ClaNFi

Clasificación de Noticias Financieras
Informe de Proyecto de Grado

Sebastián Pizard

Tutores
MSc Guillermo Moncecchi
MSc Diego Garat

Tutores Externos
Ing Camilo Cerchiari
Ing Martín Varela
Ing Javier Frank

Instituto de Computación
Facultad de Ingeniería
Universidad de la República Oriental del Uruguay

Agosto 2007

Resumen

Predecir el comportamiento de los mercados financieros ha sido siempre el objetivo de miles de analistas de mercados. Estos profesionales monitorean en tiempo real los mercados, realizan seguimientos de cotizaciones, leen noticias financieras, crean proyecciones a corto y largo plazo, etc. Todo esto con el fin de encontrar oportunidades de inversión que sean en lo posible rentables y sin demasiado riesgo asociado. La globalización permite que la información de negocios se genere y publique de forma masiva en tiempo real. De este modo, el objetivo de este proyecto es facilitar el análisis financiero, clasificando en tiempo real las noticias publicadas en relación al impacto que podrían llegar a ocasionar sobre el mercado.

En primer lugar se realizó un estudio del estado del arte en referencia a la clasificación de documentos y un relevamiento en busca de trabajos previos relacionados al análisis automático de noticias financieras. El proceso incluyó una etapa de construcción del corpus de referencia. Fueron recolectadas miles de noticias en formato RSS a partir de varios sitios financieros de la World Wide Web.

En una fase de especificación de la solución, se seleccionó una arquitectura flexible y modularizada sobre la cual se construyó la herramienta. Se seleccionaron varios algoritmos de clasificación (Bayesiano Simple, Winnow Positivo y Weighted Majority) y varios modelos para representar los documentos (Palabras Claves, tokens de Freeling, HMFrames, entre otros).

Como resultado se presenta una herramienta, realizada en lenguaje Java utilizando software libre, que analiza y clasifica noticias financieras. En un primer análisis se reconocen las acciones involucradas en la noticia, luego se realiza una predicción del impacto de la noticia sobre el valor de la acción referenciada; de este modo se indica si la noticia tendrá un impacto positivo o negativo. Las pruebas indican que dada una noticia donde se identifica una acción, es posible indicar con una precisión de un poco más del 60 % si la noticia tendrá un impacto positivo o negativo sobre el valor de la acción en el mercado bursátil.

Palabras Clave: Clasificación Automática, Noticias Financieras, Procesamiento de Lenguaje Natural, Aprendizaje Automático

Tabla de Contenidos

Capítulo 1

Introducción

Cada día surgen miles de noticias financieras en el mundo. El trabajo de los inversores se dificulta enormemente a raíz de la cantidad de información que deben considerar a la hora de seleccionar acciones sobre las cuales invertir.

Tradicionalmente las noticias financieras se transmitían mediante la prensa escrita o a través de llamados telefónicos. En este ámbito el periodismo financiero era considerado un tema de menor interés.

En la última década del siglo XX en Estados Unidos la información financiera deja de ser un asunto de especialistas, para convertirse en un tipo de información seguida por millones de personas. Sobre todo, la televisión e Internet fueron responsables de que el creciente número de pequeños inversores individuales siguieran con avidez las noticias de canales temáticos como la CNBC o los comentarios de especialistas en sitios web como TheStreet.com [Kur00].

En este contexto el periodismo financiero evoluciona, se transforma, deja su papel de relator de crónicas y pasa a ser un actor más del mundo financiero. De modo que se inventan noticias, se manipula información, se publican grandes rumores sin confirmación. En unos años de gran optimismo económico, gran impulso financiero y extraordinarias expectativas empresariales abiertas por las nuevas tecnologías, el periodismo financiero pasa a cumplir un rol mucho más activo a cualquiera que haya tenido antes.

Todos estos cambios ocurrían propiciados por un incremento en la inversión; en el año 1999 alrededor de 11 millones de estadounidenses mantenían inversiones financieras desde internet. De este modo, y citando a Howard Kurtz [Kur00], "El mercado pasa a ser ahora una parte integral de la cultura pop Americana".

En la actualidad la información financiera abunda: desde canales de televisión dedicados exclusivamente al área, pasando por sitios web financieros, periódicos especializados, blogs, grupos de noticias. Dentro de este amplio espectro, se encuentran los boletines de noticias RSS[1]. La familia de formatos

[1] RSS (*Really Simple Syndication*) es un formato para la sindicación de contenidos de páginas web. En este marco sindicación se entiende como la publicación de artículos simultáneamente en diferentes medios a través de una fuente a la que pertenecen.

RSS, surgida también a comienzos del siglo XXI, permite que los usuarios utilicen una única aplicación para recibir periódicamente el contenido que publican los distintos sitios web en los cuales están interesados. De esta forma, las noticias financieras llegan con un formato preestablecido, de forma periódica, a una sola aplicación, donde el inversor puede leerlas, descartarlas o ignorarlas.

Aún con las ventajas obvias que proporciona este mecanismo, los cientos de noticias que se generan por sitio web financiero diariamente hacen que sea casi imposible su lectura en tiempo real. Por otro lado, la actividad en los mercados bursátiles sí es en tiempo real y, de una forma u otra, toda la información disponible es considerada. De esta forma, estamos ante una situación donde la información es tan abundante que muchas veces no es posible su procesamiento a tiempo.

En este trabajo se propone una herramienta que actúa como lo haría un experto en finanzas; estableciendo qué acciones involucra la noticia y su impacto sobre el mercado, ya sea de signo positivo o negativo.

1.1. Objetivo

El proyecto se enmarca dentro del objetivo general de lograr predecir el comportamiento de una acción (como instrumento financiero), a través de la captura de la percepción que el mercado tiene de su comportamiento futuro. La intención es utilizar, entre otras fuentes de información, la opinión de los analistas financieros, con el fin de modelar tal percepción y su efecto sobre los actores del mercado.

De esta forma se plantea el objetivo especifico de construir una herramienta que permita el análisis automático de noticias financieras de modo de identificar las acciones referenciadas en su contenido y clasificar cada noticia en términos de su impacto estimado sobre el valor de la acción identificada en el mercado, indicando si la noticia tendrá un impacto positivo o negativo.

Quizás sea un poco difícil entender la meta principal de este trabajo, por lo que a continuación se presenta un ejemplo motivador.

Supongamos que un usuario recibe un noticia como la presentada en el Cuadro [1.1]. En primer lugar identifica las acciones involucradas en el titulo y contenido de la noticia. En este caso, la empresa involucrada es Heinz, cuya acción es HNZ.

Luego, analiza de alguna forma la noticia, y trata de decidir si la noticia será positiva o negativa para la empresa involucrada, o sea Heinz. Inicialmente, el usuario lee atentamente la noticia y arma mentalmente un análisis de su impacto. En gran parte de los casos el texto de la noticia contiene mucha información relevante, a partir de la cual quizás ya sea posible evaluar si tendrá un impacto positivo o negativo sobre las acciones involucradas. Un analista financiero podría, además, hacer uso de su juicio experto, o quizás analizar la situación

```
<title> Heinz to cut 2,700 jobs and $355m in costs </title>
<date> 01-06-2006 12:22:00 -0300 </date>
<description>      Heinz,the consumer products group best known
for its tomato ketchup, announced plans to cut 8 per cent of
its workforce in a bid to achieve savings of more than $355m
in the next two years to restore profits and sales growth.
</description>
<channel> US companies news - FT.com </channel>
```

```
<title> Heinz recorta 2,700 puestos de trabajo y $355m en costos
</title>
<date> 01-06-2006 12:22:00 -0300 </date>
<description>           Heinz, el grupo de productos de consumo
mejor conocido por su salsa de tomate, anuncia planes para
recortar un 8 por ciento de su fuerza de trabajo con el
fin de alcanzar ahorros de más de $355m en los dos próximos
años y así restaurar beneficios y crecimiento en las ventas.
</description>
<channel> US companies news - FT.com </channel>
```

Cuadro 1.1: Ejemplo de Noticia RSS.

bursátil previa de esas empresas o del sector industrial al cual pertenecen.

En la noticia de ejemplo, la situación planteada es bastante ambigua, lo que dificulta enormemente que un usuario, como cualquiera de nosotros, pueda decidir solamente leyendo una noticia si su impacto será positivo o negativo. De hecho se pierde demasiado tiempo analizando cada noticia, lo cual ligado a la gran cantidad de noticias publicadas diariamente, deriva en que el análisis de una noticia cueste demasiado en relación al beneficio que puede obtenerse. Esto se debe principalmente a que son muy pocas las noticias que proporcionan oportunidades de inversión.

Adicionalmente se plantea el concepto de la *certeza* de ocurrencia de una noticia, él cual refiere a la certeza de ocurrencia del evento referido en la noticia. Por ejemplo: si la noticia se refiere a la suba de precios del petróleo en el día de ayer, la certeza será alta, ya que el evento ya ocurrió. La *certeza* está estrechamente vinculada al carácter especulativo de las noticias y es un tema bastante complicado de definir. Este tema quedó fuera del alcance final de este proyecto, ya que presenta problemas que deben ser resueltos con un análisis de texto más profundo (análisis sintáctico y semántico).

1.2. Abordaje Propuesto

El problema plantea diferentes dificultades y desafíos. En primer lugar, es necesario definir sobre qué tipo de noticias se va a trabajar. Así es que las noticias financieras que se consideran en este proyecto son aquellas que se reciben en suscripciones RSS, de modo de facilitar la tareas de recolección y unificación de formatos.

Luego de recolectadas las noticias, es requerido un esfuerzo mínimo para unificar formatos; incluyendo, también, la eliminación todo tipo de publicidad que existe agregada a su contenido. De este modo, se realiza un preprocesamiento adecuado.

Llega el momento donde es necesario identificar las acciones involucradas. Una tarea para nada fácil, luego del estudio de varias opciones se incluye una solución que no descuida perfomance, pero que permite la existencia de una lista de acciones a reconocer que puede ser actualizada por el usuario.

Por último, se debe realizar una categorización de la noticia, indicando si su impacto sobre el mercado, en referencia a las acciones involucradas, será negativo o positivo. Aquí es donde se encuentran las mayores dificultades. El problema de la clasificación de documentos es un tema bastante estudiado, pero en esta instancia surgen algunas complicaciones adicionales: las noticias están conformadas en realidad por un texto bastante escueto, lo que agrega, como veremos después, cierta dificultad a las tareas clasificación.

Adicionalmente, uno de los requerimientos establece que la aplicación construida soporte nuevas fuentes de noticias. Las nuevas fuentes de noticias RSS utilizarán en general el idioma inglés, aunque esto no está garantizado. Esto plantea un gran desafío, ya que distintas fuentes manejan distintos formatos de noticias: se deben descartar gran cantidad de técnicas de categorización basadas en reglas. Estas técnicas se basan en la detección de características comunes en la estructura de los documentos a estudiar.

Finalmente, se plantea el problema de la retroalimentación. O sea, si decidimos que una noticia tendrá un impacto positivo, ¿cómo podemos asegurar si estuvimos en lo cierto? La retroalimentación realizada por un experto en el área no fue posible, dado que no contamos en el proyecto con un analista financiero con disponibilidad para responder a nuestras consultas. A causa de esto, se propone un método de retroalimentación, él cual se realiza de forma automática a partir de información bursátil existente en la web y de algunos indicadores derivados del estado de cada acción luego de publicada cada noticia.

1.3. Guía de Lectura

El resto del informe está organizado de la siguiente forma: en el Capítulo 2 se definen algunos conceptos básicos sobre el dominio, sobre la representación y clasificación de documentos. En el Capítulo 3 se presenta un Clasificador de Noticias Financieras (ClaNFi), herramienta que permite descargar y procesar noticias financieras. En el capítulo 4 se presentan y analizan las pruebas realizadas. Luego, en el Capítulo 5 se presentan las conclusiones y se proponen posibles líneas de trabajo a futuro. Finalmente en el Anexo A se presenta el cronograma del proyecto.

Capítulo 2

Marco Teórico

En este capítulo se realiza una breve introducción al dominio del problema: el ámbito financiero, él cual en general es bastante difícil de comprender, de modo que aquí se hace una reseña centrada en la teoría del Mercado Eficiente, que relaciona indirectamente los movimientos bursátiles con el periodismo financiero. Quizás sea conveniente recordar al lector la existencia de un glosario al final del informe donde se describen algunos conceptos utilizados en la siguiente sección.

En la sección [2.2] se realiza una introducción al problema de la clasificación y luego en la sección [2.3] se hace referencia a la representación de documentos en general, detallando que problemas y posibilidades pueden surgir en soluciones especificas como la buscada en el marco de este proyecto.

2.1. Sobre el Dominio

Nuestro sentido común nos indica que el valor de una acción en el mercado surge como el resultado de analizar la información que se encuentra disponible en referencia a ella. Los comentarios y noticias sobre compañías, ciertas o no, mejor o peor fundadas, pueden mover —de hecho muchas veces mueven— las cotizaciones bursátiles.

Existe una manera formal de plantear esta idea, llamada la teoría del Mercado Eficiente. Esta teoría ha sido bastante usada como referencia, por ejemplo: en un trabajo realizado sobre categorización temática de noticias financieras James Thomas [Tho03] plantea el uso de este enfoque. Como veremos más adelante parece bastante atractivo utilizar esta teoría como base de todo trabajo que se relacione con la recuperación de información para la asistencia de tomas de decisiones financieras.

La idea base del Mercado Eficiente es simple y se puede resumir como sigue: *El mercado procesa eficientemente toda la información relevante en un único precio.* De esta forma, se afirma que el precio de los activos negociados en los mercados financieros refleja toda la información conocida por los miembros del

mercado y todas las creencias de los inversores sobre el futuro [MEF07]. O esa, la teoría del Mercado Eficiente establece que el valor de una acción surge como el efecto producido en el mercado por el conocimiento de toda la información disponible de esa acción, incluyendo precios históricos, reportes anuales, noticias, rumores, etc. No es posible, entonces, predecir de forma consistente los efectos del mercado, excepto a través de la suerte o de información privilegiada.

Sin embargo, también existen anomalías, patrones inexplicables en el comportamiento del mercado. Estos patrones permiten que sea posible obtener un beneficio extraordinario. Algunas de las anomalías más conocidas incluyen efectos del tamaño, donde las empresas más pequeñas ofrecen mayor retorno de la inversión que las empresas grandes; y efectos del calendario —como por ejemplo el *January Effect* (Efecto Enero)— que parece indicar que los mayores retornos a la inversión pueden ganarse en el primer mes del año.

Dentro de la teoría del mercado eficiente, se distinguen tres hipótesis, en función de lo que se entienda por información disponible:

Eficiencia Débil (*Weak Form Efficiency*): La información relevante es únicamente el histórico de precios. Los precios incorporan la información que se deriva de la evolución histórica de las cotizaciones y volúmenes. No se toma en cuenta otro tipo de información disponible: ni pública (informes anuales, etc), ni privilegiada (por ej. detalle de anuncios antes de que se hagan públicos).

Eficiencia Semi-Fuerte (*Semi-Strong Form*): La información relevante es toda la información pública disponible. Es decir, los precios no incluyen sólo la información que hace referencia a los volúmenes y precios, sino también la referente a sus fundamentos (crecimiento en resultados, situación financiera, situación competitiva, etc). No se toma en cuenta la información privilegiada.

Eficiencia Fuerte (*Strong Form*): La información relevante se compone de toda la información pública disponible y de la información privilegiada, o sea, toda información es relevante. Los precios incorporan toda la información referente a una empresa, incluso la privada o privilegiada.

La teoría Mercado Eficiente es bastante compleja, es posible encontrar una explicación más detallada por parte de LeBaron y Vaitilingam [DL01].

La mayoría de los economistas trabajan sobre la base de una Eficiencia Débil, aunque a la hora de invertir se debe tener en cuenta toda la información con la que se dispone.

Este trabajo, al igual que el presentado por Thomas [Tho03] se construye sobre la teoría del Mercado Eficiente en su forma de Eficiencia Semi-Fuerte. Esto implica que consideraremos que toda la información relevante para el mercado es aquella de carácter público y disponible. De esta forma, se pretende utilizar parte de toda esa información, recuperando noticias financieras en formato RSS

y presentarlas al usuario. Antes de presentar cada noticia, se analiza su contenido de forma de intentar agregar información útil. Así se identifican acciones involucradas y se realiza una predicción que indica si cada noticia tendrá un impacto positivo o negativo sobre el mercado.

2.2. Clasificación de Documentos

Uno de los principales objetivos del proyecto es la construcción de una herramienta que sea capaz de categorizar una noticia, indicando si su impacto sobre el mercado, en relación a los activos negociables (acciones) involucrados, será negativo, positivo o neutro. Como se explicó previamente en la introducción, aquí es donde surgen mayores problemas. Aún cuando el problema de la clasificación de documentos es un tema bastante estudiado, se encuentran algunas dificultades adicionales. Esta sección introduce brevemente al problema de la clasificación de documentos en general, y presenta los fundamentos sobre los cuales se basa la aplicación construida. Se describe una serie de métodos de clasificación y se discute su adaptación al problema planteado.

2.2.1. Clasificación y Representación

La clasificación de documentos se refiere a la tarea de asignar un documento a una o más categorías, basándose en su contenido. Por ejemplo, podemos clasificar noticias financieras en categorías que dependan de su contenido. Así, una noticia que trata sobre un anuncio de planes, como es el caso de nuestro ejemplo (donde Heinz anuncia planes para recortar gastos), la categoría podría ser Anuncio de Planes.

La mayoría de los algoritmos de clasificación no trabajan sobre el texto del documento, sino que utilizan una versión simplificada o que ha sido adaptada previamente. En una etapa previa a la clasificación, generalmente, se transforma cada documento a una forma más simple. De esta manera, la prosa de cada documento es vectorizada: se identifica la unidad mínima de cada documento —a la que llamaremos rasgo (*feature*)— y se construye un vector. A partir de ese momento todas las tareas de clasificación o análisis se realizan sobre el vector obtenido para cada documento.

En el Cuadro [2.1] se plantea el texto de la noticia de ejemplo presentada previamente y se incluye una posible representación basada en Palabras Clave, como veremos más adelante, esta forma de representar los documentos utiliza cada palabra como rasgo.

Mas adelante, en la sección [2.3] se plantean algunos conceptos adicionales asociados a la representación de documentos.

Texto original	Heinz, the consumer products group best known for its tomato ketchup, announced plans to cut 8 per cent of its workforce in a bid to achieve savings of more than $355m in the next two years to restore profits and sales growth.
Palabras Clave	{ Heinz, consumer, products, group, best, known, tomato, ketchup, announced, plans, cut, 8, per, cent, workforce, bid, achieve, savings, $355m, next, two, years, restore, profits, sales, growth }

Cuadro 2.1: Ejemplo de Representación de Documentos

Perfil de la clase Anuncio de Planes	{ announced, plans }

Cuadro 2.2: Ejemplo de Perfil de Clase

2.2.2. Perfiles

La extracción de rasgos de un documento y su normalización deriva, como se mostró anteriormente, en un conjunto de rasgos, el cuál se denomina *perfil del documento*. De igual forma, una clase de documentos relacionados semánticamente se puede caracterizar con un conjunto (ponderado) de rasgos, obteniéndose un *perfil de clase*. Esto puede verse como un conjunto de rasgos que es compartido (o que caracteriza) a todos los documentos de esa clase.

Siguiendo la noticia de ejemplo, podríamos definir un perfil para la clase de documentos que tratan de "Anuncio de Planes", de esta forma debemos establecer una lista de rasgos que identifiquen a los documentos de esta clase. En el Cuadro [2.2] se presenta una lista de rasgos que podríamos considerar como el perfil de la clase "Anuncio de Planes". De esta forma cuando encontremos un documento nuevo y lo queramos asignar a alguna clase, lo único que debemos hacer es analizar con cuál de los perfiles de clase tiene más coincidencias.

El perfil de clase del ejemplo anterior fue creado manualmente; en la mayoría de los casos un experto en el área de interés, utilizando un gran conjunto de documentos, determina cuáles son las características de los documentos asociados a cada clase, y determina así los perfiles de las clases a utilizar en la clasificación.

Los sistemas de clasificación basados en perfiles creados manualmente pueden alcanzar una alta precisión; pero tienen un costo demasiado alto, más aún si pensamos en lo rápido que pueden cambiar los criterios de clasificación. Es claro de ver, entonces, la importancia de obtener este clasificador de forma automática.

2.2.3. Una Taxonomía de los Algoritmos de Clasificación

En esta sección trataremos los algoritmos de *aprendizaje supervisado*, que son aquellos que pueden aprender perfiles de clases automáticamente a partir

de documentos ejemplo. En este contexto, cada documento tendrá una lista de rasgos, sobre los cuales se basa la clasificación. Los algoritmos de clasificación pueden trabajar sobre casi cualquier tipo de rasgos.

Podemos distinguir tres grandes categorías de algoritmos de clasificación automática [Kos06b]:

1. **Clasificadores basados en Reglas**
   Estos clasificadores aprenden infiriendo un conjunto de reglas (una *disyunción o conjunción de pruebas atómicas*, por ejemplo: "este rasgo tiene ese valor") a partir de documentos pre-clasificados.

   En el contexto del caso de estudio, podemos definir una regla que nos permita saber si una noticia pertenece o no a la clase "Anuncio de Planes". En este caso podemos crear una regla que establezca que si el texto *announces plans to* es encontrado en el texto, entonces la noticia pertenece a la clase "Anuncio de Planes".

   En general los clasificadores basados en reglas son bastantes rígidos, ya que ante un cambio sustancial en el formato o contenido de los documentos a clasificar, se deben revisar y ajustar todas las reglas utilizadas.

2. **Clasificadores Lineales**
   En estos algoritmos, para cada clase es computado un *perfil de clase*: un vector de pesos, uno por cada rasgo, basado en la frecuencia de ocurrencia. Para cada clase y documento, un puntaje es obtenido a partir de comparar de alguna forma el perfil de la clase y el perfil del documento.

   Siguiendo con el ejemplo, podemos utilizar un peso clásico, el cual asigna un punto por cada rasgo definido en el perfil de la clase que aparezca en el perfil del documento. De este modo, el puntaje obtenido de comparar la noticia de ejemplo y la clase "Anuncio de Planes" es 2. Este puntaje surge de sumar un uno al puntaje por cada una de las siguientes palabaras: *announced* y *plans*, las cuales se encuentran en el perfil de la clase (Cuadro [2.2]) y también en el perfil de la noticia (Cuadro [2.1]).

3. **Clasificadores basados en casos**
   Estos algoritmos clasifican un nuevo documento, a partir de buscar k documentos cercanos al mismo en el conjunto de entrenamiento y hacer algún tipo de votación por mayoría de las clases de esos vecinos cercanos.

   Para utilizar un algoritmo de clasificación basado en casos es necesario definir el concepto de cercanía (o distancia): se debe establecer una métrica que permita medir cuán similares son dos documentos. De este modo, cuando llega un documento nuevo se asigna a la clase a la cuál pertenezcan la mayoría de sus k vecinos más cercanos.

En el marco del proyecto se plantearon dos grandes restricciones: por un lado las noticias RSS no cuentan con un texto demasiado largo, generalmente no exceden las 200 palabras por noticia. Por otro lado, la herramienta a construir debe permitir cambiar la fuente de noticias con relativa facilidad. Por estas causas el enfoque del presente trabajo se centra en los clasificadores lineales, desestimando los clasificadores basados en reglas por ser altamente dependientes de características inherentes a la fuente de documentos utilizada. Los clasificadores basados en casos tampoco fueron tomados en cuenta, como el lenguaje es muy amplio y las noticias tienen tan poco contenido, es muy difícil encontrar criterios para definir la noción de cercanía entre noticias.

2.2.4. Entrenamiento y Testeo

Conceptualmente, un algoritmo de clasificación tiene una *fase de entrenamiento*, en la cual los *perfiles de clase* son aprendidos a partir de un conjunto suficientemente grande de documentos de ejemplo preclasificados (el *conjunto de entrenamiento*). Luego se presenta una *fase de aplicación*, en la cual los perfiles de clase son usados para asignar los nuevos documentos a la clase más probable. En la Figura [2.1] se muestra un diagrama con los conceptos explicados anteriormente.

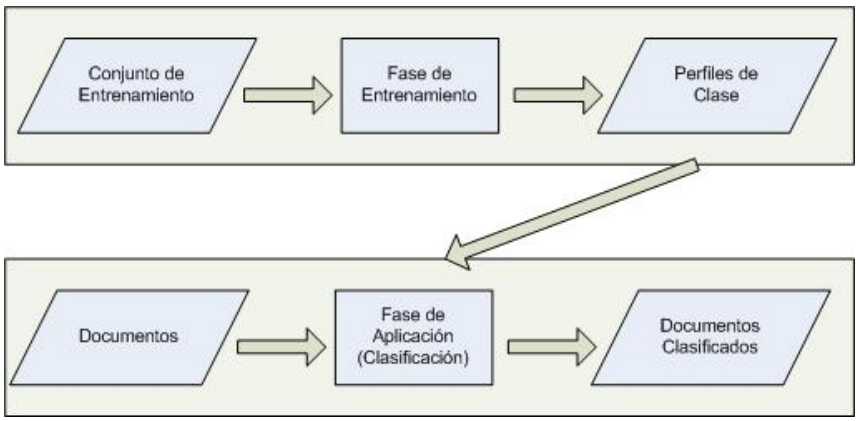

Figura 2.1: Fases de Entrenamiento y Aplicación

Se pueden distinguir dos tipos de clasificación:

1. **Mono Clasificación:** cada documento pertenece a solamente una de las clases existentes.

2. **Clasificación Múltiple:** cada documento pertenece al mismo tiempo a una o más clases.

Una medida de éxito de la clasificación es la *tasa de error*, la cual se define como la fracción de documentos asignados por el clasificador a alguna clase c, siendo errónea esa asignación.

Asumiendo una *clasificación binaria* (*relevante/no relevante*), entonces definimos las siguientes cantidades:

1. p^+ = nro. de documentos *relevantes* clasificados como *relevantes*.
2. p^- = nro. de documentos *relevantes* clasificados como *no relevantes*.
3. n^- = nro. de documentos *no relevantes* clasificados como *no relevantes*.
4. n^+ = nro. de documentos *no relevantes* clasificados como *relevantes*.

Obviamente, N el numero total de documentos se calcula como:

$$N = p^+ + n^+ + p^- + n^-$$

La relación entre las cantidades definidas previamente se muestra en la Figura [2.2].

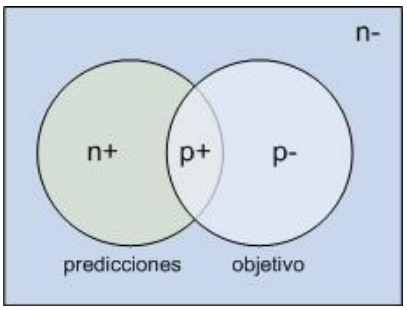

Figura 2.2: Precisión y Recuperación

Para la clase de documentos relevantes tenemos dos medidas de desempeño posibles, la Precisión (P) que indica la proporción de documentos que realmente son relevantes sobre todos los documentos señalados como tales; y la Recuperación o *Recall* (R), que indica la proporción de documentos relevantes señalados como relevantes, sobre todos los documentos relevantes existentes.

$$P = \frac{p^+}{p^+ + n^+}$$

$$R = \frac{p^+}{p^+ + p^-}$$

En un intento de obtener una única medida de desempeño, se usa la Medida-F (o *F-measure*); la cual se define como la media armónica de la Precisión y la Recuperación:

$$Medida - F = \frac{2PR}{P + R}$$

Esta medida también es llamada Medida-F_1, ya que la precisión y la recuperación son ponderadas de igual forma.

La formula general para un α no negativo es la siguiente:

$$Medida - F_\alpha = \frac{(1+\alpha)PR}{\alpha P + R}$$

Otras dos medidas-F conocidas son la medida-F_2, la cual duplica la ponderación de la recuperación sobre la precisión, y la medida-$F_{0,5}$, la cual duplica la ponderación de la precisión sobre la recuperación.

Cuando hay más de dos clases, usualmente se habla de promedio de precisiones. De este modo hay dos formas de obtener dichos promedios:

1. **Promedio Micro**
   Las cantidades p^+, n^+, p^- y n^- son sumadas sobre todas las clases. Este promedio estará dominado por las *clases grandes* (clases con gran cantidad de documentos), ya que al sumar cantidades las *clases grandes* se ponderan de mayor forma.

2. **Promedio Macro**
   La Precisión y la Recuperación se calculan sobre todas las clases y se dividen por el número de clases. Este promedio estará dominado por las *clases pequeñas* (clases con pocos documentos), ya que al sumar medidas de desempeño no importan las cantidades, entonces las *clases pequeñas* son ponderadas de igual forma que las *clases grandes* y como en general hay más *clases pequeñas* que *clases grandes*, este promedio estará dominado por las *clases pequeñas*.

En este trabajo se optó por trabajar con clasificadores binarios, ya que son más fáciles de construir y testear.

2.3. Representación de Documentos

Como se explicó en la sección anterior, generalmente el contenido de cada documento se transforma a una forma más simple y estructurada. De esta forma, se facilita y hace posible la ejecución de las siguientes etapas de procesamiento a realizar sobre el contenido de cada documento. La manera en que los documentos son representados juega un papel muy importante, tanto para el desempeño en tiempo y forma de cualquier algoritmo de clasificación que sea utilizado.

En este proyecto se consideró al tema de la representación de documentos como un tema de alta relevancia. Así es que se consideraron varias formas de representación y para algunas de estas formas se utilizaron mejoras planteadas en diferentes trabajos previos. Esta sección introduce algunos conceptos en el área que luego serán utilizados al explicar el diseño de la aplicación.

2.3.1. El problema

Con el objetivo de reducir la complejidad de los documentos y permitir una manipulación más sencilla, los documentos son transformados de su versión de *texto completo* a un *vector de documento*, el cual describe el contenido del documento. Como se explica anteriormente, este proceso consiste en identificar

la unidad mínima de cada documento (llamada rasgo o *feature*) y luego transformar cada documento a un vector de rasgos. Teóricamente es posible una definición más abstracta de un documento, de modo que un documento es la unión de los términos que tienen varios patrones de ocurrencia.

Como los patrones de ocurrencia son difíciles de definir y por ende calcular, son frecuentemente omitidos y en cambio se utilizan estadísticas de palabras sueltas (ej. índices de documentos). El pionero en el área de IR (*Information Retrieval*) Luhn [LUH58] utilizó la frecuencia de palabras como rasgos descriptivos del contenido de los documentos en varios de sus trabajos de la década de los años 50, siendo aún hoy uno de los métodos de descripción de documentos más utilizado.

Aunque esta es una técnica bastante utilizada, al representar un documento como una lista de palabras sueltas, se pierde gran parte del significado que surge en el sentido de los términos asociados, por ejemplo, cuando algunas palabras tienen diferente sentido al estar juntas (las llamadas Colocaciones Léxicas *Collocations*, por ejemplo: *trasplantar un órgano*) o son simplemente utilizadas de manera redundante (ej. *consenso de opinión*). Recientemente, se han implementado varios vectores de contexto que permiten explotar esos patrones de co-ocurrencia. Dentro de este marco veremos los HM Frames al final de la sección.

2.3.2. Palabras Clave

Tradicionalmente, la IR considera a un documento como un conjunto de palabras claves, omitiendo la mayor parte de la información proporcionada por la estructura lingüística. Formalmente se puede decir que:

Hipótesis de Palabras Clave. *Un documento es un conjunto de palabras, una consulta también es un conjunto de palabras; entonces si la palabra x ocurre en el documento, el documento trata sobre x [Kos06b].*

Esta representación es una de las más utilizadas para tareas de recuperación y clasificación de información en la actualidad. Es, por ejemplo, la base de grandes plataformas de búsqueda de documentos, como son Google y Yahoo!.

Es claro ver que al utilizar ésta representación el mayor conjunto del contenido semántico del documento se pierde. Gran parte o todo el sentido que se deriva de la estructura lingüística y sintáctica del documento se omite deliberadamente.

Aunque los resultados de realizar búsquedas o análisis basados en palabras claves han sido bastantes buenos, existen varias dificultades que deben ser contempladas [Kos06a]. Como los documentos son tratados como palabras sueltas, surgen dificultades en cuanto a que una palabra puede tener diferentes significados (*polisemia*), o que diferentes palabras pueden tener el mismo significado (*sinonimia*). El problema, tal como se presentó en la introducción al tema, consiste en la existencia de errores u omisiones que surgen como consecuencia de las simplificaciones realizadas. Representar un texto como una lista de palabras sueltas, hace que se pierda el sentido de la asociación entre las palabras. Entonces, tenemos por ejemplo que la colocación léxica *trasplantar un órgano* al ser

representada en palabras sueltas pierde el significado y trasplantar puede pasar a ser el verbo *trasplantar* en su sentido asociado a su significado más común: "Trasladar plantas del sitio en que están arraigadas y plantarlas en otro" [Esp06].

Para minimizar el impacto de éstos y otros aspectos, se han creado distintas mejoras a la representación basada en palabras clave, se enumeran algunas a continuación:

Stop Words: Se eliminan de los distintos índices las palabras que carecen de importancia (llamadas *Stops Words* o Palabras Vacías), por ser sumamente populares en las construcciones de texto, ej. ["de, en, los"].

Stemming: Método que permite igualar términos relacionados morfológicamente (por ejemplo: "caminar", "caminando" y "encaminado"). La idea central es reducir cada palabra a su *Radical* o *Stem*, eliminando posibles sufijos y prefijos.

Clustering: La idea es agrupar sinónimos en grupos de equivalencia, a fin de permitir igualarlas al momento realizar la clasificación o recuperación. Por ejemplo, al buscar por *roca* se devuelven también los documentos que contengan la palabra *piedra*.

Las mejoras presentadas anteriormente permiten incrementar en gran medida el desempeño de algoritmos de recuperación de información y de clasificación de documentos basados en la representación utilizando palabras clave [Kos06b]. Aun así se han planteado otros enfoques sobre el problema de la representación de documentos, a continuación se realiza una breve introducción a los modelos de representación que utilizan frases.

2.3.3. Frases

Este otro enfoque, la búsqueda basada en frases clave, establece que:

Hipótesis de Frases. *Un documento es un conjunto de términos, una consulta también es un conjunto de términos; entonces si el termino x ocurre en el documento, el documento trata sobre x.*

Donde un término puede ser una palabra [clave] o una frase. Cabe destacar que con este enfoque seguimos descuidando la estructura del discurso.

Aunque existen sólidos fundamentos teóricos avalan esta teoría, la búsqueda basada en frases ha sido encontrada decepcionante en reiteradas situaciones [CK06].

A pesar de que actualmente las *grupos nominales* son aceptadas como términos a indizar, los sistemas de IR con búsquedas basadas en frases no parecen tener mejor desempeño que aquellos que se basan en palabras clave. Aún así, se han realizados intentos de mejorar la representación de documentos mediante frases, un ejemplo son los HM Frames, que veremos a continuación.

Texto	I saw this man eating chips in the garden.

Árbol	{P:I ,[V:saw ,[N:man ,[V:eating ,N:chips]]|in N:garden]}
	
	`P:I, *`
	`/ \`
	`v:saw, * |in N:garden`
	`/ \`
	`n:man, *`
	`/ \`
	`V:eating, N:chips`

Lista HM Frames	["P:I", "V:saw"] ["V:saw", "N:man"] ["V:saw", "in N:garden"] ["N:man", "V:eating"] ["V:eating", "N:chips"]]

Cuadro 2.3: Ejemplo de representación de textos utilizando HM Frames

HM Frames

En el contexto de la clasificación de documentos, Koster [CK06] sugiere representar cada documento como una bolsa de términos, donde los términos son *Head/Modifier frame*. Estos HM frames son derivados de la frases del documento por un proceso de *transducción*. Cada frase se transduce en uno o más HM frames.

Un HM frame es un par denotado como: `[head, modifier]` donde ambos elementos (`head` y `modifier`) son consistentes con una palabra extraída del documento o una secuencia de estas palabras o (recursivamente) otros frames.

La intuición detrás de esto es que el `modifier` se une al `head` para hacerlo más preciso, como por ejemplo para distinguirlo de otros con el mismo significado (tratando de evitar la sinonimia). Por ejemplo, hay muchas formas de ingeniería, pero podemos enfocarnos en `[engineering, software]`.

De esta forma, es posible utilizar los HM Frames para construir representaciones alternativas de los documentos. Cada documento puede ser representado por un árbol de HM Frames o por una lista de HM Frames (realizando previamente un proceso de desanidamiento de árbol, *unnesting*). En el Cuadro [2.3] se presenta un caso que ejemplifica estas representaciones.

Koster y Verbruggen [CHAK] plantean los HM Frames como una alternativa a los modelos de representación de texto clásicos (por ejemplo: Palabras Clave). Aún así los trabajos previos con esta representación solamente incluyen tareas de extracción y recuperación de información [KT97]: por ejemplo el proyecto IRENA, cuyo objetivo era la extracción de información de artículos de música pop o el proyecto ELSA, que contemplaba la recuperación de información sobre

artículos de química; en ambos casos los autores indican una gran mejora en los tiempos de respuesta en relación a otras herramientas de procesamiento y recuperación de información.

Capítulo 3

ClaNFi

Como ya se presentó en el Capítulo 1, este proyecto se focaliza en el desarrollo de una herramienta de procesamiento y análisis de noticias financieras; más específicamente un Clasificador de Noticias Financieras, de ahora en más ClaNFi. La herramienta debe permitir la identificación de acciones en el contenido de las noticias y su clasificación en términos del tipo de impacto estimado sobre el valor de las acciones identificadas.

Durante la etapa de análisis de los requerimientos, se detectó la necesidad de construir una herramienta altamente flexible y fácilmente adaptable. Se requería, por un lado, que ClaNFi pudiera ser utilizado fácilmente por un usuario inexperto para obtener noticias financieras ya clasificadas según su impacto; pero además se requería que la herramienta fuera altamente adaptable, que se pudieran agregar o quitar fuentes de noticias.

De este modo, surgieron naturalmente dos usuarios diferentes, un usuario, inexperto, para el cual ClaNFi brinda las funcionalidades básicas de descarga y clasificación de noticias financieras según su impacto sobre el mercado; y un usuario más técnico, quién deberá, de ser necesario, agregar o quitar componentes de clasificación o de procesamiento de texto y fuentes de noticias. Para esto, ClaNFi proporciona mecanismos de ajuste permitiendo, también, agregar nuevos algoritmos.

ClaNFi provee una plataforma para la clasificación de documentos en general, incluyendo mecanismos para agregar, quitar, configurar y testear componentes de clasificación y de procesamiento de texto. Con este fin, el motor de ClaNFi es totalmente independiente de los algoritmos de clasificación y procesamiento que se utilicen. Adicionalmente, provee facilidades para agregar y ajustar nuevos componentes, por ejemplo: un componente que dada la implementación de un algoritmo de clasificación binaria, un conjunto de entrenamiento y un conjunto de testeo, realiza el entrenamiento, el testeo y el cálculo de medidas de desempeño (precisión, recuperación y medida-F_α) de forma automática.

Con el fin de crear una herramienta que soporte estos requerimientos, fue necesario separar las funcionalidades que se debían ofrecer según los perfiles de los usuarios; en este contexto surgen dos procesos bien definidos. Por un lado,

el Proceso de Clasificación y, por otro lado, el Proceso de Ajuste, los cuales se detallan a continuación.

3.1. Arquitectura

La arquitectura propuesta de ClaNFi, comprende dos grandes procesos que se presentan en la Figura [3.1]. En esta arquitectura se identifican dos procesos básicos, el *Proceso de Clasificación* y el *Proceso de Ajuste*.

El *Proceso de Clasificación* comprende la descarga de noticias RSS, la identificación de acciones, la construcción de representaciones según modelos y, finalmente, la clasificación de noticias según su impacto estimado. Éste es el proceso principal de la aplicación. El usuario interactúa solamente para iniciar o detener la descarga de noticias RSS y para iniciar el proceso de clasificación. Cuando finaliza el proceso de clasificación, la aplicación despliega los resultados obtenidos.

Por otro lado, existe un proceso accesorio, el *Proceso de Ajuste*. El *Proceso de Ajuste* permite ajustar los distintos algoritmos utilizados en el *Proceso de Clasificación*, incluye: las tareas de análisis de estado de acciones, clasificación de noticias según proyección de estado de cada acción, entrenamiento de algoritmos y análisis de los resultados del testeo y ajuste de los parámetros de los componentes.

3.1.1. Proceso de Clasificación

El *Proceso de Clasificación* tiene como objetivo la descarga y clasificación de noticias. De este modo incluye un módulo para la descarga de noticias RSS, otro módulo para la identificación de acciones, un módulo para la construcción de representaciones según modelos y por último un módulo para clasificación de documentos. A continuación se describen las principales características de cada uno de los módulos nombrados anteriormente.

Descarga de Noticias RSS

El módulo de *Descarga de Noticias RSS* realiza una inspección periódica de las fuentes de noticias RSS ingresadas por el usuario en busca de nuevas noticias. Cada vez que alguna noticia nueva es detectada, se descarga y se almacena para su posterior análisis.

Las noticias descargadas se agrupan por el día de su publicación a fin de facilitar su lectura.

Identificación de Acciones

El componente que realiza la *Identificación de Acciones* realiza una búsqueda de referencias a acciones que puedan existir en el texto de cada noticia. Las referencias son conjuntos de palabras que identifican unívocamente a una acción.

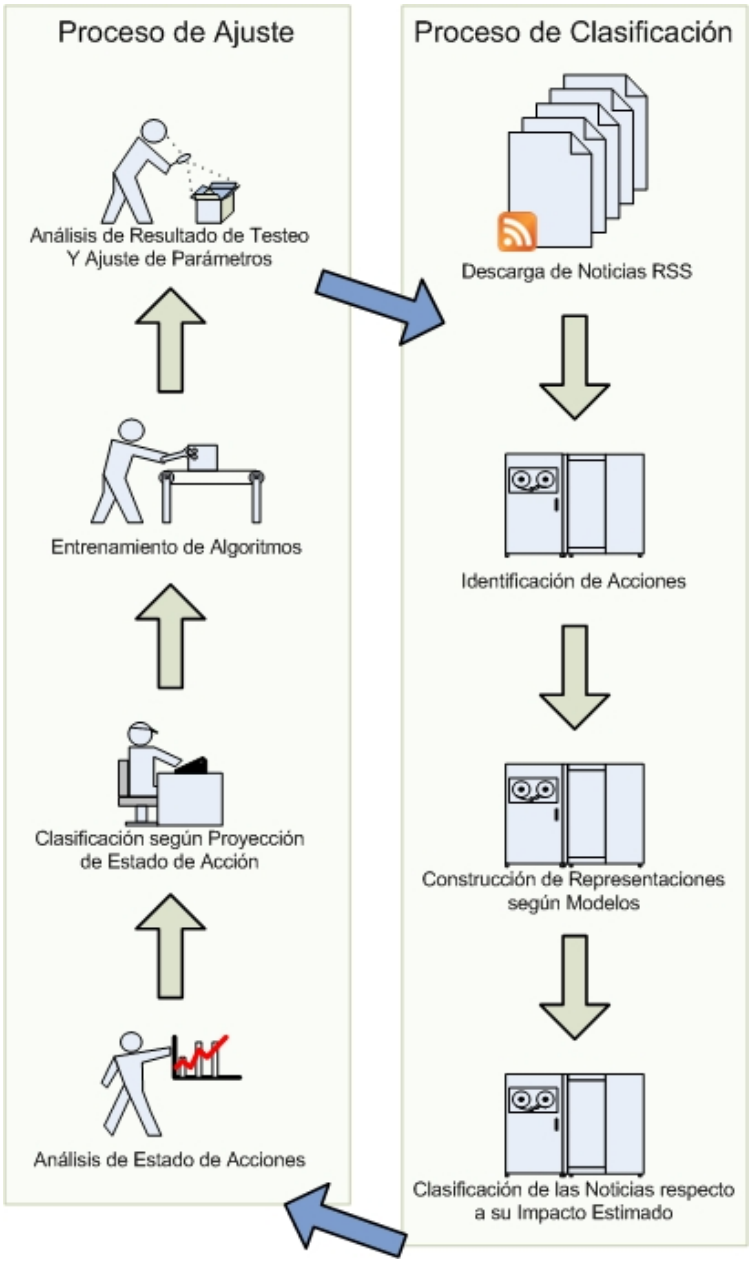

Figura 3.1: Procesos a contemplar por el Clasificador de Noticias Financieras

Por ejemplo: para la acción `COKE` la lista de referencias puede ser la siguiente:{*Coca Cola Company*, *Coca Cola Bottling*, *Coca Cola*}. De esta forma, cada vez que el componente encuentra una de esta referencias de inmediato identificará la acción `COKE`.

El componente de identificación permite que el usuario técnico defina que acciones desea identificar junto con la lista de referencias para cada una de las acciones. En la versión actual de ClaNFi se establece una lista de 9306 acciones que cotizan en Estados Unidos, construida a partir de varias fuentes, entre las cuales se destaca la lista de empresas incluidas en el índice Russell 3000[1].

Construcción de representaciones según Modelos

El módulo de Modelos de Representación de Documentos permite llevar el contenido de cada documento a una representación más adecuada para su manejo. El problema de hallar una representación para los documentos se ha presentado en la sección [2.3] y juega un papel muy importante en el posterior desempeño de los algoritmos de clasificación.

Como resultado de aplicar un modelo de representación al texto original se obtiene una lista de rasgos (*feaures*) para cada documento. Esta lista no tiene repetidos, sino que, para cada rasgo, se retorna también un número que indica la cantidad de sus ocurrencias en el texto.

ClaNFi permite agregar o quitar componentes de procesamiento de modelos de representación sin necesidad de grandes ajustes. Más adelante se presentan los componentes de procesamiento de modelos de representación construidos en el marco del proyecto

Clasificación de Documentos

El módulo de Clasificación de Documentos permite utilizar uno o más clasificadores para categorizar documentos. Para esto, el usuario técnico establece, en el proceso de ajuste, una lista de componentes de clasificación o de representación de documentos. Luego, en el proceso de clasificación, y cada vez que llega un nuevo documento, el motor de ClaNFi ejecuta cada componente de la lista; acumulando los resultados, que serán desplegados al usuario al finalizar el procesamiento.

Adicionalmente se provee un mecanismo de tipo *plug-in*, donde se pueden agregar o quitar componentes de clasificación. En la sección [3.2] se presentan los componentes de clasificación que acompañan la actual versión de la herramienta.

[1] El índice Russell 3000 es promedio ponderado de las 3000 empresas compañías más grandes en Estados Unidos tomando como referencia su capitalización en el mercado. Se estima que representan aproximadamente el 98 % del mercado estadounidense.

3.1.2. Proceso de Ajuste

El *Proceso de Ajuste* tiene objetivo el ajuste de los distintos componentes y algoritmos utilizados en el *Proceso de Clasificación*. Incluye componentes que realizan las tareas de análisis de estado de acciones, clasificación de noticias según proyección de estado de cada acción, entrenamiento de algoritmos y análisis de los resultados del testeo y ajuste de los parámetros de los componentes. A continuación se despliegan detalles de cada uno de los módulos que agrupan las tareas antes enumeradas.

Análisis de Estado de las Acciones

Con el fin de evaluar el desempeño de los clasificadores, es necesario definir un criterio que indique la correctitud de una predicción acerca del impacto de una noticia, ya sea positivo o negativo.

En un primer intento de definir un criterio de este estilo, se pensó en que la retroalimentación que recibiría ClaNFi sería ingresada manualmente por un usuario humano. Se percibió luego que se contaba con cantidades enormes, para ser manejadas por una persona, de noticias y acciones referenciadas. Adicionalmente, la capacitación y habilidades requeridas para realizar un análisis financiero de este tipo, indicaban la necesidad de contar con una persona calificada en el tema, siendo esta una de las limitantes encontradas en el proyecto.

En cambio, se optó por definir un criterio que pudiera ser expresado y calculado de forma automática. De esta forma, se implementó una solución que contempla los valores de cierre de las acciones involucradas en las noticias. Dada una noticia, se toman el valor de cierre del día de su publicación y el valor de cierre del siguiente día. Si el valor experimentó un incremento quiere decir que la noticia tuvo un impacto positivo, de otro modo la noticia tuvo un impacto negativo. Con el fin de utilizar este mecanismo, fue necesario contar con un corpus de noticias viejas, para cada una se identificaron las acciones involucradas, descargando luego los estados de cada acción (valores de apertura y cierre y volumen negociado). El algoritmo se presenta en detalle en el Cuadro [3.1].

Sin retroalimentación no hubiera sido posible realizar el entrenamiento ni las pruebas de los algoritmos implementados. El método planteado en esta sección permitió superar la ausencia de una retroalimentación que debería haber realizado una persona calificada en el tema. Aún así, presenta algunas limitaciones que deben ser tenidas en cuenta, por ejemplo: la retroalimentación se mide en la comparación del estado de las acciones entre el día de la publicación de una noticia y el día siguiente, cuando en general el impacto de las noticias se puede observar en los siguientes minutos a su publicación.

Clasificación según Proyección de Estado de las Acciones

En un intento de obtener un clasificador de referencia (*baseline*), se construyó un clasificador sencillo, que sin tomar en cuenta el contenido de la noticia, pudiera ofrecer una predicción relativamente válida.

1. Antes que nada se establece cual será el *Mercado de Referencia*. Se carga las horas de apertura y cierre y de los días en los que permanecerá cerrado.
2. Para cada una de las acciones a identificar se descarga los estados de situación para cada día dentro del período estudiado. El estado de situación consiste en el valor de cierre y el volumen negociado durante el día.
3. Para cada acción a_i que sea identificada en la noticia n se calcula el impacto que tuvo la noticia sobre dicha acción de la siguiente forma.

   Dadas dos fechas, una fecha final (f_f) y una fecha inicial (f_i), el estado de una acción entre ambas se calcula con la siguiente expresión:

$$puntaje = \frac{valorCierre(f_f)}{valorCierre(f_i)} - 1$$

4. Luego si el puntaje es positivo entonces el estado de la acción en la fecha f_f en relación a la fecha f_i es positivo. En otro caso el estado es negativo.
5. En el caso de que la fecha final (f_f) y la fecha inicial (f_i) coincidan, la expresión de cálculo se modifica levemente de la siguiente manera.

$$puntaje = \frac{valorCierre(f_f)}{valorApertura(f_f)} - 1$$

Cuadro 3.1: Algoritmo de retroalimentación

Dada un noticia, se identifican las acciones. Luego para cada acción se obtiene su estado (valor de cierre, de apertura y volumen negociado) en el día de la publicación y se compara con respecto al día hábil anterior más cercano. Si el estado promedio es positivo, entonces la tendencia se mantendrá, por lo que el clasificador predice que la noticia tendrá un impacto positivo. De igual forma, si la proyección de los estados de las acciones identificadas es negativa, se predice un impacto negativo de la noticia.

Figura 3.2: Valores de Cierre de Heinz

Siguiendo con el ejemplo, clasificaremos la noticia de Heinz [1.1] según la proyección de estado de las acciones. Recordemos que la noticia involucraba a la acción HNZ (Heinz) y fue publicada el día 1° de Junio de 2006. La proyección del estado implica observar el estado de la acción en el día de publicación y realizar una comparación con el día anterior. En este caso, según la Figura [3.2][2], la tendencia es negativa, por lo que la noticia tendrá un impacto negativo.

Entrenamiento de Algoritmos

De forma adicional ClaNFi provee un componente que automatiza las etapas de entrenamiento y testeo de cualquier clasificador binario. Para esto el usuario debe proveer el componente que implementa el clasificador (junto con los valores de posibles parámetros), el conjunto de entrenamiento y el conjunto testeo. ClaNFi cuenta, también, con un componente que crea conjuntos de entrenamiento para clasificadores binarios. Luego de ejecutada la fase de entrenamiento, el componente que realiza la fase de testeo, despliega finalmente los resultados, incluyendo la Precisión y Recuperación obtenidas.

Ajuste de parámetros

El ajuste de parámetros es la única tarea que corresponde realizar en su totalidad a un usuario humano. Comprende la creación de instancias de clasifi-

[2]Google Finance - http://finance.google.com/finance

cadores, modelos de representación o componentes en general y la configuración de los parámetros de dichas instancias.

3.2. Prototipo

Como se explicó en la sección anterior, el motor de ClaNFi es independiente de los algoritmos de clasificación o procesamiento que se utilicen. De esta forma, es posible agregar o quitar algoritmos de clasificación o, en forma más genérica, de procesamiento de documentos.

Para cumplir con los requerimientos de este proyecto fue necesario incluir varios componentes de representación de documentos así como componentes de clasificación. En esta sección se presentan los diferentes componentes que acompañan la actual versión de la herramienta.

3.2.1. Componentes de Representación de Documentos

En cuanto a los componentes de representación de documentos, se incluyeron entidades de procesamiento de Palabras Clave, tokens de Freeling y HM Frames. A continuación se explica en detalle cada uno de estos modelos.

Modelo de Palabras Clave

El modelo de palabras clave, explicado en detalle en la sección [2.3.2] considera a un documento como un conjunto de palabras claves, omitiendo la mayor parte de la información proporcionada por la estructura lingüística.

Varios son los componentes que realizan tareas asociadas a este modelo de representación.

- Preprocesamiento: Las noticias descargadas poseen, por lo general, en su contenido publicidad, tags html, u otras formas de ruido adicional a la información relevante. ClaNFi cuenta con un módulo de preprocesamiento, el cual elimina tags html, publicidad conocida, *tokens* y caracteres no deseados.
- Construcción de la Lista de Palabras Claves: El preprocesamiento se realiza sobre la tira de caracteres del contenido original de la noticia. Luego es necesario obtener los *tokens* correspondientes a las Palabras Clave. Para esto se cuenta con un componente que convierte el contenido de la noticia en palabras clave. El usuario puede definir cual será la expresión regular que se utilizará para realizar la *tokenización*, por defecto la expresión regular utiliza los espacios y los signos de puntuación para separar en *tokens*.
- Eliminación de Palabras Vacías: En una etapa posterior del procesamiento es posible eliminar las palabras que carecen de importancia (llamadas *Stops Words* o Palabras Vacías), por ser sumamente populares en las construcciones de texto, ej. [“de”, “en”, “los”]. Este paso es opcional y puede no ser requerido, dependiendo de las características del algoritmo que será utilizado.

- *Stemming*: Finalmente ClaNFi cuenta con un componente que realiza *stemming* sobre la lista de Palabras Clave obtenidas hasta el momento. La idea central de este paso (como fue presentada en la sección [2.3.2]) es reducir cada palabra a su *Radical* o *Stem*, eliminando posibles sufijos y prefijos. El algoritmo que se utiliza para realizar *stemming* es el algoritmo de Porter [Por80].

Modelo de Tokens de Freeling

El modelo de tokens de Freeling considera a un documento como un conjunto (o posiblemente un conjunto múltiple) de pares `(token, tag)`. Estos pares se calculan mediante *Pos Tagging*, donde cada tag corresponde a la categoría morfosintántica del término analizado. Para esto se utiliza el paquete *Freeling*[3], el paquete provee servicios para el análisis del lenguaje natural (análisis morfológico, reconocimiento de fechas, Pos Tagging, entre otros).

Modelo de HM Frames

Este modelo utiliza el concepto de HM Frames para representar los documentos [2.3.3].

Se decidió contar con este modelo ya que plantea algunos aspectos lingüísticos innovadores en el campo de la clasificación, como por ejemplo la identificación y eliminación de construcciones sintácticas que no contribuyen al sentido *central* del contenido del texto (adverbios, auxiliares de tiempo y modalidad, entre otras). Cabe destacar que el sentido *central* del contenido del texto del que se habla refiere al sentido que toma el texto para las tareas de recuperación de información y clasificación de información, actividades en las cuales, por lo general, no es necesario contar con análisis semánticos.

3.2.2. Componentes de Clasificación

Luego de analizar algunos trabajos previos; fundamentalmente un estudio de Koster [Kos06b], donde plantea tres algoritmos lineales y su desempeño sobre un corpus de artículos de periódicos (*el corpus Reuters-8119*), se tomó la decisión de utilizar principalmente este tipo de clasificadores. La familia de algoritmos lineales, permite superar la rigidez que caracteriza a los clasificadores basados en reglas. Por otro lado, los trabajos previos existentes en cuanto a clasificación de noticias no utilizan clasificadores basados en ejemplos.

Los clasificadores lineales representan cada documento como un conjunto de rasgos $d = \{f_1, f_2, ..., f_m\}$, donde m es la cantidad de *rasgos activos* en el documento, o sea, la cantidad de rasgos que ocurren realmente en el documento. La *fuerza* de cada rasgo f en el documento d es denotada como $s(f, d)$. La fuerza es usualmente una función del número de veces que ocurre f en d (su *frecuencia*, denotada por $n(f, d)$). La fuerza puede ser utilizada meramente para indicar la presencia o la ausencia de f en el documento, en cuyo caso toma solamente los valores 1 o 0; de otra manera puede se igual a $n(f, d)$, o tomar otros valores

[3] http://garraf.epsevg.upc.es/freeling/

para reflejar, por ejemplo: el tamaño del documento.

Para clasificar los documentos, cada clase mantiene una función S_c, la cual evaluada en d, produce un puntaje $S_c(d)$. En una mono-clasificación, el documento es asignado a la clase para la cual tiene mayor puntaje.

Un *clasificador lineal* representa cada perfil de clase como un *vector de pesos*:

$$w_c = (w(f_1, c), w(f_2, c), ..., w(f_n, c)) \equiv (w_1, w_2, ...w_n)$$

donde n es el total de rasgos en el dominio y $w(f, c)$ es el peso del rasgo f para la clase c. El puntaje de cada documento se evalúa de la siguiente manera:

$$S_c(d) = \sum_{f \in d} s(f, d) \cdot w(f, c)$$

El documento es asignado entonces a la clase para la cual obtenga mayor puntaje.

La tarea del algoritmo de aprendizaje para un clasificador lineal es encontrar un vector de pesos que clasifique de mejor manera a los nuevos documentos.

Algoritmo Bayesiano Simple

El clasificador Bayesiano simple es un método probabilístico de clasificación [Kos06b]. Trata de determinar la probabilidad de que un documento pertenezca a una clase c dada la certeza que los rasgos están presentes (o no), esto es $P(c|X)$, o sea la probabilidad de c dado X, donde X es un vector, tal que X_i es un atributo que expresa la presencia o la ausencia del rasgo i. De acuerdo a la regla de Bayes, esto es equivalente a :

$$\frac{P(X|c) \cdot P(c)}{P(X)}$$

donde $P(c)$ es la probabilidad de que cualquier documento pertenezca a la clase c y $P(X)$ es la probabilidad de que el vector X ocurra. Tomando $X = (a_1, a_2, ...a_n)$ y asumiendo que $P(X)$ es igual para todos los valores posibles, esto es proporcional a:

$$P(a_1, a_2, ...a_n|c) \cdot P(c)$$

Asumiendo independencia de atributos, esto se puede computar como:

$$\prod_{j=1,n} P(a_j|c) \cdot P(c)$$

Es fácil notar que para que esto funcione, todas las probabilidades deben ser diferentes de cero. En general, este es un problema bien conocido y al igual que en el trabajo de Koster [Kos06b] se puede optar por algún valor razonable para $P(a|c)$, por ejemplo: 0.001.

Teóricamente, este es un muy buen método, ya que toma en cuenta toda la información disponible. Por esta razón también, se debe mantener acotado

fuertemente el tamaño del conjunto de rasgos.

Este clasificador presenta la ventaja de ser simple de construir, además tiene un relativo buen desempeño. Por estas razones es que fue elegido para ser incluido en ClaNFi.

Algoritmo de Winnow

El *algoritmo de Winnow* es en realidad una familia de algoritmos. Seguimos la explicación de [Kos06b], donde se presenta el funcionamiento del algoritmo sobre una clasificación binaria (relevante/no relevante), la extensión para otras clasificación es bastante trivial.

Un documento d se representa como un vector $s = (s_1, s_2, ..., s_n)$, donde s_i indica la fuerza del i-ésimo rasgo en el documento d. Como definición podemos agregar que los rasgos activos de un documento son aquellos que ocurren en él. Se toman en cuenta, solamente los rasgos activos, ya que por definición los demás tienen fuerza cero. El algoritmo utiliza tres parámetros: un *umbral* θ, un *parámetro de promoción* α y un *parámetro de degradación* β. Además se cumple que:

1. θ es usualmente 1
2. $0 < \beta < 1$
3. $\alpha > 1$

El algoritmo de Winnow mantiene un conjunto de pesos (uno para cada rasgo del perfil de la clase), y sí el algoritmo predice 0 cuando el documento esta etiquetado como 1, entonces se multiplican por α los valores de los pesos de los rasgos que predijeron 1. Sí, en cambio, el algoritmo predice 1 cuando el documento esta etiquetado como 0, entonces se multiplican por β los valores de los pesos de los rasgos que predijeron 0. El parámetro θ se utiliza como referencia para inicializar los pesos de todos los rasgos.

El algoritmo de Winnow tiene un máximo de $2 + 3r(1 + logn)$ errores, donde r es el número de rasgos del documento y n es el número de rasgos del perfil de la clase. Se ha mostrado que este límite corresponde, en la práctica, a una muy pequeña cantidad [Blu96].

Winnow Positivo El algoritmo original llamado Winnow Positivo (*Positive Winnow*), mantiene un vector de pesos de n-dimensiones $w = (w_1, w_2, ...w_n)$, que se actualiza *cada vez que se comete un error*, de la siguiente manera:

1. **ejemplo positivo**
   Si el algoritmo predice 0 y el documento esta etiquetado como 1, entonces los pesos de todos los rasgos activos se promueven, multiplicando cada uno por α.
2. **ejemplo negativo**
   Si el algoritmo predice 1 y el documento esta etiquetado como 0, entonces los pesos de todos los rasgos activos se degradan, multiplicando cada uno por β.

Algorithm	Version					
	Basic	**Norm**	**θ-range**	**Lin-fq**	**Sqrt-fq**	**Discard**
Balanced Winnow	64.87	NA	69.66	72.11	71.56	73.2
Positive Winnow	55.56	63.56	65.80	67.20	69.67	70.0
Perceptron	65.91	NA	63.05	66.72	68.29	70.8

Cuadro 3.2: Recall/Precisión en porcentajes para distintas variantes del algoritmo de Winnow

En ambos casos, los pesos de los rasgos inactivos mantienen su valor. Inicialmente, todos los pesos tienen un valor positivo pequeño, por ejemplo: θ/d donde θ es umbral y d es la cantidad promedio de rasgos en el documento.

Perceptron Esta variante corresponde a las primeras formas de las redes neuronales. Es similar al anterior, excepto que en este caso los pesos se actualizan de una manera *aditiva*: "Un peso es promovido sumandole α y es degradado al restarle α".

Winnow Balanceado En la variante balanceada del algoritmo, para cada rasgo se mantienen dos pesos, w^+ y w^-. El coeficiente de un rasgo es la $diferencia$ entre esos dos pesos, permitiendo ahora coeficientes negativos. Para cada documento $s = (s_1, s_2, ...s_n)$ el algoritmo predice el puntaje 1 si y sólo si:

$$\sum_{j=1,m} (w_j^+ - w_j^-)s_j > \theta$$

Inicialmente, los pesos w^+ se establecen en $2\theta/d$ y los pesos w^- en θ/d, para obtener un promedio de θ en cada clase. En caso de error, solamente los pesos de los rasgos activos son actualizados, de la siguiente forma:

1. **ejemplo positivo**
   La parte positiva del peso es promovida, al ser multiplicada por α; la parte negativa es degradada, al ser multiplicada por β. De esta forma el coeficiente $(w_j^+ - w_j^-)$ se incrementa.

2. **ejemplo negativo**
   La parte negativa se promueve y la parte positiva se degrada, decrementando el coeficiente.

Consideraciones Winnow Balanceado y Perceptron tienen una ventaja importante sobre el Winnow Positivo: la presencia de coeficientes negativos los hace más robustos contra variaciones en el tamaño de los documentos. Con Winnow Positivo puede que para un documento extenso se obtenga un puntaje alto, que corresponde a la ocurrencia de muchos rasgos de bajo peso. Si se utiliza una función de fuerza normalizada en el tamaño del documento:

$$s^n(f,d) = \frac{s(f,d)}{\sum_{f \in d} s(f,d)}$$

se obtiene un Winnow Positivo bastante más competitivo (véase las primeras dos columnas del Cuadro [3.2]).

En esta tabla se muestran los resultados de un estudio presentado por Dagan [IDR97], cada ítem es un promedio de dos pares de conjuntos de entrenamiento y testeo, cada uno con 2000 documentos de entrenamiento y 1000 documentos de testeo. Para cada conjunto de entrenamiento, los algoritmos se corrieron como máximo unas 50 veces o hasta que ningún error era cometido.

Mejoras a Winnow En ese mismo estudio, Dagan propone algunas mejoras posibles:

1. Utilizar un *rango de umbrales* $[\theta^+, \theta^-]$ en lugar de un umbral solamente. Durante el entrenamiento, el algoritmo predice 0 si el puntaje del documento se encuentra por debajo del valor θ^-, y 1 si esta por encima de θ^+. Todos los ejemplos en el rango $[\theta^+, \theta^-]$ son tratados como errores. El rango elegido fue [0,9, 1,1]. Los resultados se muestran en la tercer columna del Cuadro [3.2].

2. Los resultados anteriores se obtuvieron con $s(f, d) = 1$ o 0, dependiendo de la presencia del rasgo f en el documento d. Esta *ponderación binaria* no refleja si el rasgo aparece más de una vez en el documento. Otras opciones son la *ponderación lineal*, donde $s(f, d) = n(f, d)$ (ver cuarta columna en el Cuadro [3.2]) o la *ponderación sqrt*, donde $s(f, d) = \sqrt{n(f, d)}$ (ver quinta columna en el Cuadro [3.2]).

3. Otra manera de mejorar este algoritmo se logra descartando los rasgos que no contribuyen a la clasificación. Esto se realiza de la siguiente manera. Primero se ejecuta el algoritmo un numero de veces tal que la tasa de error se estabilice por debajo de cierto nivel. Luego se eliminan todos aquellos rasgos cuyo peso se encuentre entre una promoción y una degradación del valor inicial (anterior a ejecutar el algoritmo). De esta forma dos tercios de los rasgos son eliminados. El resultado obtenido se presenta en la última columna del Cuadro [3.2].

El algoritmo de Winnow fue el primer algoritmo de clasificación seleccionado para su inclusión en el proyecto, dados los buenos resultados obtenidos por Koster [3.2] en corpus similares.

Por una cuestión de tiempo no se incluyen todas las variantes pertenecientes a esta familia de algoritmos; solamente fueron implementados *Winnow Positivo* y *Winnow Balanceado*. Al igual que en los trabajos realizados por Koster [Kos06b] se introdujeron las mejoras propuestas por Dagan [IDR97].

Algoritmo de Weighted-Majority

Además de incluir clasificadores basados en algoritmos lineales, se decidió agregar un clasificador que permita minimizar la cota de errores de otros clasificadores.

```
Para todo i, w_i:=1
Para cada ejemplo <x, c(x)>
  .q_0:=q_1:=0
  .Para cada L_i
    - Si L_i(x)=0 =>q_0:=q_0+w_i
    - Si L_i(x)=1 =>q_1:=q_1+w_i
  .Si q_0>q_1 =>predecir 0
  .Si q_1>q_0 =>predecir 1
  .Si q_1=q_0 =>predecir al azar
  .Para cada L_i equivocado =>w_i:=βw_i
```

Cuadro 3.3: Algoritmo Weighted-Majority

De esta forma se optó por construir un clasificador basado en Weighted-Majority. El algoritmo Weighted-Majority realiza predicciones utilizando una ponderación de las predicciones de otros algoritmos [Mit97].

Una propiedad interesante del algoritmo es la de permitir inconsistencias en los datos de entrenamiento. Esto es posible pues no elimina hipótesis que sean inconsistentes con parte del conjunto de entrenamiento, sino que reduce su ponderación.

Otra propiedad interesante es que podemos acotar la cantidad de errores realizados por el Weighted-Majority en términos de los errores cometidos por el conjunto de algoritmos de predicción utilizados. Formalmente, dados n algoritmos que cometen un mínimo de k errores con la secuencia de ejemplos D, Weighted-Majority comete a lo sumo: $2{,}4 * (k + \log_2 n)$ si $\beta = 0{,}5$.

En el Cuadro [3.3] se presenta el pseudo-código del algoritmo [Mit97].

3.3. Implementación

Para el desarrollo del prototipo se utilizó el lenguaje Java y recursos libres disponibles en la web. A un nivel bastante macro el prototipo esta organizado utilizando el patrón de arquitectura en capas; el mismo es relajado y se conforma de tres capas. El diagrama de la Figura [3.3] presenta la Arquitectura del Sistema[4].

Para la *Persistencia de Datos* se utilizan archivos XML, junto con la biblioteca XStream que permite realizar la persistencia de clases Java mediante el API Reflection de Java. Estas herramientas, en conjunto, permiten que sea posible agregar componentes de procesamiento de texto (identificadores de acciones, clasificadores, extractores de información, etc) en tiempo de ejecución.

La *Capa Lógica* se implementó siguiendo los lineamientos descriptos en la sección anterior. Se ideó un núcleo central basado en interfaces que permite el manejo de documentos en general, sin que estos deban ser noticias financieras.

[4] XStream es una biblioteca que provee persistencia de clases Java de forma análoga a la serialización

Esta generalidad permite, junto con el uso de XStream, que sea posible agregar componentes en tiempo de ejecución y/o utilizar la aplicación para otras tareas de procesamiento de lenguaje natural.

En la Figura [3.4] se incluye un diagrama correspondiente al modelo de dominio.

Si se desea agregar componentes se deben respetar las siguientes interfaces:

clanfi.pers.IPersistencia Interfaz que debe cumplir la entidad que realice la persistencia.

clanfi.ker.IMDocumentos Interfaz que debe cumplir la entidad que realice la administración de noticias. Para cambiar el manejador de documentos se debe indicar el identificador en el archivo /ruta instalacion /ClaNFi /clanfiXML /Properties.xml bajo la etiqueta "manejador de documentos".

clanfi.ker.IEntidad Proc Todo componente de clasificación o procesamiento de texto debe cumplir esta interfaz.

clanfi.ker.IClasificador Adicionalmente si se trata de un clasificador debe cumplir también esta interfaz.

clanfi.drep.IRModelo Interfaz que debe cumplir todo componente que realice el procesamiento de documentos y construya representaciones según uno o más modelos.

clanfi.acc.IMAcciones Interfaz que debe cumplir la entidad que realice la administración de acciones. Para cambiar el manejador de documentos se debe indicar el identificador en el archivo /ruta instalacion/ ClaNFi/ clanfiXML/

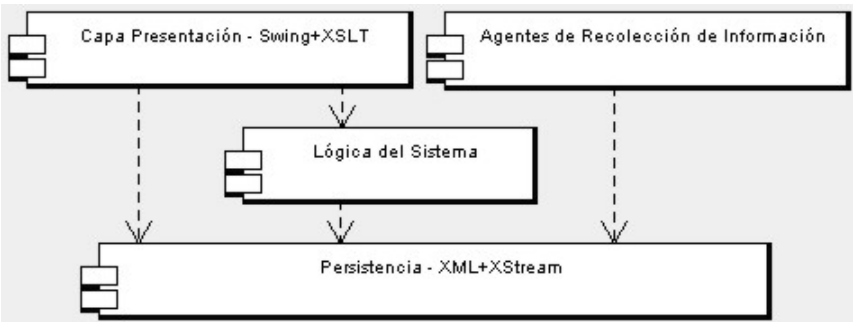

Figura 3.3: Arquitectura del Sistema.

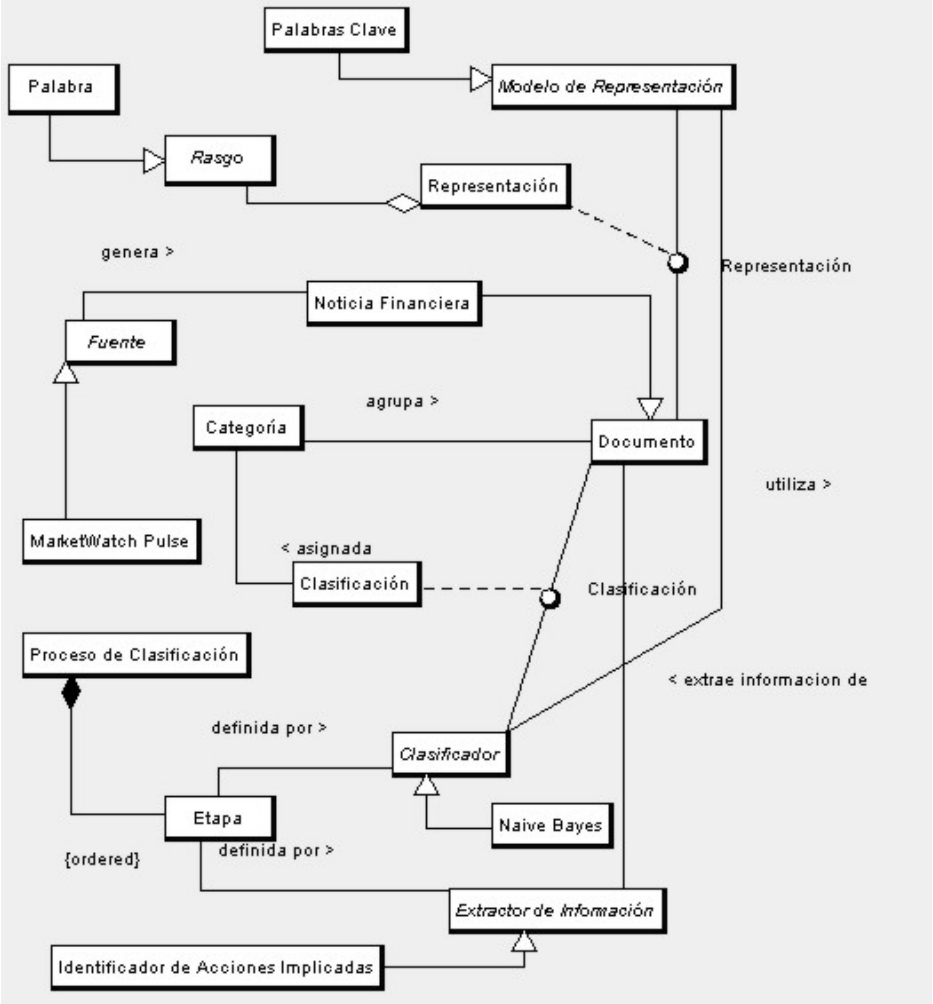

Figura 3.4: Modelo de Dominio

Properties.xml bajo la etiqueta “manejador de acciones”.

clanfi.acc.IRetroalimentador Interfaz que debe cumplir la entidad que realice la retroalimentación.

Por otro lado, `ClaNFi` cuenta con algunos componentes ya creados. Estos son:

Modelos.Palabras.Preprocesamiento Quita la basura del texto del documento. El componente permite definir qué se debe tomar como basura. Recibe en la propiedad “texto” el texto original del documento. Retorna en la propiedad “TextoLimpio” el texto sin basura.

Modelos.Palabras.ModeloPalabras Procesa el texto y devuelve la lista de palabras clave del texto del documento. Recibe en la propiedad “TextoLimpio” el texto a procesar. Retorna en la propiedad “PalabrasClave” una lista de rasgos ponderados correspondientes a las palabras clave del texto.

Modelos.Palabras.PalabrasVacias Quita las palabras vacías de un vector de palabras clave. Recibe en la propiedad “PalabrasClave” el vector de rasgos ponderados correspondientes a las palabras clave del texto. Retorna en la propiedad “PalabrasClave” el vector de rasgos ponderados correspondientes a las palabras clave del texto, sin las palabras vacías.

Modelos.Palabras.Stemming.Porter A partir de un vector de palabras claves realiza el stemming y retorna el vector resultante. Recibe en la propiedad “PalabrasClave” el vector de rasgos ponderados correspondientes a las palabras clave del texto. Retorna en la propiedad “StemsClaves” el vector de rasgos ponderados correspondientes a los stems del texto.

Modelos.Freeling.Tokenizador Procesa el texto y devuelve la lista de tokens obtenidos mediante el POS tagging realizado por Freeling. Recibe en la propiedad “TextoLimpio” el texto a procesar. Retorna en la propiedad “tokens” una lista de rasgos ponderados correspondientes a las tokens obtenidos mediante el POS tagging realizado por Freeling.

Modelos.Freeling.FPalabrasVacias Quita los tokens vacíos de un vector de tokens de Freeling. Por tokens vacíos se refiere a tokens que no aportan al contenido del texto, similar a las palabras vacías. Recibe en la propiedad “tokens” una lista de rasgos ponderados correspondientes a las tokens obtenidos mediante el POS tagging realizado por Freeling. Retorna en la propiedad “tokens” una lista de rasgos ponderados correspondientes a las tokens obtenidos

mediante el POS tagging realizado por Freeling, sin los tokens vacíos.

Modelos.HMFrames.HMFrameProxy Procesa el texto y devuelve la lista de hm frames que se obtienen utilizando AGFL y EP4IR. Recibe en la propiedad "TextoAcc" el texto del documento. En el caso particular de las noticias financieras se obtienen mejores resultados si el texto ya viene sin las referencias a acciones – o sea sí se sustituyen las referencias por comodines. Retorna en la propiedad HMFrames una lista de rasgos ponderados correspondientes a los hmframes.

EInfo.AIdent.AccionIdent Realiza la identificación de acciones dentro de una noticia, Recibe en la propiedad "texto" el texto de la noticia. Retorna en la propiedad "TextoAcc" el texto sin las referencias a las acciones encontradas (cada una se sustituye por un comodín). En la propiedad "Acciones" se retorna la lista de identificadores de las acciones encontradas.

La *Presentación* se realiza utilizando las transformaciones XSLT y lenguaje JavaScript para desplegar los archivos XML generados por la aplicación. De este modo, se cuenta con una interfaz altamente flexible que permite agregar formatos de archivos XML nuevos y configurar como se desplegarán al usuario. En la Figura [3.5] se presenta la interfaz gráfica de la aplicación.

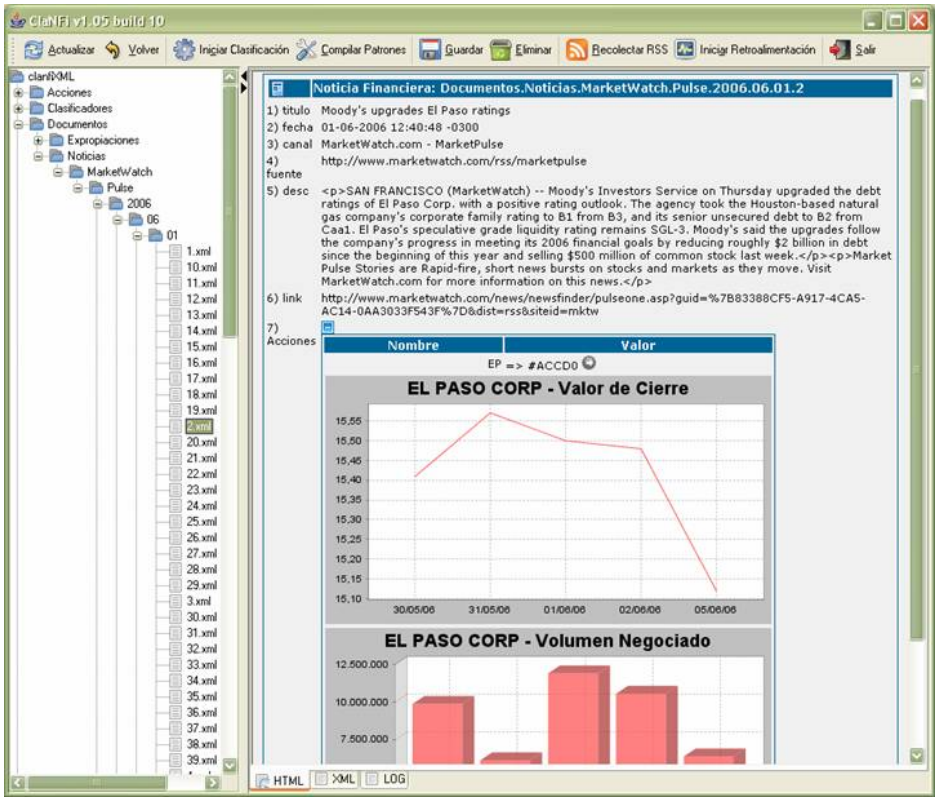

Figura 3.5: ClaNFi

Capítulo 4

Pruebas

Luego de contar con un prototipo estable de ClaNFi se proyectó la realización de las correspondientes pruebas. De esta forma fue necesario contar con un corpus de referencia y con la definición de una secuencia de componentes a probar. Esto último se refiere a la necesidad de establecer cuáles clasificadores utilizar, con qué parámetros y con qué modelo (o modelos) de representación para los documentos.

4.1. Corpus de Referencia

El corpus fue construido a partir de noticias RSS descargadas entre el 1° de Junio y el 21 de Agosto del 2006. En principio se utilizaron distintas fuentes en idioma inglés, y dentro de cada fuente varios canales de noticias RSS. El idioma inglés fue elegido pues es el idioma para el cual se publica mayor cantidad de contenido de tipo financiero.

Luego se seleccionaron dos fuentes candidatas: la primera correspondía al canal *Most Popular* de *Yahoo! Finance*, y la segunda, al canal *Market Pulse* de *Market Watch*. Esta selección fue necesaria para descartar fuentes que, por ejemplo, no publicaron una cantidad suficiente de noticias, o que incluían noticias no financieras en sus publicaciones.

Finalmente el corpus de referencia correspondió a las noticias obtenidas de *Market Watch*. La principal razón de esta elección corresponde a que las noticias de esta fuente tienen más contenido que las de *Yahoo! Finance*. De esta forma se cuenta con más información analizable en cada noticia.

El corpus obtenido cuenta con 4689 noticias y 616986 palabras, lo que hace un promedio de 132 palabras por noticia.

4.2. Pruebas Realizadas

Luego de realizar varias pruebas aisladas, se decidió proyectar un plan de pruebas que contemplara las combinaciones modelos de representación - clasificadores más prometedoras. De esta forma se realizaron ordenadamente las

Conjunto	Cantidades según Impacto Calculado					
	Negativo		Positivo		Neutro	
Entrenamiento	619	52 %	496	41 %	85	7 %
Testeo	89	50 %	78	43 %	13	7 %

Cuadro 4.1: Composición de los conjuntos de Entrenamiento y Testeo

pruebas que se documentan a continuación.

En una primera instancia se procesaron todas las noticias, identificando las acciones referenciadas en cada una de ellas. De esta forma se descartaron para la siguiente etapa las noticias dentro de las cuales no se reconoció ninguna acción.

Luego, utilizando los criterios definidos en el módulo de *Análisis de Estado de las Acciones* (sección [3.1.2]), se determinó automáticamente si el impacto real de la noticia había sido positivo o negativo. Recordemos que, con el fin de estimar el impacto de una noticia, se comparan los estados de las acciones involucradas en el día de publicación con respecto al día siguiente. Así se construyó un corpus de noticias, para las cuales había estimado si su impacto había sido positivo, negativo o neutro.

De esta forma las pruebas se realizaron sobre un subconjunto de 1380 noticias extraídas del corpus original. Para el entrenamiento se utilizó el 85 % dejando el restante 15 % para la etapa de testeo. Las características de ambos conjuntos se presentan en el Cuadro [4.1].

Recordemos que en el marco de este proyecto solamente se trabajó sobre clasificadores binarios (son aquellos que indican si un documento pertenece o no a una clase). Y como el objetivo del proceso de clasificación es indicar si una noticia tendrá un impacto positivo, negativo o neutro, fue necesario realizar dos conjuntos de clasificadores. El primer conjunto trata de predecir si la noticia tendrá un impacto negativo o no. El segundo conjunto, en cambio, intenta predecir si una noticia tendrá un efecto positivo o no. Para ambos conjuntos de clasificadores se utilizó el mismo corpus de referencia, cuyas características se mencionaron anteriormente.

Por último se efectuaron las pruebas de los clasificadores establecidos. En el Cuadro [4.2] se presentan los resultados para el conjunto de clasificadores *Negativos*, los cuales indican si una noticia tendrá un impacto negativo o no. Por otro lado, en el Cuadro [4.3] se presentan los resultados para el conjunto de clasificadores *Positivos*, que son aquellos que indican si una noticia tendrá un impacto positivo o no.

En la primera fila se presentan los resultados obtenidos para el clasificador basado en Winnow Positivo. Los diferentes conjuntos de resultados corresponden a los distintos modelos de representación de documentos utilizados.

En la siguiente fila se presentan los resultados obtenidos para el clasificador Bayesiano Simple. De igual forma, los diferentes conjuntos de resultados corres-

Clasificador		Resultados		
Algoritmo	**Modelo**	**Precisión**	**Recall**	$F_{0,5}$
Winnow Positivo	HMFrames	59 %	20 %	36
	Stemming	52 %	43 %	49
	Freeling	53 %	39 %	47
	Palabras Clave	43 %	37 %	41
Bayesiano Simple	HMFrames	51 %	78 %	58
	Stemming	52 %	87 %	60
	Freeling	53 %	87 %	61
	Palabras Clave	54 %	83 %	61
Weighted-Majority		66 %	52 %	61
Proyección de Estado		64 %	83 %	69

Cuadro 4.2: Resultados del Testeo para predicción de Impacto Negativo

ponden a los distintos modelos de representación de documentos utilizados.

Dentro del conjunto de clasificadores *Negativos* se seleccionaron dos clasificadores para utilizar en el algoritmo de Weighted-Majority, los seleccionados en este caso fueron Winnow Positivo con HM Frames y el Bayesiano Simple que utilizaba Palabras Clave. Ambos fueron seleccionados a razón de que presentaban los mejores resultados en cuanto a precisión. Los resultados de un Weighted-Majority de estas características se presentan en la tercer fila del Cuadro [4.2].

De igual forma, dentro del conjunto de clasificadores *Positivos* se seleccionaron dos clasificadores para utilizar en el algoritmo de Weighted-Majority, los seleccionados en este caso fueron Winnow Positivo con tokens de Freeling y el Bayesiano Simple que utilizaba HM Frames. Ambos fueron seleccionados a razón de que presentaban los mejores resultados en cuanto a precisión. Los resultados de un Weighted-Majority de estas características se presentan en la tercer fila del Cuadro [4.3].

Por último se ejecutó el clasificador basado en la proyección de los estados de las acciones en la fecha de la publicación de cada noticia para contar con una medida base contra la cual comparar. Los resultados obtenidos se muestran en la última fila de cada cuadro.

4.3. Análisis de Resultados

El objetivo de las pruebas realizadas es evaluar la utilización de recursos lingüísticos para la clasificación de noticias en relación al impacto estimado que tendrán sobre el mercado. En primer lugar se debe evaluar el módulo de identificación de acciones, luego es necesario evaluar la clasificación de noticias. A continuación se presenta el análisis de las evaluaciones realizadas.

De los resultados obtenidos se puede observar que en general el identificador de acciones presenta el desempeño esperado, reconociendo las acciones cuyas referencias fueron correctamente registradas en ClaNFi. Para mejorar el desem-

Clasificador		Resultados		
Algoritmo	**Modelo**	**Precisión**	**Recall**	$F_{0,5}$
Winnow Positivo	HMFrames	-	-	-
	Stemming	46 %	29 %	38
	Freeling	62 %	29 %	45
	Palabras Clave	48 %	33 %	42
Bayesiano Simple	HMFrames	61 %	10 %	23
	Stemming	-	-	-
	Freeling	-	-	-
	Palabras Clave	-	-	-
Weighted-Majority		42 %	8 %	17
Proyección de Estado		25 %	41 %	29

Cuadro 4.3: Resultados del Testeo para predicción de Impacto Positivo

peño del componente que realiza el reconocimiento de acciones, recordemos que es posible agregar referencias de búsqueda (Por ejemplo: podemos agregar la referencia *Coke* para las acciones de *Coca Cola Company*).

Por otro lado, la clasificación de noticias según su impacto esperado funciona relativamente bien, es posible determinar un clasificador que establezca si una noticia tendrá un impacto negativo o positivo con una precisión por encima del 60 %.

Recordemos que para poder comparar, creamos un clasificador sencillo, que no toma en cuenta el contenido de la noticia, sino que predice el impacto de una noticia según la tendencia de las acciones referenciadas (la explicación completa del algoritmo utilizado se encuentra en la sección 3.1.2).

En cuanto a los clasificadores *Negativos*, el mejor clasificador resultó ser el que utilizaba el algoritmo de *Weighted-Majority* llegando a un 66 % de Precisión; aún así si tomamos como base de comparación la medida-$F_{0,5}$, el clasificador de *Proyección de Estado* tuvo mejores resultados (Cuadro [4.2]). Esto resultó bastante sorprendente, no solamente que el clasificador basado en la proyección de estados haya tenido mejor desempeño que los demás, sino que tiene muy buen desempeño. Esto parece indicar que cuando una acción esta perdiendo valor la mayoría de las noticias que se publiquen tendrán un impacto negativo: manteniendo o agravando la tendencia.

En cuanto a los clasificadores *Positivos*, el mejor clasificador resultó ser el que utilizaba el algoritmo de Winnow Positivo con *tokens* de Freeling, llegando a un 62 % de Precisión. En este caso, el clasificador de *Proyección de Estado* obtuvo unos resultados bastante inferiores, logrando apenas una 25 % de Precisión (Cuadro [4.3]). Algunas de las combinaciones de componentes de clasificación y representación de documentos clasificaron todos los documentos como *No Positivos* (en cuanto a su impacto), logrando así que su Precisión y Recuperación no sean calculables. Revisando las distintas fases de entrenamiento y testeo de estos clasificadores, se puede distinguir las siguientes observaciones. En el caso de *Winnow Positivo* con HMFrames el entrenamiento culmina con una cantidad

aceptable de errores, pero luego no tiene un desempeño adecuado en el conjunto de testeo. Analizando el detalle del clasificador obtenido, podemos inducir que está ajustado para clasificar documentos muy parecidos a los del conjunto de entrenamiento (esto se llama *Sobreajuste*). Puede que esto sea causa de la definición misma de los HMFrames, recordemos que capturan parte de la información sintáctica del texto. Por otro lado, el problema del clasificador basado en el algoritmo *Bayesiano Simple* parece ser el gran tamaño del conjunto de rasgos, como se observó en trabajos previos realizados por Koster [Kos06b], siendo esta una fuerte limitante del algoritmo.

Comparativamente no se pueden establecer demasiadas observaciones concluyentes sobre las distintas combinaciones de modelos de representación y algoritmos de clasificación. En general, y como consecuencia del ajuste de parámetros, el componente basado en el algoritmo de Winnow Positivo obtuvo alta precisión pero baja recuperación. Por otro lado, los clasificadores que utilizaron HMFrames obtuvieron en algunos casos mejor precisión, pero siempre contaron con menor recuperación que los demás. Seguramente, y como se explicó anteriormente, esto se deba principalmente a la definición de los HMFrames. En ambos conjuntos de clasificadores (*Negativos* y *Positivos*), y sin tener en cuenta el clasificador basado en *Weighted Majority*, las mejores precisiones se obtuvieron con modelos de representación que aportaban mayor nivel de análisis al texto: en el caso de los clasificadores *Negativos* los HMFrames y en el caso de los clasificadores *Positivos* los tokens de Freeling (que incluyen tags de *Postagging*).

Se debe tener en cuenta que los resultados están altamente ligados a la retroalimentación proporcionada. En este caso la retroalimentación de los algoritmos de clasificación se realiza de forma automática, utilizando como unidad temporal *el día*: se evalúa si una noticia fue positiva o negativa en relación al valor de la acción del día siguiente a la fecha de publicación de una noticia. Esta limitante puede levantarse consiguiendo fuentes que permitan evaluar el estado de las acciones en periodos más pequeños (por ejemplo: horas). En este proyecto no se contaba con los recursos necesarios para adquirir información con ese grado de detalle, ya que datos tan específicos no se encuentran disponibles en forma gratuita.

Capítulo 5

Conclusiones y Trabajo Futuro

Este proyecto se enmarca en un objetivo más general, que tiene como meta final la predicción del comportamiento de las acciones (activos de inversión), a través de la captura de la percepción que el mercado tiene de su comportamiento futuro. En particular, este proyecto pretende evaluar la construcción de una herramienta que permita realizar predicciones sobre el movimiento de las acciones utilizando como referencia la información periodística. De esta forma se plantea el objetivo especifico de construir un dispositivo que permita el análisis automático de noticias financieras, de modo de identificar las acciones referenciadas en su contenido y clasificar cada noticia en términos de su impacto estimado relacionado al valor de la acciones involucradas en el mercado, indicando si la noticia tendrá un impacto positivo o negativo.

En primer lugar, se diseño una plataforma para clasificar y procesar documentos. El motor de ClaNFi es altamente flexible e independiente del tipo de documento a manejar y de los componentes o módulos de procesamiento a utilizar. Es posible agregar, quitar y configurar los componentes de procesamiento o clasificación que se aplicarán a los distintos documentos. Adicionalmente, el motor de ClaNFi provee facilidades para la clasificación de documentos, como por ejemplo: un componente que dado un clasificador binario, conjuntos de entrenamiento y testeo, realiza las actividades de entrenamiento, testeo y el cálculo de medidas de desempeño (precisión, recuperación y medida-F_α) de forma automática.

En relación al objetivo particular de la clasificación de noticias financieras se realizó un prototipo sobre la plataforma antes mencionada. Este prototipo, llamado ClaNFi, permite recolectar noticias RSS, realizando luego un análisis automático de cada una. El análisis consiste en la identificación de las acciones referenciadas y la estimación del impacto que tendrá la noticia sobre el mercado en referencia a las acciones involucradas. Para la realización de este prototipo se estudiaron trabajos previos en temas relacionados a la clasificación y representación de documentos, de forma de incorporar los algoritmos y componentes más prometedores. Así se incluyeron varios algoritmos de clasificación (*Naive*

Bayes, *Winnow Positivo*, *Weighted Majority*) y varios modelos de representación de documentos (Palabras Clave, *tokens* de Freeling, *HM Frames*). Luego se proyectaron y efectuaron varias pruebas, que incluyeron diferentes combinaciones de clasificadores y modelos de representación. Se obtuvieron, entonces, resultados de desempeño de la solución planteada, los cuales son satisfactorios si tenemos en cuenta que es un prototipo inicial: ClaNFi predice si el impacto de una noticia será positiva, negativa o neutra con una precisión por encima del 60 %.

El desempeño del prototipo en referencia la clasificación de noticias no resultó demasiado prometedor. Las causas pueden ser varias: quizás los algoritmos de clasificación no fueron los correctos o funcionaron mal, del mismo modo, quizás los modelos de representación de documentos no fueron los adecuados. A lo largo del proyecto se intentó mitigar el riesgo de ocurrencia de estos problemas, así es como se utilizaron varios algoritmos de clasificación y varios modelos de representación diferentes. Por otro lado el problema se podría originar en el conjunto de hipótesis seleccionado: de este modo, quizás la teoría del mercado eficiente no sea del todo correcta. Mucho se ha escrito sobre la validez de esta teoría, siendo la forma Semi-Fuerte la variante más controvertida. Algunos estudios empíricos en el área (siendo quizás el más relevante él de Clarke, Mandelker y Jandik [JC01]) indican que el mercado reacciona con extrema rapidez a las noticias y anuncios financieros, en la mayoría de los casos se habla de minutos; de modo que el modesto desempeño del prototipo podría deberse al mecanismo de retroalimentación planteado (recordemos que establece el impacto de un noticia utilizando el estado de las acciones referenciadas en el día de publicación y en el día siguiente). Finalmente, se podría suponer que el problema planteado es realmente muy complicado para modelar y encontrar una solución de forma automática.

En cuanto al desempeño de los algoritmos de clasificación y modelos de representación de documentos utilizados, se realizaron algunas observaciones relevantes: el algoritmo de *Winnow Positivo* presenta algunas ventajas importantes sobre el Bayesiano Simple, la más importante es que permite tener un gran conjunto de rasgos en los documentos. El Bayesiano Simple, por su lado, requiere una cantidad acotada de rasgos para un funcionamiento adecuado. En referencia al desempeño de los modelos de representación, las mejores medidas de precisión se obtuvieron de los clasificadores que utilizaron *HM Frames* o *tokens* de Freeling, lo cual se debe seguramente a que estos modelos agregan un análisis más profundo al texto.

El proyecto realiza otros aportes destacables en el área de la clasificación de noticias, por ejemplo: se presenta un método de retroalimentación automático. Este método consiste en la definición y automatización de un criterio que indique la correctitud de una predicción acerca del impacto de una noticia, ya sea positivo o negativo. Adicionalmente se presenta un clasificador de referencia (*baseline*) para la clasificación de noticias según su impacto en referencia al valor de las acciones involucradas, ya sea negativo, positivo o neutro. Este clasificador utiliza una proyección del estado de las acciones, de modo que, sin tomar en cuenta el contenido de las noticias, realiza una clasificación basada en la tendencia del valor de las acciones en los días previos a la publicación de la

noticia.

5.1. Trabajo Futuro

La herramienta plantea varias mejoras. Por un lado, sería muy positivo contar con un corpus de noticias clasificadas por un humano. De esta forma, los resultados que se obtengan serán mucho más confiables que los que se obtuvieron en este trabajo. Esto se debe a las incertidumbres inherentes al método utilizado para la *Clasificación según Proyección de Estado de las Acciones* [3.1.2]. De modo de introducir al lector al problema se plantea la siguiente situación: supongamos tenemos dos noticias financieras que refieren a la misma acción y son publicadas en el mismo día. Una noticia es claramente positiva, en referencia a su impacto sobre el mercado, y la otra es claramente negativa. Finalmente, sabemos que la tendencia del valor de la acción será positivo para los próximos días. Entonces, nuestro clasificador basado en la proyección de estado nos indicará que ambas noticias tendrán un impacto positivo. De hecho, sabemos que es posible la existencia de errores que hayan sido introducidos por el método de evaluación de los clasificadores. De este modo, sería bastante prometedora la evaluación de ClaNFi contra criterios de técnicos expertos en el área.

ClaNFi no realiza una predicción sobre el grado de impacto de las noticias. O sea, además de predecir si la noticia será positiva o negativa, también sería deseable que indicará en el grado del impacto (por ejemplo: alto, medio o bajo). Este problema plantea nuevos desafíos. En principio es necesario definir claramente el alcance de las categorías correspondientes al grado de impacto. Luego, habría que seleccionar un corpus de referencia y lograr obtener una clasificación real. Llegado a este punto, sería interesante realizar pruebas sobre los clasificadores que ya están incluidos en ClaNFi, de modo de evaluar los algoritmos ya implementados. Dependiendo de los resultados que se obtengan quizás sea necesaria la construcción de nuevos clasificadores o componentes de procesamiento de lenguaje natural.

Un tema planteado al inicio pero que finalmente quedo afuera del alcance de ClaNFi es el concepto de la *certeza* de ocurrencia de una noticia. Esto se refiere a la posibilidad de determinar automáticamente la certeza de ocurrencia del evento referido en la noticia. Por ejemplo: si la noticia se refiere a la suba de precios del petróleo en el día de ayer, la certeza será alta, ya que el evento ya ocurrió. La *certeza* está estrechamente vinculada al carácter especulativo de las noticias y es un tema bastante complicado de definir. Agregar un análisis de *certeza* permitiría al usuario un mejor manejo de la noticia, pero comprende un procesamiento del lenguaje natural más sofisticado que el que actualmente realiza ClaNFi, se debería incluir análisis sintáctico y semántico.

Apéndice A

Cronograma

En la figura [A.1] se presenta el cronograma del proyecto. En el cual se identifican algunas etapas bien diferentes: por un lado, la construcción del conjunto de testeo consumió bastante tiempo del proyecto, dadas las características específicas con las cuales debía contar. Adicionalmente, su posterior análisis también llevó un tiempo considerable. Luego, fue posible a partir de los resultados obtenidos, diseñar un prototipo adecuado. El proceso de testeo del prototipo (incluido dentro de la etapa de construcción) implicó un trabajo arduo que debió ser acompañado de algunos ajustes al prototipo, de modo de contemplar algunos casos particulares o configuraciones adicionales.

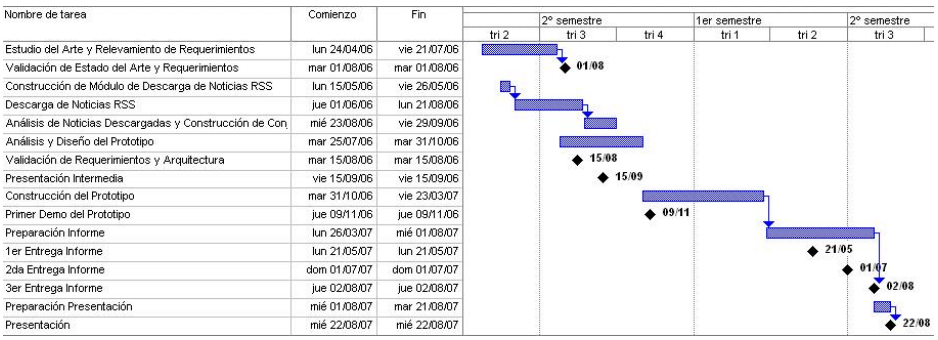

Figura A.1: Cronograma del Proyecto

Glosario

Acciones Las acciones son el producto del mercado bursátil más conocido por los inversores. Las empresas tienen su capital dividido en acciones. Cuando una empresa cotiza en Bolsa, sus acciones pueden negociarse en el mercado bursátil; los compradores y vendedores determinan el precio de las acciones. El resultado de multiplicar el precio de la acción en el mercado por el número de acciones existentes es igual al valor bursátil o capitalización de la empresa. Este criterio es muy útil para determinar el valor real de una empresa. La determinación del precio de acciones de las empresas supone, en definitiva, la valoración que hace el mercado sobre las expectativas de las empresas que cotizan.

Analizador Sintáctico Un analizador sintáctico en informática y lingüística es un proceso que analiza secuencias de tokens para determinar su estructura gramatical respecto a una gramática formal dada.

API Una API (del inglés Application Programming Interface - Interfaz de Programación de Aplicaciones) es el conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

Baseline Referencia inicial; también: valor de referencia, punto de partida, situación inicial, punto de comparación.

Canales de noticias RSS Los canales de noticias (*rss channels*) son agrupaciones temáticas que se realizan sobre las noticias rss que se publican en una fuente. De este modo, es posible, que una fuente como *El País digital* pueda publicar sus noticias agrupadas por tema, mediante canales, como por ejemplo *Finanzas*, *Clasificados* o *Sociales*.

Corpus Conjunto de textos o analizar. Es el cuerpo del lenguaje, así como del texto y del discurso, que proporciona las bases para el análisis del lenguaje para estableces sus características y verificar empíricamente una teoría referida al lenguaje, entre otras cosas. Los corpus están sujetos a un proceso conocido como etiquetado. Un ejemplo de esto es el etiquetado del tipo de palabra, en el cual se añade en forma de etiqueta la información gramatical sobre cada palabra.

Mercados Bursátil En él se negocian las acciones de las empresas. Aquí se encuentra lo que comúnmente se entiende por Bolsa.

Mercados Financieros Los mercados son lugares donde se pueden comprar y vender cosas, financiero es, esencialmente, dinero. El concepto parece simple, pero en realidad el objeto de las transacciones es el tiempo, pues se compra o se vende capacidad adquisitiva en el futuro.

Plug-in Un plugin (o plug-in –en inglés enchufar—, también conocido como addin, add-in, addon o add-on) es una aplicación informática que interactúa con otra aplicación para aportarle una función o utilidad específica, generalmente muy específica, como por ejemplo servir como driver en una aplicación, para hacer asi funcionar un dispositivo en otro programa. Ésta aplicación adicional es ejecutada por la aplicación principal.Se utilizan como una forma de expandir programas de forma modular, de manera que se puedan añadir nuevas funcionalidades sin afectar a las ya existentes ni complicar el desarrollo del programa principal.

POS tagging El propósito principal del proceso llamado POS tagging (o part-of-speech tagging) es asociar cada palabra a un texto con su categoría morfosintántica (esta representada por un tag).

Red Neuronal Las redes de neuronas artificiales son un paradigma de aprendizaje y procesamiento automático inspirado en la forma en que funciona el sistema nervioso de los animales. Se trata de un sistema de interconexión de neuronas en una red que colabora para producir un estímulo de salida. Una de las misiones en una red neuronal consiste en simular las propiedades observadas en los sistemas neuronales biológicos a través de modelos matemáticos recreados mediante mecanismos artificiales (como un circuito integrado, un ordenador o un conjunto de válvulas).

Reflection El API Reflection es un conjunto de funciones y procedimientos que acompañan a distintas versiones de Java. Con éste API es posible por ejemplo: crear una instancia de una clase cuyo nombre es desconocido hasta el momento de la ejecución; o invocar un método en un objeto, aún si el método es desconocido hasta el momento de la ejecución.

Retroalimentación La retroalimentación se produce cuando las salidas del sistema o la influencia de las salidas del sistemas en el contexto, vuelven a ingresar al sistema como recursos o información. La retroalimentación permite el control de un sistema y que el mismo tome medidas de corrección en base a la información retroalimentada.

RSS RSS es parte de la familia de los formatos XML desarrollado específicamente para todo tipo de sitios que se actualicen con frecuencia y por medio del cual se puede compartir la información y usarla en otros sitios web o programas. A esto se le conoce como redifusión o sindicación. El RSS no es otra cosa que un sencillo formato de datos que es utilizado para sindicar (redifundir) contenidos a suscriptores de un sitio web. El formato permite distribuir contenido sin necesidad de un navegador.

Sobreajuste En estadística, el sobreajuste es el ajuste a un modelo estadístico que tiene muchos parámetros. Un modelo absurdo y falso puede ajustarse perfectamente a la realidad si tiene suficiente complejidad en comparación a la cantidad de información disponible.

Token Unidad mínima de un lenguaje.

Valor de Cierre Es el valor de una acción al cierre del mercado financiero donde cotiza al fin del día.

Volumen Negociado Es el valor de una acción al cierre del mercado financiero donde cotiza al fin del día. Es un buen indicador de la actividad en el mercado. Además el volumen debe acompañar la tendencia del precio. De modo que cuando el volumen crece confirma la tendencia actual del precio.

XML XML, sigla en inglés de eXtensible Markup Language («lenguaje de marcas extensible»), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML). Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Algunos de estos lenguajes que usan XML para su definición son XHTML, SVG, MathML. XML no ha nacido sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable.

XSLT XSLT o XSL Transformaciones es un estandar de la organización W3C que presenta una forma de transformar documentos XML en otros e incluso a formatos que no son XML. Las hojas de estilo (aunque el termino de hojas de estilo no se aplica sobre la función directa del XSLT) XSLT realizan la transformación del documento utilizando una o varias reglas de plantilla: unidas al documento fuente a transformar, esas reglas de plantilla alimentan a un procesador de XSLT, el cual realiza las transformaciones deseadas colocando el resultado en un archivo de salida o, como en el caso de una página web, directamente en un dispositivo de presentación, como el monitor de un usuario.

Bibliografía

[Blu96] Avrim Blum, *On-line algorithms in machine learning*, Online Algorithms, 1996, pp. 306–325.

[CHAK] Erik Verbruggen Cornelis H. A. Koster, *The agfl grammar work lab*, Proceedings FREENIX/Usenix 2002, pp. 13–18.

[CK06] T. Verhoeven C.H.A. Koster, *Head/modifier frames for information retrieval*, http://www.cs.ru.nl/agfl/papers/hmex.pdf, 06 2006.

[DL01] Romesh Vaitilingam Dean LeBaron, *The ultimate investor: The people and ideas that make modern investment*, Capstone, 02 2001.

[Esp06] Real Academia Española, *Diccionario de la lengua española*, http://buscon.rae.es/draeI/, 2006.

[IDR97] Y. Karov I. Dagan and D. Roth, *Mistake-driven learning in text categorization*, Proceedings of the Second Conference on Empirical Methods in NLP, 1997.

[JC01] Tomas Jandik Jonathan Clarke, Gershon Mandelker, *The efficient markets hypothesis*, a book chapter published in Expert Financial Planning: Investment Strategies from Industry Leaders, 2001.

[Kos06a] C.H.A. Koster, *From key-words to key-phrases*, http://www.cs.ru.nl/peking/ict.pdf, 06 2006.

[Kos06b] ———, *Lectures notes for an advanced course in information retrieval*, http://www.cs.ru.nl/~kees/ir2/, 06 2006.

[KT97] C.H.A. Koster and C.P.A. Tiberius, *Agfl grammars for full-text information retrieval*, A. Ralli et al (Eds.), Working Papers in Natural Language Processing, Ekdosis Diavlos, 1997, pp. 139–154.

[Kur00] Howard Kurtz, *The fortune tellers: Inside wall street's game of money, media, and manipulation*, Free Press, 09 2000.

[LUH58] H.P. LUHN, *The automatic creation of literature abstracts*, IBM Journal of Research and Development **2** (1958), 159–165.

[MEF07] *Hipótesis de eficiencia de los mercados*, http://es.wikipedia.org/wiki/Hip%C3%B3tesis_de_eficiencia_de_los_mercados, 06 2007.

[Mit97] Tom M. Mitchell, *Machine learning*, McGraw-Hill Science/Engineering/Math, 1997.

[Por80] Martin F. Porter, *An algorithm for suffix stripping*, Program **14** (1980), 130–137.

[Tho03] James D. Thomas, *News and trading rules*, Ph.D. thesis, Carnegie Mellon University, 2003.