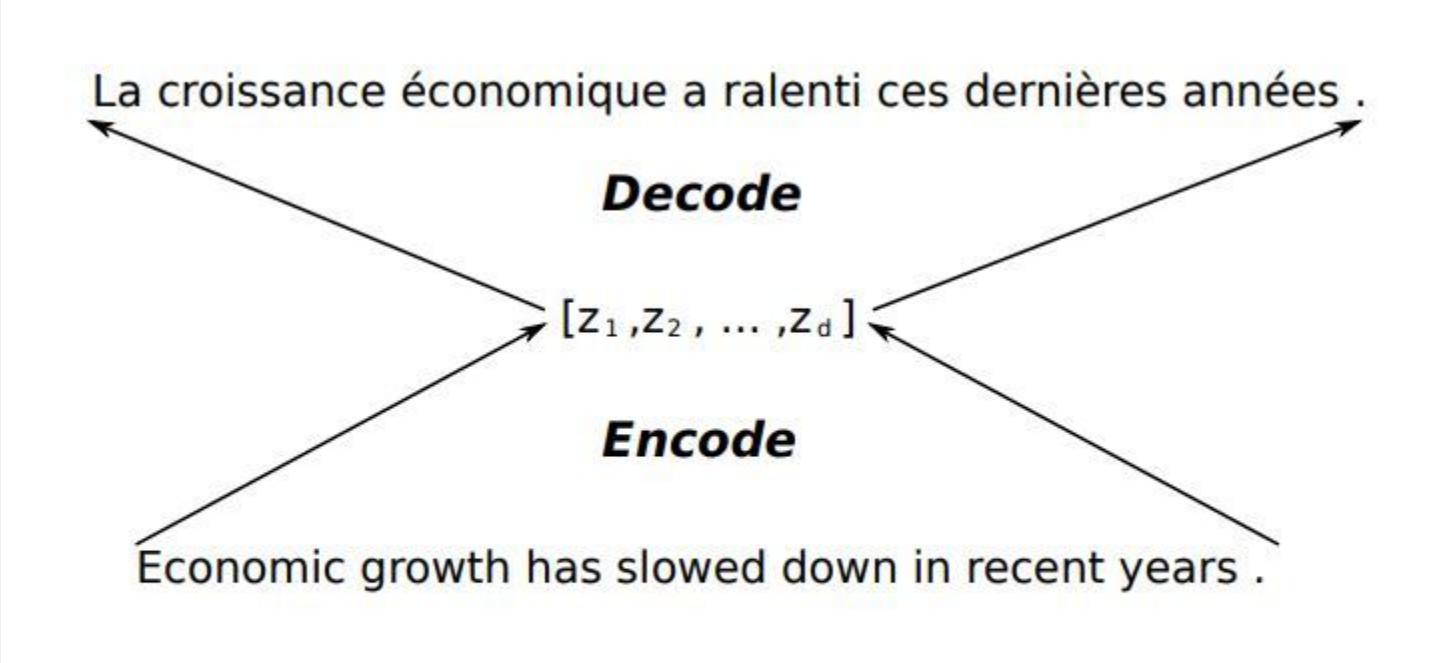
Practical AI Seminars

Week 4
Training your model

NMT



https://developer.nvidia.com/blog/introduction-neural-machine-translation-with-gpus/
https://developer.nvidia.com/blog/introduction-neural-machine-translation-gpus-part-2/
https://developer.nvidia.com/blog/introduction-neural-machine-translation-gpus-part-3/

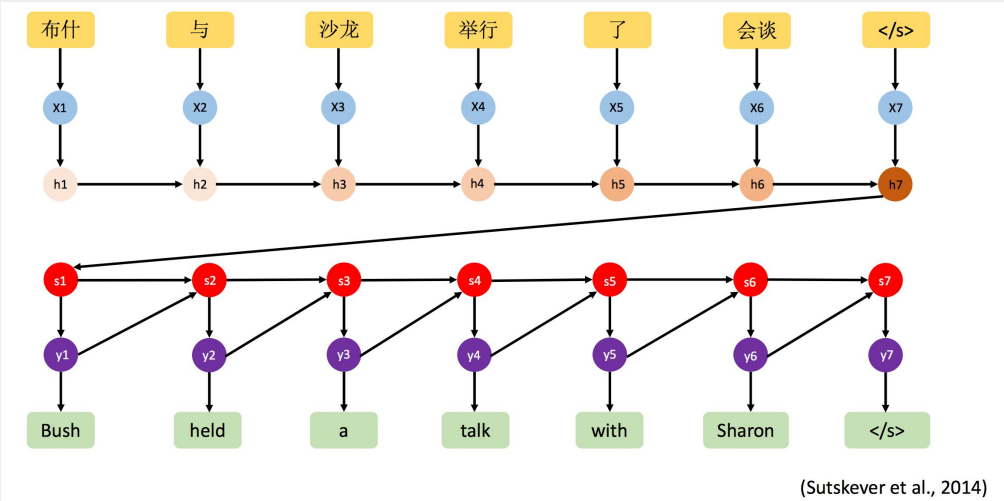
Neural Machine Translation

Encoder and decoder:

- Encoder reads an input sentence and converts it into a vector c
- Encoder reads word by word

- Decoder reads this vector c and generates a translation sentence
- Decoder generates one word at a time

- Encoder and decoder are neural networks
- Neural networks learn mathematical functions



Learning from parallel data

(src 1) When I woke up, I was sad.
(trg 1) Nuair a dhúisigh mé, bhí brón orm.

(src 2) I have to go to bed.
(trg 2) Caithfidh mé dul a chodladh.

(src 3) I love you.
(trg 3) Táim i ngrá leat.

(src 4) I love you.
(trg 4) Tá grá agam duit.

(src 5) Congratulations!
(trg 5) Comhghairdeas!

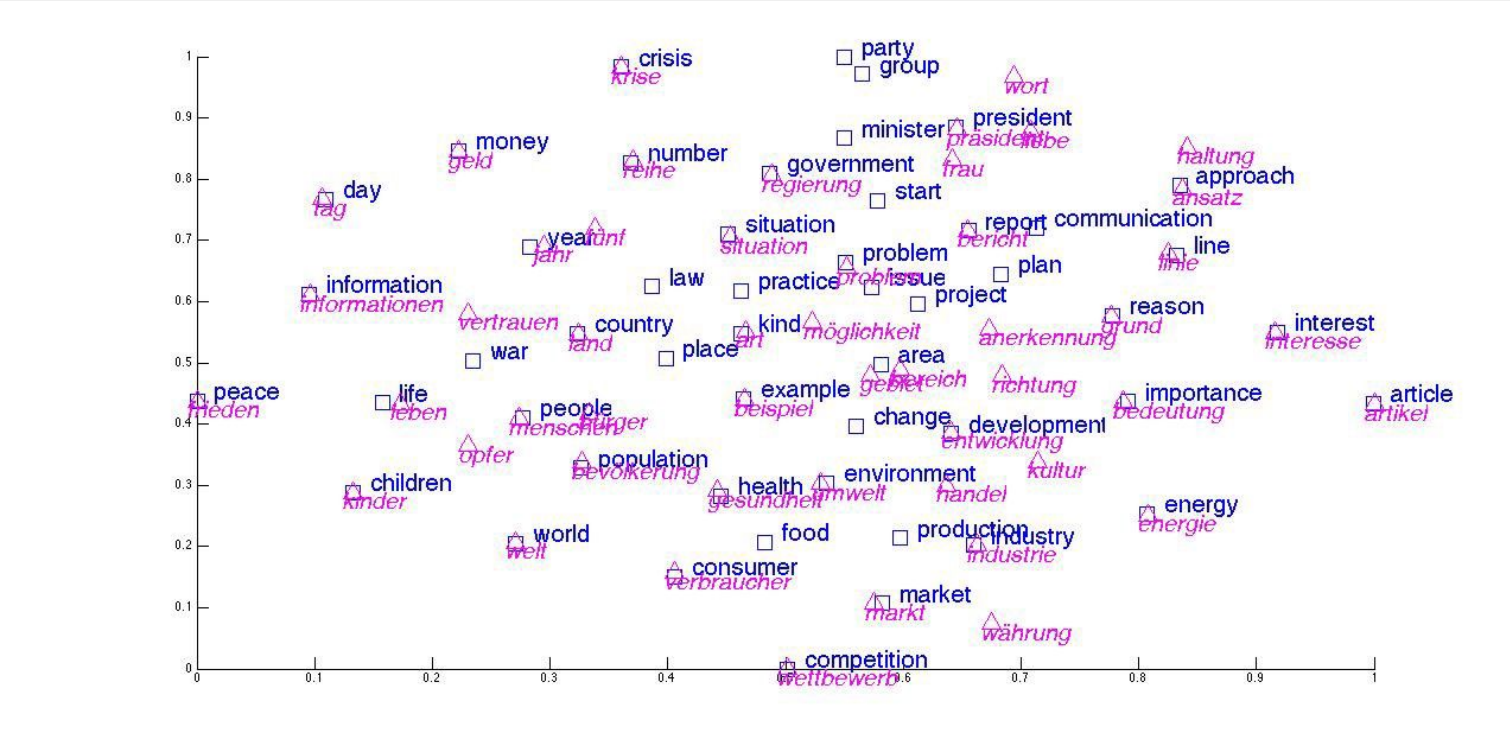
(src 6) This is a pun.
(trg 6) Is imeartas focal é seo.

(src 7) This is a pun.
(trg 7) Imeartas focal is ea é seo.

(src 8) You're an angel!
(trg 8) Is aingeal thú!

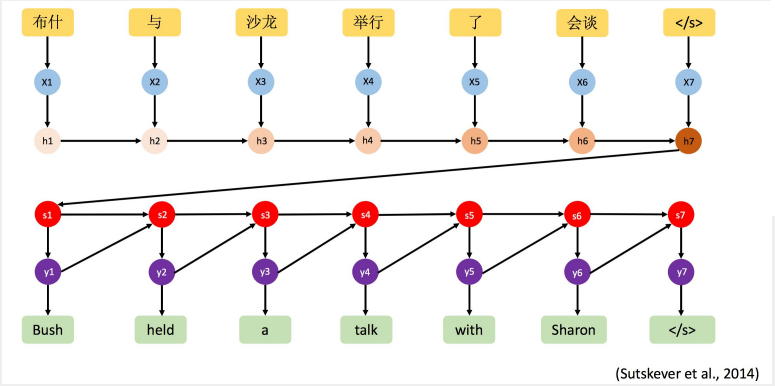
(src 9) I have a dream.
(trg 9) Tá aisling agam.

Learning from parallel data

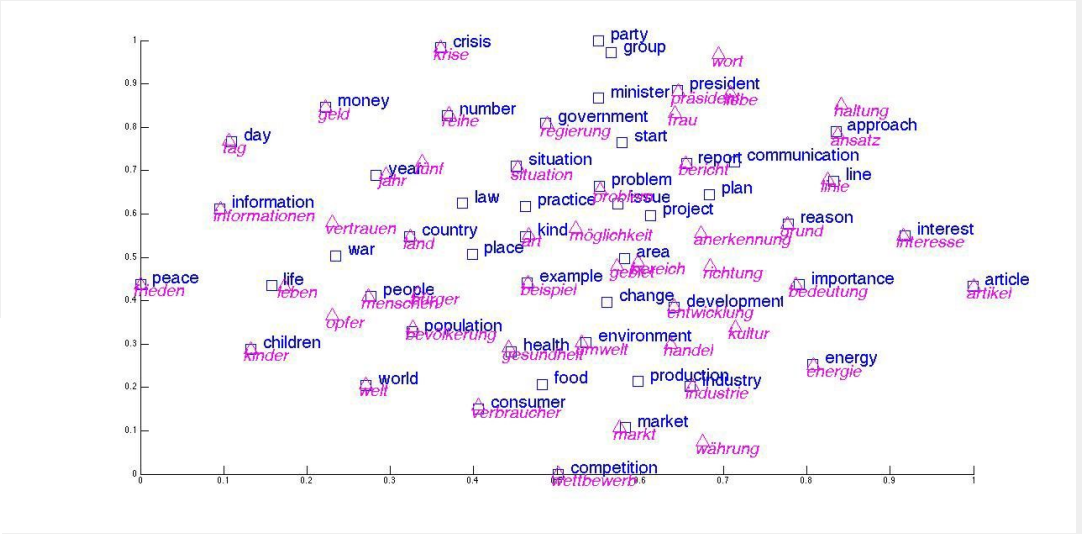


Translating

- For a given sentence in the source find the most similar one in the target
- Neural networks will compute the vector c
- Then will generate one word after another a target that minimizes the distance



(Sutskever et al., 2014)



Steps to train your own NMT system

1. Split into train, test and development sets
2. Tokenise and clean
3. Truecase
4. Create dictionaries
5. Train

Steps to train your own NMT system

1. **Split into train, test and development sets**
2. Tokenise and clean
3. Truecase
4. Create dictionaries
5. Train

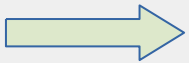
Steps to train your own NMT system

1. Split into train, test and development sets
2. **Tokenise and clean**
3. Truecase
4. Create dictionaries
5. Train

(src 1) When I woke up, I was sad.
(trg 1) Nuair a dhúisigh mé, bhí brón orm.

(src 2) I have to go to bed.
(trg 2) Caithfidh mé dul a chodladh.

(src 3) I love you.
(trg 3) Táim i ngrá leat.



(src 1) When I woke up , I was sad .
(trg 1) Nuair a dhúisigh mé , bhí brón orm .

(src 2) I have to go to bed .
(trg 2) Caithfidh mé dul a chodladh .

(src 3) I love you .
(trg 3) Táim i ngrá leat .

Steps to train your own NMT system

1. Split into train, test and development sets
2. Tokenise and clean
3. **Truecase**
4. Create dictionaries
5. Train

SKIP IT :)

Steps to train your own NMT system

1. Split into train, test and development sets
2. Tokenise and clean
3. Truecase
4. **Create dictionaries**
   a. **Byte-pair encoding (BPE)**

      Tall@ er

      Low@ est

      Tallest, Lower, Tall, Law, er, est,

      Taller and Lower

1. Train

(src 1) When I woke up , I was sad .
(trg 1) Nuair a dhúisigh mé , bhí brón orm .

(src 1) 1 2 3 4 5 2 6 7 8
(trg 1) 1 2 3 4 5 6 7 8 9

Steps to train your own NMT system

1. Split into train, test and development sets
2. Tokenise and clean
3. Truecase
4. Create dictionaries
5. **Train**

No GPU needed

GPU needed

Monitor power and carbon

1. nvidia-smi dmon
   nvidia-smi dmon -i 0 -s mpucv -d 1 -o TD > gpu.log &

1. https://github.com/lfwa/carbontracker

Resources

Data: https://opus.nlpl.eu/

NMT:

- OpenNMT (tutorial): https://github.com/ymoslem/OpenNMT-Tutorial
- JoeyNMT: https://github.com/joeynmt/joeynmt

Evaluation metrics for MT: https://github.com/mjpost/sacrebleu

Environment: Anaconda / virtualenv

Summarisation (some tutorial):
https://www.analyticsvidhya.com/blog/2023/07/build-a-text-summariser-using-llms-with-hugging-face/

LLMs: https://huggingface.co/models?pipeline_tag=summarization

Project organisation (suggested)

1. Folder structure:
    a. Data
        i. train, test, val splits in separate folders or in separate files
        ii. raw data better not be there
    b. Model
        i. keep intermediate models
        ii. don’t use weird labeling
    c. Logs
        i. GPU logs
        ii. Model logs
    d. Config: all config files
2. Preprocess off-line -> train online
3. SLURM scripts:
    a. One script per task
    b. Cheat sheet: https://www.carc.usc.edu/user-information/user-guides/hpc-basics/slurm-cheatsheet

SLURM example

```
#!/bin/bash
#SBATCH -p GPU # partition (queue)
#SBATCH -N 1 # number of nodes
#SBATCH -o slurm.%N.%j.out # STDOUT
#SBATCH -e slurm.%N.%j.err # STDERR
#SBATCH --gres=gpu:1
#SBATCH --mail-type=BEGIN,END,FAIL
#SBATCH --mail-user=d.shterionov@tilburguniversity.edu
#SBATCH -w byzantium

source activate SignON
cd /home/shterion/Projects/SignON/InterL/second-adaptable-pipeline
echo "STARTING NVIDIA DMON"
nvidia-smi dmon -i 0 -s mpucv -d 1 -o TD > gpu.log &
A="$!"
echo "STARTING TRAINING"
python train.py --mode=text2text args.json
kill $A
echo "DONE"
```