REDES NEURONALES II
Práctica VII - Inteligencia Artificial Avanzada

Abián Santana Ledesma
Samuel Frías Hernández
Alexia Sánchez Cabrera
Javier García Santana

Introducción

A continuación, se emplearán corpus de canciones tradicionales y actuales de la música canaria para generar nuevas composiciones. Para ello, se entrenó un modelo de lenguaje basado en n-gramas con suavizado laplaciano. En este proceso, se tokenizó y preparó el corpus, se calcularon las probabilidades de los n-gramas y se evaluaron las sentencias generadas. Las nuevas letras ('lyrics') se crearon mediante muestreo top-k, y el rendimiento del modelo se midió utilizando la métrica de perplejidad.

Además, se compararán los resultados obtenidos con los generados por el modelo KenLM. Posteriormente, se construirá un modelo Word2Vec a partir de las letras generadas, utilizando la técnica Skip-gram para analizar las relaciones entre palabras. Este modelo también permitirá identificar las pérdidas durante el entrenamiento, detectar palabras similares y visualizar dichas relaciones mediante técnicas como PCA y t-SNE, cuyos resultados se presentarán al final del informe.

Canciones generadas: Modelo local frente a KenLM

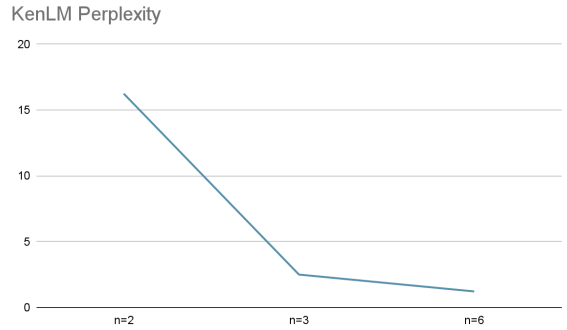
Para comparar estos modelos, se ajustará únicamente la variable 'n', que determina el tamaño del n-grama. Por ejemplo, si establecemos 'n=2' —el máximo que podemos procesar localmente—, obtendremos un modelo de bigramas, en el que la probabilidad de cada palabra depende exclusivamente de la anterior. Si 'n' fuera igual a cinco, la probabilidad dependería de las cuatro palabras previas.

Ambos modelos comparten la ventaja de incorporar la variable 'top-k', que limita la selección de palabras al generar las letras ('lyrics') a las 'k' más probables, lo que mejora la precisión y se considera una práctica óptima a priori. En el caso de KenLM, esta variable 'top-k' está integrada, permitiendo al modelo elegir las 'k' palabras con mayor probabilidad. Para las pruebas realizadas, se fijará 'k = 10'.

En este análisis, tomaremos el resultado generado por el modelo local con 'n=2' y lo compararemos con el de KenLM, incrementando progresivamente el valor de 'n' en este último.

Corpus tradicional

En el modelo local, comenzamos con un modelo de bigramas que presenta una perplejidad de 370, lo que indica que el modelo considera aproximadamente 370 palabras como posibles opciones. Al variar el valor de 'n', se obtienen los siguientes resultados de perplejidad:



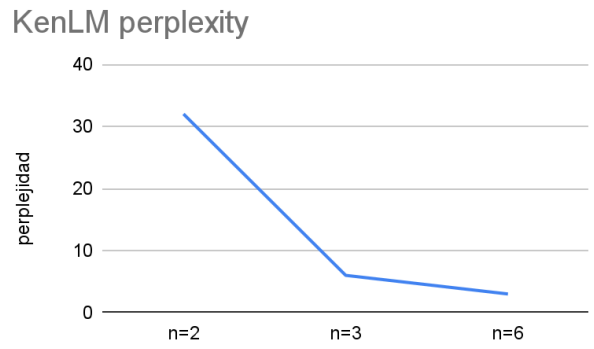
Se observa una disminución en la perplejidad, lo que sugiere que el modelo realiza predicciones más precisas al disponer de mayor contexto con valores más altos de 'n'.

Canciones generadas con KenLM

En conclusión, tras evaluar la calidad de las letras generadas, se puede determinar que, aunque KenLM presenta una menor incertidumbre al dudar entre menos tokens, no logra alcanzar la coherencia del modelo local, independientemente de cuánto se incremente el valor de 'n'.

Corpus moderno

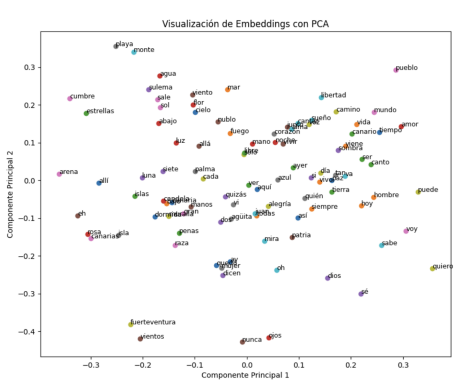
Debido a dificultades para procesar todas las canciones descargadas, en el modelo local se obtuvo una perplejidad de 1573 con un modelo de bigramas, lo que indica que el modelo duda entre aproximadamente 1573 palabras. A través del siguiente gráfico [Nota: se asume que incluirías el gráfico], se observa cómo la perplejidad disminuye al aumentar la variable 'n'. Es importante señalar que, para este ejemplo, se utilizó la misma canción que en el caso del modelo local anterior.

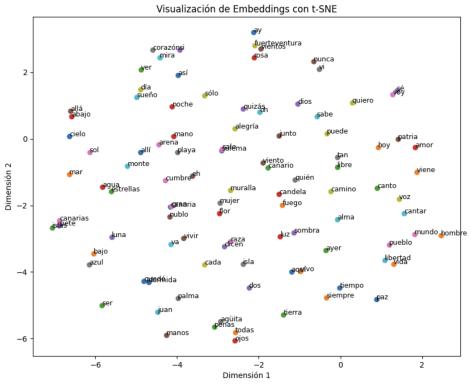


Las canciones generadas pueden consultarse en el mismo enlace compartido previamente. Al analizar los textos generados, se aprecia que, en general, el modelo local exhibe mayor coherencia e incluso logra producir canciones con rimas en algunos casos. Por su parte, el modelo basado en KenLM tiende a combinar palabras que, aunque individualmente pueden tener más sentido entre sí, no generan una coherencia general en el texto.

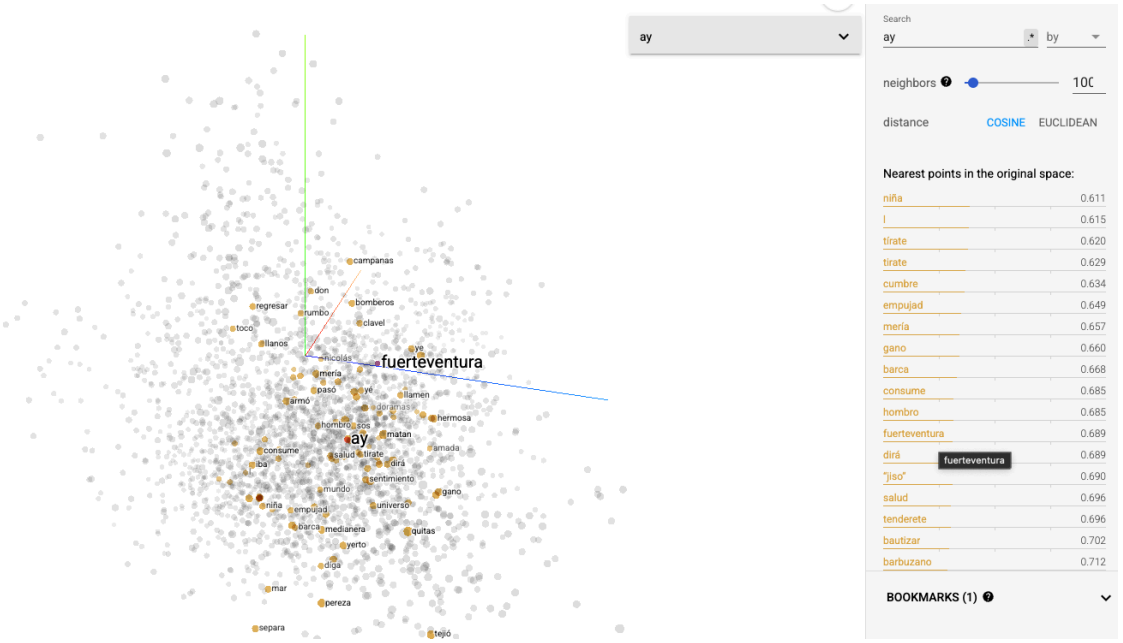
Implementación Word2Vec

Corpus tradicional

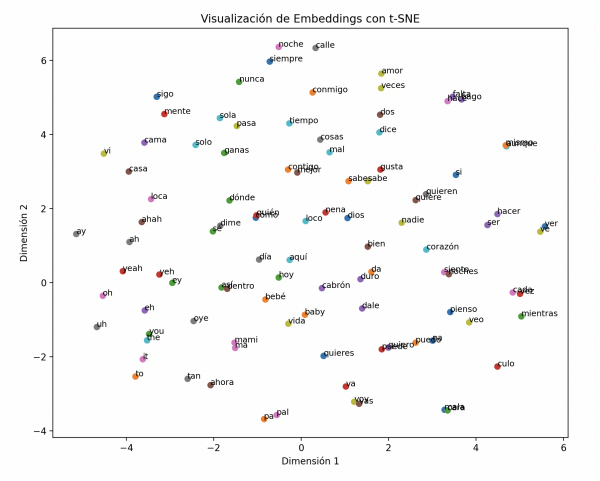


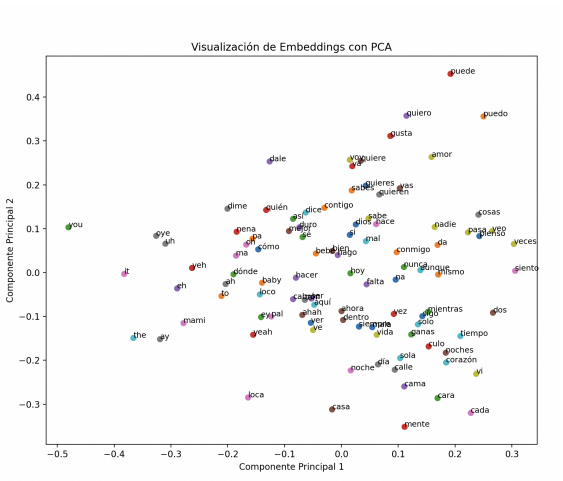


Tras analizar los embeddings y la red neuronal, podemos determinar con mayor precisión qué palabras están más relacionadas entre sí. En la práctica anterior, identificamos los trigramas, bigramas y unigramas más frecuentes en las canciones de la artista Mestisay. Entre los trigramas más utilizados destacó 'ay fuerteventura rosa'. Estas tres palabras aparecen muy próximas entre sí tanto en la visualización con t-SNE como en la proyección tridimensional, lo que confirma que el modelo genera embeddings de manera efectiva.

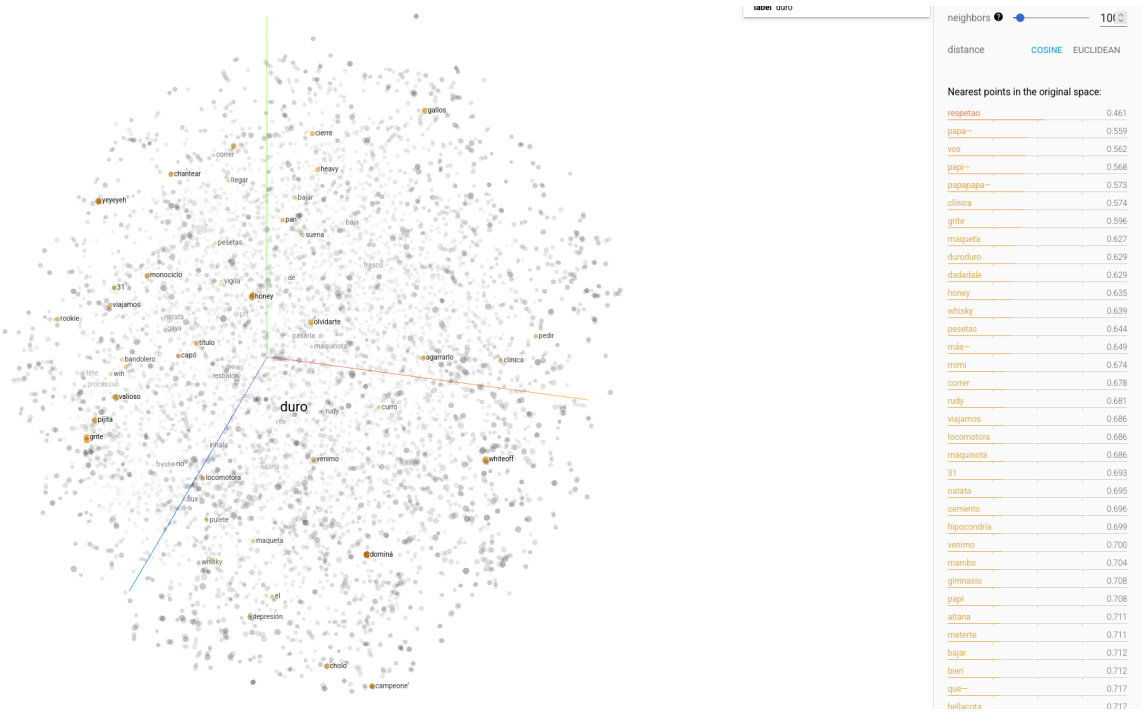


Corpus Moderno





Al analizar el corpus moderno en comparación con el corpus tradicional, se observa que utiliza un lenguaje notablemente más coloquial. Además, contiene numerosas palabras que, en su mayoría, son abreviaturas o variaciones mínimas de un mismo término, como ‘maa’ y ‘maaaaa’, las cuales se repiten con frecuencia. Centrándonos en el artista Quevedo, en un análisis previo se identificó que el trigrama predominante era ‘duro duro duro’, influido por la canción homónima. Sin embargo, como se puede apreciar en el siguiente fragmento:



Tras ampliar el conjunto de letras, la palabra más probable después de ‘duro’ pasa a ser ‘respetao’, lo que marca una diferencia. El trigrama ‘duro duro duro’ aparece relegado a una posición más baja en la lista.

Participación

- **Javier García Santana:** Testeo de corpus con modelos propuestos, elaboración del informe (corpus tradicional) e instalación librerías pytorch y gensim. (25%)
- **Samuel Frías Hernández:** Programación de los scripts python empleados en la experimentación y corrección y formato del informe. (25%)
- **Alexia Sánchez Cabrera:** Testing de corpus actual y ligera modificación del código.(25%)
- **Abián Santana Ledesma:** Desarrollo de script para instalar las librerías necesarias y elaboración de informe con los resultados obtenidos.(25%)

Referencias

Enlace al repositorio con código fuente y resultados obtenidos:

https://github.com/javiergarciasantana/Genius-PLN