Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC2613 – Inteligencia Artificial (I/2019)

Tarea 1: compresión de texto usando aprendizaje supervisado

En esta tarea, deberán aplicar todo el conocimiento que han adquirido en aprendizaje supervisado, para construir un sistema que permita comprimir archivos de texto.

Codificación basada en frecuencia

Los esquemas tradicionales de codificación de texto, utilizan una cantidad fija de bits para representar todos los caracteres. Por ejemplo, en el caso de la codificación ASCII, se utiliza 1 byte (8 bits) para representar cualquiera de los 256 caracteres que soporta. A pesar de que esta simplicidad trae grandes beneficios, los archivos de texto que obedecen a este estilo de codificación tienden a ser más pesados de lo necesario, debido a que contienen una gran cantidad de redundancia en la codificación.

Una forma de remediar esto, es utilizar esquemas de codificación basados en frecuencia. Estos se basan en el hecho de que de manera natural, algunas letras, sílabas y/o palabras ocurren con mayor frecuencia que otras en los lenguajes humanos. De esta manera, los caracteres con más probabilidad de aparecer reciben una codificación que requiere menos bits que las codificaciones de aquellos caracteres que ocurren menos frecuentemente.

Siguiendo la misma lógica, es posible extender los esquemas de codificación basados en frecuencia, al tomar en consideración el hecho que, dependiendo del contexto, hay algunos caracteres, palabras y/o sílabas que son más probables que ocurran que otras.

Tomando todo lo anterior en consideración, en esta tarea deberá implementar un esquema de codificación (y decodificación) de texto adaptativo, de tal manera que sea capaz de comprimir archivos de texto simple, y luego sea capaz de descomprimirlos, sin que exista pérdida de información. Para lograr esto, deberá utilizar los algoritmos de aprendizaje supervisado vistos hasta ahora en clases, en conjunto con algún tipo de codificador de información basado en frecuencia, tales como el **codificador de Huffman**, o el **codificador aritmético**.

Set de datos

Para entrenar sus modelos, puede usar cualquier set de datos de texto disponible, tales como **20 Newsgroups**, o el **Reuters News dataset**. Se recomienda que el set de datos sea lo más extenso posible, con el fin de capturar la mayor cantidad de patrones.

Modelos de aprendizaje supervisado

En la tarea puede utilizar cualquier modelo de aprendizaje de los vistos hasta el momento en clases. No se permite el uso de modelos más avanzados, o de modelos previamente entrenados. Con el fin de facilitar su labor, se recomienda utiliza la biblioteca *scikit-learn*, que provee implementaciones eficientes de una gran cantidad de algoritmos de aprendizaje supervisado.

Entrega y evaluación

La tarea debe ser realizada en Python 3.5 o superior. Su entrega debe incluir, al menos, dos archivos. El primero, llamado `tarea1.py`, debe implementar el compresor/descompresor de archivos de texto. El segundo archivo, llamado

`tarea1_train.ipynb`, debe contener un Jupyter Notebook donde se implemente y describa el proceso de entrenamiento y construcción del compresor/descompresor.

Con respecto a la ejecución del compresor/descompresor, este debe permitir su ejecución a través de la línea de comandos, utilizando parámetros para especificar su modo de ejecución. Específicamente, el sistema deber ser llamado utilizando el siguiente esquema:

- `python tarea1 -c <src> <dst>`: comprime el archivo de texto de nombre `<src>`, generando el archivo comprimido de nombre `<dst>`.
- `python tarea1 -d <src> <dst>`: descomprime el archivo comprimido de nombre `<src>`, generando el archivo de texto de nombre `<dst>`.

Todas las tareas serán evaluadas utilizando el mismo conjunto de archivos de prueba, todos codificados en ASCII. Existirán bonos para las tres tareas que presenten mejor razón de compresión en los archivos de prueba.

La entrega de la tarea tiene como fecha límite el lunes 30 de mayo a las 23:59, y debe realizarse en la carpeta `T1` del repositorio en GitHub que será asignado a cada uno. Para fines de corrección, se revisará la última versión subida al repositorio. Finalmente, se aplicará un descuento de 1.0 ptos. cada 6 horas o fracción de atraso.

Algunas preguntas interesantes optativas (con bono, obviamente)

Intente responder las siguientes preguntas, indicando su respuesta en el archivo `tarea1_train.py`. Sólo se considerarán para bono aquellas respuestas que estén correctamente fundamentadas, ya sea con código o datos. No se considerarán respuestas genéricas.

- ¿Cómo se comporta la razón de compresión del sistema en idiomas distintos al que fue entrenado?
- ¿Mejora el rendimiento si además de los caracteres, se consideran bigramas, trigramas, etc, en la codificación?
- ¿Como se ve afectada la razón de compresión promedio el sistema, al aumentar o disminuir el tamaño del set de entrenamiento?
- ¿Cuán sensible es el sistema al sobreentrenamiento generado por la complejidad del modelo?

Política de Integridad Académica

Los alumnos de la Escuela de Ingeniería deben mantener un comportamiento acorde al Código de Honor de la Universidad:

"Como miembro de la comunidad de la Pontificia Universidad Católica de Chile me comprometo a respetar los principios y normativas que la rigen. Asimismo, prometo actuar con rectitud y honestidad en las relaciones con los demás integrantes de la comunidad y en la realización de todo trabajo, particularmente en aquellas actividades vinculadas a la docencia, el aprendizaje y la creación, difusión y transferencia del conocimiento. Además, velaré por la integridad de las personas y cuidaré los bienes de la Universidad."

En particular, se espera que mantengan altos estándares de honestidad académica. Cualquier acto deshonesto o fraude académico está prohibido; los alumnos que incurran en este tipo de acciones se exponen a un procedimiento sumario. Ejemplos de actos deshonestos son la copia, el uso de material o equipos no permitidos en las evaluaciones, el plagio, o la falsificación de identidad, entre otros. Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente política de integridad académica en relación a copia y plagio: Todo trabajo presentado por un alumno (grupo) para los efectos de la evaluación de un curso debe ser hecho individualmente por el alumno (grupo), sin apoyo en material de terceros. Si un alumno (grupo) copia un trabajo, se le calificará con nota 1.0 en dicha evaluación y dependiendo de la gravedad de sus acciones podrá tener un 1.0 en todo ese ítem de evaluaciones o un 1.1 en el curso. Además, los antecedentes serán enviados a la Dirección de Docencia de la Escuela de Ingeniería para evaluar posteriores sanciones en conjunto con la Universidad, las que pueden incluir un procedimiento sumario. Por "copia" o "plagio" se entiende incluir en el trabajo presentado como propio, partes desarrolladas por otra persona. Está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, siempre y cuando se incluya la cita correspondiente.