Lab Week 10

Sentence similarity using Transformers

Simon Mille, Michela Lorandi
simon.mille@adaptcentre.ie, michela.lorandi@adaptcentre.ie

Code: https://github.com/mille-s/Sentence-similarity

Overview of Week 10 Lab

- We will use the **nli-distil-roberta-base** model to determine whether two sentences convey the same information, in two different ways:
    - Using **nli-distil-roberta-base** to produce (separate) sentence embeddings for two sentences, then computing the cosine similarity between them
    - First finetuning **nli-distil-roberta-base** on the supervised task of directly predicting the similarity between two sentences by minimising the difference between the cosine similarity and the target output (a similarity score), then applying it as above
- The finetuning dataset contains data-to-text data (WebNLG'20) where the input data first needs to be converted into something resembling a sentence, so that an embedding for it can be obtained and its semantic similarity to the text can be determined.
- Another challenge is that we need data/text pairs of different levels of similarity, but WebNLG'20 only gives us semantically identical pairs, so the non-identical pairs need to be synthetically created
- Will look at different ways of doing the input data conversion and non-identical data creation
- Finally a semantic-similarity threshold needs to be determined above which data and text are considered to convey the same information

Plan

- Together, we will:
  - Have an overview of the task and the challenges
  - Have a look at the data
  - Evaluate an off-the-shelf model
  - Create a simple dataset to fine-tune the model
  - Fine-tune the model
  - Evaluate the created model
- Then, you will:
  - Come up with your own experiments, e.g. changing data or basic fine-tuning parameters
  - Fine-tune one or more model(s)
  - Evaluate your model
  - Present conclusions on your experiment(s)

What matters is not necessarily to improve the accuracy of the model, but to **test some hypotheses**

A negative result is still a good result!

- ***Code: Lab_Week10_colab.ipynb***
  - *Choose **Runtime type = T4 GPU***
  - *2 main parts: Evaluation (EV) and Fine-tuning (FT)*
  - *Blue pointers in slides, e.g. **EV1.2***

The general context

- Task: Obtain a semantic similarity score between 2 sentences
- How: Apply a pre-trained model that outputs a score between -1 and 1
  - (i) Get embeddings for each sentence, (ii) calculate cosine similarity between embeddings
  - **Try it**: Lab_Week10_eval_run, cell **EV-0.0**
    - [‘The cat sits outside’, ‘The dog plays in the garden’]: ???
    - [‘A man is playing guitar’, ‘A woman watches TV’]: ???
    - [‘The new movie is awesome’, ‘The new movie is so great’]: ???

Today: calculate similarity between structured data and text

- Why not just play with similarity between text and text?
  - The model is not designed to do data/text: fine-tuning it is more interesting!
- Structured data is for example DBpedia triples
  - Go to https://dbpedia.org/page/Aarhus_Airport:
    - Subject (about): Aarhus_Airport
    - Property: left column
    - Object (value): right column
  - Triple for *dbo:location*: 'Aarhus_Airport | location | Tirstrup'
  - Triple for *dbp:cityServed*: 'Aarhus_Airport | cityServed| Aarhus,_Denmark'
- Use cases:
  - Find sentences that express exactly a triple or a triple set
  - Evaluate the quality of **Natural Language Generation** systems that produce texts from triples
  - Evaluate the quality of **Information Extraction** systems that produce triples from text
  - etc.

Challenges

- The model was trained to calculate similarity between:

  [Sentence1, Sentence2]

- and now we have:

  [Triple, Sentence]

- Examples:

[‘Aarhus_Airport | cityServed | Aarhus,_Denmark’, ‘Aarhus Airport serves the city of Aarhus, Denmark’]: ???

['Aarhus_Airport | location | Tirstrup', 'Aarhus Airport is in Tirstrup']: ???

Try it: Lab_Week10_eval_run, cell **EV-0.0**; uncomment the Triples (l.12-13) and the corresponding sentences.

- Comments?

Can we improve the scores?

Starting point, with nli-distilroberta-base-v2; the scores don't look like they indicate a match:

['Aarhus_Airport | cityServed | Aarhus,_Denmark', 'Aarhus Airport serves the city of Aarhus, Denmark']: 0.8245

['Aarhus_Airport | location | Tirstrup', 'Aarhus Airport is in Tirstrup']: 0.7943

What if we make the triple look more like a sentence? ->TextTriples

['Aarhus Airport city served Aarhus, Denmark', 'Aarhus Airport serves the city of Aarhus, Denmark']: ???

['Aarhus Airport location Tirstrup', 'Aarhus Airport is in Tirstrup']: ???

Try it: Lab_Week10_eval_run, cell **EV-0.0**; uncomment the "Textified" triples (l.15-16) and the corresponding sentences.

- Comments?

Can we further improve the scores?

Starting point, with nli-distilroberta-base-v2; the scores are better:

[‘Aarhus Airport city served Aarhus, Denmark’, ‘Aarhus Airport serves the city of Aarhus, Denmark’]: 0.9802

['Aarhus Airport location Tirstrup', 'Aarhus Airport is in Tirstrup']: 0.9270

- Let’s see how nli-distilroberta-base-v2 performs on a larger scale with this task.
- Then let’s think of ways to make it better at the task by fine-tuning it:
  a. Create a fine-tuning dataset.
  b. Fine-tune the model.
  c. Evaluate the fine-tuned model and compare it to the off-the-shelf model.
- Before evaluating, let’s have a look at the data we’ll use for fine-tuning and for evaluation

Creating a dataset to help the model score more accurately

- The base (off-the-shelf) model is pre-trained on **Sentence-Sentence** pairs
- We want to show it how to score **TextTriples-Sentence** pairs
- We will then need to compile data in this format
- There is an existing dataset of **Triples-Sentences** pairs: WebNLG+

Overview of the WebNLG+ dataset

- Documentation available at https://synalp.gitlabpages.inria.fr/webnlg-challenge/docs/
- We will load the dataset via HuggingFace; doc here: https://huggingface.co/docs/datasets/index
- Here are two data points of the dataset, with one/two triple(s) and two possible verbalisations each:

triple ->

target texts ->

```
<entry category="Airport" eid="Id4" shape="(X (X))" shape_type="NA" size="1">
  <modifiedtripleset>
    <mtriple>Aarhus_Airport | location | Tirstrup</mtriple>
  </modifiedtripleset>
  <lex comment="good" lid="Id1">Aarhus Airport is located in Tirstrup.</lex>
  <lex comment="good" lid="Id2">The location of Aarhus Airport is Tirstrup.</lex>
</entry>
```

triples ->

target texts ->

```
<entry category="ComicsCharacter" eid="Id7" shape="(X (X) (X))" shape type="sibling" size="2">
  <modifiedtripleset>
    <mtriple>Arion_(comicsCharacter) | creator | Jan_Duursema</mtriple>
    <mtriple>Arion_(comicsCharacter) | alternativeName | "Ahri'ahn"</mtriple>
  </modifiedtripleset>
  <lex comment="good" lid="Id1">The comic character Arion, also known as Ahri'ahn, was created by Jan Duursema.</lex>
  <lex comment="good" lid="Id2">Created by Jan Duursema, the comic character Arion is also known as Ahri'ahn.</lex>
</entry>
```

Data splits, use

- Like most datasets, WebNLG+ is split into
  - Training: ~80%
  - Development: ~10%
  - Test: ~10%
- WebNLG+ is usually used to train:
  - Natural language generation systems (NLG)
    - triples-> text
  - Information extraction systems (IE)
    - text-> triples

	Training (#)	Development (#)
Triple sets - all sizes (1-7)	13,211	1,667
Texts - all sizes (1-7)	35,426	4,464
Triples sets - size 1	**3,107**	**401**
Texts - size 1	**7,630**	**961**

NLG
Input: triple
Output: text

```
<entry category="Airport" eid="Id4" shape="(X (X))" shape_type="NA" size="1">
  <modifiedtripleset>
    <mtriple>Aarhus_Airport | location | Tirstrup</mtriple>
  </modifiedtripleset>
  <lex comment="good" lid="Id1">Aarhus Airport is located in Tirstrup.</lex>
  <lex comment="good" lid="Id2">The location of Aarhus Airport is Tirstrup.</lex>
</entry>
```

IE
Input: text
Output: triple

How we'll use the dataset: fine-tuning (training split)

- As we've just seen, the dataset consists of (triple, sentence) pairs:

```
<entry category="Airport" eid="Id4" shape="(X (X))" shape_type="NA" size="1">
  <modifiedtripleset>
    <mtriple>Aarhus_Airport | location | Tirstrup</mtriple>
  </modifiedtripleset>
  <lex comment="good" lid="Id1">Aarhus Airport is located in Tirstrup.</lex>
  <lex comment="good" lid="Id2">The location of Aarhus Airport is Tirstrup.</lex>
</entry>
```

Pair#1: ('Aarhus_Airport | location | Tirstrup', 'Aarhus Airport is located in Tirstrup.')
Pair#2: ('Aarhus_Airport | location | Tirstrup', 'The location of Aarhus Airport is Tirstrup.')

- We know that each sentence matches the meaning of the triple:

Input#1: ('Aarhus_Airport | location | Tirstrup', 'Aarhus Airport is located in Tirstrup.')
Output#1: similarity score = 1

Input#2: ('Aarhus_Airport | location | Tirstrup', 'The location of Aarhus Airport is Tirstrup.')
Output#2: similarity score = 1

Input#3: ('Aarhus_Airport | location | Tirstrup', 'Alan Bean was born in Wheeler, Texas.')
Output#3: similarity score = 0

….

We'll just need to make the inputs look more like text!
Triple -> *TextTriple*

How we'll use the dataset: fine-tuning (training split)

	Training (#)	Development (#)
Triple sets - all sizes (1-7)	13,211	1,667
Texts - all sizes (1-7)	35,426	4,464
Triples sets - size 1	**3,107**	401
Texts - size 1	**7,630**	961

Basic idea:

We can then have **7,630** pairs like this:
Input#*i*: (*TextTriple*, *Sentence*)
Output#*i*: similarity score = *1*

And **~3,107*7,627** pairs like this:
Input#*j*: (*TextTriple*, *Sentence*)
Output#*j*: similarity score = *0*

There are approx. 3 verbalisations per triple in the dataset (score = 1), so 7,630 - 3 = 7,627 are not verbalisations.

How we'll use the dataset: evaluation (dev split)

- Task: given an input triple (-> TextTriple), find its verbalisation(s) in a pool of candidate sentences

	Training (#)	Development (#)
Triple sets - all sizes (1-7)	13,211	1,667
Texts - all sizes (1-7)	35,426	4,464
Triples sets - size 1	3,107	**401**
Texts - size 1	7,630	**961**

401 TextTriples

Alba Iulia country Romania
11 Diagonal Street building end date 1983
Apollo 14 operator NASA
Adare Manor country Republic of Ireland
AIDA Cruises location Rostock
…

961 Candidate sentences

Alba Iulia is located in Romania.
Alba Iulia is in Romania.
11 Diagonal Street was completed in 1983.
Apollo 14 was operated by NASA.
NASA operated Apollo 14.
Adare Manor is located in the Republic of Ireland.
The Adare Manor is in the Republic of Ireland.
AIDA Cruises are located at Rostock.
AIDA Cruises is based in Rostock.
The AIDAluna is a Sphinx class cruise ship.
The Fylde is the home ground of AFC Fylde.
….

How we'll use the dataset: evaluation (dev split)

- Task: given an input triple (-> TextTriple), find its verbalisation(s) in a pool of candidate sentences

	Training (#)	Development (#)
Triple sets - all sizes (1-7)	13,211	1,667
Texts - all sizes (1-7)	35,426	4,464
Triples sets - size 1	3,107	**401**
Texts - size 1	7,630	**961**

401 TextTriples

Alba Iulia country Romania
11 Diagonal Street building end date 1983
Apollo 14 operator NASA
Adare Manor country Republic of Ireland
AIDA Cruises location Rostock
…

961 Candidate sentences

Alba Iulia is located in Romania.
Alba Iulia is in Romania.
11 Diagonal Street was completed in 1983.
Apollo 14 was operated by NASA.
NASA operated Apollo 14.
Adare Manor is located in the Republic of Ireland.
The Adare Manor is in the Republic of Ireland.
AIDA Cruises are located at Rostock.
AIDA Cruises is based in Rostock.
The AIDAluna is a Sphinx class cruise ship.
The Fylde is the home ground of AFC Fylde.
….

How we'll use the dataset: evaluation (dev split)

- Task: given an input triple (-> TextTriple), find its verbalisation(s) in a pool of candidate sentences

	Training (#)	Development (#)
Triple sets - all sizes (1-7)	13,211	1,667
Texts - all sizes (1-7)	35,426	4,464
Triples sets - size 1	3,107	**401**
Texts - size 1	7,630	**961**

401 TextTriples

Alba Iulia country Romania
11 Diagonal Street building end date 1983
Apollo 14 operator NASA
Adare Manor country Republic of Ireland
AIDA Cruises location Rostock
…

961 Candidate sentences

Alba Iulia is located in Romania.
Alba Iulia is in Romania.
11 Diagonal Street was completed in 1983.
Apollo 14 was operated by NASA.
NASA operated Apollo 14.
Adare Manor is located in the Republic of Ireland.
The Adare Manor is in the Republic of Ireland.
AIDA Cruises are located at Rostock.
AIDA Cruises is based in Rostock.
The AIDAluna is a Sphinx class cruise ship.
The Fylde is the home ground of AFC Fylde.
….

How we'll use the dataset: evaluation (dev split)

- Task: given an input triple (-> TextTriple), find its verbalisation(s) in a pool of candidate sentences

	Training (#)	Development (#)
Triple sets - all sizes (1-7)	13,211	1,667
Texts - all sizes (1-7)	35,426	4,464
Triples sets - size 1	3,107	**401**
Texts - size 1	7,630	**961**

401 TextTriples

Alba Iulia country Romania
11 Diagonal Street building end date 1983
Apollo 14 operator NASA
Adare Manor country Republic of Ireland
AIDA Cruises location Rostock
…

961 Candidate sentences

Alba Iulia is located in Romania.
Alba Iulia is in Romania.
11 Diagonal Street was completed in 1983.
Apollo 14 was operated by NASA.
NASA operated Apollo 14.
Adare Manor is located in the Republic of Ireland.
The Adare Manor is in the Republic of Ireland.
AIDA Cruises are located at Rostock.
AIDA Cruises is based in Rostock.
The AIDAluna is a Sphinx class cruise ship.
The Fylde is the home ground of AFC Fylde.
….

How we'll use the dataset: evaluation (dev split)

- Task: given an input triple (-> TextTriple), find its verbalisation(s) in a pool of candidate sentences

	Training (#)	Development (#)
Triple sets - all sizes (1-7)	13,211	1,667
Texts - all sizes (1-7)	35,426	4,464
Triples sets - size 1	3,107	**401**
Texts - size 1	7,630	**961**

401 TextTriples

Alba Iulia country Romania
11 Diagonal Street building end date 1983
Apollo 14 operator NASA
Adare Manor country Republic of Ireland
AIDA Cruises location Rostock
…

961 Candidate sentences

Alba Iulia is located in Romania.
Alba Iulia is in Romania.
11 Diagonal Street was completed in 1983.
Apollo 14 was operated by NASA.
NASA operated Apollo 14.
Adare Manor is located in the Republic of Ireland.
The Adare Manor is in the Republic of Ireland.
AIDA Cruises are located at Rostock.
AIDA Cruises is based in Rostock.
The AIDAluna is a Sphinx class cruise ship.
The Fylde is the home ground of AFC Fylde.
….

How we'll use the dataset: evaluation (dev split)

- Task: given an input triple (-> TextTriple), find its verbalisation(s) in a pool of candidate sentences

	Training (#)	Development (#)
Triple sets - all sizes (1-7)	13,211	1,667
Texts - all sizes (1-7)	35,426	4,464
Triples sets - size 1	3,107	**401**
Texts - size 1	7,630	**961**

401 TextTriples

Alba Iulia country Romania
11 Diagonal Street building end date 1983
Apollo 14 operator NASA
Adare Manor country Republic of Ireland
AIDA Cruises location Rostock
…

961 Candidate sentences

Alba Iulia is located in Romania.
Alba Iulia is in Romania.
11 Diagonal Street was completed in 1983.
Apollo 14 was operated by NASA.
NASA operated Apollo 14.
Adare Manor is located in the Republic of Ireland.
The Adare Manor is in the Republic of Ireland.
AIDA Cruises are located at Rostock.
AIDA Cruises is based in Rostock.
The AIDAluna is a Sphinx class cruise ship.
The Fylde is the home ground of AFC Fylde.
….

How we compute the evaluation results

Each candidate sentence will be assigned a similarity score for each TextTriple

TextTriple[124]:
Adare Manor country Republic of Ireland .
Candidate, score (sorted by score):
['Adare Manor is located in the Republic of Ireland.', 0.9908198118209839]
['The Adare Manor is in the Republic of Ireland.', 0.9873427152633667]
['Dublin is in the Republic of Ireland.', 0.8550620079040527]
['Swords, Dublin is led by a county manager.', 0.722202718257904]
['Swords, Dublin is part of the Dublin European Parliament constituency.', 0.708570122718811]
['The English language is the main language of the Republic of Ireland.', 0.703508198261261]
['The County Manager is the leader of Swords, Dublin.', 0.6999536752700806]
['One language used in the Republic of Ireland is English.', 0.6869048476219177]
['The title of the leader of Swords, Dublin is County Manager.', 0.6826339960098267]
['The leader title of Swords, Dublin is County Manager.', 0.6804064512252808]
['Swords is a part of the Dublin European Parliamentary constituency.', 0.6753323078155518]
['Dublin is part of Leinster.', 0.6584078073501587]
['Patrick McLoughlin is the leader of Derbyshire Dales.', 0.6280174255371094]
['Patrick McLoughlin is a leader in Derbyshire Dales.', 0.6258329153060913]
['Ireland official language is English.', 0.6230773329734802]
['Patrick McLoughlin is a leader in the Derbyshire Dales.', 0.618850827217102]
['Swords belongs to the Dublin constituency of the European Parliament.', 0.5804030895233154]
['Footballer, Alan Martin, plays for the club Crewe Alexandra F.C.', 0.4355524480342865]
['Adam McQuaid was born in Charlottetown.', 0.4347069263458252]
…

At which threshold do we get the best F1?

True positive (tp): A matching sentence is selected.
True negative (tn): A non-matching sentence is not selected.
False positive (fp): A non-matching sentence is selected.
False negative (fn): A matching sentence is not selected.

precision = tp / (fp + tp)
recall = tp / (fn + tp)

F1 = (2 * precision * recall) / (precision + recall)

How we compute the evaluation results

Threshold = 0.8

TextTriple[124]:
Adare Manor country Republic of Ireland .
Candidates & score:
['Adare Manor is located in the Republic of Ireland.', 0.9908198118209839]
['The Adare Manor is in the Republic of Ireland.', 0.9873427152633667]
['Dublin is in the Republic of Ireland.', 0.8550620079040527]
['Swords, Dublin is led by a county manager.', 0.722202718257904]
['Swords, Dublin is part of the Dublin European Parliament constituency.', 0.708570122718811]
['The English language is the main language of the Republic of Ireland.', 0.703508198261261]
['The County Manager is the leader of Swords, Dublin.', 0.6999536752700806]
['One language used in the Republic of Ireland is English.', 0.6869048476219177]
['The title of the leader of Swords, Dublin is County Manager.', 0.6826339960098267]
['The leader title of Swords, Dublin is County Manager.', 0.6804064512252808]
['Swords is a part of the Dublin European Parliamentary constituency.', 0.6753323078155518]
['Dublin is part of Leinster.', 0.6584078073501587]
['Patrick McLoughlin is the leader of Derbyshire Dales.', 0.6280174255371094]
['Patrick McLoughlin is a leader in Derbyshire Dales.', 0.6258329153060913]
['Ireland official language is English.', 0.6230773329734802]
['Patrick McLoughlin is a leader in the Derbyshire Dales.', 0.618850827217102]
['Swords belongs to the Dublin constituency of the European Parliament.', 0.5804030895233154]
['Footballer, Alan Martin, plays for the club Crewe Alexandra F.C.', 0.4355524480342865]
['Adam McQuaid was born in Charlottetown.', 0.4347069263458252]
...

At which threshold do we get the best F1?

True positive (tp): A matching sentence is selected.
True negative (tn): A non-matching sentence is not selected.
False positive (fp): A non-matching sentence is selected.
False negative (fn): A matching sentence is not selected.

precision = tp / (fp + tp)
recall = tp / (fn + tp)

F1 = (2 * precision * recall) / (precision + recall)

How we compute the evaluation results

Threshold = 0.99

TextTriple[124]:
Adare Manor country Republic of Ireland .
Candidates & score:
['Adare Manor is located in the Republic of Ireland.', 0.9908198118209839]
['The Adare Manor is in the Republic of Ireland.', 0.9873427152633667]
['Dublin is in the Republic of Ireland.', 0.8550620079040527]
['Swords, Dublin is led by a county manager.', 0.722202718257904]
['Swords, Dublin is part of the Dublin European Parliament constituency.', 0.708570122718811]
['The English language is the main language of the Republic of Ireland.', 0.703508198261261]
['The County Manager is the leader of Swords, Dublin.', 0.6999536752700806]
['One language used in the Republic of Ireland is English.', 0.6869048476219177]
['The title of the leader of Swords, Dublin is County Manager.', 0.6826339960098267]
['The leader title of Swords, Dublin is County Manager.', 0.6804064512252808]
['Swords is a part of the Dublin European Parliamentary constituency.', 0.6753323078155518]
['Dublin is part of Leinster.', 0.6584078073501587]
['Patrick McLoughlin is the leader of Derbyshire Dales.', 0.6280174255371094]
['Patrick McLoughlin is a leader in Derbyshire Dales.', 0.6258329153060913]
['Ireland official language is English.', 0.6230773329734802]
['Patrick McLoughlin is a leader in the Derbyshire Dales.', 0.618850827217102]
['Swords belongs to the Dublin constituency of the European Parliament.', 0.5804030895233154]
['Footballer, Alan Martin, plays for the club Crewe Alexandra F.C.', 0.4355524480342865]
['Adam McQuaid was born in Charlottetown.', 0.4347069263458252]
...

At which threshold do we get the best F1?

True positive (tp): A matching sentence is selected.
True negative (tn): A non-matching sentence is not selected.
False positive (fp): A non-matching sentence is selected.
False negative (fn): A matching sentence is not selected.

precision = tp / (fp + tp)
recall = tp / (fn + tp)

F1 = (2 * precision * recall) / (precision + recall)

How we compute the evaluation results

Threshold = 0.9

TextTriple[124]:
Adare Manor country Republic of Ireland .
Candidates & score:
['Adare Manor is located in the Republic of Ireland.', 0.9908198118209839]
['The Adare Manor is in the Republic of Ireland.', 0.9873427152633667]
['Dublin is in the Republic of Ireland.', 0.8550620079040527]
['Swords, Dublin is led by a county manager.', 0.722202718257904]
['Swords, Dublin is part of the Dublin European Parliament constituency.', 0.708570122718811]
['The English language is the main language of the Republic of Ireland.', 0.703508198261261]
['The County Manager is the leader of Swords, Dublin.', 0.6999536752700806]
['One language used in the Republic of Ireland is English.', 0.6869048476219177]
['The title of the leader of Swords, Dublin is County Manager.', 0.6826339960098267]
['The leader title of Swords, Dublin is County Manager.', 0.6804064512252808]
['Swords is a part of the Dublin European Parliamentary constituency.', 0.6753323078155518]
['Dublin is part of Leinster.', 0.6584078073501587]
['Patrick McLoughlin is the leader of Derbyshire Dales.', 0.6280174255371094]
['Patrick McLoughlin is a leader in Derbyshire Dales.', 0.6258329153060913]
['Ireland official language is English.', 0.6230773329734802]
['Patrick McLoughlin is a leader in the Derbyshire Dales.', 0.618850827217102]
['Swords belongs to the Dublin constituency of the European Parliament.', 0.5804030895233154]
['Footballer, Alan Martin, plays for the club Crewe Alexandra F.C.', 0.4355524480342865]
['Adam McQuaid was born in Charlottetown.', 0.4347069263458252]
...

At which threshold do we get the best F1?

True positive (tp): A matching sentence is selected.
True negative (tn): A non-matching sentence is not selected.
False positive (fp): A non-matching sentence is selected.
False negative (fn): A matching sentence is not selected.

precision = tp / (fp + tp)
recall = tp / (fn + tp)

F1 = (2 * precision * recall) / (precision + recall)

Find the best threshold

0.75

Evaluated model:
[True Positives, False Positives]: [939, 132]
[False Negatives, True Negatives]: [24, 384266]
Precision: 0.877
Recall: 0.975
accuracy: 0.998
f1-score: 0.923

0.76

Evaluated model:
[True Positives, False Positives]: [932, 106]
[False Negatives, True Negatives]: [31, 384292]
Precision: 0.898
Recall: 0.968
accuracy: 0.998
f1-score: 0.932

0.77

Evaluated model:
[True Positives, False Positives]: [925, 86]
[False Negatives, True Negatives]: [38, 384312]
Precision: 0.915
Recall: 0.961
accuracy: 0.998
f1-score: 0.937

0.78

Evaluated model:
[True Positives, False Positives]: [914, 77]
[False Negatives, True Negatives]: [49, 384321]
Precision: 0.922
Recall: 0.949
accuracy: 0.998
f1-score: 0.936

0.79

Evaluated model:
[True Positives, False Positives]: [903, 64]
[False Negatives, True Negatives]: [60, 384334]
Precision: 0.934
Recall: 0.938
accuracy: 0.998
f1-score: 0.936

0.8

Evaluated model:
[True Positives, False Positives]: [887, 60]
[False Negatives, True Negatives]: [76, 384338]
Precision: 0.937
Recall: 0.921
accuracy: 0.998
f1-score: 0.929

Evaluate a model (1)

Reminder: we want to evaluate this: ['Aarhus Airport location Tirstrup', 'Aarhus Airport is in Tirstrup']

1- Run 6 cells in **EV-1** to download and unzip the working folder and install transformers

Skip for fine-tuned model evaluation

Overview of the resources we need:

- Some input TextTriples, i.e. triples in close-to-text format
  - 401 dev triples: /content/Lab_Week10/Eval/triples_dev_1_textified.txt
- Some candidate sentences
  - 961 sentences: /content/Lab_Week10/Eval/candidateSentences_dev_1.txt
- A list with the target sentences (aligned with the input triples)
  - Used to check if a candidate has been correctly chosen or not as a match for the input triple
  - /content/Lab_Week10/Eval/targetSentences_dev_1.txt
- A model that outputs sentence similarity scores
  - We'll create one later
  - Produces files in /content/Lab_Week10/Eval/results_candSent folder
- A list with the input triples (you can ignore it, but don't erase it):
  - Used to compute additional results (e.g. for which properties does the model fail)
  - /content/Lab_Week10/Eval/triples_dev_1_split.txt

2- Then:

- **For fine-tuned model**: **paste path** in the form at the beginning of **EV-2.**
- Run 6 cells in **EV-2** to load files and model

Evaluate a model (2)

3- Score the candidate sentences (**Skip for off-the shelf model evaluation**)

- Run 8 cells in **EV-3 (use GPU: Runtime > Change runtime type)**
- This will create 401 files in /content/Lab_Week10/Eval/results_candSent
- Each file contains the score of each candidate sentence for one input triple

4- Get system-level scores

- **Choose the model to evaluate** in **EV-4.2**
- Run 6 cells in **EV-4**
- During this process, we calculate the F1 score for each threshold (from 0.01 to 0.99)
  - We use the scores obtained in the previous step
  - Question we answer: If I select all candidates with a score above threshold *t*, how many true/false positives/negatives do I get?

5- Read results

- Details about precision and recall for each threshold are printed first
- The aggregated results are printed at the end
- Best threshold for off-the-shelf model: 0.77 (F1 = 0.937)
  - Keep the scores to compare them to the scores of the other models!

We'll try to make a better model to improve on our best F1 (0.937).

2 steps:

- Create a dataset
- Fine-tune the model with this dataset
- We will use the code in Lab_Week10_colab.ipynb, **FT FINE TUNING** part

Create a dataset

- The current code creates a simple dataset to teach the model to scores “textified” triples instead of real sentences
- Sentence transformers needs a simple data structure to learn from
  - A list of objects with 2 attributes:
    - texts = [sentence1, sentence2] (a list with 2 sentences)
    - label = score (float)
- Experiment idea we talked about earlier:
  - Let’s use the WebNLG+ data to compile sentence pairs with 2 possible scores:
    - A score of 1.0: a TextTriple and one of its corresponding verbalisations in the dataset

```
<entry category="Airport" eid="Id4" shape="(X (X))" shape_type="NA" size="1">
  <modifiedtripleset>
    <mtriple>Aarhus_Airport | location | Tirstrup</mtriple>
  </modifiedtripleset>
  <lex comment="good" lid="Id1">Aarhus Airport is located in Tirstrup.</lex>
  <lex comment="good" lid="Id2">The location of Aarhus Airport is Tirstrup.</lex>
</entry>
```

      - texts = [‘AArhus Airport location Tirstrup’, Aarhus Airport is located in Tirstrup’]
      - score = 1.0.
    - A score of 0.0: a TextTriple and one verbalisation of a triple that has nothing in common with the original triple
      - texts = [‘AArhus Airport location Tirstrup’, Poaceae belongs to the order of Commelinids.’]
      - score = 0.0.

Create a dataset

- What we need for this experiment:
  - For each triple *T* we want:
    - Sentences that verbalise exactly *T*
      - *From the WebNLG+ training data*
    - Sentence that verbalise triples that have neither the Subject, the Property nor the Object in common with *T*
      - *From the WebNLG+ training data*
    - A version of *T* that looks more like a sentence (*TextTriple*)
      - *We have to create this*
  - Then, we need to format the data like this:
    - [Object$_i$ { texts = [‘TextTriple$_i$’, Sentence$_i$’], score = s$_i$}, Object$_j$ { texts = [‘TextTriple$_j$’, Sentence$_j$’], score = s$_j$}, etc.]

Create a dataset

Remember what we've seen earlier (what WebNLG+ provides us with):

Input#1: ('Aarhus_Airport | location | Tirstrup', 'Aarhus Airport is located in Tirstrup.')
Output#1: similarity score = 1

Input#2: ('Aarhus_Airport | location | Tirstrup', 'The location of Aarhus Airport is Tirstrup.')
Output#2: similarity score = 1

Input#3: ('Aarhus_Airport | location | Tirstrup', 'Alan Bean was born in Wheeler, Texas.')
Output#3: similarity score = 0

Create a dataset

- Now let's run the code:
  - Run cells **FT-1.1** to load the WebNLG+ dataset
    - Optional: Explore the dataset to get familiar with it and understand what it does
  - Run cells **FT-1.2** to extract pairs of sentences and their similarity scores
    - Explore the code and the data
  - Run cells **FT-1.3** to produce a simple verbalisation of the input triples (TextTtriples)
  - Run cells **FT-1.4** to save the dataset that combines the TextTtriples and the sentences in the format required
    - Explore the code and the data
- We now have a fine-tuning dataset!
  - /content/fine_tuning_dataset.txt

Fine-tune the model

- This part is very straightforward:
  - Run cell **FT-2.1** to set parameters
    - **Choose a path** to save the model in **FT-2.1**
      - by default in `'/content/drive/MyDrive/Colab-dump/Lab_Week10/MyModel-...'`
      - Use your drive or save it locally on the Colab server
  - Run cells **FT-2.2** to load the created dataset and create the data splits
  - Run cells **FT-2.3** to continue training the model
    - **Use GPU for this step**
    - You will need to run all the **FT** cells again if you need to change the runtime

Why do we set random seeds?

- The code sometimes has to select randomly a subset of the data:
  - Data creation: for creating data splits (e.g. when you balance the data splits)
  - Fine-tuning: for batching the training samples
- With different data splits or different batches, the results can be different!
- In summary
  - Keep track of seeds!
  - For more solid results, replicate an experiment several times with different seeds and average the scores

Train and Test data need to match!

- Always make sure that Train and Test data match in your experiments!
  - $\text{textifiedTriple}_{\text{Train}}$ == $\text{textifiedTriple}_{\text{Test}}$
  - $\text{candidateSentence}_{\text{Train}}$ == $\text{candidateSentence}_{\text{Test}}$
- This is already the case in today's exercise, nothing to change.

Create fine-tuning data

- **Edit** **FT-1.2** (2 cells for extracting and storing data) and **FT-1.4** (1 cell for selecting data subsets)
- **Run** all **FT-1** cells

Fine-tune model

- **Edit** **FT-2.1** and/or **FT-2.2** (1 cell each for fine-tuning parameters and/or random seed)
- **Run** all **FT-2** cells
  - Model is saved by default in `'/content/drive/MyDrive/Colab-dump/Lab_Week10/MyModel-...'`

Evaluate your updated model

- In summary: **paste** fine-tuned model path in **EV-2** and **run** **EV-2**, **EV-3** and **EV-4**.
  - For more details, go back to slides 25 and 26 and follow instructions
- Are you happy about the new score (**0.937 with baseline**)?

Design one or more experiments to carry out (+ expected outcome)! Some ideas:

- Modify the data
  - More or less negative and/or positive data?
  - More/less score categories?
  - Different shuffling/random seeds?
- Change parameters of fine-tuning
  - More/less epochs?
  - Different random seeds (e.g. how much variance in the scores with 5 different seeds)?