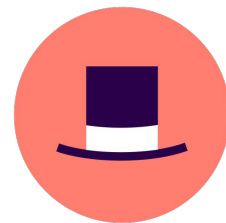


## MLLM project



Leibniz-Informationszentrum  
Wirtschaft  
Leibniz Information Centre  
for Economics

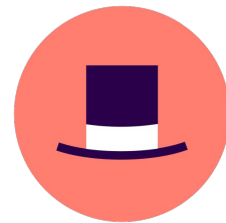


## Exercise 5: Set up and train a MLLM project

Let's set up a MLLM project. MLLM is an algorithm for lexical automated subject indexing, i.e. matching terms in document text to terms in a controlled vocabulary.

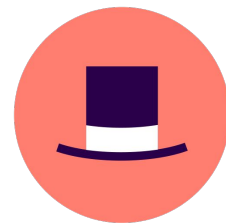
MLLM (Maui-like Lexical Matching) is inspired by Maui, an older lexical automated subject indexing tool, but it is implemented in Python within Annif.

# Define MLLM project in projects.cfg



```
[yso-mllm-en]  
name=YSO MLLM project  
language=en  
backend=mllm  
vocab=yso-en  
analyzer=snowball(english)
```

```
[stw-mllm-en]  
name=STW MLLM project  
language=en  
backend=mllm  
vocab=stw-en  
analyzer=snowball(english)
```



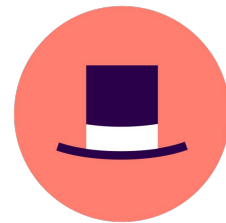
# Train MLLM on fulltext documents

MLLM requires a relatively small number (hundreds or at most a few thousand) of training documents.

The training should be done with similar documents to those we will use in testing. We will select 400 documents from the train folder for this purpose:

```
# Command for yso-nlf  
annif train yso-mllm-en --docs-limit 400 data-sets/yso-nlf/docs/validate/
```

```
# Command for stw-zbw  
annif train stw-mllm-en --docs-limit 400 data-sets/stw-zbw/docs/validate/
```



# Testing and evaluating the MLLM project

1. Test on example documents:

a. Using one sentence:

```
echo "This is an example." | annif suggest PROJECT_ID
```

b. Using a text file:

```
annif suggest PROJECT_ID <FILE.TXT
```

2. Evaluate on a corpus: `annif eval PROJECT_ID path/to/corpus`