Zurich University
of Applied Sciences

School of
Engineering
InIT Institute of Applied
Information Technology

# Project work (IT)

## Creating a data repository for language corpora with CKAN

| | |
|---|---|
| **Author** | Jan Albert<br>Kreshnik Sadriu |
| **Main supervisor** | Fernando Benites de Azevedo e Souza<br>Mark Cieliebak |
| **Date** | 20.12.2019 |

Zurich University
of Applied Sciences

School of
Engineering

# DECLARATION OF ORIGINALITY

## Project Work at the School of Engineering

By submitting this project work, the undersigned student confirm that this work is his/her own work and was written without the help of a third party. (Group works: the performance of the other group members are not considered as third party).

The student declares that all sources in the text (including Internet pages) and appendices have been correctly disclosed. This means that there has been no plagiarism, i.e. no sections of the project work have been partially or wholly taken from other texts and represented as the student's own work or included without being correctly referenced.

Any misconduct will be dealt with according to paragraphs 39 and 40 of the General Academic Regulations for Bachelor's and Master's Degree courses at the Zurich University of Applied Sciences (Rahmenprüfungsordnung ZHAW (RPO)) and subject to the provisions for disciplinary action stipulated in the University regulations.

City, Date: Signature:

..... .....

.....

.....

The original signed and dated document (no copies) must be included after the title sheet in the ZHAW version of all project works submitted.

## Abstract

The data scientists analyse a lot of data for various projects. These files however, are stored on various hard drives and on local computers. This makes accessing a certain dataset cumbersome and wastes a lot of time. Because of this we received the task to create an online data repository for the InIT department of the ZHAW. This project is relevant for system administrators who want to store large quantities of data and extend the program with custom extensions with the help of the open-source program CKAN. This program was set up through an external server which was provided to us. Initially, we compared three different data-repository platforms and concluded that CKAN was the one most suitable. Consequently, we set up CKAN with the help of Docker because there were some compatibility issues regarding the Ubuntu version we were using and the newest stable version of CKAN. Since you cannot create custom plugins with the Docker version of CKAN, we decided to try the source installation regardless of issues since there was evidence that you could get it to work with some tinkering. Thus we installed the source version with some minor tweaks. We concluded that the source installation of CKAN is better to use because of its ease of adding new plugins and its ability of creating new ones, even though it requires more effort to get the base program running.

# Contents

# 1 Introduction

## 1.1 Starting Point

There are several sources from which we were able to build upon. Firstly, there was a paper written regarding the comparison of the most popular data repository programs on the market. Unfortunately, we only came across said document after we already made our choice of which repository to build upon. [1]
Secondly, CKAN has a thorough documentation and an active forum. With the help of these resources, we could circumvent the most common errors we encountered. Especially the documentation was extremely helpful, since most the guides were written in an easy to follow step-by-step guide. They even had a detailed guide on how to extend this open-source program to support custom attributes.

## 1.2 Objective

Our goal of this project was to create a data repository for Speech & Text corpora. Currently the ZHAW InIT already has the data, but they are stored on several hard drives spread all over the office. This data repository should be accessible through an online website. When uploading a corpus, you should be able to add a licence to it. A user must also be able to add tags to a corpus, so you can filter and search for specific corpora. Since the website will be hosted on a server from the ZHAW InIT, only people within the network are able to access the repository. However, an authorization system with basic read/write permissions is crucial in order to prevent unauthorized access. Corpora can each contain thousands of files, totalling over 20GB. Therefore, one must be able to upload and download all of the data at once. A preview was also requested, so you do not have to download the data in order to view it. From the administrative side, we needed a backup system in case our server gets destroyed. Another feature which would be nice to have, is a connection with the existing LDAP of the ZHAW, so everyone can use their same login as on all the other websites from the school.

Here you can see a list of what we were tasked with and what we were able to achieve. Most of the tasks we were able to complete, but some were only completed partially or not at all due to time constraints.

| Task | Completed |
|---|---|
| **Setup server with CKAN** | YES |
| **Upload test data** | YES |
| **Display licensing for each corpus** | YES |
| **Have a backup system in place and make sure it works** | NO |
| **Create a virtual link to the 2.7 TB hard drive** | YES |
| **Ability to label corpuses with tags** | YES |
| **Ability to search corpuses with tags** | YES |
| **Display a preview to show a sampling of a corpus without needing to download it** | PARTIALLY |
| **Add a readme functionality to see what the corpus is about** | YES |
| **Create an authorisation system where only certain users can access certain data** | YES |
| **One should be able to upload a folder, so that you don't need to add everything separately** | YES |
| **Display corpus size (Amount of Tokens for Text / Amount of Hours for Audio) & Language** | YES |
| **Add a 'download all' functionality** | PARTIALLY |
| **Connect the login of CKAN with the LDAP of the ZHAW** | NO |
| **Adjust the website design** | YES |
| **Create a script in order to automatically start Docker -> deploy source install** | YES |
| **Operate derivate from corpuses(When creating a new version of a corpus you should see where it is copied from and if there any other versions, like a version tree)** | NO |

# 2 Procedure

## 2.1 Gathering Requirements

Before we could start working on the project, we needed to contact several people from the ZHAW InIT, in order to see what they would like the data repository to look like. We also categorised the data analysts into two groups, the text scientists and the audio scientists, and asked them separately what they wished from the website. Firstly, we contacted a research assistant from the text team. They had three main wishes for the project:

1. Firstly, the data repository should be accessible through a web browser. This was no problem, since all three options which were available to us are already using a website to access the contents of the repository.
2. Secondly, corpora should be able to be filtered and be searched by tags. Pius mentioned this will be especially useful if you need to train a classifier with certain data (for example “Hate speech” and “Politics”), since you can just filter for these tags and get them right away instead of searching for them manually.
3. Lastly, the files from the corpus should have a preview, so that you do not need to download the whole corpus. Pius mentioned that this would be a nice-to-have feature. The corpora can be over 20GB each and if you download a whole corpus only to find that it is not what you were looking for, then you just wasted a lot of time. A preview of files would prevent this.

Afterwards we made an appointment with the audio team. We spoke with researchers who specialise in audio. They also had three demands for the project which they wanted us to implement:

1. Firstly, the corpus should display how big it is in terms of length. An audio corpus should display the amount of hours of data there is, while a text corpus might display the amount of lines in a corpus.
2. Secondly, there should be a sampling for audio corpora. This is almost the exact same thing as the preview which Pius mentioned. However, in this case we must only load a portion of the audio file. Otherwise the browser will be overwhelmed by all of the audio data, since these are usually much bigger in terms of size than text files.
3. Lastly, a corpus should have a readme file that contains the most important information, like what the corpus is about.

After the two meetings we met with the project owners and they gave us some extra requirements. These were not major additions to the project, but instead vital components which are needed in order to make the data repository usable. They wanted a backup system in case the server dies, a licensing field where the licence of the corpus is displayed and an authorisation system where you can give or revoke permissions of a specified user. They also wanted a connection the LDAP login of the ZHAW so that you do not need to create an account for everyone at the ZHAW.

## 2.2 Timetable

At first, we spent a lot of time researching the different data repositories and gathering all of the requirements for what people were expecting from the project. After the two meetings, we decided to go with CKAN since it looked like it fits the needs of the data scientists the most.

After some tinkering with CKAN, we finally managed to get it to work on Docker. We installed some plugins and debugged them in order to get most of them to work. After two reinstallations because we wanted to hand in a clean installation of the program, we were tasked with installing CKAN with source, since there was a specific feature which required this installation.

On the next page you can see our schedule on how we were going to achieve our goals, based on the time we had available. Each task has a priority, where 0 represents the most important feature and 6 represents features, which are so called nice-to-have features. We also assigned each goal a Date, on which we were planning on having it completed. The dates matched up with our weekly meeting with the project owners. Furthermore, we created a table which represented our estimated time we needed to invest in order to get said task to work. This number is arbitrary and is meant to be a relative indicator, similar to Scrum.

| Task | Priority | Type | Week | Date | Effort |
|---|---|---|---|---|---|
| Setup server with CKAN | 0 | Administrative | 45 | 04.11.2019 | 32 |
| Upload Testdata | 1 | Data | 45 | 04.11.2019 | 16 |
| Display Liscensing for each Corpus | 1 | Feature | 45 | 04.11.2019 | 4 |
| Have a Backup System in place and make sure it works | 1 | Administrative | 46 | 11.11.2019 | 4 |
| Create a virtual Link to the 2.7 TB Harddrive | 1 | Administrative | 46 | 11.11.2019 | 1 |
| Adjust localhost redirect to the actual Website | 1 | Administrative | 46 | 11.11.2019 | 2 |
| Labeling the Corpuses with Tags | 2 | Feature | 45 | 04.11.2019 | 4 |
| Searchable with Tags | 2 | Feature | 45 | 04.11.2019 | 4 |
| A Preview to show a sampling of a Corpus without needing to download it | 2 | Feature | 47 | 18.11.2019 | 16 |
| Add a Readme functionallity to see what the corpus is about | 2 | Feature | 45 | 04.11.2019 | 4 |
| Upload Real Data from the ZHAW (First few Datasets) | 2 | Data | 48 | 25.11.2019 | 32 |
| Create a Authorisation system where only certain users can access certain data | 2 | Administrative | 47 | 18.11.2019 | 8 |
| One should be able to upload a folder, so that you don't need to add everything seperately | 2 | Feature | 51 | 16.12.2019 | 8 |
| Display Corpus Size (Amount of Tokens for Text / Amount of Hours for Audio) & Language | 2 | Feature | 51 | 16.12.2019 | 4 |
| Add Download all functionality | 3 | Feature | 48 | 25.11.2019 | 16 |
| Connect Login with LDAP (Remo Maurer) | 3 | Administrative | 49 | 02.12.2019 | 8 |
| Adjust Websize Design (CSS, ZHAW Logo, etc.) | 3 | Design | 49 | 02.12.2019 | 1 |
| Create script in order to automatically start Docker -> deploy source install | 3 | Administrative | 49 | 02.12.2019 | 8 |
| Operate derivate from corpuses | 6 | Feature | 51 | 20.12.2019 | 16 |

**DEADLINE [20. December 2019]**

## 2.3 Comparing CKAN, DSpace, the Dataverse Project

While searching for a solution for our project we had found three different Filesystem programs which could meet our needs. Each of these programs had their strengths and weaknesses. We are going to show you these three options.

### 2.3.1 Dspace

Dspace is an open source program. It has an active community. However, Dspace has a rather complicated guide. May it be for installation or the everyday use. [8]

### 2.3.2 Dataverse

Dataverse is an open source program created by Harward University. Dataverse also has good instructions sites which seems to be helpful. In addition, support from Harward is quick and supportive. However, the data limit is set to 1 TB and cannot be expanded. [9]

### 2.3.3 CKAN

CKAN is also an open source program. There is a big community that has regular updates for CKAN. Like the other two programs, CKAN can also be expanded with extensions. With their simple and simple instructions, it is possible to carry out installations and administration without problems. Furthermore, CKAN has no data limit. [7]

### 2.3.4 Why we choose CKAN

All three programs had their advantages and disadvantages. However, we had to choose a program and the decision fell on CKAN. The decision can be explained with the following points. CKAN has a very large community compared to Dataverse & Dspace. The documentation is kept simple for the size of the program and works for the most part. For this purpose, CKAN is continuously updated in regular updates and is widely distributed. It is also possible to create your own plugins with CKAN and thus respond to the needs of the users. Finally, CKAN has no data limit or data limit.

## 2.4 What is Docker?

Docker is a software which can simplify the application delivery with its containers. With this the necessary packages will be contained and become easily transportable and installed as files. The containers ensure that the used resources on one computer are separated and managed individually.

## 2.5 Setting up CKAN with Docker

At first the only real solution to install CKAN, was to set it up with Docker. Since the latest stable version of CKAN (2.8) dates back to 2017, it does not support Ubuntu 18.04, the latest Ubuntu version with all of the newest security patches. Docker allows backwards compatibility because it runs all of the different parts of a program in so called containers, which run separate from each other. Installing CKAN through Docker was in hindsight not very difficult. We encountered several errors, but most of them were due to our own mistakes. If you miss one command in the installation guide, then a few commands down the line, you will be wondering why that one command produces an error while the previous ones worked without problem. There was one bug which we encountered through a separate issue.

When we first were running the server, we could not access the website. “Connection timed out” was the only message we received, even though we were confident that CKAN was running. This was because the firewall blocked all outside access, even within the network. This was confirmed by executing

```
curl http://160.85.252.207:5000/
```

on the server. There we got back a HTML-page since the server does not block itself with the firewall. After we configured the firewall, we gained access to the CKAN website.

From then on we only had one major bug. Every link of the website redirected to your own localhost instance. To fix this, you need to go to the production.ini file with a text editor of your choice and edit the CKAN_URL to be the same as the IP-Address of the server.

We needed a lot of help from our project leader, since we were not familiar with Docker at all. But the more we worked with Docker, the more confident we got with it. The first installation took us two weeks, while the second installation after wiping all of the Docker images took us merely an hour.

Every Monday, when the servers of the ZHAW InIT usually restart, we could not start CKAN. As soon as you restart the docker-compose, everything will restart completely fine except the datapusher. The datapusher cannot be started, because the port which it uses, is already occupied (in this case by apache2).

```
ERROR: for datapusher  Cannot restart container
ba55ee05703a2bf61540955390e78cb2 901dee4f477834e6f488b0c14661fc0b:
driver failed programming external connectivity on endpoint
datapusher
(b58cbdaa1e189e922be723ae7b2c66b53616f0ef7c37ccbb9d868c9c7c3ec8e2):
Error starting userland proxy: listen tcp 0.0.0.0:8800:bind: address
already in use
```

In order to resolve this issue, we terminated the process which runs on the occupied port in order to free it so that the datapusher can access it.

```
ubuntu@data-repo:~$  sudo kill $(sudo lsof -t -i:8800)
```

After the process had been terminated, we were able to restart the docker-compose.

```
cd ckan/ckan/contrib/docker
docker-compose restart
```

Afterwards CKAN was successfully running on port 5000.

## 2.6 Displaying Size of Corpora

When you search for corpora, you want to see how big they are before downloading them. Otherwise you may be faced with several Gigabyte worth of data when you only needed a fraction of it. In order to achieve this, we needed to add a something like a custom field / attribute which you would be able to search for or filter.


CKAN already has custom fields, but they are missing the search functionality. Furthermore, we don't want the user to enter the 'Key:' field every time they upload a dataset. After some research, we stumbled upon a guide on how to add extensions to CKAN. It perfectly described the exact thing we needed. However, in order to create a custom extension, you need a source installation of CKAN. [2]

Since we used Docker in order to setup the program, we would need to install a separate instance of source CKAN. Unfortunately this is not possible with the server which was given to us, since the server runs on the latest Ubuntu version (18.04) whereas CKAN requires Ubuntu 16.04 64-bit or Ubuntu 14.04 64-bit [3].

However after some research we found a comment that indicated that you could get CKAN source to work. [5]

In the end, we successfully installed CKAN source, as documented in the point "Setting up CKAN with source". [4]

## 2.7 Used Plugins

The following plugins are installed per default in CKAN:

- stats
- text_view
- image_view
- recline_view

It is important to note, that the text_view plugin doesn't work as intended on the Docker installation. You need to comment out the ckan.preview.text_formats line and remove text_view from the plugin list in order for the files to fix the problem. This displays the text-files in the default preview.

The following plugins were added manually by editing the production.ini file:

- datastore
- datapusher
- file_uploader_ui
- resourceauthorizer
- downloadall
- pdf_view
- resource_proxy

In order to add a new plugin, one must first activate the CKAN virtual environment.

```
ubuntu@data-repo:~$  source /usr/lib/ckan/default/bin/activate
```

Afterwards you can install the plugin using pip, a package manager for python with the following command.

```
(active) ubuntu@data-repo:~$    pip install [PLUGIN_NAME]
```

Afterwards, you must locate the config file and add the plugin to the list of used plugins. The config file should already be defined in the environment variables which were setup during the basic installation. These commands will open the config file with the vim editor. If you wish to use another editor like nano or emacs, just replace "vim" with your editor of choice.

```
ubuntu@data-repo:~$  export VOL_CKAN_CONFIG=`docker volume inspect
docker_ckan_config | jq -r -c '.[] | .Mountpoint'`
ubuntu@data-repo:~$  sudo vim $VOL_CKAN_CONFIG/production.ini
```

There you can find a list of all plugins which are currently in use. You just have to add the name of the plugin to the ckan.plugin list and restart CKAN. Afterwards, you

should be able to see the changes on the website. The order of these plugins does not matter in the production.ini file, but the names are case sensitive, so you have to make sure that you spelled the plugin name correctly. Otherwise it will not work.


### 2.7.1 datastore and datapusher

Of the plugins which we added, datastore and datapusher were the most important ones, since they were required to be added during the CKAN installation. The step-by-step guide can be found under the third point of the “Installing CKAN with Docker Compose” guide [3].

### 2.7.2 file_uploader_ui

After finishing the basic CKAN setup, the first plugin we added, was the file_uploader_ui. In order to upload more than one file, you need to manually add each and every document to the server. This is extremely inefficient, especially since the main purpose of our site is to host corpora, some of which contain thousands of document totalling upwards of 20GB. The file_uploader_ui plugin adds a separate interface to the left of the regular file uploader. This enables you to add several files at once to a corpus, while also being a drag and drop field for further quality of life improvements. [6]

## 2.8 Create Symlink

The server we were given has 9 partitions. Most of them are between 5MB and 100GB. One partition however has a total of almost 4 TB. Due to the sheer volume of data which will be uploaded, it is not sustainable to store everything on a small storage space. We can create a symlink to combat this. A symlink is like a pointer from one storage space to another. Once you link these two destinations up, the system will think that it is still in the original directory, while in actuality you are on the large hard drive. To link up the different directories, you need to first find out which partition is suitable for storage. The command to list all partitions is as follows:

```
ubuntu@data-repo:~$  df –h
```

You should see a table with the partition, the amount of space on it and its respective directory. You should pick out the partition and remember the file path. Afterwards, you need to find out where the data is currently stored. There are two ways on how you can find out this information. Firstly, you can list all partitions, upload a large test corpus and list all of the partitions again. You should see that the storage of one of the partitions dropped. This is the partition from which you want to create a symlink from. Secondly, you could just check in the production.ini file where the data will be stored. There should be an environment variable called ckan_storage_path which will also take you to the directory from which you want to create a symlink.

In order to actually create a symlink, you must execute the 'ln' command. The first directory is the one where the metadata is currently stored and the second one is where it should end up in order to save storage.

```
ln -s /var/lib/docker/volumes/docker_ckan_storage/ /srv/data_repo/
```

If you want to test to see if it worked, you can always list all of the partitions again, upload test data and check the storage space on the big hard drive again. If it got smaller, then you successfully manage to create a symlink.

## 2.9 Setting up CKAN with source

Problem:

The installation of CKAN from source was necessary after the failed implementation of a self-implemented plugin via Docker. Because of that we had to de-install Ckan & Docker completely. Any remnants of Symlinks regarding the Docker installation had also to be deleted for CKAN Source to work. Without this action we would risk problems in the future.

The installation of CKAN with source wouldn't be a problem since we could use the installation guide [3]. However, since we are using Ubuntu 18.04 and not 16.04 some problems came up.

Solution:

Before we could begin the installation it was crucial that the python version is compatible with Ubuntu 16.04 installation guide. Which means that we had to downgrade the python version from 3 to 2.

```
ubuntu@data-repo:~$ python -V
Python 2.7.15+
```

*Figure 1 Command to check which version of python is installed / used*

### 2.9.1 CKAN activation

After this we continued the installation guide until we arrived at 2a. The first two commands had been implemented but the third command had to be changed to **sudo python 2 `which virtualenv` --no-site-package /usr/lib/ckan/default**. This is necessary to implement the correct virtualenv. Afterwards the last command must have been running if we wanted to install CKAN further.

```
ubuntu@data-repo:~$ . /usr/lib/ckan/default/bin/activate
(default) ubuntu@data-repo:~$
```

*Figure 2 Command to activate the virtualenv*

It is crucial that this command was active when we wanted to install anything CKAN related. Without it CKAN won't have acknowledged any input from us. We could continue to 2c. To have a stable program we didn't want to use the latest version of CKAN and copied the command and changed the version to 2.8.1.

```
pip install -e 'git+https://github.com/ckan/ckan.git@ckan-
2.8.1#egg=ckan'
```

This was needed to implement the self-programmed plugin for CKAN.

We could follow the installation until 4. The ckan.site_id could be ignored however, we had to change the site_url in **etc/ckan/default/development.ini** to http://160.85.252.207:5000

```
ckan.site_url = http://160.85.252.207:5000
```

*Figure 3 Needed input for ckan.site_url*

### 2.9.2 Jetty Problem

Afterwards the installation of Solr had to be done. The problem with Solr comes with the different version of jetty (jetty9 instead of jetty/jetty8). Jetty9 does not look for the symlink created by the solr package. In order to fix this, you just need to create an additional symlink to the destination that solr wants to point to.

```
ubuntu@data-repo:~$  sudo ln -s /etc/solr/solr-jetty.xml
/var/lib/jetty9/webapps/solr.xml
```

Because of that we had to rely from then on forward on the documentation of the latest CKAN version [3]

To make Solr work we had to follow step 5 completely from the latest installation guide. Without this deviation CKAN won't work. Afterwards we went back to the 2.8.2 guide and followed it to the end.

### 2.9.3 Deployment

As for now CKAN can be started with the paster command from the guide. But this wasn't enough for us. We wanted to deploy the source installation. With the deployment guide we managed to leave the paster command behind and exchange it for the apache2 route. But there are still somethings that needs to be tweaked. First of all we need to allow the port 8080. After that we can follow the instructions until 6. Here we must add a some extra information in /etc/apache2/sites-available/ckan_default.conf .
<VirtualHost *:8080> The first line must be changed from 127.0.0.1:8080 to *:8080. With this the apache config files works. After that the guide can be followed until the end. [10]

## 2.10 Difference between Docker and Source

The product of CKAN Docker and Source is the same. Both show the corpuses and filters in the same way. However, there is a difference in installation and maintenance between these two.

With Docker the installation has been straight forward. We could follow the installation guide to the point and have a working product in the end. Surely there were some problems on the line. These had been corrected that CKAN can work. However, the maintenance of CKAN seemed more complicated than anticipated. Because the whole CKAN program was divided in five containers further installations were complicated. Which means that adding plugins became a great hurdle instead a help.

The Source installation on the other side has been complicated. That may have been because we had a different Linux Ubuntu version than recommended. For the reason that Mr Fernando Benites has insisted that our Linux installation must be up to date since it is the ZHAW INIT policy to do so. This has caused problems in the installation department. After this installation any plugins has been installed directly in CKAN without any need to change Docker containers. This has allowed us to shorten the installation and debug time of self-programmed and imported plugins greatly.

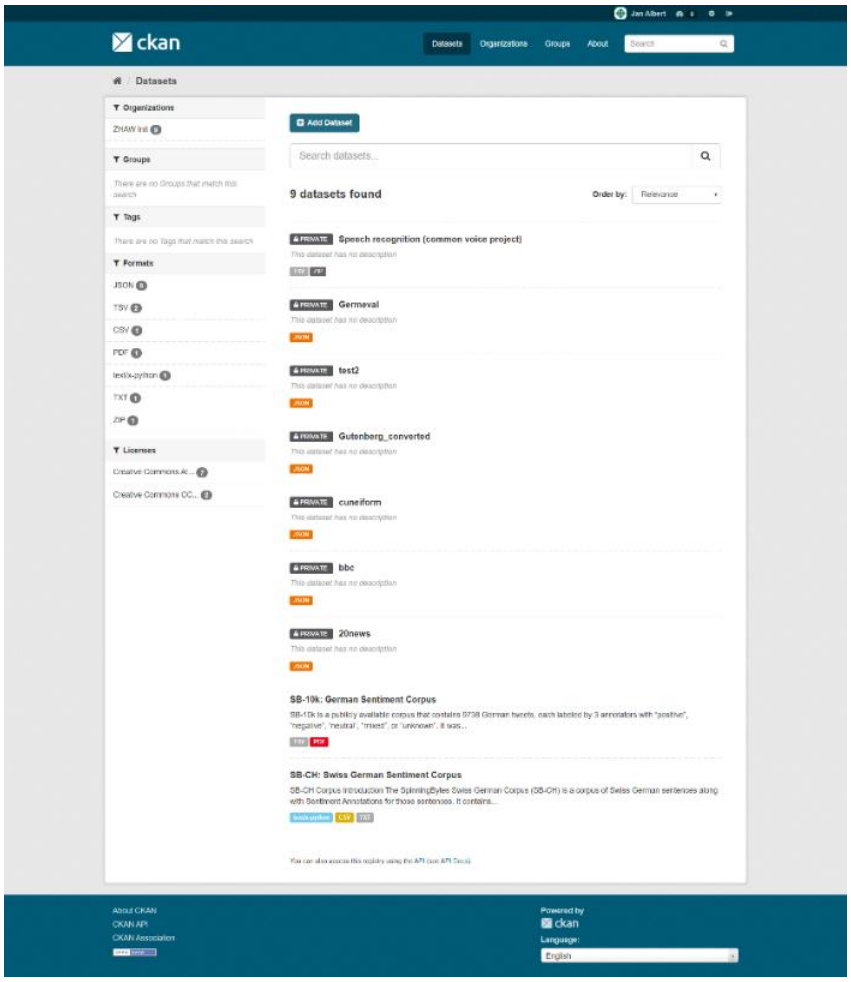
Our conclusion can be summarized in two sentences. While the Docker installation has been rather easy the maintenance has been the exactly opposite. The Source installation has been the rather hard but the maintenance and adding new plugins has never been better.

# 3 Results

The result of our project, is a fully functional website, where you can store data. Anyone who is connected to the network can create an account and view datasets. If there are certain datasets that you do not want others to view, you can set the visibility of the corpus to be only viewable by a select group of people. The administrator can set these permissions in a menu on CKAN, which is only visible to them. Users can also have permissions, which can be set to either ‘read’ or ‘read and write’. There also exists a group feature which allows you to easily manage the permissions of a team, by adding them to a so called organisation.

With the help of plugins, we also managed to easily streamline the way you handle data. You can upload a bunch of data at once with the file_uploader_ui plugin. We even managed to create a custom extension. This extension enables you to add custom field to the corpus, in our case “size”, and allows you to sort by it as well. Furthermore, we analysed the current tool and evaluated, how exactly the tool could be extended. Be it either through plugins, or manual modifications to the source code.

A screenshot of the final result can be seen on the right.



# 4 Discussion with Outlook

The installation and configuration of CKAN has been a success. Most of the tasks have been implemented and are working. But that is only the beginning. Since CKAN is an open-source program, there are no limits to expand it with plugins. May they be self-written or imported. Here are some ideas for the future.

## 4.1 Great Filtering

Now CKAN has a filter system which sadly isn't too strong. It may be possible to create a one which can filter through zip files. One can go even further and search for words in files instead just searching for the filenames.

## 4.2 Cloning and linking between Corpuses

At the moment CKAN has the plugin to clone corpuses. However, the cloned corpuses don't have a linked connection to the original. It may be possible to connect these two. With this different version of corpuses can exist without confusing the user since parent & child relation can be observed rather easily then.

## 4.3 Increase data throughput

The data throughput right now is acceptable. However, with the right plugin it may be possible to increase the data throughput tenfold in which CKAN can do right now.

## 4.4 LDAP Extension

Since the ZHAW has a LDAP environment the adding of LDAP Extension seems logical. With this the users of ZHAW InIT can work with their given accounts from the University of Applied Sciences.

## 4.5 GUI Update

Right now, CKAN has the default GUI installed. There are two possibilities to change that. One can program with HTML and JS a new frontend or can import one from the CKAN extensions. An example would be the website from opendata (opendata.admin.ch). Beware since some extension haven't been updated for a long time.

## 4.6 Audio preview

CKAN can preview some text files and PDFs. As for now the ability to preview audio files isn't one of them. The CKAN team is trying to implement a plugin as for right now. This plugin can shorten the search time for the correct audio file without downloading and opening every file.

### 4.7 Commenting datasets / corpuses

It would be helpful if users can comment the datasets in CKAN. With this the user can ask for updates or asses the datasets.

### 4.8 Adding CMS Page plugin

At the moment we can either hardcode HTML5 & JS or import a new template. However, we could install a basic CMS Page plugin which makes the editing easier and less stressful.

### 4.9 Version control

It would be useful if CKAN had functioning version control system installed. Because as of right now to update data you must delete the old input for the new. This could be extremely helpful if users want to use the old data and not the new ones.

### 4.10 Download all

As for now CKAN doesn't have the ability to download all data from a datasets to a single zip file. With this ability the users won't need to download each data by hand.

# 5 Register

Below you can find a list of references for the project. The first date after the author indicates the last time that the page had been updated, while the date at the end indicates the most recent time we accessed the source.

[1] Amorim, Ricardo & Aguiar Castro, João & Rocha, João & Ribeiro, Cristina. (2016). A comparison of research data management platforms: architecture, flexible metadata and interoperability. Universal Access in the Information Society. 16. 10.1007/s10209-016-0475-y.

[2] CKAN Association. (2019, November 21). Customizing dataset and resource metadata fields using IDatasetForm [Online]. Available: https://docs.ckan.org/en/2.8/extensions/adding-custom-fields.html [09.12.2019]

[3] CKAN Association. (2019, September 6). Installing CKAN from Source [Online]. Available: https://docs.ckan.org/en/2.8/maintaining/installing/install-from-source.html [09.12.2019]

[4] CKAN Association. (2019, September 6). Installing CKAN with Docker Compose [Online]. Available: https://docs.ckan.org/en/2.8/maintaining/installing/install-from-docker-compose.html [09.12.2019]

[5] CKAN Association. (2019, July 22). Is it possible to install ckan on ubuntu 18.04? [Online]. Available: https://github.com/ckan/ckan/issues/4311 [09.12.2019]

[6] Ori Hoch. (2018, October 17). ckanext-file-uploader-ui 0.1.4 Project description [Online]. Available: https://pypi.org/project/ckanext-file-uploader-ui/ [09.12.2019]

[7] CKAN Association. (2019, December 9). CKAN, the world’s leading Open Source data portal platform [Online]. Available: https://ckan.org/ [09.12.2019]

[8] LYRASIS. (2019, December 9). DSpace [Online]. Available: https://duraspace.org/dspace/ [09.12.2019]

[9] Harvard's Institute for Quantitative Social Science (IQSS). (2019, December 9). The Dataverse Project Open source research data repository software [Online]. Available: https://dataverse.org/ [09.12.2019]

[10] CKAN Association. (2019, September 6). Deploying a source install [Online]. Available:https://docs.ckan.org/en/2.8/maintaining/installing/deployment.html [19.12.2019]