MINERÍA DE DATOS – Proyecto de Clasificación de Secciones Médicas

Vectorización mediante: LDA y Doc2Vec – **Clasificadores:** MLP y GRU

Autor: Iker de la Iglesia Martínez

Universidad del País Vasco · Euskal Herriko Unibertsitatea

BILBOKO INGENIARITZA ESKOLA
ESCUELA DE INGENIERÍA DE BILBAO

Introducción

OBJETIVO (TASK)

Dado un fragmento de texto de ámbito médico clasificarlo en una de las secciones permitidas obteniendo el **F-Score** más alto posible.

ASPECTOS A INVESTIGAR

- **Q1 -** Mejor representación vectorial.
- **Q2 –** Análisis del rendimiento de los MLP.
- **Q3 –** Análisis del rendimiento de redes GRU.

Dataset

- **Dataset** [1]**:** Conjunto de textos de ámbito médico formal en castellano. Cada texto forma parte de una sección y se refiere a una enfermedad.
  Instancias: 8 083.
- **Clases:** Se tienen 8 clases indicando la sección de cada texto. La distribución de las clases en las instancias está bastante balanceada . **[Ver Figura 1]**

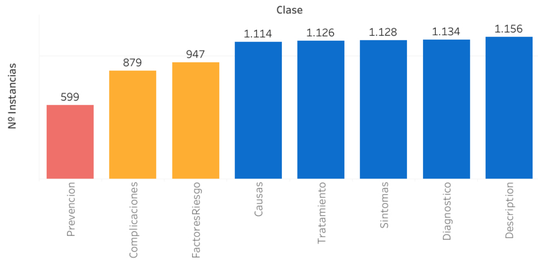

Figura 1: Distribución de las instancias en las clases.

Preproceso

Al texto se le ha aplicando *tokenización* y *lematización*, además de eliminación de *stopwords*, signos de puntuación y acentos.

Se ha logrado una reducción considerable en el número de tokens, especialmente en los textos cuya longitud superaba la media **[Ver Figura 2]** .

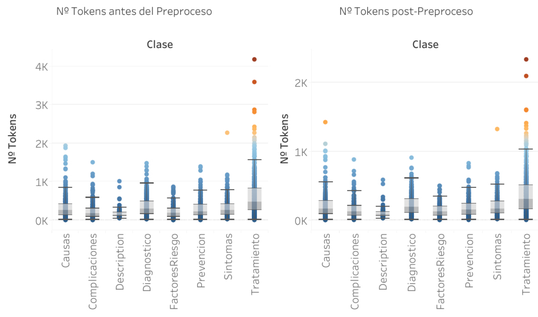

Figura 2: Distribución de las longitudes de las instancias según la clase (pre y post preproceso)

Representación Vectorial

En los experimentos se emplean 2 algoritmos de vectorización: LDA y Doc2Vec.

En el caso del Doc2Vec se ofrecen 2 variantes con el fin de analizar si los embeddings mejoran con mayor cantidad de corpus.

La dimensión se ha seleccionada arbitrariamente para no ser excesiva y ser lo suficientemente grande para poder extraer varias características de los textos.

Nombre	Corpus	Iteraciones	Parámetros	Dimensión del Vector
LDA	Pubmed + CodiEsp 2020	1000	α , β : default ;	200
Doc2Vec_Little	Dataset	200	PV-DM ; *min_count* : 50	200
Doc2Vec_Big	Dataset + Pubmed + WikiMed	100	PV-DM ; *min_count* : 50	200

Tabla 1: Especificaciones de los 3 vectorizadores empleados en los experimentos.

División del Dataset - Empleando Stratified Hold-Out : Train (70%) ; Dev (20%) ; Test (10%)

NOTA: *Algunos gráficos se han recortado para ofrecer una mejor comparativa de los datos relevantes (Todos datos se han tenido en cuenta al calcular métricas: mediana, varianza...).*

Q1 – Representación Vectorial

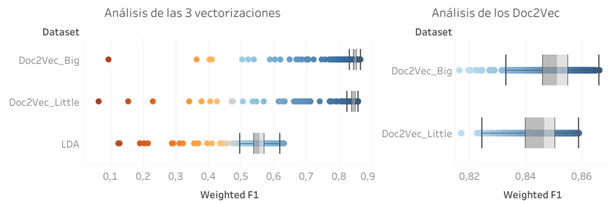

Figura 3: Comparación de los resultados de las distintas representaciones vectoriales en 1300 experimentos.

Como se aprecia en la **Figura 3**, la representación empleando LDA es la que peor resultados da con una **mediana** de 0,5564 ; la **mediana** de ambos Doc2Vec supera el 0,845.

Por otro lado, al comparar de cerca los dos Doc2Vec, se aprecia que el entrenado con mayor corpus ofrece mejores resultados y con una **varianza** menor : 0,00331 (Big) ; 0,00832 (Little).

Q2 – Análisis del rendimiento de los MLP [Sólo Doc2Vec ; 1000 pruebas]

ANÁLISIS EMPLEANDO DISTINTAS TOPOLOGÍAS

En la **Figura 4** se observa, como modelos con **una sola capa** y con un **dropout alto** ofrecen mejores resultados, debido a que estas características reducen el overfitting. El **número de unidades por capa** no ofrece una mejora considerable a partir de las ~50 neuronas.

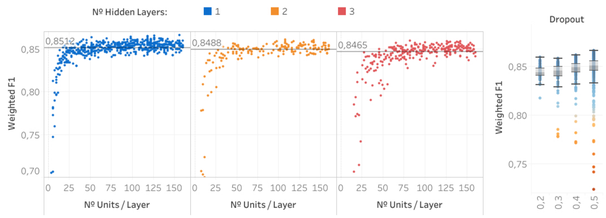

Figura 4: Comparativa de las distintas topologías probadas y de distintos valores dropout para reducir el overfitting.

ANÁLISIS CON DISTINTAS FUNCIONES DE ACTIVACIÓN Y BATCH SIZE

La **función de activación** y el **batch size** no tienen un impacto relevante en las métricas, aunque influyen principalmente en el tiempo requerido para entrenar el modelo. **[Figura 5]**

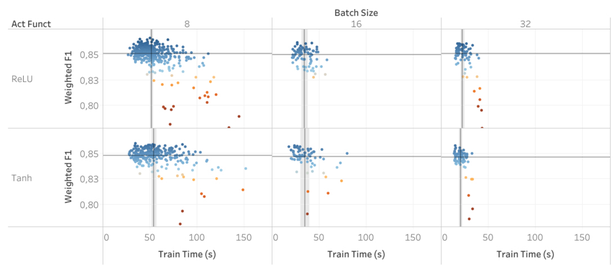

Figura 5: Comparativa del tiempo de entrenamiento y de los resultados en relación a la función de activación y batch size.

ANÁLISIS DEL WEIGHTED F-SCORE Y F-SCORE POR CLASE

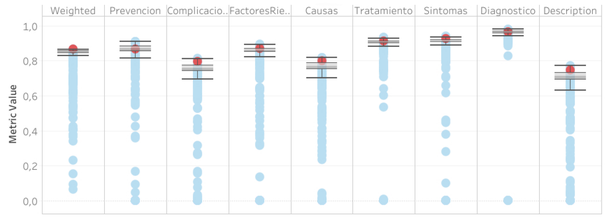

Figura 6: Comparativa de los distintos F-Score (weighted y por clase). Destacado en rojo el mejor resultado (weighted).

Q3 – Análisis del rendimiento de redes GRU

Embeddings: Para estos experimentos se han extraído los Word-Embeddings de los Doc2Vec .

REDUCCIÓN DE LOS TEXTOS PARA ACELERAR EL TIEMPO DE ENTRENAMIENTO

Los textos contienen al principio características suficientes para definir la sección a la que pertenecen, por tanto, según la **mediana** de la longitud de los textos (177), se han tenido en cuenta únicamente los primeros 175 tokens de cada texto. Ejemplo (tokenizado):

> ***respuesta*** *alergico cacahuate soler* ***desencadenar*** *alguno minuto despues exposicion* ***signo síntoma*** *alergia cacahuate poder ser* ***reacciones*** *piel urticaria enrojecimiento hinchazón ...* **[Clase Síntomas]**

ANÁLISIS DE DISTINTAS VARIACIONES DE REDES GRU

Se han ejecutado 500 **experimentos** con un **batch size** de 32 y **función de activación** ReLU en los MLP. En las **figuras 7 y 8** se observa el impacto del resto de hiperparámetros.

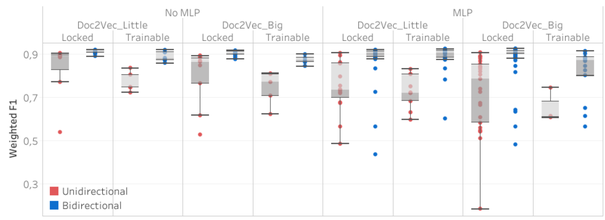

Figura 7: Análisis del impacto de las siguientes características: Embeddings empleados; capacidad de actualizar los embeddings; Conectar un MLP encima de la red GRU; Red GRU unidireccional o bidireccional. (309 experimentos)

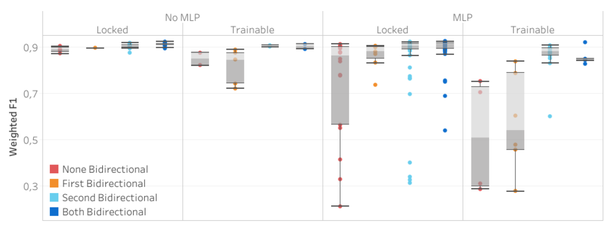

Figura 8: Comparativa de los resultados de **2 redes GRU apiladas** empleando los embeddings Doc2Vec_Big y variando las siguientes características: Capacidad de actualizar los embeddings; Conectar un MLP encima de la red GRU; Redes GRU unidireccionales o bidireccionales (cada una configurada independientemente). (191 experimentos)

Los embeddings entrenados con mayor corpus y no actualizables ofrecen mejor resultado, especialmente al añadir un MLP encima de GRU y concatenar 2 redes GRU. Las redes bidireccionales mejoran considerablemente las métricas en todos los casos. **[Figuras 7 y 8]**

ANÁLISIS DE LOS DISTINTOS TAMAÑOS DE LA RED GRU Y DEL DROPOUT

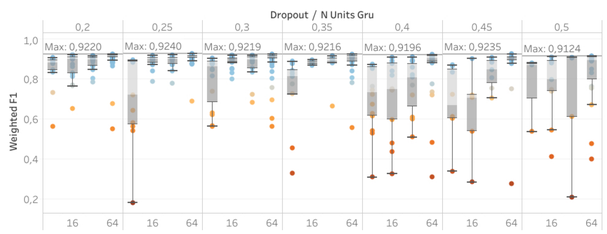

Figura 9: Comparativa de las distintas topologías probadas y de distintos valores dropout para reducir el overfitting. Se observa claramente que las redes GRU con tamaño 64 son

ANÁLISIS DEL WEIGHTED F-SCORE Y F-SCORE POR CLASE

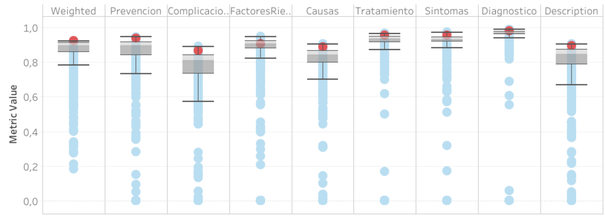

Figura 10: Comparativa de los distintos F-Score (weighted y por clase). Destacado en rojo el mejor resultado (weighted).

Análisis de los Resultados

MEJOR MLP

En la **sección Q2** podemos ver que el mejor MLP ofrece un **weighted F-Score** de 0,8661 con los siguientes hiperparámetros:

Función de activación: ReLU; ***Optimizador:*** Adamax; ***Batch Size:*** 8; ***Dropout:*** 0,5 ***Nº Hidden Layers:*** 1; ***Nº Neuronas por capa:*** 156; ***Vectorización:*** Doc2Vec_Big.

MEJOR RED GRU (MEJOR RESULTADO GLOBAL)

En la **sección Q3** podemos ver que la mejor red GRU ofrece un **weighted F-Score** de 0,9240 con los siguientes hiperparámetros:

Función de activación: ReLU; ***Optimizador:*** Adamax; ***Batch Size:*** 32; ***Dropout:*** 0,25; ***Nº Hidden Layers:*** 3; ***Nº Neuronas por capa (GRU y MLP):*** 64 ***GRUs apiladas:*** sí; ***GRUs bidireccionales:*** sí, ambas; ***Embeddings:*** Doc2Vec_Big ***Actualización de embeddings:*** embeddings no actualizables (locked).

EVALUACIÓN DEL MEJOR MODELO RESPECTO AL CONJUNTO TEST

Con el objetivo de evaluar la sensibilidad del modelo, se han juntado los conjuntos train y dev y se ha hecho un 10-fold CV. A la hora de evaluar la parte de validación de cada split se le ha añadido el conjunto test original generado por Hold-Out:

Modelo	Mean Weighted F-Score (Split Val. + Hold-Out Test)	Desviación Típica	Mín	Máx
Mejor MLP	0,86294	0.00753	0.85334	0.88044
Mejor GRU	0.91872	0.00968	0.90166	0.93677

Modelo	Mean Weighted F-Score (Hold-Out Test)	Desviación Típica	Mín	Máx
Mejor MLP	0.86469	0.00779	0.85386	0.87803
Mejor GRU	0.91911	0.01098	0.89933	0.93306

Conclusiones

- La vectorización con Doc2Vec ha sido la mejor, y mejora y es más consistente cuanto más corpus se emplee en su generación. Esto se aprecia en la **Figura 7** donde la versión Big obtiene mejores resultados sin ser actualizada, mientras que la Little mejora si se permite su actualización.
- Ambos tipos de modelo muestran la misma relación entre las métricas por clase, siendo *Description*, *Causas* y *Complicaciones* las que peor se predicen. Esto puede ser debido a que su contenido puede ser más variado léxicamente que en el resto de las clases. Por otro lado, pese a que *Prevención* es la clase con menos instancias, es una de las que mejor se predicen. **[Figuras 6 y 10]**
- En el caso de los MLP los modelos son capaces de memorizar el training set y por ello las herramientas para evitar overfitting (modelos pequeños , dropout y early-stopping) son eficaces. En las redes GRU no tienen tanto impacto.
- Las redes GRU bidireccionales son muchos más eficaces.
- Apilar redes GRU mejora las métricas, pero el aumento es desdeñable.

Trabajo Futuro

- Probar embeddings entrenados con mayor corpus.
- Aplicar otras técnicas de regularización: L1 y L2
- Mejorar los experimentos en redes GRU: mayor tamaño de las capas GRU, tamaños independientes, comparación respecto a LSTM ...
- Generar ensembles con distintos modelos para mejorar el resultado. Como se aprecia en las **Figuras 6 y 10**, el mejor en general no es el mejor en ninguna de las clases, esto da pie a una posible mejora combinando distintos modelos.

Referencias

1. Spanish Mayo Clinic Diseases (Kaggle), Jesús Utrera
2. WikiMed, Wikimedia Foundation
3. Pubmed Spanish, Courtesy of the U.S. National Library of Medicine
4. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, Cho, Kyunghyun; van Merrienboer, Bart & others (2014)
5. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems
6. Tune: A Research Platform for Distributed Model Selection and Training, Liaw, Richard and Liang, Eric and Nishihara, Robert and others (2018)
7. Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. Bergstra, J., Yamins, D., Cox, D. D. (2013)
8. Research Poster Template, University at Buffalo