Curso Bases de datos y Programación Orientada a Objetos
Master Universitario en Modelización Matemática, Estadística y Computación

Problema JAVA 2019-20
(I/O y acceso a bases de datos
Pre-procesamiento de datos para almacenamiento)

Vamos a plantear el pre-procesamiento de unos datos relativamente complejos, para extraer determinados parámetros de interés que serán almacenados en una base de datos. De este modo, posteriormente podrán elaborarse análisis en base a ellos.

El ejercicio se plantea como un caso típico en que puede ser precisa bastante elaboración hasta disponer de los valores a almacenar en el formato filas_de_una_tabla (una tabla en nuestro caso, pero podrían ser varias). Usaremos la orientación a objetos como un mecanismo que ayuda a la estructuración de la solución. Una vez dispuesta esta información, es sencillo almacenarla en la base de datos y, del mismo modo, no es demasiado complejo recuperarla para realizar diferentes elaboraciones de la misma (donde también la orientación a objetos muestra su utilidad).

Los datos serán los mismos utilizados en la última clase del curso: datos reales de un procesamiento automático del habla que asocia (de la mejor manera posible) el texto recogido por transcriptores manuales en las actas de las sesiones plenarias del Parlamento Vasco, con el texto reconocido por un decodificador automático de la voz.

***URLs de los ficheros de datos**: http://gtts.ehu.es/German/Docencia/assets/<id>.txt, donde <id> es cada uno de los siguientes identificadores: {20171201, 20171211, 20171214, 20171215}*

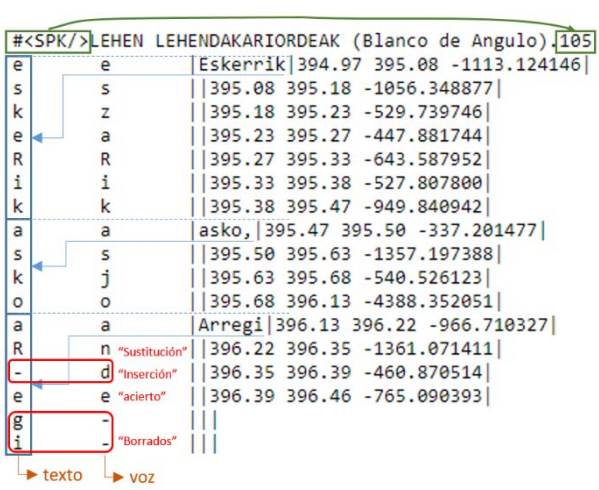
Formato de los datos a procesar

Comencemos por entender la información que encontramos en los ficheros de datos (nos limitaremos a trabajar con cuatro). No es necesario conocerlo a fondo, pero para que no surja ninguna duda, lo veremos completamente:

La asociación entre ambos textos se hace fonema a fonema, por lo que en cada línea las dos primeras columnas se corresponden con dos caracteres que representan un fonema. El primero es el del texto transcrito manualmente y el segundo el reconocido automáticamente. Leyendo de arriba abajo la primera columna, pueden leerse las palabras transcritas, mientras que haciendo lo mismo con la segunda columna, normalmente se obtiene un texto ininteligible. Esto último es debido a que el reconocedor utilizado prescinde de todo conocimiento léxico y se limita a identificar fonemas "sueltos", estrictamente por las características de la señal acústica (el nivel de acierto del reconocedor limitado de este modo es tan sólo del orden del 60%)

En ocasiones, en lugar de un fonema se encuentra un guion. Esto significa que, para obtener el mejor emparejamiento posible, en dicha posición se deja un hueco vacío (si el guion está en la primera columna se dice que el reconocedor automático ha "insertado" un fonema, y si está en la segunda que lo ha "borrado". Por lo demás, si los fonemas coinciden se habla de "aciertos" y si no de "sustituciones". Esto es relevante para un parámetro a extraer y llevar a la base de datos)

A la derecha de estas dos columnas tenemos una tercera compuesta de dos campos separados y enmarcados por el carácter "|". El primero sólo contiene de vez en cuando una palabra, que proviene del acta manual, y lo hace cuando se entiende que comienza la misma. El segundo tiene contenido siempre que hay fonema reconocido automáticamente, y consta de tres campos numéricos: segundo de inicio, segundo de fin y logaritmo de la probabilidad de acierto asignada por el propio reconocedor automático (esta última cantidad la ignoraremos en nuestro ejercicio)



A parte de estas líneas con información sobre el texto/audio, hay otras que comienzan con el carácter "#" y son meramente comentarios o contienen metainformación. En nuestro ejercicio usaremos exclusivamente las que comienzan por "#<spk/>" que llevan información sobre la identidad de quien habla a continuación.

Parámetros que queremos extraer y llevar a la base de datos.

```
DROP DATABASE IF EXISTS mummec19;
CREATE DATABASE mummec19;
USE `mummec19`;
CREATE TABLE reducidos (
  `session` VARCHAR(8) NOT NULL,
  `speaker` INT NOT NULL,
  `numturns` INT NOT NULL,
  `time` DOUBLE NOT NULL,
  `pra` DOUBLE NOT NULL)
```

Lo que pretendemos con el ejercicio es cumplimentar una tabla de base de datos que construiremos con el script de la figura. Cada fichero de los que procesaremos se corresponde con una sesión, y esta consta de una serie de turnos de palabra. En cada turno de palabra tenemos una secuencia de palabras que están formadas por secuencias de fonemas. Queremos llevar a la tabla, por cada sesión una fila de datos por cada participante (identificado por su número) con información de en cuantos turnos ha intervenido (cuantas veces ha tenido turno de palabra), cuanto tiempo han durado en total sus intervenciones, y qué "porcentaje real de acierto" (PRA) ha tenido el reconocimiento automático con sus intervenciones. (el PRA es el número de fonemas acertados dividido entre el total de líneas que le corresponden –aciertos+sustituciones+inserciones+borrados-)

Una vez que hayamos llevado todos los datos a la tabla esta información puede ser usada con otro programa Java para extraer otras elaboraciones de la información. Eso lo planteamos más adelante como ejercicio opcional.

Cómo resolver el ejercicio.

El planteamiento propuesto para resolver el ejercicio es el de crear unas clases con los conceptos que hemos visto aparecer en nuestros ficheros de datos (Sesión, Turno, Palabra, Fonema), y dotarlas de las capacidades que nos faciliten resolver con un pequeño programa principal.

El grueso de la información se encuentra en la documentación de estas clases, que ha sido generada con el programa "javadoc", indicándole que documente todo (incluidos elementos privados y "package"), y escribiendo información específica de la solución adoptada (en este sentido, para una documentación correcta, sobrarían todos los detalles de uso que se han incluido).

Para acceder a esta documentación se debe seguir este link

***Nota**. El planteamiento no dice nada de control de duplicados en las entradas de la tabla, por lo que, si se ejecutan pruebas, la tabla irá creciendo con copias repetidas de datos. Será necesario borrar la tabla desde el workbench para comprobar que todo va bien. Si hacemos que la tabla no pueda contener copias, deberemos controlar los errores que se obtendrán desde nuestro programa Java (es una opción que podéis seguir, pero igualmente hay que borrar el contenido para repetir pruebas).*

Ejercicio opcional ----------

Una vez resuelto el ejercicio y con la base de datos ya conteniendo la información extraída, se propone hacer lo siguiente:

Listar, para cada uno de los hablantes en la tabla, la información de su participación en cada sesión, y si ha participado en más de una, el número de turnos y el tiempo de intervención totales, así como su PRA medio y la desviación estándar del mismo (Se pueden elaborar con Java estos resultados agregados, o –más cómodo- solicitárselos directamente a la base de datos). Este puede ser el resultado (formateado como se quiera).

```
PONENTE: 54
sesión     turnos     tiempo        PRA
20171201        1    215.090      0.694

PONENTE: 56
sesión     turnos     tiempo        PRA
20171201        2    235.210      0.352

PONENTE: 65
sesión     turnos     tiempo        PRA
20171215        2    242.650      0.541

PONENTE: 88
sesión     turnos     tiempo        PRA
20171201        4    472.550      0.568
20171214        1    273.370      0.545
20171215        4    800.700      0.511
turnos=9, tiempo=1546.620, media(PRA)=0.541 desv(PRA)=0.023
(cont…)
```

***Nota**. Esto puede no ser un ejercicio aparte, sino algo que se lleva a cabo a continuación tras la ejecución del procesamiento de los ficheros.*