03_convert

May 26, 2024

1 Convert Text to IDs

1.1 Content

1. Import data
2. extract data and add IDs
3. save to csv

[14]:
```
# imports
import pandas as pd
from db import get_database
```

Import data

[15]:
```
# get data from mongodb
dbname = get_database()
collection = dbname["joined"]

documents = collection.find()
df = pd.DataFrame(list(documents))

df.head(2)
```

[15]:
```
                        _id display  \
0  6649fb988a508636be6bfa35    long
1  6649fb988a508636be6bfa36    long

                                           occupation  \
0  {'code': '27-2011.00', 'title': 'Actors', 'tag…
1  {'code': '23-1021.00', 'title': 'Administrativ…

                                               tasks  \
0  {'task': [{'id': 7646, 'green': False, 'relate…
1  {'task': [{'id': 7627, 'green': False, 'relate…

                                    technology_skills  \
0  {'category': [{'related': 'https://services.on…
1  {'category': [{'related': 'https://services.on…
```

```
                                          tools_used  \
0  {'category': [{'related': 'https://services.on…
1  {'category': [{'related': 'https://services.on…

                                     tools_technology  \
0  {'tools': {'category': [{'related': 'https://s…
1  {'tools': {'category': [{'related': 'https://s…

                                           knowledge  \
0  {'element': [{'id': '2.C.7.c', 'related': 'htt…
1  {'element': [{'id': '2.C.8.b', 'related': 'htt…

                                              skills  \
0  {'element': [{'id': '2.A.1.a', 'related': 'htt…
1  {'element': [{'id': '2.A.1.b', 'related': 'htt…

                                           abilities  …  \
0  {'element': [{'id': '1.A.1.a.3', 'related': 'h…  …
1  {'element': [{'id': '1.A.1.b.5', 'related': 'h…  …

                                        work_context  \
0  {'element': [{'id': '4.C.1.b.1.e', 'related': …
1  {'element': [{'id': '4.C.2.a.1.a', 'related': …

                                           job_zone  \
0  {'value': 2, 'title': 'Job Zone Two: Some Prep…
1  {'value': 5, 'title': 'Job Zone Five: Extensiv…

                                           education  \
0  {'level_required': {'category': [{'name': 'Les…
1  {'level_required': {'category': [{'name': 'Doc…

                                           interests  \
0  {'element': [{'id': '1.B.1.c', 'related': 'htt…
1  {'element': [{'id': '1.B.1.f', 'related': 'htt…

                                        work_styles  \
0  {'element': [{'id': '1.C.3.a', 'related': 'htt…
1  {'element': [{'id': '1.C.5.c', 'related': 'htt…

                                        work_values  \
0  {'element': [{'id': '1.B.2.d', 'related': 'htt…
1  {'element': [{'id': '1.B.2.a', 'related': 'htt…

                                related_occupations  \
0  {'occupation': [{'href': 'https://services.one…
1  {'occupation': [{'href': 'https://services.one…
```

```
                                  additional_information isco08       Name_de
0  {'source': [{'url': 'https://www.actorsequity…   2655  Schauspieler
1  {'source': [{'url': 'https://www.americanbar.o…   2612       Richter

[2 rows x 22 columns]
```

[16]:
```
# create dfs
skills = pd.DataFrame()
abilities = pd.DataFrame()
```

Extract data and add IDs

[17]:
```
# extract all skills from the onet data
skills = df['skills'].apply(lambda x: x['element']).explode().
 ↪reset_index(drop=True)

# Convert the list of dictionaries into a DataFrame
skills_df = pd.json_normalize(skills)

# Drop duplicates based on 'id'
skills_df = skills_df.drop_duplicates(subset='id')

# Reset the index
skills_df = skills_df.reset_index(drop=True)

# drop cols
skills_df = skills_df.drop(columns=["related", "score.scale", "score.
 ↪important", "score.value"])

# add numeric id col
skills_df['skill_id'] = range(1, len(skills_df) + 1)

skills_df.head(3)
```

[17]:
```
        id                    name  \
0  2.A.1.a  Reading Comprehension
1  2.A.1.d               Speaking
2  2.A.1.b       Active Listening

                                         description  skill_id
0  Understanding written sentences and paragraphs…         1
1  Talking to others to convey information effect…         2
2  Giving full attention to what other people are…         3
```

[18]:
```
# extract all abilities from the onet data
abilities = df['abilities'].apply(lambda x: x['element']).explode().
 ↪reset_index(drop=True)
```

```
# Convert the list of dictionaries into a DataFrame
abilities_df = pd.json_normalize(abilities)

# Drop duplicates based on 'id'
abilities_df = abilities_df.drop_duplicates(subset='id')

# Reset the index
abilities_df = abilities_df.reset_index(drop=True)

# drop cols
abilities_df = abilities_df.drop(columns=["related", "score.scale", "score.
 ↪important" ,"score.value"])

# add numeric id col
abilities_df['ability_id'] = range(1, len(abilities_df) + 1)

abilities_df.head(3)
```

[18]:

	id	name	description	ability_id
0	1.A.1.a.3	Oral Expression	The ability to communicate information and ide…	1
1	1.A.1.a.1	Oral Comprehension	The ability to listen to and understand inform…	2
2	1.A.1.d.1	Memorization	The ability to remember information such as wo…	3

Save to csv

[19]:
```
abilities_df.to_csv("files/onet_abilities.csv", index=False)
skills_df.to_csv("files/onet_skills.csv", index=False)
```