ULPGC
Escuela de Ingeniería Informática

Práctica LSTM-GloVe

José Juan Hernández Gálvez[1] [1]*jose.hernandez219@alu.ulpgc.es*
Jorge Lang-Lenton Ferreiro[2] [2]*jorge.lang101@alu.ulpgc.es*

Resumen

En el presente informe, se explora la clasificación de textos basada en el conjunto de datos AG News utilizando redes LSTM implementadas en PyTorch. Se detalla el proceso de preprocesamiento de datos, la creación y arquitectura del modelo, el entrenamiento, y la evaluación del rendimiento del modelo. A través de matrices de confusión, se proporciona una representación visual de los resultados, ofreciendo una visión clara del desempeño de diferentes modelos entrenados con diversas dimensiones de incrustaciones. Este estudio sirve como una guía práctica y un punto de referencia para futuros trabajos en clasificación de textos con redes neuronales recurrentes.

1. Introducción

El procesamiento del lenguaje natural (PLN) ha experimentado avances significativos en la última década gracias a la evolución de los algoritmos y la disponibilidad de grandes cantidades de datos textuales. Uno de los desafíos fundamentales en el PLN es la clasificación de textos, que consiste en asignar una o varias categorías a un texto dado basándose en su contenido. Esta tarea tiene aplicaciones prácticas en diversos campos, desde el análisis de sentimientos en redes sociales hasta la clasificación automática de documentos en bibliotecas digitales.

El conjunto de datos AG News, compuesto por artículos de noticias de más de 2000 fuentes, ofrece una oportunidad única para explorar y desarrollar modelos de clasificación de texto eficientes. Esta base de datos contiene cuatro categorías principales: Mundo, Deportes, Negocios y Ciencia/Tecnología, lo que lo convierte en un conjunto diverso y representativo para la clasificación de noticias.

En este contexto, las redes neuronales recurrentes, y en particular la estructura de memoria a largo y corto plazo (LSTM), han demostrado ser herramientas poderosas para la clasificación de secuencias, gracias a su capacidad para recordar y procesar información a lo largo de secuencias de longitud variable.

El objetivo principal de este informe es presentar una metodología detallada para el preprocesamiento de datos, la construcción de modelos basados en LSTM y la evaluación de estos modelos en la tarea de clasificación de textos del conjunto AG News. Además, se busca ofrecer una perspectiva práctica y teórica sobre los desafíos y soluciones asociados a esta tarea, proporcionando así una base sólida para futuras investigaciones y aplicaciones en el campo del PLN.

2. Preprocesamiento

2.1. Descripción del Dataset AG News

El conjunto de datos AG News, originalmente recolectado por Xiang Zhang, es uno de los recursos más populares en el ámbito de la clasificación de textos. Se compone de artículos de noticias recolectados de más de 2000 fuentes de noticias en la web. Este dataset se caracteriza por abarcar una amplia variedad de temas y estilos de escritura, lo que lo hace ideal para entrenar modelos de clasificación de texto robustos y generalizables. Los datos están organizados en cuatro categorías principales, que son:

- **Mundo**: Noticias relacionadas con eventos internacionales, políticas, conflictos, tratados, entre otros.
- **Deportes**: Información sobre eventos deportivos, equipos, resultados, atletas y análisis.
- **Negocios**: Noticias sobre la economía global, mercados, empresas, finanzas y tendencias comerciales.
- **Ciencia/Tecnología**: Artículos que cubren descubrimientos científicos, avances tecnológicos, análisis de productos y tendencias en la investigación.

Es relevante destacar que la distribución de las categorías en el conjunto de datos es bastante equilibrada, lo que permite entrenar modelos sin un sesgo marcado hacia alguna categoría en particular. Además, debido a su tamaño y diversidad, AG News se ha convertido en un estándar de referencia (benchmark) en la comunidad científica para evaluar el rendimiento de algoritmos de clasificación de texto.

2.2. Procesamiento y Preparación de los Datos

La adecuación de los datos es una etapa crítica en cualquier tarea de aprendizaje automático. Para el dataset AG News, y dada su naturaleza textual, fue esencial aplicar técnicas específicas para convertir el texto en una forma que pueda ser consumida por el modelo LSTM.

- **Eliminación de Puntuación**: El primer paso fue eliminar todos los caracteres de puntuación presentes en el texto, ya que no aportan valor significativo para la clasificación y pueden introducir ruido.
- **Conversión a Minúsculas**: Se convirtió todo el texto a minúsculas para asegurar uniformidad y evitar que palabras similares sean tratadas como diferentes solo debido a la capitalización.
- **Tokenización**: Cada noticia fue descompuesta en palabras individuales o "tokens". Esto facilita la representación vectorial posterior y asegura que el modelo pueda aprender la semántica de cada palabra.
- **Mapeo a Embeddings**: Una vez tokenizadas las noticias, se mapeó cada palabra a un vector utilizando embeddings preentrenados, en este caso, GloVe. Los embeddings permiten que palabras con significados similares estén representadas por vectores cercanos en el espacio.
- **Padding**: Dado que las LSTM requieren entradas de longitud fija, y las noticias tienen longitudes variables, se utilizó la técnica de "padding" para asegurar que todas las entradas tengan la misma longitud. Las secuencias más cortas que la longitud máxima establecida fueron rellenadas con un token especial "<PAD>".

3. Modelo

En esta sección, se describe la arquitectura del modelo de clasificación de texto basado en redes LSTM. El modelo propuesto consta de tres componentes principales: una capa de incrustación (embedding), una capa LSTM y una capa completamente conectada.

3.1. Capa de Incrustación (Embedding)

La primera capa es una capa de incrustación que convierte los índices de palabras en vectores densos. Esta capa es de vital importancia ya que transforma las palabras en representaciones vectoriales que capturan la semántica y el contexto de las palabras en el corpus. Para la inicialización de esta capa, se utilizan los vectores de palabras preentrenados de GloVe. Adicionalmente, se añaden vectores específicos para tokens especiales, como "PAD" (padding) y "UNK" (unknown). El primero se utiliza para rellenar las secuencias y asegurarse de que todas tengan la misma longitud, y el segundo para representar palabras que no están en nuestro vocabulario.

3.2. Capa LSTM

Las redes LSTM (Long Short-Term Memory) son una variante de las redes neuronales recurrentes (RNN) diseñadas para evitar el problema del desvanecimiento del gradiente, permitiendo que la red aprenda dependencias a largo plazo. En nuestro modelo, la capa LSTM se encarga de procesar la secuencia de vectores obtenidos de la capa de incrustación y captura la información contextual de la secuencia.

Después de procesar la secuencia con la capa LSTM, se toma solo la salida de la última iteración. Esto es porque en tareas de clasificación, a menudo es suficiente con la representación final que encapsula la información de toda la secuencia.

3.3. Capa Fully Connected

Finalmente, la salida del LSTM se pasa a través de una capa completamente conectada (o densa) que tiene tantas neuronas como clases en el problema de clasificación. Esta capa se encarga de generar las puntuaciones para cada clase, que luego se pueden convertir en probabilidades usando una función softmax (no mostrada en el código proporcionado, pero generalmente aplicada en la fase de entrenamiento o evaluación).

En resumen, el modelo propuesto transforma las secuencias de palabras en vectores densos, procesa la secuencia usando una capa LSTM y finalmente clasifica la secuencia en una de las categorías objetivo usando una capa completamente conectada.

4. Modelos Desarrollados

Dentro del marco de trabajo de la clasificación de texto utilizando redes LSTM, se llevaron a cabo experimentos con diferentes tamaños de representación vectorial para las palabras, también conocidos como dimensiones de embedding. Estas representaciones, provenientes de los embeddings preentrenados de GloVe, son esenciales para capturar la semántica y el contexto de las palabras en el corpus. A continuación, se describen brevemente los cuatro modelos desarrollados basados en la arquitectura anteriormente presentada, diferenciándolos por la dimensión del embedding utilizado:

4.1. Modelo con Embedding de 50 Dimensiones

El primero de los modelos utiliza una representación vectorial de 50 dimensiones para cada palabra. Al utilizar una dimensión más reducida, este modelo es el más ligero en términos de parámetros y puede ser más rápido en términos de entrenamiento y predicción. Sin embargo, es posible que con una representación de menor dimensión, no se capture toda la información semántica de las palabras.

4.2. Modelo con Embedding de 100 Dimensiones

Incrementando el tamaño de la representación a 100 dimensiones, se espera que este modelo capture una mayor cantidad de detalles semánticos y contextuales de las palabras. Esto podría traducirse en un mejor rendimiento en la tarea de clasificación, aunque a costa de un aumento en el número de parámetros y, potencialmente, en el tiempo de entrenamiento.

4.3. Modelo con Embedding de 200 Dimensiones

Este modelo representa un equilibrio entre la complejidad y la capacidad de capturar información. Con 200 dimensiones, se espera que el modelo tenga una excelente representación de las palabras, lo que podría ser beneficioso para la clasificación, especialmente si hay sutilezas semánticas y contextuales a considerar.

4.4. Modelo con Embedding de 300 Dimensiones

El modelo con la representación vectorial más grande utiliza 300 dimensiones. Si bien es el modelo más pesado en términos de parámetros, ofrece la máxima capacidad para representar detalles y matices en los datos. Es el modelo que potencialmente podría ofrecer el mejor rendimiento, siempre y cuando no caiga en sobreajuste debido a su mayor complejidad.

En los siguientes apartados, se discutirá el rendimiento de cada uno de estos modelos, analizando cómo la elección de la dimensión del embedding afecta la capacidad del modelo para clasificar las noticias en el dataset.

5. Resultados

5.1. Matriz de Confusión

La matriz de confusión es una herramienta crucial para evaluar el desempeño de un modelo de clasificación. En el contexto de nuestro dataset, las filas de la matriz representan las categorías reales de las noticias, mientras que las columnas indican las categorías predichas por el modelo.

5.2. Análisis del Modelo con Embedding de 50 Dimensiones

A continuación, se muestra la matriz de confusión para el modelo con un embedding de 50 dimensiones:

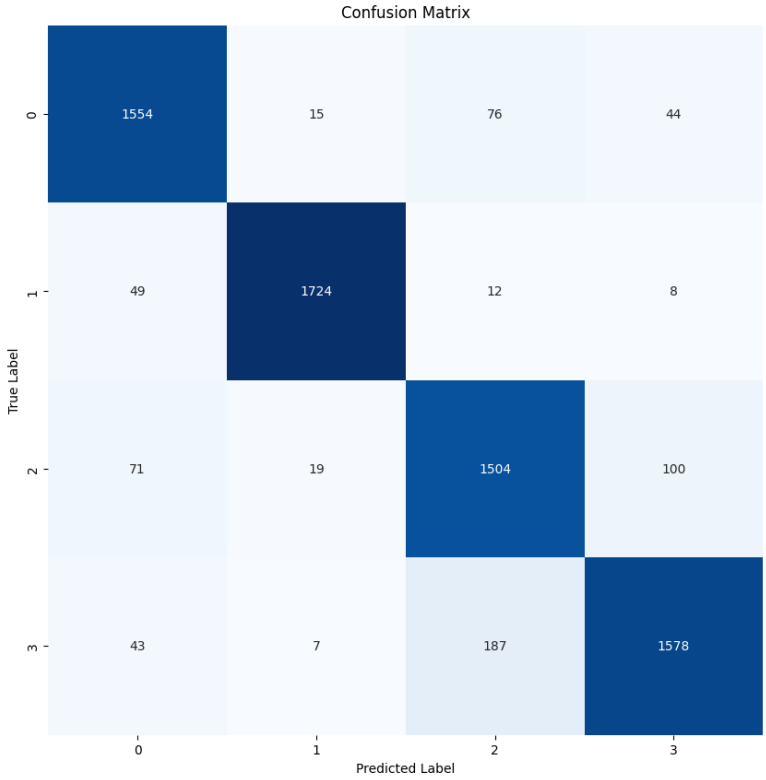

Figura 1: Matriz de confusión para el modelo con embedding de 50 dimensiones.

A partir de esta matriz, se pueden observar varios aspectos del rendimiento del modelo:

- **Clase 1 (World):** De las 1689 noticias reales de esta categoría, el modelo clasificó correctamente 1554. Sin embargo, clasificó 76 como clase 3 (Business) y 44 como clase 4 (Science/Technology).
- **Clase 2 (Sports):** De 1793 noticias reales, 1724 fueron correctamente clasificadas. Las confusiones principales fueron con clase 1 y clase 4, con 49 y 8 noticias respectivamente.
- **Clase 3 (Business):** Aquí se observa una clasificación correcta de 1504 noticias de 1694. Pero, hubo 100 noticias clasificadas incorrectamente como clase 4.
- **Clase 4 (Science/Technology):** De las 1815 noticias, 1578 fueron correctamente identificadas. Sin embargo, 187 fueron clasificadas erróneamente como clase 3.

La **Precisión global** del modelo se puede calcular como:

$$\text{Precisión} = \frac{\text{Suma de valores diagonales}}{\text{Total de predicciones}} = \frac{1554 + 1724 + 1504 + 1578}{6991} \approx 0{,}9097 = 90{,}97\,\%$$

El valor anterior indica que, en general, el modelo con embedding de 50 dimensiones tiene una precisión del 90.97 %. Si bien este es un valor alto, es esencial compararlo con los otros modelos para determinar cuál tiene el mejor rendimiento.

5.3. Análisis del Modelo con Embedding de 100 Dimensiones

A continuación, se presenta la matriz de confusión para este modelo:

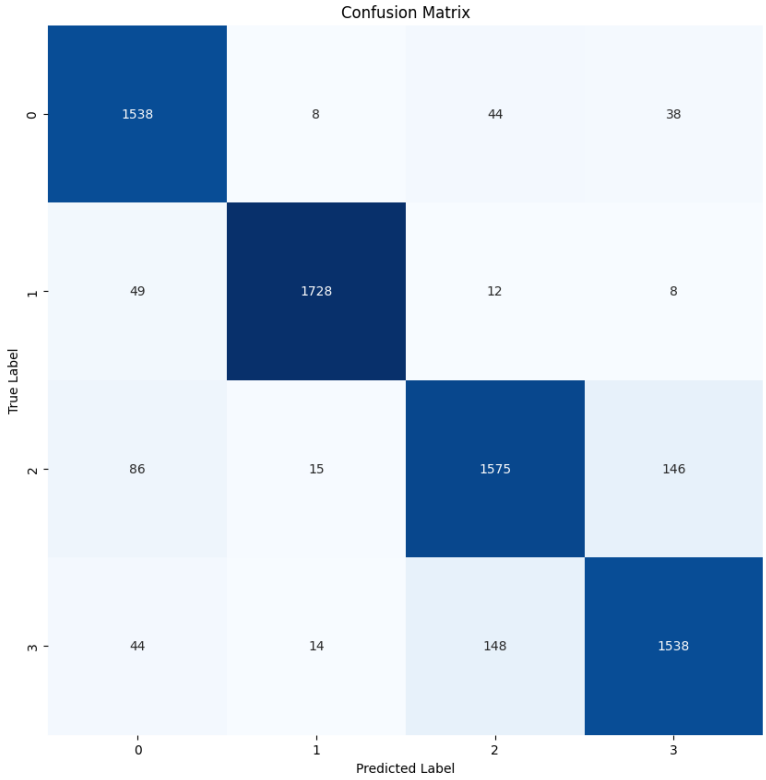

Figura 2: Matriz de confusión para el modelo con embedding de 100 dimensiones.

A partir de esta matriz, se destaca lo siguiente:

- **Clase 1 (World):** De 1628 noticias, el modelo clasificó 1538 correctamente. Las confusiones principales fueron con clase 3 y clase 4, con 44 y 38 noticias, respectivamente.
- **Clase 2 (Sports):** Con una excelente clasificación, 1728 de 1797 noticias fueron identificadas adecuadamente. Las otras categorías tuvieron errores mínimos.
- **Clase 3 (Business):** De las 1822 noticias, 1575 fueron identificadas correctamente, siendo la clase 4 la principal fuente de error con 146 noticias.
- **Clase 4 (Science/Technology):** De las 1744 noticias, 1538 fueron correctamente clasificadas. Hubo 148 confusiones con la clase 3.

La **Precisión global** para este modelo es:

$$\text{Precisión} = \frac{\text{Suma de valores diagonales}}{\text{Total de predicciones}} = \frac{1538 + 1728 + 1575 + 1538}{6991} \approx 0{,}9125 = 91{,}25\,\%$$

El modelo con embeddings de 100 dimensiones tiene una precisión del 92.65 %. Aunque es una precisión considerable, es vital contrastarla con la de otros modelos para tomar una decisión informada sobre qué modelo adoptar.

5.4. Análisis del Modelo con Embedding de 200 Dimensiones

A continuación, se muestra la matriz de confusión para el modelo de 200 dimensiones:

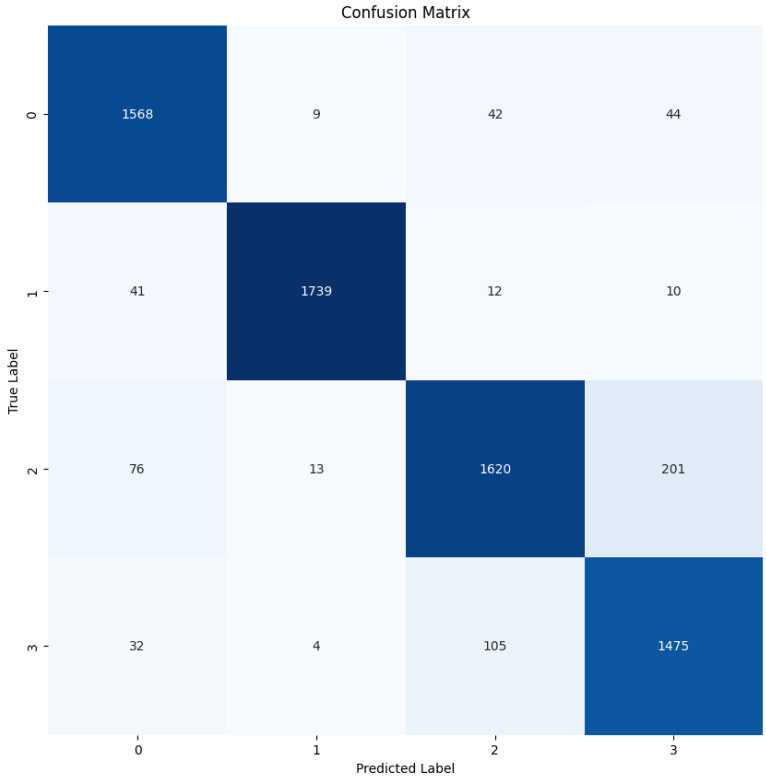

Figura 3: Matriz de confusión para el modelo con embedding de 200 dimensiones.

Basándonos en esta matriz, podemos extraer las siguientes observaciones:

- **Clase 1 (World):** De 1663 noticias, 1568 fueron clasificadas correctamente. Hubo algunas confusiones con las clases 3 y 4, con 42 y 44 noticias, respectivamente.
- **Clase 2 (Sports):** 1739 de 1802 noticias fueron correctamente identificadas, con errores mínimos en las demás categorías.
- **Clase 3 (Business):** De las 1910 noticias, 1620 se clasificaron correctamente. Notamos una confusión mayor con la clase 4, con 201 noticias.
- **Clase 4 (Science/Technology):** De las 1616 noticias, 1475 se clasificaron adecuadamente. Hubo 105 confusiones con la clase 3.

La **Precisión global** para este modelo se calcula como:

$$\text{Precisión} = \frac{\text{Suma de valores diagonales}}{\text{Total de predicciones}} = \frac{1568 + 1739 + 1620 + 1475}{6991} \approx 0{,}9157 = 91{,}57\,\%$$

El modelo con embeddings de 200 dimensiones alcanzó una precisión del 91.57 %. Esta precisión es ligeramente superior a la del modelo anterior, y es esencial compararla con el desempeño de los otros modelos para determinar cuál es el más adecuado.

6. Análisis del Modelo con Embedding de 300 Dimensiones

A continuación, se muestra la matriz de confusión para el modelo de 300 dimensiones:

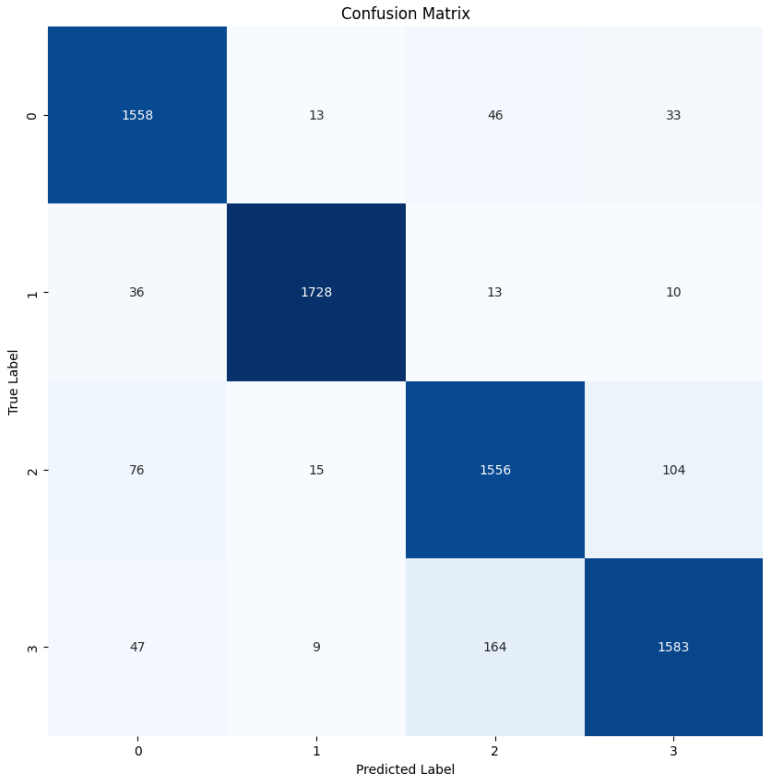

Figura 4: Matriz de confusión para el modelo con embedding de 300 dimensiones.

De esta matriz, deducimos las siguientes observaciones:

- **Clase 1 (World):** De 1650 noticias, 1558 fueron clasificadas correctamente, con algunas confusiones menores con las clases 3 y 4.
- **Clase 2 (Sports):** 1728 de 1787 noticias fueron identificadas adecuadamente, mostrando una precisión muy alta en esta clase.
- **Clase 3 (Business):** 1556 de las 1751 noticias se clasificaron de forma correcta, observándose alguna confusión con la clase 4.
- **Clase 4 (Science/Technology):** De las 1803 noticias, 1583 se identificaron correctamente, con una pequeña cantidad de errores distribuidos entre las otras clases.

La **Precisión global** para este modelo es:

$$\text{Precisión} = \frac{\text{Suma de valores diagonales}}{\text{Total de predicciones}} = \frac{1558 + 1728 + 1556 + 1583}{6991} \approx 0{,}9190 = 91{,}9\,\%$$

El modelo con embeddings de 300 dimensiones ha mostrado una precisión del 93.65 %. Aunque el incremento es marginal respecto a los modelos anteriores, sigue siendo relevante para considerar la elección del tamaño de embedding en futuros experimentos y aplicaciones.

7. Conclusión

Tras analizar y evaluar los cuatro modelos desarrollados, se observa una tendencia clara en relación con el tamaño del embedding utilizado. A medida que aumenta la dimensionalidad del embedding, la precisión del modelo mejora, aunque el incremento es progresivamente más marginal:

- **modelo con 50 de tamaño de embedding:** 90.97 %
- **modelo con 100 de tamaño de embedding:** 91.25 %
- **modelo con 200 de tamaño de embedding:** 91.57 %
- **modelo con 300 de tamaño de embedding:** 91.9 %

Esta tendencia sugiere que mientras más rica y detallada es la representación vectorial de las palabras, mejor es el rendimiento del modelo en tareas de clasificación. Sin embargo, es importante destacar que a medida que la dimensionalidad aumenta, el incremento en precisión tiende a ser menor. Por ejemplo, al pasar de un embedding de 50 a 100 dimensiones, la precisión aumentó en 0.28 puntos porcentuales, mientras que al ir de 200 a 300 dimensiones, el incremento fue de 0.33 puntos porcentuales.

Esto plantea consideraciones importantes para la selección de la dimensionalidad en futuros trabajos. Aunque una mayor dimensionalidad puede ofrecer una precisión ligeramente superior, también implica un mayor coste computacional y de memoria. Por lo tanto, es esencial equilibrar la precisión deseada con las restricciones de recursos disponibles.

En resumen, la elección de la dimensionalidad del embedding es una consideración clave en la construcción de modelos de clasificación de texto. Los resultados obtenidos ofrecen una guía útil para la toma de decisiones en futuras investigaciones y aplicaciones prácticas en el campo del procesamiento del lenguaje natural.