Inteligencia Artificial II Propuesta de trabajo Detección de correo basura (anti-spam) mediante técnicas de Inteligencia Artificial

José F. Quesada

1 Introducción

La idea de detección o filtrado de correo basura se puede describir de la siguiente forma: "Dado un mensaje nuevo M, determinar si este mensaje debe ser clasificado como correo adecuado (aceptable o ham) o como correo no adecuado (spam o basura)."

Por tanto, en última instancia podemos observar que se trata de un problema de clasificación simple, en el cual el valor de la clasificación es booleano. Así pues, la detección de correo basura es un ejemplo de la técnica de Filtrado de Documentos (Document Filtering), que a su vez es una especialización de la técnica de Clasificación de Documentos (Document Classification).

El objetivo de este trabajo se estructura en tres etapas:

- Estudio del problema de filtrado de documentos y su aplicación al filtrado de correo, y selección de un algoritmo basado en técnicas de Inteligencia Artificial.
- Implementación de dicho algorimo en uno de los lenguajes y entornos permitidos.
- Evaluación de la funcionalidad de la implementación, de acuerdo con los criterios que se detallan más adelante.

2 Algoritmo de filtrado de correo basura

Uno de los algoritmos que se vienen usando con mayor asiduidad en los últimos años para realizar esta tarea es el denominado algoritmo bayesiano ingenuo, que se basa en la aplicación de los conceptos de probabilidad, aunque

aplicando varios supuestos que permiten simplificar el problema (de ahí la motivación del nombre "ingenuo").

No obstante, existen varios algoritmos y enfoques para este problema. En un artículo de 2007 varios profesores analizaban brevemente distintas técnicas de Inteligencia Artificial para la creación de modelos anti-spam:

`http://polar.lsi.uned.es/revista/index.php/ia/article/viewFile/533/517`

En concreto se mencionan 4 algoritmos principales:

- Algoritmos bayesianos ingenuos
- Clasificadores basados en memoria
- Máquinas de soporte vectorial
- Sistemas de razonamiento basados en casos

Este trabajo no impone ningún algoritmo específico ni se limita a los 4 previemante mencionados. El alumno debe elegir un algoritmo que considere adecuado para realizar la tarea descrita. La primera sección de la memoria presentada deberá indicar el algoritmo seleccionado, así como una breve descripción (2 ó 3 páginas) del mismo (incluyendo las bases teóricas y relacionándolo con los temas vistos en la asignatura).

3 Corpus de referencia

Existen múltiples corpus que se pueden utilizar para analizar los algoritmos y sus implementaciones.

Como referencia se puede visitar la siguiente página:

`http://www.aclweb.org/aclwiki/index.php?title=Spam_filtering_datasets`

De entre los corpus (o datasets) disponibles se debe utilizar el Enron en formato preprocesado, formado por 6 bloques descritos y disponibles en el siguiente link:

`http://www.aueb.gr/users/ion/data/enron-spam/`

Es importante tener en cuenta que el trabajo debe ser capaz de utilizar parte de este corpus como información de entrenamiento (por ejemplo los bloques 1 a 5 de enron) para posteriormente analizar los porcentajes de clasificación correcta del último bloque (en este caso el 6).

O bien, se podría aplicar el entrenamiento sobre un porcentaje de cada uno de los bloques (el 75%) y posteriormente hacer la evaluación sobre el reto (el 25% en este caso).

4 Implementación del algoritmo

Una vez seleccionado el algoritmo que se utilizará, se debe realizar la implementación del mismo.

La implementación se podrá hacer en alguno de los siguientes lenguajes: Lisp, C o Java.

Es importante tener en cuenta que uno de los objetivos clave del trabajo es evaluar la implementación del algoritmo, por tanto se considerará copia si se utiliza alguna librería o componente auxiliar no desarrollado directamente por el alumno. Por tanto, cualquier librería o módulo auxiliar utilizado en el desarrollo se deberá indicar explícitamente en la documentación. Se considerará correcto utilizar módulos o librerías auxiliares para procesos laterales del algoritmo (por ejemplo, librerías de gestión de cadenas de texto, o similares).

En la documentación aportada se deberá incluir el análisis realizado para el problema, así como una descripción de alto nivel del diseño e implementación realizada.

Desde el punto de vista funcional, el sistema deberá incluir tres módulos operativos:

- **Módulo de entrenamiento**: Este módulo deberá ser capaz de recibir la lista de correos (correctos - ham, y basura - spam), y a continuar llevar a cabo el proceso de aprendizaje correspondiente generando la representación correspondiente (que deberá almacenar en uno o más ficheros).

- **Módulo de evaluación**: Este módulo debe ser capaz de cargar la representación correspondiente al aprendizaje generado por el módulo de entrenamiento. Así mismo deberá cargar un conjunto de mensajes (tanto correctos - ham, como basura - spam) y obtener a continuación las estadísticas de clasificación. Estas estadísticas indicarán los porcentajes siguientes:

  1. Correctos clasificados correctamente (Ham $->$ Ham)
  2. Correctos clasificados incorrectamente (Ham $->$ Spam)
  3. Incorrectos clasificados correctamente (Spam $->$ Ham)
  4. Incorrectos clasificados incorrectamente (Spam $->$ Spam)

- **Módulo de clasificación**: Al igual que el módulo anterior deberá ser capaz de cargar la representación resultado del aprendizaje, pero su funcionalidad será determinar si un correo recibido como entrada es basura (spam) o no (ham).

5 Evaluación

En la documentación correspondiente del trabajo se deben entregar los resultados de evaluación correspondientes a los dos siguientos modelos:

- **Evaluación 1**: Utilizando enron1 a enron5 como entrenamiento y enron6 como evaluación.
- **Evaluación 2**: Partiendo de una división (aleatoria) de los mensajes de enron1 a enron6 en dos bloques (de entrenamiento con el 75% y de evaluación con el 25%).

6 Memoria e implementación

La presentación del trabajo incluirá por tanto dos partes:

6.1 Memoria

La memoria del trabajo tendrá el siguiente índice:

- Identificación
- Descripción del problema: algoritmo, bases teóricas y relación con las técnicas de Inteligencia Artificial
- Análisis y diseño del algoritmo y su implementación
- Instrucciones de instalación y uso
- Modelos de evaluación utilizados (además de los modelos básicos exigidos se pueden seleccionar más estrategias para analizar el comportamiento del algoritmo)
- Resultados del entrenamiento y evaluación (indicando las características técnicas del sistema utilizado para realizar la evaluación, los datos relevantes -tiempo, memoria, etc.- y los resultados de clasificación obtenidos)
- Análisis crítico del proceso y sugerencias para poder mejorar el algoritmo y su implementación

6.2 Implementación

Se debe adjuntar todo el código fuente necesario para compilar y ejecutar el sistema.

Es importante describir la estructura de directorios y ficheros utilizados, y sobre todo debe ser coherente con las instrucciones de instalación y uso descritas en la memoria.

7 Bibliografía

- Sahami et al (1998): A Bayesian approach to filtering junk email. AAAI TEchnical Report WS-98-05, pp. 55-62

  $http://ftp.research.microsoft.com/pub/ejh/junkfilter.pdf$

- Metsis et al (2006): Spam Filtering with Naive Bayes - Which Naive Bayes?

  $http://www.aueb.gr/users/ion/docs/ceas2006\_paper.pdf$

- José R. Méndez, Florentino Fdez-Riverola, Fernando Díaz, Juan M. Corchado (2007): Sistemas inteligentes para la detección y filtrado de correo spam: una revisión.

  http://polar.lsi.uned.es/revista/index.php/ia/article/viewFile/533/517