

## INPUT FILE

art00143\_frag02382\_DLL2\_[Brückner][Werner]\_518char.txt

## INPUT TEXT

Brückner, Werner Geburtsdatum: \* 1944 \* 1944; 1979–82 Stud. am Inst. f. Lit. in Leipzig, Schriftst., lebt in Dobbrun/Kr. Osterburg bei Magdeburg. Schriften: Der Fiedler im Stroh ( Gesch.) 1986; Später Sommer. Vom Ende eines Märchens, 1988; Das Mädchen aus dem Café Chantant (Gesch.) 1998. Deutsches Literatur-Lexikon Online Deutsches Literatur-Lexikon. Das 20. Jahrhundert Band 4 Herausgeber: Lutz Hagedstedt Begründet von Wilhelm Kosch De Gruyter (Berlin; New York; Boston) 2002 <https://doi.org/10.1515/dllo.zw.004.476>

## PROMPT (shortened)

From the INPUT extract structured information units that each have these fields: 'SUBJECT', 'EVENT\_TYPE', 'ATTRIBUTE', 'VALUE' Whenever a field cannot be filled, mark this field with 'Unknown'. Only extract information present in the INPUT. Do not add new information or any information from your training data. Consider only the following EVENT\_TYPES and ATTRIBUTES, skip all others: [('Acquaintance', ['acquaintanceType', 'begin', 'durationOfEvent', 'end', 'participants']), ('Birth', ['place', 'begin']), ('Citizenship', ['begin', [...], 'place', 'surveillingInstitution'])]

Keep the original wording of the text fragment where the extracted information was found. Do not change the original wording. It is important that you extract as much information as possible!! In your response, do not add any explanation or comments, instead return only the structured data.

Each element in the final output must have the following structure:

```
{'fragment_text': 'Peter Schmidt wurde am 30. Juni 1960 in Berlin geboren', 'extracted_info':  
[{'SUBJECT': 'Peter Schmidt', 'EVENT_TYPE': 'birth', 'ATTRIBUTE': 'begin', 'VALUE': '1975-06-30'},  
{'SUBJECT': 'Peter Schmidt', 'EVENT_TYPE': 'birth', 'ATTRIBUTE': 'place', 'VALUE': 'Berlin'}]}
```

## RAW MODEL RESPONSE

~~Do not add any Explanation or comments.~~ ```json { "fragment\_text": "Brückner, Werner Geburtsdatum: \* 1944 \* 1944", "extracted\_info": [ { "SUBJECT": "Brückner, Werner", "EVENT\_TYPE": "Birth", "ATTRIBUTE": "begin", "VALUE": "1944" } ] }, { "fragment\_text": "1979–82 Stud. am Inst. f. Lit. in Leipzig", "extracted\_info": [ { "SUBJECT": "Brückner, Werner", "EVENT\_TYPE": "Education", "ATTRIBUTE": "begin", "VALUE": "1979" }, { "SUBJECT": "Brückner, Werner", "EVENT\_TYPE": "Education", "ATTRIBUTE": "end", "VALUE": "1982" }, { "SUBJECT": "Brückner, Werner", "EVENT\_TYPE": "Education", "ATTRIBUTE": "institution", "VALUE": "Inst. f. Lit." }, { "SUBJECT": "Brückner, Werner", "EVENT\_TYPE": "Education", "ATTRIBUTE": "place", "VALUE": "Leipzig" } ] }, { "fragment\_text": "lebt in Dobbrun/Kr. Osterburg bei Magdeburg.", "extracted\_info": [ { "SUBJECT": "Brückner, Werner", "EVENT\_TYPE": "Flat", "ATTRIBUTE": "place", "VALUE": "Dobbrun/Kr. Osterburg bei Magdeburg" } ] } } ``` ~~This captures the key events and attributes provided in the text, using the specified format. Each event type and attribute is mapped to the corresponding value found in the text. The subject for all events is "Brückner, Werner". Any missing information is left as `Unknown`. This approach ensures that all relevant information from the input is captured accurately. Note: The text does not provide explicit details about duration of events, social classes, or other attributes that might be required by the schema but are not mentioned in the input. Therefore, those fields are marked as `Unknown` if needed. However, based on the provided guidelines, we only extract what~~

## (PROCESSED) OUTPUT

```
[{'extracted_info': [{'ATTRIBUTE': 'begin',  
    'EVENT_TYPE': 'Birth',  
    'SUBJECT': 'Brückner, Werner',  
    'VALUE': '1944'}]},  
    'fragment_text': 'Brückner, Werner Geburtsdatum: * 1944 * 1944'},
```

(1a)

```
{'extracted_info': [{'ATTRIBUTE': 'begin',  
    'EVENT_TYPE': 'Education',  
    'SUBJECT': 'Brückner, Werner',  
    'VALUE': '1979'}]},
```

(2a)

```
{'ATTRIBUTE': 'end',  
    'EVENT_TYPE': 'Education',  
    'SUBJECT': 'Brückner, Werner',  
    'VALUE': '1982'}]},
```

(2b)

```
{'ATTRIBUTE': 'institution',  
    'EVENT_TYPE': 'Education',  
    'SUBJECT': 'Brückner, Werner',  
    'VALUE': 'Inst. f. Lit.'}},
```

(2c)

```
{'ATTRIBUTE': 'place',  
    'EVENT_TYPE': 'Education',  
    'SUBJECT': 'Brückner, Werner',  
    'VALUE': 'Leipzig'}]},
```

(2d)

```
'fragment_text': '1979-82 Stud. am Inst. f. Lit. in Leipzig'},
```

```
{'extracted_info': [{'ATTRIBUTE': 'place',  
    'EVENT_TYPE': 'Flat',  
    'SUBJECT': 'Brückner, Werner',  
    'VALUE': 'Dobbrun/Kr. Osterburg bei Magdeburg'}]},  
    'fragment_text': 'lebt in Dobbrun/Kr. Osterburg bei Magdeburg.'}]
```

(3a)

## Precision-oriented evaluation

(„What's the best gold match for each extracted tuple?“)

Field	Sim Score	Type	Candidate	Gold
event_type	1.00	<class 'str'>	<b>Birth</b>	Birth
attribute	1.00	<class 'str'>	<b>begin</b>	begin
value	0.74	<class 'datetime.datetime'>	1944	1944-01-01
FINAL_SCORE	0.910			
gated_pass	True			

*! Mistake in Eval Data  
✓ Score proves robust*

Field	Sim Score	Type	Candidate	Gold
event_type	1.00	<class 'str'>	<b>Education</b>	Education
attribute	1.00	<class 'str'>	<b>begin</b>	begin
value	0.74	<class 'datetime.datetime'>	1979	1979-01-01
FINAL_SCORE	0.910			
gated_pass	True			

*(same as previous)*

Field	Sim Score	Type	Candidate	Gold
event_type	1.00	<class 'str'>	<b>Education</b>	Education
attribute	1.00	<class 'str'>	<b>end</b>	end
value	0.74	<class 'datetime.datetime'>	1982	1982-01-01
FINAL_SCORE	0.910			
gated_pass	True			

*(same as previous)*

Field	Sim Score	Type	Candidate	Gold
event_type	1.00	<class 'str'>	<b>Education</b>	Education
attribute	1.00	<class 'str'>	<b>institution</b>	institution
value	0.14	<class 'str'>	Inst. f. Lit.	Literaturinstitut Johannes R. Becher""
FINAL_SCORE	0.701			
gated_pass	True			

*Model uses literal wording from  
input text, partial score **punished**  
**harshly**, final score **acceptable***

Field	Sim Score	Type	Candidate	Gold
event_type	1.00	<class 'str'>	<b>Education</b>	Education
attribute	1.00	<class 'str'>	<b>place</b>	place
value	1.00	<class 'str'>	<b>Leipzig</b>	Leipzig
FINAL_SCORE	1.000			
gated_pass	True			

*Perfect precision unit*

Field	Sim Score	Type	Candidate	Gold
event_type	0.29	<class 'str'>	<b>Flat</b>	Education
attribute	1.00	<class 'str'>	<b>place</b>	place
value	0.26	<class 'str'>	<b>Dobbrun/Kr. Osterburg bei Magdeburg</b>	Leipzig
FINAL_SCORE	0.000			
gated_pass	<b>False</b>			

*Correct information but not contained in  
eval data. Gets **gated out** due to low  
partial scores - OK*

## Recall-oriented evaluation

(„Which extracted tuple ist he best match for each gold information?“)

Field	Sim Score	Type	Candidate	Gold
event_type	1.00	<class 'str'>	Birth	Birth
attribute	1.00	<class 'str'>	begin	begin
value	0.63	<class 'datetime.datetime'>	1944	1944-09-15
FINAL_SCORE	0.869			
gated_pass	True			

*Date recalled but with lower granularity.  
☑ Score proves robust*

Field	Sim Score	Type	Candidate	Gold
event_type	1.00	<class 'str'>	Birth	Birth
attribute	1.00	<class 'str'>	begin	begin
value	0.74	<class 'datetime.datetime'>	1944	1944-01-01
FINAL_SCORE	0.910			
gated_pass	True			

*(see above)*

Field	Sim Score	Type	Candidate	Gold
event_type	1.00	<class 'str'>	Birth	Birth
attribute	0.53	<class 'str'>	begin	place
value	0.28	<class 'str'>	1944	Aue
FINAL_SCORE	0.607			
gated_pass	True			

*Mismatch so final score is too high (0.61) → gating too permissive*

Field	Sim Score	Type	Candidate	Gold
event_type	1.00	<class 'str'>	Education	Education
attribute	1.00	<class 'str'>	begin	begin
value	0.74	<class 'datetime.datetime'>	1979	1979-01-01
FINAL_SCORE	0.910			
gated_pass	True			

*(see above)*

Field	Sim Score	Type	Candidate	Gold
event_type	1.00	<class 'str'>	Education	Education
attribute	1.00	<class 'str'>	end	end
value	0.74	<class 'datetime.datetime'>	1982	1982-01-01
FINAL_SCORE	0.910			
gated_pass	True			

*(see above)*

Field	Sim Score	Type	Candidate	Gold
event_type	1.00	<class 'str'>	Education	Education
attribute	1.00	<class 'str'>	institution	institution
value	0.14	<class 'str'>	Inst. f. Lit.	Literaturinstitut Johannes R. Becher"
FINAL_SCORE	0.701			
gated_pass	True			

*Gold information successfully recalled but score is relatively low due model using the original abbreviated wording*

Field	Sim Score	Type	Candidate	Gold
event_type	1.00	<class 'str'>	Education	Education
attribute	0.47	<class 'str'>	institution	educationType
value	0.01	<class 'str'>	Inst. f. Lit.	OTHER_UNIVERSITY
FINAL_SCORE	0.494			
gated_pass	True			

*Recall challenging, but found best match, contributes with moderate score → OK*

Field	Sim Score	Type	Candidate	Gold
event_type	1.00	<class 'str'>	Education	Education
attribute	1.00	<class 'str'>	place	place
value	1.00	<class 'str'>	Leipzig	Leipzig
FINAL_SCORE	1.000			
gated_pass	True			

*Perfectly recalled gold information*

## ANALYSIS

MaxScores\_Precision (MSP) = [0.910, 0.910, 0.910, 0.701, 1.0, 0.0]  
 → Precision =  $\text{sum}(\text{MSP}) / \text{len}(\text{MSP})$   
 = 0.739

### Intuitive assessment:

- extracted years (1a, 2a, 2b) → very good precision, though missing some granularity.
- extracted education institution (2c) → accurate but difficult to evaluate due to abbreviations
- extracted education place (2d) → perfect extraction
- extracted flat place (3a) → not in evaluation data (although correct)

⇒ 0.739 seems adequate

MaxScores\_Recall (MSR) = [0.869, 0.910, 0.607, 0.910, 0.910, 0.701, 0.494, 1.0]  
 Recall(uncapped) =  $\text{sum}(\text{MSR}) / \text{len}(\text{MSR})$   
 = 0.800125  
 Maximum Recall possible =  $\text{len}(\text{extracted}) / \text{len}(\text{gold})$   
 = 0.75  
 → Recall = 0.75

### Intuitive assesment:

- 2 dates recalled with limited precision → contributes to recall as expected
- 1 birth place Aue not recalled but → contributes to recall higher than expected
- 2 dates recalled with limited precision → contributes to recall as expected
- 1 Ed. institution recalled but score != 1 → contributes to recall lower than expected
- 1 Education educationType not recalled → contributes to recall higher than expected
- 1 Ed. place perfectly recalled → contributes to recall as expected

⇒ 0.75 seems slightly too optimistic since traditional recall would be  $(6-1)/8 = 0.625$