDC. 2005. ACE (Automatic Content Extraction) English Annotation Guidelines for Events. *Linguistic Data Consortium*.

Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021. Text2Event: Controllable sequence-to-structure generation for end-to-end event extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2795–2806, Online. Association for Computational Linguistics.

Petya Osenova, Kiril Simov, Iva Marinova, and Melania Berbatova. 2022. The Bulgarian event corpus: Overview and initial NER experiments. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3491–3499, Marseille, France. European Language Resources Association.

Hao Peng, Xiaozhi Wang, Feng Yao, Kaisheng Zeng, Lei Hou, Juanzi Li, Zhiyuan Liu, and Weixing Shen. 2023. The devil is in the details: On the pitfalls of event extraction evaluation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9206–9227, Toronto, Canada. Association for Computational Linguistics.

Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Étienne Simon, Helene Olsen, Huiling You, Samia Touileb, Lilja Øvrelid, and Erik Velldal. 2024. Generative approaches to event extraction: Survey and outlook. In *Proceedings of the Workshop on the Future of Event Detection (FuturED)*, pages 73–86, Miami, Florida, USA. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023b. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Quan Yuan, Xiang Ren, Wenqi He, Chao Zhang, Xinhe Geng, Lifu Huang, Heng Ji, Chin-Yew Lin, and Jiawei Han. 2018. Open-schema event profiling for massive news corpora. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 587–596, New York, NY, USA. Association for Computing Machinery.