Name	Matr.Nr.	Due Date
Mohamed Abdelaziz	K12137202	19.11.2021

Hands-on AI I

Unit 2 -- Reading, handling and visualization of datasets ¶

Authors: Brandstetter, Schäfl, Parada-Cabaleiro, Schörgenhumer
Date: 18-10-2021

In [116]:
```
# Required packages and the u2_utils file
import u2_utils as u2
import IPython.display as ipd
import numpy as np
import spacy
from matplotlib.image import imread

u2.check_module_versions()
```

```
Installed Python version: 3.9 (✓)
Installed numpy version: 1.21.1 (✓)
Installed pandas version: 1.3.1 (✓)
Installed scikit-learn version: 1.0 (✓)
Installed matplotlib version: 3.4.3 (✓)
Installed scipy version: 1.7.1 (✓)
Installed spacy version: 3.1.3 (✓)
```

Exercise 1

Following the instructions given in the lecture notebook, perform the tasks below:

- Plot the image of a tulip named `flower.jpg` (you will find this image in the folder `resources`)
- Load the image and print its dimensions (you will see that height and length/width are different to those from the image used in the lecture notebook)

1.1. Plot the flower image.

In [117]:
```
image_path = "resources/flower.jpg"
u2.plot_image(image_path)
```

1.2. Load and print the image's dimensions.

In [118]:
```
img = imread(image_path)
print(img.shape)
```

```
(650, 830, 3)
```

Exercise 2

By using the functions from `u2_utils.py` and working with the flower image, perform the tasks below:

- Plot the image with a transparency of 35%
- Plot the image by flipping it horizontally
- Plot the histograms for each color channel in the image and try to find the good **color threshold** values to segment the image into flower and background

2.1. Plot with transparency = 35%.

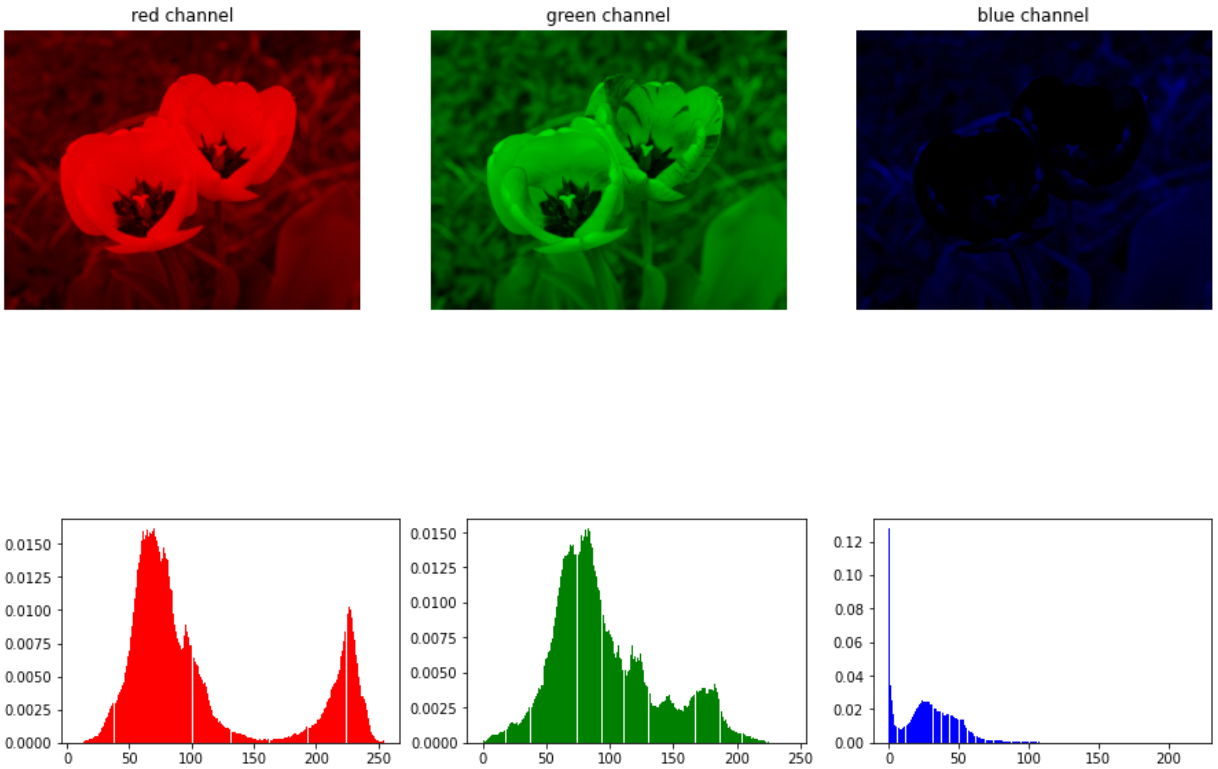
In [119]:
```
u2.plot_image_rgba(image_path, alpha=0.35)
```

2.2. Plot horizontally flipped image.

In [120]:
```
u2.plot_flipped_image(image_path, flipping="horizontal")
```

2.3. Plot RGB channels and histograms.

In [121]:
```
u2.plot_image_channels_rgb(image_path)
u2.plot_color_histograms(image_path)
```



2.4. Segment the image.

In [123]:
```
u2.segment_image(image_path, lower_threshold_g=135, lower_threshold_b=1)
u2.segment_image(image_path, lower_threshold_g=35, upper_threshold_g=135, lower_t
```

Exercise 3

With the functions from the file `u2_utils.py`, as shown in the lecture notebook, perform the tasks below:

- Generate three sine waves (all of them with a duration of 1 second, at a sampling rate of 24000 Hz) with the following frequencies: 349.228 Hz, 440 Hz and 523.251 Hz (in music terms, this is called an F major chord, here with the notes F4, A4 and C5). Then, add them together into a complex sound, plot the wave with `u2.plot_wave(...)` and generate a player to listen to it
- Use the `u2.apply_fourier_transform(...)` function to apply the discrete Fourier transform (DFT) to this complex sound and plot the output, i.e., decompose the complex sound into the original sine waves. Choose a meaningful maximum frequency when plotting the spectrum.

3.1. Generate three sine waves, sum them up, plot the combined wave and generate a player to listen to the complex sound.

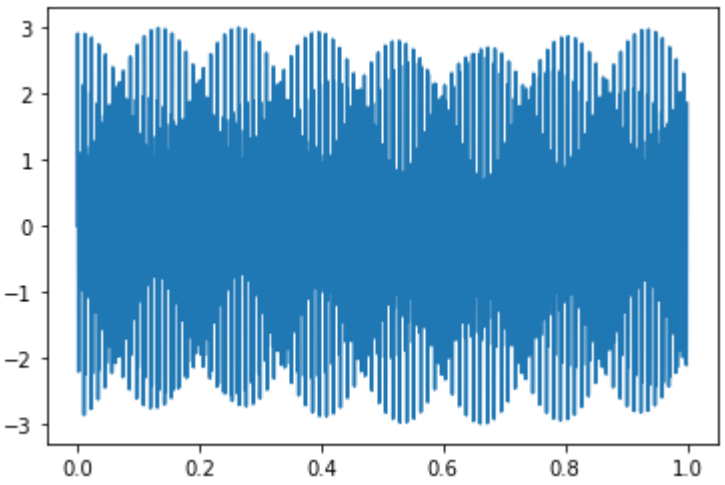
In [124]:
```
# Duration
T = 1
# Sampling Rate
sampling_rate = 24000

# Frequencies
f1 = 349.228
f2 = 440
f3 = 523.251

# Wave Points
wp_1 = u2.generate_wave(f1, T, sampling_rate)
wp_2 = u2.generate_wave(f2, T, sampling_rate)
wp_3 = u2.generate_wave(f3, T, sampling_rate)

# Combines Wave Points
wp = wp_1 + wp_2 + wp_3

# Plotting the wave
u2.plot_wave(wp, T, sampling_rate)
ipd.Audio(wp, rate=sampling_rate)
```

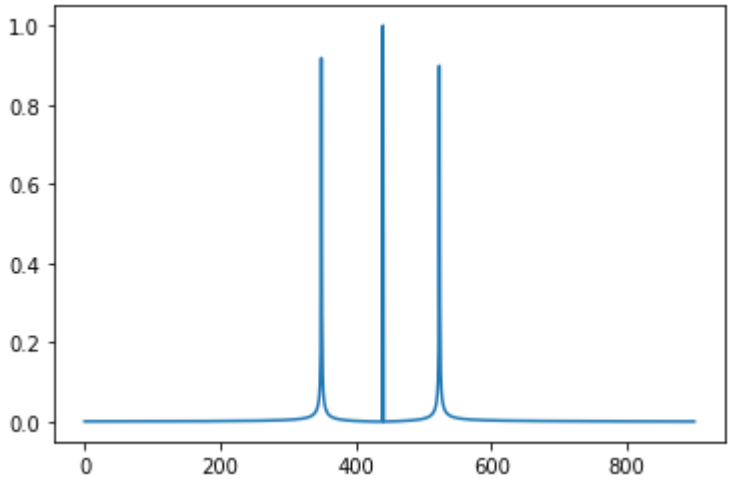


Out[124]:

0:00 / 0:01

3.2. Compute the discrete Fourier transform and plot the output.

In [125]:
```
fourier = u2.apply_fourier_transform(wp)
u2.plot_spectrum(fourier, sampling_rate, max_freq=900)
```



Exercise 4

By using the functions `u2.read_wav_file(...)` and `u2.apply_fourier_transform(...)` from `u2_utils.py` , perform the task below:

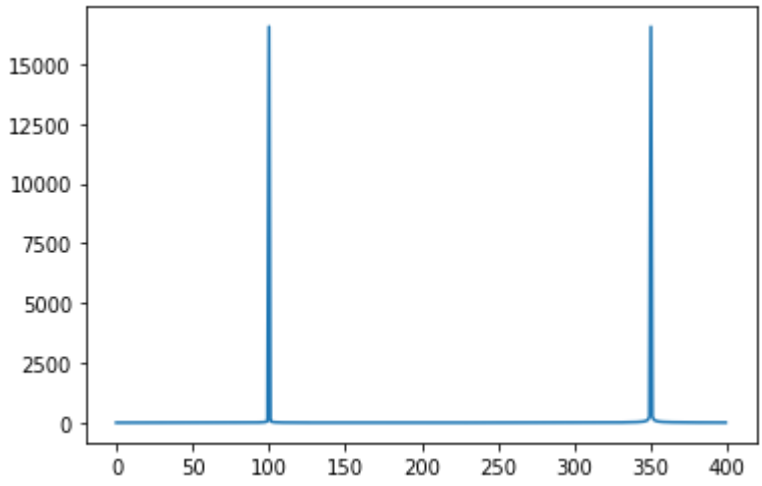
- Read the audio file named `audio.wav` from the `resources` folder.
- Apply Fourier analysis and plot the Fourier spectrum.
- Identify the frequencies of the two hidden sine waves. Hint, adapt `max_freq` from the `u2.plot_spectrum(...)` function.

4.1. Read audio file, apply Fourier analysis and plot the resulting spectrum.

In [126]:
```
audio_file = "resources/audio.wav"
T = 1
wp_audio, sampling_rate_audio = u2.read_wav_file(audio_file, time=T)

# u2.plot_wave(wp_audio, T, sampling_rate_audio)
# ipd.Audio(wp_audio, rate=sampling_rate_audio)

fourier_audio = u2.apply_fourier_transform(wp_audio)
u2.plot_spectrum(fourier_audio, sampling_rate_audio, max_freq=400)
```



4.2. What are the two frequencies?

Frequencies = (100,350)

Exercise 5

Following the instructions given in the lecture notebook, perform the tasks below:

- Download the word embedding model and load it by using the `spacy` library
- Evaluate which two words have a higher similarity: "notebook" and "pen" or "cable" and "waterfall". Then, print out the result. Hint: You could use an if/else statement to make the comparison (feel free to investigate this option on the internet). Otherwise, simply evaluate both similarities manually and print the higher one.

5.1. Download and load the word embedding.

In [127]:
```
# Download Worf Embedding
!python -m spacy download en_core_web_md
# Load Embedding
word_embedding = spacy.load('en_core_web_md')
```

```
Collecting en-core-web-md==3.1.0
  Downloading https://github.com/explosion/spacy-models/releases/download/en_core_web_md-3.1.0/en_core_web_md-3.1.0-py3-none-any.whl (https://github.com/explosion/spacy-models/releases/download/en_core_web_md-3.1.0/en_core_web_md-3.1.0-py3-none-any.whl) (45.4 MB)
Requirement already satisfied: spacy<3.2.0,>=3.1.0 in c:\users\moham\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from en-core-web-md==3.1.0) (3.1.3)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.8 in c:\users\moham\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from spacy<3.2.0,>=3.1.0->en-core-web-md==3.1.0) (3.0.8)
Requirement already satisfied: typer<0.5.0,>=0.3.0 in c:\users\moham\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from spacy<3.2.0,>=3.1.0->en-core-web-md==3.1.0) (0.4.0)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in c:\users\moham\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from spacy<3.2.0,>=3.1.0->en-core-web-md==3.1.0) (1.0.5)
Requirement already satisfied: packaging>=20.0 in c:\users\moham\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from spacy<3.2.0,>=3.1.0->en-core-web-md==3.1.0) (21.0)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in c:\users\moham\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from spacy<3.2.0,>=3.1.0->en-core-web-md==3.1.0) (3.0.5)
Requirement already satisfied: thinc<8.1.0,>=8.0.9 in c:\users\moham\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from spacy<3.2.0,>=3.1.0->en-core-web-md==3.1.0) (8.0.10)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in c:\users\moham\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from spacy<3.2.0,>=3.1.0->en-core-web-md==3.1.0) (2.0.6)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in c:\users\moham\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from spacy<3.2.0,>=3.1.0->en-core-web-md==3.1.0) (4.62.1)
Requirement already satisfied: setuptools in c:\program files\windowsapps\pythonsoftwarefoundation.python.3.9_3.9.2032.0_x64__qbz5n2kfra8p0\lib\site-packages (from spacy<3.2.0,>=3.1.0->en-core-web-md==3.1.0) (57.4.0)
Requirement already satisfied: wasabi<1.1.0,>=0.8.1 in c:\users\moham\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from spacy<3.2.0,>=3.1.0->en-core-web-md==3.1.0) (0.8.2)
Requirement already satisfied: blis<0.8.0,>=0.4.0 in c:\users\moham\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from spacy<3.2.0,>=3.1.0->en-core-web-md==3.1.0) (0.7.4)
```

```
Requirement already satisfied: pathy>=0.3.5 in c:\users\moham\appdata\local\pac
kages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packag
es\python39\site-packages (from spacy<3.2.0,>=3.1.0->en-core-web-md==3.1.0) (0.
6.0)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in c:\users\moham\appdata\lo
cal\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local
-packages\python39\site-packages (from spacy<3.2.0,>=3.1.0->en-core-web-md==3.
1.0) (2.0.5)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in c:\users\moham\appdat
a\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\l
ocal-packages\python39\site-packages (from spacy<3.2.0,>=3.1.0->en-core-web-md=
=3.1.0) (2.26.0)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<1.9.0,>=1.7.4 in c:\users
\moham\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0
\localcache\local-packages\python39\site-packages (from spacy<3.2.0,>=3.1.0->en
-core-web-md==3.1.0) (1.8.2)
Requirement already satisfied: jinja2 in c:\users\moham\appdata\local\packages
\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\py
thon39\site-packages (from spacy<3.2.0,>=3.1.0->en-core-web-md==3.1.0) (3.0.2)
Requirement already satisfied: srsly<3.0.0,>=2.4.1 in c:\users\moham\appdata\lo
cal\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local
-packages\python39\site-packages (from spacy<3.2.0,>=3.1.0->en-core-web-md==3.
1.0) (2.4.1)
Requirement already satisfied: numpy>=1.15.0 in c:\users\moham\appdata\local\pa
ckages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packa
ges\python39\site-packages (from spacy<3.2.0,>=3.1.0->en-core-web-md==3.1.0)
 (1.21.1)
Requirement already satisfied: pyparsing>=2.0.2 in c:\users\moham\appdata\local
\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-pa
ckages\python39\site-packages (from packaging>=20.0->spacy<3.2.0,>=3.1.0->en-co
re-web-md==3.1.0) (2.4.7)
Requirement already satisfied: smart-open<6.0.0,>=5.0.0 in c:\users\moham\appda
ta\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache
\local-packages\python39\site-packages (from pathy>=0.3.5->spacy<3.2.0,>=3.1.0-
>en-core-web-md==3.1.0) (5.2.1)
Requirement already satisfied: typing-extensions>=3.7.4.3 in c:\users\moham\app
data\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcach
e\local-packages\python39\site-packages (from pydantic!=1.8,!=1.8.1,<1.9.0,>=1.
7.4->spacy<3.2.0,>=3.1.0->en-core-web-md==3.1.0) (3.10.0.0)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\moham\appdata
\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\lo
cal-packages\python39\site-packages (from requests<3.0.0,>=2.13.0->spacy<3.2.0,
>=3.1.0->en-core-web-md==3.1.0) (1.26.7)
Requirement already satisfied: idna<4,>=2.5 in c:\users\moham\appdata\local\pac
kages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packag
es\python39\site-packages (from requests<3.0.0,>=2.13.0->spacy<3.2.0,>=3.1.0->e
n-core-web-md==3.1.0) (3.2)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\moham\appdata\loc
al\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-
packages\python39\site-packages (from requests<3.0.0,>=2.13.0->spacy<3.2.0,>=3.
1.0->en-core-web-md==3.1.0) (2021.10.8)
Requirement already satisfied: charset-normalizer~=2.0.0 in c:\users\moham\appd
ata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache
\local-packages\python39\site-packages (from requests<3.0.0,>=2.13.0->spacy<3.
2.0,>=3.1.0->en-core-web-md==3.1.0) (2.0.7)
Requirement already satisfied: colorama in c:\users\moham\appdata\local\package
s\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\p
```

```
ython39\site-packages (from tqdm<5.0.0,>=4.38.0->spacy<3.2.0,>=3.1.0->en-core-w
eb-md==3.1.0) (0.4.4)
Requirement already satisfied: click<9.0.0,>=7.1.1 in c:\users\moham\appdata\lo
cal\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local
-packages\python39\site-packages (from typer<0.5.0,>=0.3.0->spacy<3.2.0,>=3.1.0
->en-core-web-md==3.1.0) (8.0.1)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\moham\appdata\local
\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-pa
ckages\python39\site-packages (from jinja2->spacy<3.2.0,>=3.1.0->en-core-web-md
==3.1.0) (2.0.1)
[+] Download and installation successful
You can now load the package via spacy.load('en_core_web_md')


WARNING: You are using pip version 21.2.4; however, version 21.3 is available.
You should consider upgrading via the 'C:\Users\moham\AppData\Local\Microsoft\W
indowsApps\PythonSoftwareFoundation.Python.3.9_qbz5n2kfra8p0\python.exe -m pip
install --upgrade pip' command.
```

5.2. Evaluate similarity: "notebook" and "pen" vs. "cable" and "waterfall".

In [132]:
```
# initialize Variables
notebook = word_embedding('notebook')
pen = word_embedding('pen')
cable = word_embedding('cable')
waterfall = word_embedding('waterfall')

# Similarity
notebook.similarity(pen)
cable.similarity(waterfall)
```

Out[132]:
```
0.11750528297547742
```

Exercise 6

With the functions from the file `u2_utils.py`, including `u2.apply_tsne(...)`, i.e., the function to apply the t-SNE algorithm for dimensionality reduction already described in the first lecture, perform the tasks below:

- Following the example given in the lecture notebook, create a list containing these words: "car", "cloud", "dark", "diesel", "exercise", "grade", "grim", "homework", "lecture", "motor", "night", "petrol", "rain", "school", "storm", "study", "teacher", "thunder", "tire", "transmission", "wheel". Hint: A list can be created with `my_list = [...]`, where `...` is the content.
- Get the embeddings of the words and display the result data frame.
- Down-project the 300-dimensional representation of each word to 2 dimensions by applying the t-SNE method, i.e., the function `u2.apply_tsne(...)`. Then, plot it in a 2D representation by using the function `u2.plot_word_embeddings_2d(...)`. Look for a fitting `perplexity` value in order to group the words into meaningful clusters. Hint: Setting the seed for the random number generator `np.random.seed(seed=...)` is not necessary but useful if you want reproducible results.
- Verify that the perplexity was correct by applying the PCA down-projection method and plotting the result in a 2D representation (this will clearly show the different clusters)

6.1. Create a list with the indicated words.

In [133]:
```
word_list = ["car", "cloud", "dark", "diesel", "exercise", "grade", "grim", "home
```

6.2. Get embeddings and display the data frame.

In [134]:
```
df_embeddings = u2.get_word_vectors(word_embedding, word_list)
df_embeddings
```

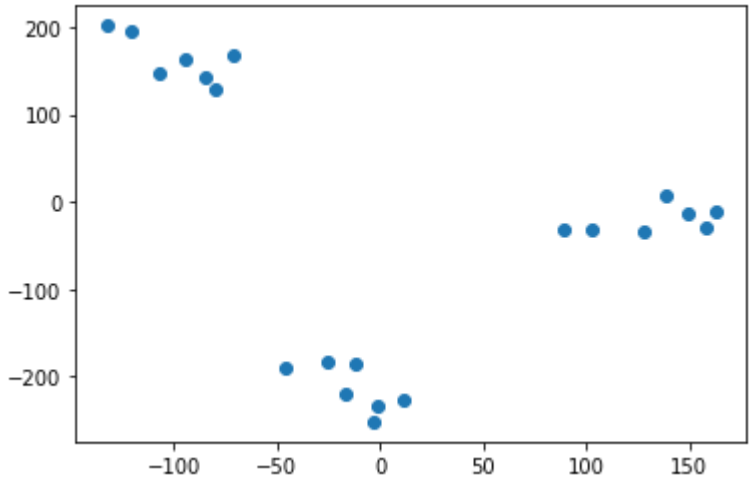
Out[134]:

	0	1	2	3	4	5	6	7
car	0.209870	0.464810	-0.242380	-0.065751	0.608560	-0.346980	-0.253310	-0.425900
cloud	0.110850	-0.274010	0.497370	-0.096619	-0.419780	-0.077946	-0.227320	0.536840
dark	0.236970	-0.012691	0.025805	0.087690	0.004066	-0.151640	-0.317490	0.239520
diesel	0.074851	0.350290	0.190260	-0.333050	0.425360	0.359850	-0.121180	-0.346060
exercise	-0.042700	0.278030	0.153170	-0.091364	-0.496480	0.206410	-0.417420	-0.288840
grade	-0.656050	0.483440	-0.083531	0.338010	0.095574	-0.056138	0.347310	0.291710
grim	0.187990	-0.031330	0.245440	0.014962	-0.288610	-0.296700	0.042305	0.400510
homework	-0.537830	0.123420	-0.320550	0.047442	-0.489640	0.154940	0.142420	-0.194470
lecture	-0.224790	0.070714	-0.029326	0.532260	-0.091217	0.113060	0.165450	-0.376670
motor	0.725850	0.466320	0.000849	-0.592510	-0.135420	-0.346780	0.171730	-0.583980
night	0.333580	0.159440	0.180540	0.071267	-0.041976	-0.206280	0.081888	-0.252500
petrol	-0.182910	0.243110	0.433100	-0.094089	0.141000	-0.385810	0.094114	0.008765
rain	0.324600	0.526210	0.136880	-0.144660	-0.463330	0.100970	-0.294250	0.411170
school	0.102310	0.277400	-0.057097	-0.166040	0.291030	-0.157290	-0.331180	0.115920
storm	0.166880	0.216020	0.563300	-0.124580	-0.204020	-0.145840	-0.140640	0.689530
study	-0.063718	-0.095939	0.007423	-0.223330	-0.617150	0.069799	-0.326370	0.216910
teacher	-0.275350	0.416730	-0.259720	0.494890	0.418850	-0.092241	-0.202580	-0.222150
thunder	0.223790	0.448800	0.370240	-0.420080	0.819880	0.054773	0.182930	0.380190
tire	0.241080	0.831030	-0.511740	-0.388330	0.116610	-0.275020	0.211430	-0.111200
transmission	0.212660	0.347000	-0.069504	-0.330860	-0.592680	0.218030	0.114600	-0.243440
wheel	0.507110	0.724710	0.038327	-0.679270	-0.066860	0.084498	0.106990	-0.165120

21 rows × 300 columns

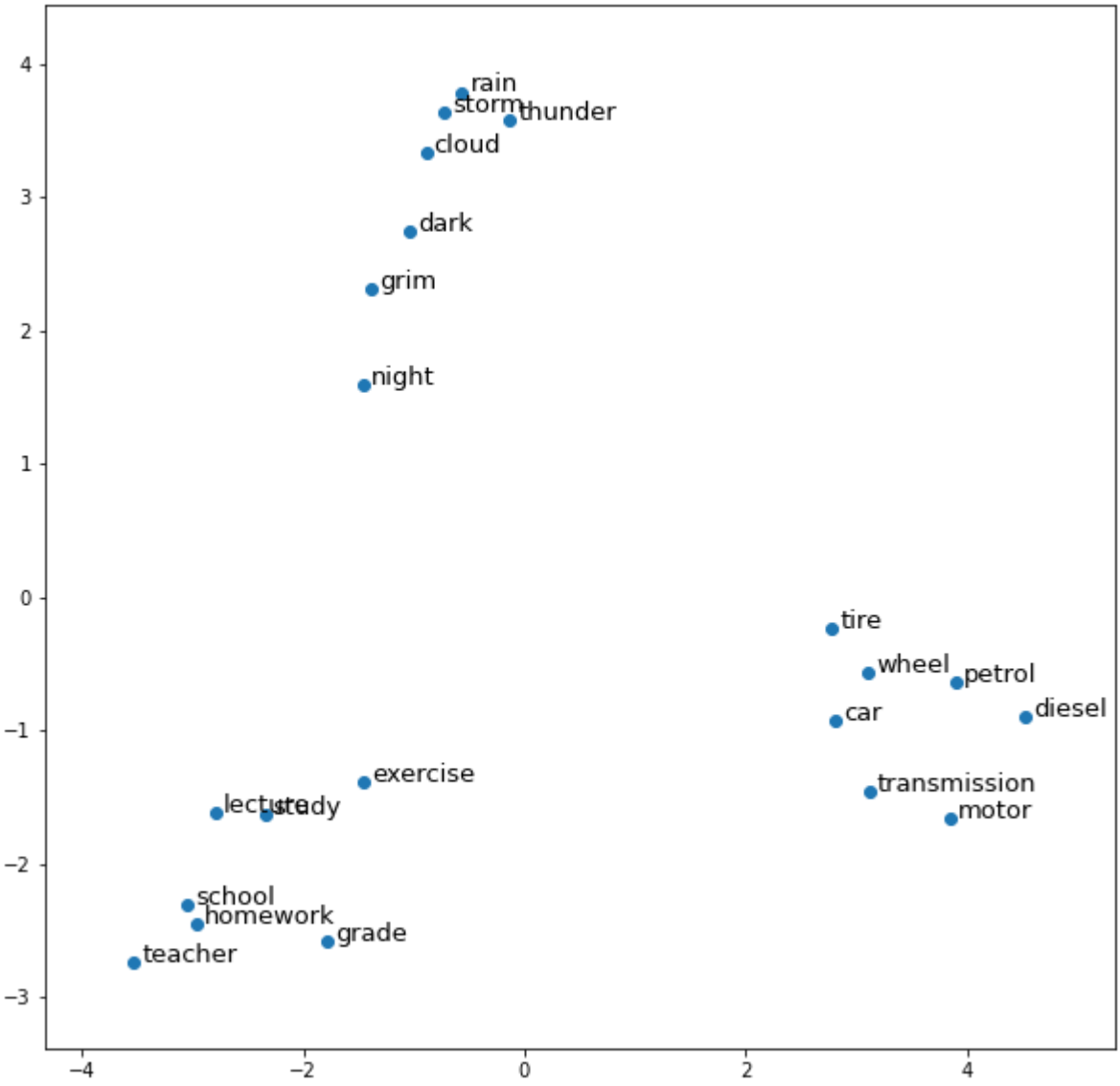
6.3. Apply t-SNE and plot the resut in a 2D representation.

In [135]:
```
np.random.seed(seed=21)
embeddings_tsne = u2.apply_tsne(n_components=2, data=df_embeddings, perplexity=3.
u2.plot_points_2d(data=embeddings_tsne)
```



6.4. Apply PCA and plot the result in a 2D representation.

In [136]:
```
embeddings_pca_2d = u2.apply_pca(n_components=2, data=df_embeddings)
u2.plot_word_embeddings_2d(embeddings_pca_2d)
```



In [ ]: