

Traducción automática neuronal e inteligencia artificial: entrenamiento, evaluación e integración

1. Introducción a la inteligencia artificial para la traducción

Antoni Oliver (aoliverg@uoc.edu)

Presentación del seminario

Traducción automática neuronal e inteligencia artificial: entrenamiento, evaluación e integración

- [WIKI](#)
- [Repositorio](#)

1. Introducción a la inteligencia artificial para la traducción

Traducción automática neuronal e inteligencia artificial: principios teóricos

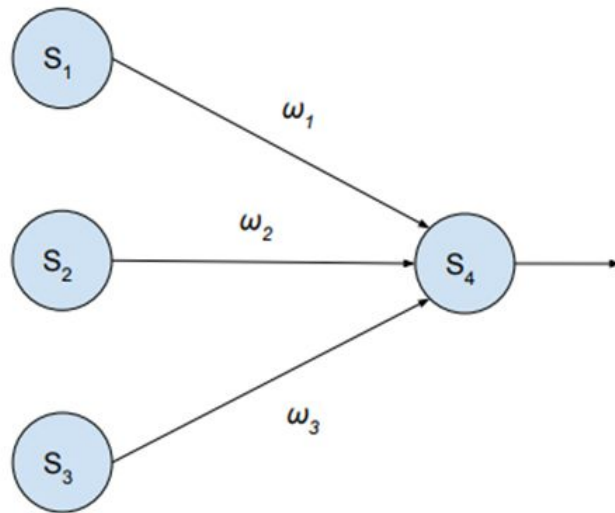
Tipos de sistemas de traducción automática

- Basado en reglas
- Basado en corpus

Historia de la traducción automática

1930	1940	1950	1960		1970	1980	1990	2000	2010	2020		
Precusores y pioneros			Grandes expectativas	ALPAC	Década silenciosa	Sistemas operativos y comerciales	TA estadística		TA neuronal	LUM's		

Neurona artificial

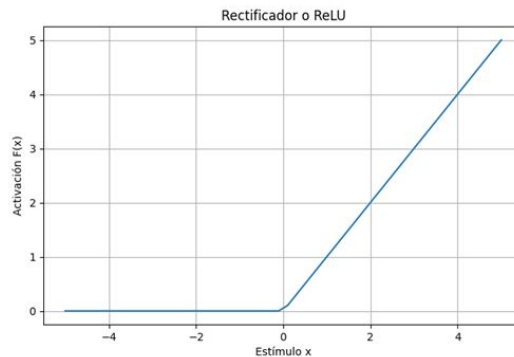
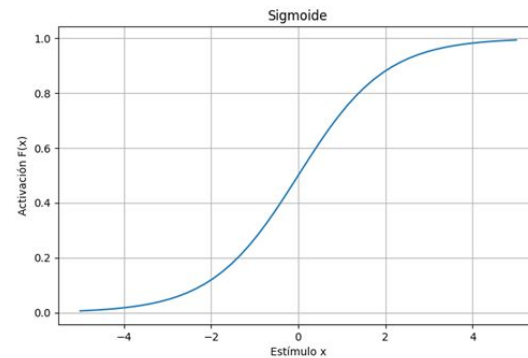
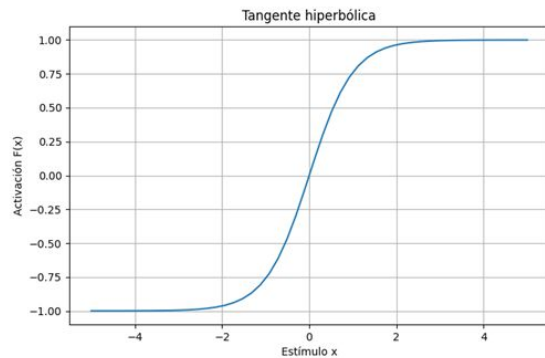


$$S_4 = F (\omega_1 \times S_1 + \omega_2 \times S_2 + \omega_3 \times S_3)$$

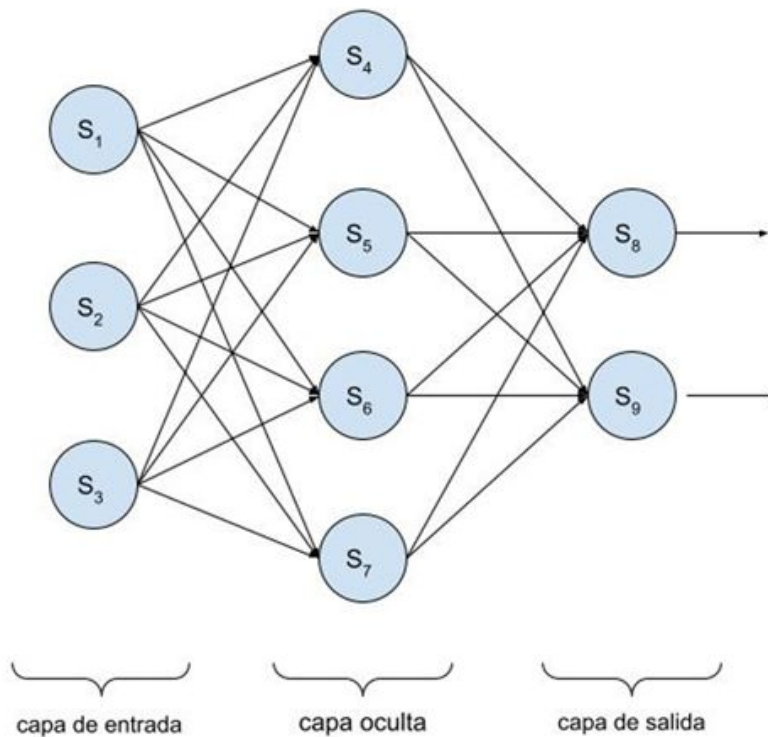
Funciones de activación

Funció	Fórmula	Rang
Tangent hiperbòlica	$\tanh(x) = \frac{\sinh(x)}{\cosh(x)}$ $= \frac{e^x - e^{-x}}{e^x + e^{-x}}$	de -1 a +1
Sigmoide	$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}}$	de 0 a +1
Rectificador o ReLU	$\text{relu}(x) = \max(0, x)$	de 0 a ∞

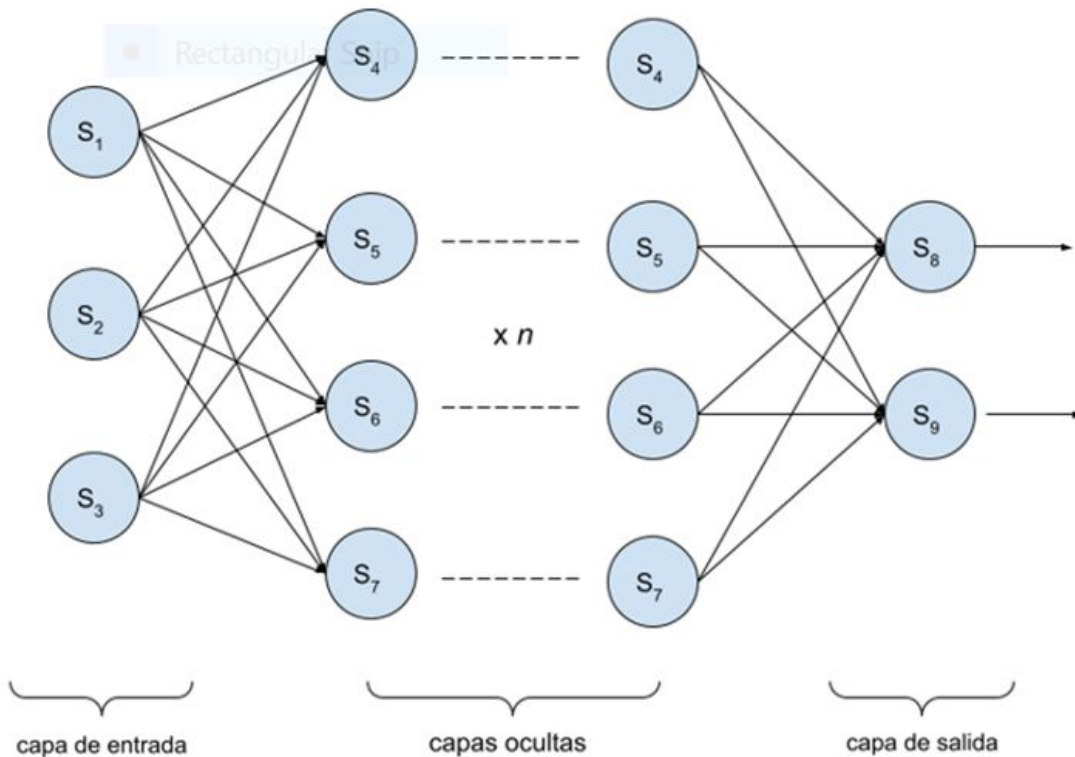
Funciones de activación



Redes neuronales



Aprendizaje profundo



Ejemplo de red neuronal sencilla

<https://github.com/aoliverg/materiales-TAN>

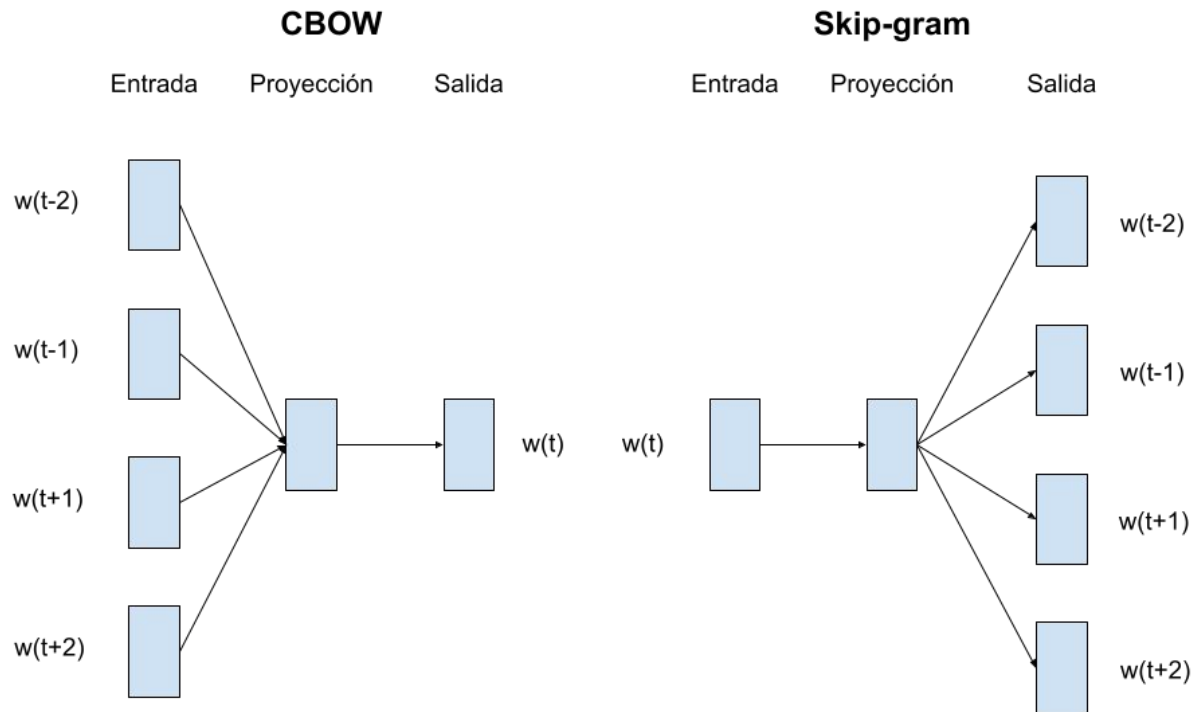
- Demo neurona artificial
- Demo red neuronal

Words embeddings

c=0	La matrícula de mi coche era fácilmente identificable en una ciudad pequeña como la nuestra.
c=1	La matrícula de mi coche era fácilmente identificable en una ciudad pequeña como la nuestra.
c=2	La matrícula de mi coche era fácilmente identificable en una ciudad pequeña como la nuestra.
c=3	La matrícula de mi coche era fácilmente identificable en una ciudad pequeña como la nuestra.

- **CBOW** (Continuous Bag of Words), que lee las palabras del contexto e intenta predecir la palabra central más probable.
- **Modelo Skip-Gram**, que predice las palabras del contexto a partir de la palabra central.

CBOW - Skip Gram



Operaciones con Word Embeddings

- Similitud
- Operaciones

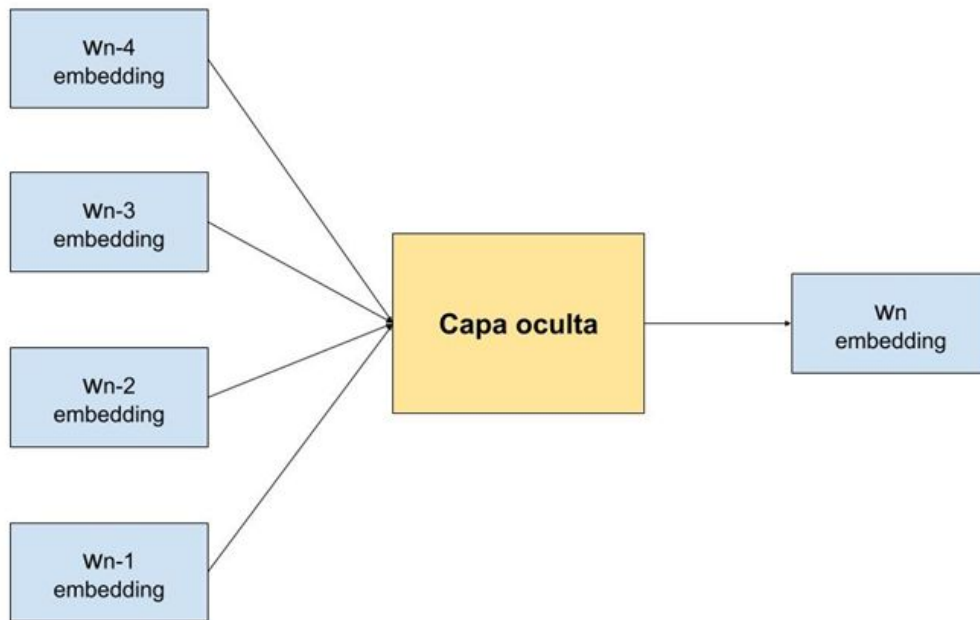
Grandes modelos de Word Embeddings

- <https://docs.google.com/open?id=0B7XkCwpl5KDYNINUTTlSS21pQmM>

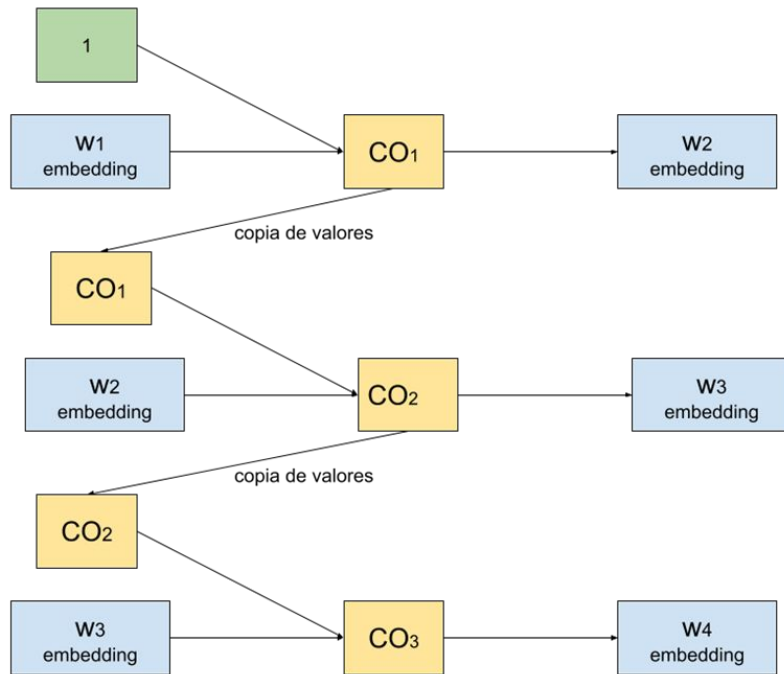
Sentence Embeddings

- La misma idea que Word Embeddings, pero ahora se pretende representar toda una oración.
- Retomaremos esto en la alineación de documentos con una estrategia que se llama *bilingual sentence mining*

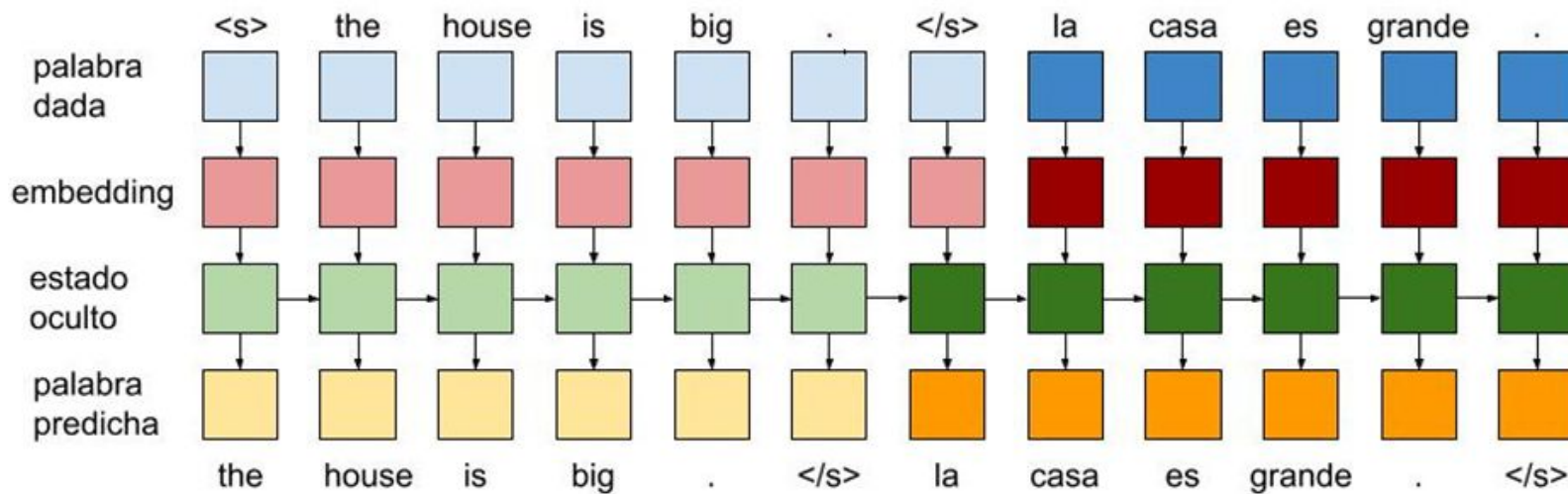
Modelos de lenguaje neuronales



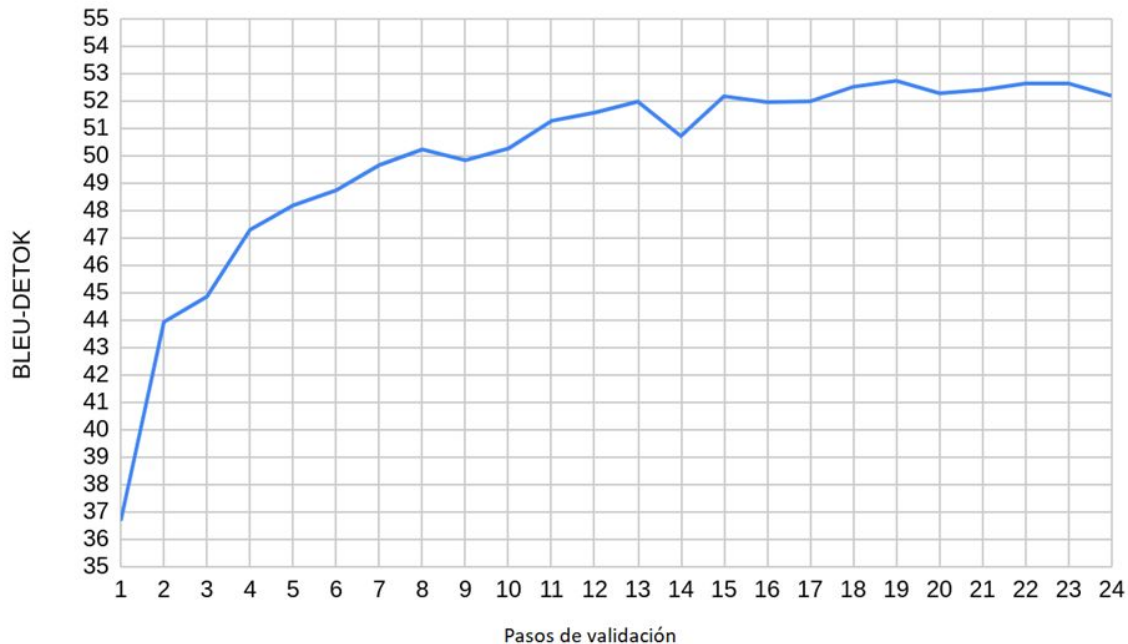
Modelo de lenguaje neuronal recurrente



Modelos de traducción neuronal



Entrenamiento



Toolkits de entrenamiento y uso de motores de traducción automática neuronal

Toolkits (no neuronales)

- [Apertium](#): transferencia sintáctica superficial
- [Moses](#): Traducción automática estadística

Toolkits neuronals

- [Marian](#)
- [OpenNMT](#)
- [Fairseq](#)
- [Transformers](#)

Modelos de traducción automática neuronal libres

OpusMT

<https://github.com/Helsinki-NLP/Opus-MT>

<https://huggingface.co/Helsinki-NLP/opus-mt-en-es>

Ejecución desde un script

```
from transformers import MarianMTModel, MarianTokenizer

src_text = ["This is a simple translation test.", "And this is another sentence."]

model_name = "Helsinki-NLP/opus-mt-en-es"
tokenizer = MarianTokenizer.from_pretrained(model_name)
model = MarianMTModel.from_pretrained(model_name)
translated = model.generate(**tokenizer(src_text, return_tensors="pt",
padding=True))
res = [tokenizer.decode(t, skip_special_tokens=True) for t in translated]
print(res)
```

NLLB - No Language Left Behind

<https://ai.meta.com/research/no-language-left-behind/>

<https://huggingface.co/facebook/nllb-200-distilled-600M>

Ejecución desde un script

```
from transformers import AutoTokenizer, AutoModelForSeq2SeqLM, pipeline

tokenizer = AutoTokenizer.from_pretrained("facebook/nllb-200-distilled-600M")
model = AutoModelForSeq2SeqLM.from_pretrained("facebook/nllb-200-distilled-600M")

translator = pipeline('translation', model=model, tokenizer=tokenizer,
src_lang='eng_Latn', tgt_lang='spa_Latn', max_length = 200)
src_text = ["This is a simple translation test.", "And this is another sentence."]
res = translator(src_text)
print(res)
```

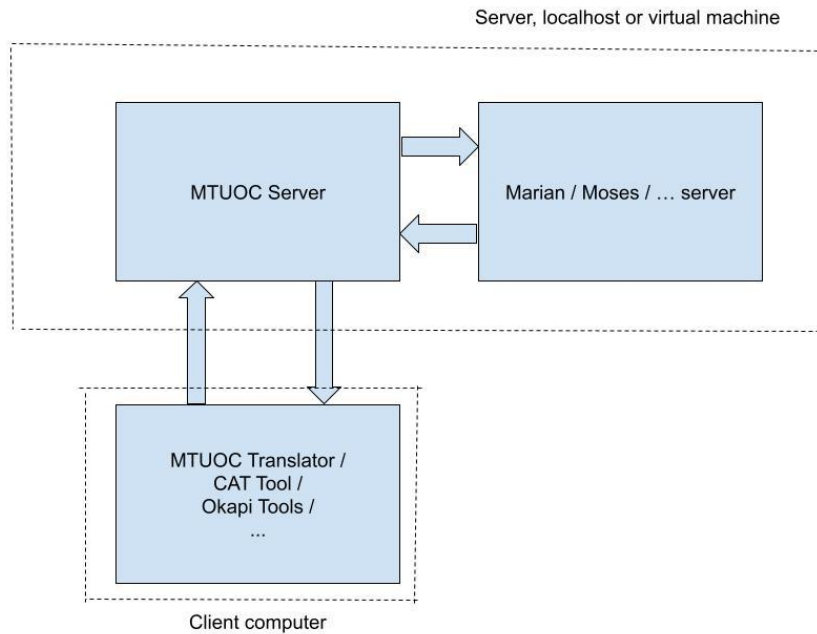
El proyecto MTUOC

Objetivos del proyecto

Facilitar la creación de corpus paralelos, el entrenamiento, evaluación e integración de motores de traducción automática neuronales (y estadísticos).

<https://mtuoc.github.io/>

MTUOC-server



config-server.yaml

```
MTEngine:
  MTengine: OpusMT
  #one of Marian, OpenNMT, Moses, GoogleTranslate, DeepL, Lucy, OpusMT, NLLB, Softcatalà, Apertium, Transformers, Aina
  SLcode: en
  TLcode: es
  multilingual: False
  #False or <tgtlang> or any multilingual code used by the system.

MTUOCServer:
  port: 8000
  type: MTUOC
  #one of MTUOC, Moses, ModernMT, OpenNMT, NMTWizard
  verbosity_level: 3
  log_file: log.log
  ONMT_url_root: "/translator"
  #specific configuration when acting as ONMT server
...
Transformers:
#use the same configuration for OpusMT
  model_path: ../opus-mt-en-es
  #model_path: Helsinki-NLP/opus-mt-tc-big-en-cat_oci_spa
  beam_size: 5
  num_hypotheses: 5
```

Servidor en funcionamiento

```
2024-11-19 13:35:25.470616      3      MTUOC server started using MTUOC protocol
MTUOC server IP:      192.168.1.51
MTUOC server port:      8000
MTUOC server type:      MTUOC
```

MTUOC Translator v.1.2

Connect to: **TEST** | TEXT FILE | XLIFF

Server IP:

Server port:

Server type:

Source Language:

Target Language:




☒ Segmentation

MTUOC Translator v.1.2

Connect to: **TEST** | TEXT FILE | XLIFF

¡Gracias por vuestra atención!

Antoni Oliver
aoliverg@uoc.edu

 UOC.universitat
 @UOCuniversitat
 UOCuniversitat

Universitat Oberta
de Catalunya

Uoc

 UOC.universitat
 @UOCuniversitat
 UOCuniversitat
