Toxic Spans Detection

Mario H. Garrido Czacki E. David Guzmán Ramírez Alejandro Hernández Rodríguez

Licenciatura en Ciencia de Datos
Procesamiento del Lenguaje Natural
Dra. Helena Gómez Adorno
Dra. Gemma Bel Enguix

28 de enero de 2021

Contenidos

Introducción

La moderación es crucial para promover discusiones sanas en línea. Aunque se ha publicado varios conjuntos de datos y modelos de detección de toxicidad (el cual en este trabajo se refiere como lenguaje abusivo) la mayoría de ellos clasifican comentarios o documentos completos y no identifican los tramos que hacen que un texto sea tóxico.

El objetivo de la competencia es detectar el conjunto de subcadenas que se consideran tóxicas de un texto. Por ejemplo:

- "This is a stupid example, so thank you for nothing a!@#!@."
- [10,11,12,13,14,15,51,52,53,54,55,56]

En el primer punto se muestra el texto, y en el segundo los índices de los caracteres de las subcadenas tóxicas, que vienen siendo las cadenas **stupid** y **a!@#!@**.

Descripción del corpus

Los datos se encuentran en Toxic Spans Detection, son alrededor de 10 mil publicaciones que provienen del Civil Comments dataset y que fueron anotados por personas, a las que se les pidió que extrajeran los tramos tóxicos de cada publicación.

En algunas publicaciones tóxicas el mensaje central que se transmite puede ser inherentemente tóxico(por ejemplo, una publicación sarcástica) y, por lo tanto, puede ser difícil atribuir la toxicidad de esas publicaciones a tramos particulares.

Tramo	**Texto**
[4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17]	How fucking stupid are you?
[20, 21, 22, 23]	You are obviously a fool
[ ]	Dumb shit left shit in toilet, now he's up shit creek.
[ ]	This is a garbage country with garbage people.

Tabla 1: Ejemplos de anotaciones en el conjunto de entrenamiento.

Descripción del corpus

Algunas estadísticas básicas del conjunto de datos se encuentran en la tabla 2, en el conjunto de entrenamiento un 6.11 % de los posts fueron anotados con [ ], ya sea porque eran completamente tóxicos, contenían sarcasmo, o simplemente no eran tóxicos. Estos posts resultan un problema ya que es posible que bajen el rendimiento de nuestro modelo, por lo que se intentaron abordar desde distintos ángulos.

Dataset	**Tóxico**	**Vacío**	**Total**
Train	7454	485	7939
Test	647	43	690
Evaluation	-	-	2000

Tabla 2: Estadísticas del conjunto de datos.

Metodología - Preprocesamiento

- **Puntuación:** Se decidió quitar la mayoría de los signos de puntuación que estaban marcados como tóxicos.
- **Spans vacíos:** Posteriormente se trató el tema de las publicaciones que estaban etiquetadas como [ ] con tres distintas estrategias.
- **Minúsculas y tokenización:** Por último, se pasó a minúsculas el conjunto de entrenamiento y se tokenizó con ayuda de spaCy.
- **GloVe:** Se utilizaron embeddings de GloVe preentrenados en un corpus de Twitter.

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

Metodología - Modelo

- **Modelo:** Se encontró un modelo muy aceptable usando una LSTM bidireccional con 6 capas, un tamaño de vector oculto de 600×2 y una probabilidad de dropout de $20\,\%$.
- **Criterio:** Debido a que se entrenó con una función de pérdida de entropía cruzada binaria y se evalúa con F1, se usaron los pesos que optimizaran esta última.
- **Posprocesamiento:** Se usó un enfoque basado en tokens pero se busca una respuesta basada en índices del texto original. Se realiza la conversión y posteriormente una corrección para spans detectados muy juntos:

$$[1, 9] \rightarrow [1, 2, 3, 4, 5, 6, 7, 8, 9]$$

$$[1, 10] \rightarrow [1, 10]$$

Resultados

- La red generada tiene una arquitectura de LSTM bidireccional de 6 capas, seguida de una capa densa de salida a una función sigmoide.
- Se intentaron distintos tipos de preprocesamiento y posprocesamiento para el corpus.

Preprocesamiento	**F1 train**	**F1 test**
Dejar las publicaciones con [ ] tal como están	0.6198	0.6258
Poner las publicaciones con [ ] como completamente tóxicos	0.6498	0.6526
Remover las publicaciones con [ ] del conjunto de entrenamiento	0.7260	0.6459

Tabla 3: Resultados con distintos preprocesamientos.

Resultados - F1 y perdida

Al final se logró obtener un F1 score de 0.6488 en la competencia de Codalab.

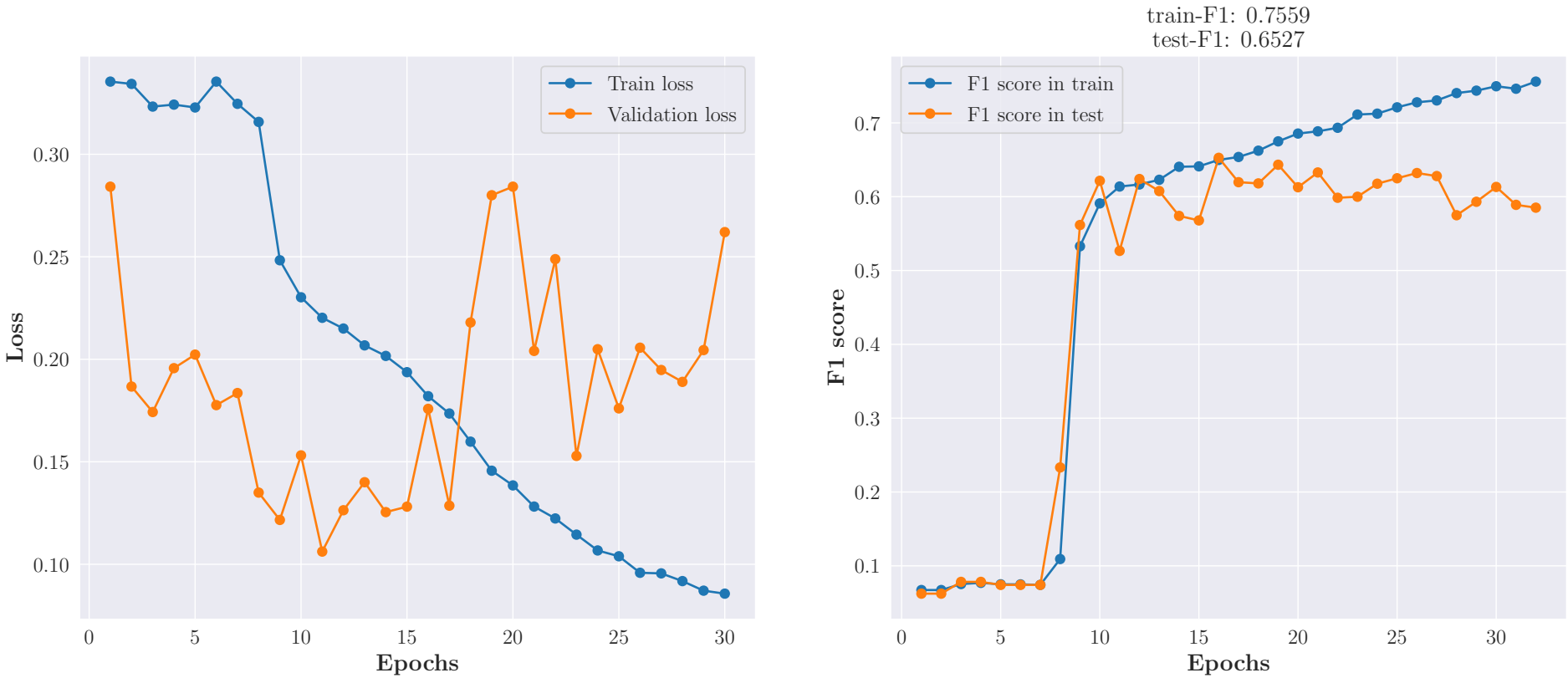

Figura 1: Gráfica de funciones de perdida y F1 score utilizadas en el entrenamiento. Debe observarse que se logró el mejor desempeño en la métrica F1 de evaluación en la época 16 de entrenamiento, por lo que esa fue la utilizada para el modelo final.

Conclusiones

Dados los resultados de otros participantes en Codalab, nuestro modelo tuvo un desempeño aceptable para la tarea establecida. Dado que la máxima calificación obtenida parece ser un 0.7 en F1 score y nuestro modelo presentó un 0.65 en la misma métrica, creemos que tenemos un modelo funcional. Si el tiempo y los recursos lo hubieran permitido, teníamos algunas ideas para mejorar el desempeño más:

- Generar un conjunto de datos aumentados para entrenar la red con más datos.
- Utilizar un modelo de secuencia basado en caracteres en vez de tokens.
- Implementar la generación de características mediante convolución de los caracteres en el token para ampliar el poder predictivo de los embeddings.
- Experimentar con un modelo basado en atención.

Bibliografía

SemEval 2021. *Toxic spans detection.* https://competitions.codalab.org/competitions/25623.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press 2016. https://www.deeplearningbook.org/.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. *Glove: Global vectors for word representation*. 2014. https://nlp.stanford.edu/projects/glove/.