Universitat Autònoma de Barcelona

escola d'enginyeria

Proyecto final de carrera

(5641 - Clasificación automática de textos y explotación BI)

Javier Buill Vilches

Directors:

Dr. Ramon Grau Sala.

Dr. Joan-Josep Vallbé Fernández.

Departaments:

Dep. Arquitectura de Computadors (Universitat Autònoma de Barcelona)

Dep. Dret Constitucional i Ciència Política (Universitat de Barcelona)

El sotasignat, Dr. Ramon Grau Sala

professor de l'Escola d'Enginyeria de la UAB,

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per en Javier Buill Vilches

I per a que consti firma la present.

Signat: Dr. Ramon Grau Sala

Bellaterra, 19 de Juny de 2014

El sotasignat, Dr. Joan-Josep Vallbé Fernández

professor del Dep. Dret Constitucional i Ciència Política (Universitat de Barcelona),

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per en Javier Buill Vilches

I per a que consti firma la present.

Signat: Dr. Joan-Josep Vallbé Fernández

Bellaterra, 19 de Juny de 2014

Resumen

El presente proyecto tiene como objetivo desarrollar una tecnología que permita codificar grandes cantidades de texto de manera automática para posteriormente ser visualizada y analizada mediante una aplicación diseñada en Qlikview.

El motor de la investigación e implementación de este proyecto se ha encontrado en la incipiente presencia de tecnologías informáticas en los procesos de codificación para ciencias políticas. De esta manera, el programa creado tiene como objetivo automatizar un proceso que se desarrolla comúnmente de manera manual y, por ende, las ventajas de introducir técnicas informáticas son notablemente valiosas.

Estas automatizaciones permiten ahorrar tanto en tiempo de codificación, como en recursos económicos o humanos. Se ha elaborado una revisión teórica y metodológica que han servido como instrumentos de estudio y mejora, con el firme propósito de reducir al máximo el margen de error y ofrecer un instrumento de calidad con salida de mercado real.

El método de clasificación utilizado ha sido Bayes, y se ha implementado utilizando Matlab. Los resultados de la clasificación han llegado a índices del 99.2%.

En la visualización y análisis mediante Qlikview se pueden modificar los parámetros referentes a partido político, año, categoría o región, con lo que se permite analizar numerosos aspectos relacionados con la distribución de las palabras repartidas entre las diferentes categorías y en el tiempo.

Abstract

The aim of the present project is to develop a technology capable of codifying a huge amount of text automatically in order to be analysed through a Qlikview application.

The main reason for the investigation and implementing of this project has been found due to the new presence of information technology in codifying processes for political science. Thus, the program created seeks to automate a usually hand-made process, and the advantages of introducing these techniques are remarkably valuable.

This automates allow to save time both in time and in economic or human resources. In this point, there has been a theoretic and methodological revision that worked out as study and development instruments, with the aim to reduce the margin of error and offering a quality tool with access to real market.

The classification method used has been Bayes, and it has been implemented by using Matlab. The classification results have reached 99.2% success.

In the visualization and analysis with Qlikview the values of political party, year, category or region can be modified allowing analyse numerous aspects related to the word distribution between the categories and through time.

Resum

El present projecte té com objectiu desenvolupar una tecnologia que permeti codificar grans quantitats de text de forma automàtica per a posteriorment ser visualitzada i analitzada mitjançant una aplicació dissenyada en Qlikview.

El motor de la investigació i implementació d'aquest projecte s'ha trobat en la incipient presència de tecnologies informàtiques als processos de codificació per a ciències polítiques. D'aquesta forma, el programa creat té com a objectiu automatitzar un procés que es desenvolupa comunament de forma manual i, per tant, els avantatges d'introduir tècniques informàtiques són molt valuoses.

Aquestes automatitzacions permeten tant estalviar temps de codificació, com en recursos econòmics o humans. S'ha preparat una revisió teòrica i metodològica que ha funcionat com a instrument d'estudi i de millora, amb el ferma propòsit de reduir al màxim el marge d'error i oferir un instrument de qualitat amb sortides al mercat real.

El mètode de classificació utilitzat ha estat Bayes, i s'ha implementat mitjançant Matlab. Els resultats de la classificació han arribat a índexs del 99.2%.

A la visualització i anàlisis mitjançant Qlikview es poden modificar el paràmetres referents a partit polític, any, categoria o regió, amb el que es poden analitzar nombrosos aspectes relacionats amb la distribució de les paraules repartides entre les diferents categories i en el temps.

Índice

Índice de tablas y figuras

1 Introducción

La informática ha ido conquistando progresivamente todos los aspectos de nuestra vida cotidiana, con la misión de simplificar los procedimientos manuales más complejos. De este modo, se hacen latentes las demandas reales de informatización en procedimientos o tareas repetitivas, que permitirían incorporar inconmensurables ventajas. Asimismo, en las ciencias sociales se destaca la necesidad de utilizar metodologías científicas en sus estudios que permitan reflejar empíricamente sus resultados. En este punto, la codificación de los datos es uno de los procedimientos más densos pero imprescindibles para estos estudios.

Es por ello que el objetivo central de este proyecto sea ofrecer una herramienta automática de análisis para la rama de las ciencias políticas que utiliza la codificación manual como *modus operandi* en parte de su estudio. De este modo se facilita el trabajo de los investigadores sociales al permitir tratar un gran volumen de información, evaluándola y clasificándola, para posteriormente dar respuestas numéricas y estadísticas a sus estudios. Así pues, se plantea la informática como un instrumento de mejora en las tareas, no únicamente como un mero traspaso de información tradicional a la era digital.

Existen numerosos grupos de investigación que trasladan y explotan el contenido de, por ejemplo, los programas electorales, los debates de investidura o las noticias de los periódicos a los soportes informáticos de manera manual.

Las ventajas de informatizar los procedimientos manuales son muchas, y entre ellas las más destacadas podrían serla velocidad en obtener los resultados y el poco margen de error conseguido con el proceso automático. Cabe mencionar también, que cambiando los datos a analizar, la misma herramienta se puede seguir utilizando sin tener que modificar nada. En cambio, cada vez que se codifica manualmente, se ha de empezar el proceso desde cero, con todo lo que ello conlleva.

1.1 Metodología actual - Motivación

Con el fin de obtener conclusiones a partir de información escrita, académicos del área de las ciencias políticas han creado una metodología para clasificar textos, y posteriormente procesar los datos recogidos y presentar los resultados de una manera clara.

La metodología empleada se basa, primeramente, en la creación de unas categorías en las que poder definir los datos a estudiar. Estas categorías se representan con un código numérico de entre 2 y 4 caracteres en los que el primer carácter nos indica el dominio sobre el que se

engloban las descripciones, y a partir de este número se desglosa el resto de categorías. A continuación se muestra, como ejemplo, una tabla de códigos con su descripción[1]:

Domain 1: External Relations
101 Foreign Special Relationships: Positive
102 Foreign Special Relationships: Negative
103 Anti-Imperialism: Positive
104 Military: Positive
105 Military: Negative
106 Peace: Positive
107 Internationalism: Positive
108 European Integration: Positive
109 Internationalism: Negative
110 European Integration: Negative
Domain 2: Freedom and Democracy
201 Freedom and Human Rights: Positive
202 Democracy: Positive
203 Constitutionalism: Positive
204 Constitutionalism: Negative

Tabla 1. Códigos y descripciones

Por otro lado, existe un segundo código que no hace referencia al contenido de la oración, sino al ámbito territorial en el que dicha frase tiene su afectación. En este caso podemos tener ámbito local, autonómico o estatal, entre otros. De tal modo:

Level of government (first digit):
1 The local level
2 The regional (provincial, state) level
3 The national level
8 The European level
9 The international/global level
Preferred degree of authority (second digit):
1 The text unit (i.e. the quasi-sentence) claims less authority for the respective level
2 The text unit (i.e. the quasi-sentence) claims more authority for the respective level
0 The text unit (i.e. the quasi-sentence) contains no authority claim. It only states the level of government addressed by the policy preference, without claiming more or less competences for that particular level of government in that policy area.

Tabla 2. Códigos y descripciones (ámbito territorial)

[1] En el anexo se ofrece la lista de códigos y sendas descripciones.

Por lo tanto, de la unión de estos dos códigos obtenemos un tercero que aglutina y determina la frase con el siguiente criterio: "ámbito/contexto_temática". Es preciso destacar que en muchas ocasiones, la frase analizada no proporciona información sobre el ámbito del que hace referencia, por lo que los codificadores deben recurrir a leer la noticia (si lo que se está clasificando son los titulares) o el contexto en el que se encuentra.

En segundo lugar y, recuperando las cuestiones metodológicas, una vez se han establecido los criterios bajo los que se va a clasificar el texto, el siguiente paso a realizar será la separación del texto en frases o "cuasi-frases[2]" (que pueden estar formadas desde una única palabra, hasta una oración compleja). Este proceso es totalmente manual y requiere una persona especializada que conozca bien el tema que se está estudiando. Huelga destacar que para desempeñar las tareas de codificación es necesario un entrenamiento previo, que permita al codificador hacerse con los códigos y sus definiciones; que pueda captar los diferentes matices que coexisten entre códigos.

Hemospodido,enestecontexto,desarrollarpolíticasdevertebración territorialparacombatirlosdesequilibriosdemográficosinternosyluchar contraeldespoblamientorural,/
creerenunaeconomíasostenible, diversificada,equilibradaytecnológicamenteavanzada,/
construirunentramadodeserviciossociales,educativosysanitariosdecalidadfortaleciendoelestadodelbienestar,/
ponerenvalorlaspotencialidadesendógenasdenuestroterritorio,/

Tabla 3. Ejemplos de cuasi-frases

En este tercer punto, el texto está totalmente preparado para ser codificado, así que un experto en la temática, a partir de la lectura de cada frase o cuasi-frase puede establecer en qué ámbito se enmarca la frase en cuestión (primeros 2 dígitos) y sobre qué está hablando (siguientes 3-4 dígitos).

Según Laura Cabeza, codificadora experta del proyecto RMP, el tiempo estimado en realizar una codificación de un texto de alrededor de 200 páginas (sobre 4000 cuasi-frases/ frases) es de 2 meses dedicando un mínimo de media jornada únicamente a este cometido.

Conociendo estas cifras, se puede constatar la principal motivación y la gran necesidad que existe en automatizar este proceso, ya que muchas veces los datos que se consiguen extraer de estos textos, se encuentran ya fuera de contexto (como podría ser el estudio de los programas electorales para las elecciones), ya que ha pasado demasiado tiempo como para

[2] "Las palabras como unidad de codificación a veces se analizan dentro de unidades contextuales más amplias a las que pertenecen, principalmente la frase para asegurarse de que su significado queda recogido de forma correcta. […] Debido a que las frases largas pueden contener más de un argumento, se dividen a veces en cuasi-frases." *(Alonso et al., 2012)*

poder realizar un estudio en el que poder comparar diferentes partidos y que pueda ayudar a elegir al votante, por ejemplo.

1.2 Propuesta de solución

La propuesta ofrecida para la realización del proyecto se seguirá en dos pasos principales:

1. Extracción y clasificación de los datos
2. Tratamiento y muestra de resultados

En primer lugar, la extracción de los datos se basará en recopilar los textos referentes a programas electorales de diferentes partidos, en fechas distintas y de diferentes comunidades autónomas para, posteriormente, identificar sobre qué están hablando (*topicmodels*[3]). Se probará en este caso el algoritmo de clasificación *NaiveBayes*, ofreciéndole únicamente unos datos de muestra (entrenamiento) ya clasificados previamente.

Una vez clasificados todos los titulares, se procederá a la extracción de esta información para procesarla en Qlikview, programa que permite visualizar de manera organizada, rápida e intuitiva gran cantidad de datos. Los pasos estandarizados a seguir dentro de un proyecto de Qlikview siguen un proceso E-T-L (Extract, Transform, Load, o Extracción, Transformación y Carga).

Dado que en este caso, tanto el proceso de análisis como la carga de datos final la vamos a realizar de manera autónoma, se intentará en la medida de lo posible, minimizar el proceso de transformación de manera que los datos que extraigamos de la clasificación se encuentren en la forma que más rápidamente nos permita cargar, de manera modular y dinámica, toda la información en referencia al tema o temas de los que se vaya a realizar el estudio. De esta manera se garantiza disponer de una herramienta que se amolde a cualquier tipo de proyecto o análisis posterior.

Por tanto, en este trabajo se utilizarán como bases de datos las elaboradas por *Regional Manifesto Project* (RMP), pero no se descarta que en un futuro este mismo instrumento sirva para otro tipo de bases que tengan un objeto de estudio diferente.

[3]Un *topic model* es un tipo de modelo estadístico para descubrir temas abstractos que ocurren en una colección de documentos.

1.3 Estado del arte

En primer lugar, conviene destacar que no se ha encontradoninguna aplicación dedicada a clasificar noticias o textos políticos utilizando los parámetros demandados por estudios de esta índole, aunque sí que es bien conocido que importantes empresas utilizan este tipo de metodologías para analizar los intereses de los usuarios o consumidores, como es, por ejemplo, a través de las redes sociales o el correo electrónico.

En este sentido, abunda la documentación y bibliografía sobre cómo realizar clasificaciones automáticas a partir de unos datos de entrada, así como la información referente a diferentes algoritmos que están siendo estudiados y perfeccionados. La cuestión que pretendemos desarrollar puede ser definida como *topicmodel,* que es aquella que alude al descubrimiento de temas desconocidos o abstractos, intentando determinar cual es el contenido semántico de cada uno de los textos analizados.

En la actualidad, en referencia al estudio del contenido de estos datos políticos, existen diversas plataformas[4]que se dedican manualmente a categorizar y clasificar la información, tanto a nivel estatal, como a nivel internacional. Una de estas plataformas es el *Regional Manifesto Project* (MRP)[5], cuyas bases de datos han sido utilizadas en este proyecto como materia objeto de estudio. En este caso, los investigadores se han ocupado de distintos aspectos del funcionamiento de los partidos políticos, así como de la estructura y el desarrollo de los sistemas de partidos.

Respecto a la clasificación automática del contenido generado, se han consultado diferentes algoritmos para realizar la clasificación a partir de cálculos estadísticos. Dada la información obtenida por diversas fuentes, un algoritmo con alto índice de acierto en clasificaciones de texto es el algoritmo clasificador bayesiano ingenuo o 'NaiveBayes', el cual mediante unas probabilidades previas a la clasificación (dadas por la frecuencia de aparición de cada palabra en los datos de entrenamiento) y un resultado de los datos de entrenamiento, se calculan unas nuevas probabilidades que nos permiten determinar si una instancia pertenece o no al grupo que queremos clasificar.

[4]Se pueden encontrar ejemplos de estas clasificaciones en: https://manifesto-project.wzb.eu/ o http://www.regionalmanifestosproject.com/espaol/descarga-de-datos

[5]El proyecto desarrollado por el RMP se basa en análisis de contenido cuantitativo de los programas electorales de más de 50 países que cubren todas las elecciones libres y democráticas desde 1945. El RMP tiene como objetivo medir las preferencias políticas de los partidos que compiten en elecciones regionales, empleando para ello el análisis de contenido cuantitativo de los programas electorales.

Por otro lado, otros algoritmos que también pueden funcionar con un buen índice de aciertos es la *máquina de vectores de soporte* o 'SVM' (*Support Vector Machine*), que dado un conjunto de entrenamiento, se pueden etiquetar cada instancia a una clase y posteriormente entrenarlo para construir un modelo que permita la predicción a una clase de las nuevas muestras, del cual se hablará más adelante.

1.4 Objetivos

El principal objetivo de este proyecto es proporcionar un instrumento de análisis automático para los estudios en ciencia política que utilizan la codificación manual como metodología científica en parte de sus investigaciones. Con el producto resultante se pretende facilitar el trabajo a los investigadores, que reducirán el tiempo y mano de obra al realizar el proceso codificador.

Las ventajas de la informatización del proceso son cuantiosas, ya que en la eliminación de un proceso manual se incluyen mejoras temporales, puesto que se consigue reducir el tiempo de espera para conseguir los resultados. Así mismo, no es necesario esperar a que el codificador finalice sus funciones, sino que basta con incorporar las bases de datos en el programa para dejar que se encargue él de su catalogación, la cual realizará en un tiempo considerablemente inferior.

Por lo tanto, el objetivo no es sólo reducir el tiempo de la propia codificación, sino lograr evitar el tiempo previo en formación del investigador y en el tiempo posterior de análisis y visualización de los datos, puesto que con QlikView[6] el investigador no debe manipular los datos, sino que estos se codifican y se visualizan de acuerdo a las necesidades prescritas. Cabe destacar que el ahorro de tiempo se produce en todas las fases de un mismo proyecto, así como en otros que decidan utilizar este programa, ya que únicamente cambiarán los datos, puesto que los códigos seguirán siendo los mismos y el tiempo de espera se reducirá al tiempo de cálculo de la máquina.

Uno de los motivos centrales de este trabajo es poder ofrecer una herramienta útil, que se enmarque dentro de un nicho de mercado real. Para ello, el reto se encuentra en poder crear un programa con un porcentaje de acierto elevado, que permita reducir el margen de error a su máxima expresión y sea un producto plenamente fiable. Como es lógico, se busca ofrecer una herramienta que simplifique un procedimiento complejo además de tener un manejo simple e intuitivo.

[6]Qlikview, herramienta de análisis empresarial rápido, potente y altamente interactivo. http://www.qlik.com/

Gracias a una herramienta de estas características, el proceso se agilizará en diferentes fases puesto que los datos le vendrán dados al codificador de acuerdo a sus intereses. Y de esta manera, el investigador podrá dedicar el tiempo que se ahorra a otros aspectos de la investigación y centrarse, más tarde, en interpretar los datos obtenidos.

Existen múltiples ópticas a través de las cuales analizar el proyecto, como son: reducir el margen de error en la clasificación, ser capaz de aprender más rápidamente o necesitar un conjunto de datos de entrenamiento más pequeño. En este sentido, el interés principal de este proyecto reside en reducir al máximo los fallos de la clasificación y obtener el máximo porcentaje de aciertos.

2 Marco teórico

La clasificación de texto es una metodología empleada en gran parte de las ciencias sociales, que la emplean como mecanismo de conversión de información cualitativa a cuantitativa.

Esto se puede conseguir de manera manual o automática en base a un algoritmo que se procesa en una computadora. La clasificación intelectual de documentos ha sido el área de trabajo de la biblioteconomía, mientras que la clasificación algorítmica de documentos se usa principalmente en las ciencias de la información y computación.

Existen dos tipos de clasificación: basada en el contenido e indexada. En la primera, el peso dado a sujetos particulares en un documento determina la clase a la cual el documento se asigna. Esto es, por ejemplo, como en la regla de clasificación de libros en la que al menos el 20% del contenido del libro debe hablar sobre la clase a la que el libro está asignada. En cambio, en la clasificación indexada el usuario es quien decide en base a qué parámetros se va a realizar la clasificación.

2.1 Clasificación automática

Se ha hablado de que existen tanto la clasificación manual como la automática. En la clasificación automática existe un amplio campo de estudio y trabajo dentro de las ciencias de la computación, en el área de la Inteligencia Artificial[7] (IA).

En la IA, este tipo de clasificación se enmarca dentro de la capacidad de aprendizaje. Sin esta capacidad, los programas deberían estar dotados de inteligencia durante su diseño, y por tanto, la inteligencia emanaría del diseñador y su capacidad de prever la diversidad del entorno y sus posibles cambios. El aprendizaje es la capacidad que permitirá dotar al programa de la posibilidad de adquirir conocimiento y por tanto, proporcionarle autonomía.

[7] *"Computational intelligence is the study of the design of intelligent agents. An agent is something that acts in an environment- it does something. Agents include worms, dogs, thermostats, airplanes, humans, organizations, and society. An intelligent agent is a system that acts intelligently: What it does is appropriate for its circumstances and its goal, it is flexible to changing environments and changing goals, it learns from experience, and it makes appropriate choices given perceptual limitations and finite computation.*
The central scientific goal of computational intelligence is to understand the principles that make intelligent behavior possible, in natural or artificial systems. The main hypothesis is that reasoning is computation. The central engineering goal is to specify methods for the design of useful, intelligent artifacts."(Poole, Mackworth& Goebel 1998)

"*The goal of text categorization is the classification of documents into a fixed number of predefined categories*. […] Using machine learning, the objective is to learn classifiers from examples which do the category assignments automatically."[8]

El aprendizaje computacional se puede definir como aquel conjunto de métodos computacionales que aprenden de la experiencia, mejoran el rendimiento y permiten hacer cosas a un programa para las cuales no se había programado explícitamente.

El aprendizaje comprende un amplio espectro de metodologías, desde los procesos de memorización hasta la generalización de conceptos o el descubrimiento.

Existen diferentes técnicas y métodos utilizados para la clasificación automática, que son comentados en el siguiente apartado.

2.2 Métodos de clasificación automática

En la actualidad existen diferentes métodos de clasificación automática, y dependiendo de qué se quiera clasificar cada método puede aportar una serie de ventajas y hacer que la clasificación tenga unos mejores resultados.

Podemos separar los métodos de clasificación automática en dos grandes ramas: aprendizaje supervisado y aprendizaje no supervisado (clusterización).

2.2.1 Aprendizaje supervisado

El programa dispone de una serie de parejas (s_i, a_i) donde si corresponde a una percepción concreta y a_i a la acción que va asociada a ella, a las que se llaman ejemplos.

Estos ejemplos se suponen muestras de una función *f*. El programa debe realizar una inferencia inductiva de manera que construya una función *h*, llamada hipótesis, que sea coherente con los ejemplos mostrados, $h(s_i) = a_i$, y que se comporte bien con los casos no enseñados.

Consideremos por ejemplo encontrar una función que interpole o aproxime un conjunto de punto sobre el plano. Con las mismas restricciones de interpolación o aproximación tenemos muchas posibles funciones "solución", todas diferentes. La preferencia de una hipótesis sobre

[8]Joachims, Thorsten. Text categorization with support vector machines: Learning with many relevant features. London, UK, 1998

otra se llama similitud. Todos los algoritmos de aprendizaje tienen algún tipo de similitud, y es importante conocer cuál.

Si la función tiene valores discretos (simbólicos) entonces a los a_i se les llama clases y al proceso clasificación. Si a_i sólo tiene valores binarios, entonces a la función aprendida se le llama concepto y al método de aprendizaje conceptual. Un ejemplo de este último caso es "aprender" el concepto de "objeto rígido".

2.2.2 Aprendizaje no supervisado (clusterización)

Si el agente no dispone de las parejas (s_i, a_i) estamos en el caso no supervisado. Una forma de este tipo de aprendizaje es el aprendizaje por refuerzo, que utiliza una señal que le indica si lo que ha aprendido es correcto o no. Por ejemplo jugar al ajedrez utilizando como refuerzo el número de fichas perdidas y el número de fichas que ha perdido el adversario.

Otras formas de aprendizaje no supervisado son muy dependientes del objetivo. Los métodos de agrupación o clústering son muy utilizados en el reconocimiento de patrones. La idea es elegir una partición del espacio de las entradas en un número fijo de subconjuntos o clústeres, de manera que los elementos de un mismo clúster sean cercanos entre ellos, dada una cierta métrica.

Como se ha comentado anteriormente, el aprendizaje bayesiano es idóneo para la clasificación de textos, y por tanto es el método que vamos a detallar más exhaustivamente.

2.2.3 Aprendizaje bayesiano

El aprendizaje Bayesiano se basa en la asunción de que las cantidades de interés están gobernadas por distribuciones de probabilidades y que las decisiones óptimas se puede hacer razonando sobre estas probabilidades y sobre los datos observados.

Para este método, se supone que la respuesta a la incógnita de encontrar la mejor hipótesis de un espacio *H*, dado *D*, es la más probable dado ese espacio *D* y un conocimiento inicial sobre las probabilidades a priori de cada hipótesis.

Para obtener una solución válida y con fundamento en la realización de este proyecto se ha utilizado el ya conocido teorema de Bayes, enunciado por Thomas Bayes en 1763.

Este teorema expresa la probabilidad condicional de un evento aleatorio *X* dado en *Y* términos de la distribución de probabilidad condicional del evento *X* dado *Y*y la distribución de probabilidad marginal[9] de sólo *A*.

De una manera general, este teorema nos vincula la probabilidad de *X* dado *Y*y la probabilidad de *Y* dado *X*. Así que, por ejemplo, sabiendo la probabilidad de que suceda un evento como tener sueño dado que "es de noche", se podría saber la probabilidad de saber si es de noche cuando tienes sueño.

Se define:

Dado {$X_1, X_2, \ldots, X_i, \ldots, X_n$}, un conjunto de sucesos mutuamente excluyentes y exhaustivos[10], y tales que la probabilidad de cada uno de ellos es distinta de cero (*0*).
Sea *Y* un suceso cualquiera del que se conocen las probabilidades condicionales *P(Y|X)*.

Entonces, la probabilidad $P(X_i|Y)$ viene dada por la expresión (Ec.1):

$$P(X_i|Y) = \frac{P(Y|X_i)P(X_i)}{P(Y)} \qquad \textit{Ecuación 1}$$

donde:

- $P(X_i)$son las probabilidades a priori.
- $P(Y|X_i)$es la probabilidad de Y en la hipótesis A_i.
- $P(X_i|Y)$son las probabilidades a posteriori.

Con base en la definición de Probabilidad condicionada, obtenemos la Fórmula de Bayes (Ec.2), también conocida como la Regla de Bayes:

$$\mathrm{P(A_i|B)} = \frac{\mathrm{P(Y|X_i)P(X_i)}}{\sum_{k=1}^{n} \mathrm{P(Y|X_k)P(X_k)}} \qquad \textit{Ecuación 2}$$

Cuando se hace un recuento para calcular las probabilidades de ciertas muestras para determinadas categorías, nos podemos encontrar con que no existe ninguna coincidencia, y por tanto el valor de la probabilidad para este caso en concreto es *0*.

[9] Probabilidad condicional es la probabilidad de que ocurra un evento *A,* sabiendo que también sucede otro evento *B*. La probabilidad condicional se escribe *P(A|B).*
[10] Se consideran todos los posibles resultados.

Esto puede generar problemas, porque si este caso se repite con mucha frecuencia (normalmente con volúmenes muy grandes de datos que están agrupados en pocas características), al calcular el producto de las probabilidades va a salir igual a *0*.

Para evitarlo se utiliza la técnica llamada “Laplace Smoothing” (Ec.3) que se basa en

$$\mathrm{P}(\mathrm{X_i} = \mathrm{x_{ij}} | \mathrm{Y} = \mathrm{y_k}) = \frac{\#\mathrm{D}\{\mathrm{X_y}=\mathrm{x_{ij}}\ \hat{}\ \mathrm{Y}=\mathrm{y_k}\}+\mathrm{l}}{\#\mathrm{D}\{\mathrm{Y}=\mathrm{y_k}\}+\mathrm{lM}}, \mathrm{l} = 1 \qquad \textit{Ecuación 3}$$

donde $l = 1$ y M el número de diferentes valores de X_i.

Existen casos en que uno se puede encontrar con muchas características, que determinan si el elemento que se está estudiando pertenece o no a una determinada clase. En este caso, existe un grave problema con la fórmula habitual del cálculo de la probabilidad causado por imprecisión numérica, ya que si se multiplican muchos elementos con valor entre *0* y *1*, el resultado tenderá a *0*, y por tanto no se podrá clasificar correctamente.

Para poder solucionar este inconveniente se cambia la fórmula habitual, y en vez de aplicar multiplicaciones de la probabilidad (Ec. 4) se cambia a la suma de logaritmos (Ec. 5), pasando de

$$\mathrm{v} = \arg max_{\mathrm{v_j}\in\mathrm{V}}(\mathrm{P}(\mathrm{v_j}) \prod_{\boldsymbol{a_i}\in\boldsymbol{X}}(\boldsymbol{P}(\boldsymbol{a_i}|\boldsymbol{v_j}))) \qquad \textit{Ecuación 4}$$

a

$$\mathrm{v} = \arg max_{\mathrm{v_j}\in\mathrm{V}}(\log\mathrm{P}(\mathrm{v_j}) + \sum_{\mathrm{a_i}\in\mathrm{X}} \log\mathrm{P}(\mathrm{a_i}|\mathrm{v_j})) \qquad \textit{Ecuación 5}$$

2.2.3.1 Ejemplo.

Ponemos como ejemplo un caso en el que vamos a seguir todos los pasos del algoritmo de una manera visual, para comprender cómo funciona y poder explicar todos los detalles.

Utilizamos para este ejemplo 1000 muestras de fruta, de las cuales tenemos 3 características que nos definen cómo son estas frutas.

Las piezas de fruta que tenemos son plátanos, naranjas y “otros” (sin identificar).

- La primera característica describe la forma, si es redonda o no.
- La segunda característica es sobre el sabor y nos dice si es dulce o no.
- La tercera y última característica es referente al color y nos dice si es amarilla o no.

En la siguiente tabla se muestra un recuento de las frutas:

	Alargada	**No alargada**	**Dulce**	**No dulce**	**Amarilla**	**No amarilla**	**Total**
Plátano	400	100	350	150	450	50	500
Naranja	0	300	250	50	300	0	300
Otro	100	100	50	150	150	50	200
Total	500	500	650	350	800	200	1000

Tabla 4. Ejemplo de recuento

La *tabla 4* contiene toda la información necesaria para llevar a cabo el algoritmo.
Por un lado vamos a necesitar calcular las probabilidades a priori (que si no tuviéramos ningún dato referente a los atributos de cada fruta sería el valor que usaríamos para estimar si una fruta es una de las listadas.

P(plátano) = 500/1000 = 0.5 50% de probabilidad
P(naranja) = 300/1000 = 0.3 30% de probabilidad
P(otro) = 200/1000 = 0.2 20% de probabilidad

También calculamos la probabilidad de la evidencia (probabilidad de las características):
P(Alargado) = 0.5
P(Dulce) = 0.65
P(Amarillo) = 0.8

Por último calculamos la probabilidad condicionada P(A|B).

P(alargado |plátano) = 0.8	P(dulce |plátano) = 0.7	P(amarillo |plátano) = 0.9
P(alargado | naranja) = 0	P(dulce | naranja) = 0.75	P(amarillo | naranja) = 1
P(alargado | otro) = 0.5	P(dulce | otro) = 0.25	P(amarillo | otro) = 0.75

Ahora ya disponemos de todos los datos necesarios para calcular la probabilidad de cualquier nueva entrada.

Ahora hacemos la prueba, y nos viene una fruta con las siguientes características:
Alargada, dulce y amarilla.

Pasamos a calcular:

$$P(plátano|alargado, dulce, amarillo) = \frac{P(alargado|plátano)P(dulce|plátano)P(amarillo|plátano)}{P(alargado)P(dulce)P(amarillo)}$$

$$P(naranja|alargado, dulce, amarillo) = \frac{P(alargado|naranja)P(dulce|naranja)P(amarillo|naranja)}{P(alargado)P(dulce)P(amarillo)}$$

$$P(otro|alargado, dulce, amarillo) = \frac{P(otro|naranja)P(dulce|otro)P(amarillo|otro)}{P(alargado)P(dulce)P(amarillo)}$$

Y obtenemos los siguientes resultados:
P(plátano | alargado, dulce, amarillo) = 0.252/P(evidencia)
P(naranja | alargado, dulce, amarillo) = 0
P(otro | alargado, dulce, amarillo) = 0.01875/P(evidencia)

Comprobamos y vemos que 0.252>0.01875 y por tanto el algoritmo retorna que esta fruta, a partir de las citadas características es un plátano ya que tiene más probabilidades de serlo.

2.2.4 Máquinas de vectores de soporte (SVM)

Las máquinas de soporte vectorial o máquinas de vectores de soporte (Support Vector Machines, SVMs) son un conjunto de algoritmos de aprendizaje supervisado desarrollados por Vladimir Vapnik y su equipo.(Cortes y Vapnik, 1995)

Estos métodos están propiamente relacionados con problemas de clasificación y regresión. Dado un conjunto de ejemplos de entrenamiento (muestras) podemos etiquetar las clases y entrenar una SVM para construir un modelo que prediga la clase de una nueva muestra. Intuitivamente, una SVM es un modelo que representa a los puntos de muestra en el espacio, separando las clases por un espacio lo más amplio posible. Cuando las nuevas muestras se ponen en correspondencia con dicho modelo, en función de su proximidad pueden ser clasificadas a una u otra clase.

2.2.5 K Nearest Neighbors (K-NN)

El algoritmo K-nearestneighbor es el método basado en instancias más básico. Es un método de clasificación supervisada que sirve para estimar la probabilidad a posteriori de que un elemento x pertenezca a la clase C_j a partir de la información proporcionada por el conjunto de prototipos

Permite a una instancia arbitraria x ser descrita por el vector de características

[a1(x), …, an(x)],

donde ar(x) denota el r-ésimo atributo de la instancia x. Después la distancia entre dos instancias xi y xj se define por la distancia euclidiana d(xi, xj)[11.]

Para el caso de aproximar funciones de valores discretos f: Rn en V, donde V es el set finito {V1, …, Vn}, el algoritmo devuelve

$$f(x_q) = \operatorname{argmax}_{v \in V} \sum_{i=1}^{k} \delta(v, f(x_i)) \qquad \textit{Ecuación 6}$$

Donde *δ(a, b)=1* si *a=b* y *δ-(a,b)=0* en caso contrario, y $x_1, \ldots, x_k$ denotan las k instancias que son cercanas a x_q.

[11] La distancia euclidiana o métrica euclídea es la distancia ordinaria entre dos puntos que se puedan medir con una regla, y está dada por la fórmula de Pitágoras.

3 Herramienta Business Intelligence

3.1 Qué es Business Intelligence

Antes de poder explicar qué es el *BI* (o Business Intelligence), hay que entender una serie de conceptos y problemáticas que se pretenden solucionar.

3.1.1 La problemática de los *ERP* o Sistemas de Gestión

Los sistemas *ERP*[12] han tenido una gran acogida durante los últimos años. Han ayudado a las empresas a adquirir un gran volumen de datos sin prescindir del control. No obstante, la sensación de muchos directivos es que estos programas no son adecuados para disponer de la información necesaria para tomar decisiones (capa decisional[13]). Se pueden extraer listados e incluso gráficos, pero llegar a la información que se necesita resulta extremadamente complicado y tedioso.

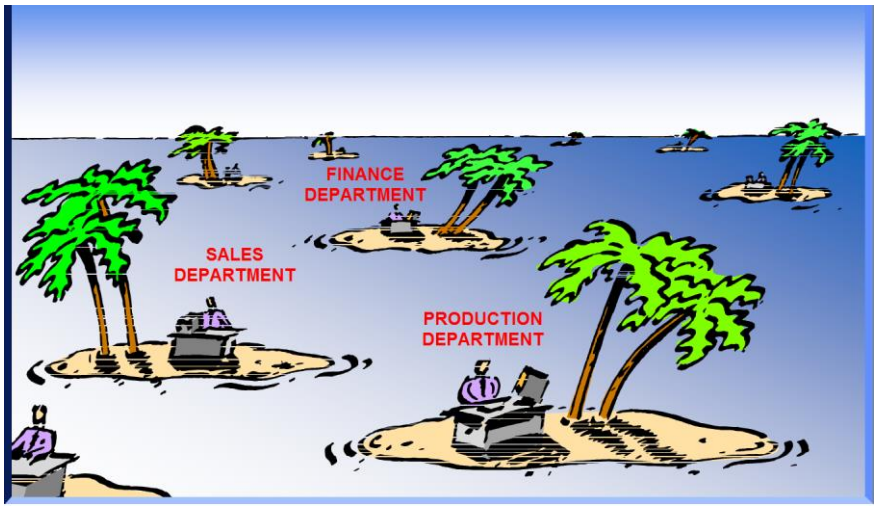


Figura 1. Islas de información (extraído de: http://www.baesis.com/Portals/52892/images/Islands%20of%20Info.png)

De la misma manera, la complejidad de los negocios implica que muchas empresas hayan adquirido distintas soluciones para resolver problemáticas específicas del sector como podría

[12]Enterprise ResourcePlanning (ERP) o Sistemas de Planificación de Recursos Empresariales se llama a los sistemas de planificación de gerencia que integran y manejan muchos de los negocios asociados con las operaciones de producción y de los aspectos de distribución de una compañía en la producción de bienes o servicios.

[13]Se llama capa decisional a los miembros de una empresa que se dedican a tomar decisiones sobre las acciones que va a llevar a cabo la empresa, normalmente directivos y altos cargos.

ser un Control de Presencia para controlar las horas que se trabaja en taller, Sistemas de *CAD*/*CAM*[14] para desarrollar y escandallar productos, herramientas de control como los presupuestos y sistemas de retribución variables desarrollados en *Excel*...

El problema es, que estas islas de información y la enorme dispersión de datos, hace que cada vez sea más difícil ponerle orden y, por ende, cada vez es más complicado satisfacer las necesidades de información del directivo.

En definitiva, el *ERP* no es una herramienta para uso gerencial y directivo en general, y por tanto se deben buscar nuevas soluciones.

3.1.2 BI o Business Intelligence

Los sistemas *BI* sirven para resolver muchos de los puntos débiles de los *ERP*: Recogen información de diferentes fuentes. Da igual donde ubique la información y el formato en el que lo almacene. *BI* captura y almacena la información proveniente de cualquier origen de datos.

Hace una ordenación de los Datos. Organiza todos los datos recabados, unifica criterios ya que cada aplicación maneja sus propios maestros (por ejemplo, clientes y proveedores), jerarquiza los distintos conceptos (por ejemplo: Gama de Productos → Familias de Producto → Producto) y permite calcular indicadores tan agregados como sea necesario (desde la Facturación hasta el Beneficio Neto, por ejemplo).

Todo ello, para visualizar los resultados en una serie de Cuadros de Control que permiten una enorme interactividad. La información es interactiva, se puede filtrar haciendo clic sobre las casillas y bucear por los distintos indicadores a través de las distintas jerarquías.

[14] CAD/CAM son las siglas que definen a "diseño asistido por computador" y "fabricación asistida por computador".

3.2 Qué es Qlikview

Qlikview es una solución líder de *BI* que permite recolectar datos desde diferentes orígenes, basados en *ERP*, *CRM*[15], *datawarehouses*[16], bases de datos *SQL*, *MSExcel*, etc., modelar al gusto del usuario para facilitar su manejo y presentar esos datos de forma muy visual y atractiva.

3.2.1 Qué aporta

Una de las principales ventajas del uso de *Qlikview* es la diversidad de plataformas en las que se puede utilizar, ya que se puede disponer de su uso desde sólo un ordenador personal, hasta la versión Server, en la que se disponen diferentes programas ya desarrollados y accesibles desde diferentes plataformas, como pueden ser móviles, tablets/ipad, ordenadores (en diferentes sistemas operativos) todo gracias a su versión web…

Qlikview aporta un lenguaje propio de modelado de datos, lo cual permite trabajar con la información directamente desde la base de datos y transformarla adecuadamente. Cabe destacar que este lenguaje es complicado para aquel que sea ajeno a la programación, aunque en su defensa podemos comentar que existe una gran comunidad[17] con soluciones y aportaciones para poder desarrollar las aplicaciones que necesites.

A diferencia de muchas otras herramientas de BI, Qlikview utiliza un modelo asociativo de datos, que se carga directamente en memoria.

Existe disparidad de opiniones en cuanto a este asunto, ya que unos defienden que este tipo de modelo no es óptimo para volúmenes grandes de datos, ya que se convierte en un gran problema de necesidad de hardware al no poder cargar tantos datos en memoria y por tanto no es recomendable utilizar este sistema.

El otro enfoque es que las posibilidades que ofrece este modelo teniendo todos los datos cargados, es la velocidad para poder visualizarlos y la potencia de ser capaces de verlos, tanto

---

[15] Software para la administración de la relación con los clientes. Sistemas informáticos de apoyo a la gestión de las relaciones con los clientes, a la venta y al marketing. Con este significado CRM se refiere al sistema que administra un data warehouse con la información de la gestión de ventas y de los clientes de la empresa.

[16] Almacenamiento de información homogénea y fiable

[17] http://community.qlik.com/welcome

los que están relacionados con la selección actual, como los datos que no tienen nada que ver, que siguen cargados y se pueden observar.

Este último punto es muy importante ya que permite ampliar el abanico de posibilidades en el Business Discovery[18]. La forma de presentar esta funcionalidad es cambiando el color de la selección de los datos (*Fig 2*). Los datos que tienen relación con el valor seleccionado, que estará de color verde, se mostrarán en color blanco, y todos aquellos valores que no tengan una conexión con la selección actual se mostrarán en color gris.

Figura 2. *Selección*

Gracias a esto se pueden detectar problemas de falta de información o falta de integridad en la base de datos.

Una de las mejores aportaciones que ofrece Qlikview es la visualización de datos. A parte de la gran cantidad de gráficos y tablas que se pueden utilizar, el concepto utilizado para la selección de datos es una gran ventaja.

[18]A diferencia del BI tradicional, en el que sólo unas cuantas personas están implicadas en la creación de conocimiento, el Business Discovery permite a todos crear conocimiento, pues da acceso a los datos a cada grupo de trabajo, departamento y unidad de negocio para que puedan tomar las mejores decisiones. Las empresas pueden llevar el conocimiento a todos los puntos de su organización, permitiendo a todos los usuarios hacer su trabajo de una forma mucho más rápida y eficaz que nunca. Permite a todos los usuarios crear un conocimiento personalizado que satisfaga sus necesidades únicas de negocio así como sus plazos.

3.3 Fases de un proyecto de BI (ETL)

Como en cualquier proyecto técnico que se pretenda realizar, existe un seguido de fases y pasos que se deben completar, como puede ser una toma de requerimientos, una planificación, un estudio de viabilidad, etc.

El objetivo de un proyecto de BI es dotar a la empresa de los medios necesarios para que obtenga la información precisa para la toma de decisiones. Así pues, la idea es que cualquier responsable de la empresa pueda disponer de información relevante y actualizada antes de tomar cualquier decisión importante para la empresa. Este objetivo puede implicar a diferentes niveles de la empresa desde el operativo hasta el estratégico.

El primer paso que hay que realizar es el análisis, esta fase consiste en comprobar las necesidades actuales y futuras. Esta es una parte clave del proyecto. Si no se identifican correctamente las necesidades, difícilmente se podrán satisfacer.

En la siguiente fase, de análisis, se debe ser capaz de dar respuesta a dos preguntas básicas: ¿qué información se necesita para desempeñar el actual trabajo? y ¿qué información gustaría tener para poder hacer mejor el trabajo?

Una vez establecidos los objetivos específicos del proyecto, lo siguiente es definir el modelo de datos**.** El modelo de datos es una representación conceptual de las magnitudes que se manejan en la empresa y las relaciones que hay entre éstas**.**

Los requerimientos de información identificados durante la anterior fase proporcionarán las bases para realizar el diseño y la modelización del Data Warehouse, a partir de ahora DW. La misión del equipo del proyecto es conseguir un modelo de datos que sea homogéneo y claro, y que sea capaz de dar cabida a todas las necesidades de la empresa**.**

Más adelante se identificarán las fuentes de los datos (sistema operacional, fuentes externas,..) y las transformaciones necesarias para, a partir de dichas fuentes, obtener el modelo lógico de datos del DW. Este modelo estará formado por entidades y relaciones que permitirán resolver las necesidades de negocio de la organización.

El modelo lógico se traducirá posteriormente en el modelo físico de datos que se almacenará en el DW y que definirá la arquitectura de almacenamiento del mismo, adaptándose al tipo de explotación que se realice de él.

La mayor parte de las definiciones de los datos del DW estarán almacenadas en los metadatos y formarán parte del mismo Data Warehouse.

La implantación de un Data Warehouse lleva implícitos los siguientes pasos:

3.3.1 1ª fase: Extract (extracción)

En esta primera fase únicamente se buscan TODOS los datos existentes en las diferentes fuentes y sistemas de los que se dispone y se extraen todos los datos al programa.

Normalmente los datos provienen de orígenes y sistemas diferentes y cada uno de ellos en un formato distinto.

Los formatos de las fuentes normalmente se encuentran en bases de datos relacionales o ficheros planos, pero pueden incluir bases de datos no relacionales u otras estructuras diferentes. La extracción convierte los datos a un formato preparado para iniciar el proceso de transformación.

3.3.2 2ª fase: Transform (transformación)

Una vez están "cargados" todos los datos dentro del programa, entra en juego toda la lógica de negocio, y para poder calcular todo aquello que posteriormente se quiere mostrar, se ha de hacer un amplio estudio y análisis.

Se deben realizar una serie de modificaciones y transformaciones con el fin de obtener los datos que se querrán mostrar más adelante. Algunos de estos cambios pueden ser:

- Seleccionar datos de sólo algunas columnas
- Codificar valores
- Unir (join) tablas de múltiples fuentes
- Transponer tablas
- Crear agregaciones
- Generar valores de índice

3.3.3 3ª fase: Load (carga)

Con todos los datos cargados y pre-calculados se llega al momento en el que traducir todos los números en gráficos y tablas organizadas, además de otros objetos que se pueden cargar (mapas, indicadores…).

Por norma general, se colocan unos cuadros en los que se puedan seleccionar las diferentes dimensiones de las que se compone el modelo de datos, y con su selección se filtran los datos que se muestran en el centro de la pantalla en forma de tabla o gráfico.

La siguiente fase consiste en la realización de la capa de presentación o de los informes, cuadros de mandos y otros elementos de visualización a través de los cuales los usuarios accederán a la información. Cuando la información está bien presentada y el formato de los informes ofrecidos tiene un buen diseño, los usuarios son capaces de interpretar la información con mucha más facilidad, eficiencia y con un menor número de errores.

3.4 Ejemplo de Dashboard para ciencias políticas

El siguiente ejemplo está sacado de la web de demostración de Qliktech[19]. Es una aplicación en la que revisar, estudiar y analizar datos sobre las elecciones presidenciales para la campaña de Obama en las elecciones estadounidenses del pasado año 2012.

En este caso se trata de un informe de análisis de sentimiento, en el que se puede buscar información tanto sobre Barak Obama como de su contrincante MittRomney y comprobar las tendencias que hay sobre los temas listados con cada uno de los candidatos a partir de datos obtenidos de las redes sociales como es Twitter.

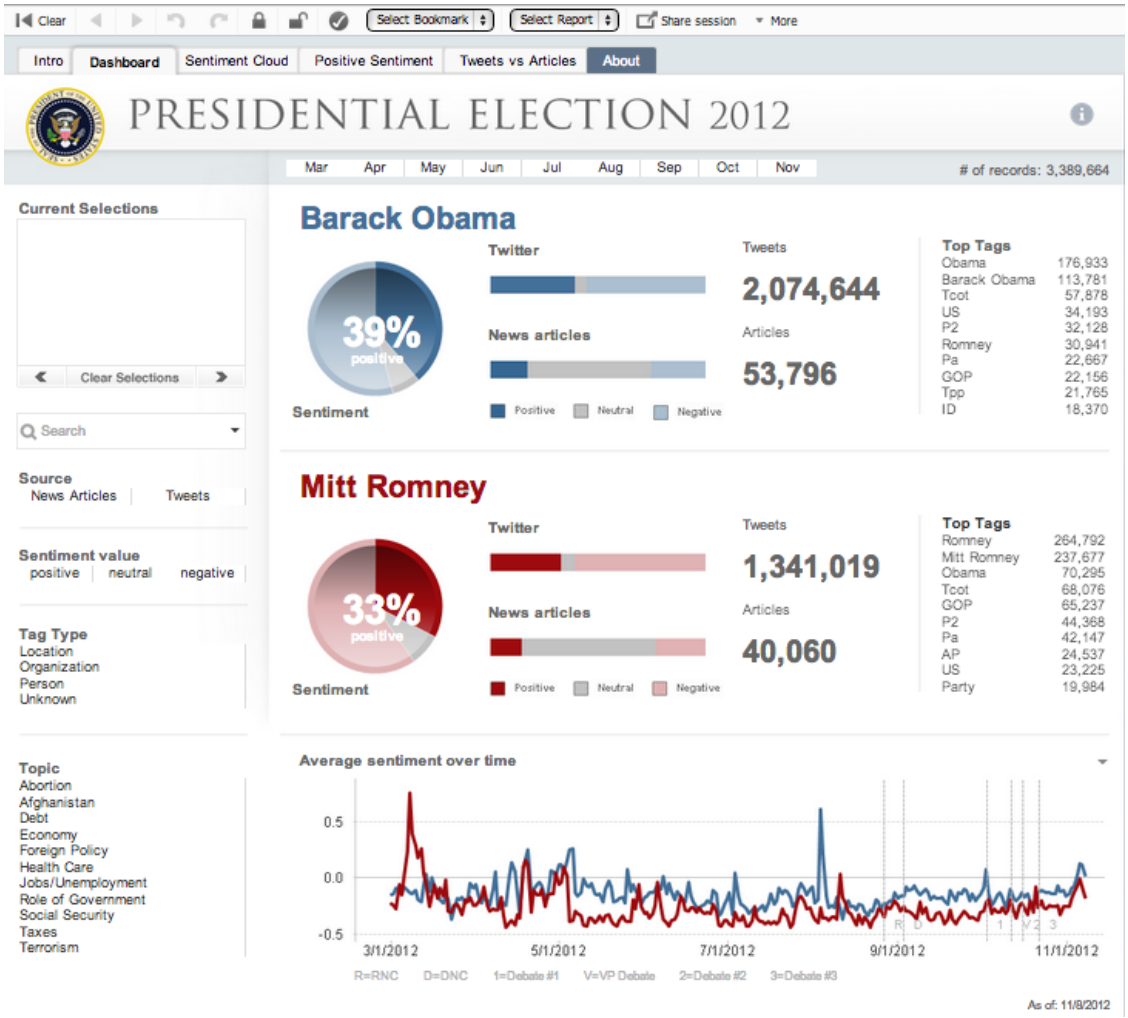


Figura 3. *Ejemplo de Dashboard de ciencias políticas (extraído de: http://eu-b.demo.qlik.com/QvAJAXZfc/opendoc.htm?document=qvdocs%2F2012%20Presidential%20Election.qvw&host=demo11&anonymous=true)*

[19]El anterior ejemplo y muchos otrosmás se puedenconsultar en: http://eu-b.demo.qlik.com/

4 Diseño general y datos técnicos

Para implementar el código necesario para la clasificación del texto se ha utilizado el entorno de desarrollo y lenguaje Matlab, ya que es un entorno conocido por su gran capacidad con el trato de matrices y la velocidad que se obtiene en los cálculos con ellas .

Se han utilizado tanto funciones propias del lenguaje, como otras que se han creado a partir de la necesidad del problema a tratar.

Ha sido de mucha ayuda el uso del foro de ayuda de Matlab[20], así como su manual. Y cabe destacar la gran ayuda proporcionada por el resto de usuarios de Matlab al exponer y comentar sus dudas por diferentes webs, sobre todo en stackoverflow[21].

Los datos de entrada han sido cedidos por el grupo de investigación de Regional Manifesto Project que sin ninguna reserva han entregado sus textos clasificados manualmente para poder llevar a cabo el proyecto y tener unos datos con los cuales poder comparar el resultado obtenido por el algoritmo.

Estos datos han sido entregados en formato Word (*Tabla 5*), y constaban de una serie de tablas con la cuasi-frase en la primera casilla y el código que representa a esa frase.

MODERNIZAR LAS ESTRUCTURAS ECONÓMICAS PARA CREAR EMPLEO ESTABLE Y DE CALIDAD	
La política económica que defendemos los socialistas es la que favorece el desarrollo de nuestra sociedad, desde el punto de vista cuantitativo pero sobre todo cualitativo, para cada individuo pero sobre todo para la colectividad, para el conjunto de la población. /	20_408
Es verdad que sólo una economía fuerte, sustentada en un tejido empresarial sólido, competitivo e innovador es capaz de garantizar el estado de bienestar que defendemos y al que en ningún caso renunciamos. /	20_504
Educación, sanidad y políticas sociales para todos, basadas en la solidaridad intergeneracional que permita asegurar las mejores	20_504

Tabla 5. *Ejemplo de datos extraídos de fichero MS Word.*

El modus operandi seguido para la transformación de estos datos ha sido copiar el contenido de la tabla a una hoja de cálculo (Excel) en la cual se ha acabado de formatear para quitar los espacios en blanco y otros.

[20] http://www.mathworks.es/es/help/index.html

[21] http://stackoverflow.com/

Una vez obtenida una tabla sin “huecos”, se crean dos archivos de texto (txt formateados como ANSI[22]) y en el primero de ellos introducimos el texto y en el segundo introducimos el código. De esta manera, tenemos la primera frase (o cuasi-frase) en la primera fila, y su código correspondiente en la primera fila del archivo de códigos.

Los nombres utilizados para estos archivos se diferencian en “_text” al final del nombre para los archivos que contienen el texto y “_sol” para aquellos que tienen los códigos o soluciones.

Otro programa que se ha utilizado ha sido un “stemmer” que es un programa en el que a la entrada de una palabra, retorna la raíz de la misma. Esto sirve para no diferenciar términos que son el mismo pero en el que se ha cambiado el género o número. Por ejemplo para el conjunto de las palabras “rojo”, “roja”, “rojos”, “rojas”, la salida del stemmer devuelve “roj”. De esta manera todas las probabilidades de las palabras “modificadas” no desvirtúan el valor real de la probabilidad del concepto que las engloba.

Para crear el reporte/informe/cuadro de mando se ha utilizado Qlikview, herramienta de Business Intelligence para la explotación y tratamiento de datos.

En el cuadro de mando creado para el proyecto se pueden elegir fechas y otros filtros que nos permiten adentrarnos en los datos e investigar de una manera fácil, rápida y visual. “La información está a un solo clic de distancia”.

[22]Estándar para la codificación del juego de caracteres relacionado con Microsoft y una modificación del juego de caracteres ASCII. El código ASCII utiliza 7 bits para representar a cada carácter, mientras que el formato ANSI utiliza 8 bits para cada uno.

5 Diseño técnico e implementación

A continuación se detallan los pasos seguidos a la hora de diseñar e implementar la solución. Este apartado está dividido en cuatro sub-apartados; en el primero se detalla el proceso de clasificación a partir de la entrada de datos hasta aplicar el método de Bayes ingenuo, en el segundo se explica lo referente a la creación de un juego de pruebas y test para evaluar los resultados, en el tercero se detallan esos resultados y les da sentido, y por último en el cuarto apartado se detalla la creación de un "cuadro de mando" capaz de mostrar los datos procesados de una manera fácil para la comprensión y estudio.

5.1 Clasificación

5.1.1 Datos de entrada

Los datos de entrada que se han utilizado como fuente constan de unos archivos en formato Word en los que hay unas tablas con el par (frase | código) codificado de forma manual previamente por personas expertas en el tema.

Posteriormente se transforman a partir de importar estas tablas a MS Excel se crean 2 archivos *txt* a partir del anterior, uno para las frases (cada frase en una línea nueva) y otro para los códigos (cada código en una línea nueva).

El principal motivo de haber utilizado este tipo de datos ha sido por la facilidad de utilizar para un usuario sin experiencia en formatos fuera de nivel usuario y por tanto que su uso sea lo más cómodo como sea posible para personas ajenas a la informática.

5.1.2 Pre-procesado de los datos de entrada

Los datos que llegan en origen no pueden ser utilizados directamente por el clasificador, ya que dentro de todos los textos se encuentran muchos caracteres que no interesan, como pueden ser signos de puntuación, números o caracteres especiales.

También se debe tener en cuenta que si el objetivo es conocer el tema del que habla una frase, existen ciertas palabras que no tienen un significado relevante a ésta, y por tanto no deben ser tenidas en cuenta.

Además de todo esto, si queremos ser precisos en nuestros procesos, existen también muchas palabras que son derivadas de una misma palabra o raíz teniendo todas el mismo significado para ese mismo concepto.

Por tanto, los pasos a seguir para filtrar y eliminar todo aquello que no resulta útil para el clasificador son los que siguen:

- Lo primero de todo es eliminar los números que encontramos en el texto, ya que sólo nos vamos a fijar en las palabras (letras).
- Existen también caracteres especiales, como los puntos, comas etc. Todos estos caracteres son eliminados del texto.
- Se pasa un programa externo que devuelve la palabra base de la que derivan sus derivados encontrados en el texto (Stemmer[23]). Por ejemplo, si en el texto encontramos las palabras “rojos”, “rojas”, “rojo”, “roja”, para el mismo concepto (rojo) tenemos 4 posibilidades, así que este programa nos devuelve “roj” para cada una de ellas. De esta manera, haciendo que la lista de palabras que nos retorne el programa sea única, sólo debemos quedarnos con el concepto “roj” una vez.
- Por último, eliminamos todas esas palabras que no tienen un significado relevante a priori en ninguna oración, como son los pronombres, determinantes o preposiciones.

Completando todos estos pasos, conseguimos un texto “limpio” en el que vamos a encontrar muy pocas interferencias procedentes de duplicados de palabras (en concepto) o diferentes y diversas causas.

Con todos los pasos anteriores procesados, la información relevante al texto, se copia con el mismo formato de filas a un nuevo texto, cambiando cada palabra única por un valor numérico, referente a la posición que ocupa dentro de la lista de palabras únicas, identificando así cada palabra con un valor único.

[23]Programa obtenido de la web http://snowball.tartarus.org/ que utiliza el sistema Snowball para generar stemmers para diferentes idiomas.

5.1.3 Método Bayes

Para la implementación del método de Bayes se han seguido una serie de pasos y se ha organizado el código. Se ha separado la implementación en funciones para facilitar las tareas, tener claridad y modular el código:

- Lanzador
- Recuento
- Probabilidad
- Transformación
- Asignación
- Validación

Para una muestra del flujo (*Fig. 4*) que sigue el programa, se muestra el siguiente diagrama de flujo:

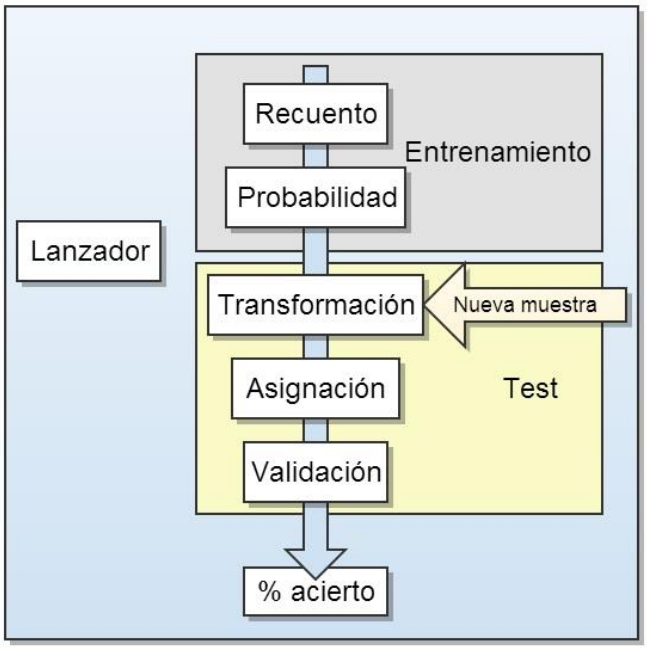


Figura 4. *Diagrama de flujo de ejecución*

Yendo en el orden en el que se ejecuta el proceso, lo primero que hay que explicar el lanzador, qué es y en qué consiste. Este proceso no es más que un elemento de conexión y orden entre los diferentes estados en los que va a seguir la información.

Dentro encontramos unos bucles que van a servir para las pruebas, en caso de saber con qué parámetros lanzar el proceso (optimización). Se aporta una versión sin los bucles y con las variables disponibles para la parametrización elegida.

Los bucles que encontramos son referentes a Stem/NoStem, StopWords/NoStopWords, Laplace/NoLaplace, Sum/Prod, Priori/NoPriori que detallamos más adelante.

El primer paso del algoritmo consiste en leer un archivo de texto, donde tenemos las frases que nos van a servir de entrenamiento, que identificamos como “XXX_text.txt”, y también el archivo “XXX_sol.txt” en el que tenemos las clasificaciones o clases asignadas manualmente.

En el caso del archivo de soluciones (o clasificaciones) cargamos todas las líneas y las se almacenan en una lista, a la que posteriormente se le aplica una reducción para eliminar todos los elementos repetidos y así obtener una lista de clasificaciones única.

Para el archivo con el texto el proceso es más elaborado. Primero se carga entero y se eliminan todos aquellos caracteres que sean números. Se hace también un recuento de las palabras diferentes que hay, y sabiendo esta cantidad se creará una matriz de tantas filas como palabras únicas hay en el texto, y tantas columnas como códigos únicos haya en el fichero de soluciones.

Mientras se realiza el proceso de copiado, se comprueba cada palabra leída (de la fila conocida) y en la misma fila del archivo de soluciones se comprueba el valor al que está asignada la palabra, y en la matriz de recuento a la palabra que estamos analizando le sumamos un 1. Así conseguimos contar todas las palabras utilizadas en cada frase para cada categoría.

En este momento cabe destacar que si la opción de stop words está activada en vez de comprobar todas las palabras, antes de leer el texto, se pasa la lista de palabras únicas a otra función que extrae todas aquellas palabras que se han determinado “sin significado relevante” y de esta manera para cada palabra que comprobemos del texto, nunca las tendremos en cuenta.

También es ahora cuando se comprueba si la opción de stemwords está activa, ya que si es así también se pasará por una función externa que transformará todas las palabras eliminando el sufijo que hace variar la palabra desde su forma básica. Realizando esta acción se consigue unificar muchas variantes de la misma palabra que de otra forma habrían quedado separadas y tenidas en cuenta como si fueran conceptos diferentes, alterando así el resultado a la hora de clasificar.

El siguiente paso del proceso total calcular el porcentaje relativo para cada palabra en cada clase. Teniendo calculada la matriz de recuento es un cálculo muy fácil que sólo consiste en dividir cada fila entre el total de la suma de la propia fila. Así conseguimos tener la probabilidad condicionada de cada palabra para cada clase $P(Wi|Ci)$.

En este caso, también existe la variante de si se quiere aplicar Laplace Smoothing o no, que consiste en sumar un 1 al numerador y el valor del total de palabras diferentes del texto en el denominador.

En este proceso se calcula también cual es la probabilidad a priori que tiene cada clase, la cual se calcula a partir de (número de veces que aparece la clase Ci)/(total de clases).

Con todos estos datos calculados el proceso se encuentra listo para poder calcular un dato de entrada y poder clasificarlo según la probabilidad hallada con la fórmula de Bayes.

En este apartado también existe una bifurcación, y es según si vamos a comprobar frases nuevas, o vamos a hacer test para comprobar cómo reacciona el algoritmo con los mismos datos con los que ha entrenado.

El caso con el que más se va a tratar es con nuevas frases/cuasi-frases, ya que es para lo que está destinado el algoritmo, clasificar texto nuevo. Pero para la realización de la mayoría de los test, se han utilizado los mismos datos de entrenamiento que los de test, ya que el ámbito que abarcan las palabras de cada texto de entrenamiento no tienen por qué coincidir con las de los test. En el caso ideal sí sería así, pero se considera que hacen falta muchos más datos de entrenamiento para que la muestra sea suficientemente significativa.

Utilizando nuevas oraciones, se debe pasar por un nuevo proceso de transformación, donde se volverá a pasar por todos los filtros anteriores (stem, stop words) para poder escribir un nuevo fichero con la forma predefinida para poder comparar.

Teniendo ya transformado el fichero de test, se lee línea a línea, y por cada palabra (que es un número referente a la fila de la matriz de recuento) se carga su información a una nueva matriz auxiliar, a la que se efectuarán los cálculos pertinentes.

Estos cálculos también requieren conocer si el parámetro de suma/producto está activado o no, ya que el cálculo será diferente en función de él. Teniéndolo activado, significa que la fórmula se efectuará calculando el producto de las probabilidades, y si está desactivado será la suma de logaritmos de las probabilidades, para intentar evitar el error producido por las palabras inexistentes en la matriz de recuento que harían que el producto fuera 0.

Una vez hecha la multiplicación o la suma, se busca el valor más alto (máximo) para cada frase, y éste determina cual es la clasificación con mayor probabilidad de pertenencia, y por tanto la que sale elegida.

Para finalizar se pasa por el proceso de validación, que recoge los resultados obtenidos en el punto anterior y los compara con los resultados reales provenientes del archivo de soluciones, y la división entre la cantidad de aciertos y el total de muestras da el porcentaje de acierto que se ha tenido que es devuelto al programa principal (lanzador) y se guarda en un archivo de texto, junto con el tiempo que ha tardado el proceso en completarse.

5.2 Validación y pruebas

Se han efectuado una serie de pruebas y validaciones para testear los resultados obtenidos cambiando el máximo de parámetros para encontrar la mejor configuración que permita clasificar los textos con un porcentaje de acierto lo más elevado posible, independientemente del coste computacional, ya que la mejora entre el tiempo empleado en la clasificación automática contra la manual es incuestionable.

Se ha creado un juego de pruebas que combinan todos los parámetros y variables que se pueden cambiar para obtener un resultado diferente, y comprender a qué atiende cada archivo para la clasificación.

El primer juego de pruebas que se ha creado, ha sido para cada archivo (referente a un parido político determinado en un año determinado).

En el segundo juego de pruebas, se ha agrupado a los partidos políticos en el mismo archivo, y se ha lanzado agrupado.

La última prueba realizada ha sido con todos los archivos juntos en uno solo, creando así el archivo de entrenamiento que debiera ser usado para clasificar cualquier entrada.

En vista a los resultados que se iban obteniendo se ha observado que utilizar unas frases “nuevas” que no pertenezcan al grupo de entrenamiento carece de sentido, ya que no conocemos el universo[24] del que está formado cada archivo de texto, y por tanto, es posible que intentemos clasificar palabras que no pertenecen al conjunto del texto en particular. No se debe actuar de esta manera con el texto que agrupa todos los archivos, ya que entendemos que es suficientemente significativo como para poder clasificar correctamente.

Así que lo que se ha decidido es comprobar en cada texto como clasificaría el mismo texto, o al menos aquellas palabras relevantes de cada frase del mismo texto, y poder comprobar entonces si el

[24] Es la totalidad de elementos o características que conforman el ámbito de un estudio o investigación.

peso otorgado por el algoritmo a cada palabra es suficiente como para poder clasificar, al menos, los conceptos de los que habla.

A continuación se detallan los parámetros que se han cambiado para utilizar diferentes configuraciones.

5.2.1 Con/sin stemming

El proceso de stemming, de utilizar o no derivados de palabras nos permite, si lo aplicamos, eliminar el género, número, conjugación y más formas que puedan tener las palabras. De esta manera sólo se tiene en cuenta el concepto de la palabra base y el peso de ésta no se ve perjudicado por distribuir las probabilidades entre diferentes derivados de la misma.

Lo que se espera conseguir con este proceso es poder reducir el número de palabras diferentes con el mismo concepto, y así hacer que la precisión en el acierto sea mayor.

5.2.2 Con/sin stop words

Las “stop words” son palabras que no aportan significado dentro de una oración, como pueden ser los artículos, preposiciones…

La eliminación de esas palabras permite que el resto de palabras obtengan un peso más elevado dentro de cada oración, aportando así un dato más fiable (a priori) para el cálculo de las probabilidades.

Para el caso del castellano, que es el idioma elegido para esta clasificación, la lista de palabras “stop words” es la que sigue:

ahí, tal, de, aquí, allí, allá, la, que, el, en, y, a, los, del, se, las, por, un, para, con, no, una, su, al, lo, como, más, pero, sus, le, ya, o, este, sí, pues, decir, entonces, vez, porque, esta, entre, cuando, muy, sin, sobre, también, me, hasta, hay, donde, quien, desde, todo, nos, durante, todos, uno, les, ni, contra, otros, ese, eso, ante, ellos, e, esto, mí, antes, algunos, qué, unos, yo, otro, otras, otra, él, tanto, esa, estos, mucho, quienes, nada, muchos, cual, poco, ella, estar, estás, algunas, algo, nosotros, mi, mis, tú, te, ti, tu, tus, ellas, nosotras, vosotros, vosotras, os, mío, mía, míos, mías, tuyo, tuya, tuyos, tuyas, suyo, suya, suyos, suyas, nuestro, nuestra, nuestros, nuestras, vuestro, vuestra, vuestros, vuestras, esos, esas, ser, haber, tener, hacer, estar.

5.2.3 Con/sin Laplace Smoothing

En este apartado nos encontramos con dos casos diferenciados:

- Palabras que no aparecen en la categoría X_i y se aplica Laplace Smoothing.
- Palabras que sólo aparecen 1 vez en la categoría X_i y no se aplica Laplace Smoothing.

En ambos casos, nos encontramos con que al hacer el recuento de palabras, se puede comprobar que determinada palabra no se encuentra ninguna vez en alguna de las categorías. Si sucede esto, como el sistema está diseñado para que se realicen unas multiplicaciones con las frecuencias relativas obtenidas de dicho recuento, hará que la multiplicación dé 0.

Esto es un problema, dado que con un sistema muy grande, en el que se contemplen muchas posibles categorías, este caso se dará en más de una ocasión originando tales problemas.

Con tal de solucionarlo, en el momento de calcular la matriz de frecuencia dada la matriz de recuento, se cambia la fórmula y se aplica una pequeña corrección sumando en todos los elementos 1 unidad, evitando de esta forma los ceros.

Se han tomado métricas con ambas opciones, tanto teniendo en cuenta esto, y por tanto aplicando Laplace Smoothing, y por lo contrario sin hacerlo y calculando la frecuencia de la manera habitual.

5.2.4 Con todas las categorías/ sólo sub-códigos

Dado que se dispone un código de categoría que consta de dos sub-códigos, se puede plantear la resolución del problema desde el punto de vista de usar estos dos códigos por separado, o intentar clasificar todos los datos utilizando el código completo.

A primera vista, parece que tiene más sentido separar los dos códigos, ya que son independientes en cuanto a concepto, y por tanto no debería incluirse la información que proporciona la asignación a un sub-código con la que proporciona el otro.

5.2.5 Por partido político

Se ha comprobado que si se quiere clasificar un texto completo con diferentes partidos políticos, de ideologías diferentes, el resultado de esta clasificación cambia del orden de más del 10% de precisión a peor, haciendo que no sea tan ajustado.

Teniendo en cuenta este factor, se ha pensado que la mejor solución es agrupar a los partidos entre ellos, de manera que una primera clasificación nos indique qué partido es el autor de las frases que se están clasificando. Una vez conocido el partido, se vuelve a lanzar una clasificación, esta vez sabiendo con qué grupo de entrenamiento pertenece (referente al partido político elegido).

5.2.6 Con probabilidades *a priori*

En ocasiones, el proceso se encuentra en la situación en que diversas palabras comparten diferentes clasificaciones, y esto puede inducir a problemas. El motivo que induce este tipo de conflicto yace en que si las mismas palabras se encuentran con una frecuencia similar en 2 o más clases, éstas no serán capaces de mostrar si la frase que se analiza pertenece a una clase u otra.

Debido a esta necesidad se debe tener en cuenta el valor obtenido por la probabilidad *a priori*, ya que éste va a determinar si existe más probabilidad de que una palabra forme parte de una clase en mayor o menor proporción. Si de lo contrario se comprueba que existe un problema con el balanceo de las palabras respecto las clases a las que pertenecen, se puede dar el caso en que una misma palabra aparezca en más ocasiones dentro de una clase, pero que el número de ocurrencias de la misma y de toda la clase sea muy pequeña en comparación con la clase predominante. En este caso, si se omite la probabilidad *a priori*, lo que prima en este caso será el porcentaje de las palabras únicamente, y entonces la clasificación se verá modificada.

5.2.7 Resumen de pruebas

Los resultados obtenidos realizando las pruebas comentadas anteriormente se muestran en una tabla ordenada para visualizar mejor la gran cantidad de datos recogidos.

Las pruebas han consistido en 32 configuraciones diferentes por cada archivo de texto con sus soluciones. Se detallan los parámetros a continuación:

Como se puede observar en la *tabla 6*, se utiliza un código binario para identificar si se está usando un parámetro (1) o no (0), y de tal manera viene indicado en la siguiente tabla, donde encontramos el porcentaje de aciertos de cada ejecución del algoritmo para 15 casos diferentes (archivos individuales).

Se muestra en las siguientes páginas los resultados (en porcentaje de aciertos) obtenidos para cada una de las configuraciones mostradas anteriormente.

Stemmedwords	Stop words	Laplace smoothing	Prod(prob) /sum(log)	Prob priori
0	0	0	0	0
0	0	0	0	1
0	0	0	1	0
0	0	0	1	1
0	0	1	0	0
0	0	1	0	1
0	0	1	1	0
0	0	1	1	1
0	1	0	0	0
0	1	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	0
0	1	1	0	1
0	1	1	1	0
0	1	1	1	1
1	0	0	0	0
1	0	0	0	1
1	0	0	1	0
1	0	0	1	1
1	0	1	0	0
1	0	1	0	1
1	0	1	1	0
1	0	1	1	1
1	1	0	0	0
1	1	0	0	1
1	1	0	1	0
1	1	0	1	1
1	1	1	0	0
1	1	1	0	1
1	1	1	1	0
1	1	1	1	1

Tabla 6. *Configuraciones de ejecución*

Además de las pruebas comentadas antes, se ha hecho una prueba con la unión de 15 de los textos de los que se disponía. En dicha prueba se ha querido comprobar cómo funciona el programa con una cantidad de datos más grande, y en este caso se ha entrenado para determinar qué partido ha sido el autor de la frase analizada.

En esta primera tabla (7) se hallan los resultados de las pruebas 1-16

% ACIERTOS	00000	00001	00010	00011	00100	00101	00110	00111	01000	01001	01010	01011	01100	01101	01110	01111
ARA PSOE 2011	97.44	97.44	97.44	97.44	32.35	28.86	32.35	28.86	98.06	97.60	98.06	97.60	83.24	73.55	83.24	73.55
AST PSOE 2007	93.72	93.60	93.72	93.60	35.14	33.43	35.14	33.43	95.35	94.68	95.35	94.68	67.91	61.58	67.87	61.58
AST PSOE 2009	93.73	93.42	93.73	93.42	33.05	31.82	33.05	31.82	94.87	94.32	94.87	94.32	60.82	55.81	60.82	55.81
CLM PSOE 1983	99.04	99.04	99.04	99.04	41.97	37.41	41.97	37.41	99.28	98.80	99.28	98.80	94.96	86.33	94.96	86.57
GEN PP 2011	97.70	97.44	97.70	97.44	40.14	36.74	40.14	36.74	98.28	98.23	98.28	98.23	86.33	77.92	86.33	77.92
GEN PSOE 2011	97.10	97.05	97.10	97.05	42.29	40.91	42.29	40.91	97.38	97.15	97.38	97.15	80.16	71.17	80.11	71.17
MAD PSOE 2011	96.50	96.35	96.50	96.35	40.22	37.92	40.22	37.92	97.34	97.13	97.39	97.13	80.54	71.73	80.59	71.73
NAV PSOE 2011	93.59	93.33	93.59	93.33	39.34	37.29	39.34	37.29	95.31	94.63	95.31	94.63	73.15	67.13	73.19	67.13
PV PNV 2005	96.89	96.47	96.89	96.47	40.45	37.01	40.45	37.01	97.98	97.31	97.98	97.31	87.22	74.26	86.96	74.26
PV PP 2005	97.33	97.27	97.33	97.27	34.66	32.56	34.66	32.56	98.30	97.90	98.30	97.90	82.56	73.24	82.61	73.24
PV PSOE 2005	94.83	94.48	94.83	94.48	28.01	26.53	28.01	26.53	96.16	95.07	96.16	95.07	69.33	57.94	69.33	57.94
PV PSOE 2009	93.82	93.48	93.82	93.48	35.13	33.28	35.13	33.28	68.86	68.43	68.86	68.43	55.03	48.70	55.03	48.70
PV PSOE 2012	95.67	95.48	95.67	95.48	33.29	30.74	33.26	30.71	96.35	96.13	96.35	96.13	75.42	69.15	75.45	69.15
VAL PP 2011	94.63	94.34	94.63	94.34	31.05	29.30	31.05	29.30	78.03	77.63	78.03	77.63	60.93	52.80	60.93	52.80
VAL PSOE S007	94.77	94.58	94.77	94.58	38.97	37.31	38.97	37.31	95.86	95.43	95.86	95.43	67.79	58.40	67.76	58.40

Tabla 7. *Porcentajes de acierto de las configuraciones analizadas*

En esta segunda tabla (8) se hallan los resultados de las pruebas 17-32

% ACIERTOS	10000	10001	10010	10011	10100	10101	10110	10111	11000	11001	11010	11011	11100	11101	11110	11111
ARA PSOE 2011	94.03	93.87	94.03	93.87	28.86	26.69	28.86	26.69	94.88	94.72	94.88	94.72	67.26	59.35	67.26	59.35
AST PSOE 2007	85.79	85.42	85.79	85.42	31.20	30.27	31.20	30.27	88.25	87.32	88.25	87.32	55.34	50.50	55.34	50.50
AST PSOE 2009	82.20	81.92	82.20	81.92	28.92	27.86	28.92	27.86	85.85	84.87	85.85	84.87	49.26	46.09	49.26	46.09
CLM PSOE 1983	98.32	98.32	98.32	98.32	37.89	32.13	37.89	32.13	98.08	98.08	98.08	98.08	88.97	79.62	89.21	79.62
GEN PP 2011	94.68	94.26	94.68	94.26	35.18	32.57	35.23	32.57	96.09	95.77	96.09	95.77	73.38	65.40	73.33	65.40
GEN PSOE 2011	91.96	91.82	91.96	91.82	38.49	37.77	38.49	37.77	93.01	92.63	93.01	92.63	64.08	57.90	64.08	57.90
MAD PSOE 2011	91.29	90.98	91.29	90.98	36.41	35.00	36.41	35.00	92.80	92.23	92.80	92.23	66.88	58.48	66.82	58.48
NAV PSOE 2011	85.68	85.32	85.68	85.32	33.52	31.50	33.52	31.50	88.55	87.63	88.55	87.63	59.32	55.32	59.32	55.32
PV PNV 2005	94.28	93.94	94.28	93.94	38.77	34.99	38.77	34.99	95.96	94.87	95.96	94.87	77.38	65.60	77.38	65.60
PV PP 2005	94.32	94.03	94.32	94.03	30.51	28.52	30.51	28.52	95.40	94.89	95.40	94.89	67.95	60.00	67.90	60.00
PV PSOE 2005	87.90	87.57	87.90	87.57	25.06	24.09	25.06	24.09	90.47	89.05	90.47	89.05	51.74	44.07	51.74	44.07
PV PSOE 2009	85.61	85.22	85.61	85.22	31.27	29.80	31.27	29.80	64.50	63.85	64.50	63.85	45.12	41.10	45.17	41.10
PV PSOE 2012	87.98	87.67	87.98	87.67	27.40	25.54	27.38	25.51	90.50	89.85	90.53	89.85	58.77	52.71	58.71	52.71
VAL PP 2011	87.64	87.29	87.64	87.29	27.90	26.25	27.90	26.25	73.55	72.80	73.55	72.80	49.00	42.80	49.00	42.80
VAL PSOE S007	87.59	87.37	87.59	87.37	35.75	34.43	35.75	34.43	89.63	88.85	89.63	88.85	55.01	49.03	55.04	49.03

Tabla 8. *Porcentajes de acierto de las configuraciones analizadas*

5.2.8 Equipo utilizado (hardware)

El equipo que se ha utilizado para programar todo el proceso y para realizar todas estas pruebas ha sido el siguiente:

Procesador: Intel® Core™ i5-2500K Processor (6M Cache, up to 3.70 GHz)

Memoria RAM: G.SkillRipjaws X DDR3 1333 PC3-10666 8GB 2x4GB CL9
Disco: 2x320Gb 5400rpm Seagate Barracuda
Tarjeta gráfica: ATI Radeon HD 6850 1GB DDR5

5.3 Resultados

Para cada una de las configuraciones descritas anteriormente se ha realizado una matriz de confusión[25], la cual nos permite visualizar aquellos elementos que no han sido bien clasificados, en qué lugar se han clasificado.

A modo de introducción se puede indicar que los mejores resultados se han obtenido utilizando stemmed words y eliminando las stop words, ya que ha reducido considerablemente el número de palabras sin significado en cada frase, y ha permitido encontrar el núcleo o concepto principal de cara oración.

Todos los resultados que se muestran a continuación han sido de clasificaciones del post código, a no ser que se mencione lo contrario.

5.3.1 Con/sin stemming

% ACIERTOS	ARA PSOE 2011	AST PSOE 2007	AST PSOE 2009	CLM PSOE 1983	GEN PP 2011	GEN PSOE 2011	MAD PSOE 2011	NAV PSOE 2011	PV PNV 2005	PV PP 2005	PV PSOE 2005	PV PSOE 2009	PV PSOE 2012	VAL PP 2011	VAL PSOE S007
10000	97.44	93.72	93.73	99.04	97.70	97.10	96.50	93.59	96.89	97.33	94.83	93.82	95.67	94.63	94.77
00000	94.03	85.79	82.20	98.32	94.68	91.96	91.29	85.68	94.28	94.32	87.90	85.61	87.98	87.64	87.59

Tabla 9. *Porcentajes de acierto con/sin stemming*

Eliminando todas aquellas palabras que son derivados de la misma, se reduce el número total de palabras distintas de todo el conjunto de los textos, con lo que el recuento de cada palabra

[25]Herramienta de visualización que se emplea en aprendizaje supervisado. Cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias en la clase real. Uno de los beneficios de las matrices de confusión es que facilitan ver si el sistema está confundiendo dos clases.

por categoría se ve incrementado. Esto hace que los conceptos referentes a las raíces de las palabras ganen importancia respecto al total de palabras.

Tal y como se puede observar en la *tabla 9*, el porcentaje de aciertos en 15 textos a los que se ha aplicado el método de clasificación, únicamente cambiando el parámetro referente a "stemming", produce una sensible mejora entre ambos modelos. Estos resultados demuestran que siempre que se usa este método se consigue una mejora en el resultado final.

Se encuentra una mejora en el valor medio obtenido utilizando el filtrado de las stemmed words de valor 5.92% de media.

En el peor de los casos (CLM PSOE 1983) la mejora es sólo del 0.68%, aumentando así desde el 98.32% hasta el 99.04%, poco significativo dado que el margen de mejora en este caso era de sólo el 1.68%.

En cambio, en el caso en el que más evidente es la mejora (AST PSOE 2009), alcanza el 11.53% de diferencia, llevando el valor desde 82.90% hasta el 93.73%.

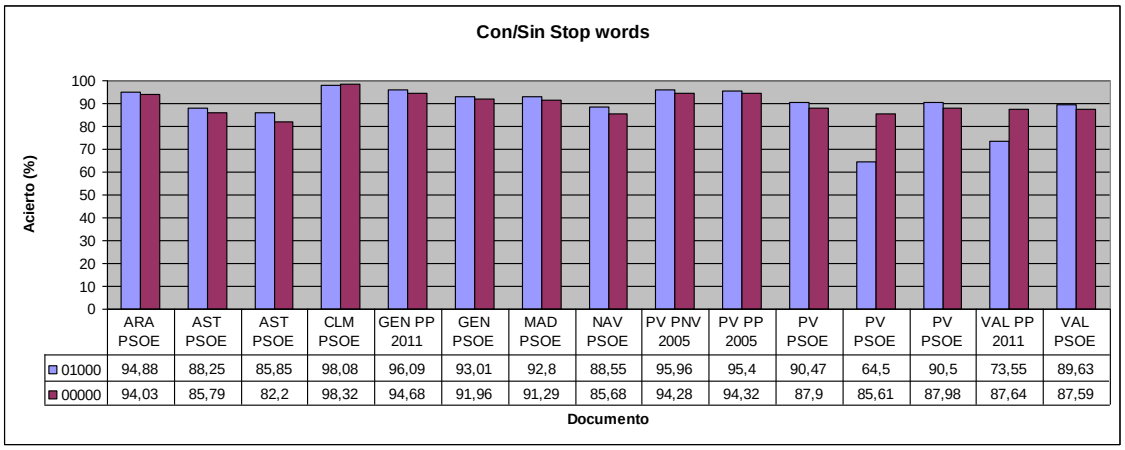


Figura 5. Comparativa con/sin stemmed words

5.3.2 Con/sin stop words

% ACIERTOS	ARA PSOE 2011	AST PSOE 2007	AST PSOE 2009	CLM PSOE 1983	GEN PP 2011	GEN PSOE 2011	MAD PSOE 2011	NAV PSOE 2011	PV PNV 2005	PV PP 2005	PV PSOE 2005	PV PSOE 2009	PV PSOE 2012	VAL PP 2011	VAL PSOE S007
01000	94.88	88.25	85.85	98.08	96.09	93.01	92.80	88.55	95.96	95.40	90.47	64.50	90.50	73.55	89.63
00000	94.03	85.79	82.20	98.32	94.68	91.96	91.29	85.68	94.28	94.32	87.90	85.61	87.98	87.64	87.59

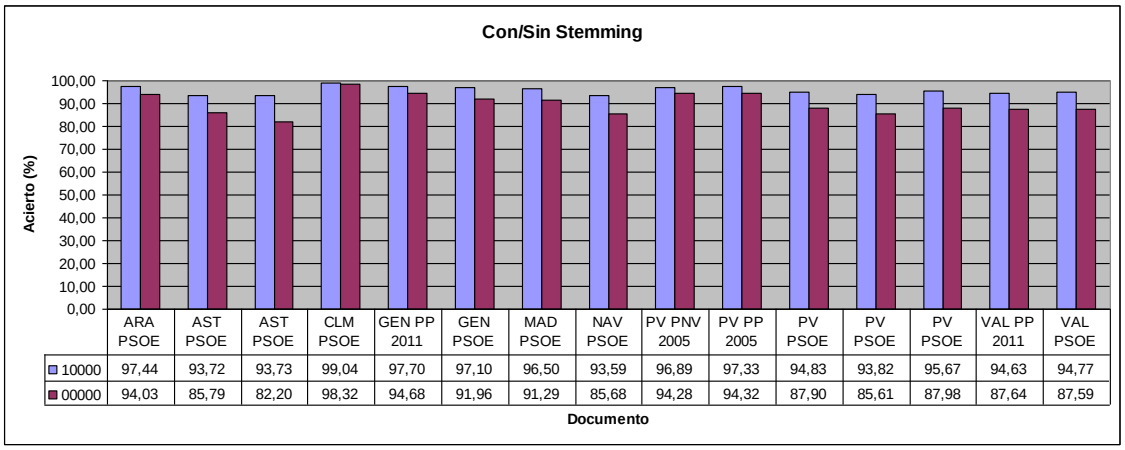
Tabla 10*. Porcentajes de acierto con/sin stop words*

El uso del filtro de stop words (ver *tabla 10*) permite eliminar del recuento todas las palabras añadidas en la lista con el mismo nombre. Como en el caso anterior, stemmed words, el objetivo es dar más peso a los conceptos más relevantes de cada oración.

El principal problema es que no se encuentran suficientes palabras que identifiquen una categoría en particular, ya que las palabras que la forman serán comunes a muchas categorías diferentes.

De manera similar a la aplicación del filtro de stemmed words, aplicar este método proporciona mejores resultados que dejarlo desactivado. Aunque en este caso existen ciertos textos que no cumplen con la hipótesis planteada. El más destacado es en el texto PV PSOE 2009, en el que se alcanza un 21.11% negativo. Esto es causado por la distribución de las palabras, y la cantidad de éstas contadas en cada una de las clases. El principal problema es que no se encuentran suficientes palabras que identifiquen una categoría en particular, ya que las palabras que la forman serán comunes a muchas categorías diferentes.

En el caso de emplear el filtro StopWords tan sólo se encuentra una mejora en el valor medio obtenido de un 1.16%.



	ARA PSOE	AST PSOE	AST PSOE	CLM PSOE	GEN PP 2011	GEN PSOE	MAD PSOE	NAV PSOE	PV PNV 2005	PV PP 2005	PV PSOE	PV PSOE	PV PSOE	VAL PP 2011	VAL PSOE
10000	97,44	93,72	93,73	99,04	97,70	97,10	96,50	93,59	96,89	97,33	94,83	93,82	95,67	94,63	94,77
00000	94,03	85,79	82,20	98,32	94,68	91,96	91,29	85,68	94,28	94,32	87,90	85,61	87,98	87,64	87,59

Figura 6. Comparativa con/sin stop words

5.3.3 Con/sin palabras frecuencia = 0 (Laplace)

% ACIERTOS	ARA PSOE 2011	AST PSOE 2007	AST PSOE 2009	CLM PSOE 1983	GEN PP 2011	GEN PSOE 2011	MAD PSOE 2011	NAV PSOE 2011	PV PNV 2005	PV PP 2005	PV PSOE 2005	PV PSOE 2009	PV PSOE 2012	VAL PP 2011	VAL PSOE S007
00100	28.86	31.20	28.92	37.89	35.18	38.49	36.41	33.52	38.77	30.51	25.06	31.27	27.40	27.90	35.75
00000	94.03	85.79	82.20	98.32	94.68	91.96	91.29	85.68	94.28	94.32	87.90	85.61	87.98	87.64	87.59

Tabla 11. *Porcentajes de acierto Laplace*

Tal y como se comprueba en los porcentajes de acierto proporcionados en la *tabla 11*, los valores que se obtienen aplicando o no Laplace Smoothing tienen una notable diferencia. Mientras que los valores en el caso en el que no se aplica el filtro son ciertamente altos, al aplicar el método sufren un gran cambio y la precisión baja considerablemente. Esto se debe a que los valores obtenidos en el cálculo de la matriz de frecuencia son muy bajos, causado por el bajo número de palabras a repartir entre todas las categorías. Dado que se habla de muchos temas diferentes, cuanto más bajo es el número de palabras a tener en cuenta se pierde efectividad.

Para que la aplicación del método de *Laplace Smoothing* sea efectiva, se debería disponer de un texto lo suficientemente largo y con un mínimo de repeticiones de palabras para que al aplicar la división a la matriz de recuento no hiciera que los valores resultantes en gran parte de la matriz fueran 0.

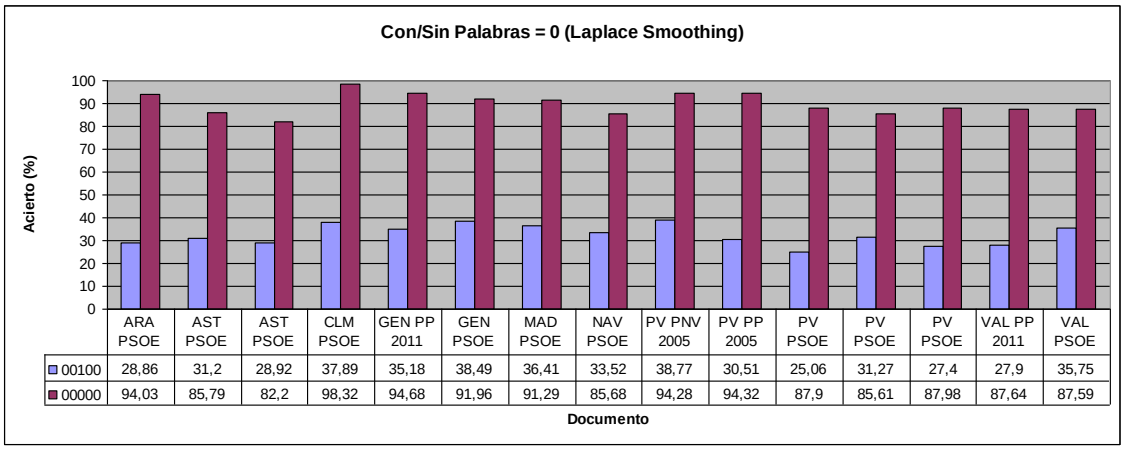


Figura 7. Comparativa con/sin Laplace Smoothing

5.3.4 Prod(prob) / sum(log)

% ACIERTOS	ARA PSOE 2011	AST PSOE 2007	AST PSOE 2009	CLM PSOE 1983	GEN PP 2011	GEN PSOE 2011	MAD PSOE 2011	NAV PSOE 2011	PV PNV 2005	PV PP 2005	PV PSOE 2005	PV PSOE 2009	PV PSOE 2012	VAL PP 2011	VAL PSOE S007
00010	94.03	85.79	82.20	98.32	94.68	91.96	91.29	85.68	94.28	94.32	87.90	85.61	87.98	87.64	87.59
00000	94.03	85.79	82.20	98.32	94.68	91.96	91.29	85.68	94.28	94.32	87.90	85.61	87.98	87.64	87.59

Tabla 12. *Porcentajes de acierto Prod(prob)/sum(log)*

Como se puede observar en la *tabla 12*, el hecho de utilizar sólo la suma de logaritmos o el producto de las probabilidades no afecta en nada a la clasificación.

Esto es debido a que la distribución de las palabras es uniforme en las palabras que más se repiten, lo que quiere decir que las mismas palabras son las más comunes en todas las categorías, y por tanto el peso de estas palabras no es determinante.

Otra razón de peso es que la probabilidad a priori en este caso se impone frente a las probabilidades de las palabras, dando así mucha más importancia al uso de esta probabilidad sin que el valor del resto de probabilidades, tanto en el productorio como en la suma de logaritmos, sea de importancia.

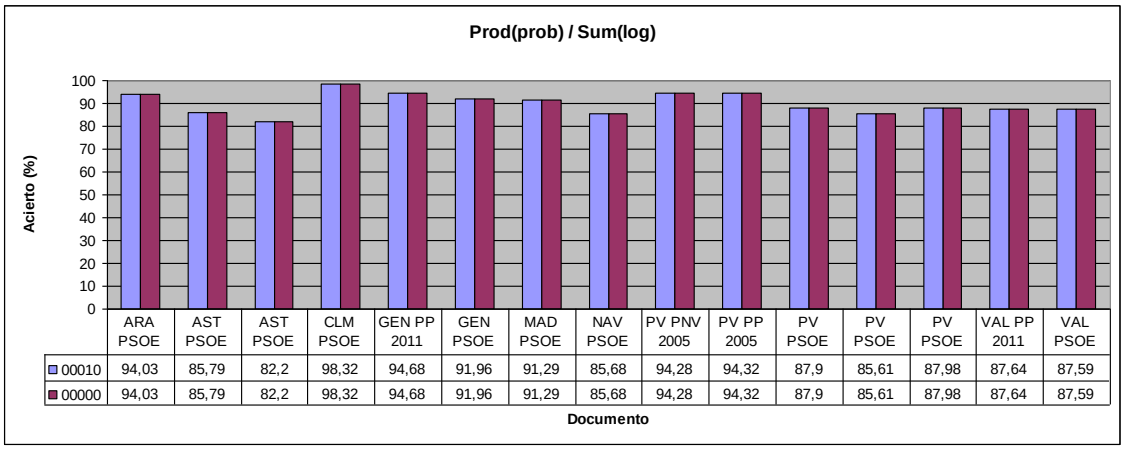


Figura 8. Comparativa prod(prob) / sum(log)

5.3.5 Con / Sin probabilidad a priori

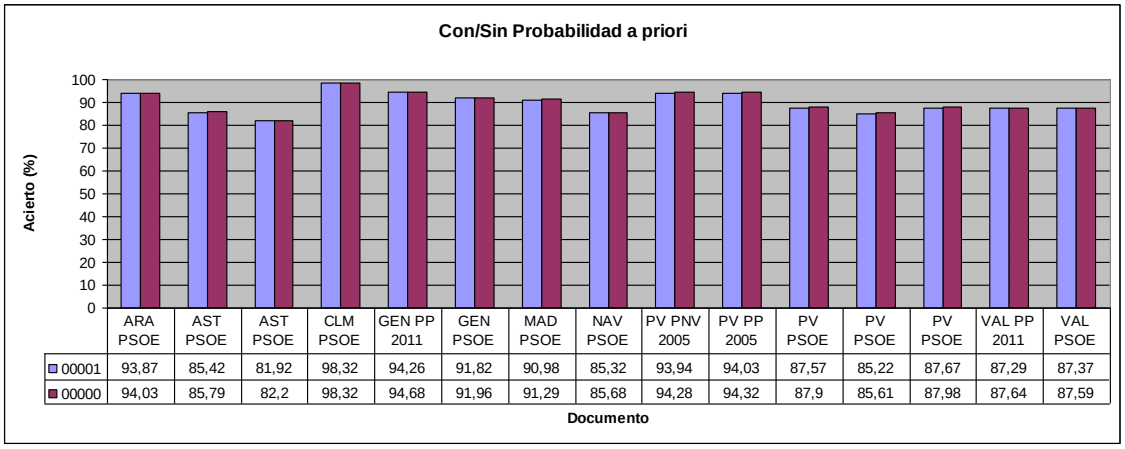
% ACIERTOS	ARA PSOE 2011	AST PSOE 2007	AST PSOE 2009	CLM PSOE 1983	GEN PP 2011	GEN PSOE 2011	MAD PSOE 2011	NAV PSOE 2011	PV PNV 2005	PV PP 2005	PV PSOE 2005	PV PSOE 2009	PV PSOE 2012	VAL PP 2011	VAL PSOE S007
00001	93.87	85.42	81.92	98.32	94.26	91.82	90.98	85.32	93.94	94.03	87.57	85.22	87.67	87.29	87.37
00000	94.03	85.79	82.20	98.32	94.68	91.96	91.29	85.68	94.28	94.32	87.90	85.61	87.98	87.64	87.59

Tabla 13*. Porcentajes de acierto con/sin probabilidad a priori*

Para conjuntos en los que las probabilidades de los datos estén muy desbalanceadas y se encuentran clases con mucha cantidad de información frente a clases con muy poca, puede ser buena idea observar el comportamiento del algoritmo sin utilizar las probabilidades a priori de cada clase y así comprobar el peso que tienen las palabras sin más.

En caso de conjuntos de datos relativamente pequeños como los que se tienen en cuenta en este estudio (ver *tabla 13*), la orientación (significado) de los textos está muy enfocada a temas (clases) muy determinados. Fruto de este hecho es que los datos obtenidos en la clasificación muestran muy poca afectación en cuanto al porcentaje de aciertos resultante.

En cambio, con textos que incluyan otras temáticas además de las que existen en estos, sí que puede ser útil utilizar este filtro para comprobar resultados. Poniendo un ejemplo sencillo, si en dos categorías se utilizan las mismas palabras, pero en una se tiene 10 veces más datos que en la otra, aunque el porcentaje que representa la palabra a evaluar sea más alta en la clase con menos datos, casi seguro saldrá elegida la clase con más probabilidad a priori, cuando probablemente no debiera ser así.



	ARA PSOE	AST PSOE	AST PSOE	CLM PSOE	GEN PP 2011	GEN PSOE	MAD PSOE	NAV PSOE	PV PNV 2005	PV PP 2005	PV PSOE	PV PSOE	PV PSOE	VAL PP 2011	VAL PSOE
00001	93,87	85,42	81,92	98,32	94,26	91,82	90,98	85,32	93,94	94,03	87,57	85,22	87,67	87,29	87,37
00000	94,03	85,79	82,2	98,32	94,68	91,96	91,29	85,68	94,28	94,32	87,9	85,61	87,98	87,64	87,59

Figura 9. Comparativa con/sin probabilidad a priori

5.3.6 Por partido político

Dado que clasificar directamente qué dice cada frase puede no ser la mejor idea, se ha pensado que es preciso incorporar un paso anterior. Este paso consiste en realizar una pre-clasificación sobre quién ha dicho esa frase, y una vez conocido este dato, cambiar la base del entrenamiento por sólo el grupo específico que lo ha dicho, y entonces clasificar en referencia a estos nuevos datos.

Todo este planteamiento viene justificado porque, por norma general y hablando sobre programas electorales dentro de los mismos grupos, éstos tratan todos los temas con el mismo enfoque, y no existe diversidad de opiniones para las mismas ideas.

El crear esta primera fase de clasificación ha obligado a utilizar sólo a los dos partidos mayoritarios en nuestro país (PP/PSOE), dado que la información de la que se disponía era insuficiente para que las muestras estuvieran compensadas y tener, al menos, la misma cantidad de documentos de cada partido.

5.3.7 Sin filtro

La prueba con el archivo que incluye la totalidad de los textos que se han probado anteriormente no ha funcionado, a causa de que el texto estaba totalmente desbalanceado: casi el 85% de las frases pertenecían a la misma clase y se clasificaban como si fueran de esa clase, con lo que se obtenían resultados cercanos a este porcentaje.

Por tanto, se ha optado por crear un nuevo texto largo en el que las clases estuvieran más balanceadas y poder conseguir una mejor clasificación. El total de resultados se muestran en el apéndice.

5.4 Tiempos de ejecución

Aunque carece de sentido comparar los tiempos de ejecución entre la codificación manual y la automática, se detallan los datos de ciertas ejecuciones realizadas utilizando toda la casuística presentada.

Se muestran todos los tiempos referentes al proceso completo de test, en el que se hace el entrenamiento del texto y después se comprueba el mismo texto con los datos obtenidos, así que no es sólo la clasificación, sino que es la suma de ambos.

El total de los tiempos de ejecución se muestra en el apéndice.

tiempo (s)	00000	00001	00010	00011	00100	00101	00110	00111	01000	01001	01010	01011	01100	01101	01110	01111
ARA PSOE 2011	129,13	129,10	128,92	128,92	129,08	129,11	128,92	128,92	71,45	71,48	71,32	71,33	71,46	71,50	71,32	71,33
AST PSOE 2007	420,44	420,52	420,08	420,09	420,47	420,56	420,08	420,09	228,25	228,24	227,86	227,88	228,20	228,29	227,87	227,88
AST PSOE 2009	820,69	820,78	820,20	820,23	820,75	820,85	820,21	820,22	447,45	447,54	447,04	447,05	447,47	447,60	447,04	447,05
CLM PSOE 1983	15,75	15,76	15,71	15,72	15,75	15,76	15,71	15,72	8,94	8,95	8,92	8,92	8,95	8,96	8,91	8,92
GEN PP 2011	177,23	177,30	176,98	176,98	177,26	177,33	176,98	176,98	96,00	96,06	95,78	95,79	96,02	96,09	95,78	95,79
GEN PSOE 2011	321,36	321,43	321,05	321,06	321,41	321,48	321,05	321,06	174,60	174,68	174,35	174,35	174,65	174,72	174,35	174,35
MAD PSOE 2011	197,82	197,86	197,61	197,62	197,84	197,89	197,61	197,61	108,85	108,91	108,67	108,67	108,87	108,92	108,67	108,67
NAV PSOE 2011	479,21	479,29	478,82	478,85	479,25	479,35	478,82	478,83	258,64	258,74	258,32	258,34	258,69	258,78	258,32	258,33
PV PNV 2005	54,01	54,04	53,88	53,89	54,02	54,06	53,88	53,89	30,41	30,44	30,29	30,30	30,42	30,46	30,29	30,30
PV PP 2005	216,40	216,45	216,17	216,17	216,43	216,49	216,17	216,18	117,25	117,30	117,05	117,06	117,28	117,33	117,05	117,06
PV PSOE 2005	607,36	607,49	606,87	606,89	607,42	607,54	606,87	606,89	328,44	328,57	328,05	328,08	328,54	328,66	328,06	328,05
PV PSOE 2009	760,75	760,88	760,14	760,17	760,81	760,96	760,15	760,17	416,71	416,85	416,20	416,23	416,78	416,92	416,20	416,23
PV PSOE 2012	738,19	738,30	737,68	737,70	738,25	738,37	737,68	737,70	391,93	392,04	391,50	391,54	392,02	392,13	391,53	391,53
VAL PP 2011	355,86	355,94	355,49	355,50	355,90	355,98	355,49	355,50	193,25	193,34	192,93	192,95	193,29	193,37	192,93	192,95
VAL PSOE S007	451,84	451,94	451,42	451,44	451,89	451,99	451,42	451,44	245,04	245,15	244,67	244,70	245,08	245,18	244,68	244,69

Tabla 14. *Tiempos de ejecución*

tiempo (s)	10000	10001	10010	10011	10100	10101	10110	10111	11000	11001	11010	11011	11100	11101	11110	11111
ARA PSOE 2011	106,27	106.30	106.12	106,13	106,28	106,32	106,11	106,12	58,44	58,48	58,32	58,32	58,46	58,49	58,32	58,32
AST PSOE 2007	333,33	333,39	332,96	332,97	333,35	333,44	332,96	332,97	180,73	180,81	180,43	180,44	180,76	180,85	180,43	180,44
AST PSOE 2009	649,67	649,76	649,18	649,20	649,72	649,82	649,18	649,19	347,36	347,46	346,97	346,98	347,41	347,52	346,97	346,98
CLM PSOE 1983	13,84	13,85	13,81	13,81	13,85	13,85	13,81	13,81	7,84	7,85	7,82	7,82	7,85	7,86	7,82	7,82
GEN PP 2011	141,30	141,36	141,04	141,05	141,33	141,39	141,05	141,05	77,78	77,84	77,55	77,56	77,80	77,87	77,56	77,56
GEN PSOE 2011	259,09	259,17	258,77	258,78	259,13	259,19	258,78	258,80	141,08	141,15	140,81	140,83	141,11	141,18	140,81	140,82
MAD PSOE 2011	158,03	158,08	157,82	157,83	158,06	158,10	157,82	157,83	87,42	87,48	87,24	87,25	87,44	87,49	87,24	87,25
NAV PSOE 2011	376,75	376,84	376,36	376,37	376,79	376,88	376,37	376,37	204,79	204,88	204,46	204,48	204,82	204,92	204,47	204,48
PV PNV 2005	46,28	46,32	46,15	46,16	46,29	46,33	46,15	46,15	27,53	27,56	27,40	27,41	27,54	27,57	27,40	27,41
PV PP 2005	173,26	173,31	173,02	173,04	173,30	173,36	173,04	173,04	93,97	94,02	93,78	93,78	94,00	94,05	93,79	93,80
PV PSOE 2005	480,93	481,06	480,44	480,46	480,99	481,12	480,44	480,45	260,51	260,62	260,09	260,10	260,55	260,68	260,09	260,10
PV PSOE 2009	602,95	603,10	602,35	602,38	603,01	603,16	602,35	602,37	330,68	330,82	330,17	330,19	330,73	330,87	330,17	330,19
PV PSOE 2012	583,08	583,21	582,58	582,60	583,16	583,24	582,56	582,57	314,10	314,22	313,67	313,69	314,16	314,31	313,70	313,71
VAL PP 2011	284,10	284,18	283,74	283,74	284,15	284,24	283,73	283,74	156,21	156,30	155,89	155,91	156,25	156,33	155,89	155,91
VAL PSOE S007	361,42	361,52	361,00	361,03	361,48	361,60	361,01	361,01	197,17	197,27	196,81	196,83	197,20	197,31	196,80	196,83

Tabla 15. *Tiempos de ejecución*

5.5 Visualización en Qlikview

Después de analizar los requerimientos del sistema, y evaluar cuáles son las necesidades que hay que cubrir, se ha estimado crear un cuadro general en el que poder visualizar todos los datos obtenidos tanto el proceso de la clasificación, como los tiempos de ejecución y porcentajes de acierto de cada uno de los textos.

5.5.1 Proceso ETL

Para poder realizar correctamente la carga de los datos obtenidos de la clasificación se ha debido hacer también un estudio y análisis para crear el modelo de datos apropiado con el que conseguir el máximo de información posible y con el que se pueda obtener todo aquello que se desea ver en el Dashboard final.

En el primer paso, referente a la extracción de datos (EXTRACT), se ha elegido cargar toda la información referente al conteo de palabras, así como la matriz de frecuencia, las listas de palabras únicas/stemizadas/sin stop words y también las clases. Dado que el nombre del archivo contiene información sobre el partido político, la comunidad autónoma y el año en que se ha hecho el programa, esta información también está extraída y preparada para modelar.

Para la siguiente fase, transformación, se enlazan todos los datos extraídos anteriormente en tablas sin relación para convertirlo en un modelo de datos relacionado y así aprovechar las ventajas de Qlikview con el uso bases de datos relacionales.

El modelo obtenido se muestra en la imagen siguiente:

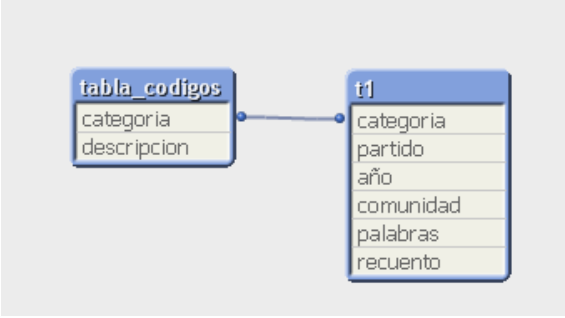


Figura 10*. Modelo de tablas*

Donde sólo encontramos 2 tablas, una con la información referente al recuento de las palabras, y otra con relación a las categorías, ambas unidas por el campo categoría.

En el último paso, la carga o LOAD, se cargan los datos extraídos y calculados para mostrarlos en un Dashboard en el que se pueden seleccionar diferentes opciones y filtros con

tal de "navegar" por los datos y así poder entender y sacar conclusiones a partir de los gráficos obtenidos y predefinidos anteriormente.

Estos filtros son selectores que permiten gracias a su representación por colores (verde, blanco o gris), conocer si la selección actual contiene valores disponibles o que están fuera del grupo.

5.5.2 Dashboard

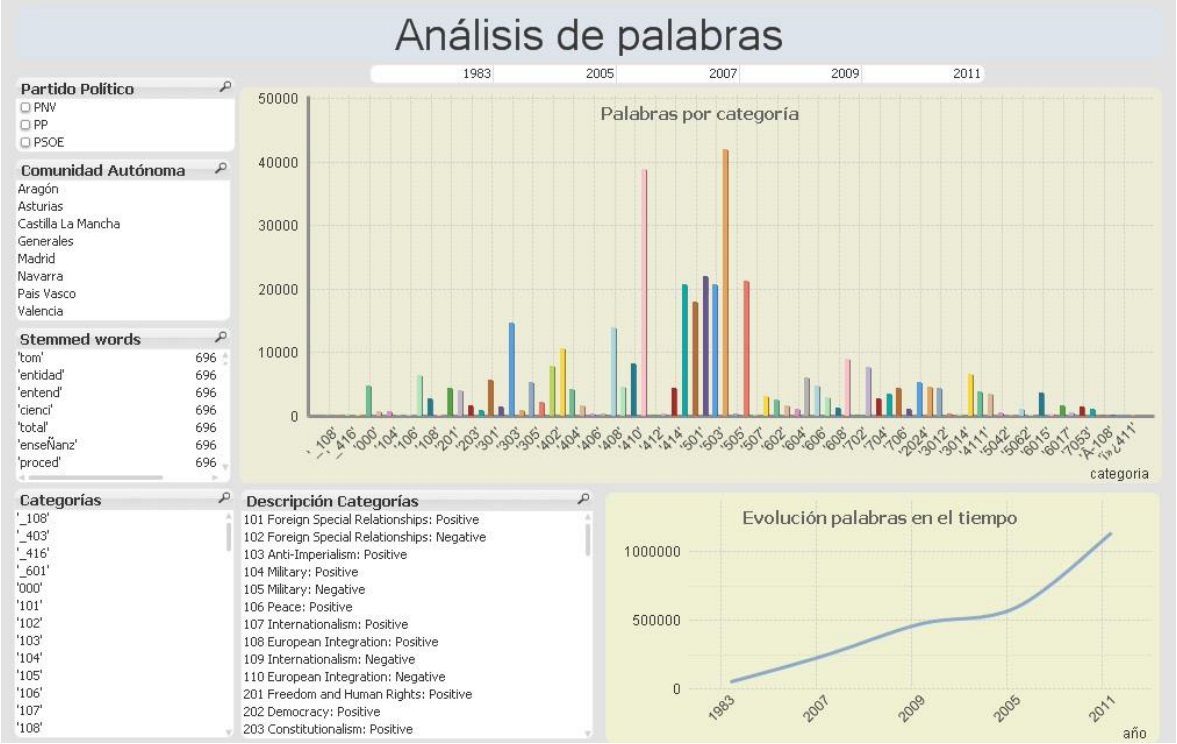


Figura 11. *Dashboard Análisis de palabras*

En el Dashboard que se muestra en la figura 6, se expone un ejemplo para el estudio de las palabras utilizadas en los diferentes archivos usados en la clasificación. Se tienen selectores para:

- Fecha: Se puede elegir el año del que se quiere ver la información
- Partido Político: Se elige el o los partidos políticos de los que se quiere observar sus datos
- Comunidad autónoma: Selector entre las diferentes comunidades autónomas
- Stemmed words: Lista de palabras en referencia a los documentos seleccionados por años, partidos políticos y comunidades autónomas.
- Categorías: Con este selector se eligen aquellas categorías a las que se quiere prestar atención y estudiar con detalle.
- Descripción Categorías: El nombre de las categorías, para mejorar la comprensión.

Se muestran dos gráficas: en la primera “Palabras por categoría” se muestra el conteo de las palabras diferentes que se han dicho para cada categoría. En la siguiente gráfica “Evolución palabras en el tiempo” se muestra la cantidad de palabras ordenadas a lo largo de los años.

Utilizando todos los filtros anteriores se pueden comprobar todos los datos en referencia a las palabras usadas tanto en el tiempo como por cada categoría, conociendo la descripción de la misma, así como de las fechas, partidos políticos o comunidades autónomas donde se han utilizado.

6 Planificación

El proyecto estaba planteado, en un principio, para que a partir de la explotación de unos datos ya clasificados (a mano o automáticamente) ofrecer una herramienta de BI que permitiera la visualización y utilización de la base de datos, además de una rápida y fácil obtención de los mismos. No obstante, se ha tenido en cuenta la opción de que estas clasificaciones no se hicieran manualmente, ya que el coste temporal de clasificar tal cantidad de información hace inviable tal opción, ya que la actualización de estas bases de datos ha de ser realizada por gente especializada en estos temas y han de ir registro a registro leyendo, comprendiendo y clasificando cada uno de ellos.

Así pues, los primeros meses del proyecto se han dedicado al estudio bibliográfico de la materia, que permitiera contextualizar y conocer otras investigaciones que se hubieran desarrollado en proyectos similares. La segunda parte de este trabajo se ha dedicado a poner en práctica, ensayar y mejorar los conocimientos aplicados.

6.1 Diagrama de Gantt

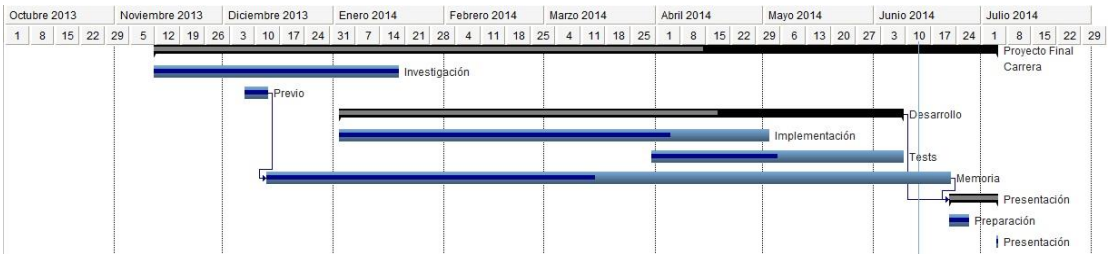


Figura 12. *Diagrama de Gantt*

6.2 Distribución del tiempo

Como se puede observar en el anterior diagrama de Gantt, la mayor parte del tiempo de la realización del proyecto ha sido de implementación o desarrollo del programa y redacción de la memoria. Del total de 450 horas destinadas al proyecto, 300 horas se han destinado a estas tareas (150 y 150 horas, respectivamente). Como parte de desarrollo se debe tener en cuenta también el tiempo de pruebas (50 horas).

Contando el total de horas a una media de 8€/hora, el presupuesto mínimo (sin tener en cuenta el ordenador utilizado y el software) suma 3600€.

El tiempo previo a todo el desarrollo de implementación y memoria se ha destinado a investigación y análisis de métodos y herramientas para llevar a cabo todo el desarrollo. Este tiempo ha sido de 100 horas más.

A todo este trabajo se ha de sumar el tiempo invertido en la realización del previo y todos los tiempos de espera de ejecuciones, ya que este tiempo no se puede medir ni como test ni como implementación, porque en la mayor parte de los casos se ha procurado ejecutar durante la noche para afectar lo menos posible.

6.3 Problemas

Es muy difícil planificar correctamente un proyecto como este y es muy importante tener claro qué hay que hacer y seguir una metodología clara con el fin de no llegar apurado de tiempo y sin haber pensado en todos los pasos a desarrollar.

A nivel técnico los principales problemas han surgido al intentar entender los datos mostrados, ya que en un inicio parecían no tener mucho sentido, pero después de analizar más profundamente se han podido detectar los patrones que afectan a la clasificación.

Ha habido muchos problemas con el formato de los datos, ya que al trabajar con diferentes sistemas operativos (Windows/Linux/OS) existía incompatibilidad y muchas de las palabras acentuadas o de cualquier otra índole se han visto afectadas.

7 Conclusiones

Las conclusiones se separarán en dos grupos, el primero, con un único apartado referente a las conclusiones de interés general, y un segundo con dos apartados, sobre las técnicas o de implementación, donde se desglosará entre las conclusiones referentes a la clasificación automática y las del visionado de datos en la herramienta de *BI* elegida.

7.1 Conclusiones generales

Trabajar analizando textos periodísticos o políticos requiere una gran inversión de tiempo y esfuerzos para clasificar y ordenar toda la información. Más allá de la clasificación también existen procesos repetitivos en el momento de analizar los datos los cuales requieren también una inversión importante de recursos.

Se ha proporcionado un conjunto de herramientas que permiten agilizar estos procesos para poder llevar a cabo el análisis y obtener los datos de manera automática.

La principal conclusión que se extrae es que el tiempo invertido en la clasificación automática, convierte un hasta ahora problema en un simple trámite. Por tanto, el provecho que se puede obtener de los datos es prácticamente instantáneo y se debe considerar como la principal ventaja.

7.2 Conclusiones técnicas

Disponer de un cuadro de mando preparado para recibir datos formateados, permite estandarizar las consultas a esos datos, y por tanto, conocer qué se va a ver y cómo se va a ver. Simplemente al cargar nuevos datos procesados se dispone de información de calidad que ya está procesada para analizar con eficacia.

7.2.1 Análisis de la clasificación

Debemos preguntarnos, ¿cuál es el error que comete una persona experta cuando clasifica manualmente este tipo de datos? En base a la respuesta a esta pregunta, podemos concluir cómo funciona este método.

Como la respuesta a esta pregunta es subjetiva, debemos basarnos en un porcentaje relativo. Por tanto, el objetivo marcado ha de ser el 100% e intentar acercarse a este valor tanto como sea posible.

Los resultados obtenidos no distan mucho de este valor, alcanzando índices de hasta el 99.2% de acierto en su clasificación, en el caso de CLM_PSOE_1983. Aunque dependiendo del texto y los parámetros usados puede variar mucho este resultado.

Cuando se calcula el acierto uniendo todos los textos (conjunto inicial de 15 textos) no acaba de dar un buen resultado causado porque la distribución de las palabras con respecto a los partidos políticos está completamente desbalanceada, con lo que es necesario disponer de más datos codificados de manera más uniforme entre los diferentes partidos políticos, ya que así se obtendrán mejores clasificaciones.

Los textos utilizados hablan sobre temáticas diferentes y siempre existe una "inclinación" a hablar más de unos temas que de otros. Esto afecta en que las ocurrencias de las palabras no están balanceadas y hace que la probabilidad *a priori* de una clase sobre el resto tenga un efecto que desvirtúe el resultado del cálculo de probabilidades. Esto está causado porque hay muchas palabras que se usan tanto en un contexto como en otro diferente, y podemos encontrar las mismas palabras en diferentes categorías pero si de una categoría tenemos el doble de palabras (las mismas) que en otra categoría, es mucho más probable que salga elegida la categoría con más ocurrencias.

7.2.2 Análisis de los datos (Qlikview)

El Dashboard o Cuadro de mando final, dispone de una serie de selectores que nos permiten filtrar la información a partir de las variables dentro de cada dimensión, así pues, podemos definir qué queremos ver:

- Por año: con el uso de esta selección, toda la información mostrada es filtrada por el año en que se publicó el programa electoral.
- Por partido político: donde elegir filtrar los datos por los partidos que sean de interés
- Por categoría: eligiendo de esta manera las categorías a analizar
- Por comunidad autónoma: permitiendo elegir las zonas en las que se han publicado los programas electorales

Se muestra asimismo una lista con las palabras utilizadas, que varía dependiendo de la selección de los anteriores selectores, mostrando sólo aquellas palabras utilizadas en el contexto que se haya elegido a través de ellos.

7.3 Mejoras

Después de realizar el proyecto se han detectado aspectos en el programa a mejorar y ampliar en próximas versiones.

Como objetivo principal se debe intentar mejorar el porcentaje de acierto obtenido sin depender del texto con el que se haya entrenado, y para ello van a hacer falta muchos más datos de muestra para entrenar y que éstos estén más balanceados (tanto a nivel de palabras como a nivel del partido político al que pertenezcan).

Sería muy interesante hacer que el programa fuera capaz de detectar el idioma en el que está escrito y pudiera separar estos datos para testear con otros juegos de pruebas. En el caso del multi idioma se puede considerar a traducir las palabras al idioma base y clasificar a partir de estas nuevas palabras, o considerarlo como dos grupos diferentes.

Dado que el programa del stemmer no funciona para todos los casos y devuelve palabras que pueden seguir siendo variaciones, se debe buscar una alternativa, ya sea crear un diccionario con todos los términos posibles o mejorar el algoritmo de stemming.

Con ánimo de filtrar mejor las palabras “stop words” se deben añadir nuevas palabras que no se hayan considerado hasta ahora y que puedan afectar a la clasificación, y puede que eliminar a alguna de ellas que no afecte positivamente.

Aunque carece de sentido preocuparse por el tiempo de respuesta del algoritmo, es una cuestión a tener en cuenta para una nueva versión, e intentar que el tiempo de respuesta del programa sea más rápido.

En cuanto a la visualización con Qlikview, se debe realizar un estudio más preciso que las necesidades y analizar mejor los datos a mostrar, ya que hasta ahora solo se muestra una pequeña parte de todo lo que se puede llegar a visualizar.

8 Bibliografía y Webgrafía

Budge, I.; Klingemann, H.D.; Volkens, A.; Bara, J.; Tanembaum, E. (2001): Mapping policy preferences: estimates for parties, electors, and governments, 1945:1998. Oxford: Oxford University Press, cap. 1 [pp. 19250.]

Budge, I.; Klingemann, H.D.; Volkens, A.; Bara, J.; Tanembaum, E. (2001): Mapping policy preferences: estimates for parties, electors, and governments, 1945: 1998. Oxford: Oxford University Press, cap. 2 [pp. 51273.]

Klingemann et al 2006 mapping policy preferences part2

Sonia Alonso Sáenz de Oger, Braulio Gómez Fortes. Partidos nacionales en elecciones regionales: ¿coherencia territorial o programas a la carta? Revista de Estudios Políticos (nueva época), Núm. 152, Madrid, (abril-junio 2011), págs. 183-209

Treatingwords as data with error: uncertainty in text statements of policy positions, Kenneth Benoit, Michael Laver, SlavaMikhaylov, American Journal of Political Science, Vol. 53, No. 2, (April 2009), Pp. 495–513

Mapping Policy Preferences II. Estimates for Parties, Electors, and Governments in Eastern Europe, European Union, and OECD 1990-2003. Hans-Dieter Klingermann, Andrea Volkens, Judith L Bara, Ian Budge, and Michael D. McDonald. (2006)

Estimating Policy Positions from Political Texts. Michael Laver and John Garry. American Journal of Political Science, Vol 44, No. 3 (Jul., 2000), pp. 619-634.

The Mandate Process. McDonald Budge 2005.

Sonia Alonso, Andrea Volkens, Braulio Gómez. Análisis de contenido de textos políticos. Un enfoque cuantitativo. CIS Centro de investigaciones sociológicas, (2012).

Cortes C., Vapnik V., (1995). Support-Vector networks, Machine learning, Vol 20 / 3: 273-297

Jordi Vitrià. Intel·ligència Artificial II.Apunts de l'assignatura apunts (2001).

Juan de Dios Álvarez Romero. Clasificación Automática de Textos usando Reducción de Clases basada en Prototipos. (Enero 2009).

Silvia Cristina Machado Ferreira, Aplicació basada en tècniques de Natural Language Programming per a la detecció de sentiments a la web (Septiembre 2013).

Joachins, Thorsten. Text categorization with support vector machines: Learning with many relevant features.

Manifesto project Database [Online] **https://manifesto-project.wzb.eu/**

Spanish Policy Agendas [Online] **http://www.ub.edu/spanishpolicyagendas/**

Ayuda de matlab [Online] **http://www.mathworks.es/es/help/index.html**

Ayuda en la red [Online] **http://stackoverflow.com/**

Web de Snowball, lenguaje de procesado de cadenas de caracteres, para crear algoritmos de stemming. [Online] **http://snowball.tartarus.org/**

Computational intelligence and knowledge. Poole, Mackworth& Goebel (1998) [Online] **http://people.cs.ubc.ca/~poole/ci/ch1.pdf**

Coneixement, Raonament i Incertesa. Aprendre amb Bayes II - Machine Learning 10-70. Tom M. Mitchell. Machine Learning Department. Carnegie Mellon University January 18, 2011 [Online] **http://www.cs.cmu.edu/~tom/10601_fall2012/slides/MLE_MAP_9-11-12.pdf**

Teorema de Bayes [Online] **http://en.wikipedia.org/wiki/Bayes%27_theorem**

Teoria de Support Vector Machines (SVM) [Online] **http://en.wikipedia.org/wiki/Support_vector_machine**

Teoría de K-Nearest Neighbors (KNN) [Online] **http://en.wikipedia.org/wiki/KNN**

Business Intelligence [Online] **http://www.sinnexus.com/business_intelligence/**

Inteligencia Empresarial [Online] **http://es.wikipedia.org/wiki/Inteligencia_empresarial**

Josep Lluís Cano, Business Intelligence: Competir con información [Online] **http://itemsweb.esade.edu/biblioteca/archivo/Business_Intelligence_competir_con_informacion.pdf**

David M. Blei, Probabilistic Topic Models **http://www.cs.princeton.edu/~blei/papers/Blei2012.pdf**

Stanford Artificial Intelligence laboratory [Online] **http://ai.stanford.edu/**

9 Anexo

9.1 RMP regional manifesto project

El Manifesto Project ha desarrollado un sistema de categorías donde cada cuasi-frase de cada manifesto sólo se clasifica en una única de las 56 categorías estándar. Estas 56 categorías están agrupadas en siete áreas políticas y han sido diseñadas para ser comparables entre partidos, países, elecciones y a lo largo del tiempo.

Domain 1: External Relations
101 Foreign Special Relationships: Positive
102 Foreign Special Relationships: Negative
103 Anti-Imperialism: Positive
104 Military: Positive
105 Military: Negative
106 Peace: Positive
107 Internationalism: Positive
108 European Integration: Positive
109 Internationalism: Negative
110 European Integration: Negative

Domain 2: Freedom and Democracy
201 Freedom and Human Rights: Positive
202 Democracy: Positive
203 Constitutionalism: Positive
204 Constitutionalism: Negative

Domain 3: Political System
301 Decentralisation: Positive
302 Centralisation: Positive
303 Governmental and Administrative Efficiency: Positive
304 Political Corruption: Negative
305 Political Authority: Positive

Domain 4: Economy
401 Free Enterprise: Positive
402 Incentives: Positive
403 Market Regulation: Positive
404 Economic Planning: Positive
405 Corporatism: Positive
406 Protectionism: Positive
407 Protectionism: Negative
408 Economic Goals
409 Keynesian Demand Management: Positive
410 Productivity: Positive 8
411 Technology and Infrastructure: Positive
412 Controlled Economy: Positive
413 Nationalisation: Positive
414 Economic Orthodoxy: Positive
415 Marxist Analysis: Positive
416 Anti-Growth Economy: Positive

Domain 5: Welfare and Quality of Life
501 Environmental Protection: Positive
502 Culture: Positive
503 Social Justice: Positive
504 Welfare State Expansion
505 Welfare State Limitation
506 Education Expansion
507 Education Limitation

Domain 6: Fabric of Society
601 National Way of Life: Positive
602 National Way of Life: Negative
603 Traditional Morality: Positive
604 Traditional Morality: Negative
605 Law and Order: Positive
606 Civic Mindedness: Positive
607 Multiculturalism: Positive
608 Multiculturalism: Negative

Domain 7: Social Groups
701 Labour Groups: Positive
702 Labour Groups: Negative
703 Agriculture: Positive
704 Middle Class and Professional Groups: Positive
705 Minority Groups: Positive
706 Non-Economic Demographic Groups: Positive

Territorial authority claims in party manifestos refer to the level of administration that is addressed by the policy/politics/polity preference (i.e., local, regional, national, European, and international) or, alternatively, to the relationships between the levels (cooperation, subsidiary principle, exclusivity of competences, etc.). Territorial authority claims do not necessarily reflect the real existing distribution of competences between the levels of government in the manifesto country at the time of the election. They only reflect the party's view about which level of administration is connected to which particular policy preference.

A territorial authority claim is captured by a code made up of two digits: the first digit indicates the level of government for which the policy preference is articulated and the second digit signals the preferred degree of authority for that level (i.e., more/less competences to be devolved/returned to the level of government addressed in the first digit). When the sentence does not contain a claim for more or for fewer competences for a particular level of government, the second digit is 0. This implies that the sentence is simply connecting a particular policy preference to a particular level of government (for example, the sentence says what the regional government is going to do to help single-parent families), accepting the territorial status quo (i.e. the existing distribution of competences between the levels of government).

Level of government (first digit):
1 The local level
2 The regional (provincial, state) level
3 The national level
8 The European level
9 The international/global level

Preferred degree of authority (second digit):
1 The text unit (i.e. the quasi-sentence) claims less authority for the respective level
2 The text unit (i.e. the quasi-sentence) claims more authority for the respective level
0 The text unit (i.e. the quasi-sentence) contains no authority claim. It only states the level of government addressed by the policy preference, without claiming more or less competences for that particular level of government in that policy area.
Alternatively, when more than one level of government is addressed simultaneously and the relationship between the levels is made explicit, one of the following codes for territorial authority claims applies:

01 In favour of subsidiary principle
02 In favour of clear (jurisdictional) distinction between levels (accountability)
03 In favour of shared authority between some levels, including explicit calls for cooperation or coordination between higher and lower levels (vertical cooperation)
09 More than one level addressed at the same time (including all levels addressed at the same

time: for example, a statement defending justice at all levels of governance) the classification scheme of territorial authority claims is composed of 20 different codes:

10 Local level (No explicit claim for more or less authority)
11 Less authority for the local level
12 More authority for the local level
20 Regional level (No explicit claim for more or less authority)
21 Less authority for the regional level
22 More authority for the regional level
30 National level (No explicit claim for more or less authority)
31 Less authority for the national level
32 More authority for the national level
80 European level (No explicit claim for more or less authority)
81 Less authority for the European level
82 More authority for the European level
90 International level (No explicit claim for more or less authority)
91 Less authority for the international level
92 More authority for the international level
00 No territorial authority claim is being made (no level addressed, no direction)
01 In favour of subsidiary principle
02 In favour of clear (jurisdictional) distinction between levels (accountability)
03 In favour of shared authority between some levels, including explicit calls for cooperation or coordination between higher and lower levels (vertical cooperation)
09 More than one level addressed at the same time; all levels addressed at the same time

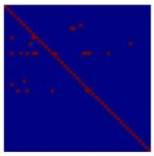
9.2 Resultados

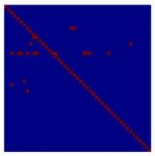
9.2.1 Clasificación

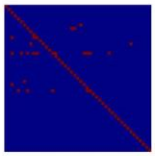
A continuación se muestran las matrices de confusión obtenidas en cada una de las pruebas realizadas en todos los archivos analizados. En ellas se puede comprobar cómo han clasificado todas las configuraciones de una manera más visual.

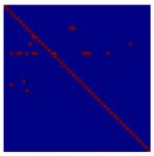
En el CD adjunto se entregan los Workspace creados para cada ejecución para comprobar los valores de todas las variables, así como de todas las tablas y listas de palabras.

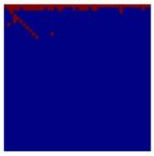
ARA PSOE 2011

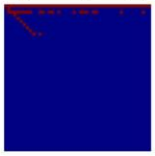

1_ARA_PSOE_2011_text.txt

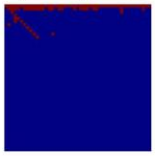

2_ARA_PSOE_2011_text.txt

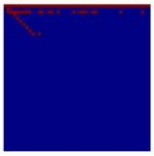

3_ARA_PSOE_2011_text.txt

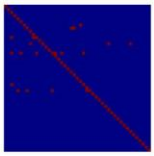

4_ARA_PSOE_2011_text.txt

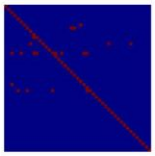

5_ARA_PSOE_2011_text.txt

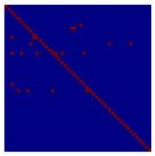

6_ARA_PSOE_2011_text.txt

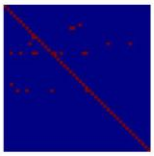

7_ARA_PSOE_2011_text.txt

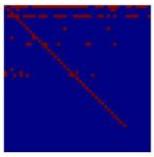

8_ARA_PSOE_2011_text.txt

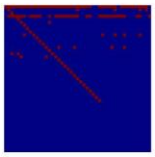

9_ARA_PSOE_2011_text.txt

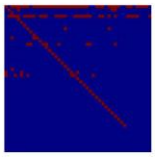

10_ARA_PSOE_2011_text.txt

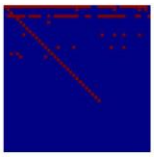

11_ARA_PSOE_2011_text.txt

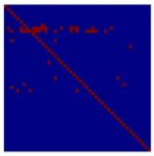

12_ARA_PSOE_2011_text.txt

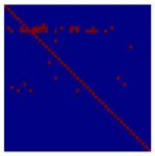

13_ARA_PSOE_2011_text.txt

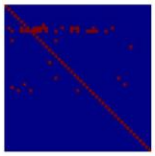

14_ARA_PSOE_2011_text.txt

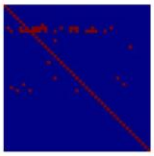

15_ARA_PSOE_2011_text.txt

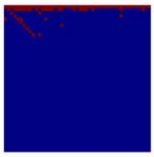

16_ARA_PSOE_2011_text.txt


17_ARA_PSOE_2011_text.txt

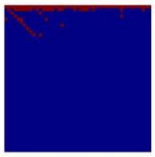

18_ARA_PSOE_2011_text.txt

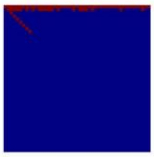

19_ARA_PSOE_2011_text.txt

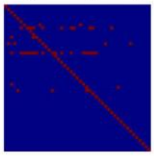

20_ARA_PSOE_2011_text.txt

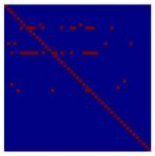

21_ARA_PSOE_2011_text.txt

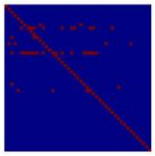

22_ARA_PSOE_2011_text.txt

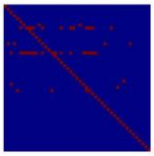

23_ARA_PSOE_2011_text.txt

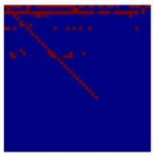

24_ARA_PSOE_2011_text.txt

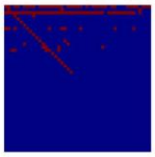

25_ARA_PSOE_2011_text.txt

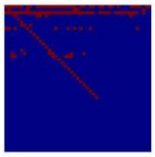

26_ARA_PSOE_2011_text.txt

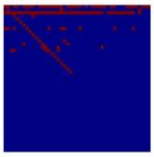

27_ARA_PSOE_2011_text.txt

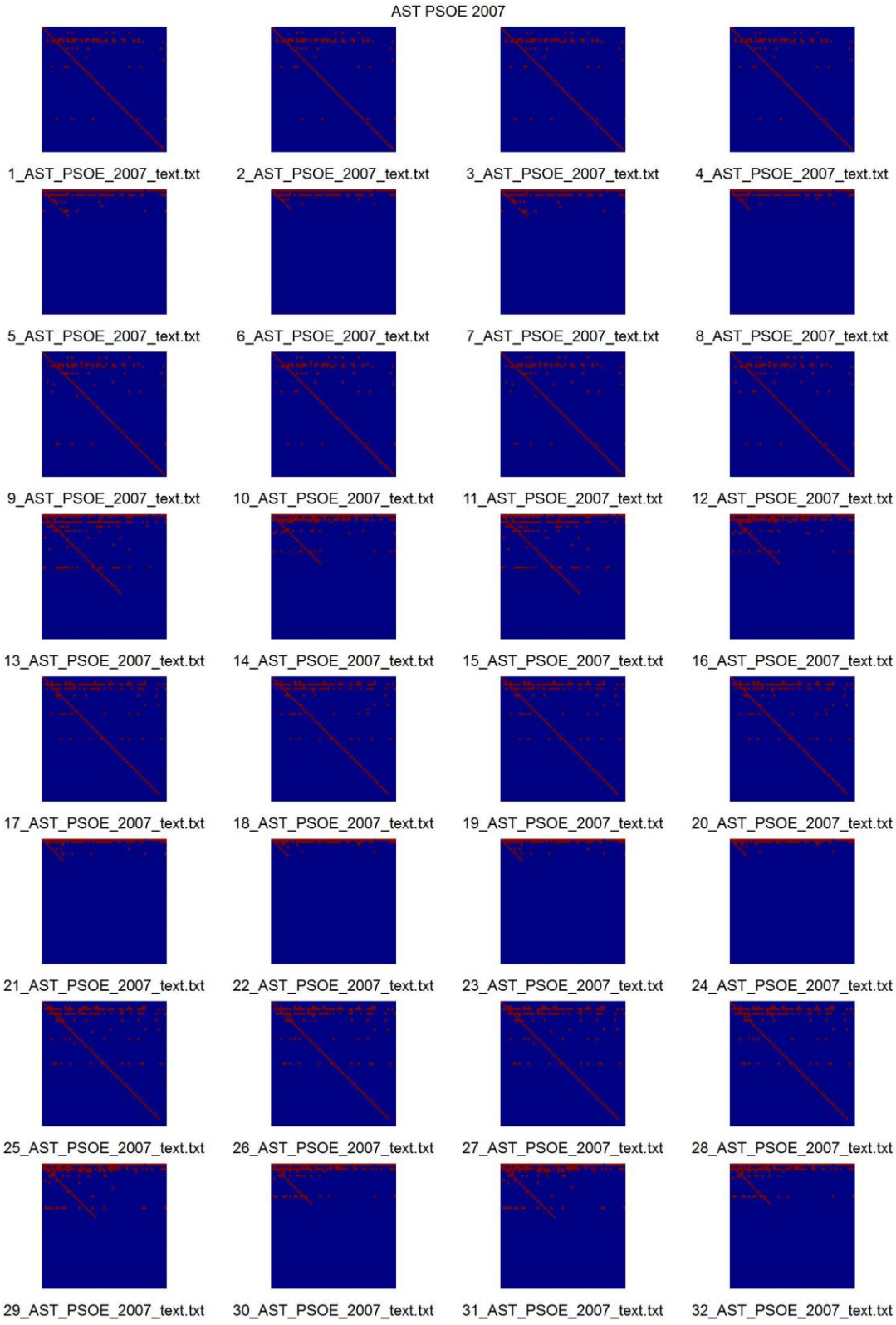

28_ARA_PSOE_2011_text.txt


29_ARA_PSOE_2011_text.txt

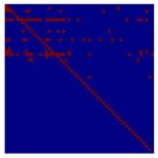

30_ARA_PSOE_2011_text.txt

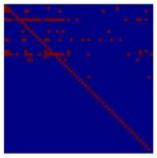

31_ARA_PSOE_2011_text.txt

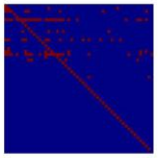

32_ARA_PSOE_2011_text.txt

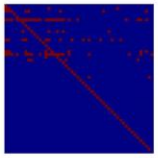

AST PSOE 2007
1_AST_PSOE_2007_text.txt
2_AST_PSOE_2007_text.txt
3_AST_PSOE_2007_text.txt
4_AST_PSOE_2007_text.txt
5_AST_PSOE_2007_text.txt
6_AST_PSOE_2007_text.txt
7_AST_PSOE_2007_text.txt
8_AST_PSOE_2007_text.txt
9_AST_PSOE_2007_text.txt
10_AST_PSOE_2007_text.txt
11_AST_PSOE_2007_text.txt
12_AST_PSOE_2007_text.txt
13_AST_PSOE_2007_text.txt
14_AST_PSOE_2007_text.txt
15_AST_PSOE_2007_text.txt
16_AST_PSOE_2007_text.txt
17_AST_PSOE_2007_text.txt
18_AST_PSOE_2007_text.txt
19_AST_PSOE_2007_text.txt
20_AST_PSOE_2007_text.txt
21_AST_PSOE_2007_text.txt
22_AST_PSOE_2007_text.txt
23_AST_PSOE_2007_text.txt
24_AST_PSOE_2007_text.txt
25_AST_PSOE_2007_text.txt
26_AST_PSOE_2007_text.txt
27_AST_PSOE_2007_text.txt
28_AST_PSOE_2007_text.txt
29_AST_PSOE_2007_text.txt
30_AST_PSOE_2007_text.txt
31_AST_PSOE_2007_text.txt
32_AST_PSOE_2007_text.txt

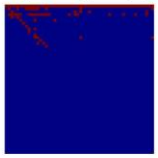
AST PSOE 2009

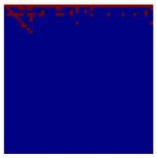

1_AST_PSOE_2009_text.txt

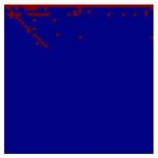

2_AST_PSOE_2009_text.txt

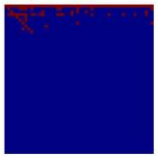

3_AST_PSOE_2009_text.txt

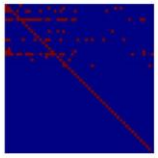

4_AST_PSOE_2009_text.txt

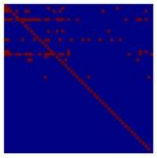

5_AST_PSOE_2009_text.txt

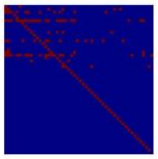

6_AST_PSOE_2009_text.txt

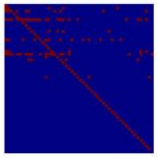

7_AST_PSOE_2009_text.txt

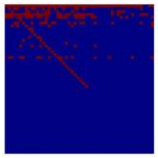

8_AST_PSOE_2009_text.txt

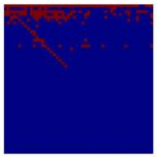

9_AST_PSOE_2009_text.txt

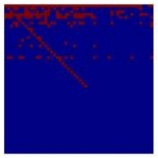

10_AST_PSOE_2009_text.txt

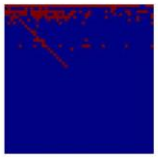

11_AST_PSOE_2009_text.txt

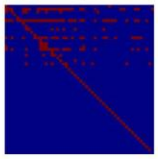

12_AST_PSOE_2009_text.txt

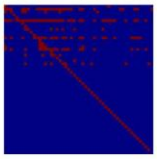

13_AST_PSOE_2009_text.txt

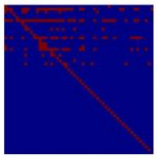

14_AST_PSOE_2009_text.txt

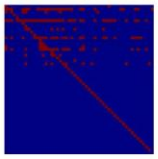

15_AST_PSOE_2009_text.txt

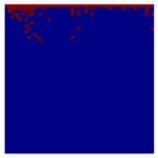

16_AST_PSOE_2009_text.txt

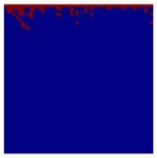

17_AST_PSOE_2009_text.txt

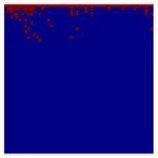

18_AST_PSOE_2009_text.txt

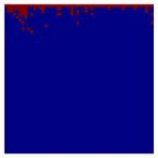

19_AST_PSOE_2009_text.txt

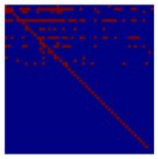

20_AST_PSOE_2009_text.txt

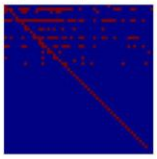

21_AST_PSOE_2009_text.txt

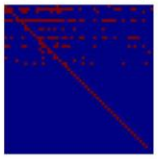

22_AST_PSOE_2009_text.txt

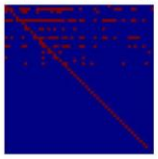

23_AST_PSOE_2009_text.txt

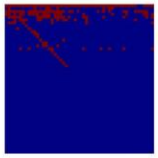

24_AST_PSOE_2009_text.txt

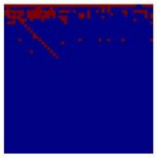

25_AST_PSOE_2009_text.txt

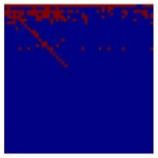

26_AST_PSOE_2009_text.txt

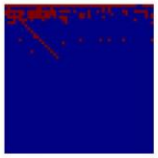

27_AST_PSOE_2009_text.txt


28_AST_PSOE_2009_text.txt

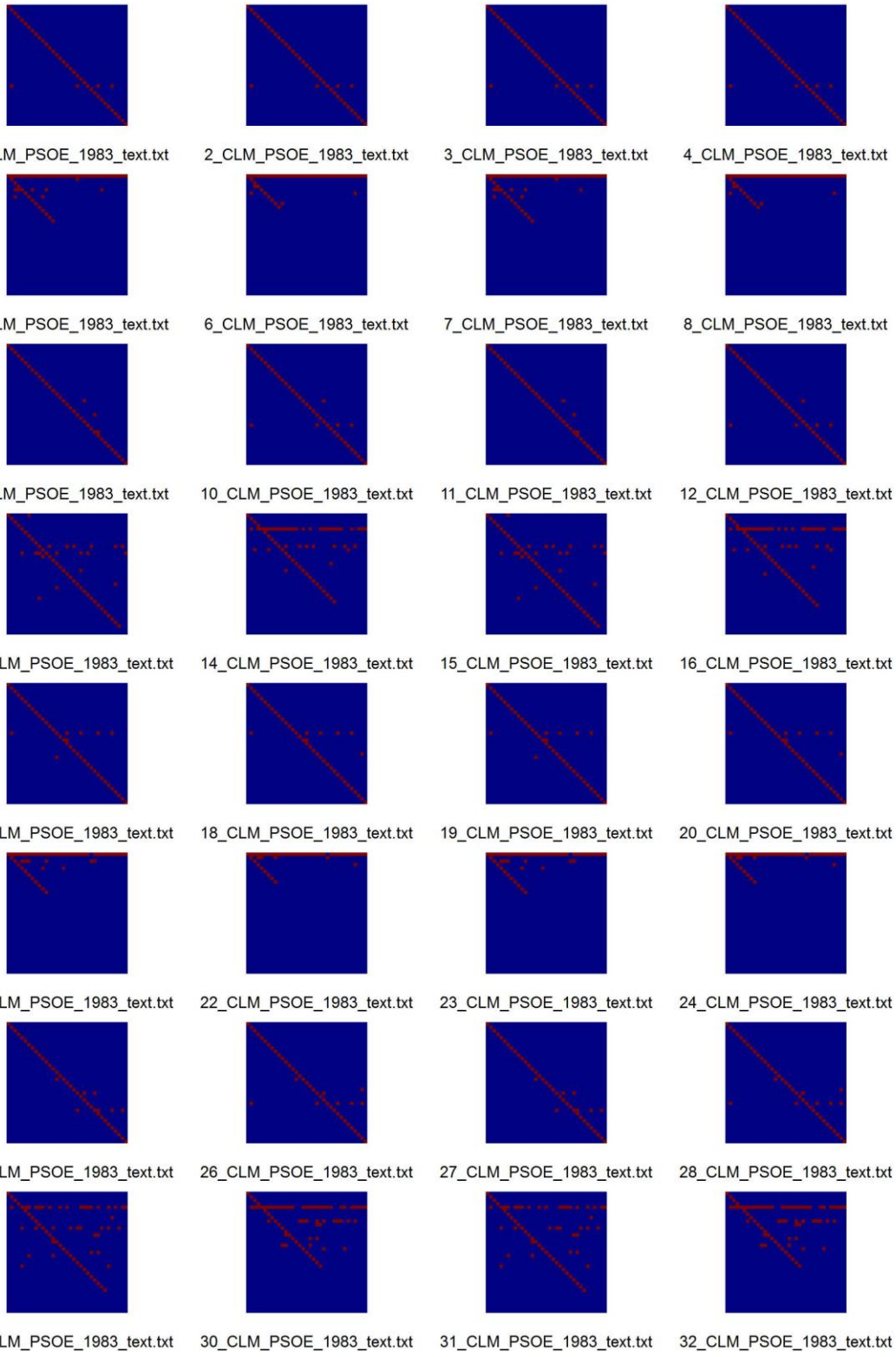

29_AST_PSOE_2009_text.txt


30_AST_PSOE_2009_text.txt

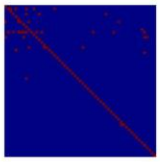

31_AST_PSOE_2009_text.txt

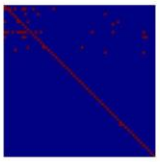

32_AST_PSOE_2009_text.txt

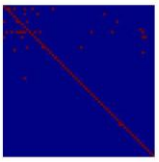
CLM PSOE 1983

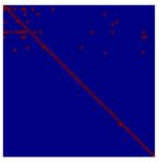


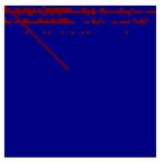
GEN PP 2011

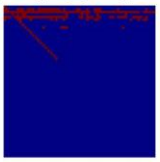

1_GEN_PP_2011_text.txt

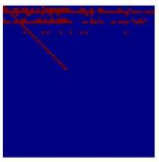

2_GEN_PP_2011_text.txt

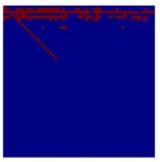

3_GEN_PP_2011_text.txt

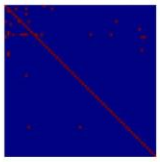

4_GEN_PP_2011_text.txt

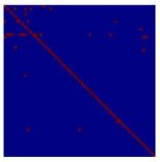

5_GEN_PP_2011_text.txt

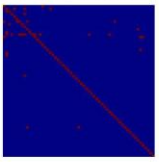

6_GEN_PP_2011_text.txt

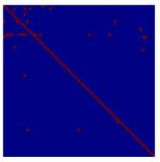

7_GEN_PP_2011_text.txt

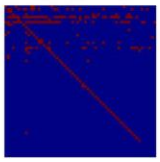

8_GEN_PP_2011_text.txt

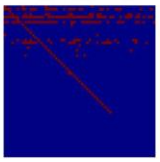

9_GEN_PP_2011_text.txt

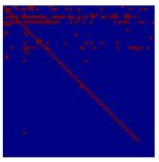

10_GEN_PP_2011_text.txt

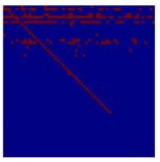

11_GEN_PP_2011_text.txt

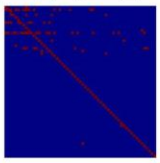

12_GEN_PP_2011_text.txt

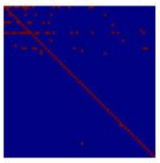

13_GEN_PP_2011_text.txt

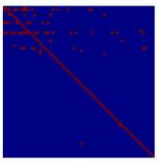

14_GEN_PP_2011_text.txt

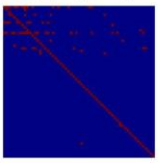

15_GEN_PP_2011_text.txt

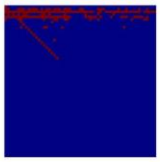

16_GEN_PP_2011_text.txt

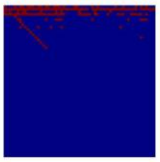

17_GEN_PP_2011_text.txt

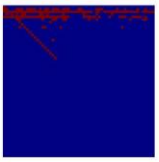

18_GEN_PP_2011_text.txt

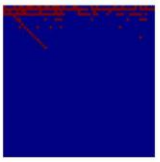

19_GEN_PP_2011_text.txt

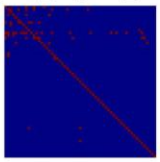

20_GEN_PP_2011_text.txt

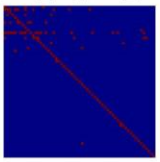

21_GEN_PP_2011_text.txt

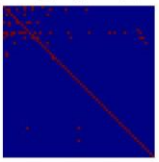

22_GEN_PP_2011_text.txt

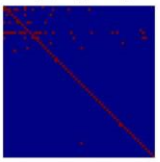

23_GEN_PP_2011_text.txt

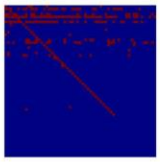

24_GEN_PP_2011_text.txt

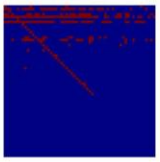

25_GEN_PP_2011_text.txt

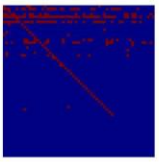

26_GEN_PP_2011_text.txt

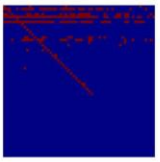

27_GEN_PP_2011_text.txt

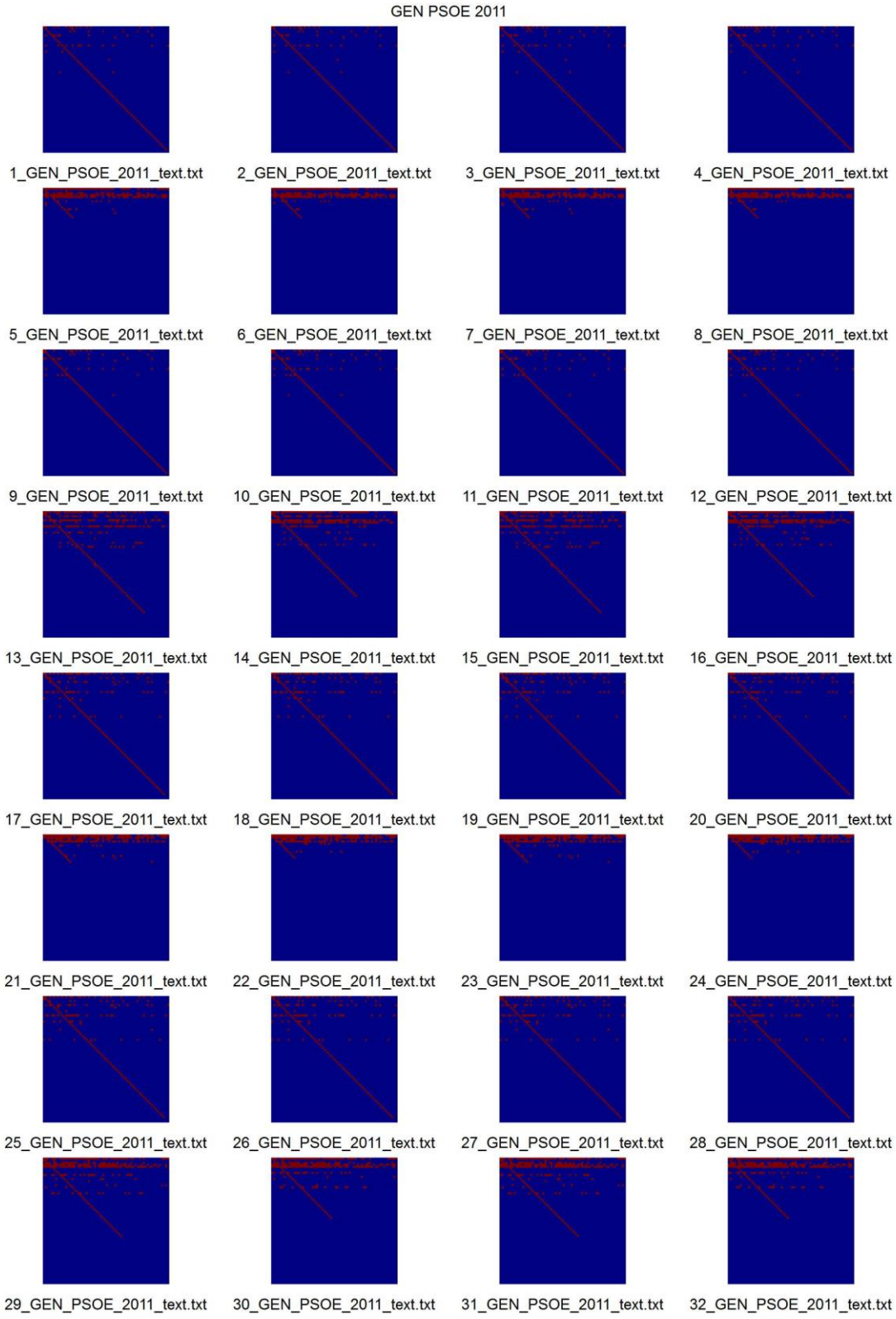

28_GEN_PP_2011_text.txt


29_GEN_PP_2011_text.txt

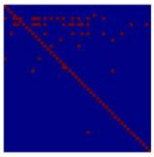

30_GEN_PP_2011_text.txt

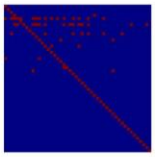

31_GEN_PP_2011_text.txt

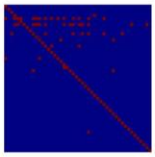

32_GEN_PP_2011_text.txt

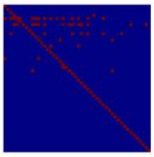

GEN PSOE 2011
1_GEN_PSOE_2011_text.txt
2_GEN_PSOE_2011_text.txt
3_GEN_PSOE_2011_text.txt
4_GEN_PSOE_2011_text.txt
5_GEN_PSOE_2011_text.txt
6_GEN_PSOE_2011_text.txt
7_GEN_PSOE_2011_text.txt
8_GEN_PSOE_2011_text.txt
9_GEN_PSOE_2011_text.txt
10_GEN_PSOE_2011_text.txt
11_GEN_PSOE_2011_text.txt
12_GEN_PSOE_2011_text.txt
13_GEN_PSOE_2011_text.txt
14_GEN_PSOE_2011_text.txt
15_GEN_PSOE_2011_text.txt
16_GEN_PSOE_2011_text.txt
17_GEN_PSOE_2011_text.txt
18_GEN_PSOE_2011_text.txt
19_GEN_PSOE_2011_text.txt
20_GEN_PSOE_2011_text.txt
21_GEN_PSOE_2011_text.txt
22_GEN_PSOE_2011_text.txt
23_GEN_PSOE_2011_text.txt
24_GEN_PSOE_2011_text.txt
25_GEN_PSOE_2011_text.txt
26_GEN_PSOE_2011_text.txt
27_GEN_PSOE_2011_text.txt
28_GEN_PSOE_2011_text.txt
29_GEN_PSOE_2011_text.txt
30_GEN_PSOE_2011_text.txt
31_GEN_PSOE_2011_text.txt
32_GEN_PSOE_2011_text.txt

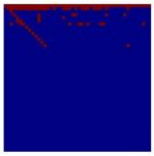
MAD PSOE 2011

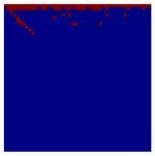

1_MAD_PSOE_2011_text.txt

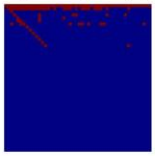

2_MAD_PSOE_2011_text.txt

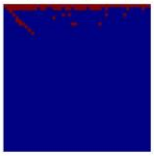

3_MAD_PSOE_2011_text.txt

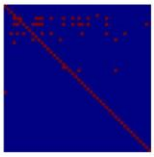

4_MAD_PSOE_2011_text.txt

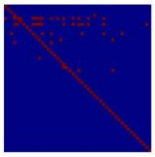

5_MAD_PSOE_2011_text.txt

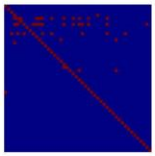

6_MAD_PSOE_2011_text.txt

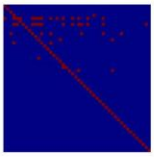

7_MAD_PSOE_2011_text.txt

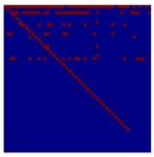

8_MAD_PSOE_2011_text.txt

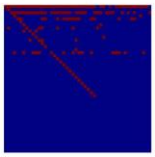

9_MAD_PSOE_2011_text.txt

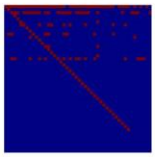

10_MAD_PSOE_2011_text.txt

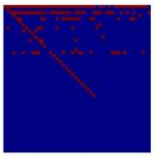

11_MAD_PSOE_2011_text.txt

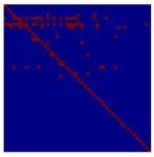

12_MAD_PSOE_2011_text.txt

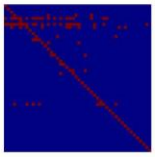

13_MAD_PSOE_2011_text.txt

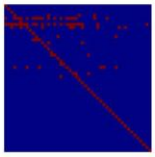

14_MAD_PSOE_2011_text.txt

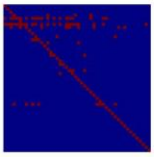

15_MAD_PSOE_2011_text.txt

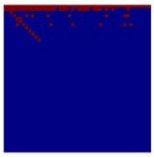

16_MAD_PSOE_2011_text.txt


17_MAD_PSOE_2011_text.txt

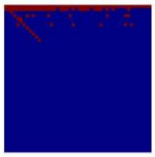

18_MAD_PSOE_2011_text.txt

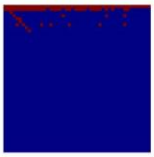

19_MAD_PSOE_2011_text.txt

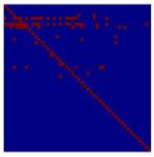

20_MAD_PSOE_2011_text.txt

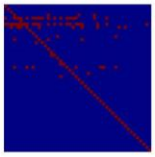

21_MAD_PSOE_2011_text.txt

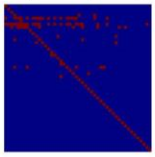

22_MAD_PSOE_2011_text.txt

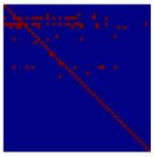

23_MAD_PSOE_2011_text.txt

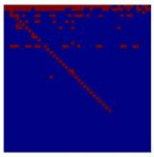

24_MAD_PSOE_2011_text.txt

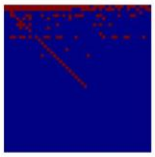

25_MAD_PSOE_2011_text.txt

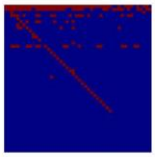

26_MAD_PSOE_2011_text.txt

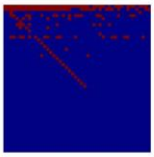

27_MAD_PSOE_2011_text.txt

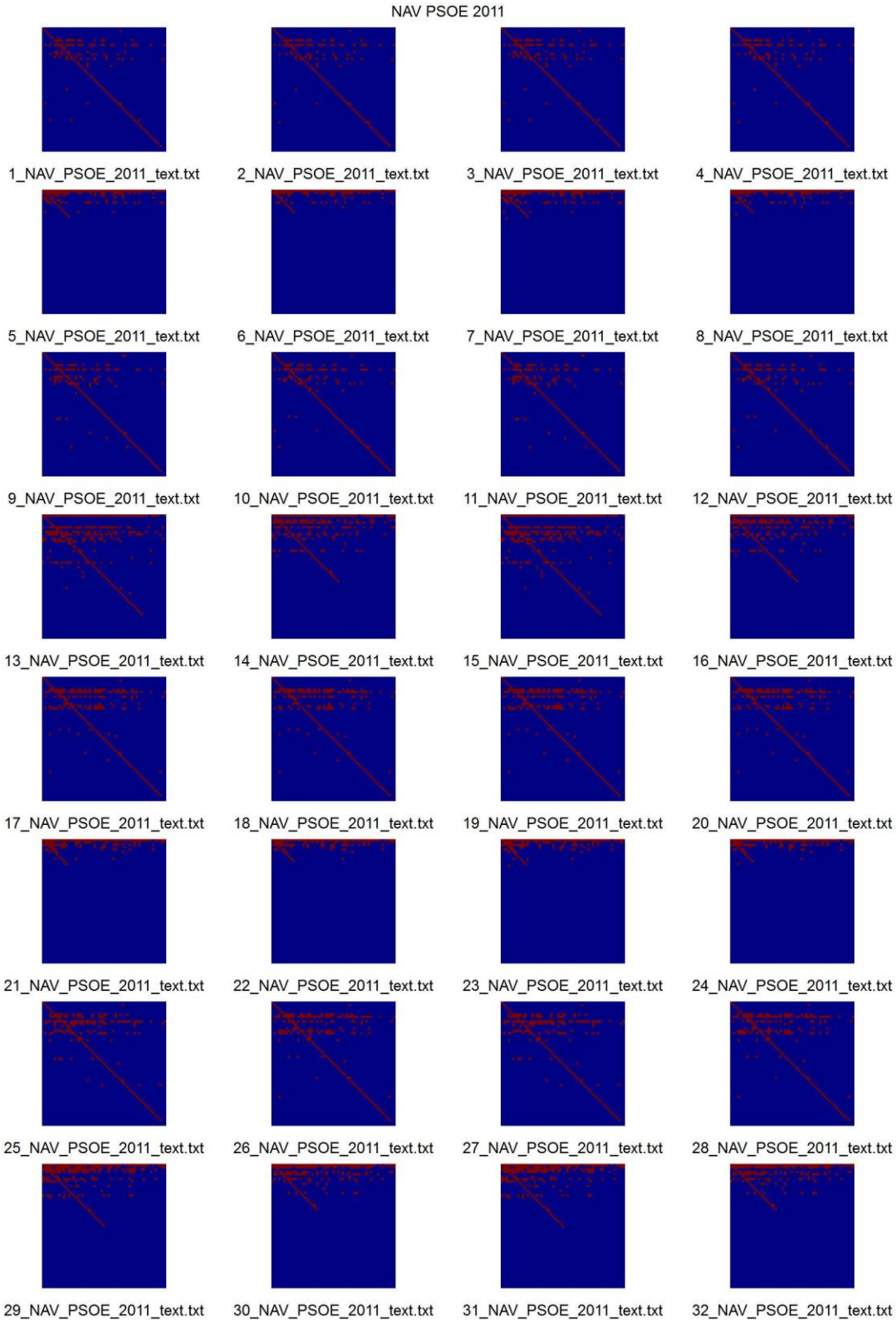

28_MAD_PSOE_2011_text.txt


29_MAD_PSOE_2011_text.txt

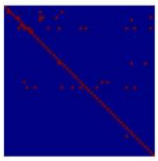

30_MAD_PSOE_2011_text.txt

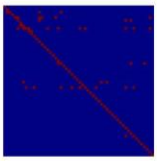

31_MAD_PSOE_2011_text.txt

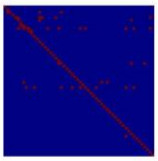

32_MAD_PSOE_2011_text.txt

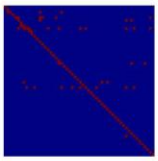

NAV PSOE 2011
1_NAV_PSOE_2011_text.txt
2_NAV_PSOE_2011_text.txt
3_NAV_PSOE_2011_text.txt
4_NAV_PSOE_2011_text.txt
5_NAV_PSOE_2011_text.txt
6_NAV_PSOE_2011_text.txt
7_NAV_PSOE_2011_text.txt
8_NAV_PSOE_2011_text.txt
9_NAV_PSOE_2011_text.txt
10_NAV_PSOE_2011_text.txt
11_NAV_PSOE_2011_text.txt
12_NAV_PSOE_2011_text.txt
13_NAV_PSOE_2011_text.txt
14_NAV_PSOE_2011_text.txt
15_NAV_PSOE_2011_text.txt
16_NAV_PSOE_2011_text.txt
17_NAV_PSOE_2011_text.txt
18_NAV_PSOE_2011_text.txt
19_NAV_PSOE_2011_text.txt
20_NAV_PSOE_2011_text.txt
21_NAV_PSOE_2011_text.txt
22_NAV_PSOE_2011_text.txt
23_NAV_PSOE_2011_text.txt
24_NAV_PSOE_2011_text.txt
25_NAV_PSOE_2011_text.txt
26_NAV_PSOE_2011_text.txt
27_NAV_PSOE_2011_text.txt
28_NAV_PSOE_2011_text.txt
29_NAV_PSOE_2011_text.txt
30_NAV_PSOE_2011_text.txt
31_NAV_PSOE_2011_text.txt
32_NAV_PSOE_2011_text.txt

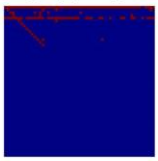
PV PP 2005

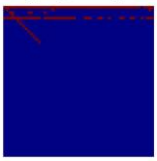

1_PV_PP_2005_text.txt

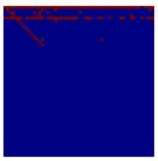

2_PV_PP_2005_text.txt

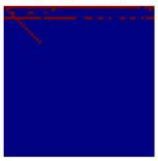

3_PV_PP_2005_text.txt

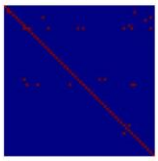

4_PV_PP_2005_text.txt

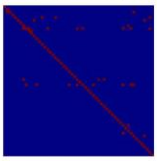

5_PV_PP_2005_text.txt

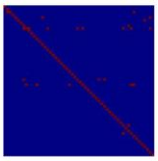

6_PV_PP_2005_text.txt

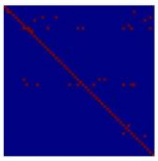

7_PV_PP_2005_text.txt

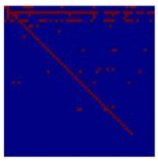

8_PV_PP_2005_text.txt

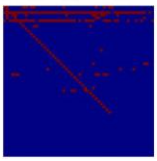

9_PV_PP_2005_text.txt

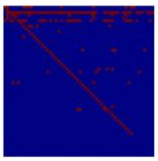

10_PV_PP_2005_text.txt

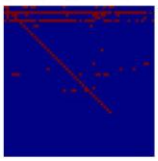

11_PV_PP_2005_text.txt

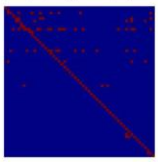

12_PV_PP_2005_text.txt

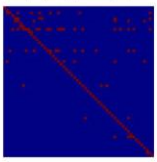

13_PV_PP_2005_text.txt

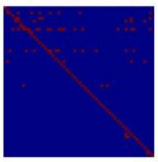

14_PV_PP_2005_text.txt

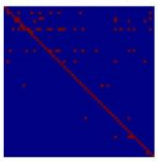

15_PV_PP_2005_text.txt

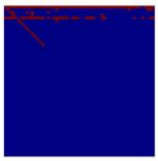

16_PV_PP_2005_text.txt

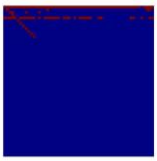

17_PV_PP_2005_text.txt

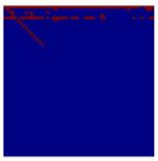

18_PV_PP_2005_text.txt

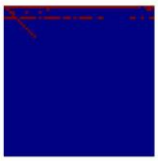

19_PV_PP_2005_text.txt

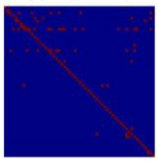

20_PV_PP_2005_text.txt

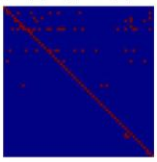

21_PV_PP_2005_text.txt

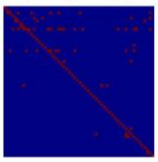

22_PV_PP_2005_text.txt

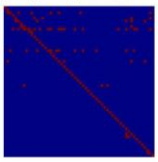

23_PV_PP_2005_text.txt

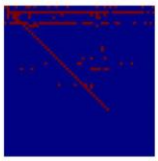

24_PV_PP_2005_text.txt

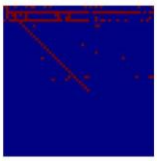

25_PV_PP_2005_text.txt

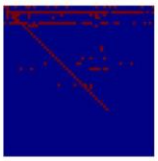

26_PV_PP_2005_text.txt

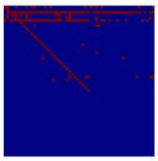

27_PV_PP_2005_text.txt

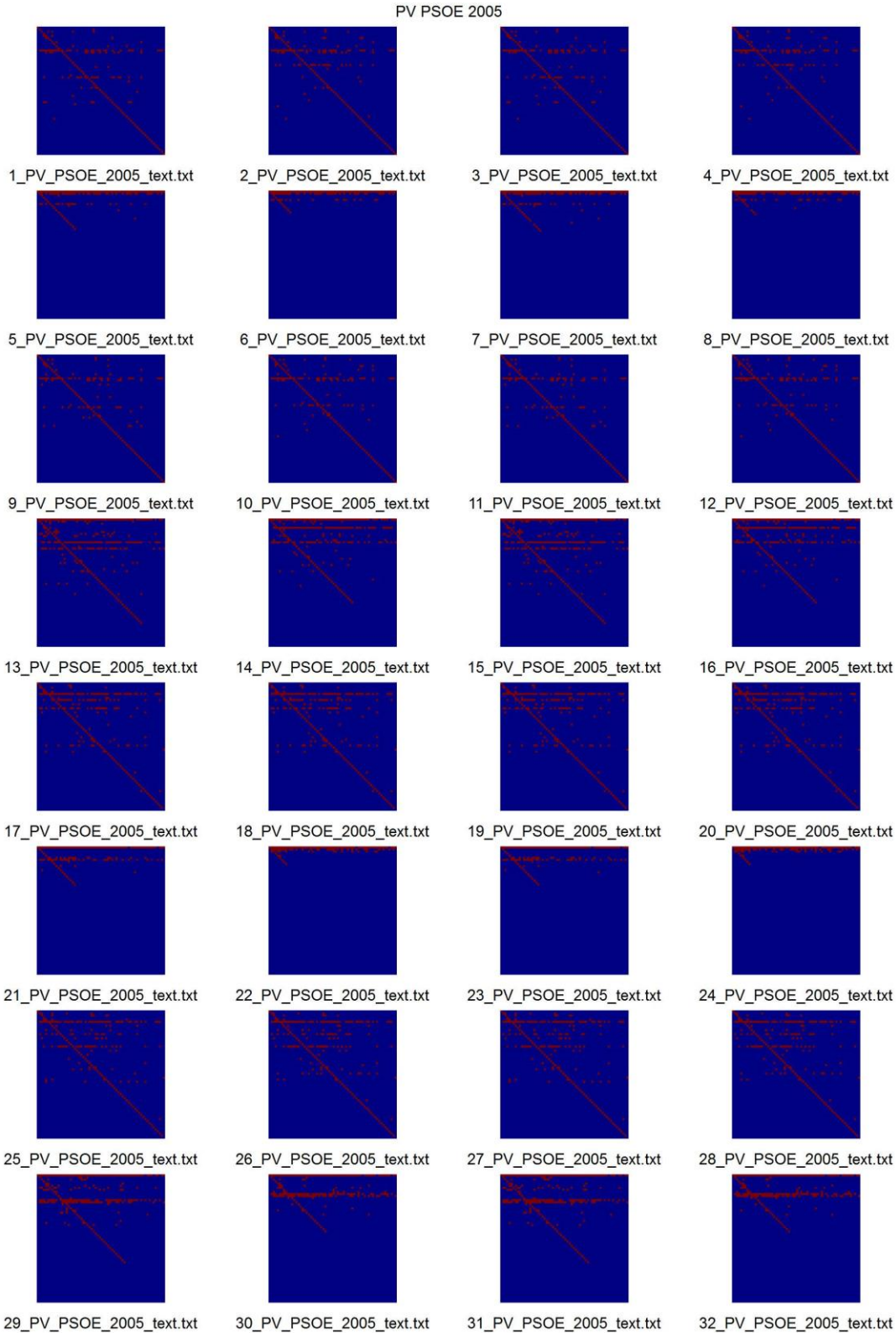

28_PV_PP_2005_text.txt


29_PV_PP_2005_text.txt

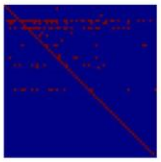

30_PV_PP_2005_text.txt

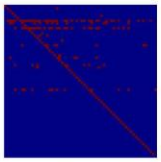

31_PV_PP_2005_text.txt

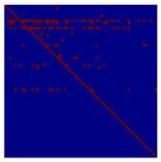

32_PV_PP_2005_text.txt

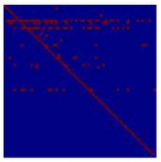

PV PSOE 2005
1_PV_PSOE_2005_text.txt
2_PV_PSOE_2005_text.txt
3_PV_PSOE_2005_text.txt
4_PV_PSOE_2005_text.txt
5_PV_PSOE_2005_text.txt
6_PV_PSOE_2005_text.txt
7_PV_PSOE_2005_text.txt
8_PV_PSOE_2005_text.txt
9_PV_PSOE_2005_text.txt
10_PV_PSOE_2005_text.txt
11_PV_PSOE_2005_text.txt
12_PV_PSOE_2005_text.txt
13_PV_PSOE_2005_text.txt
14_PV_PSOE_2005_text.txt
15_PV_PSOE_2005_text.txt
16_PV_PSOE_2005_text.txt
17_PV_PSOE_2005_text.txt
18_PV_PSOE_2005_text.txt
19_PV_PSOE_2005_text.txt
20_PV_PSOE_2005_text.txt
21_PV_PSOE_2005_text.txt
22_PV_PSOE_2005_text.txt
23_PV_PSOE_2005_text.txt
24_PV_PSOE_2005_text.txt
25_PV_PSOE_2005_text.txt
26_PV_PSOE_2005_text.txt
27_PV_PSOE_2005_text.txt
28_PV_PSOE_2005_text.txt
29_PV_PSOE_2005_text.txt
30_PV_PSOE_2005_text.txt
31_PV_PSOE_2005_text.txt
32_PV_PSOE_2005_text.txt

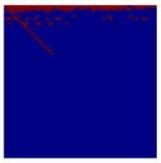
PV PSOE 2009

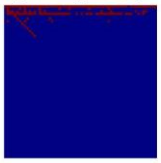

1_PV_PSOE_2009_text.txt

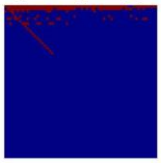

2_PV_PSOE_2009_text.txt

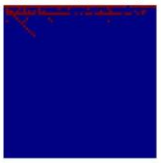

3_PV_PSOE_2009_text.txt

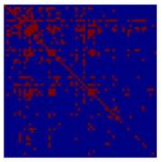

4_PV_PSOE_2009_text.txt

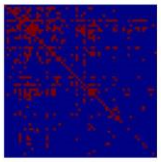

5_PV_PSOE_2009_text.txt

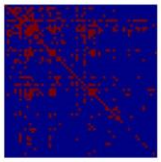

6_PV_PSOE_2009_text.txt

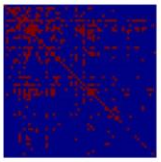

7_PV_PSOE_2009_text.txt

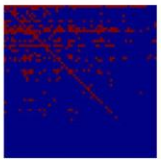

8_PV_PSOE_2009_text.txt

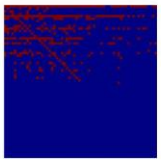

9_PV_PSOE_2009_text.txt

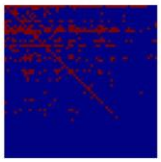

10_PV_PSOE_2009_text.txt

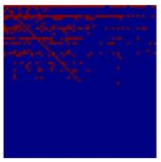

11_PV_PSOE_2009_text.txt

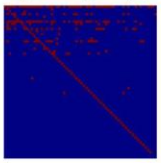

12_PV_PSOE_2009_text.txt

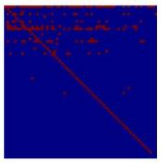

13_PV_PSOE_2009_text.txt

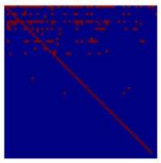

14_PV_PSOE_2009_text.txt

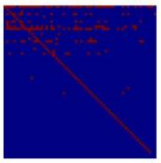

15_PV_PSOE_2009_text.txt

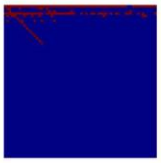

16_PV_PSOE_2009_text.txt

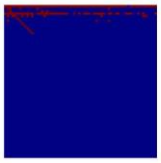

17_PV_PSOE_2009_text.txt

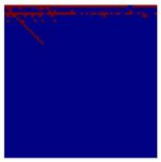

18_PV_PSOE_2009_text.txt

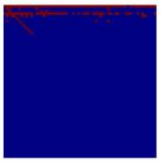

19_PV_PSOE_2009_text.txt

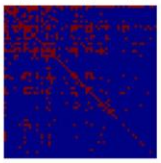

20_PV_PSOE_2009_text.txt

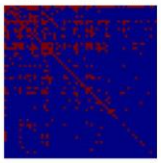

21_PV_PSOE_2009_text.txt

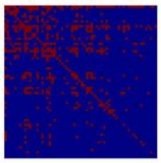

22_PV_PSOE_2009_text.txt

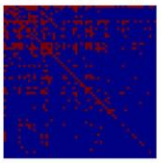

23_PV_PSOE_2009_text.txt

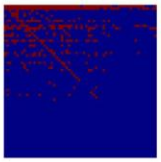

24_PV_PSOE_2009_text.txt

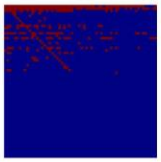

25_PV_PSOE_2009_text.txt

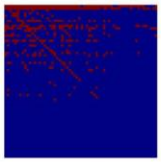

26_PV_PSOE_2009_text.txt

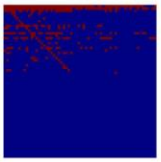

27_PV_PSOE_2009_text.txt

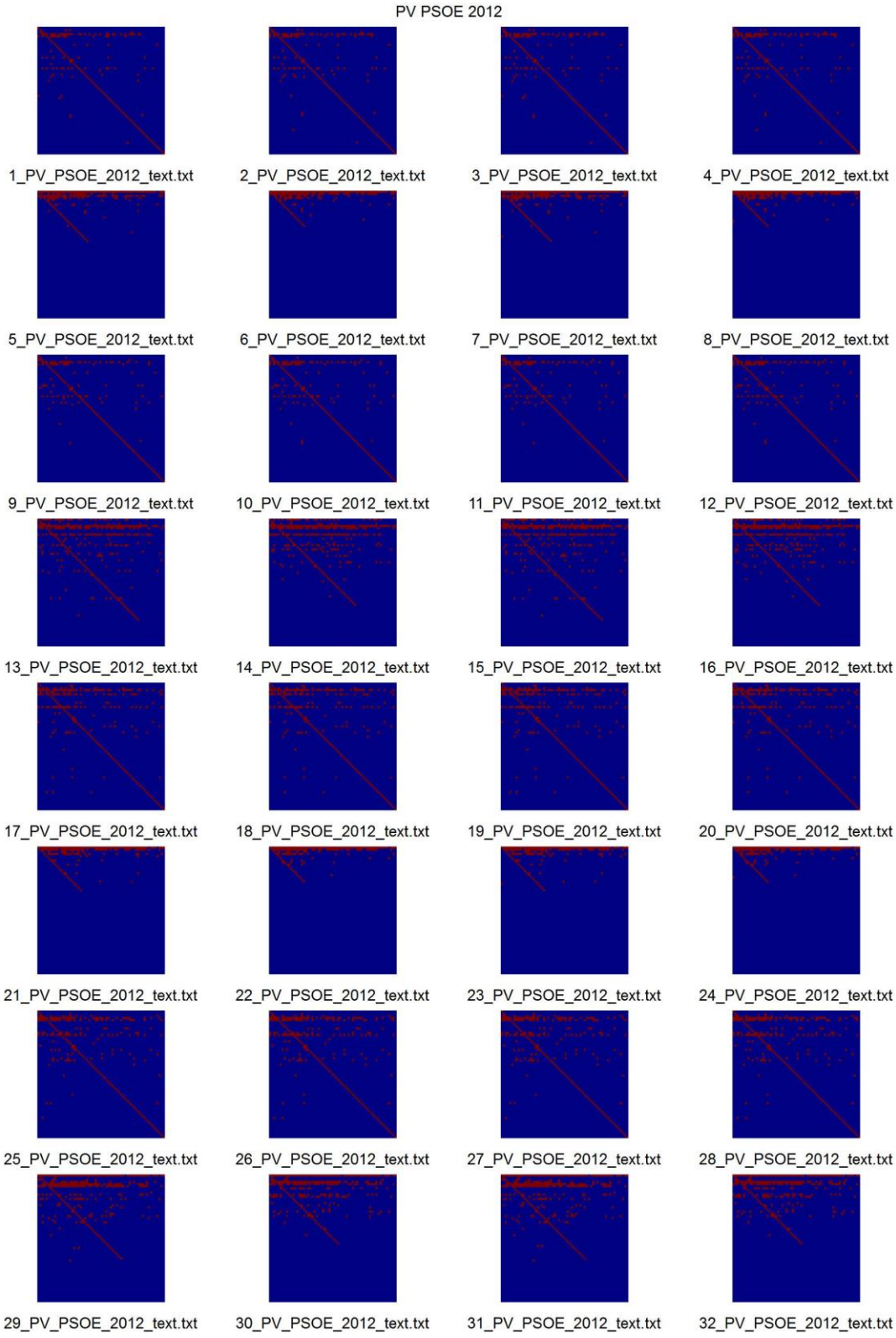

28_PV_PSOE_2009_text.txt


29_PV_PSOE_2009_text.txt

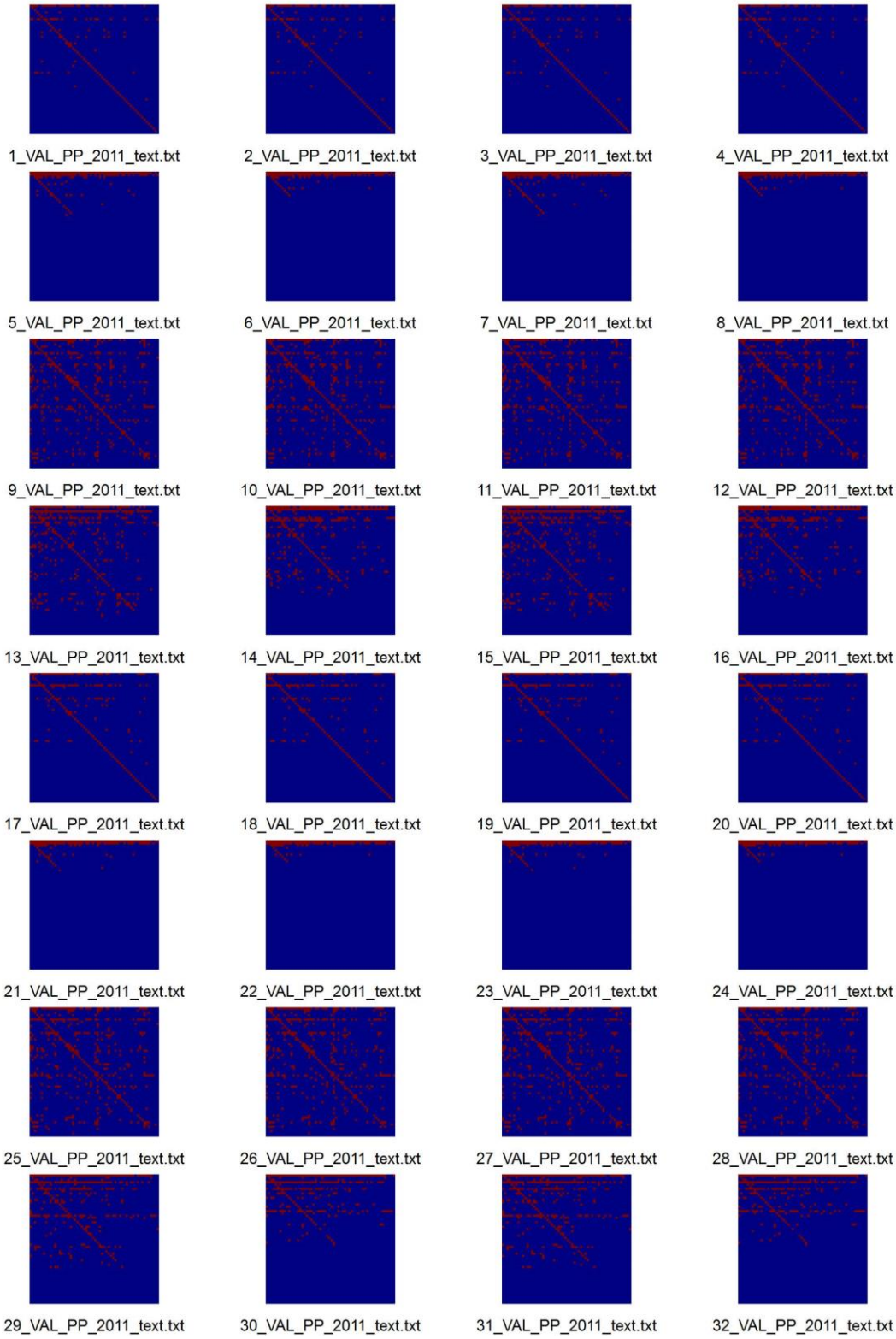

30_PV_PSOE_2009_text.txt


31_PV_PSOE_2009_text.txt

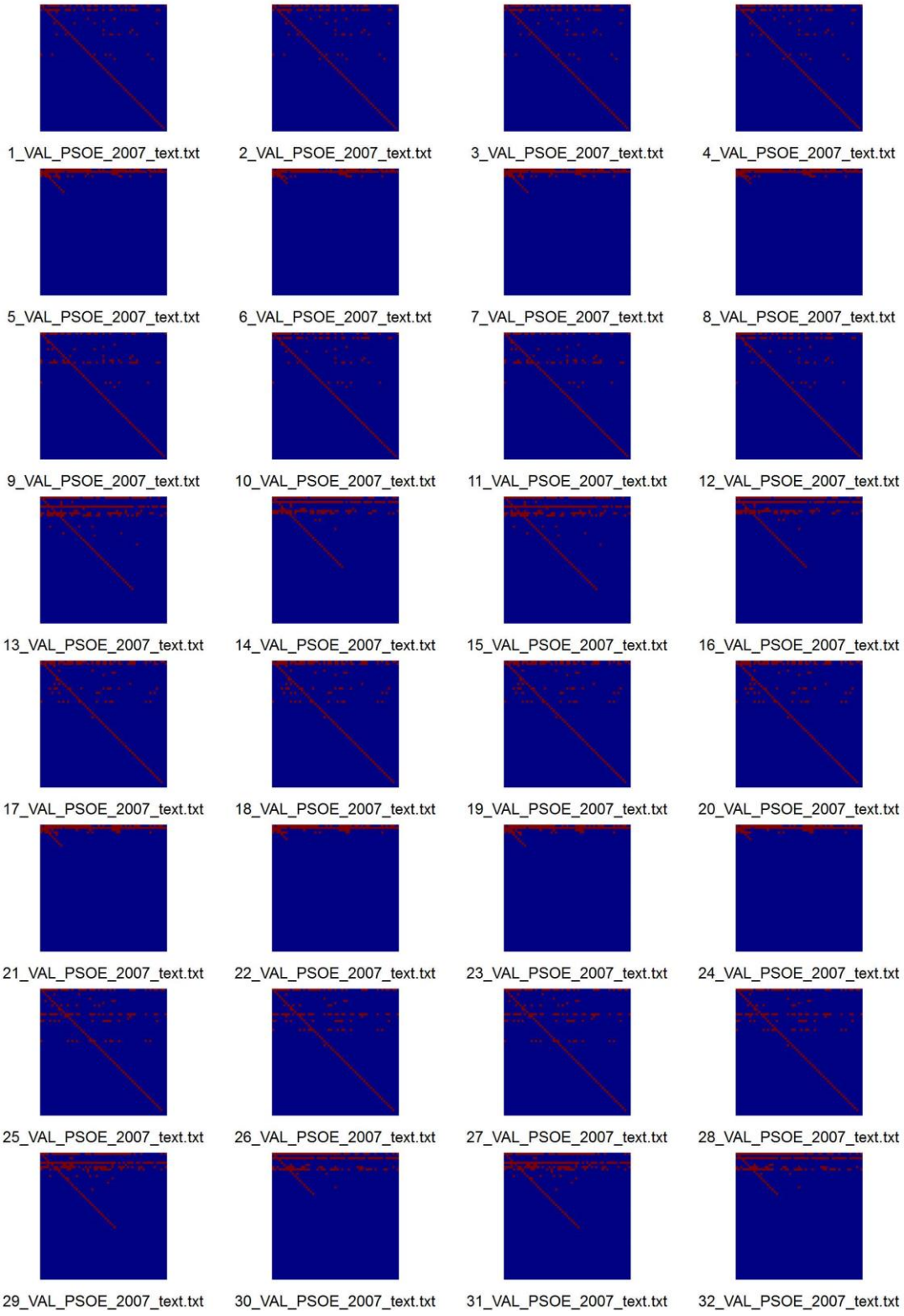

32_PV_PSOE_2009_text.txt


PV PSOE 2012
1_PV_PSOE_2012_text.txt
2_PV_PSOE_2012_text.txt
3_PV_PSOE_2012_text.txt
4_PV_PSOE_2012_text.txt
5_PV_PSOE_2012_text.txt
6_PV_PSOE_2012_text.txt
7_PV_PSOE_2012_text.txt
8_PV_PSOE_2012_text.txt
9_PV_PSOE_2012_text.txt
10_PV_PSOE_2012_text.txt
11_PV_PSOE_2012_text.txt
12_PV_PSOE_2012_text.txt
13_PV_PSOE_2012_text.txt
14_PV_PSOE_2012_text.txt
15_PV_PSOE_2012_text.txt
16_PV_PSOE_2012_text.txt
17_PV_PSOE_2012_text.txt
18_PV_PSOE_2012_text.txt
19_PV_PSOE_2012_text.txt
20_PV_PSOE_2012_text.txt
21_PV_PSOE_2012_text.txt
22_PV_PSOE_2012_text.txt
23_PV_PSOE_2012_text.txt
24_PV_PSOE_2012_text.txt
25_PV_PSOE_2012_text.txt
26_PV_PSOE_2012_text.txt
27_PV_PSOE_2012_text.txt
28_PV_PSOE_2012_text.txt
29_PV_PSOE_2012_text.txt
30_PV_PSOE_2012_text.txt
31_PV_PSOE_2012_text.txt
32_PV_PSOE_2012_text.txt

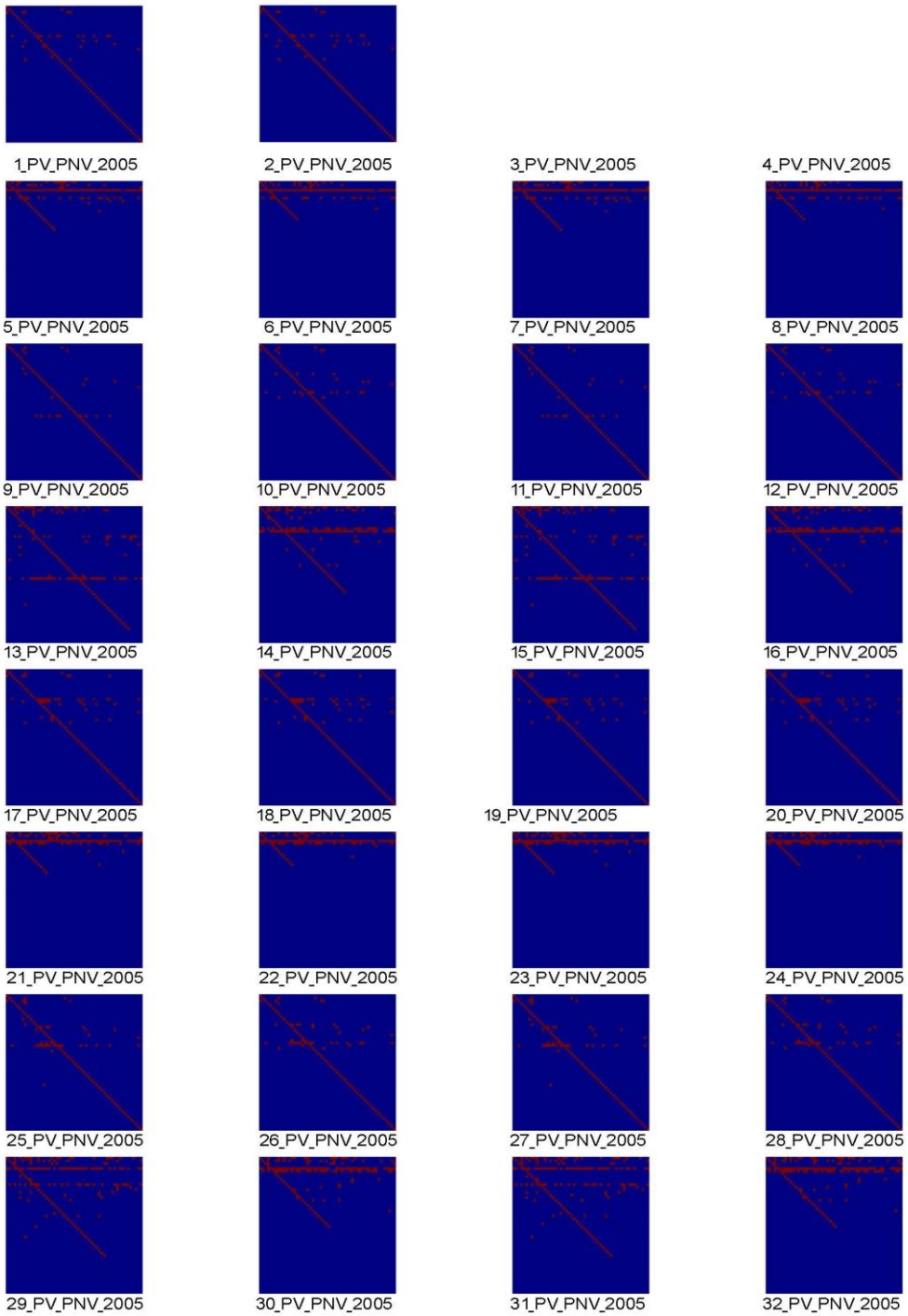
VAL PP 2011


1_VAL_PP_2011_text.txt
2_VAL_PP_2011_text.txt
3_VAL_PP_2011_text.txt
4_VAL_PP_2011_text.txt
5_VAL_PP_2011_text.txt
6_VAL_PP_2011_text.txt
7_VAL_PP_2011_text.txt
8_VAL_PP_2011_text.txt
9_VAL_PP_2011_text.txt
10_VAL_PP_2011_text.txt
11_VAL_PP_2011_text.txt
12_VAL_PP_2011_text.txt
13_VAL_PP_2011_text.txt
14_VAL_PP_2011_text.txt
15_VAL_PP_2011_text.txt
16_VAL_PP_2011_text.txt
17_VAL_PP_2011_text.txt
18_VAL_PP_2011_text.txt
19_VAL_PP_2011_text.txt
20_VAL_PP_2011_text.txt
21_VAL_PP_2011_text.txt
22_VAL_PP_2011_text.txt
23_VAL_PP_2011_text.txt
24_VAL_PP_2011_text.txt
25_VAL_PP_2011_text.txt
26_VAL_PP_2011_text.txt
27_VAL_PP_2011_text.txt
28_VAL_PP_2011_text.txt
29_VAL_PP_2011_text.txt
30_VAL_PP_2011_text.txt
31_VAL_PP_2011_text.txt
32_VAL_PP_2011_text.txt

VAL PSOE 2007



PV PNV 2005


1_PV_PNV_2005
2_PV_PNV_2005
3_PV_PNV_2005
4_PV_PNV_2005
5_PV_PNV_2005
6_PV_PNV_2005
7_PV_PNV_2005
8_PV_PNV_2005
9_PV_PNV_2005
10_PV_PNV_2005
11_PV_PNV_2005
12_PV_PNV_2005
13_PV_PNV_2005
14_PV_PNV_2005
15_PV_PNV_2005
16_PV_PNV_2005
17_PV_PNV_2005
18_PV_PNV_2005
19_PV_PNV_2005
20_PV_PNV_2005
21_PV_PNV_2005
22_PV_PNV_2005
23_PV_PNV_2005
24_PV_PNV_2005
25_PV_PNV_2005
26_PV_PNV_2005
27_PV_PNV_2005
28_PV_PNV_2005
29_PV_PNV_2005
30_PV_PNV_2005
31_PV_PNV_2005
32_PV_PNV_2005

9.3 Contenido del CD

Como contenido del CD se presenta la siguiente estructura de carpetas:

En la carpeta Documentación se encuentran por un lado la memoria del proyecto y una carpeta (Fuentes) con los archivos obtenidos vía internet de la bibliografía.

En la carpeta Código se encuentran todos los archivos .mat (Matlab) además de todas las figuras generadas por el programa. Asimismo están adjuntos todos los archivos que han servido como fuente para la clasificación, dentro de la carpeta Data.