

```
{
  "_id": ObjectId("62b4ff1a70b5f18e1b09af2f"),
  "tweet_id": 1477061496504991751,
  "lang": "en",
  "raw_text": "One last pic and i'll be gone, ma",
  "text": "One last pic and i'll be gone, make i",
  "is_retweet": false,
  "warnings": Array,
  "sui_keywords": Array,
  "author_id": "1416119651050917891",
  "drugs": Array,
    0: "My Way"
  "created_at": "2021-12-31T23:37:57.000Z",
  "sui_classifier": 0.9851861000061035
}
```

Table: examples\_example *records*

Add New Record

Enter values for the new record considering constraints

Name	Type
<b>id</b>	integer
<b>meta</b>	text
<b>filename</b>	varchar(1024)
<b>text</b>	text
<b>created_at</b>	datetime
<b>updated_at</b>	datetime
annotations_approved_by_id	integer
<b>project_id</b>	integer
<b>uuid</b>	char(32)

for query, connection through relations

Table: labels\_span *entities*

Add New Record

Enter values for the new record

Name	Type
<b>id</b>	integer
<b>prob</b>	real
<b>manual</b>	bool
<b>created_at</b>	datetime
<b>updated_at</b>	datetime
<b>start_offset</b>	integer
<b>end_offset</b>	integer
<b>example_id</b>	integer
<b>user_id</b>	integer
<b>label_id</b>	integer

Table: labels\_relation *relations*

<b>id</b>
prob
manual
created_at
updated_at
<b>example_id</b>
<b>from_id_id</b>
<b>to_id_id</b>
<b>type_id</b>
<b>user_id</b>

V1  
"0"rels  
are excl.

*users*

<b>auth_user</b>
<b>id</b>
password
last_login
is_superuser
username
last_name
email
is_staff
is_active
date_joined
first_name

query returns user id only

*projects*

<b>projects_project</b>
<b>id</b>
<b>name</b>
description
guideline
created_at
updated_at
project_type
collaborative_annotation
polymorphic_ctype_id
single_class_classification
random_order
created_by_id

*entity dict*

<b>label_types_spantype</b>
<b>id</b>
<b>text</b>
prefix_key
suffix_key
background_color
text_color
created_at
updated_at
project_id

*relations dict.*

<b>label_types_relationtype</b>
<b>id</b>
<b>text</b>
prefix_key
suffix_key
background_color
text_color
created_at
updated_at
project_id

*input*  
*output*



```
{
  "_id": ObjectId("62b4ff1a70b5f18e1b09af2f"),
  "tweet_id": 1477061496504991751,
  "lang": "en",
  "raw_text": "One last pic and i'll be gone, ma",
  "text": "One last pic and i'll be gone, make i",
  "is_retweet": false,
  "warnings": Array,
  "sui_keywords": Array,
  "author_id": "1416119651050917891",
  "drugs": Array,
    0: "My Way"
  "created_at": "2021-12-31T23:37:57.000Z",
  "sui_classifier": 0.9851861000061035
}
```

Table: examples\_example *records*

Add New Record

Enter values for the new record considering constraints

Name	Type
<b>id</b>	integer
<b>meta</b>	text
<b>filename</b>	varchar(1024)
<b>text</b>	text
<b>created_at</b>	datetime
<b>updated_at</b>	datetime
annotations_approved_by_id	integer
<b>project_id</b>	integer
<b>uuid</b>	char(32)

for query, connection through entities

V2  
"0"rels  
are incl.

Table: labels\_span *entities*

Add New Record

Enter values for the new record

Name	Type
<b>id</b>	integer
<b>prob</b>	real
<b>manual</b>	bool
<b>created_at</b>	datetime
<b>updated_at</b>	datetime
<b>start_offset</b>	integer
<b>end_offset</b>	integer
<b>example_id</b>	integer
<b>user_id</b>	integer
<b>label_id</b>	integer

Table: labels\_relation *relations*

<b>id</b>
prob
manual
created_at
updated_at
<b>example_id</b>
<b>from_id_id</b>
<b>to_id_id</b>
<b>type_id</b>
<b>user_id</b>

*input*

*output*

*projects*

Table: projects\_project

<b>id</b>
<b>name</b>
description
guideline
created_at
updated_at
project_type
collaborative_annotation
polymorphic_ctype_id
single_class_classification
random_order
created_by_id

*entity dict*

Table: label\_types\_spantype

<b>id</b>
<b>text</b>
prefix_key
suffix_key
background_color
text_color
created_at
updated_at
project_id

*relations dict.*

Table: label\_types\_relationtype

<b>id</b>
<b>text</b>
prefix_key
suffix_key
background_color
text_color
created_at
updated_at
project_id

*users*

Table: auth\_user

<b>id</b>
password
last_login
is_superuser
username
last_name
email
is_staff
is_active
date_joined
first_name

query returns user id only





F entity pairs:

1) find all sentence-wised  $e_1e_2$  combinations

1.1) both entities annotated by the same user // not true for pubmed 2

2) Drop-NonDrop relations only

3) drop same spans duplicates per sentence. covers "mirrored" spans

UNION w/ relations

Python: find NULL entity pairs that do not overlap spans with relations

↳ concatenate w/ relations