Yaml Based Exams Documentation

Release 0.2.0

Robbert Harms

May 24, 2020

Contents

CHAPTER 1

Introduction

Ybe is a software package supporting a YAML based file format for importing, exporting and storing exams in a plain text file. It supports multiple-choice, multiple-response and essay questions. Due to the extensive meta-data storage, storing exams in ybe allows you to easily search, annotate and recombine questions into new exams. Exams can be written as a LaTeX file, or be exported to the QTI format and be imported by Canvas and other educational software.

1.1 Examples

Questions can be stored in a plain text file using YAML for structure and Markdown for the content of the questions.

1.1.1 Minimal .ybe file

For example, a minimal example of a multiple choice question is given by:

```yaml
ybe_version: 0.2.0

questions:
- multiple_choice:
    id: q1
    points: 1
    text: Example multiple choice question.
    answers:
        - answer:
            text: First answer
        - answer:
            text: Second answer
            correct: true
```

This defines a list of questions with only one question. The `id` is meant to be provide a unique identifier to every question and should be unique for every question in an .ybe file. The points define the worth of the question. The text is in Markdown format and allows all Markdown operators. The answers are again a list with Markdown text blocks and a marked correct answer.

1.1.2 Exporting to QTI

If you would copy the previous Ybe content into a text file named `example.ybe`, you could export it to a QTI using:

```
from ybe import read_ybe_file
from ybe.lib.qti_writer import \
    write_qti_zip, \
    ConvertCanvasEquations

ybe_exam = read_ybe_file('example.ybe')

# QTI with Canvas style equations
write_qti_zip(ybe_exam, 'qti_canvas.zip',
              text_formatter=ConvertCanvasEquations())
```

1.1.3 Exporting to LaTeX

Alternatively, you could output your exam to a Latex file read for printing:

```
from ybe import read_ybe_file, write_latex_file

ybe_exam = read_ybe_file('example.ybe')
write_latex_file(ybe_exam, 'main.tex')
```

1.1.4 Supported question types

An example of an ybe file with all supported questions and some file meta data is given by:

```
ybe_version: 0.2.0

info:
    title: Example questions
    description: Example of all questions.
    document_version: 0.1.0
    date: 2020-05-24
    authors:
        - The Author

questions:
- multiple_choice:
    id: q1
    points: 1
    text: Example multiple choice question.
    answers:
        - answer:
            text: First answer
        - answer:
            text: Second answer
            correct: true

- open:
    id: q2
    points: 3
    text: Example open question.

- multiple_response:
    id: q3
    points: 2
```

(continues on next page)

(continued from previous page)

```yaml
    text: A multiple response question is a
        multiple choice question, where
        multiple answers are possible.
    answers:
        - answer:
            text: First answer
            correct: true
        - answer:
            text: Second answer
        - answer:
            text: Third answer
            correct: true
        - answer:
            text: Fourth answer

- text_only:
    id: q4
    text: |-
        Since "text" is short for "text_markdown",
        you can use Markdown syntax to markup
        your document.

        For example:

        1. this is a list
        2. *with this in italics*
        3. **and in bold**

        This is a famous formula inline: $E=mc^2$
        and this is a basic displayed formula:

        $$ a^2 = b^2 + c^2 $$
```

1.1.5 Adding meta-data

In addition, Ybe supports adding meta-data to your questions. This allows you to store a description of the question, the lifecycle (like, who made this question), a classification of for example the skill level and question analytics where you can store statistics of past usage of this question. This can all be stored as follows:

```yaml
questions:
- open:
    id: q5
    points: 1
    text: Example with meta data
    meta_data:
        general:
            description: Some description for yourself
            keywords: [alpha, beta]
            language: en
        lifecycle:
            author: The Author
        classification:
            skill_level: Knowledge
            related_concepts: [Ybe]
            module: Science
            chapter: 1
            difficulty: 1
        analytics:
            - exam:
```

(continues on next page)

(continued from previous page)

```
        name: 2020_qz1
        participants: 1
        nmr_correct: 0
      - exam:
        name: 2020_qz1
        participants: 200
        nmr_correct: 25
```

1.1.6 Searching your questions

If you would save the above in a file `example.ybe`, you could then search through the questions easily. For example, finding all questions that yield exactly one point can be done like:

```
from ybe import read_ybe_file

ybe_exam = read_ybe_file('example.ybe')

for question in ybe_exam.questions:
    if question.points == 1:
        print(question)
```

1.1.7 Importing from QTI

If you already have questions in Canvas or other software packages, you could export these to QTI format and convert easily into an .ybe file:

```
from ybe import read_qti_zip, write_ybe_file
from ybe.lib.utils import copy_ybe_resources

ybe_exam = read_qti_zip('qti_file.zip')

# write the ybe file
write_ybe_file(ybe_exam, './qti_to_ybe.ybe')

# and write the images referred to in the QTI
copy_ybe_resources(ybe_exam, './')
```

1.2 Summary

In general:

- Storing exams in a plain-text `.ybe` file
- Importing and exporting to and from QTI
- Write exams to LaTeX
- API for scripting exams

Technical details:

- Free software: GPL v3 license
- Full documentation: https://ybe.readthedocs.io
- Project home: https://github.com/robbert-harms/ybe

1.3 Quick installation guide

Ybe requires Python 3.8+. Either use your package manager, or install a Python distribution like Anaconda. After that it is typically as simple as:

```
pip install ybe
```

Linux

For Ubuntu 18.xx you need to install Python 3.8 first, for example see here: https://linuxize.com/post/how-to-install-python-3-8-on-ubuntu-18-04/. Afterwards, simply install using:

```
pip3 install ybe
```

For other Linux distributions the setup is typically similar, install Python 3.8 and then install ybe.

Windows

- Install Anaconda Python 3.8
- Open an Anaconda shell and type: `pip install ybe`

Mac

- Install Anaconda Python 3.8
- Open an Anaconda shell and type: `pip install ybe`