uc3m | Universidad Carlos III de Madrid

OpenCourseWare

Procesamiento de Lenguaje Natural con Aprendizaje Profundo,

Máster en Ciencia y Tecnología Informática

Tema 6 Data Augmentation

Isabel Segura Bedmar

Objetivos

- Comprender el concepto de Data Augmentation (DA) y conocer sus principales técnicas.
- Conocer las principales librerías que implementan las técnicas de DA para PLN y saber utilizarlas para generar datos sintéticos que puedan ser utilizados en tareas de PLN.

Índice

- ¿Qué es Data Augmentation (DA)?
- DA en visión artificial
- DA en PLN
- Principales técnicas / librerías de DA para PLN

Data Augmentation (DA)

- **DA** son técnicas dirigidas a **generar nuevos datos sintéticos** a partir de los datos disponibles.
- Los algoritmos de **aprendizaje automático** (supervisado) requieren **grandes colecciones de datos anotados** para la tarea a realizar (por ejemplo, clasificación de imágenes, clasificación de textos, generación de resúmenes, etc), para **obtener buenos resultados**.
- El proceso de **anotación** de estos datos es muy **costoso** (requiere conocimiento experto) y lento.
- DA es un proceso automático que puede reducir el coste en el proceso de creación de nuevos datos.

Data Augmentation (DA)

- **Generan nuevos instancias sintéticos** aplicando **modificaciones** sobre los **datos** del conjunto **original**.
- Además de aumentar la cantidad de datos, también **añaden diversidad** al conjunto de datos.
- DA actúa como una técnica de **regularización**, evitando el **sobre-abrendizaje** en los modelos.

Índice

- ¿Qué es Data Augmentation (DA)?
- **DA en visión artificial**
- DA en PLN
- Principales técnicas / librerías de DA para PLN

DA en visión artificial

- En este campo, las modificaciones más habituales son cambios en tamaño, color, contraste, brillo, rotaciones, zoom, etc.
- albumentations es una librería de Python para generar nuevas imágenes.

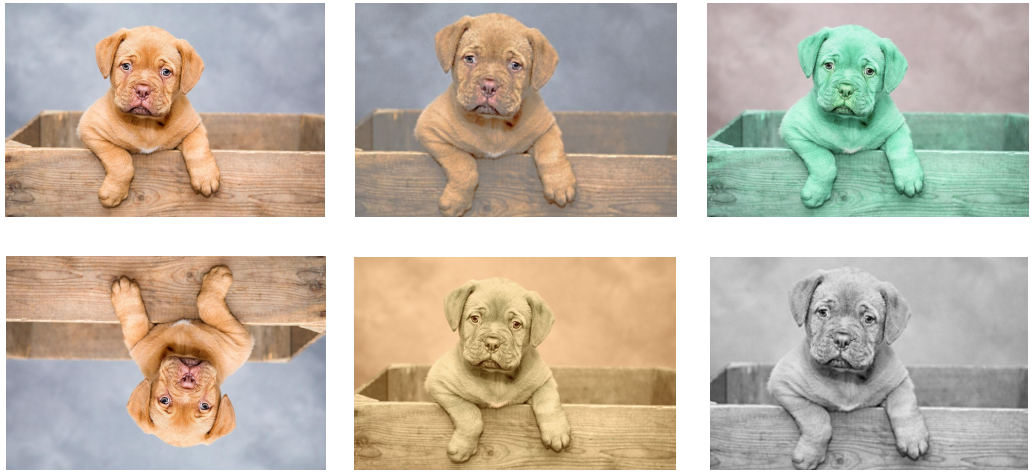


Imagen de Ilona Krijgsman en Pixabay

Índice

- ¿Qué es Data Augmentation (DA)?
- DA en visión artificial
- **DA en PLN**
- Principales técnicas / librerías de DA para PLN

DA en PLN

- Operaciones que consisten en **añadir**, **eliminar** o **intercambiar caracteres** y/o **tokens**.
- Principales enfoques (librerías) DA para PLN:
  - NLP Aug.
  - Easy Data Augmentation (EDA).
  - Back translation.

NLP Aug

- Librería de Python que proporciona una **implementación eficiente** de las técnicas de DA para texto.
- En tres **niveles**: **carácter**, **palabra** u **oración**.
- Para cada nivel:
  - eliminación aleatoria,
  - inserción aleatoria,
  - alteración del orden,
  - sustitución de sinónimos.

NLPAug: reemplazo de caracteres

- Además del reemplazo aleatorio, NLPAug incluye otros métodos para reemplazar caracteres por otros, como **Optical character recognition** o **Keyboard Augmenter**.
- Por ejemplo, **Optical character recognition (OCR)** simula errores basados en que algunos caracteres tienen grafía similar:

El barco estaba lleno de peces

***B**l barco e**8**ta**6**a **11**eno de peces*

NLPAug: reemplazo de caracteres

- Por ejemplo, **Optical character recognition (OCR)** simula errores basados en que algunos caracteres tienen grafía similar:

```python
import nlpaug.augmenter.char as nac
text = 'El Europarlamento insta a prohibir el cobro por el equipaje de mano, pero aún lo seguirás pagando.'

aug = nac.OcrAug()
augmented_texts = aug.augment(text, n=3)

print("Texto original:")
print(text)
print()
print("Textos generados:")
for new_text in augmented_texts:
    print(new_text)
```

```
Texto original:
El Europarlamento insta a prohibir el cobro por el equipaje de mano, pero aún lo seguirás pagando.

Textos generados:
El Europarlamento insta a prohibir e1 cobro por el equipaje de mano, peku aún 10 seguirás pagando.
El Europarlamento insta a pkohi6ik el cobro p0k el equipaje de manu, pero aún 10 seguirás pa9and0.
El Bukupar1amentu insta a pr0hi6ik e1 cobro puk el equipaje de mano, pero aún lo seguirás pa9andu.
```

NLPAug: Reemplazo de caracteres

- **Keyboard Augmenter:** genera nuevas instancias reemplazando algunos caracteres por otros que están próximos en el teclado

El barco estaba lleno de peces

El barvp estaba ñleno de oeves

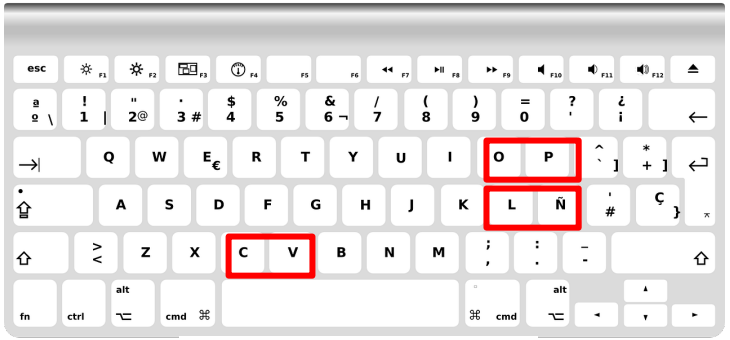


Imagen de OpenClipart-Vectors en Pixabay

NLPAug: Reemplazo de caracteres

- **Keyboard Augmenter:** genera nuevas instancias reemplazando algunos caracteres por otros que están próximos en el teclado

```python
aug = nac.KeyboardAug()
augmented_texts = aug.augment(text, n=3)
print("Texto original:")
print(text)
print()
print("Textos generados:")
for new_text in augmented_texts:
    print(new_text)
```

```
Texto original:
El Europarlamento insta a prohibir el cobro por el equipaje de mano, pero aún lo seguirás pagando.

Textos generados:
El WuroLa5laJentL insta a prohibir el cob#9 por el eAu7laje de mano, p3r) aún lo eWguieás 0zganWo.
El EKGopsFlamen$o insta a (rohigKr el coVrl por el equl)ajD de HaMo, pero aún lo serHitás pagando.
El Europarlamento 7nsfa a proM&bor el vobrp por el #qu7pxje de Hsno, pero aún lo s#gu9ráa pagando.
```

NLPAug: Reemplazo de sinónimos

- Basado en el uso de diccionarios (como WordNet) o modelos de lenguaje (modelos de word embeddings, BERT, RoBERT, GPT, etc)

El barco estaba lleno de peces

*El **navío** estaba lleno de **pescados***

NLPAug: Reemplazo de sinónimos

- Aunque el uso de sinónimos facilitar mantener el significado original, es posible que se produzcan otro tipo de errores:

El barco estaba lleno de peces

*El **embarcación** estaba lleno de **pescados***

NLPAug: Reemplazo de sinónimos

```
[ ]
aug = naw.SynonymAug(aug_src='wordnet')
text = 'The key opens all the locks in the house'
augmented_texts = aug.augment(text, n=3)

print("Texto original:")
print(text)
print()
print("Textos generados:")
for new_text in augmented_texts:
    print(new_text)
```

```
Texto original:
The key opens all the locks in the house

Textos generados:
The key give all the locks in the sign
The cardinal opens totally the locks in the theater
The key opens wholly the locks in the household
```

NLPAug: Reemplazo de sinónimos

```
aug = naw.ContextualWordEmbsAug(model_path='bert-base-uncased', action="substitute")
augmented_texts = aug.augment(text, n=3)

print("Texto original:")
print(text)
print()
print("Textos generados:")
for new_text in augmented_texts:
    print(new_text)
```

```
Texto original:
The key opens all the locks in the house

Textos generados:
a key opened all the locks to the house
the car opens all 42 doors in the house
the key opens any possible locks to the house
```

NLPAug: Reemplazo por antónimos

```python
aug = naw.AntonymAug()
text = 'A Good Life offers a very different perspective'
augmented_texts = aug.augment(text, n=3)

print("Texto original:")
print(text)
print()
print("Textos generados:")
for new_text in augmented_texts:
    print(new_text)
```

```
Texto original:
A Good Life offers a very different perspective

Textos generados:
A Bad Life offers a very like perspective
A Evil Life offers a very same perspective
A Evil Life offers a very same perspective
```

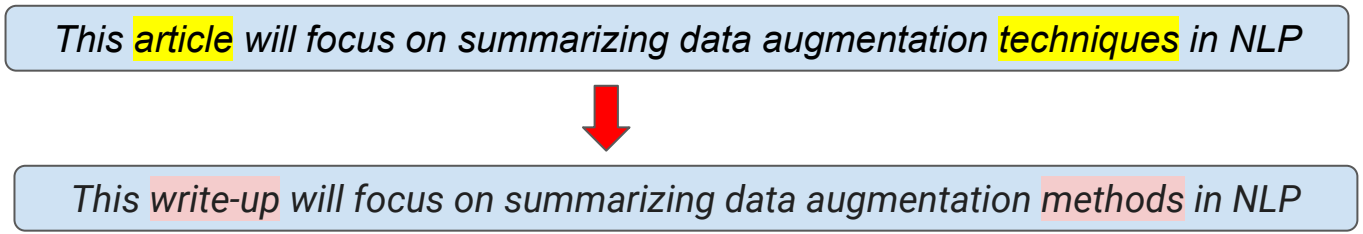
Easy Data Augmentation (EDA)

- Operaciones muy básicas para generar nuevos datos sintéticos para tareas de NLP, implementadas en la librería de Python TextAugment*:
  - sustitución de sinónimos.
  - inserción de sinónimos.
  - intercambio aleatorio.
  - borrado aleatorio.
- **TextAugment** utiliza la librería **nltk** para **obtener** la lista de **stopwords** y los **sinónimos** de WordNet (base de datos léxica con más de 150.000 palabras en inglés: nombres, verbos, adjetivos y adverbios), o de su versión multilingüe OMW.

*Wei, J., & Zou, K. (2019). Eda: Easy data augmentation techniques for boosting performance on text classification tasks. EMNLP 2019. https://aclanthology.org/D19-1670.pdf

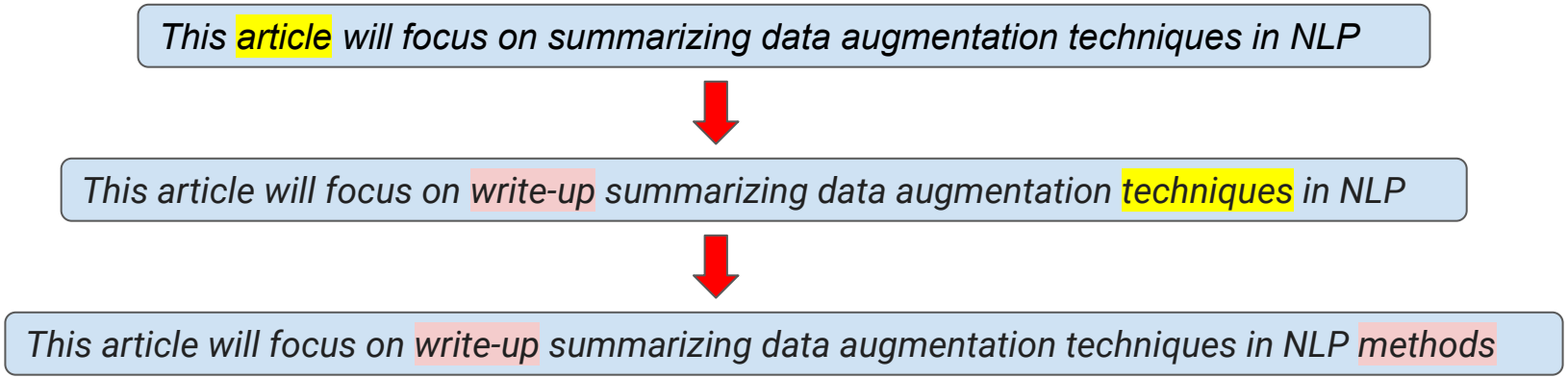
EDA: sustitución de sinónimos

- En esta operación, se seleccionan **n palabras** (que no sean stopwords) en un texto para ser **reemplazadas** por **sinónimos** para generar un nuevo texto.



EDA: inserción aleatoria

- En esta operación, se selecciona una **palabra** (que no sea una stopword) de forma **aleatoria**, e **inserta** un **sinónimo** de dicha palabra en una **posición aleatoria** en la oración. Pueden producir errores gramaticales y semánticos
- Este proceso se realiza n veces.
- Las palabras originales no son eliminadas o reemplazadas. Por ejemplo:



EDA: intercambio aleatoria

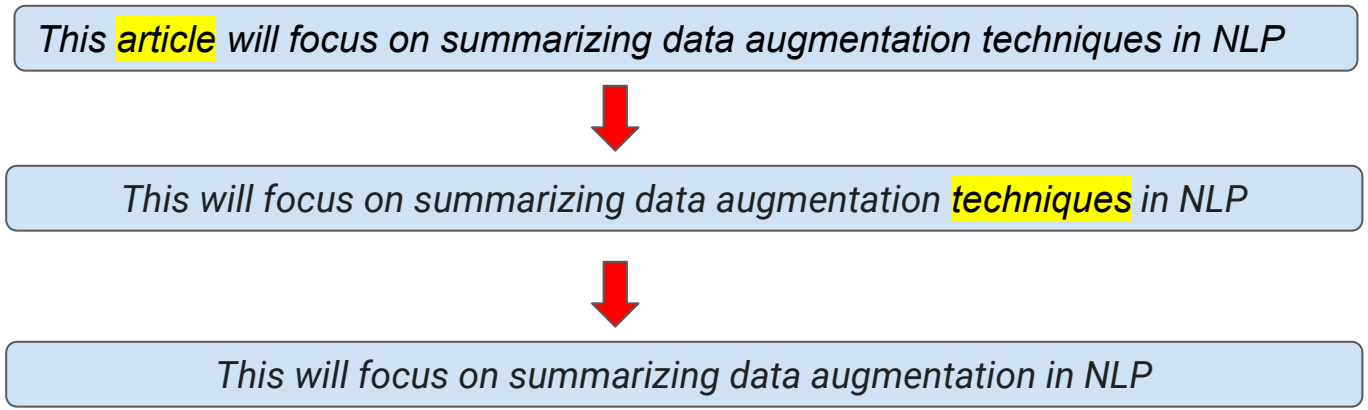
- En esta operación, se seleccionan **aleatoriamente dos palabras**, y se **intercambian** sus posiciones.
- Puede producir errores gramaticales y semánticos

This article will focus on summarizing data augmentation techniques in NLP

This techniques will focus on summarizing data augmentation article in NLP

EDA: borrado aleatorio

- Se selecciona una palabra aleatoriamente y se elimina del texto. Puede ejecutarse varias veces.
- Puede producir errores gramaticales y semánticos



Back translation

- Traducir el texto original a un idioma, y volver a traducirlo al idioma original.

No me esperes para el almuerzo

Imagen de Raphael Silva en Pixabay

Don’t wait me for lunch

Imagen de OpenClipart-Vectors en Pixabay

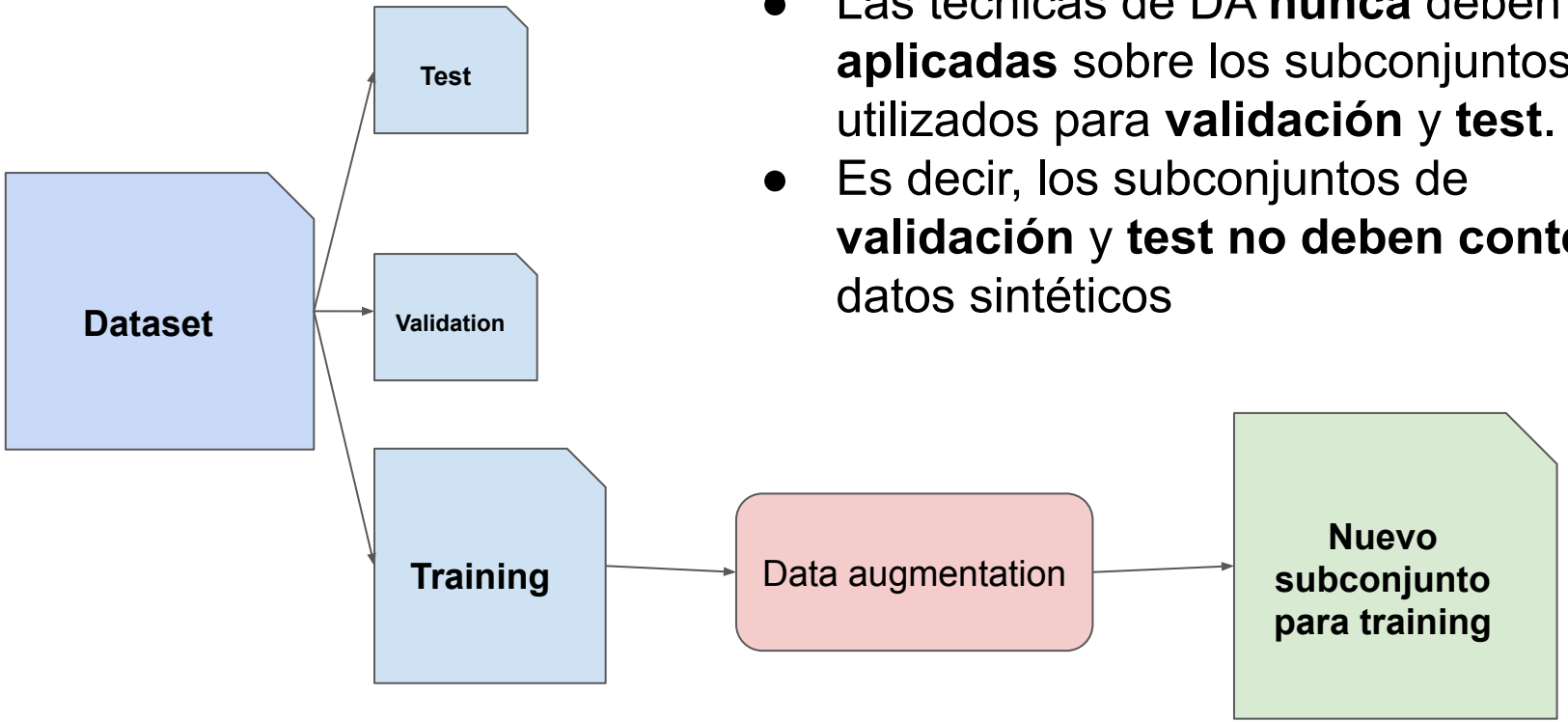
No me esperes para almorzar

Imagen de Raphael Silva en Pixabay

DA en PLN

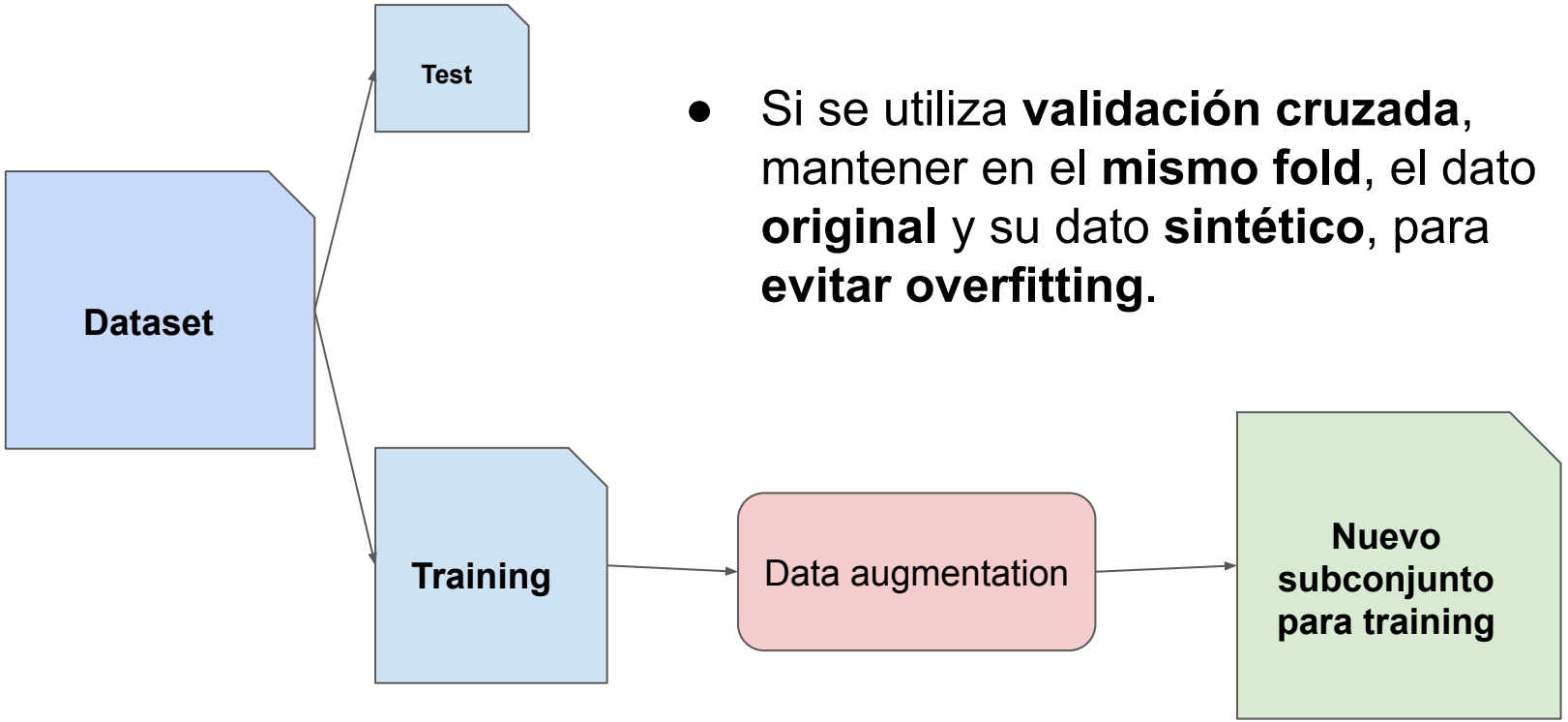
- En PLN, DA es una **tarea** muy **compleja** ya que las modificaciones realizadas sobre el texto original podrían generar un **nuevo texto** con **errores** gramaticales o que haya perdido su significado.
- Además, mientras que en visión artificial, los datos sintéticos pueden generarse durante el entrenamiento, en el caso de PLN, estos nuevos datos sintéticos deben ser generados antes de comenzar el entrenamiento.

DA en PLN



- Las técnicas de DA **nunca** deben ser **aplicadas** sobre los subconjuntos utilizados para **validación** y **test.**
- Es decir, los subconjuntos de **validación** y **test no deben contener** datos sintéticos

DA en PLN



- Si se utiliza **validación cruzada**, mantener en el **mismo fold**, el dato **original** y su dato **sintético**, para **evitar overfitting**.

Resumen

- Los algoritmos de **aprendizaje automático**, y en particular, las redes neuronales profundas **requieren grande cantidades de datos** para poder **entrenar modelos** que puedan obtener **buenos resultados**.
- Las técnicas de **DA** permiten **generar nuevos datos sintéticos** a partir de los datos del **conjunto de entrenamiento**.
- Los subconjuntos de test y validación no deben incluir datos sintéticos.
- **DA en PLN** consiste en **operaciones** para **añadir**, **borrar** o **intercambiar caracteres**, **tokens** u **oraciones** en el texto original. También se utilizan algoritmos de **traducción automática** (back translation: español -> inglés -> español).

Resumen

- **DA en PLN** es una **tarea compleja** porque el **nuevo texto** generado podría contener **errores** y haber **perdido** la **coherencia** con el significado del texto original.
- Es necesario un **estudio empírico** para determinar la **mejor combinación** de **técnicas** de DA y la cantidad de nuevos datos sintéticas, que permitan mejorar los resultados del algoritmo de aprendizaje automático.
- No siempre ayudan a mejorar los resultados en PLN.

OpenCourseWare
Procesamiento de Lenguaje Natural con Aprendizaje Profundo,

Gracias!!!

https://github.com/isegura