PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

CLASIFICACIÓN AUTOMÁTICA DE TEXTOS MULTILINGÜES

NELSON EDWARD SALAZAR WILLIAMS

Profesor Guía: **Rodrigo Alfaro Arancibia**

INFORME FINAL DEL PROYECTO
PARA OPTAR AL TÍTULO PROFESIONAL DE
INGENIERO CIVIL EN INFORMÁTICA

OCTUBRE – 2012

Pontificia Universidad Católica de Valparaíso
Facultad de Ingeniería
Escuela de Ingeniería Informática

CLASIFICACIÓN AUTOMÁTICA DE TEXTOS MULTILINGÜES

NELSON EDWARD SALAZAR WILLIAMS

Profesor Guía: **Rodrigo Alfaro Arancibia**

Profesor Co-referente: **Claudio Cubillos Figueroa**

Carrera: **Ingeniería Civil en Informática**

OCTUBRE – 2012

Dedicado a mi familia que me ha amado y apoyado en este hermoso camino del conocimiento y a quienes han hecho de esta travesía una experiencia maravillosa. Un millón de gracias a mi tutor y profesores, pues son ellos quienes alimentaron con su sabiduría todo este camino académico.

Índice

Resumen…… iv
Lista de Figuras…… v
Lista de Tablas…….. vii
Glosario de Términos…… viii
1. Introducción…… 1
2. Marco Teórico …… 2
2.1. Aprendizaje de Máquinas..... 2
2.2. Categorización de Textos..... 3
2.2.1. Categorización de Textos Multi-Etiquetas..... 3
2.3. Categorización Automática de Textos Bilingües..... 4
2.4. Wordnet..... 4
2.4.1. EuroWordnet..... 5
2.5. Traducción Automática..... 6
2.6. Reuters..... 7
3. Definición de Objetivos……...8
3.1. Objetivo General..... 8
3.2. Objetivos Específicos..... 8
4. Estado del Arte……9
4.1. Técnicas de traducción automática..... 9
4.1.1. Traducción automática basada en reglas..... 9
4.1.2. Traducción automática basada en corpus..... 10
4.2. Técnicas de Entrenamiento de documentos..... 12
4.2.1. Entrenamiento poli-lingüístico..... 12
4.2.2. Entrenamiento Lingo-Transversal..... 12
4.2.3. Lenguaje Esperanto..... 12
4.3. Representación de Documentos..... 13
4.3.1. Reducción de dimensionalidad..... 15
4.4. División de documentos..... 18
4.4.1. Grupo de Entrenamiento y Validación (TV)..... 18
4.4.2. Grupo de Pruebas (Te)..... 18
4.5. Fase de Entrenamiento..... 19

4.5.1. Mapear términos en synsets 19
4.5.2. Capturar las relaciones entre los synsets 20
4.5.3. Crear perfiles conceptuales de categorías 21
4.6. Fase de clasificación 22
4.6.1. Traducción y generación de un vector conceptual 22
4.6.2. Pesaje de las categorías 22
4.6.3. Determinación de la categoría de un documento 23
5. Método……...24
5.1. Fase de Preparación 25
5.2. Fase de Entrenamiento 25
5.3. Fase de Prueba 26
5.4. Evaluación..... 26
5.4.1. Precisión 26
5.4.2. Recall..... 27
5.4.3. Exactitud (accuracy)..... 27
5.4.4. Medición-F 27
6. Herramientas……..28
6.1. Reuters 28
6.2. Traducción: Apertium 29
6.3. Reducción de Dimensionalidad: Wordnet 31
6.4. Entrenamiento y Clasificación: Aplicación Web..... 31
7. Implementación…….33
7.1. Codificación y ejecución..... 36
7.2. Interfaz Grafica 37
7.3. Servidor..... 42
8. Análisis de Resultados…… 43
8.1. Prueba N°1: Ratio entre Archivos de Entrenamiento y Testing variables..... 44
8.1.1. Condiciones Iniciales Constantes..... 44
8.1.2. Condiciones Iniciales Variables 44
8.1.3. Resultados 45
8.1.4. Conclusiones Parciales..... 61
8.2. Prueba N°2: Cantidad de archivos procesados variable 62

8.2.1. Condiciones Iniciales Constantes 62
8.2.2. Condiciones Iniciales Variables 62
8.2.3. Resultados 63
8.2.4. Conclusiones Parciales 79
8.3. Prueba N°3: Inclución de varios idiomas variable 80
8.3.1. Condiciones Iniciales Constantes 80
8.3.2. Condiciones Iniciales Variables 80
8.3.3. Resultados 81
8.3.4. Conclusiones Parciales 89
8.4. Conclusiones generales de los resultados 90
9. Conclusiones del estudio……...91
10. Referencias……92
11. Anexo……93
11.1. Código Fuente del programa desarrollado (extracto) 93

Resumen

Este documento presentará un método que solucionará la problemática que se presenta cuando se necesita de clasificar textos en distintos lenguajes de manera automática. Se presentan los resultados de la investigación e implantación realizada sobre el tema de Clasificación Automática de Textos Multilingües. Se explican las distintas técnicas ya existentes documentadas que permiten esta categorización, junto a herramientas seleccionadas para el método propuesto como solución, hasta llegar a la implantación del mismo con sus respectivos resultados.

En este contexto, el objetivo del presente consiste en desarrollar un método que permita la clasificación automática de textos, disponibles en una variedad de idiomas, en categorías declaradas previamente. Para lograr esto se propone una secuencia de pasos ordenados que unidos desembocan en la solución tecnológica propuesta a lo largo del informe. De los resultados obtenidos se puede adelantar que la investigación ha reportado los frutos esperados y que sí es posible desarrollar esta tarea de manera automática.

El documento comienza con una introducción que contextualiza la motivación del proyecto, seguido del marco teórico que establece los conceptos utilizados a lo largo del informe. Se definen los objetivos del proyecto tanto generales como específicos. Se define el estado del arte, donde se explican en detalle todas las técnicas utilizadas en el método propuesto. Seguido de lo anterior se declara el método propuesto en sí, junto con las herramientas utilizadas para implementarlo. Ya declarado lo anterior, se detalla la implantación del método junto con los resultados obtenidos y sus respectivos análisis.

Lista de Figuras

Figura 2-1 - Ejemplo de relación jerárquica 5
Figura 4-1: Técnicas traducción 10
Figura 4-2: Representación vectorial de un texto 13
Figura 4-3: Ejemplo de reducción de dimensionalidad 16
Figura 5-1: Método Propuesto 24
Figura 7-1: Diagrama de casos de uso del método propuesto. 33
Figura 7-2: Diagrama de Actividades del proceso de preparación de documentos y entrenamiento 34
Figura 7-3: Diagrama de Actividades del proceso de clasificación de nuevos documentos. 35
Figura 7-4: Interfaz Inicial 37
Figura 7-5: Interfaz Entrenamiento 37
Figura 7-6: Interfaz Entrenamiento en ejecución 38
Figura 7-7: Interfaz Resultados entrenamiento 39
Figura 7-8: Interfaz Clasificación de 1 documento 40
Figura 7-9: Interfaz Resultados Clasificación de 1 documento 41
Figura 8-1: Categoria "corporate": 80% Entrenamiento, 20% Testing 45
Figura 8-2: Categoria "economics": 80% Entrenamiento, 20% Testing 46
Figura 8-3: Categoria "government": 80% Entrenamiento, 20% Testing 47
Figura 8-4: Categoria "markets": 80% Entrenamiento, 20% Testing 48
Figura 8-5: Categoria "corporate": 60% Entrenamiento, 40% Testing 49
Figura 8-6: Categoria "economics": 60% Entrenamiento, 40% Testing 50
Figura 8-7: Categoria "government": 60% Entrenamiento, 40% Testing 51
Figura 8-8: Categoria "markets": 60% Entrenamiento, 40% Testing 52
Figura 8-9: Categoria "corporate": 40% Entrenamiento, 60% Testing 53
Figura 8-10: Categoria "economics": 40% Entrenamiento, 60% Testing 54
Figura 8-11: Categoria "government": 40% Entrenamiento, 60% Testing 55
Figura 8-12: Categoria "markets": 40% Entrenamiento, 60% Testing 56
Figura 8-13: Categoria "corporate": 20% Entrenamiento, 80% Testing 57
Figura 8-14: Categoria "economics": 20% Entrenamiento, 80% Testing 58
Figura 8-15: Categoria "government": 20% Entrenamiento, 80% Testing 59
Figura 8-16: Categoria "markets": 20% Entrenamiento, 80% Testing 60
Figura 8-17: Categoria "corporate": 500 archivos 63
Figura 8-18: Categoria "economics": 500 archivos 64
Figura 8-19: Categoria "government": 500 archivos 65
Figura 8-20: Categoria "markets": 500 archivos 66
Figura 8-21: Categoria "corporate": 1000 archivos 67
Figura 8-22: Categoria "economics": 1000 archivos 68
Figura 8-23: Categoria "government": 1000 archivos 69
Figura 8-24: Categoria "markets": 1000 archivos 70
Figura 8-25: Categoria "corporate": 5000 archivos 71
Figura 8-26: Categoria "economics": 5000 archivos 72
Figura 8-27: Categoria "government": 5000 archivos 73
Figura 8-28: Categoria "markets": 5000 archivos 74
Figura 8-29: Categoria "corporate": 10000 archivos 75
Figura 8-30: Categoria "economics": 10000 archivos 76

FIGURA 8-31: CATEGORIA "GOVERNMENT": 10000 ARCHIVOS 77
FIGURA 8-32: CATEGORIA "MARKETS": 10000 ARCHIVOS 78
FIGURA 8-33: CATEGORIA "CORPORATE": SIN MULTILINGUAL 81
FIGURA 8-34: CATEGORIA "ECONOMICS": SIN MULTILINGUAL 82
FIGURA 8-35: CATEGORIA "GOVERNMENT": SIN MULTILINGUAL 83
FIGURA 8-36: CATEGORIA "MARKETS": SIN MULTILINGUAL 84
FIGURA 8-37: CATEGORIA "CORPORATE": CON MULTILINGUAL 85
FIGURA 8-38: CATEGORIA "ECONOMICS": CON MULTILINGUAL 86
FIGURA 8-39: CATEGORIA "GOVERNMENT": CON MULTILINGUAL 87
FIGURA 8-40: CATEGORIA "MARKETS": CON MULTILINGUAL 88

Lista de Tablas

TABLA 1 – LENGUAJES USADOS EN LA INTERNET 1
TABLA 2-1: NUMERO DE PALABRAS Y SYNSET EN WORDNET 2.1 4
TABLA 2-2: NUMERO DE HISTORIAS PRODUCIDAS POR REUTERS EN 1997 QUE FUERON EDITADAS O CORREGIDAS MANUALMENTE 7

Glosario de Términos

- Metadata: Son datos altamente estructurados que describen información, describen el contenido, la calidad, la condición y otras características de los datos. Son datos sobre datos.

- Ontología: Hace referencia a la formulación de un exhaustivo y riguroso esquema conceptual dentro de uno o varios dominios dados; con la finalidad de facilitar la comunicación y el intercambio de información entre diferentes sistemas y entidades

- Semántica: Se refiere a los aspectos del significado, sentido o interpretación del significado de un determinado elemento, símbolo, palabra, expresión o representación formal.

- Léxico: Es el vocabulario de un idioma o región, el diccionario de una lengua o el caudal de modismos y voces de un autor.

- Corpora (corpus): Es un conjunto, normalmente muy amplio, de ejemplos reales de uso de una lengua. Estos ejemplos pueden ser textos (típicamente), o muestras orales (normalmente transcritas).

1. Introducción

Con el incremento del uso de la Internet a nivel mundial, el número de documentos digitales disponibles en la comunidad ha crecido vertiginosamente a lo largo de estos años. Al momento de clasificar esta enormidad de documentos, sin la utilización de métodos automáticos, se necesita de personal experto en dicha tarea, lo que encarece los costos de ésta. Más aún, la diversidad de la disponibilidad de estos documentos en distintos lenguajes hace todavía más compleja la tarea.

Considerando esta problemática, se entiende que en la categorización de textos digitales es necesaria la utilización de un método que automatice el proceso. Es por esto que el tema del presente proyecto se basa en la necesidad de crear un método que permita clasificar textos automáticamente en distintas categorías predefinidas de la manera más precisa y exacta. El presente documento considerará la resolución de clasificación en categorías binarias, es decir, si el documento pertenece o no a alguna categoría.

Además dicha categorización deberá tener la capacidad de poder realizarse independiente del idioma en que se encuentre disponible el texto, siendo ésta, para el desarrollo del proyecto, la lengua Española, el Inglés y el Francés, pues son unos de las lenguajes más hablado en el mundo (ver **¡Error! No se encuentra el origen de la referencia.**).

Tabla 1 – Lenguajes Usados en la Internet

Lenguajes Usados en la Internet [6]					
Idioma	**#Usuarios**	**Penetración**	**Crecimiento (2000-2009)**	**%Usuarios**	**Población Esperada (2009)**
Inglés	478,442,379	37.9 %	237.0 %	27.6 %	1,263,830,976
Chino	383,650,713	27.9 %	1,087.7 %	22.1 %	1,373,859,774
Español	136,524,063	33.2 %	650.9 %	7.9 %	411,631,985

2. Marco Teórico

A continuación se describen los conceptos relevantes para la comprensión y contextualización del presente estudio.

2.1. Aprendizaje de Máquinas

Del inglés *Machine Learning* (ML), es una rama de la Inteligencia Artificial cuyo objetivo es el desarrollo de técnicas para que las máquinas puedan aprender. En el campo de la clasificación de textos, este paradigma apunta a un proceso inductivo que construye un clasificador automático de textos, vía el aprendizaje de las características de un grupo de documentos previamente clasificado en alguna de las categorías de interés [2].

Las ventajas de este enfoque radican en la exactitud de los resultados, que es comparable con aquella alcanzada por expertos humanos, lo que implica un considerable ahorro del trabajo de expertos.

En la actualidad existe una diversidad de algoritmos de aprendizaje automático dentro de las que destacan:

1. Aprendizaje Supervisado: El algoritmo se realiza en función de las entradas y las salidas: debe haber una correspondencia entre las entradas y las salidas deseadas del sistema.
2. Aprendizaje No Supervisado: El algoritmo se basa únicamente en las entradas al sistema. No tiene información de cómo deben ser las salidas.
3. Aprendizaje por Refuerzo: La entrada es el entorno que rodea al sistema, el cual aprende de éste. Este algoritmo se basa en el *feedback* que obtiene de su entorno luego de sus acciones.

Es por esto que es crítico el modelamiento del sistema y de su entorno para la obtención de resultados lo más óptimo posible.

2.2. Categorización de Textos

Categorización de textos (TC) es la actividad de etiquetar textos, escritos en lenguaje natural, en categorías con temáticas predefinidas [2]. Matemáticamente significa asignar un valor Booleano a la pareja $< d_j, c_i > \in D \, x \, C$ donde D corresponde al dominio de documentos y con $C = c_1, \ldots, c_c$ un conjunto de categorías predefinidas. Un valor verdadero T asignado a la pareja $< d_j, c_i >$ indica la decisión de archivar d_j bajo c_i, mientras que un valor falso F indica la decisión de no archivar d_j bajo c_i. Más formalmente, la tarea de la clasificación de textos, es buscar la función objetivo $\Phi : \mathrm{D \, x \, C} \rightarrow \mathrm{T, F}$ (que describe cómo deberían ser idealmente clasificados los textos), por medio de la función $\Phi : \mathrm{D \, x \, C} \rightarrow \mathrm{T, F}$ llamado clasificador, de tal manera que Φ y Φ *coincidan* lo más posible. Esta coincidencia de las funciones, cuando es medida, es llamada efectividad.

Para el desarrollo del este proyecto se deben hacer las siguientes consideraciones:

- Las categorías no representan conocimiento adicional, simplemente se consideran como etiquetas simbólicas.
- No existe conocimiento exógeno disponible, como por ejemplo lo es la metadata. Entonces, dicho conocimiento proviene únicamente de la parte endógena del sistema, principalmente obtenida de lo extraído desde los documentos.

2.2.1. Categorización de Textos Multi-Etiquetas

El apellido Multi-Etiquetas hace referencia que un documento puede ser clasificado en ninguna, una o más categorías pertenecientes al conjunto $C = c_1, \ldots, c_c$. En esta familia de categorizaciones es posible hacer una subdivisión según la cantidad de etiquetas que puede tener un documento.

En el caso que cada documento $d_j \in D$ deba ser asignado a una única categoría, corresponderá al caso de *etiquetado simple*. Cuando un documento pueda ser asociado desde cero hasta C categorías se le denominará caso de *etiquetado múltiple.* Del etiquetado simple se extiende un caso particular denominado *etiquetado binario*. En este último, cada documento $d_j \in D$ debe ser asignado a la categoría c_i o a su complemento c_i.

Desde un punto de vista teórico el caso binario (por ende también el etiquetado simple) es más general que el etiquetado múltiple. Esto es cierto pues el problema de etiquetado múltiple bajo $c_1, \ldots, c_C$ puede ser visto como C problemas independientes de clasificación binaria bajo c_i, c_i con $i = 1, \ldots, C$. Sin embargo se requiere que las categorías sean

independientes unas de otras, esto es: para cualquier c', c'' el valor de $\Phi(d_j, c')$ no dependa del valor de $\Phi(d_j, c'')$, y viceversa.

2.3. Categorización Automática de Textos Bilingües

Es una extensión de la clasificación automática de textos en la que se pretende categorizar un conjunto de textos disponible en dos idiomas distintos sin la intervención de humanos expertos. Formalmente, dado dos colecciones de documentos $D = \{D_e, D_f\}$ en dos idiomas distintos *e* y *f*, y $C = c_1, \ldots, c_c$ el conjunto de categorías predefinidas en un lenguaje pivote, se busca que el clasificador $\Phi : D \times C \rightarrow T, F$ tenga la mejor efectividad en base a la función $\Phi : D \times C \rightarrow T, F$ [4].

2.4. Wordnet

Es una ontología de herencia léxica que provee de muchas herramientas que pretende representar algunos de los aspectos semánticos del léxico. La universidad de Princetons ha construido ésta a lo largo de una década. En su versión actual (Wordnet 2.1) es posible encontrar más de 155000 *formas de palabras* organizadas en 117597 significados. Dentro de las formas de palabras se encuentran los sustantivos, verbos, adjetivos y adverbios [1] (ver Tabla 2-1).

Tabla 2-1: Numero de palabras y synset en Wordnet 2.1 [7]

	Formas de Palabra	**Synsets**
Sustantivos	117.097	81.426
Verbos	11.488	13.650
Adjetivos	22.141	18.877
Adverbios	4.601	3.644
Total	155.327	117.597

Los sustantivos están representados como una jerarquía de nodos (ver **¡Error! No se encuentra el origen de la referencia.**) donde cada uno de estos corresponde a un significado de palabra o, según es definido por Wordnet, un *synset.* Un synset es simplemente un conjunto de palabras que expresan lo mismo en al menos un contexto. Por ejemplo $automóvil, camión$ es un synset que dice lo mismo que el concepto "vehículo motorizado".

Los synsets están conectados unos a otros a través de diversas relaciones semánticas. Las más importantes relaciones entre sustantivos son las relaciones de hipónimo y de

hiperónimo. La relación de hiperónimo entre los synsets A y B quiere decir que "B es un tipo de A". Por el otro lado, un hipónimo es el inverso de un hiperónimo, es decir que "A es un tipo de B". Por ejemplo equipo computacional, procesador de data es un hiperónimo de "computador personal". Usualmente cada synset tiene un único hiperónimo, es por esto que Wordnet se estructura jerárquicamente.

Otra relación relevante que ofrece Wordnet es la de los merónimos y los holónimos. Un holónimo corresponde a la relación "es parte de" entre los synsets. Por ejemplo, el sysnet $\{teclado\}$ es un merónimo del synset $computador, equipo\ electrónico, procesador\ de\ data$.

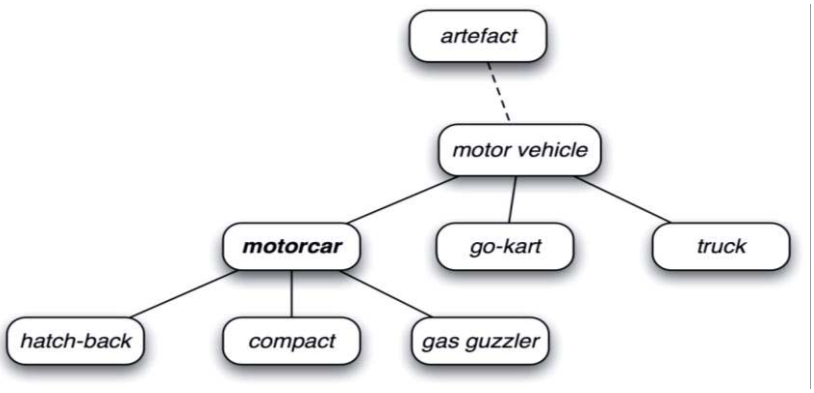

Figura 2-1 - Ejemplo de relación jerárquica

2.4.1. EuroWordnet

Es una base de datos multilingüe desarrollado por una asociación de universidades europeas. Se estructura de la misma manera que Wordnet pero con la diferencia que se encuentra en varios idiomas occidentales (holandés, italiano, español, alemán, francés, checo y estonio).

Utilizar EuroWordnet ofrece más ventajas que Wordnet, pues principalmente se libera del proceso de traducción (que es un proceso costoso). Lamentablemente éste es licenciado y queda fuera de los alcances de este proyecto.

2.5. Traducción Automática

La traducción automática (AT), conocida también del término en ingles Machine Translation (MT), se refiere a sistemas computacionales donde es posible realizar traducciones sin asistencia humana. A un nivel muy básico, la traducción consiste en la sustitución automática de términos en un texto de algún lenguaje a otro objetivo. También existen técnicas más complejas donde se puede hacer traducción de frases e inclusive expresiones idiomáticas [5]. En la práctica es complejo obtener una traducción de buena calidad sin la participación de un humano, pues al menos ésta debe ser revisada para ver si fue hecha de buena manera.

2.6. Reuters

Reuters es un corpus de 800.000 historias periodísticas que fueron manualmente clasificadas, las cuales fueron liberadas por Reuters Ltd. con fines de investigación. Esta clasificación manual se llevó una primera vez y se generó el RCV1-v1 (Reuters Corpus Volume I). Seguidamente el personal encargado de la clasificación generó una versión corregida de ésta, la cual fue denominada RCV1-v2.

El corpus se encuentra clasificado bajo tres distintas reglas:

1. Por tópicos (Topics): Los documentos son clasificados en 4 categorías jerárquicas. Como regla un documento de tener por lo menos una categoría asociada de tópico. Las categorías de tópicos son las siguientes:
    a. CCAT: Corporaciones/Industrias (Corporate/Industrial)
    b. ECAT: Economía (Economy)
    c. GCAT: Guvernamental/Social (Government/Social)
    d. MCAT: Mercado
2. Por industria (Industries): Los documentos son clasificados en 10 subjerarquias relacionadas con las industrias
3. Por región (Region): Los documentos son agrupados por la ubicación geográfica y por la asociación económica/política que tengan. Los documentos como regla tienen al menos una región asociada

El corpus fue generado en base a procedimientos manuales y automáticos (ver Tabla 2-2). Estos últimos aún así fueron corregidos por personal encargado. Con esto se recalca el hecho que la clasificación es propensa a tener errores al tratarse de un proceso subjetivo.

Tabla 2-2: Numero de historias producidas por Reuters en 1997 que fueron editadas o corregidas manualmente

		Manualmente Corregida	
		NO	**SI**
Manualmente Editada	**NO**	170.745	334.975
	SI	228.851	23.289

3. Definición de Objetivos

3.1. Objetivo General

Desarrollar un método que permita la clasificación automática de textos, disponibles en múltiples idiomas, en categorías declaradas previamente en algún idioma predefinido.

3.2. Objetivos Específicos

- Definir la representación y procesamiento que se le harán a los documentos.
- Definir el algoritmo clasificador que se utilizará para la categorización de los textos.
- Definir los criterios y medidas de la evaluación de las pruebas.
- Realizar pruebas al método elegido
- Realizar análisis sobre las pruebas realizadas.

4. Estado del Arte

Se presentan a continuación algunas de las técnicas necesarias para poder llevar a cabo la clasificación de documentos, que servirán de guía a lo largo del proyecto. Una vez explicadas, se darán a conocer los pasos a seguir para la elaboración del método, exponiendo las distintas tendencias documentadas investigadas.

4.1. Técnicas de traducción automática

Los sistemas de traducción automática se pueden clasificar entre dos grandes grupos: los que se basan en reglas lingüísticas por una parte, y los que utilizan corpus textuales por otra.

4.1.1. Traducción automática basada en reglas

Para poder convertir algún texto desde un lenguaje L_i a un lenguaje L_j, con i distinto de j, existen tres distintas técnicas: traducción directa, enfoque interlingua y el enfoque por transferencia.

a) **Traducción directa (traducción binaria):** En este enfoque el sistema traductor esta de diseñado para satisfacer los detalles particulares para un determinado par lingüístico. Esto es que el lenguaje fuente es analizado para construir representaciones únicas adecuadas para el lenguaje objetivo. Usualmente estos sistemas consisten en un gran diccionario bilingüe.

b) **Enfoque interlingua:** Se asume que es posible convertir el lenguaje fuente a una representación semántica-sintáctica en la que son más comunes con el otro lenguaje. Entonces la traducción ocurre en 2 etapas, la traducción del lenguaje fuente al interlingua y luego la traducción desde la interlingua al lenguaje objetivo. Este lenguaje intermedio (conocido usualmente como lenguaje esperanto) es complejo de construir, pues se basa de un conjunto de primitivas semánticas, convirtiéndose en un ‘lenguaje universal’.

c) **Enfoque por transferencia:** A diferencia del enfoque anterior, este enfoque utiliza tres etapas. La primera consiste en convertir el lenguaje fuente a una representación abstracta orientada de dicho lenguaje; luego se convierte la representación en una equivalente en el lenguaje objetivo; finalmente se traduce esta representación en texto de lenguaje objetivo. Entonces los sistemas de traducción por transferencia se componen de tres tipos de diccionarios, un diccionario del lenguaje fuente que contiene detallada información morfológica, gramática y semántica, un segundo

diccionario de las mismas características, pero en el lenguaje objetivo, y finalmente un diccionario bilingüe de transferencia. En la Figura 4-1 se puede apreciar una representación gráfica de las técnicas descritas.

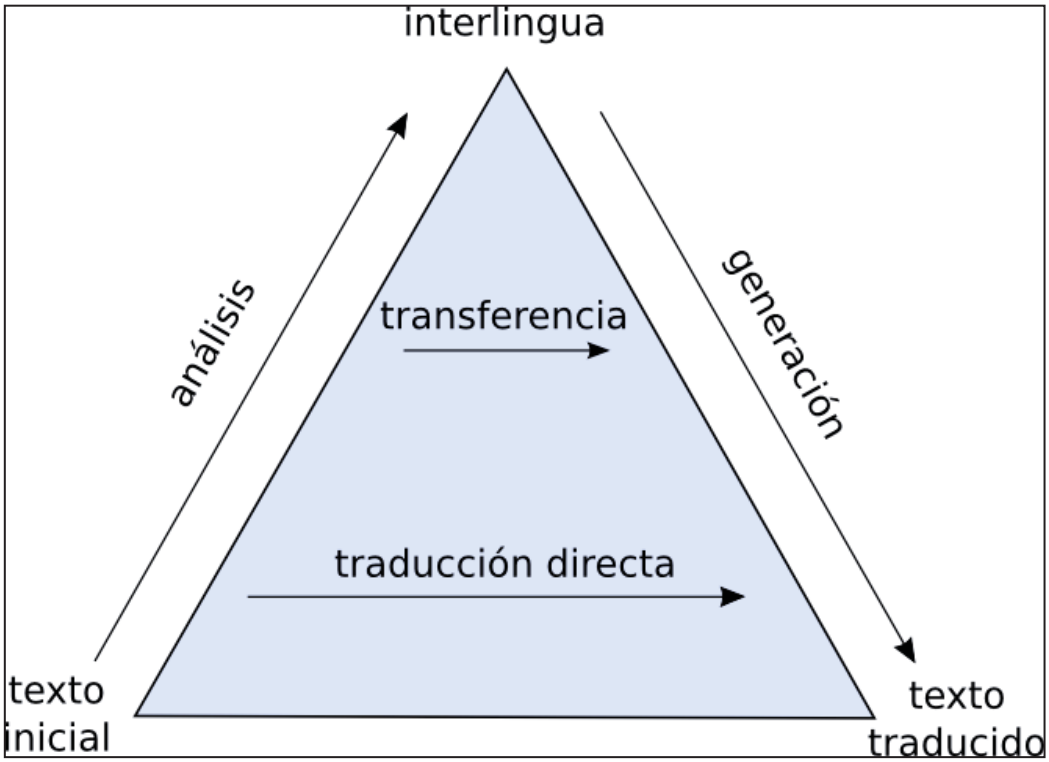

Figura 4-1: Técnicas traducción

4.1.2. Traducción automática basada en corpus

Primeramente se debe definir que es un corpus y esto es un conjunto, normalmente muy amplio, de ejemplos reales de uso de una lengua. Estos ejemplos pueden ser textos (típicamente), o muestras orales (normalmente transcritas). Entonces este tipo de traducción se basa en un corpus lingüístico y en el análisis de éste. Destacan dos maneras populares de traducción en corpus.

a) **Estadístico:** El objetivo de la traducción automática estadística, es generar traducciones a partir de métodos estadísticos basados en corpus de textos bilingües.

b) **Basados en ejemplos:** Se caracteriza por el uso de un corpus bilingüe como principal fuente de conocimiento en tiempo real. Es esencialmente una traducción por analogía y puede ser interpretada como una implementación del razonamiento por casos base

empleado en el aprendizaje automático, que consiste en la resolución de un problema basándose en la solución de problemas similares.

4.2. Técnicas de Entrenamiento de documentos

En la categorización de textos bilingües es posible encontrar 3 escenarios para realizar dicha clasificación [1]:

4.2.1. Entrenamiento poli-lingüístico

En este escenario, al sistema se le enseña usando textos de entrenamiento de ambos lenguajes. Luego es construido un único gran clasificador usando el conjunto de documentos de entrenamiento etiquetados. De esta manera, esta técnica no necesita de estrategias de traducción, por lo que no presenta distorsión o pérdida de información.

4.2.2. Entrenamiento Lingo-Transversal

El sistema realiza el entrenamiento de las etiquetas sólo en el lenguaje pivote para luego, una vez terminado, clasificar el otro idioma. Este escenario es el que se pretende tomar para el desarrollo del presente proyecto. Existe un problema con la traducción que puede ser cubierto de dos maneras:

1. Traducción del conjunto de Entrenamiento: El conjunto de documentos etiquetados es traducido al lenguaje objetivo. Luego se entrena el clasificador para dicho lenguaje. Entonces el entrenamiento se convierte en uno Poli-lingüístico.

2. Traducción del conjunto de Prueba: El Conjunto de documentos no etiquetados es traducido al lenguaje pivote. Luego estos documentos son entrenados con el conjunto de documentos de entrenamiento. De esta manera la categorización Multi-Lenguaje se convierte en uno Mono-Lenguaje

4.2.3. Lenguaje Esperanto

Este enfoque usa un lenguaje de referencia universal al cual todos los documentos son traducidos. Dicho lenguaje universal debería contener todas las propiedades de los lenguajes de interés y estar organizados en una manera semántica.

4.3. Representación de Documentos

Los textos no pueden ser directamente interpretados por un clasificador o por un algoritmo clasificador. Es por esto que se debe realizar una etapa de indexación que mapea un texto d_j en una representación compacta de su contenido, el cual es aplicado al grupo de entrenamiento, validación y testing. La manera más convencional según la literatura, es la representación por medio de un vector de pesos de los términos dj = $\{w_{1j},\ldots, w_{|T|j}\}$, donde T representa el conjunto de términos que se presentan al menos una vez dentro del documento y el peso w_{ij} representa cuanto aporta en el documento. En la Figura 4-2 se puede apreciar gráficamente lo explicado, asumiendo que el peso de un término equivale a la cantidad de veces que se encuentra repetido en un texto.

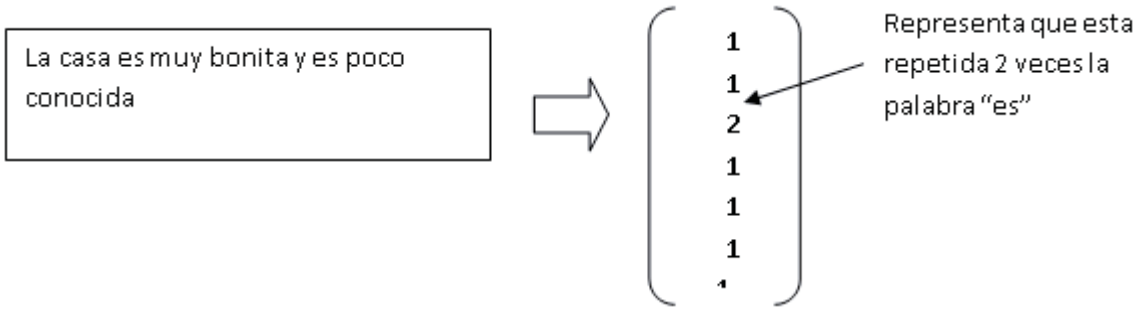

Figura 4-2: Representación vectorial de un texto

Primeramente se debe definir qué es un término para la representación. Acorde a la literatura [2], destacan dos maneras de representación: bolsa de palabras y por frases. La primera hace alusión a que cada palabra en un documento es tomada individualmente; la segunda toma una secuencia de palabras que estadísticamente tiene un significado. En este proyecto se utilizará la primera manera pues acorde a [2] no presenta mayor efectividad en comparación con otros métodos.

Luego se debe declarar los pesos de la representación anteriormente elegida. Lo más utilizado es representar que los pesos fluctúen entre los valores 0 y 1. En algunos casos se utiliza una representación binaria, es decir, el peso es 0 ó 1. El presente proyecto utilizará la primera manera.

En la literatura existen distintas técnicas de representación de textos. En el presente reporte se harán uso de 3 técnicas:

1. Frecuencia: Corresponde a la frecuencia absoluta de un término t en un documento D.

$$w(Di, tj) = fij \quad (4.3.1.1)$$

2. tf: Corresponde al valor de la frecuencia f_{ij} normalizada por el tamaño del documento.

$$w(Di, tj) = fij \,/\, |Di| \quad (4.3.1.2)$$

3. tf-idf: Corresponde a la frecuencia normalizara multiplicada por la frecuencia inversa de un término en la colección N (A más documentos donde aparezca un término t, menos representativo es)

$$\mathrm{w(Di, tj)} = (\,\mathrm{fij} \,/\, |\mathrm{Di}|\,) * -\log(\,\mathrm{nj} \,/\, \mathrm{N}\,) w(Di, tj) = fij \,/\, |Di| \quad (4.3.1.3)$$

donde n_j corresponde al número de documentos que contiene al termino t.

4.3.1. Reducción de dimensionalidad

El procesamiento de los vectores generados como ha sido mencionado, provoca una carga considerable al tratarse de vectores que superan las miles de dimensiones. Es por esto que antes de aplicar las técnicas que serán detalladas más adelante, es necesario reducir la dimensionalidad de estos. Matemáticamente es reducir el espacio vectorial de |T| a |T'|, donde |T'| << |T|.

Además de mejorar el performance de los algoritmos clasificadores, se logra reducir el sobre ajustamiento, que es el fenómeno de ajustar demasiado un clasificador con un grupo de prueba, que no será capaz de clasificar correctamente para otro grupo de prueba. Aun así no se debe caer en el otro extremo de reducir demasiado la dimensionalidad pues se puede caer en el error de perder el significado del texto.

En este proyecto se hará uso de dos técnicas para reducir el espacio vectorial: extracción de términos y el uso de Wordnet. La primera técnica consiste en eliminar las palabras funcionales tales como artículos, preposiciones y conjunciones. La segunda técnica hace uso de la herramienta Wordnet (ver apartado 6.3) y gracias a la asociación de hiperónimos u homónimos es posible calcular la distancia semántica entre dos conceptos. Aquellos que su distancia sea menor a un cierto umbral, podrá entonces ser asociada a una palabra ya existente en el vector, es decir, se aumenta el peso de la palabra en el vector en vez de crear una nueva dimensión al vector.

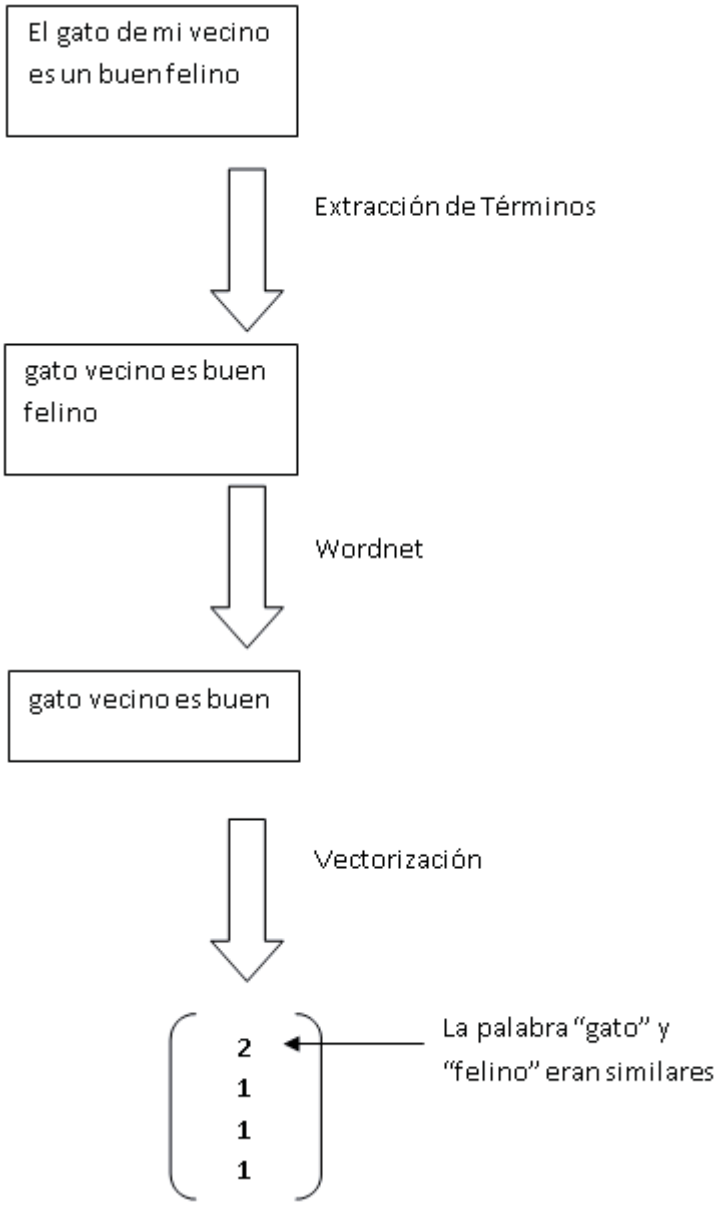

Figura 4-3: Ejemplo de reducción de dimensionalidad

En la Figura 4-3 se aprecia que el vector se reduce considerablemente, pues el texto original tiene 9 términos, lo que daría como resultado un vector de 9 dimensiones, en cambio, luego de la aplicación de la reducción, el vector se reduce a 4 dimensiones.

4.4. División de documentos

Para la realización del entrenamiento y las pruebas, es necesario la disponibilidad de un corpus inicial $\Omega = \{d_1, \dots, d_{\Omega}\} \subset D$ de documentos preclasificados en $C = \{c1, \dots, c_C\}$. En investigaciones, es siempre deseable evaluar la efectividad del clasificador Φ construido. Para ello es necesario dividir el corpus en 2 partes no necesariamente del mismo tamaño: un grupo de entrenamiento (y validación) y otro de pruebas.

4.4.1. Grupo de Entrenamiento y Validación (TV)

Se define $TV = \{d_1, \dots, d_{TV}\}$. El clasificador Φ para las categorías $C = \{c1, \dots, c_C\}$ es deductivamente construido, observando las características de los documentos.

4.4.2. Grupo de Pruebas (Te)

Se define $Te = \{d_{TV+1}, \dots, d_{\Omega}\}$. Es usado para las pruebas de efectividad de los clasificadores. Cada $d_j \in Te$ sirve de entrada para $\Phi(d_j, c_i)$ que es comparado con $\Phi(d_j, c_i)$.

Los documentos en Te no pueden participar de ninguna manera en la construcción inductiva de los clasificadores; si esta condición no fuese satisfecha, los resultados experimentales serian poco realistas y la evaluación no tendría carácter científico.

4.5. Fase de Entrenamiento

El primer tema que se debe contemplar en esta etapa es aquel referente a la representación de los documentos, de tal manera que se retenga la mayor cantidad de información y que facilite la manipulación del contenido por la máquina.

La representación más usada es la *bolsa de palabras*, la cual consiste en almacenar todas las palabras presentes en algún documento junto con su número de ocurrencias. El problema de esta representación es que no considera las relaciones entre las palabras, por lo que los algoritmos de aprendizaje se ven restringidos a sólo detectar patrones en la terminología, mientras que los patrones conceptuales son ignorados.

En este proyecto se considera la factibilidad de utilizar Wordnet pues permite crear perfiles de categorías que contienen los conceptos (synsets) dándole mayor representatividad. Para poder llevar a cabo lo anterior es necesario cumplir con los siguientes pasos:

1. Mapear los términos de los documentos en synsets usando Wordnet.
2. Capturar las relaciones entre los synsets.
3. Utilizando algún método de selección de rasgos, seleccionar los conceptos característicos que formarán los perfiles conceptuales de categoría.

4.5.1. Mapear términos en synsets

Una de las técnicas más recurrentes para representar documentos es a través de vectores. El problema de estos es la dimensionalidad que alcanzan, usualmente más de 10000 palabras [1]. Para lidiar con esta complejidad se acude a la reducción de dimensionalidad, en la que un espacio vectorial T es comprimido a $T' \ll T$. Esta reducción además subsana el problema de sobre ajustamiento. El sobre ajustamiento se detecta cuando un clasificador se dice que es bueno categorizando la data que utilizaron para entrenarse, pero malos para clasificar nueva data. Esta reducción se debe hacer con precaución para no remover información relevante.

En algunos casos, términos distintos pueden representar el mismo concepto y en otros casos cuando no es así. En estos casos el término puede ser remplazado por un concepto de mayor nivel (recordando que Wordnet trabaja jerárquicamente) sin afectar el desempeño. Es por lo anterior, que mapear los términos en conceptos permite una reducción del espacio vector.

En los casos en que una palabra tenga varios significados, implicaría que la palabra puede ser mapeada en más de un synset provocando ruido en la representación

(desembocando finalmente en perdida de información). Para estos casos de necesidad de desambiguación, Wordnet ofrece retornar una lista ordenada de synsets de cada término, que representa que tan comúnmente representa el término el synset ordenados de mayor a menor.

Actualmente existe un gran campo de investigación sobre la desambiguación de palabras, pero este proyecto busca ahondar en los conocimientos de la clasificación de textos bilingües. Es por esto que se elige una estrategia simple de desambiguación que consiste en elegir aquel significado más común de cada término como el más apropiado. Matemáticamente, la frecuencia de los synsets (sf), queda representado de la siguiente manera:

$$sf\ c_i, s\ = tf(c_i,\ t \in T\ \ primero\ Ref_s\ t\ \ = s\}) \qquad (4.5.1.1)$$

Dónde:

- c_i: la categoría i-ésima
- $tf(c_i, T')$: la suma de las frecuencias de todos los términos $t \in T$ en los documentos entrenados de la categoría c_i.
- $Ref_s\ t$: el conjunto de todos los synsets asignados al termino t en Wordnet.

4.5.2. Capturar las relaciones entre los synsets

Luego de mapear los términos en synsets, se pasa a utilizar las jerarquías de Wordnet que permite capturar relaciones entre los synsets (relación de hiperónimo). La frecuencia de los synsets será actualizado según la siguiente ecuación:

$$sf\ c_i, s\ = \sum_{b \in H(s)} sf\ c_i, s \qquad (4.5.2.1)$$

Dónde:

- c_i: la categoría i-ésima
- b y s son synsets
- $H\ s\ contiene\ los\ synsets\ que\ tienen\ el\ synset\ s\ como\ hiperónimo.$

4.5.3. Crear perfiles conceptuales de categorías

Para crear los perfiles conceptuales de categorías se hará uso del clasificador Bayesiano ingenuo, que es un clasificador probabilístico basado en el teorema de Bayes y algunas hipótesis simplificadoras adicionales. Es a causa de estas simplificaciones, que se suelen resumir en la hipótesis de independencia entre las variables predictoras, que recibe el apelativo de ingenuo.

El clasificador Bayesiano se define como sigue:

$$P(A|B) = (P(B|A) * P(A)) / P(B) \qquad (4.5.3.1)$$

La probabilidad de que A ocurra dado B está determinado por la probabilidad de B dado A, la probabilidad de que A ocurra y la probabilidad de que B ocurra. La regla de Bayes permite el cálculo de la probabilidad de que un evento A, dado que B ha ocurrido. Esto es usado en la clasificación de textos para determinar la probabilidad que el documento B es del tipo A sólo mirando la frecuencia de las palabras en el documento.

Una categoría es representada por una colección de palabras; la frecuencia es el número de veces que cada palabra ha sido detectada en el documento usado para entrenar el clasificador.

Suponiendo que hay n categorías C_0 hasta C_{n-1}. Determinar a qué categoría un documento D es el más asociable, significa calcular la probabilidad que ese documento D esté en la categoría C_i, es decir $P(C_i|D)$, para cada categoría C_i. Usando la regla de Bayes se puede calcular este último:

$$P(C_i|D) = (P(D|C_i) * P(C_i)) / P(D) \qquad (4.5.3.2)$$

$P(C_i|D)$ es la probabilidad de que el documento D esté en la categoría C_i; esto es, la probabilidad que dado un conjunto de palabras en D, éstas aparezcan en la categoría C_i. $P(D|C_i)$ es la probabilidad que para una categoria dada C_i, las palabras en D aparezcan en dicha categoría. $P(C_i)$ es la probabilidad de una categoría dada; esto es, la probabilidad de que un documento esté en la categoría C_i sin considerar sus contenidos. P(D) es la probabilidad de que ese específico documento ocurra.

4.6. Fase de clasificación

Esta fase consiste en usar los perfiles de categorías conceptuales para categorizar los documentos no etiquetados en los dos idiomas. Los pasos a seguir para conseguir esto son:

1. Traducir los documentos al lenguaje pivote y generar un vector conceptual.
2. Pesar los perfiles de categorías conceptuales y el vector conceptual de los documentos no etiquetados.
3. Calcular la distancia entre el vector conceptual y todos los perfiles de categorías conceptuales.

4.6.1. Traducción y generación de un vector conceptual

El objetivo de la traducción del texto a ser clasificado al lenguaje pivote no es producir una traducción semánticamente exacta, sino que proveer al texto de una aseguración de calidad suficiente. Los resultados dependerán del traductor utilizado. Para ello se hará uso de la herramienta llamada Apertium (ver apartado 6)

Luego de traducir, se debe hacer uso de Wordnet para la generación del vector conceptual de cada documento, mapeando los términos en synsets y capturando la relación entre estos.

4.6.2. Pesaje de las categorías

Esta etapa consiste en pesar los perfiles de categorías conceptuales y el vector conceptual de los documentos no etiquetados. Cada peso *w(s,c)* expresa la importancia del synset s en el vector de c con respecto a la frecuencia en todos los documentos de entrenamiento. Para realizar este proceso se utilizará simplemente la frecuencia de las palabras dentro de alguna categoría.

$$w\ s_k, c_i\ = tf(s_k, c_i) \qquad (4.6.2.1)$$

Dónde:

$tf(s_k, c_i)$: denota el número de veces que el synset s_k ocurre en la categoría c_i

4.6.3. Determinación de la categoría de un documento

Para poder determinar a qué categoría pertenece un documento, se debe calcular la probabilidad de que este pertenezca a una de las categorías predefinidas y luego determinar cuál de todos estos valores es el más alto, es decir, a qué categoría pertenece con mayor probabilidad.

Para calcular en qué categoría debe ir el documento D, se necesita calcular $P(C_i|D)$ para cada categoría y encontrar la probabilidad más alta. Cada uno de esos cálculos involucra el valor no conocido y fijo P(D), por lo que puede ser ignorado y se calcula de la siguiente manera:

$$P(C_i|D) = (P(D|C_i) * P(C_i)) / P(D) \quad (4.6.3.1)$$

Se puede omitir con seguridad porque interesa el valor relativo, no el valor absoluto, de $P(C_i|D)$, y P(D) simplemente actúa como un factor de escala en $P(C_i|D)$.

D es separado en un set de palabras dentro del documento, llamado W_0 a través de W_{m-1}. Para calcular $P(C_i|D)$, hay que calcular el producto de las probabilidades de cada palabra; esto es, la probabilidad de que cada palabra aparezca en C_i. Aquí está el paso "ingenuo": Se supone que las palabras aparecen de forma independiente de otras palabras (que claramente no es cierto para la mayoría de idiomas) y $P(D|C_i)$ es el simple producto de las probabilidades de cada palabra:

$$P(D|C_i) = P(W_0|C_i) * P(W_1|C_i) * \ldots * P(W_m - 1|C_i) \quad (4.6.3.2)$$

Para cualquier categoría, $P(W_j|C_i)$ se calcula como el número de veces que aparece W_j en C_i dividido por el número total de palabras en Ci. $P(C_i)$ es calculado como el número total de palabras en C_i dividido por el número total de palabras en todas las categorías juntas. Por lo tanto, $P(C_i|D)$ es:

$$P(W_0|C_i) * P(W_1|C_i) * \ldots * P(W_m - 1|C_i) * P(C_i) \quad (4.6.3.3)$$

Para cada categoría, escogiendo la mayor, se determina la categoría para el documento D.

Una crítica común de los clasificadores bayesianos ingenuos de texto es que hacen la "ingenua" suposición de que las palabras son independientes el uno del otro y, por lo tanto, estos modelos son menos precisos que otros más complejos.

5. Método

Para poder construir el método de clasificación de textos bilingües, es necesario seguir una serie de pasos fundamentales, los cuales serán divididos en 3 grandes faces: preparación del corpus, fase de entrenamiento y fase de clasificación.

En la primera fase se deben de preparar los documentos para que puedan ser procesados por el algoritmo. La segunda fase consiste en crear los perfiles de categorías conceptuales, los cuales contienen los conceptos que mejor la caracterizan con respecto al resto de las categorías. La última fase consiste en determinar la pertenencia de un documento d_j a una categoría c_i. En la Figura 5-1 se muestra gráficamente lo recién descrito.

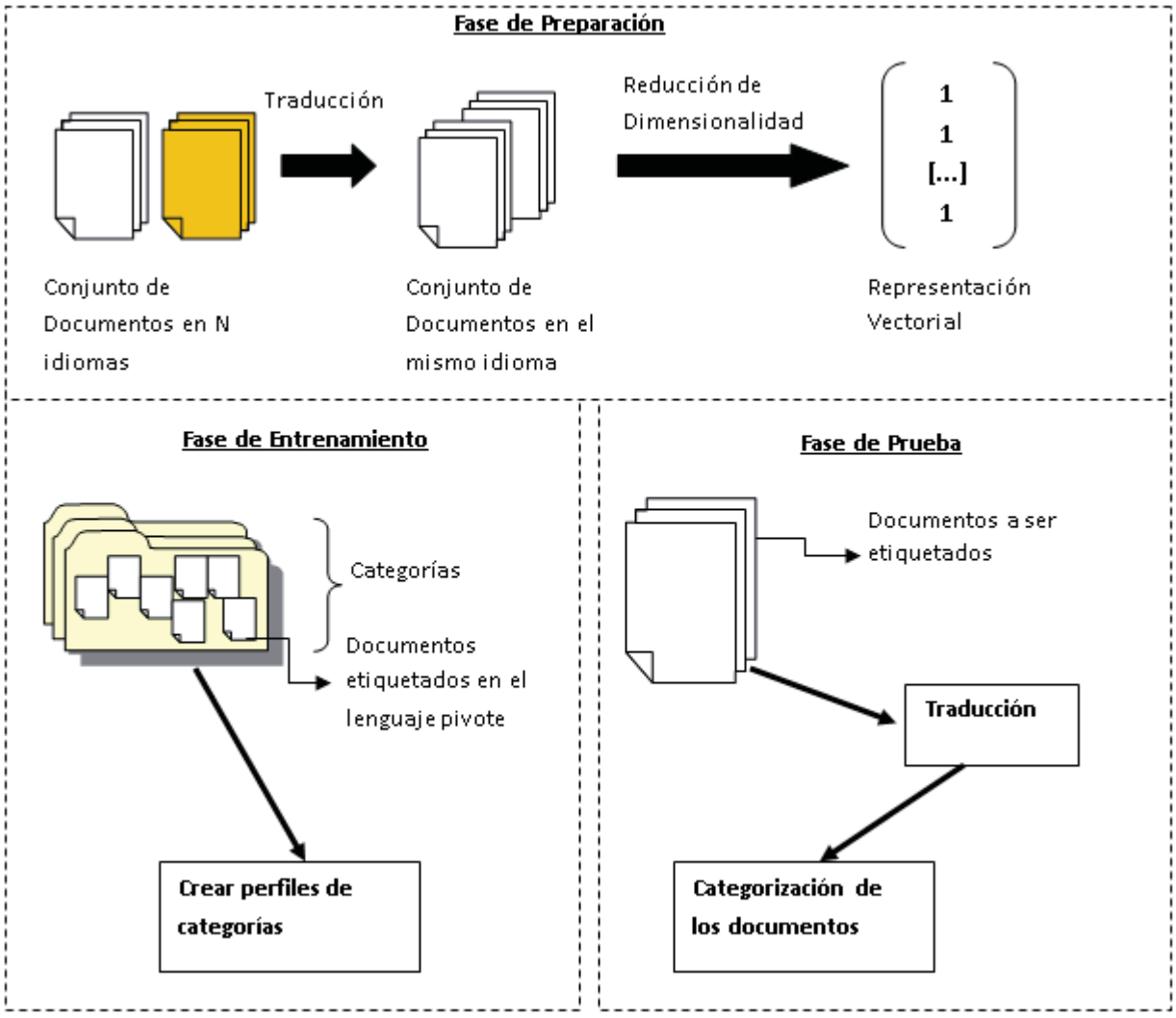

Figura 5-1: Método Propuesto

5.1. Fase de Preparación

En esta etapa se deben preparar los textos para que puedan ser procesados más adelante. Los documentos en un principio se encuentran preclasificados (ver apartado 6.1) para más detalles del corpus elegido), pero en distintos idiomas y estos para poder ser procesados deben tener una representación vectorial.

Para lograr esta representación se deben cumplir las siguientes actividades:

a) **Traducción:** Los documentos que no se encuentran en el idioma pivote (el inglés) deben de ser traducidos de manera automática. Para esto se utilizará la herramienta Apertium (ver apartado 6.2), el cual se encargara de esta tarea.
b) **Extracción de términos:** A los documentos se les quitan aquellas palabras funcionales tales como artículos, preposiciones y conjunciones para reducir el espacio vectorial a construir.
c) **Construcción del vector conceptual y mapeo de synsets:** El vector es construido con el peso de cada término en algún documento (ver apartado 4.5 para más detalles).

5.2. Fase de Entrenamiento

Tal como lo explica el apartado 4.5 los documentos ya se encuentran pre-clasificados y en notación vectorial y entonces se deben crear los perfiles de las categorías, es decir crear el vector que representa a la categoría.

Un paso entre la fase preparación y de entrenamiento es el mapeo con el fin de disminuir la dimensionalidad del vector conceptual. Para realizar lo anterior, se verifica si los términos son similares a alguno en el vector. Para definir si el término es ‘similar’ a otro se utiliza Wordnet y las capacidades de determinar si existe el hiperónimo de cada palabra, En el caso de que exista el hiperónimo, el peso de dicho término es aumentado por el peso del padre.

Para el entrenamiento de la máquina se hará uso de la herramienta desarrollada para solucionar esta problemática (ver apartado 6.4). Esta aplicación estará configurada para que haga el entrenamiento con el algoritmo de bayesiano ingenuo, el cual es un clasificador probabilístico cuyo origen es la aplicación del teorema de Bayes de formula

$$P(c_i \mid \vec{d}_j) = \frac{P(c_i)P(\vec{d}_j \mid c_i)}{P(\vec{d}_j)} \qquad (5.2.1)$$

con $P(\vec{d}_j)$ la probabilidad de un documento elegido aleatoriamente tenga como representación el vector $\vec{d}_j$, y $P(c_i)$ la probabilidad de que un documento aleatoriamente elegido pertenezca a la categoría c_i. El apellido ingenuo hace referencia que hay independencia entre las variables predictivas. Entonces de la misma manera que es explicado en el apartado 4.5.3, se crean los perfiles de categorías, los cuales son almacenados como un objeto serializado de Java en un archivo, lo que permitirá realizar distintas clasificaciones de nuevos documentos bajo un mismo clasificador en un futuro.

5.3. Fase de Prueba

En esta etapa los documentos sin clasificar son sometidos bajo el algoritmo de clasificación, utilizando el mismo método que en la fase de entrenamiento (algoritmo de bayes ingenuo). Para realizar esta actividad automáticamente, también se hace uso de la herramienta desarrollada, la cual se encargara de clasificar los nuevos documentos acorde a aquellos que ya fueron entrenados.

En la práctica, el entrenamiento y las pruebas se realizan en un mismo comando, en el cual se define que porción del corpus servirá como entrenamiento, siendo el complemento para las pruebas. Finalmente, la aplicación entregará la clasificación de cada documento nuevo, terminando así el proceso completo de clasificación de textos bilingües.

5.4. Evaluación

Para evaluar el desempeño de los clasificadores se utilizaran cuatro indicadores: precisión, recall, exactitud y Medición-F.

5.4.1. Precisión

La precisión de una clase es el número de verdaderos-positivos divididos por la suma de los falsos-positivos con los verdaderos-positivos. Verdaderos-positivos se refiere a aquellos documentos que han sido correctamente categorizados a la clase correcta; Falso positivo corresponde a los elementos incorrectamente clasificados a la clase correcta. Una precisión de valor 1.0 quiere decir que para una clase C todos los elementos clasificados en la clase C, ciertamente pertenecen a la categoría C (pero no dice nada acerca de aquellos elementos que no fueron correctamente clasificados).

$$\text{Precision} = \frac{tp}{tp + fp} \qquad (5.5.1.1)$$

5.4.2. Recall

Recall es definido como el número de verdaderos-positivos dividido por la sumad e verdaderos-positivos y falsos-negativos). Los falsos-negativos son aquellos elementos que no fueron clasificados a la clase correcta. Un Recall de valor 1.0 quiere decir que cada elemento de la clase C fue categorizado como perteneciente a C (pero no dice nada acerca de aquellos elementos que fueron incorrectamente clasificados en C).

$$\text{Recall} = \frac{tp}{tp + fn} \tag{5.5.2.1}$$

5.4.3. Exactitud (accuracy)

Corresponde a qué tan lejano el resultado obtenido es respecto a lo deseado.

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn} \tag{5.5.3.1}$$

5.4.4. Medición-F

Es una medida que combina la precisión y el recall. El factor beta que aparece en la formula indica cuantas veces se le da importancia al recall sobre la precisión.

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}} \tag{5.5.4.1}$$

6. Herramientas

A continuación se detallarán las distintas herramientas a ser utilizadas en el presente proyecto.

6.1. Reuters

Reuters es el corpus que contiene los documentos clasificados manualmente en distintas categorías. De los tipos de clasificaciones (ver punto 2.6), se eligió la clasificación por Tópicos, pues asegura que cada documento tenga al menos una categoría asociada.

Las categorías en las cuales fueron asociadas son las siguientes:

- corporate: Corporaciones/Industrias (Corporate/Industrial)
- economics: Economía (Economy)
- government: Guvernamental/Social (Government/Social)
- markets: Mercado

Originalmente se partió con RCV-v1, pero se generó el RCV-v2 que es una agrupación con menos errores y que cumple con la regla que al menos cada documento tenga una categoría y que estos no se encuentren repetidos.

Para poder realizar la clasificación multilingüe, a su vez se necesito de un corpus en varios idiomas. Es por esto que se hizo uso también de Reuters multilingüe, el cual se encuentra en 10 distintos idiomas.

6.2. Traducción: Apertium

Apertium [9] es un sistema de traducción automática de código abierto que permite la traducción entre una variedad de idiomas, basándose en la traducción por reglas.

Las traducciones basadas en reglas consisten en un gran número de reglas lingüísticas que representan a un cierto lenguaje (en formato XML), que luego son convertidas en el lenguaje objetivo. Para convertir desde el lenguaje fuente al lenguaje objetivo se tiene además un extenso léxico de información morfológica, sintáctica y semántica, y un largo conjunto de reglas de conversión.

Actualmente Apertium dispone de los siguientes pares de traducción:

- español-catalán
- español-esperanto
- español-euskera
- español-francés
- español-gallego
- español-inglés
- español-occitano
- español-portugués
- español-rumano
- catalán-esperanto
- catalán-francés
- catalán-inglés
- catalán-occitano
- catalán-portugués
- gallego-inglés
- gallego-portugués
- inglés-esperanto
- inglés-galés
- inglés-francés
- francés-bretón

Otra gran característica, es su performance: la literatura [3] indica que puede traducir miles de palabras por segundo en computadores ordinarias de escritorio.

Para trabajar a nivel de código con esta herramienta se utilizarán comandos bash bajo la plataforma Unix (usando Ubuntu 10.04). En este se le dará la ruta de que carpeta contiene

los documentos a ser traducidas, además de indicar cuál es el lenguaje fuente y el lenguaje objetivo.

6.3. Reducción de Dimensionalidad: Wordnet

Para la reducción de dimensionalidad, una técnica utilizada es la aplicación de Wordnet (ver apartado 2.4 para más detalles de Wordnet). Para poder hacer uso de Wordnet a nivel de programación, se hace uso de la librería para Python llamada NLTK [10]. NLTK proviene de las siglas del inglés *Natural Language Toolkit* y es una suite de librerías para el procesamiento del lenguaje natural.

Dentro de las características que ofrece NLTK, en el módulo del Wordnet se encuentran las siguientes:

- Sinónimos, Antónimos, Hiperónimos e Hiponimos
- Holonimos, Meronimos, Coordenadas y Similares
- Glosarios, 'Véase también', Ejemplos y Descripciones
- Distancia entre términos

Para el caso del presente proyecto se hará uso sólo del módulo de Hiperónimos, el cual será explicado más adelante.

6.4. Entrenamiento y Clasificación: Aplicación Web

Para todos los pasos del método propuesto, esta aplicación entra en juego. Ésta es la encargada de concertar la llamada a los subprocesos que permiten la exitosa ejecución de la clasificación de documentos.

1. Preparación de los documentos: En esta etapa los documentos son modificados de tal manera que para cuando se necesite de entrenar, estos se encuentren listos para ser utilizados. Entonces se encarga de la traducción de los documentos que se encuentren en otro idioma al inglés, para ser luego divididos en el grupo de entrenamiento y el otro de clasificación. Los documentos son limpiados de los *stopwords* y reordenados de tal manera que sea más sencillo la lectura de estos para su posterior procesamiento. Junto con lo anterior se limpian las carpetas que serán utilizadas en reiteradas ocasiones

2. Entrenamiento: Los documentos son leídos para rescatar las características de cada uno de ellos. Se obtienen los pesos de los términos en cada documento de cada categoría que será la fuente para la próxima clasificación. Además se aplica Wordnet una vez calculado los pesos de manera opcional.

3. Clasificación: Se recorre el o los archivos a clasificar y se calcula la probabilidad de que éste pertenezca o no a alguna categoría. De la misma manera que en el entrenamiento, se hace uso de Wordnet para encontrar relaciones entre palabras del tipo hiperónimo.

4. Evaluación: Finalmente se evalúan los puntos destacados en el apartado 5.4. Dichos resultados. Una vez realizada esta evaluación, los resultados son graficados en un ambiente web compatible con dispositivos móviles.

5. Resultados: Despliega los resultados obtenidos a partir de la clasificación y evaluación realizada a través de gráficos.

7. Implementación

Para la implementación se hicieron uso de las herramientas mencionadas en el apartado de herramientas (apartado 6). En la Figura 7-1: Diagrama de casos de uso del método propuesto. se pueden apreciar las funciones que desempeña el método para lograr el objetivo deseado.

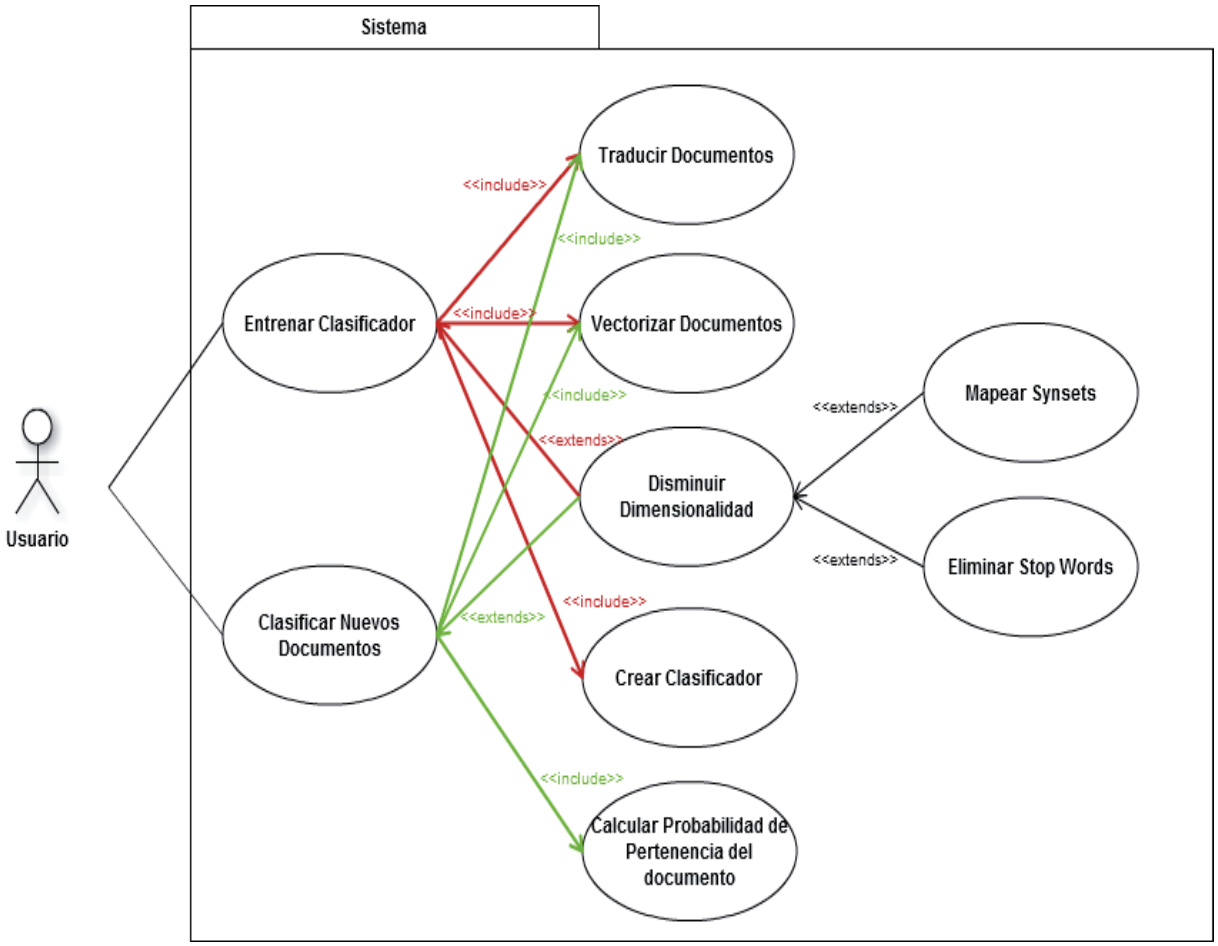

Figura 7-1: Diagrama de casos de uso del método propuesto.

De la Figura 7-1 se puede rescatar que el proceso de entrenamiento y clasificación pueden ejecutare en temporalidades distintas, es decir, que es posible que el proceso de clasificación sea realizado ya sea una vez creado el clasificador o en algún momento futuro. Esto también implica que el clasificador puede ser utilizado en más de una ocasión.

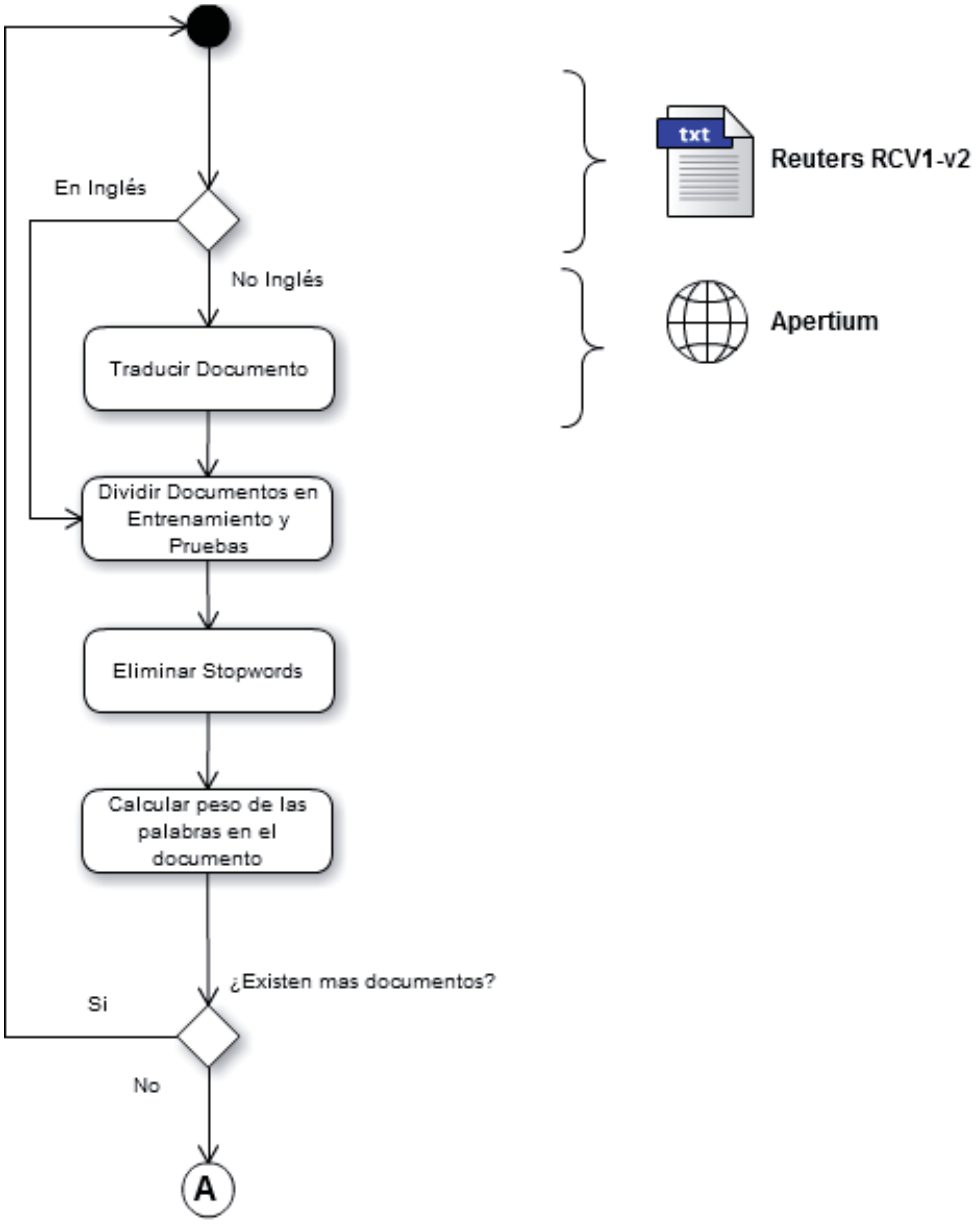

Figura 7-2: Diagrama de Actividades del proceso de preparación de documentos y entrenamiento

En la se muestra la secuencia de los pasos a seguir para poder preparar los documentos a ser clasificados o entrenados. A su vez a la derecha de la imagen se puede apreciar la herramienta que debe ser utilizada para poder realizar dicha tarea.

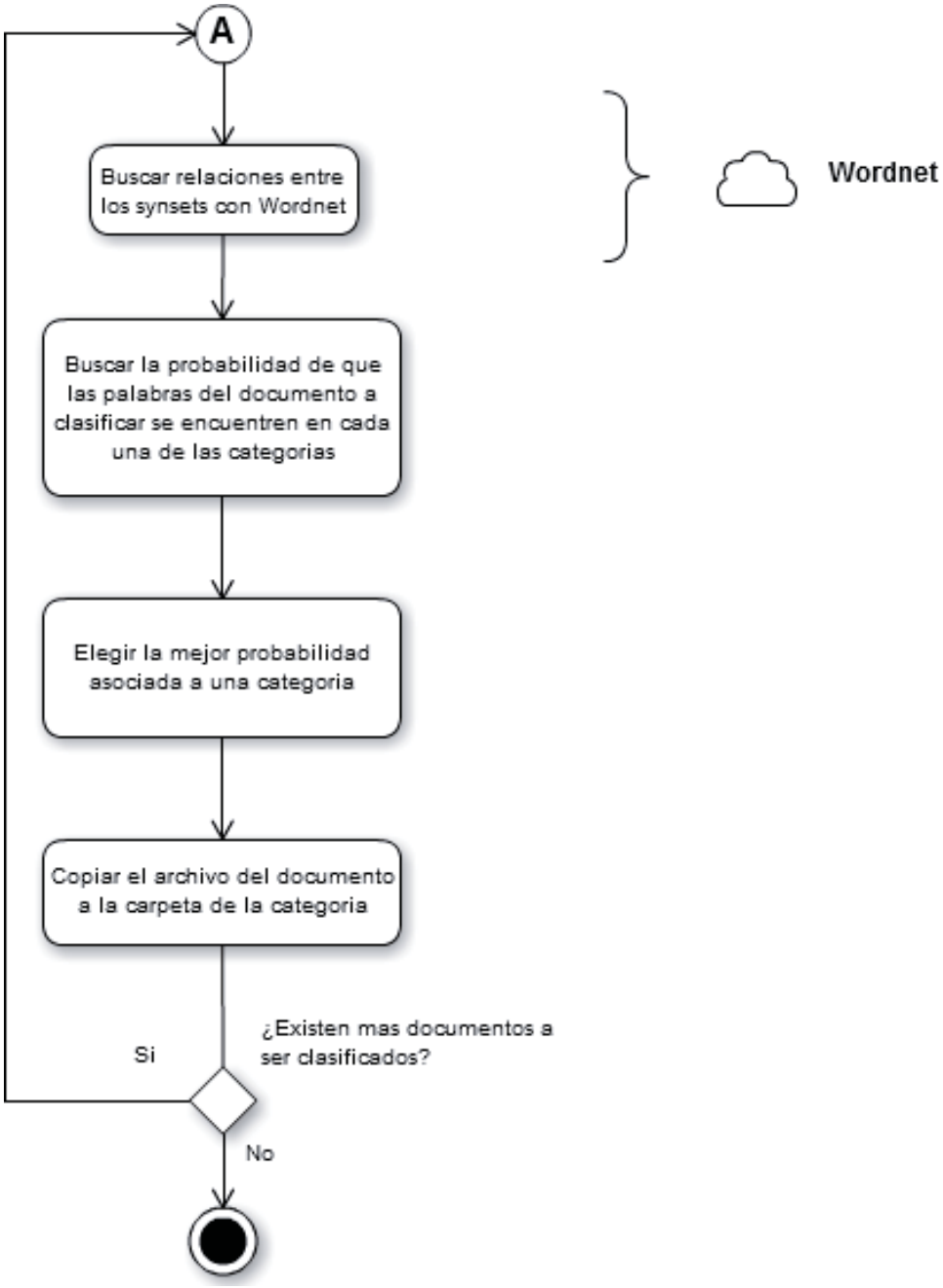

Figura 7-3: Diagrama de Actividades del proceso de clasificación de nuevos documentos.

En la Figura 7-3 se refleja el proceso de la clasificación de nuevos documentos bajo los documentos que ya fueron entrenados, previo a una tarea de preparación de los documentos.

7.1. Codificación y ejecución

El sistema fue implementado bajo el lenguaje de programación PHP5 para que funcione idealmente con interfaces web, lo cual lo hace muy versátil al momento de su utilización en una variada gama de dispositivos. El motor de clasificación fue desarrollado desde cero, lo que asegura de mejor manera que el código realice las tareas indicadas sin realizar esfuerzos extras al servidor, comprometiendo tiempo o calidad de los resultados.

La llamada a los subprocesos tales como la traducción, buscar la similitud entre palabras usando Wordnet se hacen consultando a la consola del sistema operativo del servidor de la aplicación vía comandos de consola.

La interfaz fue diseñada para que funcionase en dispositivos de escritorio y móviles gracias a la aplicación de librerías tales como *JQuery Mobile*, permitiendo a su vez un entorno amigable y de fácil uso.

7.2. Interfaz Grafica

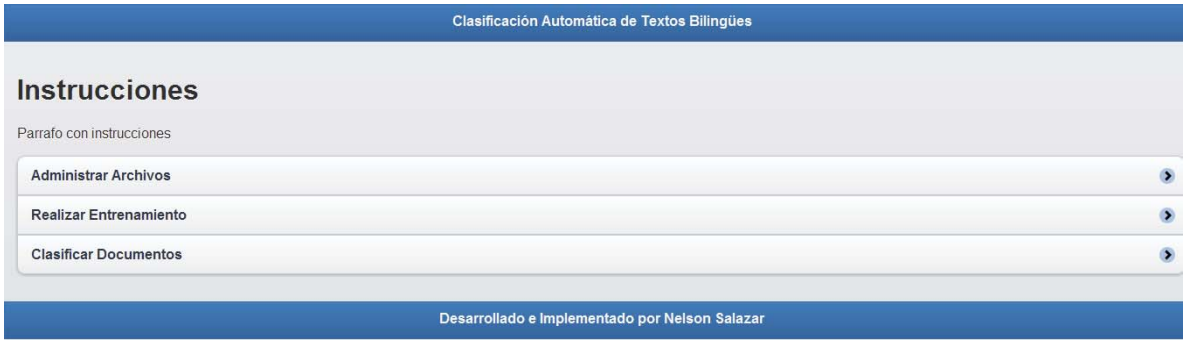

Figura 7-4: Interfaz Inicial

Figura 7-5: Interfaz Entrenamiento

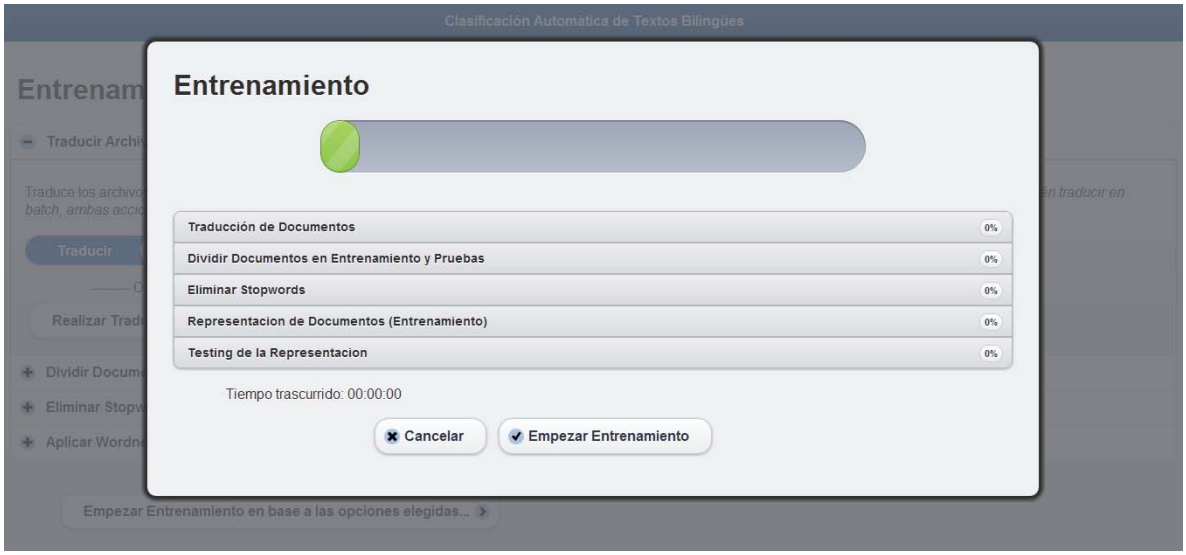

Figura 7-6: Interfaz Entrenamiento en ejecución

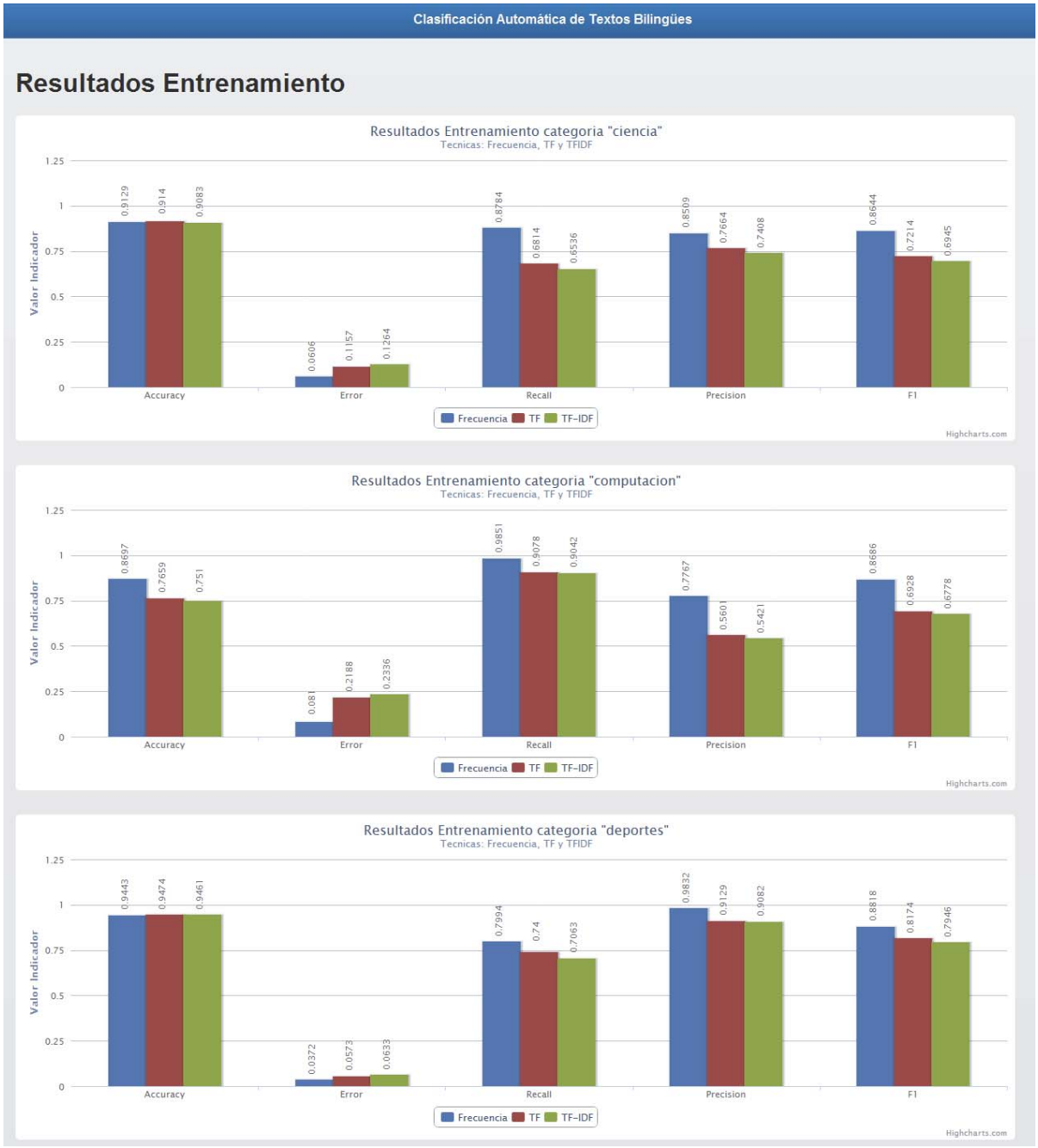

Figura 7-7: Interfaz Resultados entrenamiento

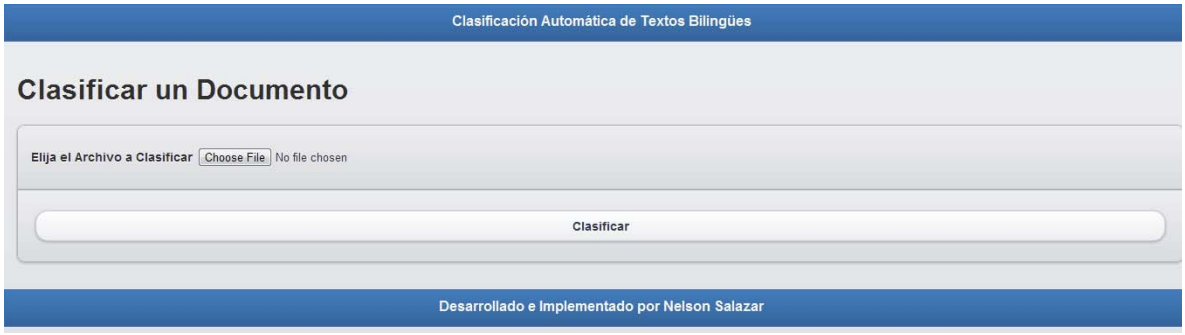

Figura 7-8: Interfaz Clasificación de 1 documento

Figura 7-9: Interfaz Resultados Clasificación de 1 documento

7.3. Servidor

El servidor en el cual el sistema fue montado posee las siguientes características:

- Sistema Operativo Servidor: Ubuntu 10.04
- Procesador Servidor: AMD Phenom(tm) II N950 Quad-Core Processor, 2100 Mhz, 4 Core(s), 4 Logical Processor(s)
- Memoria RAM Servidor: 8Gb
- Apache 2.2.14
- PHP5 2.5.8
- MySQL 14.14
- Apertium 2.0
- Wordnet 3.0
- WordNet-QueryData 1.49
- Text-Similarity 0.08
- WordNet-Similarity 2.05

8. Análisis de Resultados

Para realizar el análisis de los resultados se realizaron diversas pruebas bajo distintas condiciones con respecto a los siguientes aspectos:

- La cantidad de archivos con los que se realizaron los entrenamientos y pruebas.
- La proporción de archivos entre el entrenamiento y las pruebas.
- La inclusión o no de archivos en otros idiomas.
- La aplicación de distintas representaciones de documentos.

Cada una de estos tipos de prueba se realizó aplicando el criterio de *ceteris paribus*, es decir que si por ejemplo se modificó la proporción de archivos para el entrenamiento y pruebas, los otros aspectos quedaron constantes. La única excepción es la aplicación de distintas representaciones de documentos, en la cual para todas las pruebas se realizan los 3 tipos (frecuencia, TF, TF-IDF)

En cada iteración, los archivos seleccionados fueron escogidos aleatoriamente, provocando que cada una de éstas arroje resultados únicos.

8.1. Prueba N°1: Ratio entre Archivos de Entrenamiento y Testing variables

8.1.1. Condiciones Iniciales Constantes

- No se incluyen archivos en otros idiomas
- 1000 archivos divididos en las siguientes categorías: corporate, economics, government, markets.
- Aplicación de las 3 técnicas de representación de documentos

8.1.2. Condiciones Iniciales Variables

- 80% de los archivos son de entrenamiento; 20% de los archivos son para el testing
- 60% de los archivos son de entrenamiento; 40% de los archivos son para el testing
- 40% de los archivos son de entrenamiento; 60% de los archivos son para el testing
- 20% de los archivos son de entrenamiento; 80% de los archivos son para el testing

8.1.3. Resultados

2.1.1.1. Entrenamiento 80% / Testing 20%

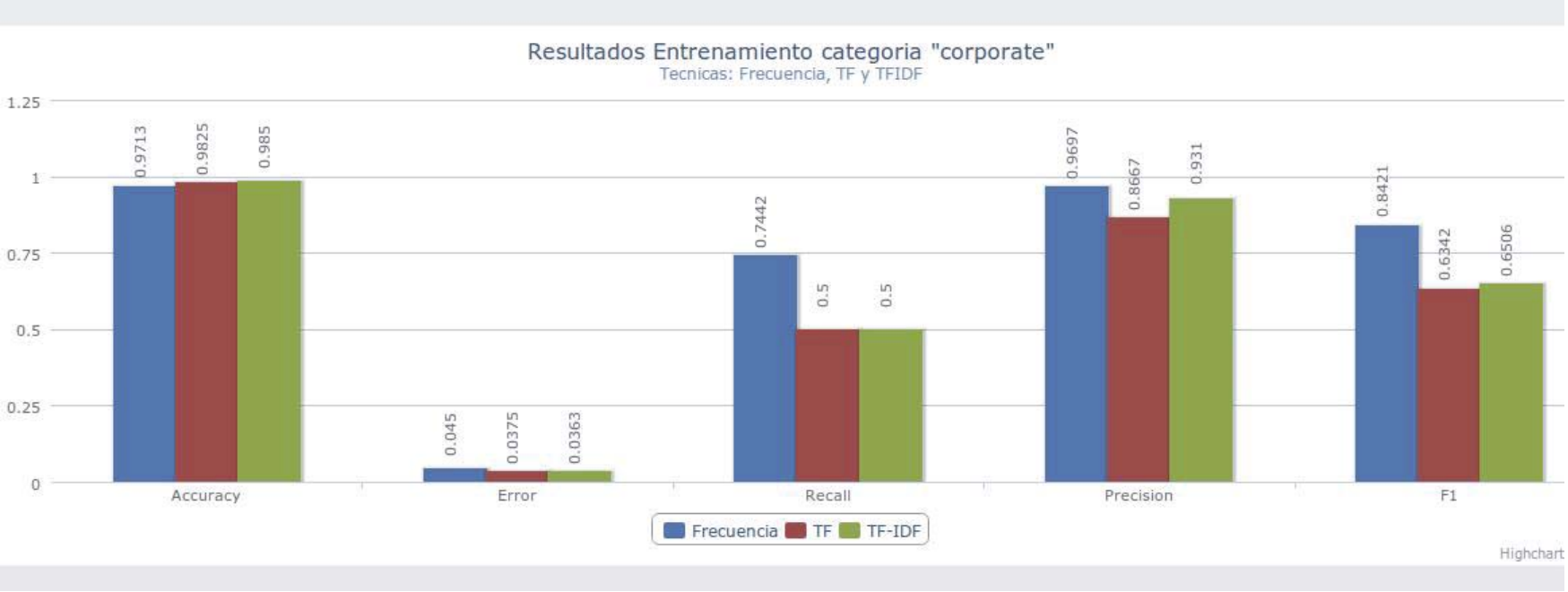

Figura 8-1: Categoría “corporate”: 80% Entrenamiento, 20% Testing

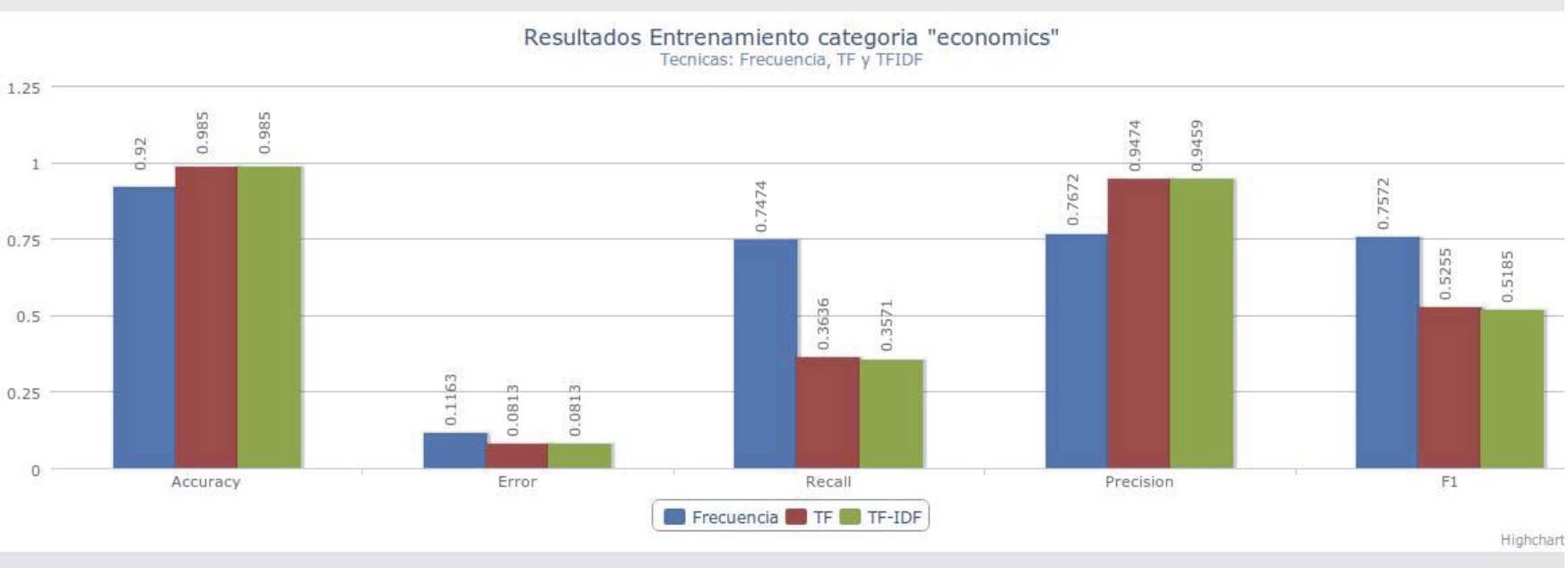

Figura 8-2: Categoría “economics”: 80% Entrenamiento, 20% Testing

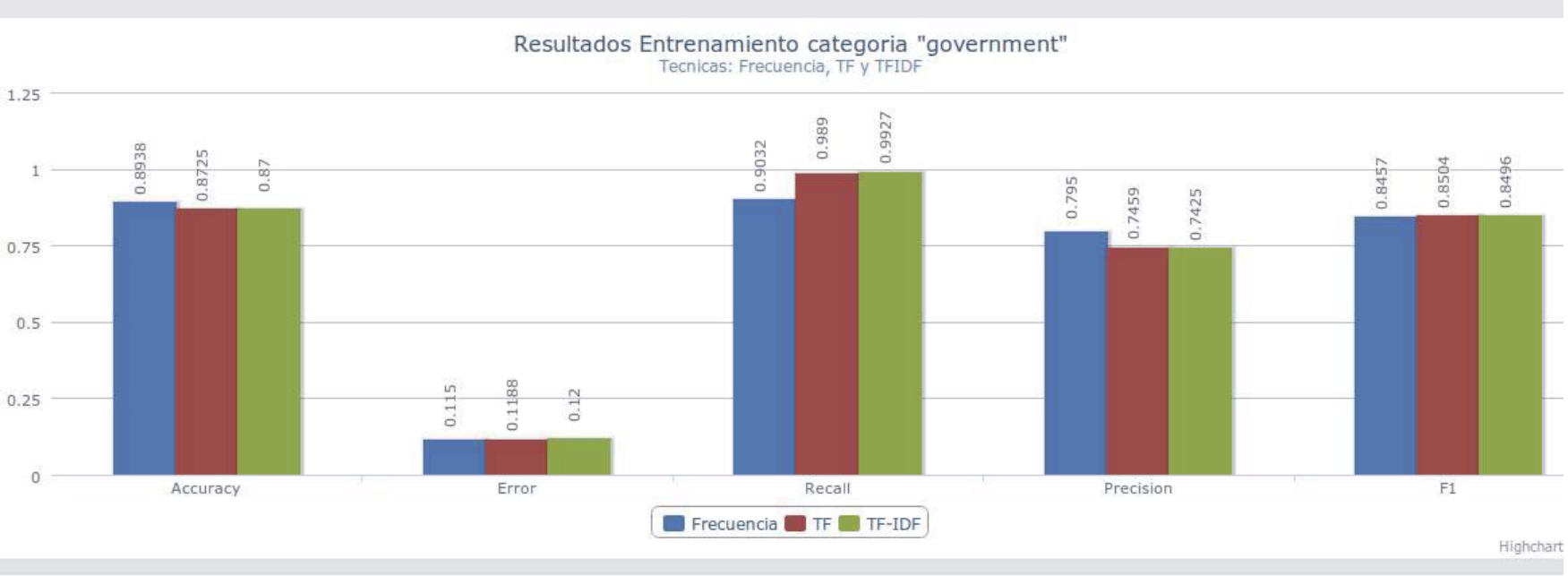

Figura 8-3: Categoría “government”: 80% Entrenamiento, 20% Testing

Figura 8-4: Categoría “markets”: 80% Entrenamiento, 20% Testing

2.1.1.2. Entrenamiento 60% / Testing 40%

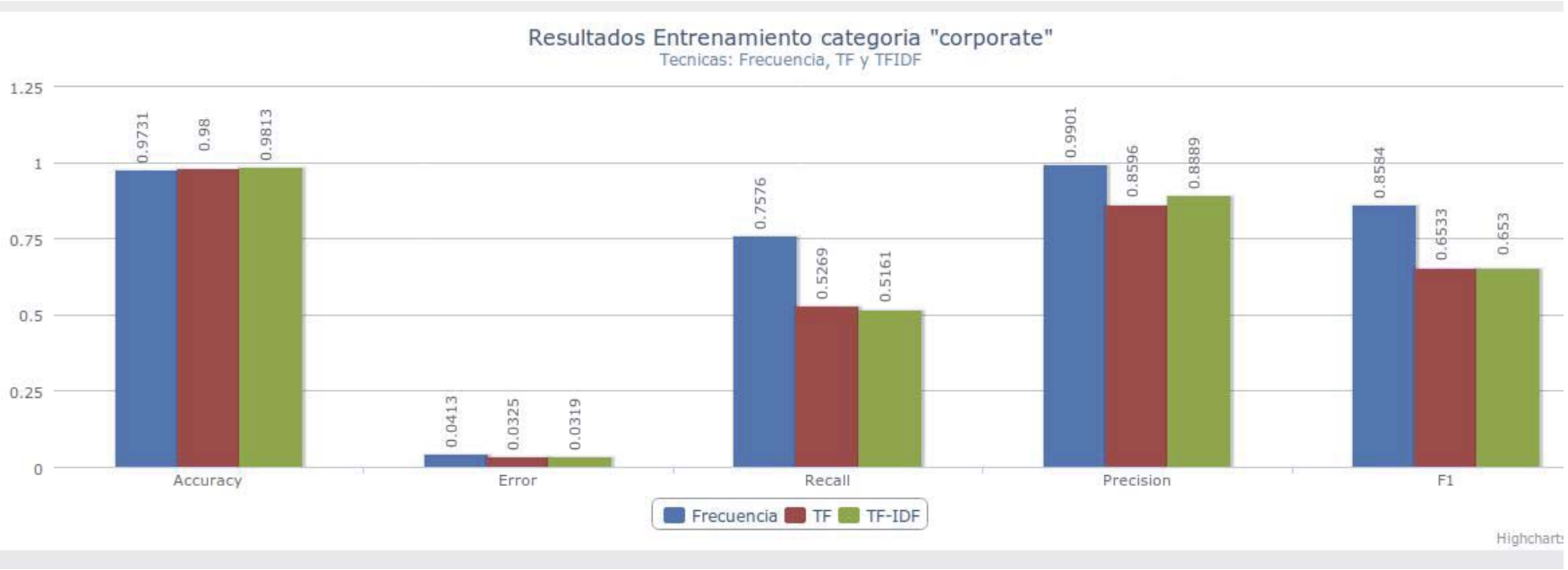

Figura 8-5: Categoría “corporate”: 60% Entrenamiento, 40% Testing

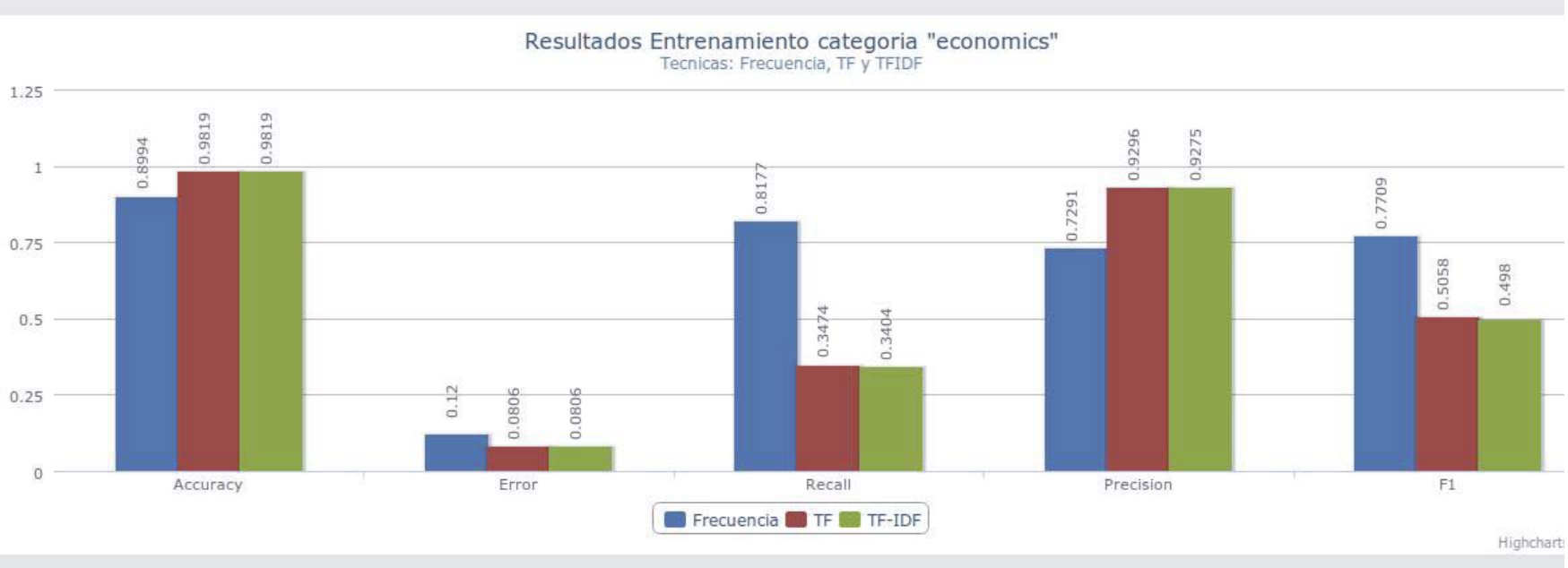

Figura 8-6: Categoría “economics”: 60% Entrenamiento, 40% Testing

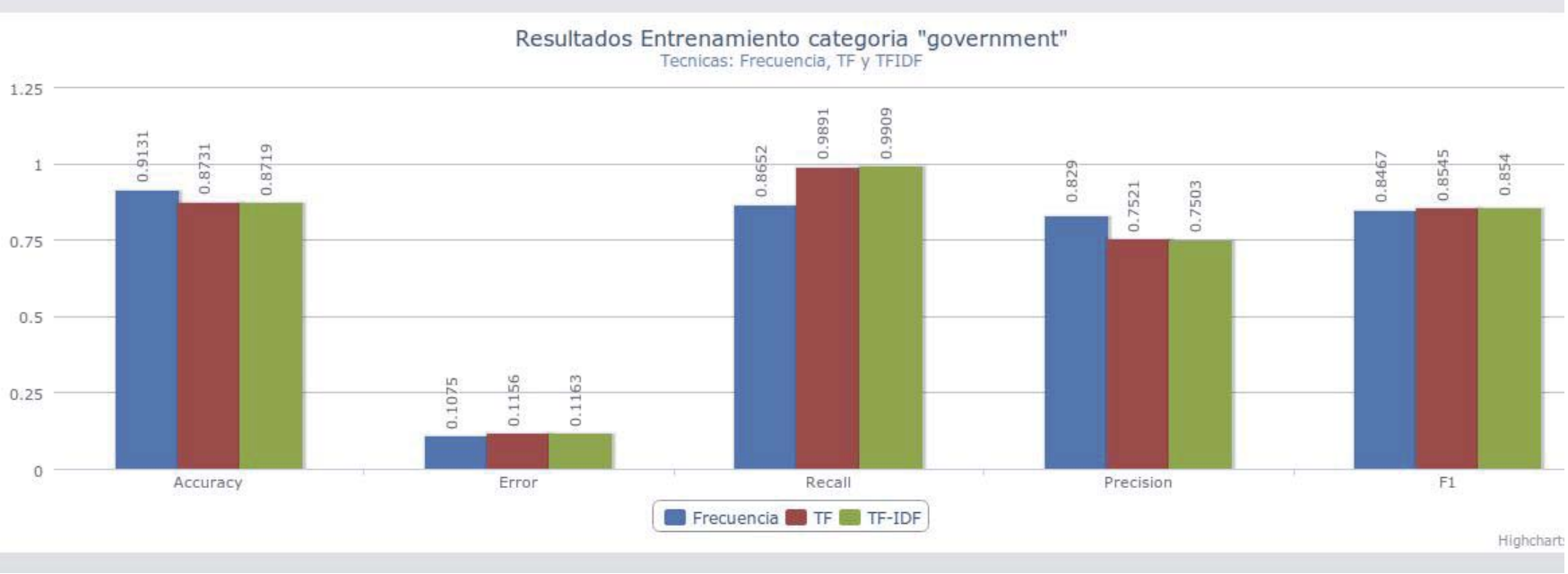

Figura 8-7: Categoría “government”: 60% Entrenamiento, 40% Testing

Figura 8-8: Categoría “markets”: 60% Entrenamiento, 40% Testing

2.1.1.3. Entrenamiento 40% / Testing 60%

Figura 8-9: Categoría “corporate”: 40% Entrenamiento, 60% Testing

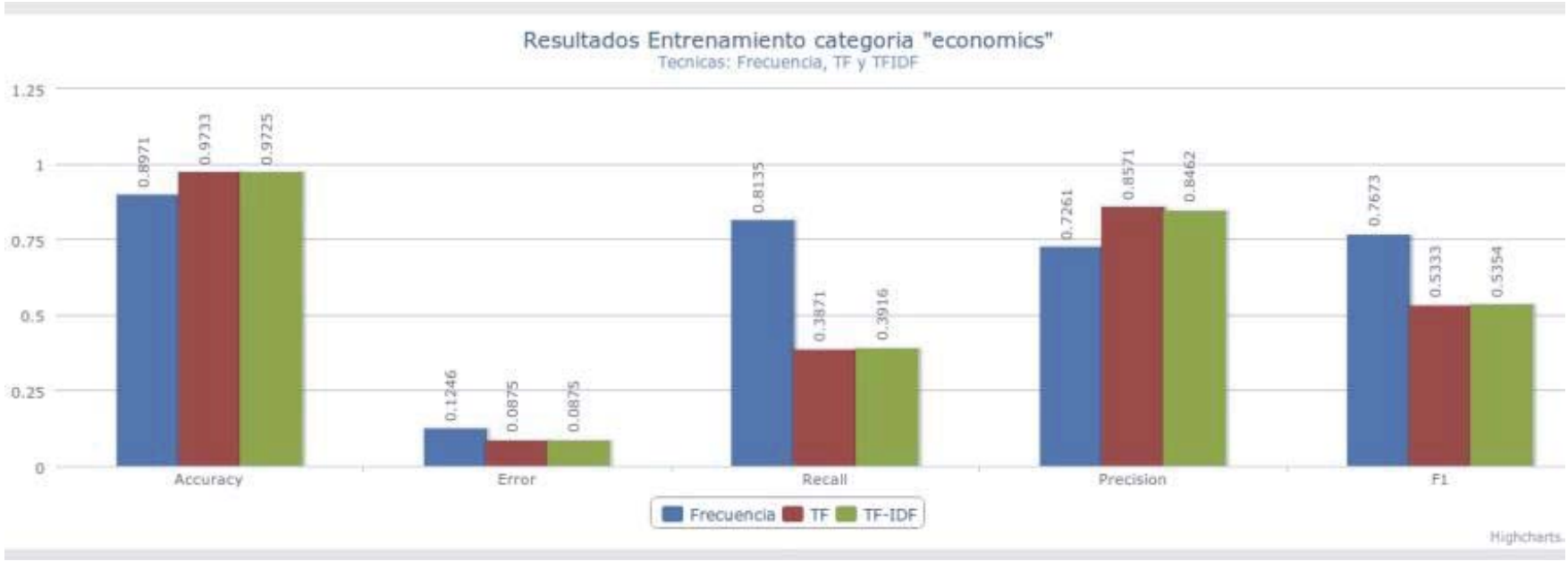

Figura 8-10: Categoría “economics”: 40% Entrenamiento, 60% Testing

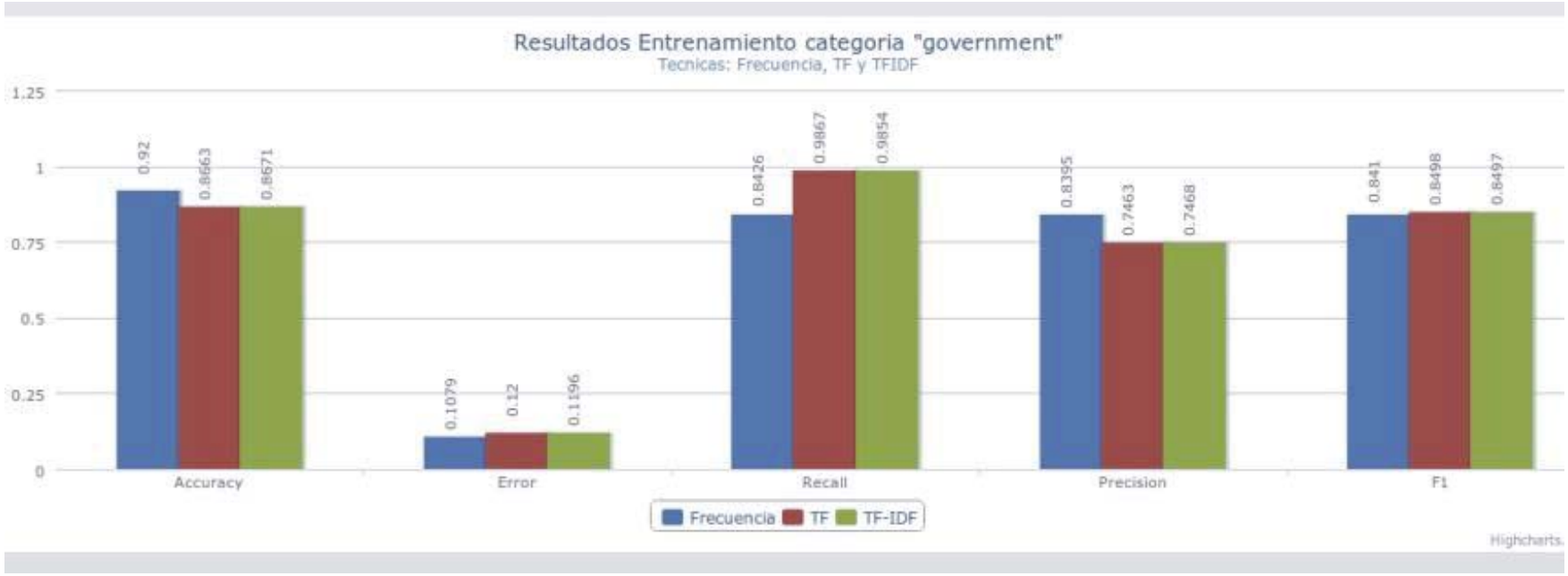

Figura 8-11: Categoría “government”: 40% Entrenamiento, 60% Testing

Figura 8-12: Categoría “markets”: 40% Entrenamiento, 60% Testing

2.1.1.4. Entrenamiento 20% / Testing 80%

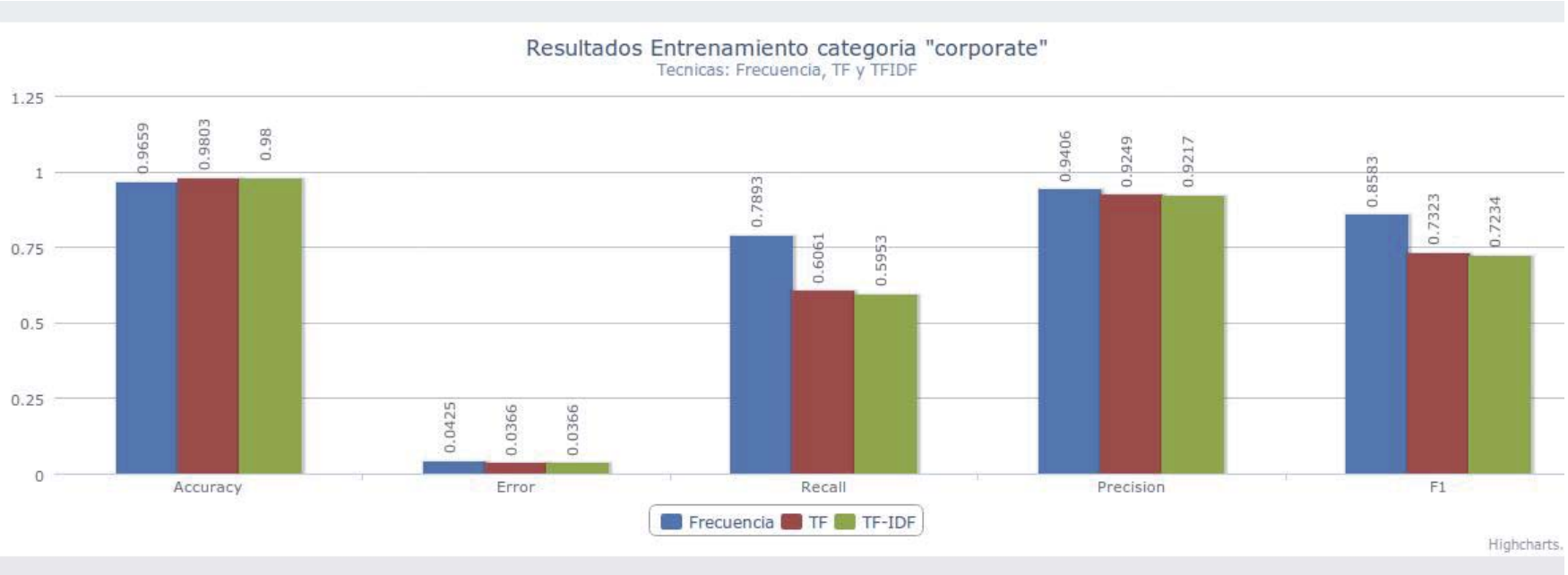

Figura 8-13: Categoría “corporate”: 20% Entrenamiento, 80% Testing

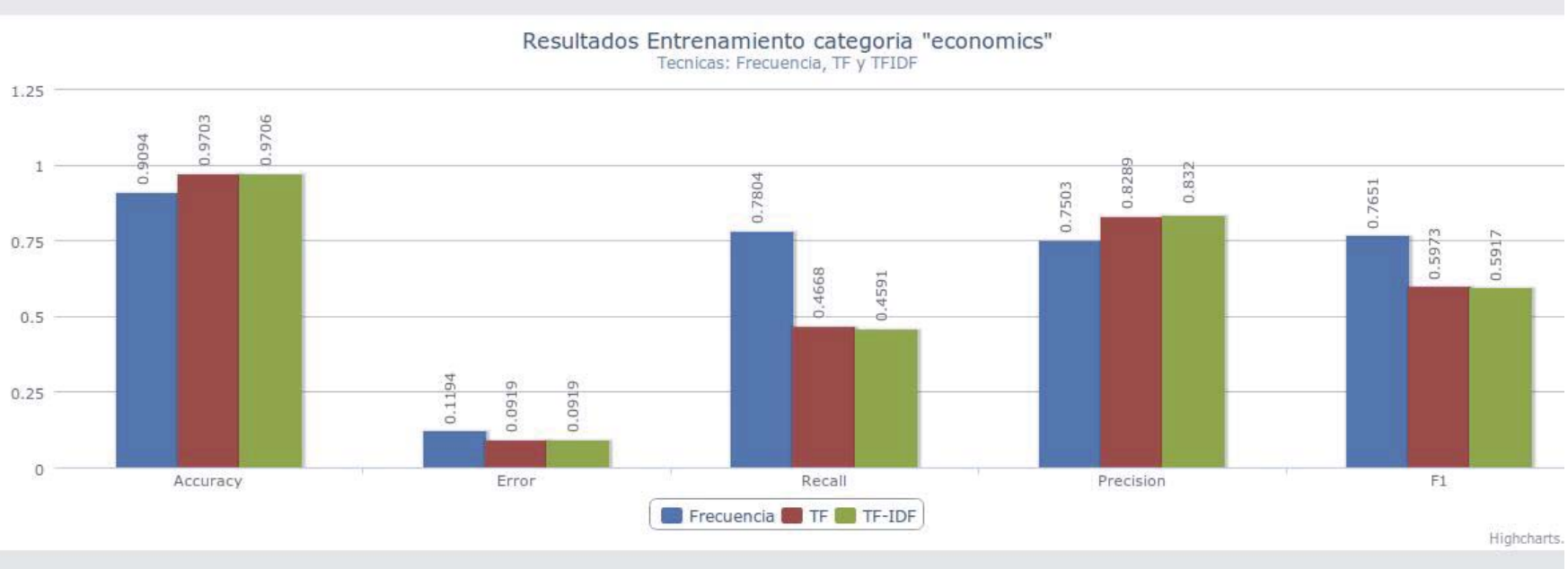

Figura 8-14: Categoría “economics”: 20% Entrenamiento, 80% Testing

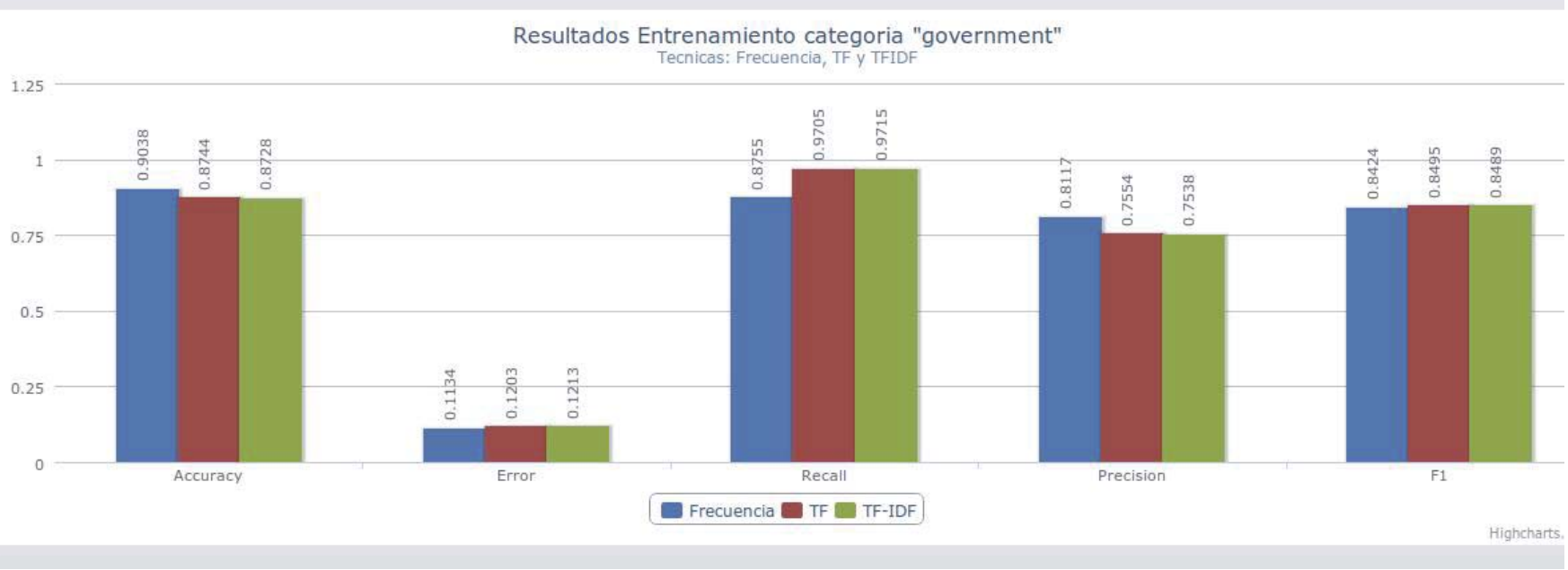

Figura 8-15: Categoría “government”: 20% Entrenamiento, 80% Testing

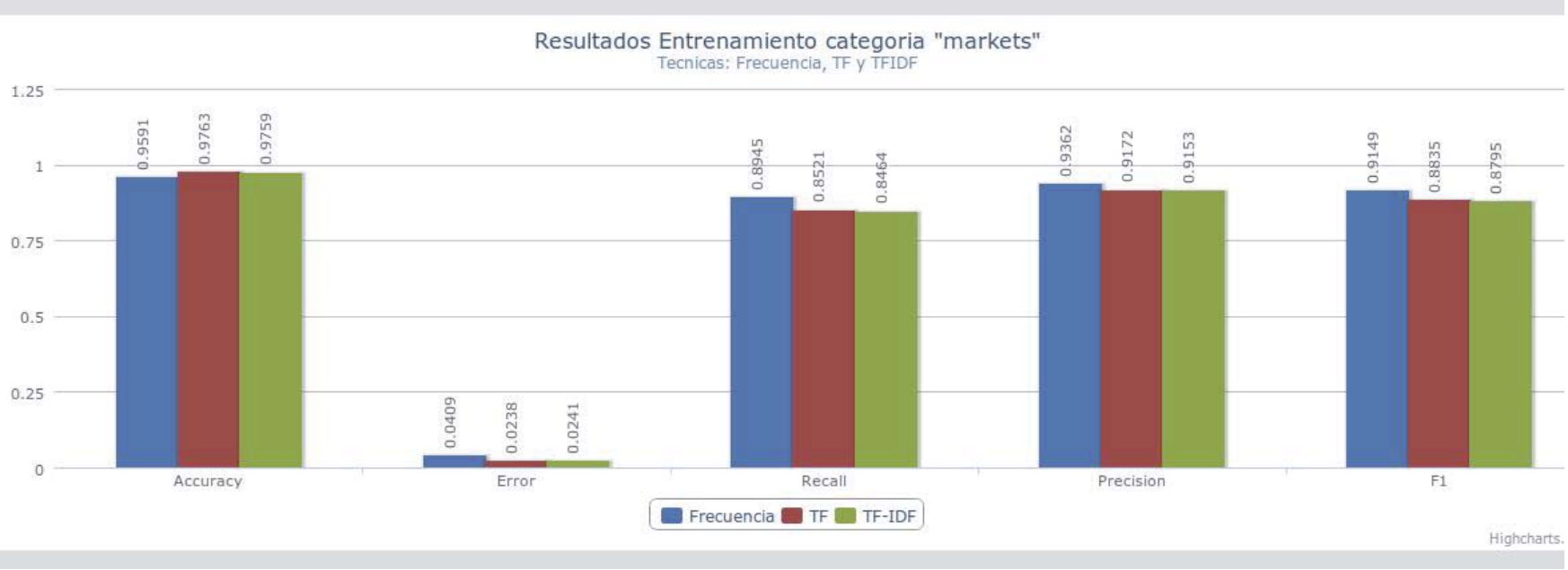

Figura 8-16: Categoría “markets”: 20% Entrenamiento, 80% Testing

8.1.4. Conclusiones Parciales

De los resultados obtenidos se puede concluir que a medida que se aumente el porcentaje de entrenamiento versus el de testing utilizando la técnica de frecuencia, los resultados no difieren mucho entre si por cada técnica de clasificación utilizada y usando distintos porcentajes de división de documentos.

Lo anterior implica que no es necesario de un gran número de documentos para obtener un clasificador que ofrezca mejores resultados, por lo que se pueden realizar nuevos entrenamientos sin tener que invertir en mucho tiempo en el procesamiento de los archivos.

Por otra parte el uso de distintas técnicas para representar los documentos influyó en los resultados, obteniéndose mejores resultados usando la representación por frecuencia por sobre tf y tf-idf en la mayoría de los casos.

8.2. Prueba N°2: Cantidad de archivos procesados variable

8.2.1. Condiciones Iniciales Constantes

- No se incluyen archivos en otros idiomas
- 60% de los archivos son de entrenamiento; 40% de los archivos son para el testing
- Aplicación de las 3 técnicas de representación de documentos

8.2.2. Condiciones Iniciales Variables

- 10.000 archivos divididos en las siguientes categorías: corporate, economics, government, markets.
- 5.000 archivos divididos en las siguientes categorías: corporate, economics, government, markets.
- 1.000 archivos divididos en las siguientes categorías: corporate, economics, government, markets.
- 500 archivos divididos en las siguientes categorías: corporate, economics, government, markets.

8.2.3. Resultados

2.1.1.5. Entrenamiento 500 archivos

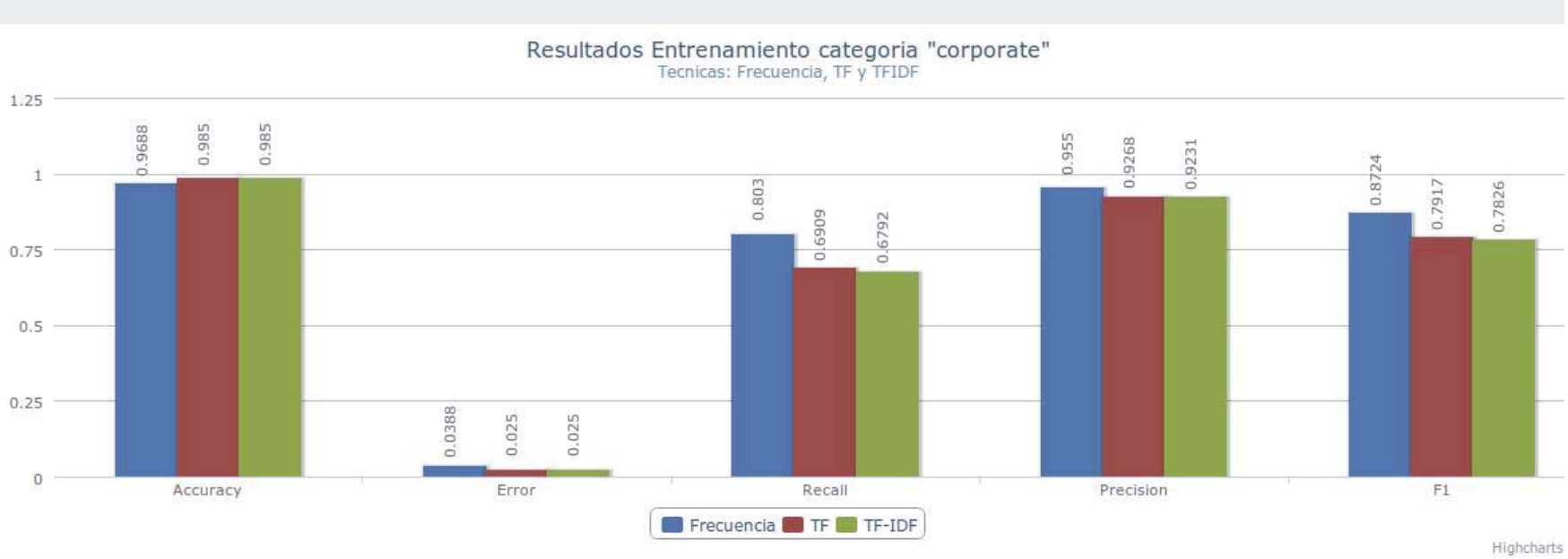

Figura 8-17: Categoría “corporate”: 500 archivos

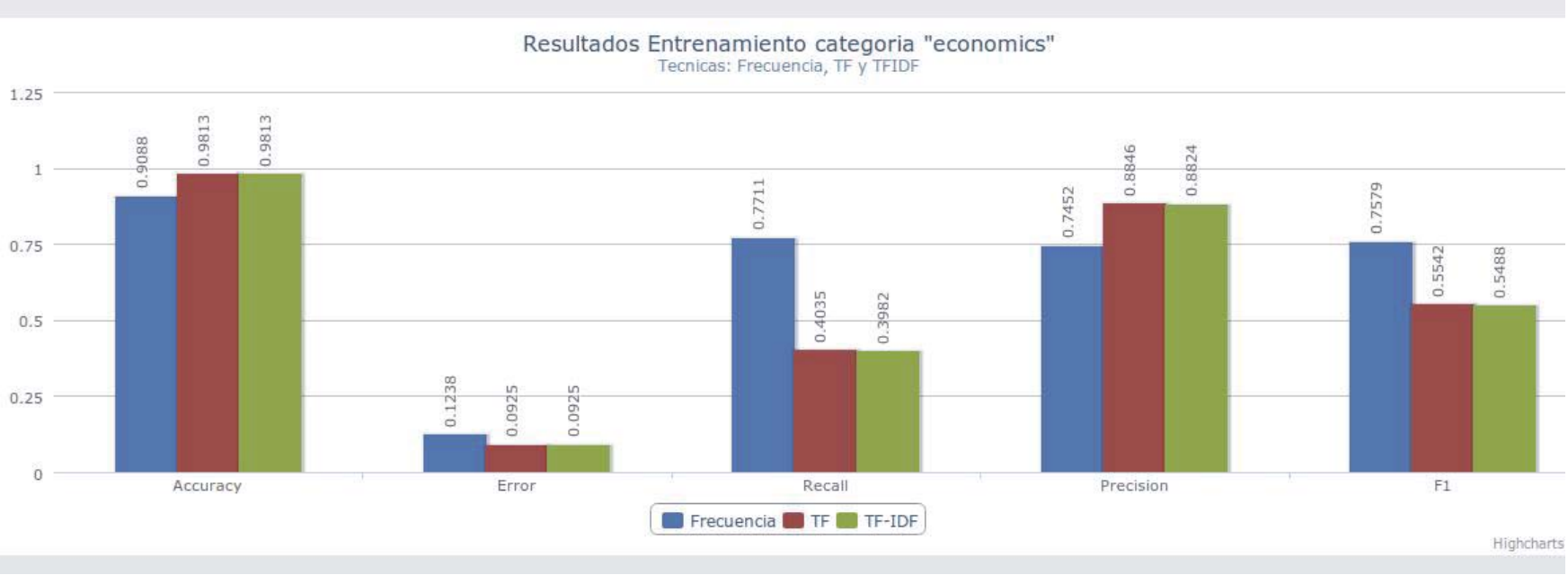

Figura 8-18: Categoría “economics”: 500 archivos

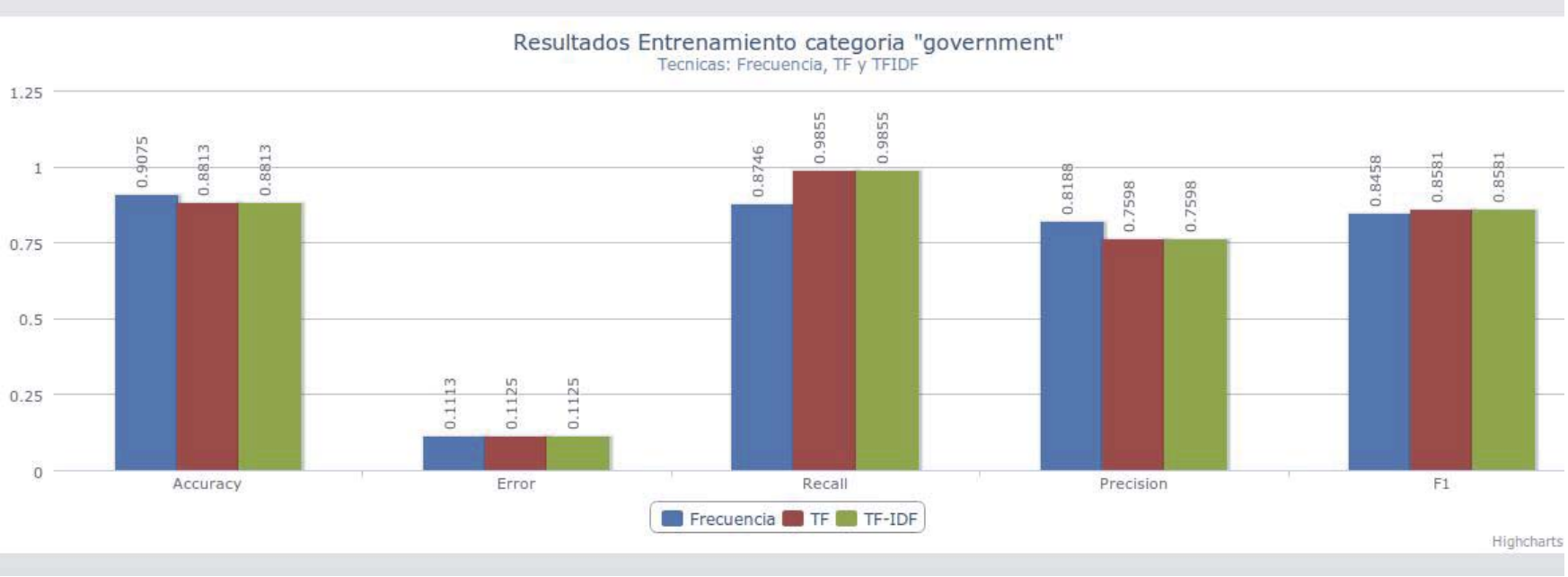

Figura 8-19: Categoría “government”: 500 archivos

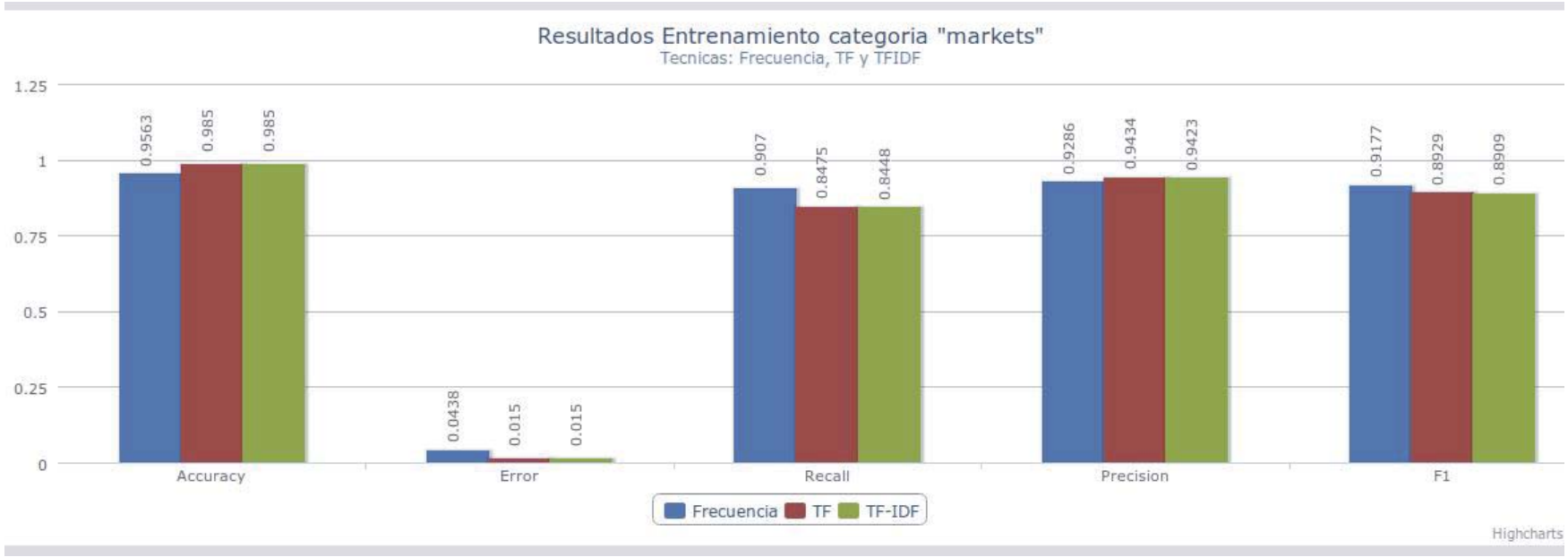

Figura 8-20: Categoría “markets”: 500 archivos

2.1.1.6. Entrenamiento 1000 archivos

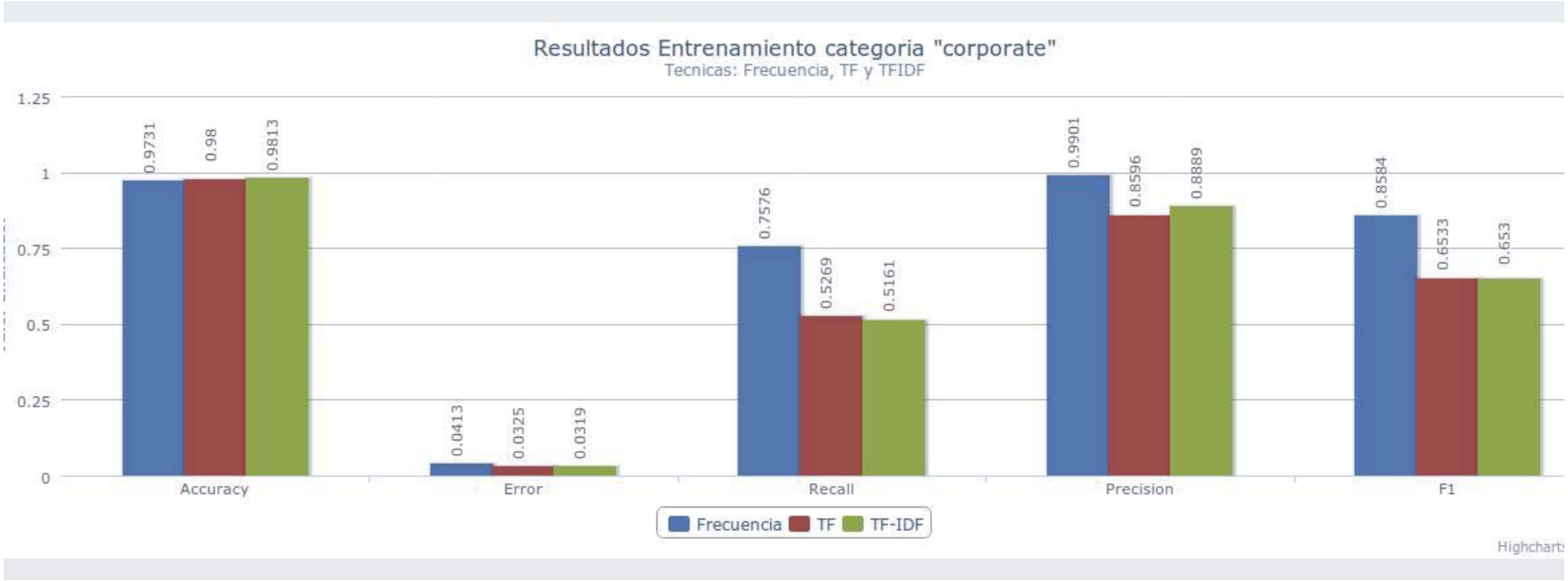

Figura 8-21: Categoría “corporate”: 1000 archivos

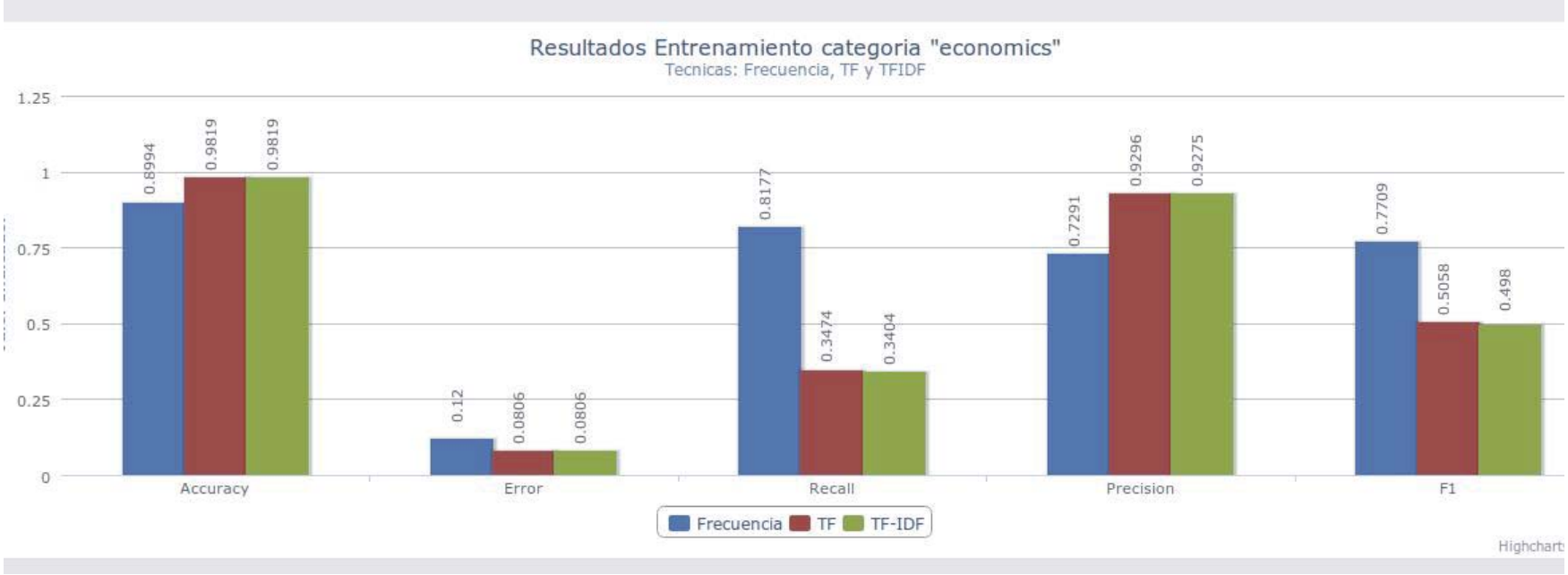

Figura 8-22: Categoría “economics”: 1000 archivos

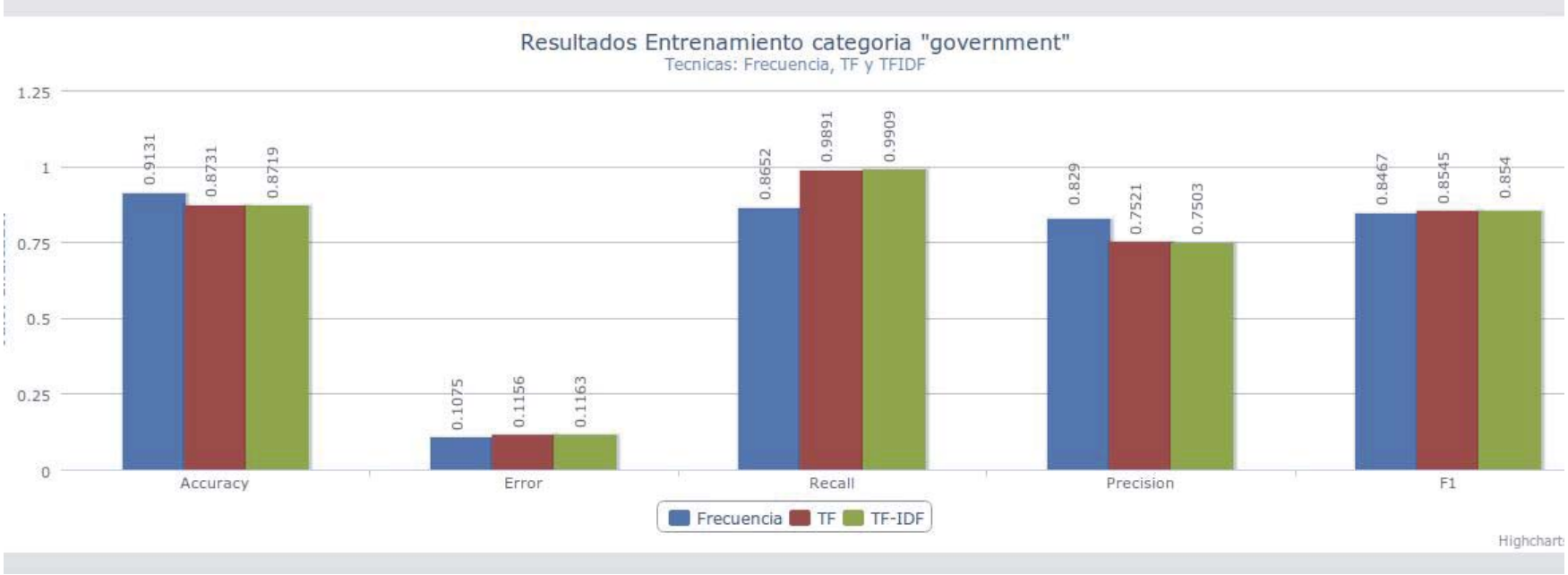

Figura 8-23: Categoría “government”: 1000 archivos

Figura 8-24: Categoría “markets”: 1000 archivos

2.1.1.7. Entrenamiento 5000 archivos

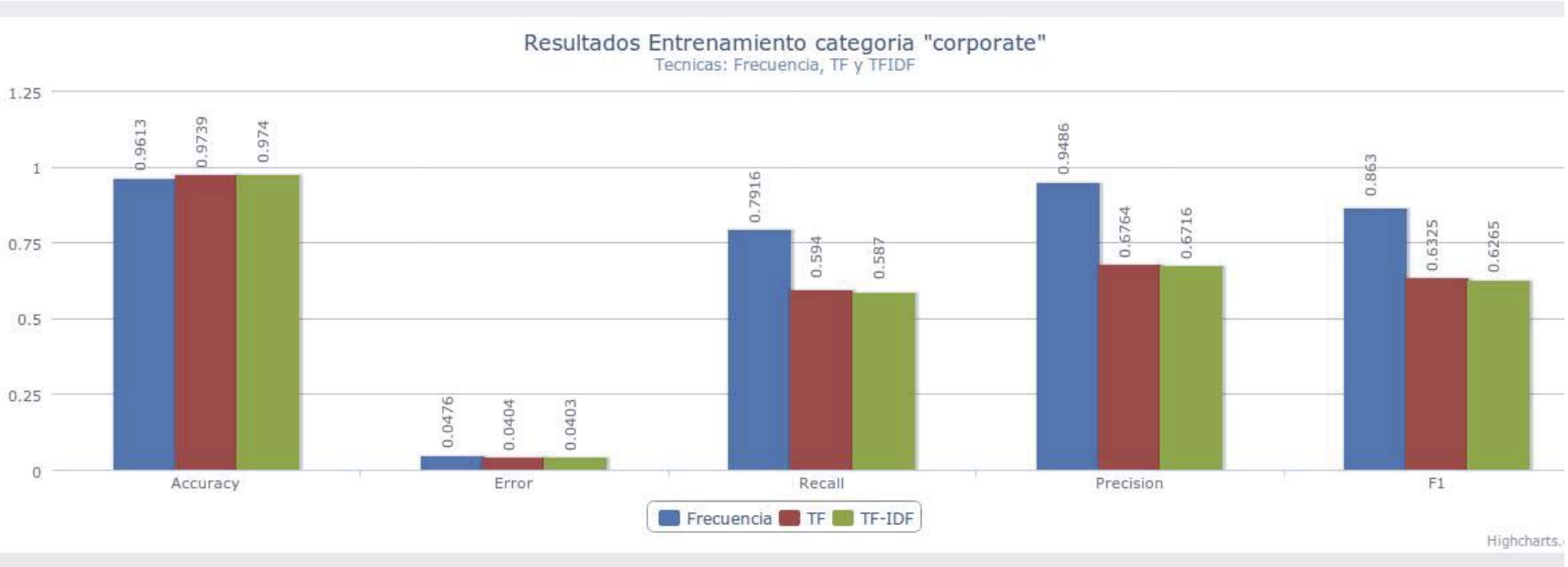

Figura 8-25: Categoría “corporate”: 5000 archivos

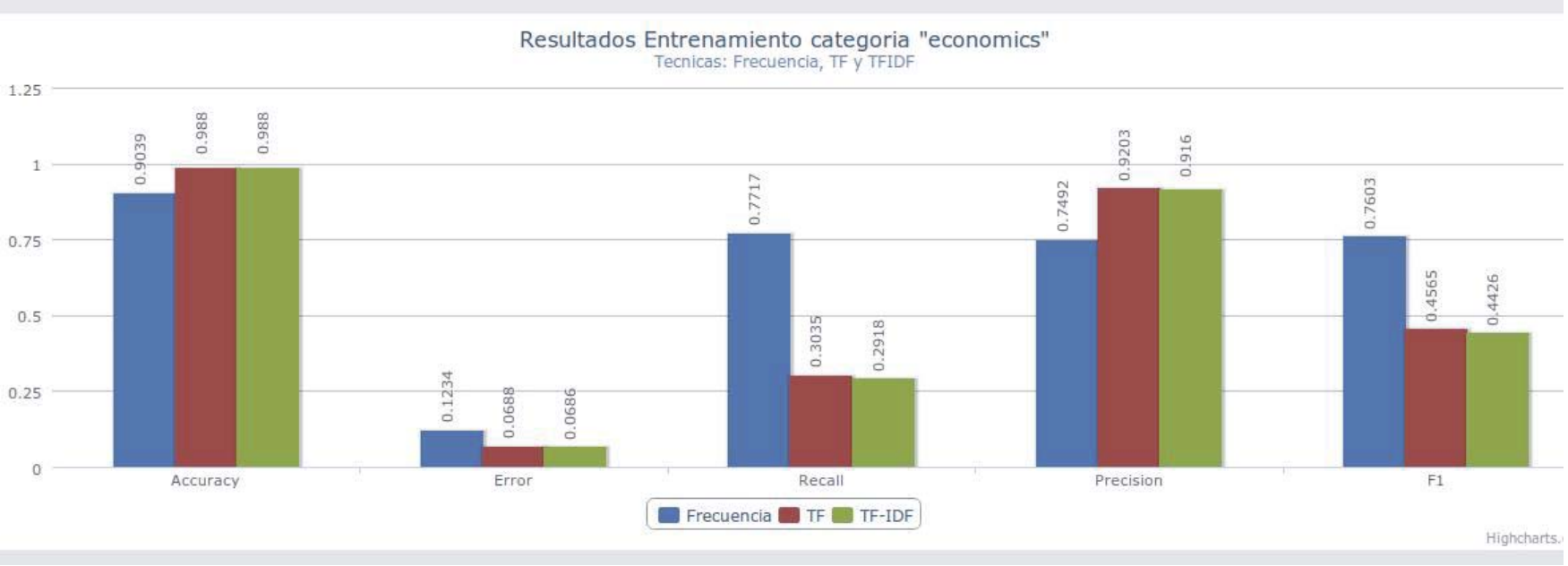

Figura 8-26: Categoría “economics”: 5000 archivos

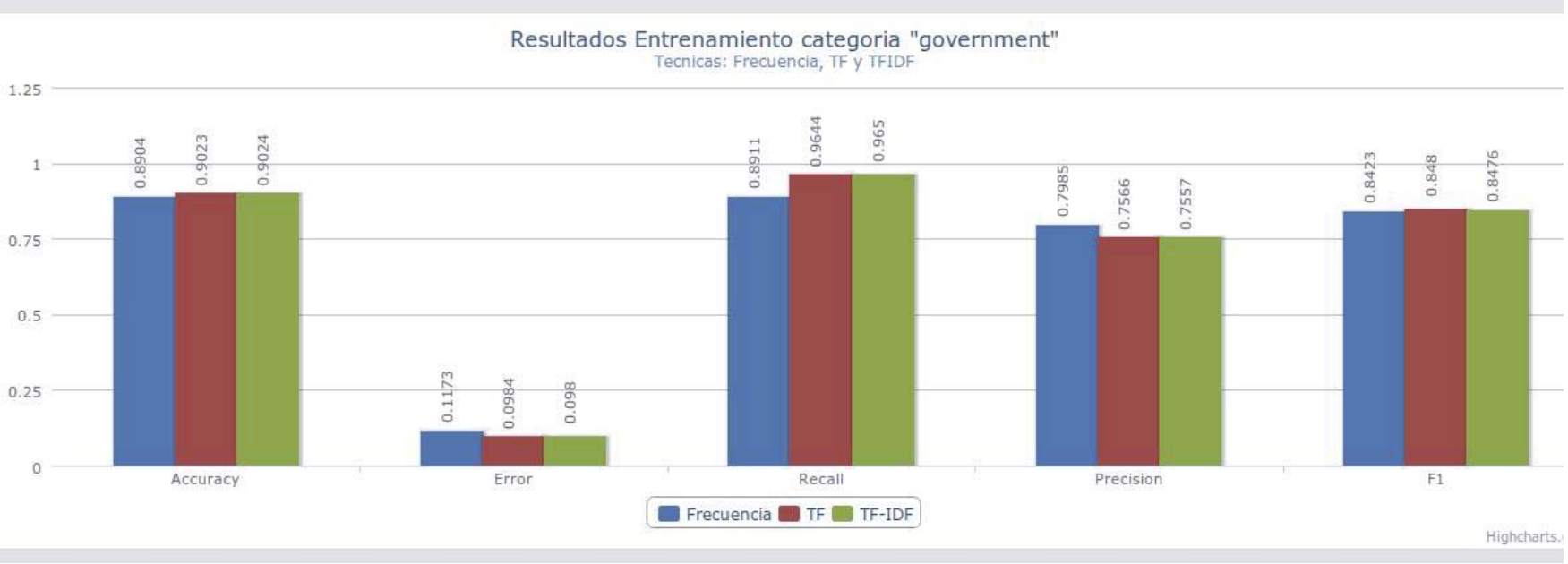

Figura 8-27: Categoría “government”: 5000 archivos

Figura 8-28: Categoría “markets”: 5000 archivos

2.1.1.8. Entrenamiento 10000 archivos

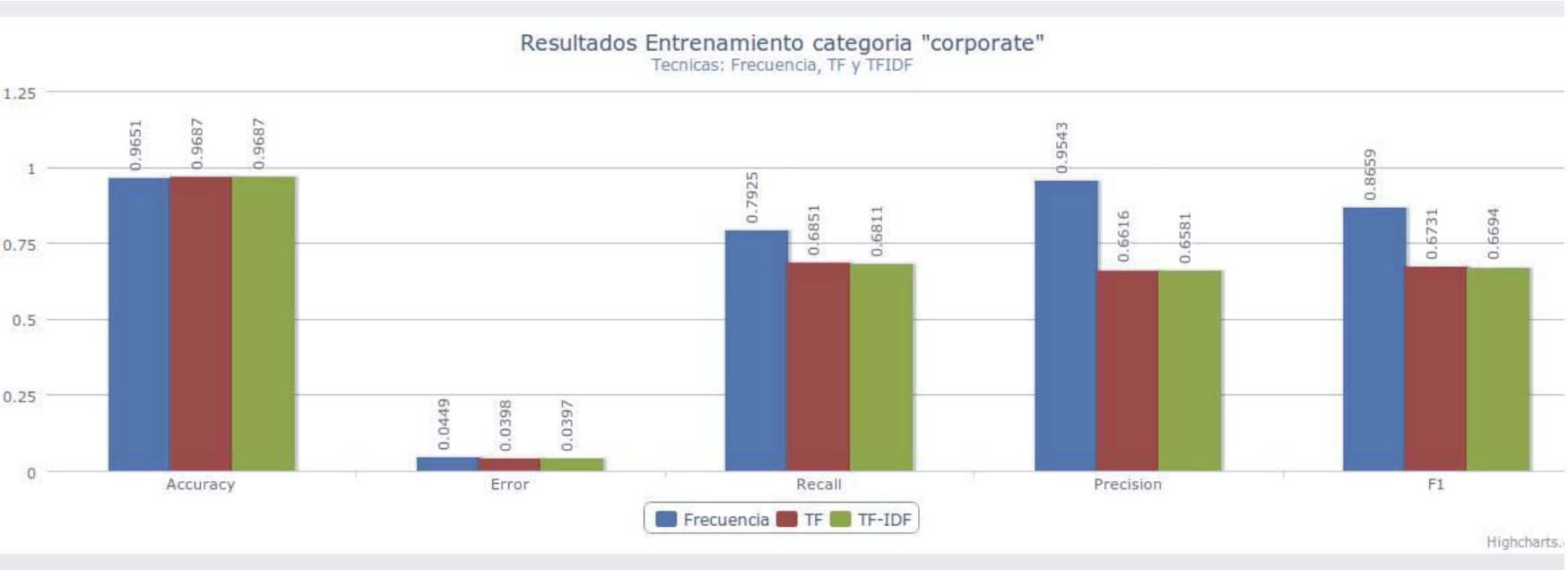

Figura 8-29: Categoría “corporate”: 10000 archivos

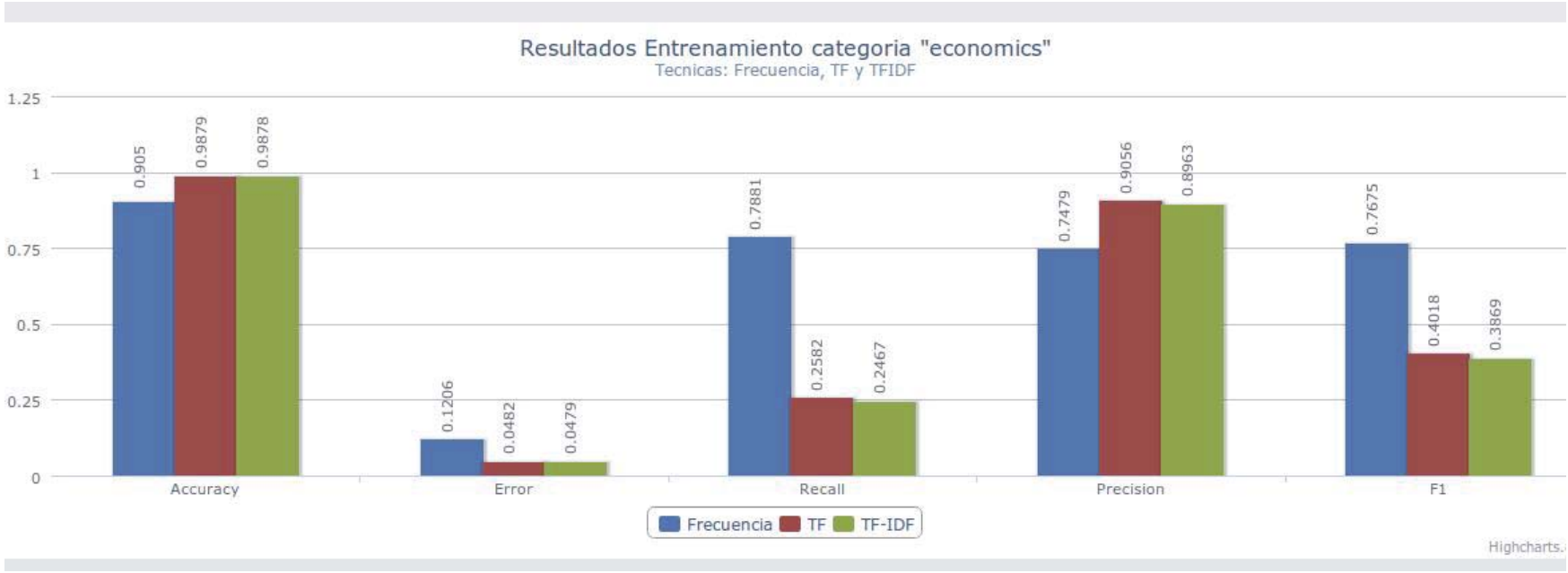

Figura 8-30: Categoría “economics”: 10000 archivos

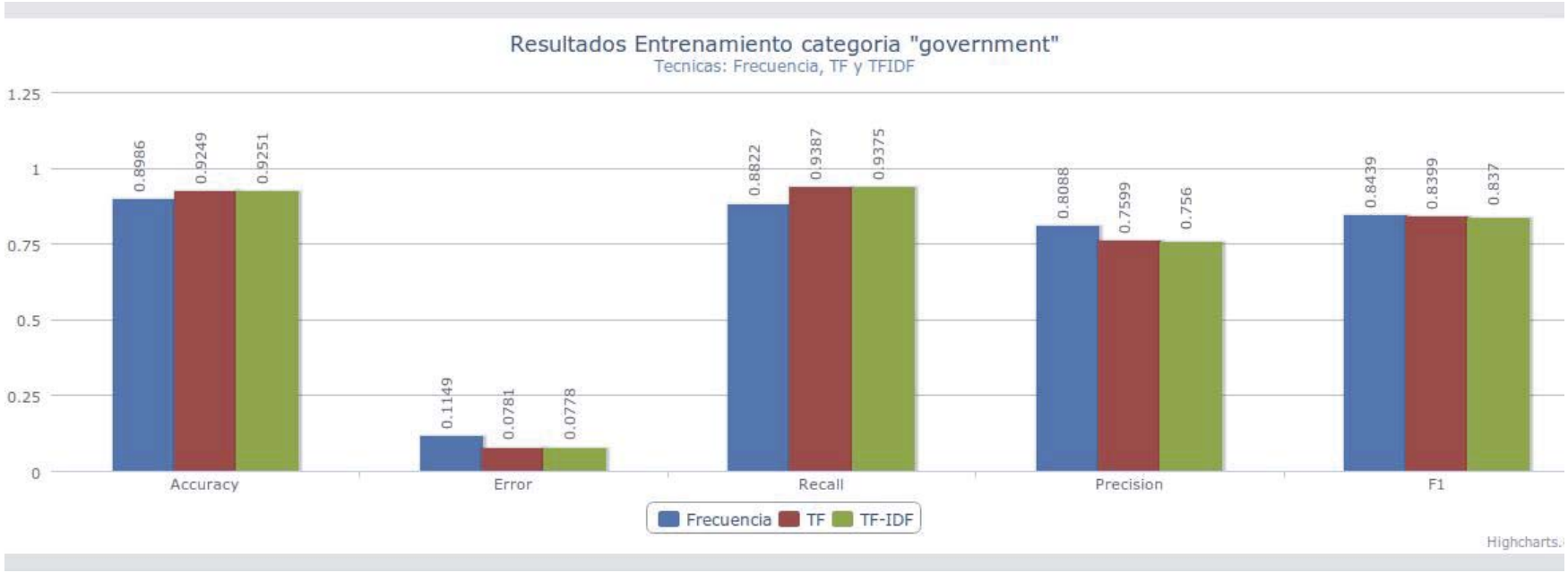

Figura 8-31: Categoría “government”: 10000 archivos

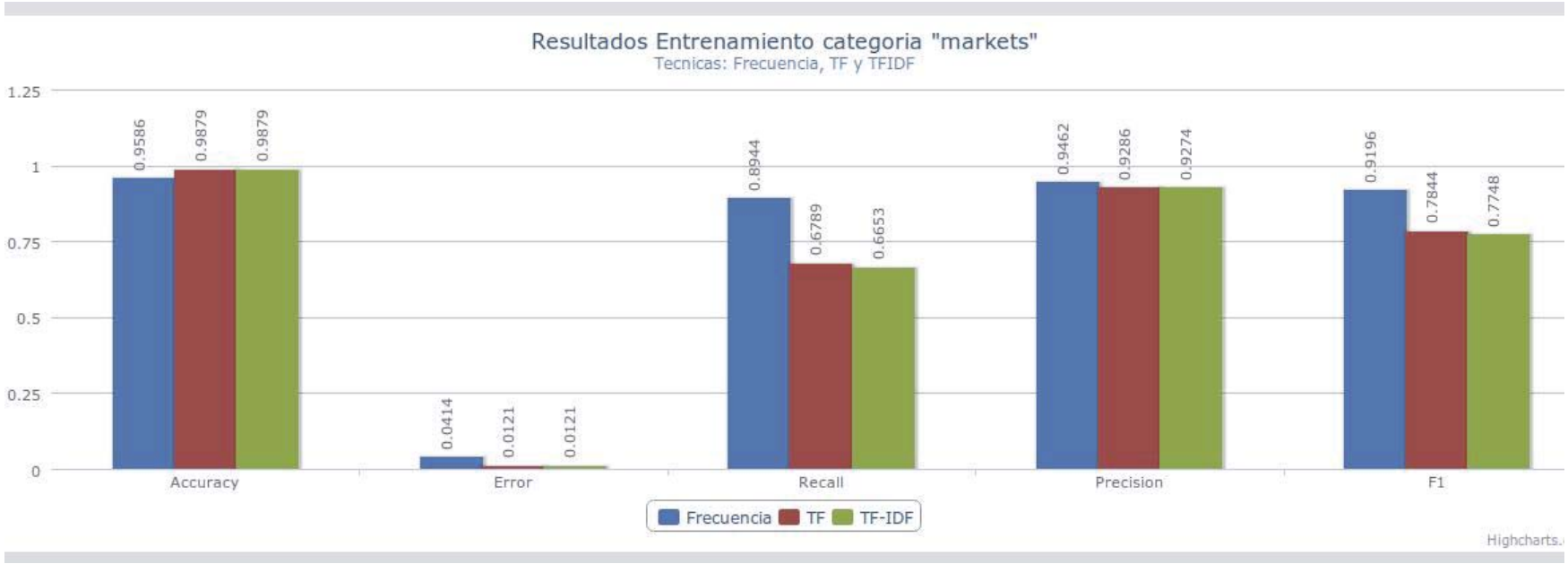

Figura 8-32: Categoría “markets”: 10000 archivos

8.2.4. Conclusiones Parciales

De los resultados obtenidos se puede concluir que a medida que se aumenten los archivos utilizados para el entrenamiento y clasificación, los resultados disminuyen en poca medida su calidad. Esto responde al hecho de que a mayor la cantidad de documentos, mayor es la cantidad de errores que se presentan y, dado los resultados esta relación es más que lineal.

Lo anterior implica que no es necesario de un gran número de documentos no implica necesariamente una mejor representación de una categoría, pues también se presenta el efecto de que también se arrastren más errores de entrenamiento.

Por otra parte el uso de distintas técnicas para representar los documentos influyó en los resultados, obteniéndose mejores resultados usando la representación por frecuencia por sobre tf y tf-idf en la mayoría de los casos.

8.3. Prueba N°3: Inclusión de varios idiomas variable

8.3.1. Condiciones Iniciales Constantes

- 60% de los archivos son de entrenamiento; 40% de los archivos son para el testing
- 1500 archivos divididos en las siguientes categorías: ciencia, computación, deportes, política, religión
- Aplicación de las 3 técnicas de representación de documentos

8.3.2. Condiciones Iniciales Variables

- No se incluyen archivos en otros idiomas
- Si se incluyen archivos en otros idiomas (inglés, francés, español)

8.3.3. Resultados

2.1.1.9. Entrenamiento con documentos sólo en Inglés

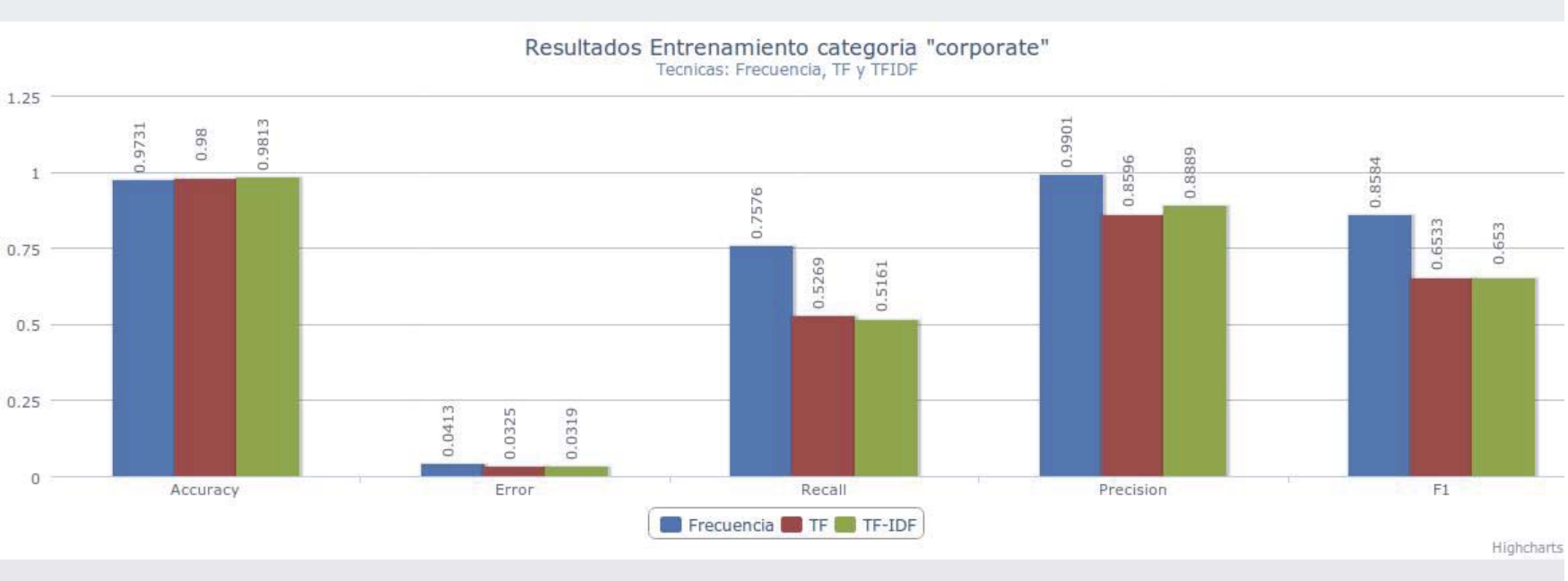

Figura 8-33: Categoría “corporate”: Sin Multilingüe

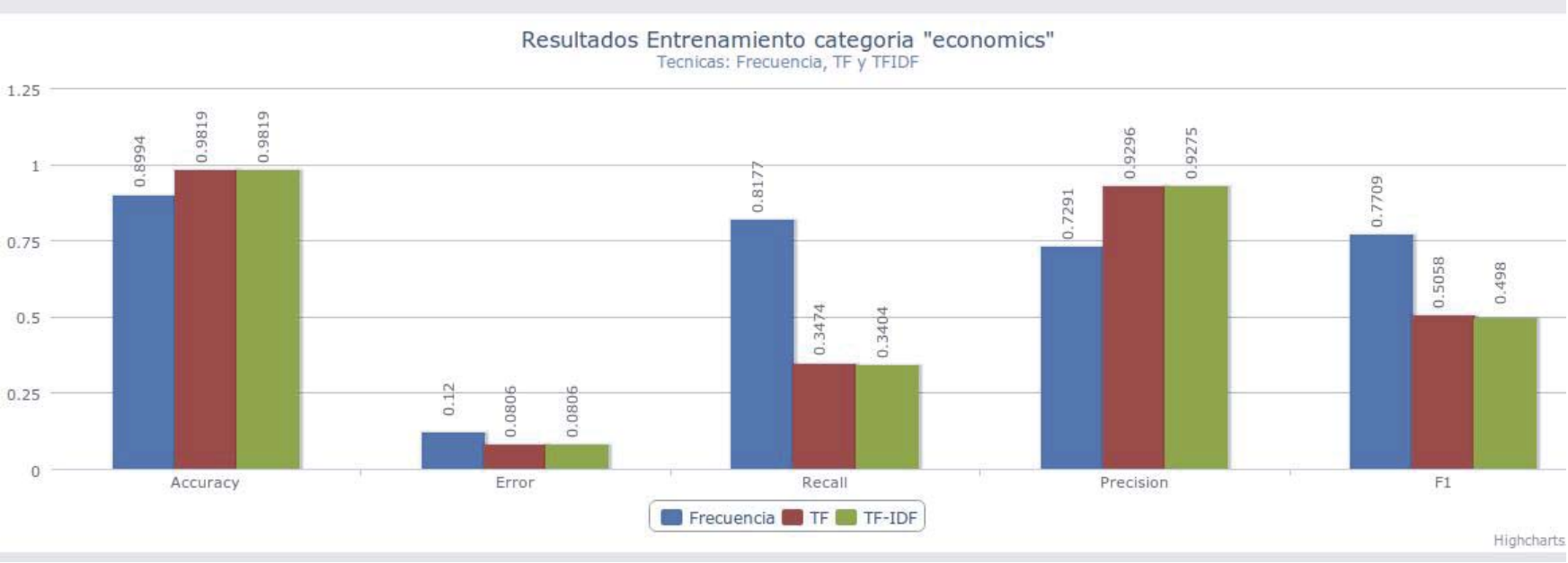

Figura 8-34: Categoría “economics”: Sin Multilingüe

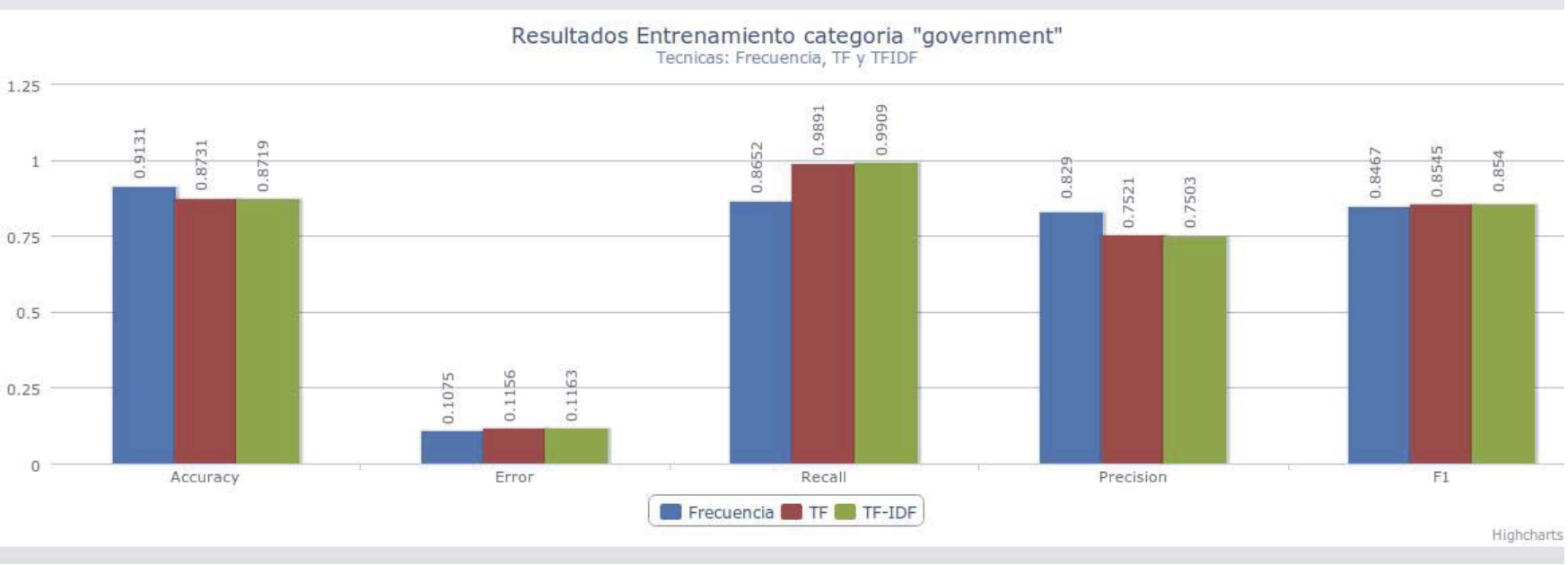

Figura 8-35: Categoría “government”: Sin Multilingüe

Figura 8-36: Categoría “markets”: Sin Multilingüe

2.1.1.10. Entrenamiento con documentos en varios idiomas (Inglés, Francés, Español)

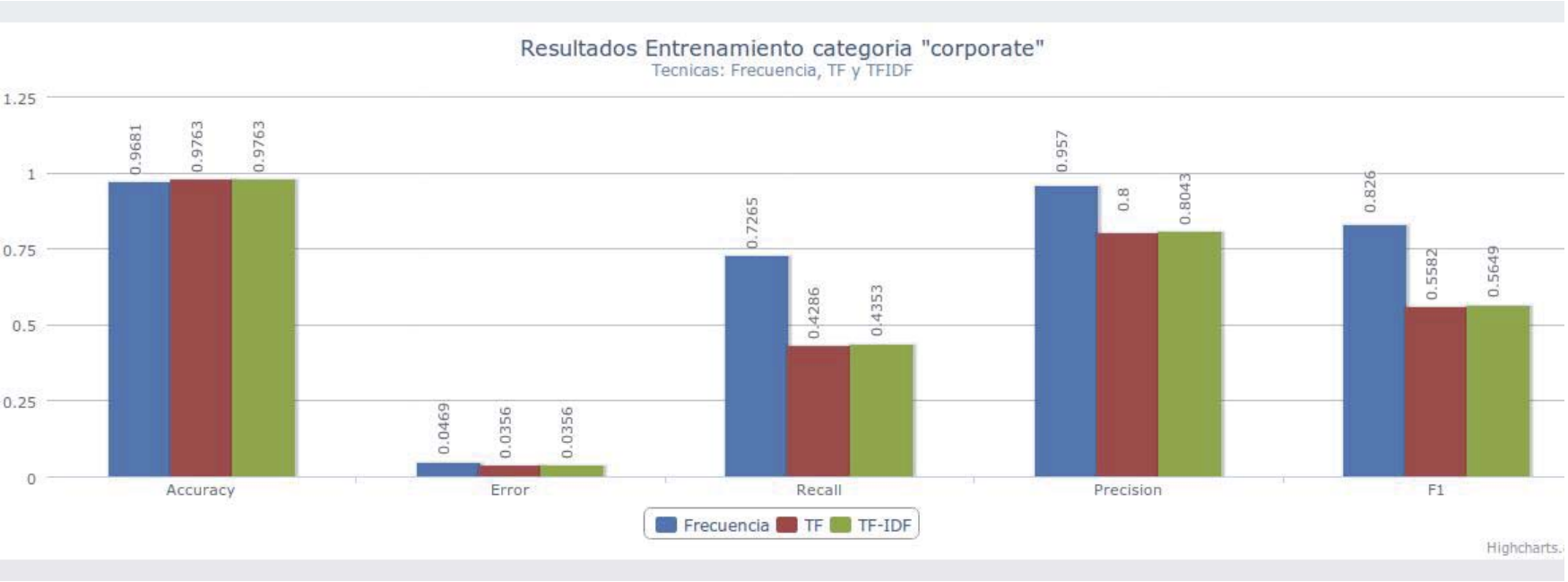

Figura 8-37: Categoría “corporate”: Con Multilingüe

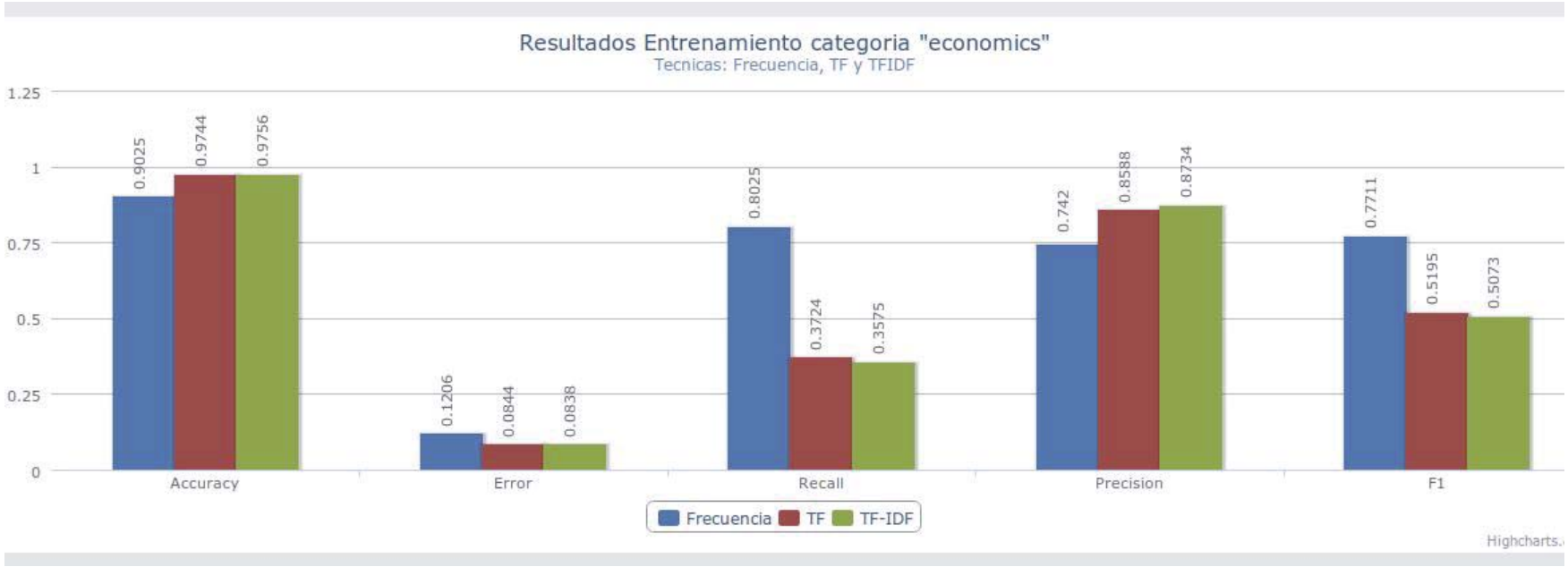

Figura 8-38: Categoría “economics”: Con Multilingüe

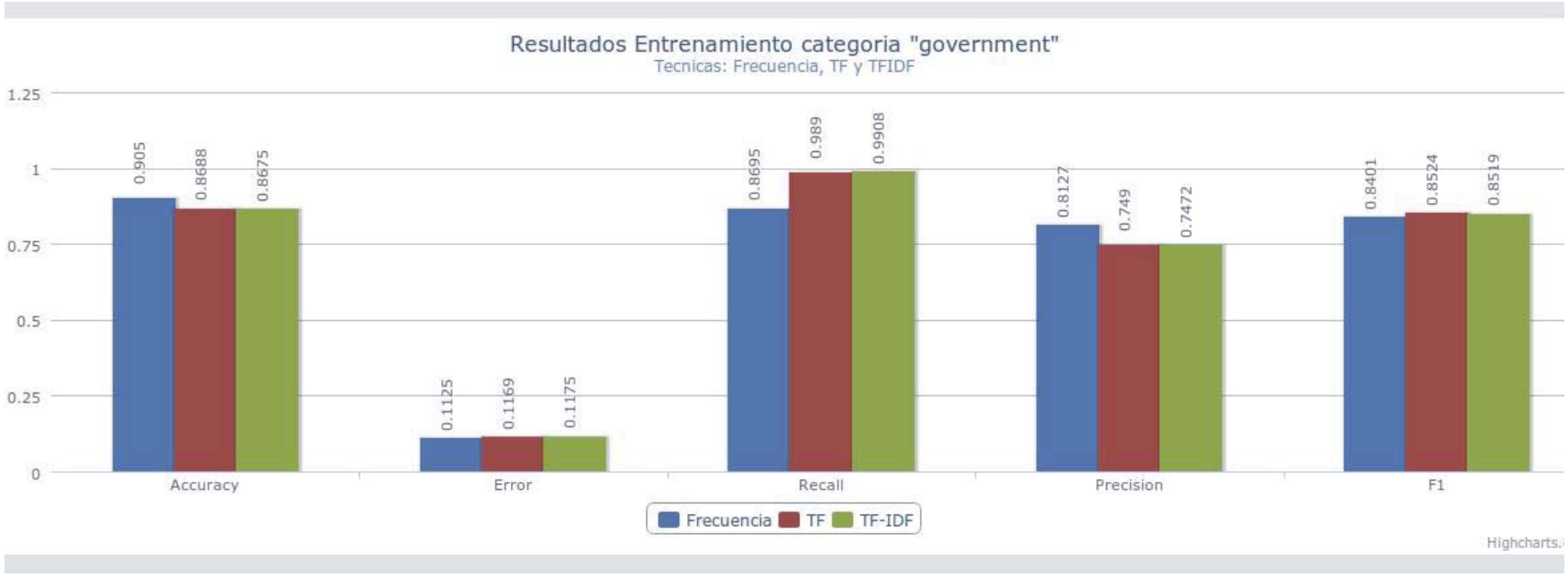

Figura 8-39: Categoría “government”: Con Multilingüe

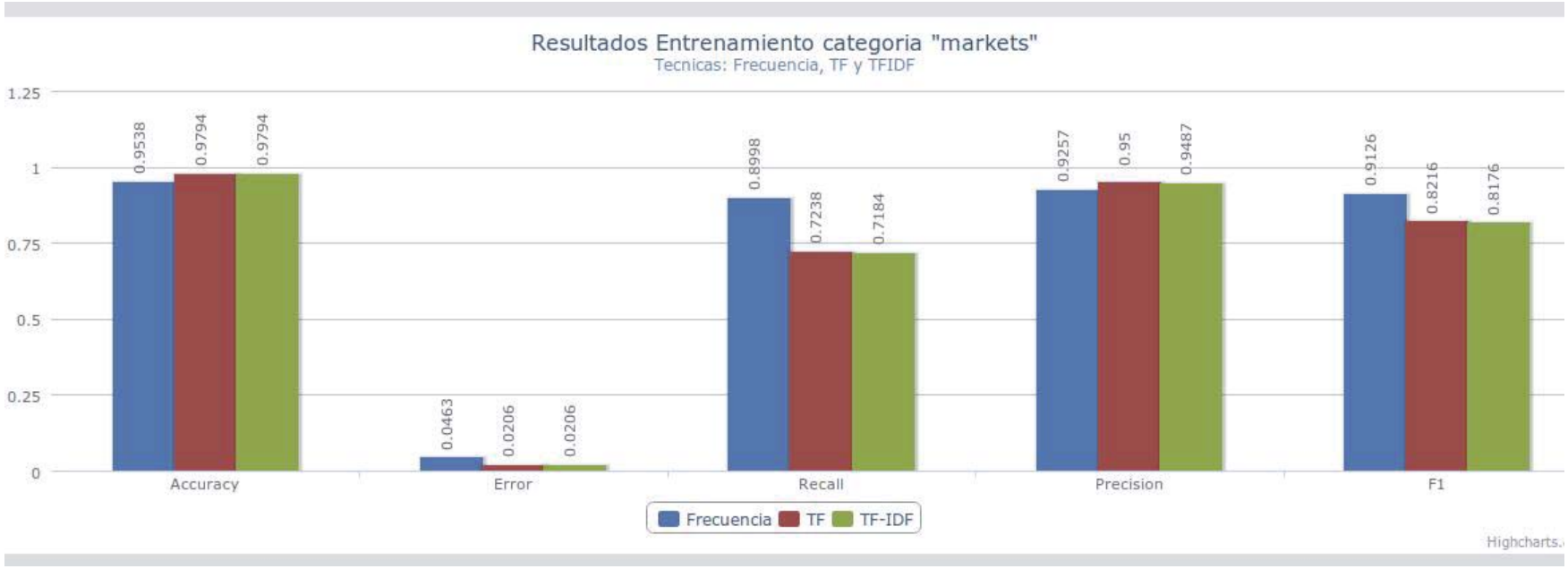

Figura 8-40: Categoría “markets”: Con Multilingüe

8.3.4. Conclusiones Parciales

De los resultados obtenidos se puede concluir que la inclusión de archivos que deben ser traducidos en el proceso de clasificación no afecta drásticamente los resultados. Esta influencia se explica debido a que el traductor utilizado no garantiza una traducción perfecta, existiendo palabras sin traducir o mal traducidas, lo que "ensucia" el entrenamiento y su posterior clasificación. Aún así se presenta un mejor resultado a modo general en los entrenamientos y clasificaciones de documentos en un idioma (inglés) por sobre los entrenados y clasificados multilingualmente.

Por otra parte el uso de distintas técnicas para representar los documentos influyó en los resultados, obteniéndose mejores resultados usando la representación por frecuencia por sobre tf y tf-idf en la mayoría de los casos, aunque con drásticas diferencias en distintas pruebas y categorías.

8.4. Conclusiones generales de los resultados

De todas las pruebas realizadas se puede concluir que la técnica de representación de documentos con mejores resultados es la de Frecuencia, seguido por TF y TF-IDF.

Si bien, el accuracy en casi todas las pruebas con todas las técnicas de clasificación resultaron muy altos, este valor no se debe considerar por sobre los demás, sino que al contrario, se debe preferir calificar por otros indicadores como F1, el Recall y la precisión. Lo anterior se explica pues para el cálculo del accuracy se consideran los errores de Verdaderos Negativos (aquellos documentos que no perteneciendo a una determinada categoría, efectivamente no fueron clasificado ahí), los cuales tienden a acaparar una gran mayoría de su valor.

El método ofrece mejores resultados con una menor proporción de archivos de entrenamiento que de testing, por lo que se infiere que se necesitan de pocos documentos para realizar una buena representación de las categorías. Además mientras más documentos se utilicen para el testing, los resultados deben tender a algo más estable, siendo acorde a los resultados, mejores a medida que existan más documentos de testing.

Finalmente, con respecto a la aplicación de método para clasificar documentos en varios idiomas, se rescata que si es factible y que es posible obtener buenos resultados, no muy distantes de los obtenidos con un solo idioma. Para disminuir la brecha entre estos resultados sólo de debe poseer de otra herramienta de traducción de textos.

9. Conclusiones del estudio

Los resultados de la investigación arrojan a luz que el método propuesto funciona según lo esperado, desde la traducción de documentos con Apertium hasta la clasificación misma. Los resultados son satisfactorios con respectos a otros estudios realizados.

La técnica usada en el presente documento de entrenamiento, utilizando el clasificador Bayesiano ingenuo arroja resultados buenos, aunque es posible obtener mayores índices usando nuevas técnicas acorde a la literatura. Si bien se pudo realizar dicho estudio con técnicas más avanzadas, se eligió dar preferencia el determinar el método en el cual la clasificación de textos en múltiple lenguajes arrojase buenos resultados. Lo anterior permite dar una base para futuros estudios relacionados con el tema.

En general, entonces se puede afirmar que el método demuestra ser un gran potencial para la clasificación de textos, pues ofrece la flexibilidad de cambiar alguna técnica intermedia (por ejemplo cambiar el clasificador bayesiano por alguna otra), lo que lo hace tremendamente versátil y adaptable a las circunstancias que se desee.

Además, gracias a la construcción de la interfaz web y móvil que utiliza el método propuesto, es posible utilizarlo en distintos dispositivos contemporáneos, tanto móviles como de escritorio.

10. Referencias

[1] Bentaallah Mohamed Amine, Malki Mimoun. *WordNet based Multilingual Text Categorization.* (2007).

[2] Fabrizio Sebastiani. *Machine Learning in Automated Text Categorization.* ACM Computing Surveys (2002).

[3] Mikel L. Forcada, Boyan Ivanov Bonev, Sergio Ortiz Rojas. Documentation of the Open-Source Shallow-Transfer Machine Translation Platform Apertium.(2010)

[4] Ke Wu, Xiaolin Wang and Bao-Liang Lu. Cross Language Text Categorization Using a Bilingual Lexicon. (2008).

[5] John Hutchins, Machine Translation: General Overview. (2003)

[6] *http://www.internetworldstats.com/stats7.htm* Información actualizada el 30 de Septiembre del 2009. Revisada el 01 de Noviembre del 2010.

[7] Princeton. *http://wordnet.princeton.edu/* Revisada el 01 de Noviembre del 2010.

[8] Lewis, D. D.; Yang, Y.; Rose, T.; and Li, F. RCV1: A New Benchmark Collection for Text Categorization Research. Journal of Machine Learning Research, 5:361-397 (2004).

[9] *http://wiki.apertium.org/wiki/Main_Page* Revisada el 01 de Noviembre del 2010.

[10] *http://www.nltk.org/* Revisada el 01 de Noviembre del 2010.

[11] Andrew McCallum, Kamal Nigamy. A Comparison of Event Models for Naive Bayes Text Classification. (1998).

[12] *http://mallet.cs.umass.edu/index.php* Revisada el 01 de Noviembre del 2010.

11. Anexo

11.1. Código Fuente del programa desarrollado (extracto)

```
<?php
ini_set('max_input_time',-1);
ini_set('upload_max_filesize','1024M');
ini_set('post_max_size','1024M');
ini_set('max_file_uploads',1000);
ini_set('memory_limit','4096M');
ini_set('max_execution_time',0);

$REPRESENTACION         = array();  //$REPRESENTACION[categoria]         -->
{[0]:{Frecuencia}, [1]:{TF}, [2]:{TF-IDF}}
$directorioTraining     = 'repositorio_training';
$contadorProcesados     = 0;
$totalArchivos          = 0;

$file                   = fopen('tmp/archivos_training.txt', "r");
$totalArchivos    =                                        (int)fread($file,
filesize('tmp/archivos_training.txt'));
fclose($file);

function iterarTodasCategorias($path) {
global $REPRESENTACION;

$ignore     = array('.','..','cgi-bin','.DS_STORE');
$categorias = scandir($path);
foreach($categorias as $categoria){
if(in_array($categoria, $ignore))
continue;
if (is_dir(rtrim($path, '/').'/'.$categoria))
$REPRESENTACION[$categoria]   =              procesarCategoria(rtrim($path,
'/').'/'.$categoria, $categoria);
}

foreach($categorias as $categoria){
if(in_array($categoria, $ignore))
```

```
continue;

$TFIDF                          = array();
$tmp_tf                 = $REPRESENTACION[$categoria][1];
$cantidadCategorias     = count($categorias);

foreach($tmp_tf as $palabra => $cantidad){
$cantidadPalabraEnOtrasCategorias = 0;          //La cantidad de categorias
que contienen la palabra de la categoria i (incluyendose)
foreach($categorias as $tmp_categoria){
if(in_array($tmp_categoria, $ignore))
continue;
if(array_key_exists($palabra, $REPRESENTACION[$tmp_categoria][1]))
$cantidadPalabraEnOtrasCategorias++;
}
$TFIDF[$palabra]     =     $cantidad     *     log($cantidadCategorias     /
$cantidadPalabraEnOtrasCategorias);
}
$REPRESENTACION[$categoria][2] = $TFIDF;
}
}

function procesarCategoria($path, $categoria) {
global $contadorProcesados;
$FRECUENCIA                             = array();
$TF                                     = array();

$ignore = array('.','..','cgi-bin','.DS_STORE');
$files = scandir($path);
foreach($files as $t) {
if(in_array($t, $ignore))
continue;
if (!is_dir(rtrim($path, '/').'/'.$t)){
$rutaArchivo            = rtrim($path, '/').'/'.$t;
$file                           = fopen($rutaArchivo, 'r');
$contenidoArchivo = fread($file, filesize($rutaArchivo));
fclose($file);
```

```
$cantidadPalabrasArchivo      = contarPalabrasEnArchivo($contenidoArchivo);
$FRECUENCIA                          =            sumaAsociativa($FRECUENCIA,
$cantidadPalabrasArchivo);
}

$breadcrum  = fopen('tmp/archivos_representacion.php', "w");
fwrite($breadcrum,      '{"procesados":      '.$contadorProcesados.',"total":
'.$totalArchivos.'}');
fclose($breadcrum);
}

$cantidadPalabrasCategoria    = count($FRECUENCIA);
foreach($FRECUENCIA as $palabra => $cantidad)
$TF[$palabra]     = $cantidad/$cantidadPalabrasCategoria;

$tmp_representacion     = array($FRECUENCIA, $TF, array());
return $tmp_representacion;
}

function sumaAsociativa($array1, $array2){
if(count($array1) == 0)
return $array2;
if(count($array2) == 0)
return $array1;
if(count($array1) >= count($array2)){
$tmp  = $array1;
$array1 = $array2;
$array2     = $tmp;
}

foreach($array1 as $palabra1 => $cantidad1){
if(array_key_exists($palabra1, $array2))
$array2[$palabra1]      += $cantidad1;
else
$array2[$palabra1]      = $cantidad1;
}
return $array2;
}
```

```
function contarPalabrasEnArchivo($contenidoArchivo){
$array_contenidoArchivo =        str_word_count($contenidoArchivo,        1,
'áéíóúÁÉÍÓÚñÑ1234567890');
$unico_contenidoArchivo = array_count_values($array_contenidoArchivo);
return $unico_contenidoArchivo;
}

function guardarRepresentacion($REPRESENTACION){
include('connect.php');
$superSQL   = 'TRUNCATE TABLE representacion_entrenamiento;';
$result     = $mysqli->query($superSQL);
$contador   = 0;
foreach($REPRESENTACION as $categoria => $tmpRepresentaciones){
foreach($tmpRepresentaciones[0] as $palabra => $frecuencia){
$tf               = $tmpRepresentaciones[1][$palabra];
$tfidf            = $tmpRepresentaciones[2][$palabra];
$superSQL   = 'INSERT INTO representacion_entrenamiento (categoria, palabra,
frecuencia,   tf,   tfidf)   VALUES   ("'.$categoria.'",   "'.$palabra.'",
'.$frecuencia.', '.$tf.', '.$tfidf.');';
$result     = $mysqli->query($superSQL);
if($mysqli->affected_rows == 0)
echo $superSQL;
}
}
}

iterarTodasCategorias($directorioTraining);
guardarRepresentacion($REPRESENTACION);
echo 1;
?>
<?php
ini_set('max_input_time',-1);
ini_set('upload_max_filesize','1024M');
ini_set('post_max_size','1024M');
ini_set('max_file_uploads',1000);
ini_set('memory_limit','4096M');
ini_set('max_execution_time',0);
```

```
$directorioClasificados       = 'repositorio_clasificados';
$directorioTraining                 = 'repositorio_training';
$categoriasIniciales          = array();
$contadorProcesados                 = 0;
$totalArchivos                      = 0;

$file                    = fopen('tmp/archivos_testing.txt', "r");
$totalArchivos    = (int)fread($file, filesize('tmp/archivos_testing.txt'));
fclose($file);

$categoriasIniciales    = scandir($directorioTraining);

$matrizResultadosFrecuencia   =
iterarTodasCategorias($directorioClasificados.'/frecuencia');
$erroresFrecuencia                  =
erroresTipoIyII($matrizResultadosFrecuencia);
$indicadoresFrecuencia        = calcularIndicadores($erroresFrecuencia);

$matrizResultadosTF     =
iterarTodasCategorias($directorioClasificados.'/tf');
$erroresTF              = erroresTipoIyII($matrizResultadosTF);
$indicadoresTF          = calcularIndicadores($erroresTF);

$matrizResultadosTFIDF  =
iterarTodasCategorias($directorioClasificados.'/tfidf');
$erroresTFIDF                 = erroresTipoIyII($matrizResultadosTFIDF);
$indicadoresTFIDF       = calcularIndicadores($erroresTFIDF);

$tmpToPrint =   array('frecuencia'   =>   $indicadoresFrecuencia,   'tf'   =>
$indicadoresTF, 'tfidf' => $indicadoresTFIDF);

echo json_encode($tmpToPrint);


function iterarTodasCategorias($path) {
$matrizResultados = array();
```

```
$ignore     = array('.','..','cgi-bin','.DS_STORE');
$categorias = scandir($path);
foreach($categorias as $categoria){
if(in_array($categoria, $ignore))
continue;
if (is_dir(rtrim($path, '/').'/'.$categoria))
$matrizResultados[$categoria] =               procesarCategoria(rtrim($path,
'/').'/'.$categoria);
}
return $matrizResultados;
}

function procesarCategoria($path) {
global $contadorProcesados, $categoriasIniciales, $totalArchivos;
$tmpMatrizResultados = array();

$ignore     = array('.','..','cgi-bin','.DS_STORE');
foreach($categoriasIniciales as $tmpCategoria){
if(in_array($tmpCategoria, $ignore))
continue;
$tmpMatrizResultados[$tmpCategoria] = 0;
}

$ignore = array('.','..','cgi-bin','.DS_STORE');
$files = scandir($path);
foreach($files as $t) {
if(in_array($t, $ignore))
continue;
if (!is_dir(rtrim($path, '/').'/'.$t)){
$tmpCategoriaNombre     = explode('_', $t, 2);
$categoriaArchivo = $tmpCategoriaNombre[0];
$tmpMatrizResultados[$categoriaArchivo]++;
}

$contadorProcesados++;
$breadcrum  = fopen('tmp/archivos_evaluacion.php', "w");
fwrite($breadcrum,      '{"procesados":      '.$contadorProcesados.',"total":
'.$totalArchivos.'}');
```

```
fclose($breadcrum);
}
return $tmpMatrizResultados;
}

function erroresTipoIyII($matrizResultados){
global $totalArchivos, $categoriasIniciales;
$falsosPositivos        = array();
$falsosNegativos        = array();
$verdaderosPositivos    = array();
$verdaderosFalsos       = array();

$ignore     = array('.','..','cgi-bin','.DS_STORE');
foreach($categoriasIniciales as $categoria){
if(in_array($categoria, $ignore))
continue;
$verdaderosPositivos[$categoria]    = 0;
$falsosNegativos[$tmpCategoria]           = 0;
$falsosPositivos[$categoria]        = 0;
}

foreach($matrizResultados as $categoria => $tmpResultados){
foreach($tmpResultados as $tmpCategoria => $valor){
if($categoria != $tmpCategoria){
$falsosNegativos[$tmpCategoria]     += $valor;
$falsosPositivos[$categoria]  += $valor;
}else{
$verdaderosPositivos[$categoria]    += $valor;
}
}
}

foreach($matrizResultados as $categoria => $tmpResultados){
$verdaderosFalsos[$categoria] =               $totalArchivos               -
$verdaderosPositivos[$categoria]     -     $falsosPositivos[$categoria]     -
$falsosNegativos[$tmpCategoria];
}
```

```
return array($falsosNegativos, $falsosPositivos, $verdaderosFalsos,
$verdaderosPositivos);
}

function calcularIndicadores($errores){
$indicadorPrecision     = calcularPrecision($errores);
$indicadorRecall  = calcularRecall($errores);
$indicadorAccuracy      = calcularAccuracy($errores);
$indicadorError         = calcularError($errores);
$indicadorF1            =       calcularF1($errores,       $indicadorRecall,
$indicadorPrecision);

return array('accuracy' => $indicadorAccuracy, 'error' => $indicadorError,
'f1' => $indicadorF1, 'precision' => $indicadorPrecision, 'recall' =>
$indicadorRecall);
}

function calcularPrecision($errores){
$indicadorPrecision           = array();
$falsosNegativos        = $errores[0];
$falsosPositivos        = $errores[1];
$verdaderosFalsos       = $errores[2];
$verdaderosPositivos    = $errores[3];

foreach($verdaderosPositivos as $categoria => $foo){
$indicadorPrecision[$categoria]     = round($verdaderosPositivos[$categoria]
/ ($verdaderosPositivos[$categoria] + $falsosPositivos[$categoria]), 4);
}

return $indicadorPrecision;
}

function calcularRecall($errores){
$indicadorRecall        = array();
$falsosNegativos        = $errores[0];
$falsosPositivos        = $errores[1];
$verdaderosFalsos       = $errores[2];
$verdaderosPositivos    = $errores[3];
```

```
foreach($verdaderosPositivos as $categoria => $foo){
$indicadorRecall[$categoria]  =  round($verdaderosPositivos[$categoria]  /
($verdaderosPositivos[$categoria] + $falsosNegativos[$categoria]), 4);
}

return $indicadorRecall;
}

function calcularAccuracy($errores){
global $totalArchivos;

$indicadorAccuracy           = array();
$falsosNegativos        = $errores[0];
$falsosPositivos        = $errores[1];
$verdaderosFalsos       = $errores[2];
$verdaderosPositivos    = $errores[3];

foreach($verdaderosPositivos as $categoria => $foo){
$indicadorAccuracy[$categoria]      =
round(($verdaderosPositivos[$categoria]  +  $verdaderosFalsos[$categoria])  /
$totalArchivos, 4);
}

return $indicadorAccuracy;
}

function calcularError($errores){
global $totalArchivos;

$indicadorError         = array();
$falsosNegativos        = $errores[0];
$falsosPositivos        = $errores[1];
$verdaderosFalsos       = $errores[2];
$verdaderosPositivos    = $errores[3];

foreach($verdaderosPositivos as $categoria => $foo){
```

```
$indicadorError[$categoria]   =    round(($falsosNegativos[$categoria]    +
$falsosPositivos[$categoria]) / $totalArchivos, 4);
}

return $indicadorError;
}

function calcularF1($errores, $indicadorRecall, $indicadorPrecision){
$indicadorF1      = array();

foreach($indicadorRecall as $categoria => $foo){
$indicadorF1[$categoria]      =  round(2  *  $indicadorRecall[$categoria]  *
$indicadorPrecision[$categoria]    /    ($indicadorRecall[$categoria]    +
$indicadorPrecision[$categoria]), 4);
}

return $indicadorF1;
}
?>


<?php
ini_set('max_input_time',-1);
ini_set('upload_max_filesize','1024M');
ini_set('post_max_size','1024M');
ini_set('max_file_uploads',1000);
ini_set('memory_limit','4096M');
ini_set('max_execution_time',0);

$REPRESENTACION         = array();  //$REPRESENTACION[categoria]         -->
{[0]:{Frecuencia}, [1]:{TF}, [2]:{TF-IDF}}
$directorioTraining     = 'repositorio_training';
$contadorProcesados     = 0;
$totalArchivos          = 0;

$file                   = fopen('tmp/archivos_training.txt', "r");
$totalArchivos    =                                    (int)fread($file,
filesize('tmp/archivos_training.txt'));
```

```
fclose($file);

function iterarTodasCategorias($path) {
global $REPRESENTACION;

$ignore     = array('.','..','cgi-bin','.DS_STORE');
$categorias = scandir($path);
foreach($categorias as $categoria){
if(in_array($categoria, $ignore))
continue;
if (is_dir(rtrim($path, '/').'/'.$categoria))
$REPRESENTACION[$categoria]   =               procesarCategoria(rtrim($path,
'/').'/'.$categoria, $categoria);
}

foreach($categorias as $categoria){
if(in_array($categoria, $ignore))
continue;

$TFIDF                        = array();
$tmp_tf                 = $REPRESENTACION[$categoria][1];
$cantidadCategorias     = count($categorias);

foreach($tmp_tf as $palabra => $cantidad){
$cantidadPalabraEnOtrasCategorias = 0;          //La cantidad de categorias
que contienen la palabra de la categoria i (incluyendose)
foreach($categorias as $tmp_categoria){
if(in_array($tmp_categoria, $ignore))
continue;
if(array_key_exists($palabra, $REPRESENTACION[$tmp_categoria][1]))
$cantidadPalabraEnOtrasCategorias++;
}
$TFIDF[$palabra]     =     $cantidad     *     log($cantidadCategorias     /
$cantidadPalabraEnOtrasCategorias);
}
$REPRESENTACION[$categoria][2] = $TFIDF;
}
}
```

```
function procesarCategoria($path, $categoria) {
global $contadorProcesados;
$FRECUENCIA                          = array();
$TF                                  = array();

$ignore = array('.','..','cgi-bin','.DS_STORE');
$files = scandir($path);
foreach($files as $t) {
if(in_array($t, $ignore))
continue;
if (!is_dir(rtrim($path, '/').'/'.$t)){
$rutaArchivo            = rtrim($path, '/').'/'.$t;
$file                          = fopen($rutaArchivo, 'r');
$contenidoArchivo = fread($file, filesize($rutaArchivo));
fclose($file);

$cantidadPalabrasArchivo      = contarPalabrasEnArchivo($contenidoArchivo);
$FRECUENCIA                          =            sumaAsociativa($FRECUENCIA,
$cantidadPalabrasArchivo);
}

$breadcrum  = fopen('tmp/archivos_representacion.php', "w");
fwrite($breadcrum,      '{"procesados":      '.$contadorProcesados.',"total":
'.$totalArchivos.'}');
fclose($breadcrum);
}

$cantidadPalabrasCategoria    = count($FRECUENCIA);
foreach($FRECUENCIA as $palabra => $cantidad)
$TF[$palabra]     = $cantidad/$cantidadPalabrasCategoria;

$tmp_representacion     = array($FRECUENCIA, $TF, array());
return $tmp_representacion;
}

function sumaAsociativa($array1, $array2){
if(count($array1) == 0)
```

```
return $array2;
if(count($array2) == 0)
return $array1;
if(count($array1) >= count($array2)){
$tmp  = $array1;
$array1 = $array2;
$array2     = $tmp;
}

foreach($array1 as $palabra1 => $cantidad1){
if(array_key_exists($palabra1, $array2))
$array2[$palabra1]      += $cantidad1;
else
$array2[$palabra1]      = $cantidad1;
}
return $array2;
}

function contarPalabrasEnArchivo($contenidoArchivo){
$array_contenidoArchivo =        str_word_count($contenidoArchivo,        1,
'áéíóúÁÉÍÓÚñÑ1234567890');
$unico_contenidoArchivo = array_count_values($array_contenidoArchivo);
return $unico_contenidoArchivo;
}

function guardarRepresentacion($REPRESENTACION){
include('connect.php');
$superSQL   = 'TRUNCATE TABLE representacion_entrenamiento;';
$result     = $mysqli->query($superSQL);
$contador   = 0;
foreach($REPRESENTACION as $categoria => $tmpRepresentaciones){
foreach($tmpRepresentaciones[0] as $palabra => $frecuencia){
$tf               = $tmpRepresentaciones[1][$palabra];
$tfidf            = $tmpRepresentaciones[2][$palabra];
$superSQL   = 'INSERT INTO representacion_entrenamiento (categoria, palabra,
frecuencia,   tf,   tfidf)   VALUES   ("'.$categoria.'",   "'.$palabra.'",
'.$frecuencia.', '.$tf.', '.$tfidf.');';
$result     = $mysqli->query($superSQL);
```

```
if($mysqli->affected_rows == 0)
echo $superSQL;
}
}
}

iterarTodasCategorias($directorioTraining);
guardarRepresentacion($REPRESENTACION);
echo 1;
?>


<!DOCTYPE html>
<html>
<head>
<?php include('header.php'); ?>
</head>

<body>
<div data-role="page" id="page-entrenar">
<div data-role="header">
<h1>Clasificaci&oacute;n Autom&aacute;tica de Textos Biling&uuml;es</h1>
</div><!-- /header -->

<div data-role="content">
<h1>Entrenamiento</h1>
<div data-role="collapsible-set" data-theme="a" data-content-theme="a">
<div data-role="collapsible" data-collapsed="false">
<h3>Traducir Archivos</h3>
<p>Traduce los archivos que han sido subidos desde el idioma origen al
pivote (Ingl&eacute;s). Elija entre traducir en modo batch o ahora. <i>(Si
decide realizarlo ahora y elige tambi&eacute;n traducir en batch, ambas
acciones ocurrir&aacute;n)</i></p>
<div id="contenedorTraducir" data-role="fieldcontain">
<label for="slider-traducir" class="ui-hidden-accessible">Foo:</label>
<select name="slider-traducir" id="slider-traducir" data-role="slider">
<option value="0">No Traducir</option>
<option value="1" selected="selected">Traducir</option>
```

```
</select>
</div>
<div id="divisorTraduccionOpciones">--------- O ---------</div>
<a onclick="doTranslate();" href="#" data-role="button" data-theme="a" data-inline="true" data-icon="arrow-r" data-iconpos="right">Realizar Traducci&oacute;n Ahora</a>
<div class="contenedorBarraProgreso contenedorBarraProgreso_traduccion">
<div class="container">
<div class="progress_bar_traduccion ui-progress-bar ui-container">
<div class="ui-progress" style="width: 4%;">
<span class="ui-label" style="display:none;">Procesando <b class="value" id="archivos_procesados"></b> de un total de <b class="value" id="archivos_total">100 archivos</b></span>
</div><!-- .ui-progress -->
</div><!-- #progress_bar -->

<div class="content mensaje_barraProgreso" style="display: none;">
<p>Traducci&oacute;n Finalizada</p>
</div><!-- #mensaje_barraProgreso -->
</div><!-- #container -->
</div>
</div>
<div data-role="collapsible">
<h3>Dividir Documentos</h3>
<p>Los documentos deben ser divididos 2 grupos: en <strong>Entrenamiento</strong> y <strong>Pruebas</strong>. Debe elegir el porcentaje de los documentos existentes ser&aacute; para entreamiento y el complemento para pruebas. Se recomienda un ratio <strong>60% / 40%</strong> entre el primero y el segundo.</p>
<div id="contenedorPorcentajesDividirDocumentos">
<div id="contenedorPorcentajeEntrenamiento">60% Entrenamiento</div>
<div id="contenedorDivisorDividirDocumentos">::</div>
<div id="contenedorPorcentajePruebas">40% Pruebas</div>
</div>
<div id="contenedorDividirDocumentos">
<label for="slider_dividirDocumentos_entrenamiento" class="ui-hidden-accessible">Foo:</label>
```

```
<input onchange="doUpdatePorcentajeEntrenamientoPruebas()" type="range"
name="slider_dividirDocumentos_entrenamiento"
id="slider_dividirDocumentos_entrenamiento" value="60" min="10" max="90"
style="display:none;" />
</div>
</div>
<div data-role="collapsible">
<h3>Eliminar Stopwords</h3>
<p>Los <i>stopwords</i> son palabras sin mayor significado como
art&iacute;culos, pronombres, preposiciones, ente otros.</p>
<div id="contenedorStopWords" data-role="fieldcontain">
<label for="slider_stopwords">Eliminar Stopwords:</label>
<select name="slider_stopwords" id="slider_stopwords" data-role="slider">
<option value="0">No Eliminar</option>
<option value="1" selected="selected">Eliminar</option>
</select>
</div>
</div>
<div data-role="collapsible">
<h3>Aplicar Wordnet</h3>
<p><i>Wordnet</i> permite relacionar los t&eacute;rminos dentro de un
documento, permitiendo la disiminuci&oacute;n del espacio vectorial
generado..</p>
<div id="contenedorStopWords" data-role="fieldcontain">
<label for="slider_wordnet">Aplicar Wordnet:</label>
<select name="slider_wordnet" id="slider_wordnet" data-role="slider">
<option value="0" selected="selected">No Aplicar</option>
<option value="1">Aplicar</option>
</select>
</div>
</div>
</div> <!-- /collapsible-set -->
<a onclick="$('#contenedorPopup').fadeIn();" id="boton-startEntrenamiento"
href="#" data-role="button" data-theme="a" data-inline="true" data-
icon="arrow-r" data-iconpos="right">Empezar Entrenamiento en base a las
opciones elegidas...</a>
<div id="contenedorBotonEjecutar">
</div><!-- /content -->
```

```
<div data-role="footer">
<h4>Desarrollado e Implementado por <strong>Nelson Salazar</strong></h4>
</div><!-- /footer -->

</div><!-- /page -->

<!-- El popup del entrenamiento -->
<div id="contenedorPopup" style="display:none;">
<div id="contenidoPopup">
<h1>Entrenamiento</h1>
<div  class="contenedorBarraProgreso  contenedorBarraProgreso_entrenamiento"
style="display: block">
<div class="container">
<div   class="progress_bar_entrenamiento   ui-progress-bar   ui-container"
style="height: 60px">
<div class="ui-progress" style="width: 7%">
</div><!-- .ui-progress -->
</div><!-- #progress_bar -->

<div class="content mensaje_barraProgreso" style="display: block;">
<p> </p>
</div><!-- #mensaje_barraProgreso -->
</div><!-- #container -->
</div>
<ul                  data-role="listview"                  data-inset="true"
id="resumenPorcentajesEntrenamiento">
<li>Traducci&oacute;n     de     Documentos     <span     class="ui-li-count"
id="porcentajeTraduccion">0%</span></li>
<li>Dividir Documentos en Entrenamiento y Pruebas <span class="ui-li-count"
id="porcentajeDivision">0%</span></li>
<li>Eliminar          Stopwords          <span          class="ui-li-count"
id="porcentajeStopwords">0%</span></li>
<li>Representacion  de  Documentos  (Entrenamiento)<span  class="ui-li-count"
id="porcentajeRepresentacion">0%</span></li>
<li>Testing     de     la     Representacion     <span     class="ui-li-count"
id="porcentajeTesting">0%</span></li>
</ul>
```

```
<p id="tiempoTranscurridoEntrenamiento">Tiempo trascurrido: <span class="ui-
label">00:00:00</span></p>
<div style="margin-left:auto; margin-right:auto; width: 510px;">
<a    id="botonCancelEntrenamiento"    href="#"    data-role="button"    data-
inline="true"                                                data-icon="delete"
onclick="doStopEntrenamiento()">Cancelar</a>
<a id="botonInEntrenamiento" href="#" data-role="button" data-inline="true"
data-icon="check" onclick="doStartEntrenamiento()">Empezar Entrenamiento</a>
</div>
</div>
</div>
</body>
</html>
```