Natural Language Processing & HPC

Artificial Intelligence in Life Sciences

Antonio Miranda - Text Mining Unit, BSC - antonio.miranda@bsc.es

0. Intro to NLP

1. What we do in NLP

Can you read this?

Fine-tuning For each task, we append the relevant predictive layer on top of CamemBERT's Transformer architecture. Following the work done on BERT (Devlin et al., 2019), for sequence tagging and sequence labeling we append a linear layer respectively to the <s> special token and to the first subword token of each word. For dependency parsing, we plug a bi-affine graph predictor head as inspired by (Dozat and Manning, 2017) following the work done on multilingual parsing with BERT by (Kondratyuk, 2019). We refer the reader to these two articles for more details on this module.

We fine-tune independently CamemBERT for each task and each dataset. We optimise the model using the Adam optimiser (Kingma and Ba, 2014) with a fixed learning rate. We run a grid search on a combination of learning rates and batch sizes. We select the best model on the validation set out of the 30 first epochs.
Although this might push the performances even further, for all tasks except NLI, we don't apply any regularisation techniques such as weight decay, learning rate warm-up or discriminative fine-tuning. We show that fine-tuning CamemBERT in a straight-forward manner leads to state-of-the-art results on most tasks and outperforms the existing BERT-based models in most cases.

1. What we do in NLP

Can you read this?

第一章

起初神创造天地。
2 地是空虚混沌。渊面黑暗。神的灵运行在水面上。
3 神说，要有光，就有了光。
4 神看光是好的，就把光暗分开了。
5 神称光为昼，称暗为夜。有晚上，有早晨，这是头一日。
6 神说，诸水之间要有空气，将水分为上下。
7 神就造出空气，将空气以下的水，空气以上的水分开了。事就这样成了。
8 神称空气为天。有晚上，有早晨，是第二日。
9 神说，天下的水要聚在一处，使旱地露出来。事就这样成了。
10 神称旱地为地，称水的聚处为海。神看着是好的。
11 神说，地要发生青草，和结种子的菜蔬，并结果子的树木，各从其类，果子都包着核。事就这样成
了。
12 于是地发生了青草，和结种子的菜蔬，各从其类，并结果子的树木，各从其类，果子都包着核。神
看着是好的。
13 有晚上，有早晨，是第三日。
14 神说，天上要有光体，可以分昼夜，作记号，定节令，日子，年岁。
15 并要发光在天空，普照在地上。事就这样成了。

https://www.wordproject.org/bibles/parallel/d/chinese.htm

1. What we do in NLP

Can you find anything here?

http://www2.humanresourcesonline.net/truth-messy-desk-gallery/

1. What we do in NLP

Fine-tuning For each task, we append the relevant predictive layer on top of CamemBERT's Transformer architecture. Following the work done on BERT (Devlin et al., 2019), for sequence tagging and sequence labeling we append a linear layer respectively to the <s> special token and to the first subword token of each word. For dependency parsing, we plug a bi-affine graph predictor head as inspired by (Dozat and Manning, 2017) following the work done on multilingual parsing with BERT by (Kondratyuk, 2019). We refer the reader to these two articles for more details on this module.

We fine-tune independently CamemBERT for each task and each dataset. We optimise the model using the Adam optimiser (Kingma and Ba, 2014) with a fixed learning rate. We run a grid search on a combination of learning rates and batch sizes. We select the best model on the validation set out of the 30 first epochs.
Although this might push the performances even further, for all tasks except NLI, we don't apply any regularisation techniques such as weight decay, learning rate warm-up or discriminative fine-tuning. We show that fine-tuning CamemBERT in a straight-forward manner leads to state-of-the-art results on most tasks and outperforms the existing BERT-based models in most cases.

1. What we do in NLP

Entity extraction

Fine-tuning For each task, we append the relevant predictive layer on top of CamemBERT's Transformer architecture. Following the work done on BERT (Devlin et al., 2019), for sequence tagging and sequence labeling we append a linear layer respectively to the <s> special token and to the first subword token of each word. For dependency parsing, we plug a bi-affine graph predictor head as inspired by (Dozat and Manning, 2017) following the work done on multilingual parsing with BERT by (Kondratyuk, 2019). We refer the reader to these two articles for more details on this module.

We fine-tune independently CamemBERT for each task and each dataset. We optimise the model using the Adam optimiser (Kingma and Ba, 2014) with a fixed learning rate. We run a grid search on a combination of learning rates and batch sizes. We select the best model on the validation set out of the 30 first epochs.
Although this might push the performances even further, for all tasks except NLI, we don't apply any regularisation techniques such as weight decay, learning rate warm-up or discriminative fine-tuning. We show that fine-tuning CamemBERT in a straight-forward manner leads to state-of-the-art results on most tasks and outperforms the existing BERT-based models in most cases.

1. What we do in NLP

第一章

起初神创造天地。
2 地是空虚混沌。渊面黑暗。神的灵运行在水面上。
3 神说，要有光，就有了光。
4 神看光是好的，就把光暗分开了。
5 神称光为昼，称暗为夜。有晚上，有早晨，这是头一日。
6 神说，诸水之间要有空气，将水分为上下。
7 神就造出空气，将空气以下的水，空气以上的水分开了。事就这样成了。
8 神称空气为天。有晚上，有早晨，是第二日。
9 神说，天下的水要聚在一处，使旱地露出来。事就这样成了。
10 神称旱地为地，称水的聚处为海。神看着是好的。
11 神说，地要发生青草，和结种子的菜蔬，并结果子的树木，各从其类，果子都包着核。事就这样成
了。
12 于是地发生了青草，和结种子的菜蔬，各从其类，并结果子的树木，各从其类，果子都包着核。神
看着是好的。
13 有晚上，有早晨，是第三日。
14 神说，天上要有光体，可以分昼夜，作记号，定节令，日子，年岁。
15 并要发光在天空，普照在地上。事就这样成了。

https://www.wordproject.org/bibles/parallel/d/chinese.htm

1. What we do in NLP

Machine translation

CHAPTER 1

In the beginning God created the heaven and the earth.
2 And the earth was without form, and void; and darkness was upon the face of the deep. And
the Spirit of God moved upon the face of the waters.
3 And God said, Let there be light: and there was light.
4 And God saw the light, that it was good: and God divided the light from the darkness.
5 And God called the light Day, and the darkness he called Night. And the evening and the morning
were the first day.
6 And God said, Let there be a firmament in the midst of the waters, and let it divide the waters
from the waters.
7 And God made the firmament, and divided the waters which were under the firmament from the
waters which were above the firmament: and it was so.
8 And God called the firmament Heaven. And the evening and the morning were the second day.
9 And God said, Let the waters under the heaven be gathered together unto one place, and let the
dry land appear: and it was so.
10 And God called the dry land Earth; and the gathering together of the waters called he Seas: and
God saw that it was good.
11 And God said, Let the earth bring forth grass, the herb yielding seed, and the fruit tree yielding
fruit after his kind, whose seed is in itself, upon the earth: and it was so.

https://www.wordproject.org/bibles/parallel/d/chinese.htm

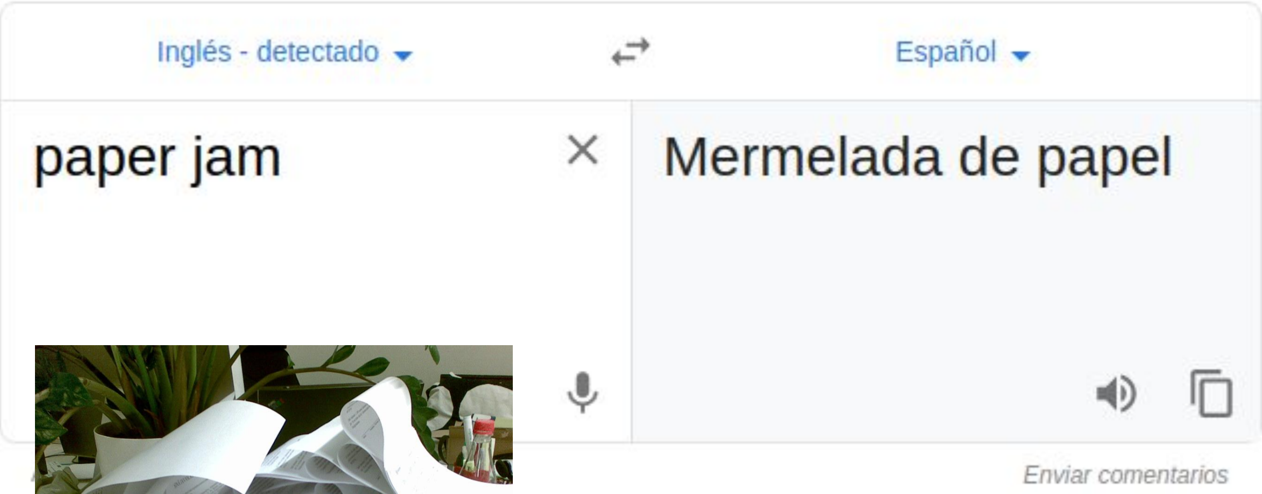
1. What we do in NLP

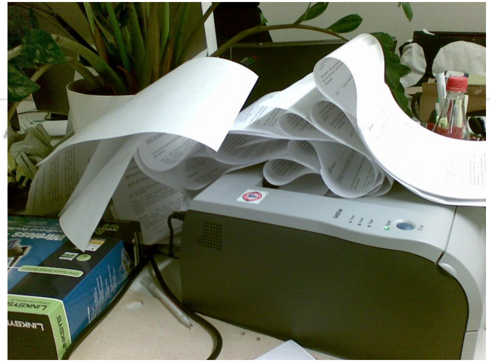
http://www2.humanresourcesonline.net/truth-messy-desk-gallery/

1. What we do in NLP

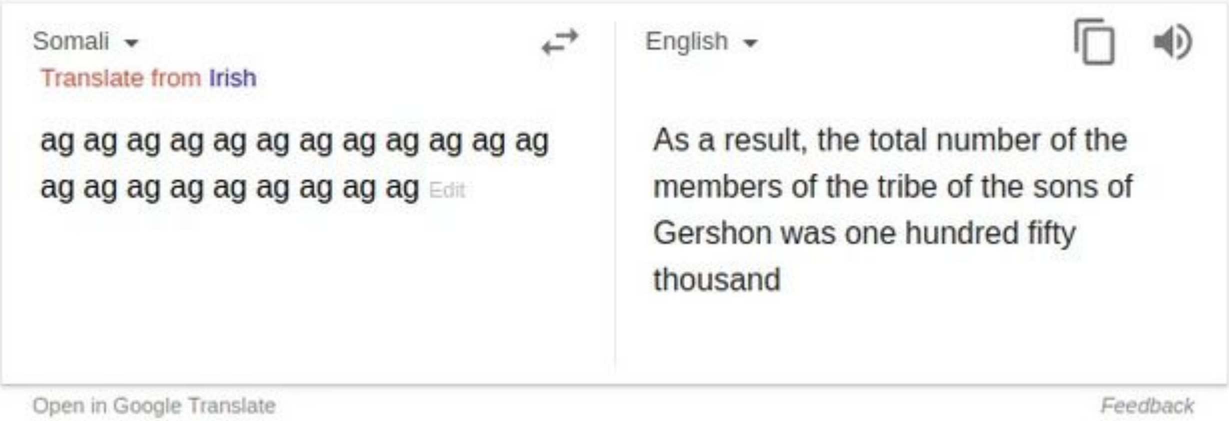
Document indexing

2. Why Google is not enough





2. Why Google is not enough



3. What NLP used to be

- Example use case: is this code Python or Java?

```
public class Test {
    public static void main(String args[]) {
        String array[] = {"Hello, World", "Hi there, Everyone", "6"};
        for (String i : array) {
          System.out.println(i);
        }
    }
}
```

3. What NLP used to be

- Example use case: is this code Python or Java?
- Which rules would you create to determine whether the code is Java or Python?

```
public class Test {
    public static void main(String args[]) {
        String array[] = {"Hello, World", "Hi there, Everyone", "6"};
        for (String i : array) {
          System.out.println(i);
        }
    }
}
```

3. What NLP used to be

Characteristics:

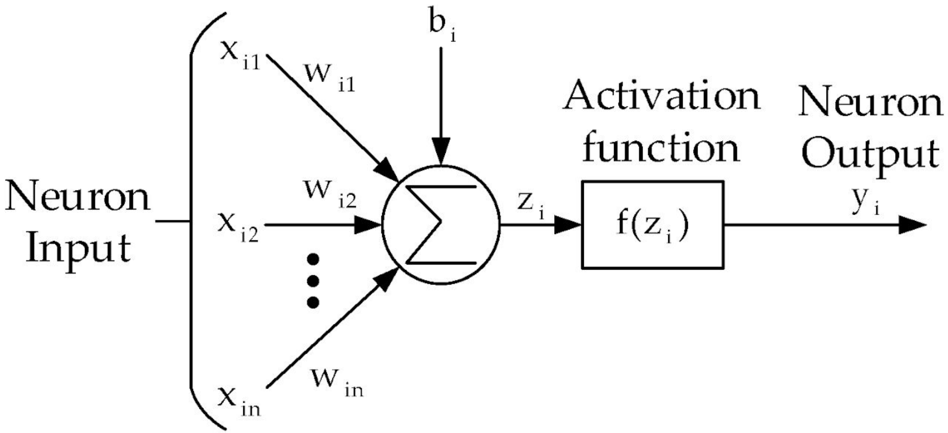
- **Rule-based**
- Focus on **pattern-matching**
- May have **high performance in specific use cases**, but often do not generalize.

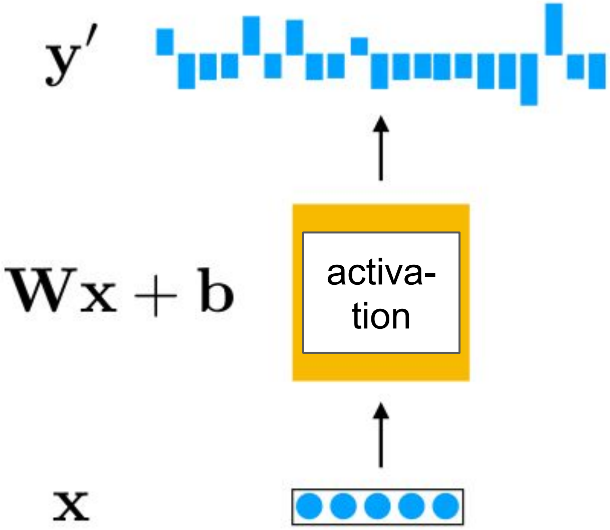
```
import tables, strutils

var wordFrequencies = initCountTable[string]()

for line in stdin.lines:
  for word in line.split(", "):
    wordFrequencies.inc(word)
```

Is this Nim of Python?
Can we use all the previous rules?

4. What NLP is today

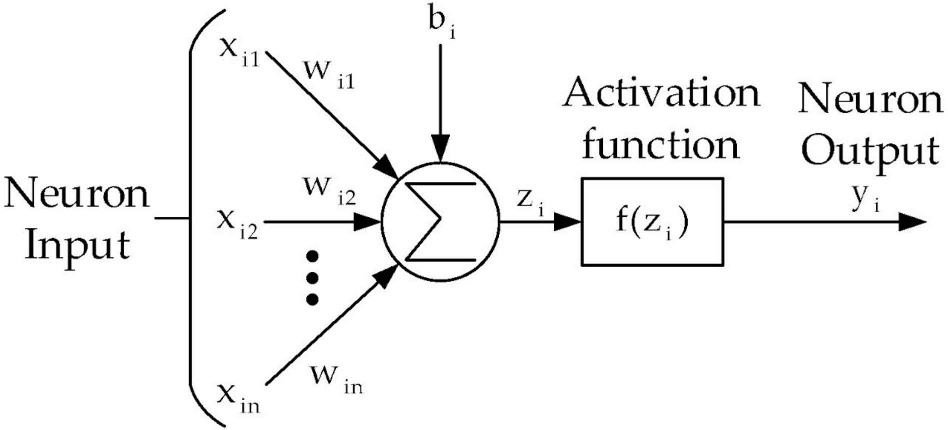
Neural networks

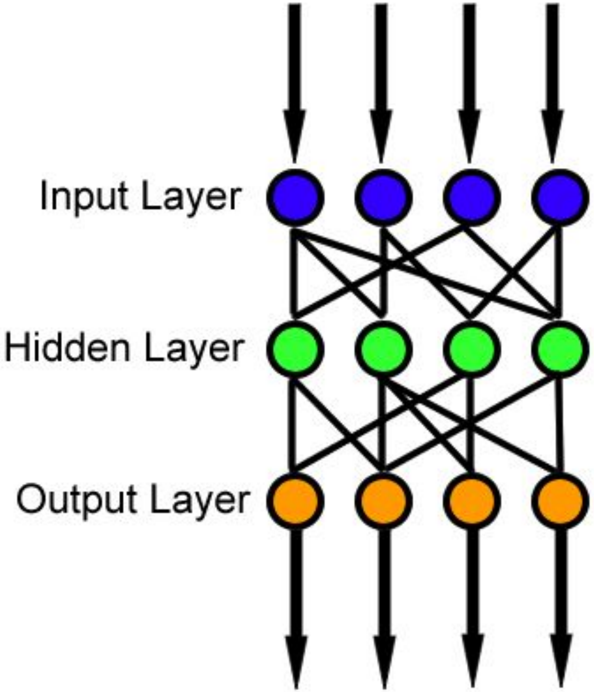




4. What NLP is today

Neural networks - FFN





4. What NLP is today

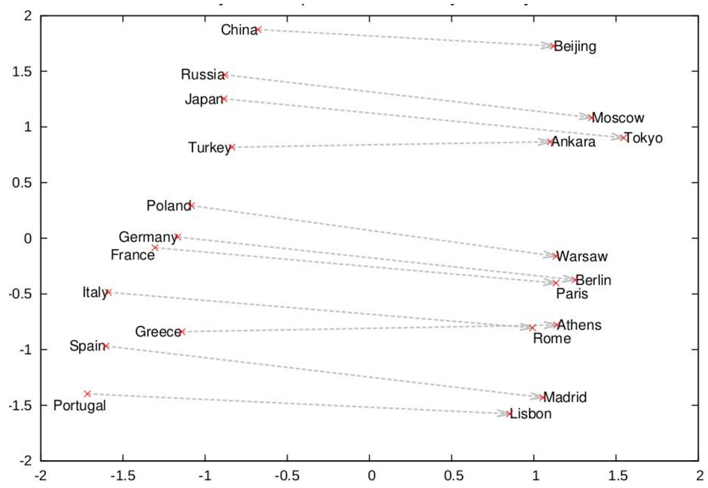
AI: Machine Learning and **Deep Learning** algorithms

1. Work with vectors
2. Need tons of data
3. Training may be expensive
4. Promise more generalization...

4. What NLP is today

1. Work with vectors: **words are transformed into vectors** through a look-up table (**word embeddings**)

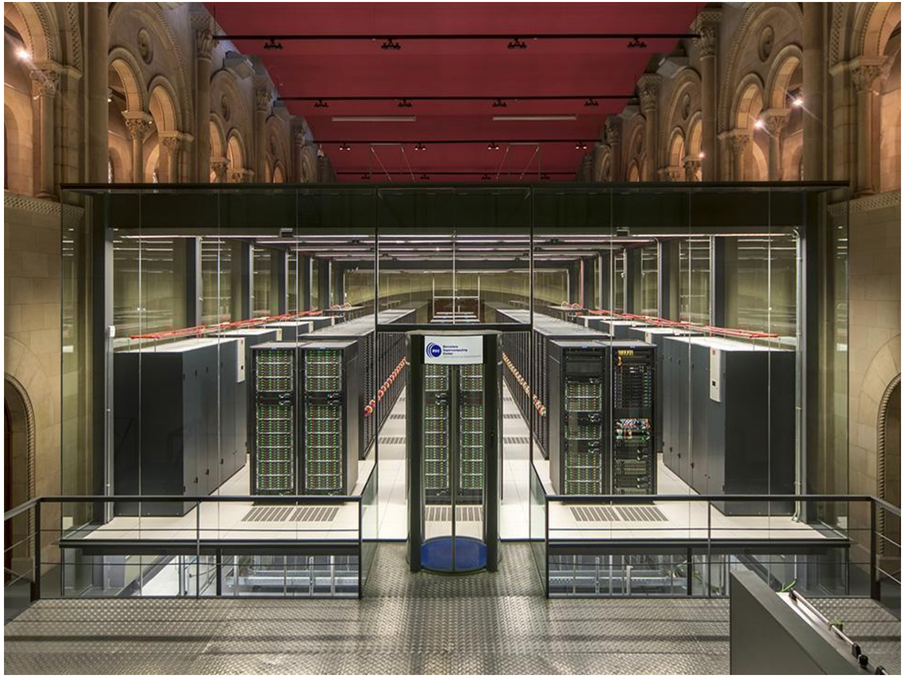
	X1	X2	...	Xn
apple	0.2	0.0	...	-0.3
doctor	0.5	-0.9	...	0.11
injury	-1.5	0.4	...	-0.3
dog	-0.11	0.6	...	-0.3



4. What NLP is today

2. Need tons of data: **Need HPC**

3. Training may be **VERY** expensive (GB of data, thousands of millions of parameters, weeks of computation): **Need HPC**



4. What NLP is today

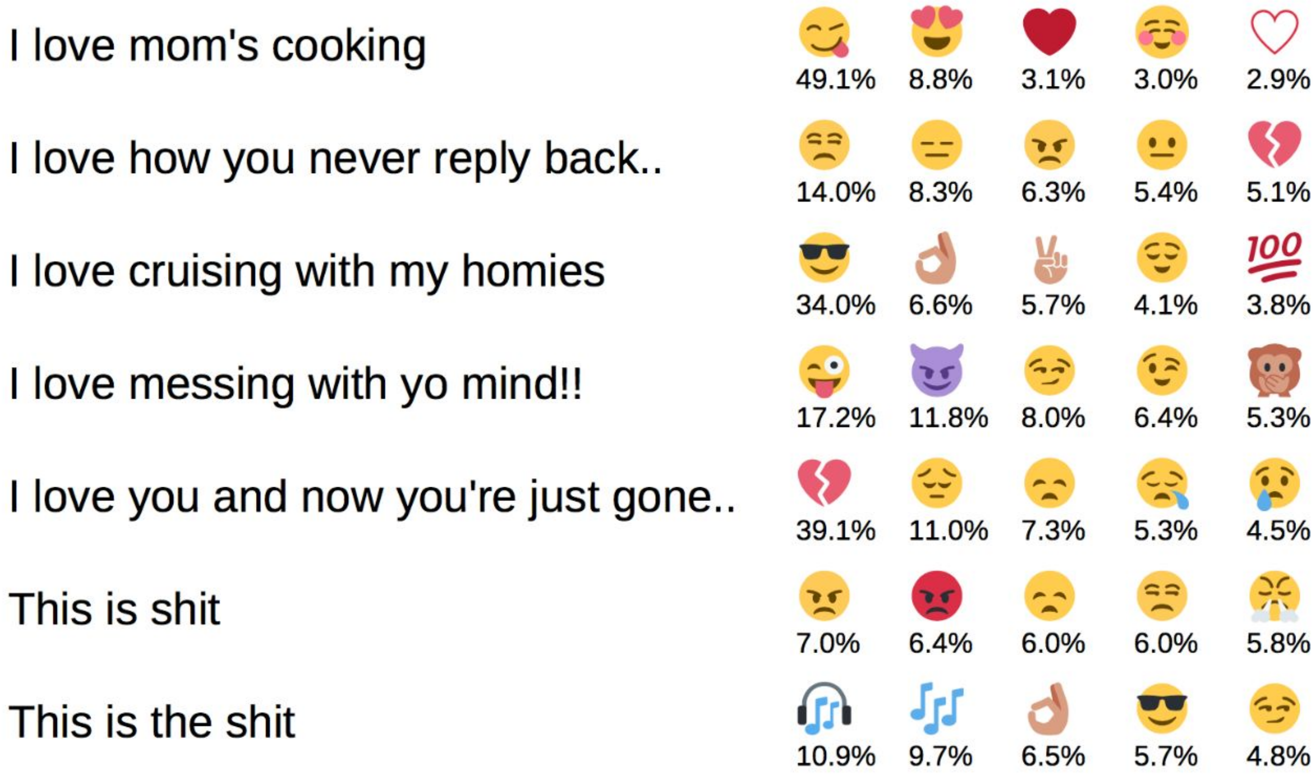
3. Training may be VERY expensive (lots of data, thousands of millions of parameters)

- **Reuse models:**
  - DeepMoji: predict emoji given the text of a tweet.

INPUT:
2 días sin salir de mi casa grabando #ROSALIA me voy a quedar loco

PREDICTION:

4. What NLP is today



Is the model "understanding" something deep?

Do you think we can **re-use** the model for anything else? **Irony** detection? **Sentiment** detection?

4. What NLP is today

3. Training may be VERY expensive (lots of data, thousands of millions of parameters)

- **Reuse models: transfer learning**
    - Interesting idea: train the algorithm to *“speak”*! If it understands how to “speak”, it will understand language!

1. Introduction to language models

The language modeling task

- Classical NLP task for learning the probability distribution of word sequences in a given language:

  P(word1, word2, …, wordN)

- **Advantages:**
  - **Unsupervised** task (tons of data available)
  - Well-known evaluation metrics (perplexity)
  - **Makes sense?**

Motivation

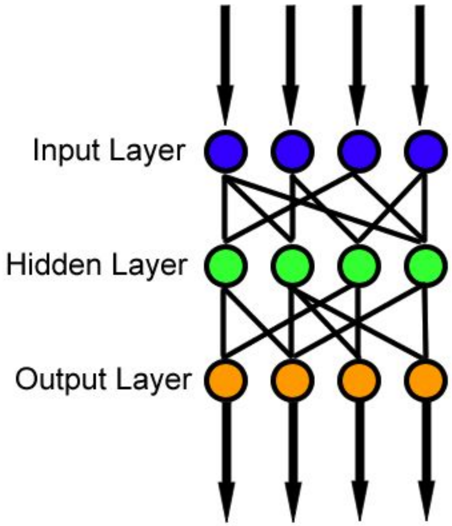
- **Direct app.**:
  - Selecting the most probable sentence:
    - Speech recognition: P(no idea) >>> P (no eye deer)
    - Machine translation: P(tall man) >>> P(large man)
  - Predicting the next word:
    - Predictive text: P(next word | word1, word2,...)

- **Indirect app.: transfer learning** capabilities.

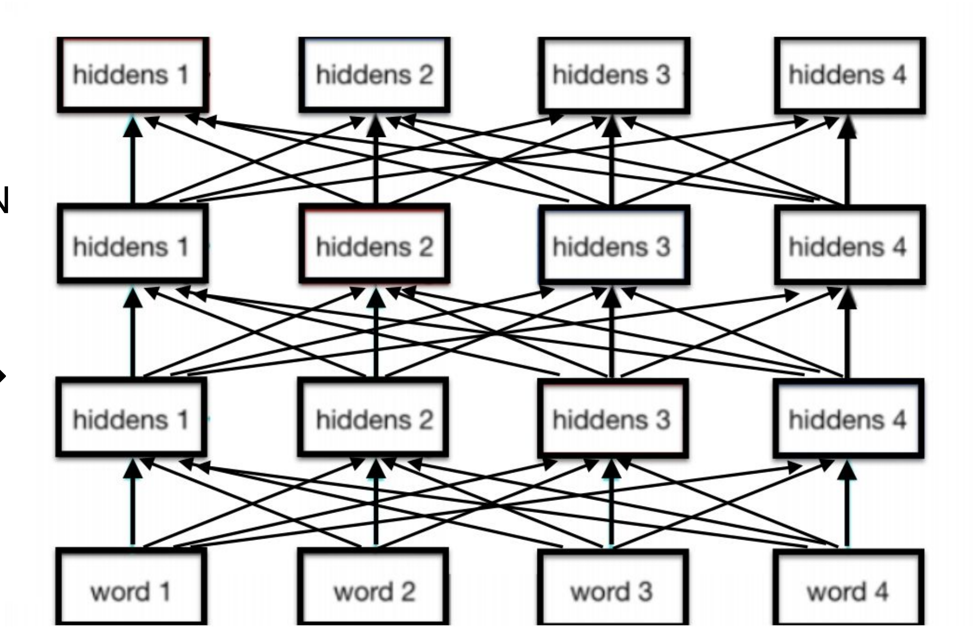
How? Language modeling architectures

- Statistical language modeling
- Neural language modeling:
  - Recurrent neural networks: LSTMs, GRUs…
  - Convolutional Neural Networks:
  - **Transformers!**

Transformer

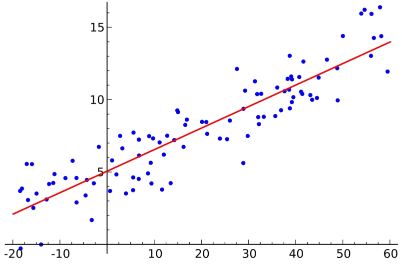


MASSIVE FFN stacked with attention



GPT-2 (2019)

OpenAI



- Parameters: **1.5 Billion** (linear regression has 2...)

- Training:
  - **8 million web pages**
  - **Predict each token given the previous ones**. Example: “The dog is playing.”
    - Input: “The” -> Predict: “dog”.
    - Input: “The dog” -> Predict “is”.
    - Input: “The dog is” -> Predict “playing”.
    - The same for very large paragraphs (**long-range dependencies!**)

SYSTEM PROMPT (HUMAN-WRITTEN)

In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

MODEL COMPLETION (MACHINE-WRITTEN, 10 TRIES)

The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez.

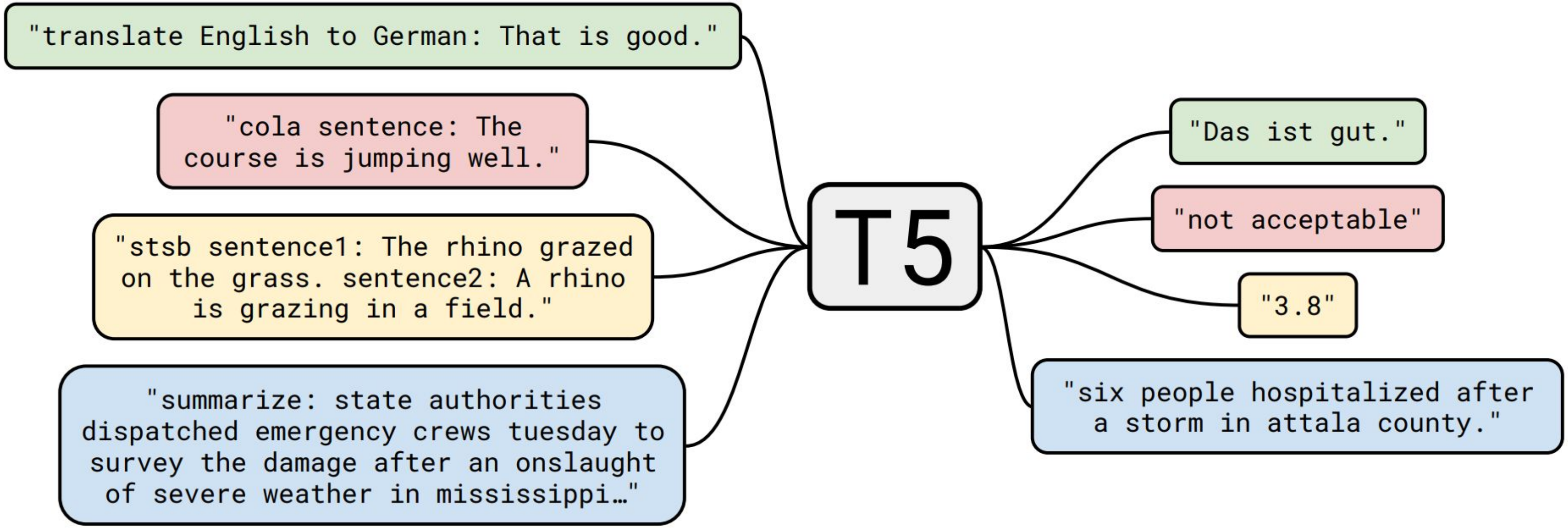
Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns.

BERT (2018)

- Parameters: **340M**

- Training:
  - ALL Wikipedia and a book corpus: **3.3B words.**
  - Task 1: **Predicting randomly masked tokens**:
    - Input: “The **[MASK]** is playing” -> Predict: “The **dog** is playing”
    - Input: “**[MASK]** dog **[MASK]** playing” -> Predict: “**The** dog **is** playing”
  - Task 2: **Next sentence prediction** (binary classification)

T5 (2019)



"translate English to German: That is good."
"cola sentence: The course is jumping well."
"stsb sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."
"summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi…"
T5
"Das ist gut."
"not acceptable"
"3.8"
"six people hospitalized after a storm in attala county."

Transformer language models

- Realization 1: The algorithms from **language modeling task** can be applied **to several NLP tasks**:

	Rank	Name	Model	URL	Score	CoLA	SST-2	MRPC	STS-B	QQP	MNLI-m	MNLI-mm	QNLI	RTE	WNLI	AX
	1	T5 Team - Google	T5		90.3	71.6	97.5	92.8/90.4	93.1/92.8	75.1/90.6	92.2	91.9	96.9	92.8	94.5	53.1
	2	ERNIE Team - Baidu	ERNIE		90.1	72.8	97.5	93.2/91.0	92.9/92.5	75.2/90.8	91.2	90.8	96.1	90.9	94.5	49.4
	3	Microsoft D365 AI & MSR AI & GATECH	MT-DNN-SMART		89.9	69.5	97.5	93.7/91.6	92.9/92.5	73.9/90.2	91.0	90.8	99.2	89.7	94.5	50.2
+	4	王玮	ALICE v2 large ensemble (Alibaba DAMO NLP)		89.7	73.2	97.1	93.9/91.9	93.0/92.5	74.8/91.0	90.8	90.6	95.9	87.4	94.5	48.7
+	5	Microsoft D365 AI & UMD	FreeLB-RoBERTa (ensemble)		88.4	68.0	96.8	93.1/90.8	92.3/92.1	74.8/90.3	91.1	90.7	95.6	88.7	89.0	50.1
	6	Junjie Yang	HIRE-RoBERTa		88.3	68.6	97.1	93.0/90.7	92.4/92.0	74.3/90.2	90.7	90.4	95.5	87.9	89.0	49.3
	7	Facebook AI	RoBERTa		88.1	67.8	96.7	92.3/89.8	92.2/91.9	74.3/90.2	90.8	90.2	95.4	88.2	89.0	48.7
+	8	Microsoft D365 AI & MSR AI	MT-DNN-ensemble		87.6	68.4	96.5	92.7/90.3	91.1/90.7	73.7/89.9	87.9	87.4	96.0	86.3	89.0	42.8
	9	GLUE Human Baselines	GLUE Human Baselines		87.1	66.4	97.8	86.3/80.8	92.7/92.6	59.5/80.4	92.0	92.8	91.2	93.6	95.9	-

Transformer language models

- Realization 1: The algorithms from **language modeling task** can be applied **to several NLP tasks**:

	Rank	Name	Model	URL	Score	CoLA	SST-2	MRPC	STS-B	QQP	MNLI-m	MNLI-mm	QNLI	RTE	WNLI	AX
	1	T5 Team - Google	T5		90.3	71.6	97.5	92.8/90.4	93.1/92.8	75.1/90.6	92.2	91.9	96.9	92.8	94.5	53.1
	2	ERNIE Team - Baidu	ERNIE		90.1	72.8	97.5	93.2/91.0	92.9/92.5	75.2/90.8	91.2	90.8	96.1	90.9	94.5	49.4
	3	Microsoft D365 AI & MSR AI & GATECH	MT-DNN-SMART		89.9	69.5	97.5	93.7/91.6	92.9/92.5	73.9/90.2	91.0	90.8	99.2	89.7	94.5	50.2
+	4	王玮	ALICE v2 large ensemble (Alibaba DAMO NLP)		89.7	73.2	97.1	93.9/91.9	93.0/92.5	74.8/91.0	90.8	90.6	95.9	87.4	94.5	48.7
+	5	Microsoft D365 AI & UMD	FreeLB-RoBERTa (ensemble)		88.4	68.0	96.8	93.1/90.8	92.3/92.1	74.8/90.3	91.1	90.7	95.6	88.7	89.0	50.1
	6	Junjie Yang	HIRE-RoBERTa		88.3	68.6	97.1	93.0/90.7	92.4/92.0	74.3/90.2	90.7	90.4	95.5	87.9	89.0	49.3
	7	Facebook AI	RoBERTa		88.1	67.8	96.7	92.3/89.8	92.2/91.9	74.3/90.2	90.8	90.2	95.4	88.2	89.0	48.7
+	8	Microsoft D365 AI & MSR AI	MT-DNN-ensemble		87.6	68.4	96.5	92.7/90.3	91.1/90.7	73.7/89.9	87.9	87.4	96.0	86.3	89.0	42.8
	9	GLUE Human Baselines	GLUE Human Baselines		87.1	66.4	97.8	86.3/80.8	92.7/92.6	59.5/80.4	92.0	92.8	91.2	93.6	95.9	-

Transformer language models

- Realization 2: **More data + bigger model = better results.**
  - GPT2 >> GPT: 10x more data and parameters.
  - BERT Large >> BERT Base: 3x more parameters
  - T5 is the best of all of them: **C O L O S S A L** model: 11 BILLION parameters

“it is often possible to achieve **better** performance simply by training a **larger model** on a **larger dataset**”

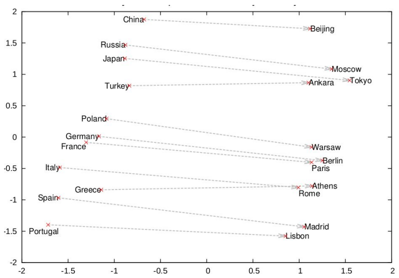
Google team

The Transformer Space Race (2018-?)

- Extreme competition between big AI players for building large Transformer language models. Massive models. *Massive* as in **M A S S I V E**.
- Models:
  - Based on BERT:
    - Generic improvements:
      - RoBERTa for “Robustly Optimized **BERT** approach”
      - DilBERT
      - ...
    - Language specific improvements:
      - CamemBERT
      - ...
  - UMLFiT: Universal Language Model Fine-tuning for Text Classification
  - ...

To sum up: Contextual embeddings

- *Vanilla* word embeddings do not take into account the context.



- Contextual embeddings (transformer models) *do* (obviously :))

“Language models are the new embeddings”

2. How to use transformer models?

Using Transformer language models

Options

- Train from scratch
- Transfer learning

Train from scratch

1. Pick up a given **architecture** (eg. BERT)
    a. Alternatively, modify the architecture (very hot research area)
2. **Preprocess your data**:
    a. Cleaning
    b. Recommended to split words into subwords
3. **Fit** the model with your data.
4. **Profit**!

Train from scratch

- We are building a **resource other people can benefit from**!

- But:
  - **Computationally expensive**
  - The **corpus** should be ***huge***

Transfer learning

1. Pick up a **pre-trained** model (eg. BERT already trained by Google)

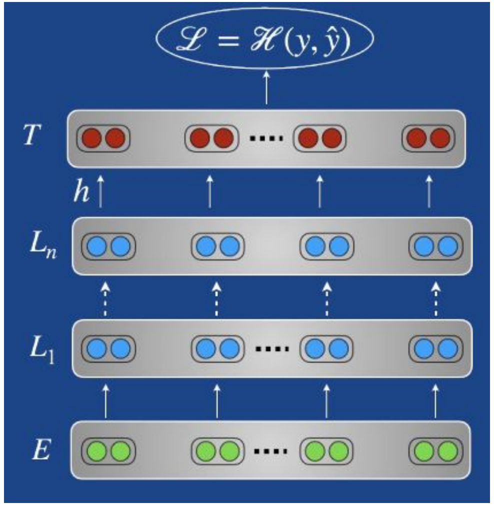
2. Preprocess your data...

3. Use it with your small data! Options:

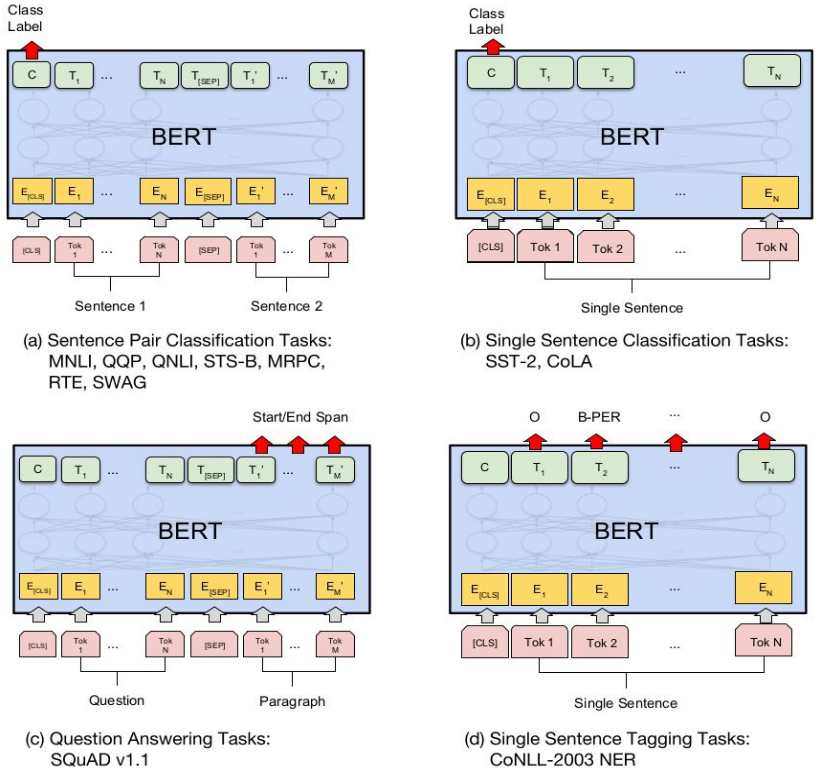
    a. Weight freezing: Add your own layers on the top!
    b. Fine tuning with your data + add your own layers!
    c. Zero shot?

Classifica
tion layer

BERT



How to do transfer learning? BERT example



Recommended libraries

- Facebook's Fairseq
- Huggingface's 🤗 Transformers
- FastAI/UMLFiT

3. Future work at BSC?

Low-hanging fruit?

- Steps to get published:
  1. Pick an NLP task of your choice
  2. Apply transfer learning from BERT
  3. Easily improve SOTA
  4. Profit!
- But:
  - Is it that easy?
  - What is our competitive advantage?

Why is BSC the right place to do more?

- High Performance Computing
- Data
- Mission to build resources for other researchers or institutions

Possible ideas

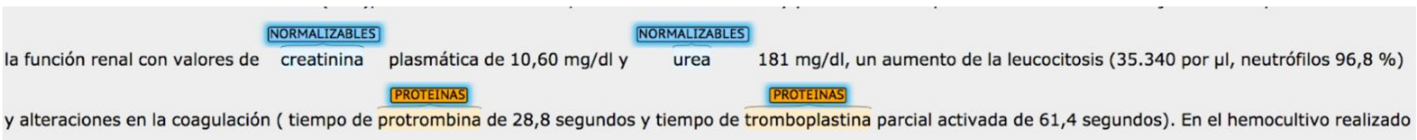
- Build a **Spanish** (Catalan, Basque…?) language model
  Loading
- Build a **domain-specific** language model (medical/bio)
  Loading
- **Investigate** approaches for domain adaptation for language models
- Conduct research to **improve** compute and data **efficiency**
  Loading

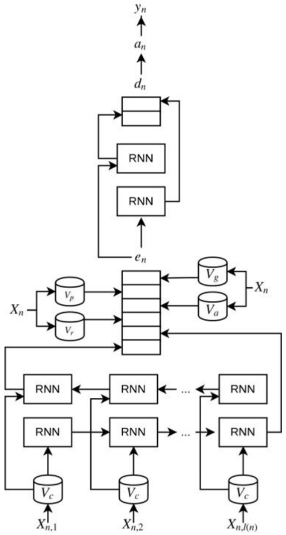
5. Why is this presentation subtitled “Artificial Intelligence in Life Sciences”?

At TeMU we do...

1. Clinical word embeddings (github.com/PlanTL-SANIDAD/Embeddings)

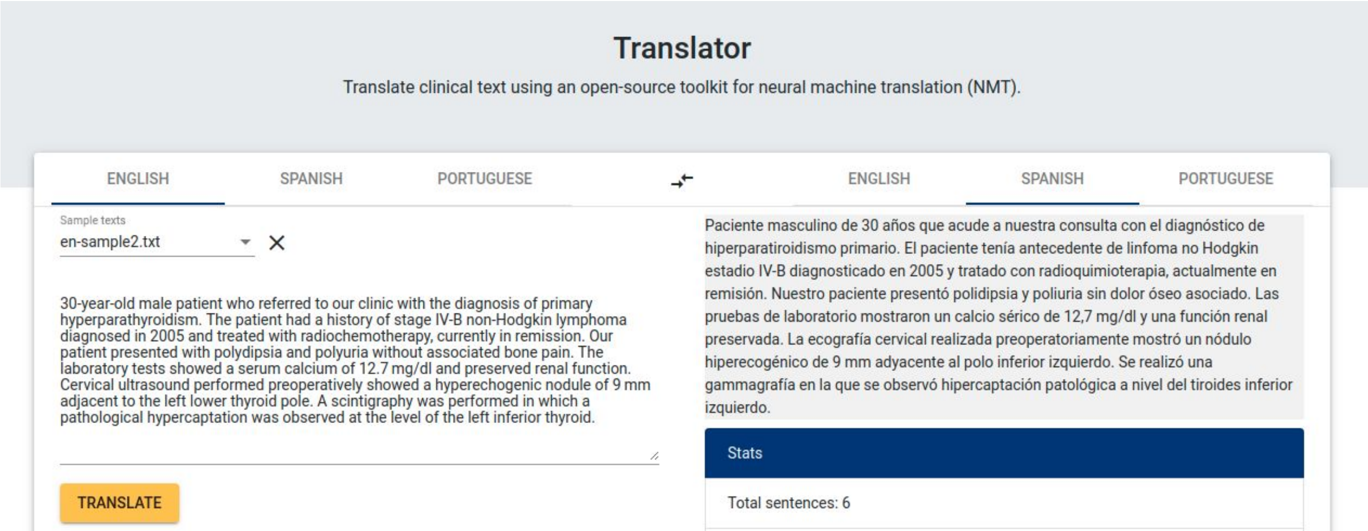
2. Clinical entity extractors: PharmaCoNER Tagger (http://temu.bsc.es/pharmaconer/)





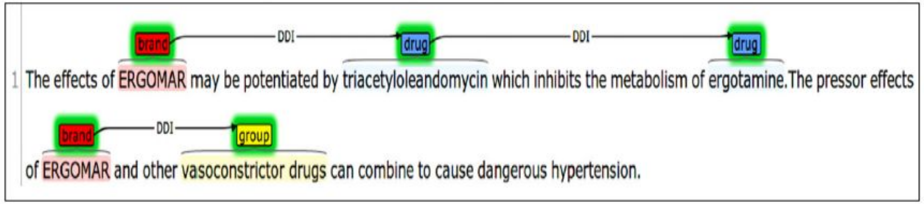
At TeMU we do...

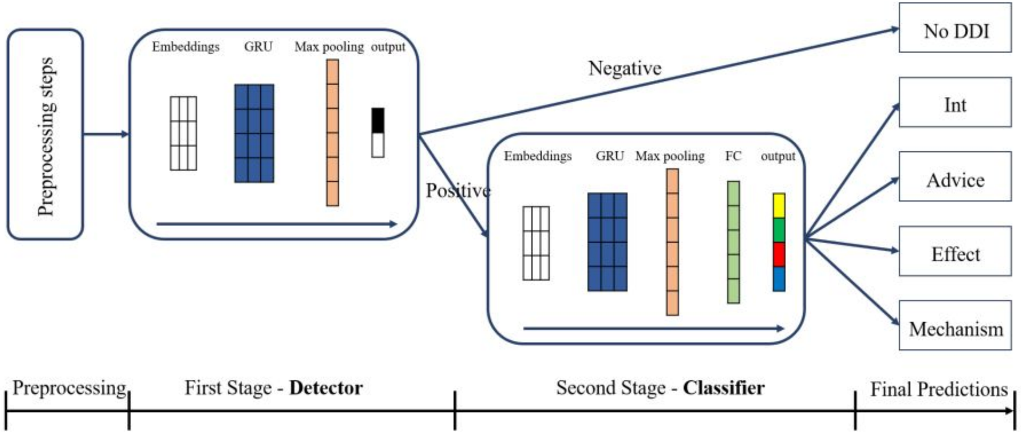
3. Medical Translator



At TeMU we do...

4. Drug-drug interactions detector





At TeMU we do...

5. ONGOING: Clinical BERT, Catalan BERT...

6. And many more! Ictusnet, meddocan, codiesp...

Ictusnet	Ictus report information extraction	
Meddocan	Clinical text anonymization	http://temu.bsc.es/meddocan/
Codiesp	Clinical text indexing	http://temu.bsc.es/codiesp/

http://temu.bsc.es/

antonio.miranda@bsc.es

Thanks to:
Jordi Armengol (jordi.armengol@bsc.es)