Ejemplos de aprendizaje automático

Héctor Jiménez Salazar
Fac. de Ciencias Físico Matemáticas, UAP
hjimenez@fcfm.buap.mx

septiembre, 2000

1 Aprendizaje Automático

En este capítulo se presenta una revisión de tres métodos de aprendizaje automático (AA) para poder seleccionar el que más se ajuste a la solución que será dada a algunos problemas del Procesamiento del Lenguaje Natural (PLN). La primera sección define los conceptos que van a ser empleados. Cada una de las tres secciones subsiguientes describe un método de AA, y, al final, se hace la elección de uno de ellos, comparando su precisión en el problema de desambiguación de palabras frecuentes con varias funciones lingüísticas en los textos. Empezaremos por motivar las nociones generales del AA.

Es general que los seres humanos asignemos un conjunto de rasgos a cada experiencia, y es común que, para resolver un problema, empecemos por clasificar sus partes. Aprendemos clasificando. La clasificación es considerada como una de las operaciones más empleadas en diversos campos de la ciencia. La relación entre aprendizaje y clasificación puede mostrarse con el área de la computación llamada *aprendizaje automático*.

El aprendizaje automático refiere a la realización automática de cambios en estructuras de información o procesos de cómputo que modifican en esencia el comportamiento de un sistema basado en esta entidad computacional, trátese de una estructura de información o de un proceso. Dichos cambios se representan en la misma entidad computacional que cambia. Es de interés, por tanto, para el aprendizaje automático estudiar la representación de entidades computacionales y, por supuesto, los métodos que subyacen a sus transformaciones. La representación de la entidad computacional sujeta a transformaciones procura ampliar su rango de aplicación, particularmente, después del aprendizaje, la nueva representación será más general. Hay dos modos de realizar el aprendizaje: supervisado, en el cual se tiene una muestra de problemas resueltos que "enseñan" al sistema cómo resolver nuevos problemas y, no supervisado, en donde deben descubrirse regularidades sin necesidad de contar con problemas ya resueltos. Nos interesa trabajar con un sistema supervisado. En este contexto, el problema de clasificación consiste en usar las regularidades "aprendidas" para resolver nuevos problemas.

Consideremos una entidad o fenómeno particular representado por sus rasgos. A dicha entidad la llamaremos *instancia*, o *ejemplar*; a cada conjunto que reuna un tipo de rasgos, *atributo*; y, consecuentemente, los *valores de un atributo* serán los *rasgos*. La clasificación supervisada se apoya en instancias previamente clasificadas. A esta colección de instancias se le designa **colección de entrenamiento**, de manera que podemos representar a la colección de entrenamiento como una reunión de *clases* de instancias.

Una forma de clasificar una instancia es adjudicándole la clase de una de las instancias de la colección de entrenamiento, aquélla que se parezca más a la instancia por clasificar. Otros métodos de clasificación tratan de ahorrar el trabajo de revisar la colección de entrenamiento completa para determinar la clase de una instancia y, de antemano, establecen reglas basadas en la colección de entrenamiento, para decidir la clase de la nueva instancia. Bastará pues, en estos casos, identificar algunos rasgos de la instancia por clasificar para determinar su clase. Presento en las siguientes secciones tres métodos dentro del campo del aprendizaje automático para analizar su empleo en el ámbito de la clasificación:

- La identificación de rasgos relevantes, para apoyar la determinación de la clase de un ejemplar.
- La síntesis de atributos que se refiere a la obtención de reglas de clasificación representadas en un árbol de decisión, y
- El razonamiento basado en memoria, cuya esencia es calcular la similitud entre instancias para determinar la clase de una nueva instancia.

En primer lugar formalizaremos algunos de los conceptos básicos de la clasificación.

2 Relaciones y entropía

Dada una colección de *atributos*, $\mathcal{A} = (A_p)_{p \leq P}$, y una función, Dom, que asigna a cada atributo su *dominio*, $A_p \mapsto \mathrm{Dom}(A_p) = D_p = \{x_{pj}\}_{j \leq n_p}$, definimos una *relación* R, con *esquema* $\mathcal{A}$, como un subconjunto del producto cartesiano de los dominios de los atributos de $\mathcal{A}$, $R \subset D = \prod_{p \leq P} D_p$. Veremos a R como la *colección de entrenamiento* y para cualquier *instancia* x, miembro de R, nos referimos indistintamente a su p-ésima componente x_p o *proyección* p de x, $\pi_p : R \to D_p$, $x \mapsto x_p$.

Debido a que la relación R contiene instancias clasificadas, consideraremos, para toda $x \in R$, uno de los rasgos, digamos x_q, como la clase a la cual pertenece x. Al atributo A_q lo llamaremos *atributo de decisión*. Nos interesa resolver el problema de caracterización de las instancias de una relación que tienen un determinado valor en la componente q. Es decir, cuáles son, si es que existen, los valores de otras componentes de las instancias de R que permitan asegurar que en la componente q se presenta un valor x_{qk}. De esta forma podrá establecerse una dependencia entre el valor de un atributo de una instancia por clasificar y la clase a la que pertenece.

Una *dependencia funcional* $A_p \to A_q$ entre dos atributos de una relación R es consistente con un subconjunto S de R, si $\forall x, y \in S(\pi_p(x) = \pi_p(y) \Rightarrow \pi_q(x) = \pi_q(y))$. Para un atributo particular A_p, consideramos una *partición* $\mathcal{R}_p$ de R en clases $R|_{x_{pj}} = \pi_p^{-1}(x_{pj}), j \leq n_p$, esto es: $\bigcup_{j \leq n_p} R|_{x_{pj}} = R$ y para $k \neq j$, $R|_{x_{pk}} \cap R|_{x_{pj}} = \emptyset$. Sean $\mathcal{R}_s$ y $\mathcal{R}_t$ particiones de R, decimos que $\mathcal{R}_s$ *refina* a $\mathcal{R}_t$ si $\forall R_s \in \mathcal{R}_s(\exists R_t \in \mathcal{R}_t$, tal que $R_s \subset R_t)$.

En las relaciones que exploramos no es común el cumplimiento cabal de una dependencia funcional consistente con toda la relación R, es por ello que requerimos determinar el 'grado' de dependencia entre dos atributos. Observemos por ejemplo que si $\mathcal{R}_p$ refina a $\mathcal{R}_q$ entonces $A_p \to A_q$ es consistente con R. Intuitivamente, hay mayor dependencia de A_q en A_p que de A_q en A_r si la partición que induce A_p refina a A_q en un subconjunto de R significativamente mayor que en el que lo hace A_r. Es en este sentido que nos referimos a que A_p es más *relevante* que A_r. La noción de grado de dependencia entre atributos puede precisarse de varias formas. En la sección 3 lo vamos a hacer con base a una prueba estadística.

Para una función f con dominio A, denotaremos con $\mathrm{argmin}_{x \in A} f(x)$ el valor que es argumento de f y cuya imagen es menor. En forma semejante $\mathrm{argmax}_{x \in A} f(x)$, es el valor que maximiza a la función.

Otro concepto muy útil es la *entropía* o *cantidad de información*. Decimos que un rasgo provee más cantidad de información que otro, si el primero tiene menos oportunidades de determinar la clase de una instancia. O bien, que entre menos relevante resulte la presencia de x_{pj}, para que cualquier instancia x con este valor pertenezca a una clase, menos cantidad de información proveerá x_{pj}. La motivación de este concepto es que los eventos muy probables tienen poca información que añadir, mientras que los eventos poco probables anuncian mucha información faltante, es decir *incertidumbre*. La *entropía* (o cantidad de información) de un estado s_i se define como

$$-\log_2 \Pr(s_i)$$

Esta medida tiene unidades de *bits*. En un sistema con estados $S = \{s_1, \ldots, s_k\}$ la cantidad de información sería:

$$H_S = -\sum_{i=1}^{k} \Pr(s_i) \log_2 \Pr(s_i). \tag{1}$$

Consideramos a la relación R con esquema $(A_1, \ldots, A_q)$ como un sistema cuyos posibles estados son las clases que define la partición, $\mathcal{R}_q$. Supongamos que $R_1 \subset R$, y que $R_1|_{x_{pj}}$ está contenido en un miembro de $\mathcal{R}_p$. La cantidad de información de R_1 según la repartición en clases que se hace con los valores de A_p, o simplemente respecto del atributo A_p, es el promedio de las cantidades de información que provee cada uno de los subconjuntos $R_1|_{x_{pj}}, x_{pj} \in D_p$, esto es:

$$H_{R_1(A_p)} = \sum_{x_{pj} \in D_p} H_{R_1|_{x_{pj}}} \frac{\#R_1|_{x_{pj}}}{\#R_1}. \tag{2}$$

La noción de cantidad de información de R_1 respecto a A_p se interpreta como una relación entre la cantidad de información de dos sistemas: la cantidad de información de R_1 (la incertidumbre de que las instancias de R_1 pertenezcan a una clase) y la cantidad de información de A_p (la incertidumbre de que las instancias de R_1 tengan un valor en su componente p). $H_{R_1(A_p)}$ es pues la cantidad de información de que dado un determinado valor de A_p una instancia de R_1 pertenezca a una clase, y es menor que la cantidad de información de R_1, ya que al conocer algo sobre R_1 no puede aumentar su incertidumbre, $H_{R_1(A_p)} \leq H_{R_1}$, más bien se dice que R_1 "pierde" información, por lo cual se define $H_{R_1} - H_{R_1(A_p)}$ como la *ganancia de información* de A_p.

La cantidad de información se empleará en dos de los algoritmos que van a exponerse. En el primero (sec. 4), la cantidad de información nos ayuda a optimizar la construcción de un árbol de decisión. En el segundo algoritmo (sec. 5), esta medida ayuda a ponderar la comparación entre los rasgos de dos instancias.

3 Identificación de rasgos relevantes

La idea central de este método ([Chan, Wong, 91][1]) es realizar una prueba de hipótesis que nos permita decidir si un atributo depende de otro. Para seleccionar los valores que determinan la clase de una instancia, el método parte de una colección de entrenamiento y revisa en cada clase si es significativo, estadísticamente, la ocurrencia de un rasgo. A cada rasgo, entonces, podrá corresponder una clase. Siguiendo este procedimiento para cada clase y para todos los rasgos, se encuentra una serie de parejas, (rasgo-clase): un rasgo que propone, con cierto nivel de confianza, la clase a la que pertenecería cualquier otra instancia que contenga dicho rasgo. Dada, pues, una instancia por clasificar, determinamos tantas clases como propuestas haga cada uno de sus rasgos que hayan sido seleccionados por el método. De todas estas clases elegimos aquélla que haya sido propuesta por el mayor número de rasgos.

3.1 Rasgos que determinan una clase

Tomemos a A_q como atributo de decisión y algún otro atributo A_p para el que deseamos identificar si depende A_q. Supongamos que $\#R = N$, y consideremos a las tablas de contingencia $O_{pq} = [o_{jk}^{pq}]_{j,k}$, donde $o_{jk}^{pq} = \#\{x|\pi_p(x) = x_{pj} \wedge \pi_q(x) = x_{qk}\}$. Recordemos que la frecuencia esperada de las instancias que tienen los valores x_{pj} y x_{qk} se calculan mediante $e_{jk}^{pq} = \sum_j o_{jk}^{pq} \times \sum_k o_{jk}^{pq}/N$. En una prueba de independencia usando χ^2 entre los atributos A_q y A_p con un nivel α, probamos que si $\sum_{j,k}(o_{jk}^{pq} - e_{jk}^{pq})^2/e_{jk}^{pq}$ es mayor al valor que determina α para χ^2 entonces se rechaza la independencia. No es posible, sin embargo, determinar, en caso de dependencia, qué clase depende de cierto valor del atributo p.

Para averiguar la caracterización de las instancias de una clase $R|_{x_{qk}}$ mediante valores del atributo A_p, comparemos $\Pr(\pi_q(x) = x_{qk}|\pi_p(x) = x_{pj})$ con $\Pr(\pi_q(x) = x_{qk})$, es decir, comparemos la probabilidad condicional de que una instancia x pertenezca a $R|_{x_{qk}}$ dado que x tiene en su p-ésima componente el valor x_{pj}, y la probabilidad de que la instancia x pertenezca a la clase $R|_{x_{qk}}$. Si

$$\Pr(\pi_q(x) = x_{qk}|\pi_p(x) = x_{pj}) > \Pr(\pi_q(x) = x_{qk}) \tag{3}$$

[1] Presentamos la base en que se sustenta este método, lo cual es motivado por el empleo de la *información mutua* que es de interés en el PLN, pero no será seguido como aparece en [Chan, Wong, 91].

entonces podríamos reconocer que es más probable que una instancia x pertenezca a $R|_{x_{qk}}$ cuando tenga en su p-ésima componente el valor x_{pj}. En este caso elegiríamos como característico de la clase $R|_{x_{qk}}$ al valor x_{pj} de la p-ésima componente. Puesto que

$$\Pr(\pi_q(x) = x_{qk} | \pi_p(x) = x_{pj}) = \Pr(\pi_q(x) = x_{qk}, \pi_p(x) = x_{pj}) / \Pr(\pi_p(x) = x_{pj}),$$

la desigualdad 3 puede ser reescrita de la siguiente manera:

$$\begin{aligned} \Pr(\pi_q(x) = x_{qk}, \pi_p(x) = x_{pj}) &> \Pr(\pi_p(x) = x_{pj}) \Pr(\pi_q(x) = x_{qk}) \\ N \cdot \Pr(\pi_q(x) = x_{qk}, \pi_p(x) = x_{pj}) &> N \cdot \Pr(\pi_p(x) = x_{pj}) \cdot N \cdot \Pr(\pi_q(x) = x_{qk}) / N \\ o_{jk}^{pq} &> o_{j+}^{pq} o_{+k}^{pq} / N = e_{jk}^{pq} \end{aligned} \tag{4}$$

donde 4 refiere al cálculo de la frecuencia esperada a partir de los totales marginales. Tenemos, por tanto, que la hipótesis de caracterizar una clase con algún valor de otro atributo se transforma en probar la hipótesis de que las frecuencias de los valores observados en un atributo de las instancias son mayores que las frecuencias de los valores esperados en ese mismo atributo. Complementariamente, si se tiene la igualdad entonces habría independencia entre los atributos A_q y A_p y, si se cumple además que $\Pr(\pi_q(x) = x_{qk} | \pi_p(x) = x_{pj}) < \Pr(\pi_q(x) = x_{qk})$, podría afirmarse que dado el valor x_{pj} en una instancia es menos probable que pertenezca a la clase $R|_{x_{qk}}$.

Dado que hay una correspondencia entre una distribución normal con media cero y varianza σ^2, $n(0, \sigma^2)$, con una distribución $\chi^2(1)$ podemos remplazar una prueba de independencia entre atributos por una prueba basada en la distribución normal de la variable aleatoria:

$$m_{jk} = \frac{o_{jk}^{pq} - e_{jk}^{pq}}{\sqrt{e_{jk}^{pq}}} \tag{5}$$

para la cual la hipótesis de independencia conduce a suponer que su media es diferente de cero. Antes de proceder con esta prueba debemos estandarizar la distribución 5 y, para hacerlo, acudimos al cálculo de su varianza mediante la siguiente fórmula [Chan, Wong, 91]:

$$v_{jk}^{pq} = \sqrt{\left(1 - \frac{o_{j+}^{pq}}{N}\right)\left(1 - \frac{o_{+k}^{pq}}{N}\right)}$$

por lo tanto

$$d_{jk}^{pq} = m_{jk}^{pq} / v_{jk}^{pq} \tag{6}$$

Así pues, d_{jk}^{pq} representa la diferencia de los valores esperados de las observaciones y las estimaciones del número de instancias que tienen los rasgos x_{pj} y x_{qk}. Según la derivación que hemos realizado, d_{jk}^{pq} tiene la densidad normal $n(\mu, \sigma^2)$, con $\mu = 0$ y $\sigma^2 = 1$,

Ahora probar que la clase k de A_q depende del valor j de A_p significa probar la hipótesis $H_0 : \mu > 0$ (vs $H_1 : \mu = 0$). En esta prueba, si $d_{jk}^{pq} > 1.96$ podemos aceptar que la clase k depende del valor x_{pj}, con 95% de confianza.

Por cada aplicación de la prueba de hipótesis es posible que encontremos el valor de un atributo del cual depende una de las clases de A_q, esto es, una pareja *(clase, rasgo indicador)*: $x_{qk} \leftarrow x_{pj}$. Al variar j, fijando un valor x_{qk}, se agregarán otros valores del atributo A_p a los rasgos indicadores que ya hemos determinado. Sucesivamente, para cada $p \in \{1, \ldots, P\} \setminus \{q\}$, encontramos todos los posibles rasgos indicadores de la clase x_{qk} que proporciona la colección de entrenamiento R: $x_{i_1 k_1} \ldots x_{i_j k_j}$.

Las parejas (clase, rasgo indicador) pueden verse como reglas que determinan que: cualquier otra instancia, x, pertenecerá a la clase que representa x_{qk}, si cualquier x contiene alguno de los valores $x_{i_1 k_1} \ldots x_{i_j k_j}$[2]. En la figura 1 se muestra de manera resumida los pasos anteriores, haciendo referencia a la ecuación 6 con la función $normaliza(p, q, j, k, R)$.

[2]Observemos que, también, podemos tener parejas que "rechacen" la asignación de una clase a una entidad, cuando $d_{jk}^{pq} < -1.96$. El método que fue implantado no usa este tipo de parejas.

Input:
The relation R, a threshold value s
and a fixed class k from the q-th attribute

Output:
Hypothesized dependencies pairs: $x_{qk} \leftarrow x_{pj_1}, \ldots$

Algorithm:
foreach $j \in [1, n_p]$
  println; print $R(k, q)$"←";
  foreach $p \in [1, P] \wedge p \neq q$
    let $d = normaliza(p, q, j, k, R)$
    if $d > s$: print $R(j, p)$
    else continue
end

Figura 1: Algoritmo para encontrar los rasgos que determinan una clase.

Observemos que aún cuando cada uno de los rasgos indicadores nos permiten, con cierto nivel de confianza, garantizar la clase de una instancia, no es necesario que todos ellos aparezcan en la instancia. Más bien, basta que uno de estos rasgos ocurra para asegurar, con cierto grado de confiabilidad, la pertenencia de la instancia a la clase que determina el rasgo. El conflicto entre las diferentes clases que pueden ser propuestas por algunos rasgos, de una instancia sin clasificar, puede ser resuelto eligiendo la clase que sea propuesta por mayor número de rasgos, considerando, en caso de empate, la clase que tenga mayor confiabilidad. Los resultados del algoritmo antes explicado se interpretan como "votaciones por una clase" [Oflazer, Tür, 97].

4 Síntesis de atributos

Hacer más sintéticos los patrones formados por valores de atributos que determinan la clase de una instancia significa, bajo algún criterio determinado, eliminar aquellos patrones y atributos considerados redundantes. El método que se expondrá está basado en los árboles de decisión. El uso de un árbol de decisión para clasificar instancias proviene del hecho de que con una colección de entrenamiento y eligiendo cualquier atributo, podemos *bifurcar* o *repartir* la colección original en tantas subcolecciones como valores tenga el atributo. Por cada *bifurcación* o *rama* asociada a un rasgo del atributo elegido, se tendrá una colección restringida a contener, en todas sus instancias, ese rasgo. El propósito de bifurcar una colección en varias subcolecciones es conocer si una subcolección es *homogénea*, es decir, si todos los miembros, de alguna de las subcolecciones que se obtienen, pertenecen a la misma clase. Si es así, puede establecerse que el rasgo usado para realizar esa bifurcación determina la clase de las instancias. En caso de que una subcolección sea heterogénea, ésta se compondrá de instancias de al menos dos clases diferentes, podremos entonces repartirla nuevamente eligiendo otro atributo[3]. Al final del proceso habremos conseguido una *estructura jerárquica* o *árbol* en el que cada punto de bifurcación representa una colección de instancias y cada bifurcación o rama que parte de un punto de bifurcación representa un rasgo del atributo elegido para llevar a cabo la repartición. Así pues, una bifurcación establece la *condición* de que todas las instancias de la colección a la que conduce tengan el mismo valor de un atributo. Al *recorrer el camino* que definen las bifurcaciones, empezando por el punto de bifurcación inicial, que es la colección de entrenamiento completa, pasaremos a través de las ramas, a cada vez más reducidas subcolecciones de instancias, hasta llegar a una colección homogénea. En

[3]Observemos que si dos instancias idénticas tienen la misma clase, estaremos frente a un problema de falta de información para poder discernir.

este recorrido, podremos acumular las condiciones que establecen las bifurcaciones contenidas en el camino. Tenemos así, que un recorrido desde la raíz hasta una colección homogénea proporciona un conjunto de rasgos que determinan, según la colección de entrenamiento, la pertenencia a la clase de la colección homogénea en la que termina ese recorrido. El problema de clasificación se traduce en que: cualquier otra instancia que coincida con alguna de las secuencias de rasgos, definidos por un recorrido, pertenecerá a la clase de la colección homogénea donde termina ese recorrido.

Puesto que la construcción de un árbol de decisión, como se ha expuesto, elige arbitrariamente los atributos para cada punto de bifurcación, existirán, para una colección de entrenamiento dada, varios posibles árboles de decisión. Claramente, el árbol óptimo es el que más favorece la determinación expedita de la clase de una instancia. Es decir, un árbol con la menor cantidad posible de ramas. En el método que vamos a explicar se elige, en cada punto de bifurcación, el atributo cuyos rasgos proporcionan la menor cantidad de información sobre cada una de las clases. La razón de esta elección descansa en el hecho de que con ese atributo se tienen más oportunidades de determinar la clase de las instancias de las colecciones derivadas de ese punto de bifurcación y, por tanto, es más probable que las instancias queden reunidas en subcolecciones homogéneas. Otra manera de ver lo anterior es refiriéndonos a la ganancia de información: se elige el atributo de máxima ganancia de información puesto que esperamos que, al repartir con él, las nuevas subcolecciones tengan reducida su cantidad de información.

4.1 Construcción de un árbol de decisión

Sea R una relación con esquema $\mathcal{A} = (A_1, \ldots, A_q)$ si $R_1, R_2 \subset R$, decimos que R_1 *precede* a R_2 en x_{pj}, $R_1 \prec_{x_{pj}} R_2$, si R_2 es el resultado de la selección de las instancias con valor x_{pj} a partir de R_1: $R_2 = R_1|_{x_{pj}}$. Llamemos *condición* a la restricción de una relación a un valor x_{pj}. Para precisar qué es un árbol de decisión, recordemos primero que un árbol W se define mediante una relación V en un conjunto de nodos E, de tal forma que existe un único nodo llamado raíz que no tiene predecesores y cuyos sucesores pueden verse como raíces de j árboles repartiendo en cada uno de ellos el resto de los nodos de E. En un árbol W, la secuencia de arcos $b_1, \ldots, b_t$, es una *trayectoria* si existe una secuencia de nodos $c_1, \ldots, c_{t+1}$ y b_h conecta a c_h con c_{h+1}, $(1 \leq h \leq t)$. Un *árbol de decisión*, W, para una relación R, se construye definiendo como raíz a R y todo nodo R_s de nivel i cumple que $\exists R_r$ de nivel $i-1$ y $x_{p_j k} \in A_{p_j}$ $(1 \leq p_j < q)$ tal que $R_r \prec_{x_{p_j k}} R_s$, además si R_s es terminal se tiene que $R_s|q_l = R_s$ para algún $q_l \in A_q$.

Si W es un árbol de decisión de R entonces cada trayectoria $b_1, \ldots, b_t$ desde la raíz hasta un nodo terminal representa una secuencia de condiciones sobre valores de atributos que son satisfechas por las instancias del nodo terminal. El problema de síntesis trata de reducir el tamaño de las trayectorias y el número de éstas. Como se ha dicho el criterio de reducción se apoya en el concepto de cantidad de información.

En un árbol de decisión W podemos ver a la clase de una instancia como un estado. Observemos, además, que si S tiene instancias de sólo una clase, entonces $H_S = 0$. Para sintetizar las instancias que contiene una relación R, debemos construir un árbol de decisión que sea óptimo, en el sentido de que sus trayectorias sean lo más cortas posibles y en menor número. Si W se construye de manera que los nodos elegidos tengan la menor cantidad de información ([Quinlan, 93]), o mayor ganancia de información, entonces estaremos eligiendo como constituyentes de esos nodos, instancias más probables y por tanto restarán menos instancias para clasificar. Al construir el árbol de esta forma es más probable que se eliminen algunas trayectorias. También, las trayectorias pueden reducirse deteniendo la bifurcación cuando $H_S = 0$.

Así pues, es necesario elegir los atributos en un orden tal que el número de trayectorias sea mínimo, y esto significa elegir cada vez el atributo que proporcione la mayor ganancia de información: $\text{argmin}_{A_r \in \mathcal{A} \setminus \{A_q\}} H_{R_i(A_r)}$. Este es el criterio usado en el algoritmo de Quinlan.

Ordenemos ahora los pasos para construir el árbol de decisión óptimo W, asociado a una relación R con esquema $\mathcal{A} = (A_1, \ldots, A_q)$ y atributo de decisión A_q. Por definición se toma a R como raíz y se itera el siguiente procedimiento para definir los arcos: de acuerdo con el criterio de Quinlan se elige un atributo para realizar una partición en el nodo en curso R_i, de tal manera que los elementos de esta partición, considerados como nuevos nodos, queden pendientes hasta que sea su turno partirlos. Eventualmente, habrá subconjuntos de R que no sea necesario partir, puesto que contendrán únicamente instancias de una sola clase de $\mathcal{R}_q$ (su entropía es cero) y que, como nodos del árbol, representan a las hojas. El algoritmo mostrado en la figura 2 precisa estos pasos.

Input:
The relation R with scheme $\mathcal{A}$ and the decision attribute A_q.

Output:
A decision tree W given as a collection of edges and its root.

Algorithm:
$push(Queue, R)$
let $E = \emptyset$
while $Queue \neq \emptyset$
  let $R_i = \text{pop}(Queue)$
  let $A_p = \text{argmin}_{A_r \in \mathcal{A} \setminus \{A_q\}} H_{R_i(A_r)}$
  if $H_{R_i(A_p)} \neq 0$
    foreach $j \in [1, n_p]$
      let $E = E \cup \{(R_i, R_i|_{x_{pj}})\}$
      $\text{push}(Queue, R|_{x_{pj}})$
end

Figura 2: Algoritmo para construir un árbol de decisión.

5 Razonamiento basado en memoria

Dentro del área de la Inteligencia Artificial un problema puede verse como un estado inicial y su solución como un estado meta, al cual se llega mediante una secuencia de operadores aplicados a partir del estado inicial. A fin de pasar a un nuevo estado, tendrá que aplicarse un operador y esto obliga a que sus condiciones sean satisfechas, es decir, las condiciones de un operador se cumplen si el estado en curso permite aplicar el operador. Este enfoque requiere de un mecanismo de búsqueda para encontrar la secuencia de operadores que producen el estado meta. Normalmente en esta tarea se hace uso de un empate exacto al comparar un estado con las condiciones del operador que se desea aplicar. Hay, sin embargo, criterios menos restrictivos que el empate exacto, como el siguiente [Stanfill, Waltz, 86]: Elegir el operador cuyas condiciones se acerquen más al estado en curso. Es necesario, entonces, medir la cercanía entre estados y, por tanto, definir una función de distancia.

Designada la búsqueda como el principal componente del razonamiento mecánico, un problema de clasificación puede expresarse de la siguiente manera: dada una nueva instancia, a partir de la colección de entrenamiento con instancias compuestas cada una de sus rasgos y su clase, debe elegirse la clase de la instancia de la colección cuyos rasgos se parezcan más a la nueva serie de rasgos. El método que va a exponerse recurre a la colección de entrenamiento completa y no sólo a rasgos particulares o a un subconjunto de éstos, para determinar la clase de una instancia. Este método se dice *basado en memoria* pues hace uso de una descripción concreta de todos los ejemplos, sin abstracciones.

Para clasificar una nueva instancia, es natural que se observe la colección de entrenamiento con el fin de adjudicarle a esta nueva instancia la clase de aquella instancia de la colección que más se parezca. El presente método no hace más que eso, precisando el significado de "parecido" o *similitud* entre instancias. El concepto de similitud también puede traducirse en *distancia*: a mayor similitud menor distancia y viceversa. Una forma de medir la "cercanía" entre dos instancias es contando el número de rasgos diferentes que hay entre éstas; instancias idénticas tendrán distancia cero. Esta formulación es un buen punto de partida, para definir otra métrica que enfrente situaciones comunes cuando comparamos instancias; como cuando hay atributos más importantes que otros, y, por tanto, la presencia o ausencia de los más importantes es más significativo que la ausencia o presencia de los menos importantes. La importancia de un atributo puede ser expresado por un *peso* del atributo. Este peso puede ser proporcionado por la ganancia de información del atributo, pues nos indica qué tanta información provee tal atributo. Resumiendo, dada una colección de entrenamiento S, una función que determina la distancia entre dos ejemplares X y Y, $\Delta(X,Y)$, y un nuevo ejemplar por clasificar, Z, asignamos a Z la clase de $\mathrm{argmin}_{X\in S}\Delta(X,Z)$.

Es importante señalar que el razonamiento basado en memoria hace uso intenso de los datos almacenados en la memoria sin que éstos sufran ninguna abstracción. Otra característica importante del razonamiento basado en memoria es que no depende[4] de modelos apoyados en el dominio [Stanfill, Waltz, 86].

5.1 Similitud entre instancias

Consideremos una colección de atributos $\mathcal{A} = (A_1, \ldots, A_m)$. Cada atributo $A \in \mathcal{A}$ tiene su dominio finito propio $\mathrm{Dom}(A)$. Sea $S \subset U$ una colección de entrenamiento y sean $C_1, \ldots, C_n \subset U$ las clases en el *universo* $U \subset \prod_{A\in\mathcal{A}} \mathrm{Dom}(A)$. Sean X y Y dos instancias representadas por la secuencia de valores de atributos $(x_1, \ldots, x_m)$ y $(y_1, \ldots, y_m)$. La distancia Δ es definida como

$$\Delta : (X,Y) \mapsto \Delta(X,Y) = \sum_{i=1}^{m} \delta(x_i, y_i) \quad \text{donde} \quad \delta(a,b) = \begin{cases} 0 & \text{si } a = b \\ 1 & \text{si } a \neq b \end{cases}$$

Para clasificar una nueva instancia, Z, es necesario obtener la clase de la instancia $\mathrm{argmin}_{X\in S}\Delta(X,Z)$.

Frecuentemente Δ, aunque es muy simple, no es la distancia más adecuada. Como alternativa, podemos asociar *pesos* a cada uno de los atributos de tal manera que, en el peor de los casos –cuando los rasgos son diferentes– sumamos pesos (y aumenta la distancia) y, en el mejor de los casos –los valores de los atributos son iguales–, no haya contribución a la distancia. La distancia se modifica así:

$$\Delta(X,Y) = \sum_{i=1}^{m} \delta(x_i, y_i) \cdot w_i. \tag{7}$$

Una técnica que ha tenido buenos resultados [Daelemans, van-den-Bosch, et. al. 98] asocia a cada atributo su *ganancia normalizada*. Ello se basa en el cálculo de la cantidad de información de los atributos, de manera semejante a la determinación de un árbol de decisión [Quinlan, 93] que discrimina atributos en los caminos desde la raíz. La ganancia normalizada se define con base en la ganancia de información:

Ganancia de información Para cualquier conjunto $S' \subset S \subset U$, usamos la cantidad de información necesaria para identificar una instancia de S' en una clase: $H_{S'}$, de acuerdo con la ecuación 1. Entonces, para cada $A \in \mathcal{A}$, podemos calcular la cantidad de información de S' respecto de A: $H_{S'(A)}$ (ec. 2). El valor $\mathit{gain}(A) = H_{S'} - H_{S'(A)}$ es la *ganancia de información* cuando S' es repartido de acuerdo con el atributo A.

Ganancia normalizada Para una colección de entrenamiento dada, S, y para cada $A \in \mathcal{A}$ sea

$$\mathit{split\ info}(A) = -\sum_{a\in\mathrm{Dom}(A)} \frac{\#S|_a}{\#S} \cdot \log_2\left(\frac{\#S|_a}{\#S}\right),$$

la ganancia normalizada se define por

$$\mathit{gain\ ratio}(A) = \frac{\mathit{gain}(A)}{\mathit{split\ info}(A)}. \tag{8}$$

[4]Esta dependencia se refiere a los cambios sustanciales que sufre un método al cambiar el dominio de aplicación.

La normalización en 8 pondera la ganancia de información con las entropías de los atributos. Los valores que calcula 8 serán utilizados como pesos en 7 para llevar a cabo nuestra aplicación.

El procedimiento del razonamiento basado en memoria, en general, consiste de reunir una colección de instancias de entrenamiento y una vez identificada la entidad a clasificar, esto es, habiendo determinado las clases posibles de la entidad, sus propiedades y los rasgos que permiten determinar su clase, tendrá que calcularse la ganancia normalizada de cada atributo, eligiendo los atributos con mayor información para definir los pesos que intervienen en la función de distancia. Como se ha dicho, para clasificar una nueva instancia z se calcula la distancia hacia cada elemento $x \in S$ en la colección de entrenamiento S, y la clase del elemento que resulte con menor distancia es elegida para clasificar a la nueva instancia z. Para las entidades lingüísticas, las instancias que formarán la colección de entrenamiento se extraen de una fuente rica de ejemplos; normalmente de un *corpus*. Los siguientes pasos, orientados a la clasificación de entidades lingüísticas, precisan la obtención de una colección de entrenamiento:

Colección de entrenamiento inicial:

- Proponer todos los atributos posibles de la entidad a clasificar y determinar las posibles clases de ésta.
- Reunir las instancias de entrenamiento compuestas por atributos contextuales y su clase. Los atributos contextuales se forman por tantos vecinos alrededor de la entidad como sean necesarios y, si se considera necesario, propiedades de los vecinos. En consecuencia, los valores de los atributos pueden ser palabras que están alrededor de la entidad a clasificar o propiedades de las palabras, estructuras gramaticales y otras entidades lingüísticas del contexto en que ocurre la entidad por clasificar. Siendo ésta una de las tareas básicas para conformar una colección de entrenamiento, ha de ser realizada por un experto (naturalmente humano) en el área. Esto obliga a que la determinación de la clase de cada instancia se haga manualmente (en contraposición a la etapa de aplicación: cuando se induce la clasificación a un texto de aplicación, partiendo del corpus de entrenamiento, se procede, en efecto, de manera algorítmica, es decir, automática).

Selección de atributos:

- Calcular y analizar la curva de ganancia normalizada para todos los atributos, con lo cual será posible discriminar los atributos que no aportan información para determinar la clase de la entidad.
- Proyectar la colección inicial de acuerdo con los atributos que ofrecen mayor información en una nueva colección de entrenamiento. Específicamente:
  - Una nueva instancia se forma extrayendo, de una instancia original, los rasgos con mayor peso informativo. En caso de formar dos instancias iguales pero con diferente clase decidir la clase de esa instancia tomando aquélla cuya frecuencia de ocurrencia sea mayor, procediendo de acuerdo con un criterio "voraz" (*greedy*): Se da una mayor oportunidad de asignar clases correctas a instancias ambiguas. Careciendo de un criterio determinista, se ha convenido en la aceptación de este criterio voraz basado en frecuencias.

6 Elección de un método

Como será visto en esta sección, el método de aprendizaje basado en memoria es el que mejor se adapta a nuestras necesidades. Con base en la precisión que tenga cada método presentado, al clasificar palabras de un texto, sustentaremos nuestra elección.

Se plantea, como problema a resolver, la clasificación de las palabras ambiguas frecuentes[5] ("la", "que",

[5]Una palabra ambigua es aquella para la que existen al menos dos contextos en los cuales tiene diferente función (por ejemplo: "la mesa" con función artículo, y "la construyó" con función pronombre). Hay muchas palabras ambiguas, aquí nos referimos a las que son muy frecuentes en el español mexicano: la, que, todo, uno, sobre, para, los, como, y algunas flexiones de éstas. Las funciones de la palabras o partes del discurso consideradas son: `ADJ`, `ADV`, `ART`, `CONJ`, `CONT`, `NOM`, `NP`, `NUM`, `PREP`, `PRON`, `VERB`, y los signos de puntuación, para adjetivo, adverbio, artículo, conjunción, contracción, sustantivo, nombre propio, número, preposición, pronombre, y verbo, respectivamente.

etc.). Específicamente, el clasificador tendrá que discernir sobre la parte del discurso que le corresponde a cada una de estas palabras al ocurrir en un texto, es decir, elegir entre pronombre y artículo la clase de "la", entre conjunción y pronombre la clase de "que", etc. Para ello se tomó un texto compuesto de 7,661 signos, cada uno con su parte del discurso. De éste se extrajeron instancias para constituir la colección de entrenamiento. Las instancias se formaron por las partes del discurso de cuatro palabras vecinas a cada palabra ambigua, dos palabras contiguas a cada lado de ella. A partir de estas instancias, 654 en total, se procedió a formar: la colección de entrenamiento, para el método de aprendizaje basado en memoria; las parejas (clase, rasgo), según el método de atributos relevantes; y el árbol de decisión definido por el criterio de Quinlan.

Hemos tomado para ejemplificar los recursos antes aludidos, lo correspondiente para el etiquetamiento de la palabra "que". En este caso, las parejas que emplea el método de rasgos relevantes se muestran en la tabla 1. La tabla está organizada en cuatro columnas, una por cada rasgo. Cada entrada de la tabla contiene una terna formada por una pareja y el valor que le corresponde, de acuerdo con la curva normal, para proponer que la clase de la pareja depende del rasgo de la pareja. Todos los valores d_{ik}^{pq} son mayores que 1.96, pues hemos establecido 95% de confianza (conforme a lo dicho en la sec. 2.2.1). El primer renglón, correspondiente al rasgo 1, lo interpretamos como: la clase de "que" es `PRON` si su rasgo 1 es `ART`, esto es, si la marca de la $(j-2)$-ésima palabra es `ART`, si "que" ocupa la posición j-ésima, pues $d_{1k}^{pq} > 1.96$[6]. De manera semejante se interpretan las demás entradas.

Clase, Rasgo $1/d_{1k}^{pq}$	Clase, Rasgo $2/d_{2k}^{pq}$	Clase, Rasgo $3/d_{3k}^{pq}$	Clase, Rasgo $4/d_{4k}^{pq}$
PRON, ART /5.95	PRON, ART /7.30	PRON, VERB /6.22	PRON, VERB /5.16
PRON, CONT /2.49	PRON, COM /2.32	PRON, ADV /2.78	PRON, COM /2.49
PRON, PREP /3.54	PRON, NOM /8.35	CONJ, ADJ /4.36	CONJ, NOM /3.23
CONJ, ADV /3.78	PRON, PTO /2.02	CONJ, NP /2.49	
CONJ, COM /3.13	PRON, NP /2.02	CONJ, ART /5.21	
CONJ, CONJ /2.78	CONJ, PREP /3.70	CONJ, CONJ /2.49	
CONJ, PRON /3.49	CONJ, VERB /10.49		
CONJ, PTO /2.02	CONJ, ADV /4.64		
	CONJ, CONJ /2.62		
	CONJ, DPT /2.02		

Tabla 1: Parejas clase-rasgo para determinar la parte de discurso de "que"

Por el método de Quinlan, procediendo según lo explicado en la sección 2.3.1, se determinaron 90 reglas para desambiguar la palabra "que". Algunas de éstas se muestran a continuación:

[6] Recordemos que este valor se usa cuando hay empate en la votación hecha por los rasgos.

ADJ NOM VERB X4 → PRON	ADJ PREP X3 NOM → CONJ	ADV NOM X3 X4 → CONJ
ADV PREP X3 VERB → CONJ	ART NOM X3 X4 → PRON	COM PREP X3 NOM → CONJ
CONJ NOM PRON X4 → PRON	CONT NOM X3 X4 → PRON	NOM NOM X3 X4 → PRON
PREP NOM PREP NOM → PRON	PREP NOM VERB ART → PRON	PREP NOM VERB NOM → CONJ
PRON GUI X3 X4 → PRON	PRON PREP X3 VERB → CONJ	VERB ADV X3 ADV → PRON
X1 ADJ ADV X4 → PRON	X1 ADJ ART X4 → CONJ	X1 ADJ COM X4 → CONJ
X1 ADJ PRON X4 → CONJ	X1 ADJ VERB VERB → PRON	X1 ADV X3 ADJ → CONJ
X1 ADV X3 VERB → CONJ	X1 ART X3 X4 → PRON	X1 COM ADJ X4 → CONJ
X1 COM PREP X4 → PRON	X1 CONJ X3 ART → CONJ	X1 CONJ X3 VERB → CONJ
X1 CONT X3 X4 → PRON	X1 NP X3 X4 → PRON	X1 PREP X3 ADJ → PRON
X1 PREP X3 PRON → PRON	X1 PRON ADV X4 → PRON	X1 PRON VERB X4 → PRON
X1 PTO X3 X4 → PRON	X1 VERB X3 X4 → CONJ	

Figura 3: Parte de las reglas usadas en la desambiguación de “que”.

Las reglas se interpretan de la manera siguiente: La palabra ambigua se etiqueta con la marca que indica el consecuente, si se cumple el antecedente. El antecedente de una regla está formado como lo hemos estado diciendo, por cuatro marcas. Para nuestro ejemplo: las marcas de dos palabras antes de “que”, y las marcas de dos palabras después de “que”. Así, la regla `ADJ NOM VERB X4` → `PRON` que ocurre en la primera entrada de la tabla 3, significa que la marca de “que” será `PRON`, si se cumple que: la palabra antes de “que” tiene la marca `NOM`, la marca antes de `NOM` es `ADJ`, la marca de la palabra que sigue a “que” es `VERB`, y no importa cual sea la marca posterior a `VERB` (`X4`).

El método de aprendizaje basado en memoria no modifica las instancias, trabaja directamente con ellas, según lo expuesto en la sección 2.4.1. Solamente fue calculada la ganancia de información para cada uno de los atributos. De un total de 189 instancias útiles en la asignación de la parte de discurso a la palabra “que”, se muestran sólo algunas de éstas en la fig. 2. Según el orden, de la marca más alejada a la izquierda hacia la marca más alejada a la derecha, los valores de ganancia de información (véase la ec. 8) son: 0.0963056, 0.161837, 0.360231 y 0.103724. Estos valores indican que la marca de la palabra que sigue a “que” es la de mayor peso informativo, en tanto que la marca que le antecede en dos posiciones es la de menor peso informativo. Se ha dicho que estos valores serán utilizados en la ec. 7 como pesos (w_i). Más adelante, con estos valores, calcularemos la distancia entre dos secuencias de rasgos.

Instancia	Clase
ART NOM PRON VERB	PRON
PTO CONJ ADJ NOM	CONJ
COM ART ADV VERB	PRON
VERB VERB ART NOM	CONJ
VERB ART ART NOM	PRON
ART NOM ADV VERB	PRON
VERB VERB ART NOM	CONJ
NOM PREP PREP PRON	PRON
VERB ART PRON VERB	PRON
ADJ COM PREP NOM	PRON
PRON VERB VERB PREP	CONJ
VERB ART PRON VERB	PRON
NOM ADJ ADV PRON	PRON

Tabla 2: Instancias formadas por dos marcas vecinas a cada lado de la palabra “que”.

Con las reglas, parejas e instancias (que hemos ejemplificado sólo para el caso de la palabra “que”) se llevó a cabo el experimento de aplicar los tres métodos expuestos a un texto de prueba[7] formado por 400 palabras. Ya que eran necesarias las partes del discurso de las palabras que son vecinas a la palabra ambigua frecuente por clasificar, se tomó el texto ya marcado y se reclasificaron las palabras ambiguas frecuentes de éste.

El texto a marcar se muestra en la fig 4. En esta figura, la primera columna representa la posición (indicada con el signo #) de la primera palabra, que está en la siguiente columna de ese mismo renglón. El texto de la columna dos es una secuencia de 15 palabras o signos, con la secuencia de marcas adjudicadas en el siguiente renglón. Las marcas que ocurren en esta secuencia se consideran con las mismas posiciones que las palabras del texto del renglón previo.

[7] Éste es un fragmento del artículo periodístico del género editorial, titulado "El congreso y la economía", de Miguel Angel Granados Chapa, tomado del *Corpus del Español Mexicano Contemporáneo* [Lara, Ham-Chande, et. al. 79].

#	Texto/Marcas
1	. lo hizo con ramplonería agravada por la insolencia que se deriva de hablar desde PTO lo VERB PREP NOM VERB PREP la NOM que PRON VERB PREP VERB PREP
16	el poder , en nombre de la mayoría . mayoría que esta vez , contra ART NOM COM PREP NOM PREP la NOM PTO NOM que ADJ NOM COM PREP
31	su costumbre , no fue silenciosa , si bien sólo alcanzó a romper su habitual ADJ NOM COM ADV VERB NOM COM CONJ ADV ADV VERB PREP VERB ADJ ADJ
46	y obligado mutismo golpeando la madera de los escaños . pero todo eso es lo CONJ ADJ NOM ADV la NOM PREP los NOM PTO CONJ todo PRON $\boxed{\text{VERB lo}}$
61	que menos importa . más relevante debió ser la toma de posición del congreso ante $\boxed{\text{que ADV VERB}}$ PTO ADV ADJ VERB VERB la NOM PREP NOM CONT NOM PREP
76	la carestía . formar un comité de emergencia supone que la situación es de emergencia la NOM PTO VERB ART NOM PREP NOM VERB que la NOM VERB PREP NOM
91	. si así es , la apreciación parlamentaria contradice el informe hecho público , el PTO CONJ ADV VERB COM la NOM ADJ VERB ART NOM VERB NOM COM ART
106	mismo día , por el banco de México . según este organismo , el índice ADJ NOM COM PREP ART NOM PREP NP PTO PREP ADJ NOM COM ART NOM
121	de precios sólo ha subido cinco por ciento de agosto a la fecha . ¿ PREP NOM ADV VERB VERB ADJ PREP NOM PREP NOM PREP la NOM PTO AIN
136	justifica esta alza el que el congreso se sienta obligado a estudiar la situación ? VERB ADJ $\boxed{\text{NOM ART que ART NOM}}$ PRON VERB VERB PREP VERB la NOM INT
151	¿ o los diputados y senadores , a pesar de todo , han podido encontrar AIN CONJ los NOM CONJ NOM COM PREP NOM PREP todo COM VERB VERB VERB
166	en la realidad datos que no corresponden a los de la estadística ? de cualquier PREP la NOM NOM que ADV VERB PREP los PREP la NOM INT PREP ADJ
181	modo , el caso es que el comité sólo tendrá una importancia retórica . nace NOM COM ART NOM VERB que ART NOM ADV VERB una NOM ADJ PTO VERB
196	sin que para ello exista fundamento legal alguno . y eso es grave , en PREP que PREP PRON VERB NOM ADJ ADJ PTO CONJ PRON VERB NOM COM PREP
211	un sistema constitucional cimentado en facultades expresas . es decir , el poder público sólo ART NOM ADJ VERB PREP NOM ADJ PTO VERB VERB COM ART NOM ADJ ADV
226	puede hacer válidamente lo que le está ordenado o permitido . de otra manera , VERB VERB $\boxed{\text{ADV lo que PRON VERB}}$ ADJ CONJ ADJ PTO PREP ADJ NOM COM
241	a diferencia del norteamericano , que en mucho le ha servido de modelo , el PREP NOM CONT NOM COM que PREP ADV PRON VERB VERB PREP NOM COM ART
256	congreso mexicano carece de facultades indagatorias , jurisdiccionales , salvo en pocas materias , ninguna NOM ADJ VERB PREP NOM ADJ COM ADJ COM ADV PREP ADJ NOM COM PRON
271	de las cuales tiene que ver con los precios . centralmente , ha de ocuparse PREP las PRON VERB que VERB PREP los NOM PTO ADV COM VERB PREP VERB
286	de legislar y de supervisar algunas actividades del poder ejecutivo . en ello debiera enfocar PREP VERB CONJ PREP VERB ADJ NOM CONT NOM ADJ PTO PREP PRON VERB VERB
301	su esfuerzo el congreso . primero , puede ejercer facultades como la que le atribuye ADJ NOM ART NOM PTO ADV COM VERB VERB NOM ADV la que PRON VERB
316	el artículo 93 de la constitución . una interpretación quizá extensiva - pero autorizada por ART NOM NUM PREP la NOM PTO una NOM ADV ADJ GUI CONJ VERB PREP
331	los hechos , pues el congreso ha llamado ante él , así sea de modo los NOM COM CONJ ART NOM VERB VERB PREP PRON COM ADV VERB PREP NOM
346	informal , aun a jefes de departamento - le permitiría interrogar al procurador del distrito NOM COM ADV PREP NOM PREP NOM GUI PRON VERB VERB CONT NOM CONT NOM
361	y territorios federales sobre el grado de aplicación del código penal en materia de delitos CONJ NOM ADJ PREP ART NOM PREP NOM CONT NOM ADJ PREP NOM PREP NOM
376	contra la economía pública . se sabría allí , con certidumbre , que jamás se PREP la NOM ADJ PTO PRON VERB ADV COM PREP $\boxed{\text{NOM COM que ADV PRON}}$
391	procesa a nadie por acaparamiento , tal como podría hacerse VERB PREP PRON PREP NOM COM PRON ADV VERB VERB

Figura 4: Texto a marcar con sus partes del discurso.

Se describe, a continuación, algunos etiquetamientos para "que", con cada uno de los métodos. En los ejemplos, hacemos referencia a los métodos: Quinlan, con Q; atributos relevantes, con A; y aprendizaje basado en memoria, con M. Al inicio de cada ejemplo, aparece la posición en la que ocurre "que" y sus marcas vecinas, contexto. En la fig 4 los ejemplos que explicaremos aparecen señalados con un recuadro.

posición 61, contexto `[VERB ART ADV VERB]`:

- **A** Se clasificó como `PRON`. Tres rasgos votaron por esta clase, según las parejas de la fig 1: el 2, 3 y 4.
- **M** La misma clase se determinó por similitud de instancias. Se empleó: `[COM ART ADV VERB]`, cuya distancia al contexto de "que" en la posición 61 fue: 0.0963056.
- **Q** La regla `X1 ART X3 X4` → `PRON` empató con el contexto y asignó, también, la misma marca.

posición 140, contexto `[NOM ART ART NOM]`:

- **A** La clase determinada por dos rasgos contra uno fue `CONJ`. A favor 3 y 4, en contra el 2.
- **Q** El método basado en árbol de decisión, aplicó la misma regla que en la posición 61, clasificándola como `PRON`.
- **M** La instancia `[VERB ART ART NOM]` cuya clase es `PRON`, fue la más cercana.

posición 230, contexto `[ADV ART PRON VERB]`:

- **A** Tenemos que los rasgos relevantes que determinaron esta clase fueron dos entre el 1, 2 y 4. La clase propuesta fue `PRON`.
- **M** La instancia `[VERB ART PRON VERB]` con clase `PRON` se acercó a una distancia 0.0963056.
- **Q** Para este contexto no hubo empate con ninguna regla de "que".

posición 388, contexto `[NOM COM ADV PRON]`:

- **A** Los rasgos 2 y 3 propusieron la clase: `PRON`.
- **M** Con la instancia `[NOM ADJ ADV PRON]` se clasificó `PRON`.
- **Q** Tampoco hubo regla que pudiera ser aplicada a este contexto.

En total, en el texto de la fig. 4 hay 13 ocurrencias de la palabra "que". El método de Quinlan tuvo 8 aciertos, el método de atributos relevantes 9 aciertos, y el método de aprendizaje basado en memoria, 10 aciertos.

Los aciertos, en general (considerando todas las palabras ambiguas frecuentes), se muestran en la siguiente tabla 3.

Método	Aciertos (%)
Síntesis de atributos	75.0
Rasgos relevantes	90.0
Similitud entre instancias	92.5

Tabla 3: Precisión de los métodos al etiquetar palabras ambiguas frecuentes.

Para que el método de Quinlan pueda clasificar correctamente un ejemplar, deberá contar con una regla sin variables en su antecedente que empate completamente con el contexto de la palabra que va a ser marcada, o bien, deberá alimentarse con suficientes ejemplos para empatar solamente con algunos rasgos contenidos en el contexto de la palabra a etiquetar, y los demás rasgos tuvo que haberlos generalizado el método. En otras palabras, para este método, las fallas en la clasificación de nuevos ejemplos, se pueden interpretar como la necesidad de más ejemplos para el método, de tal forma que realice una adecuada generalización. En el texto de la fig 4 hubo cinco casos en los que no pudo ser aplicada una regla (dos de ellos ocurren en las posiciones ejemplificadas: 230 y 388.).

En contraste con el método anterior, el método de identificación de atributos relevantes reduce la clasificación de una instancia, a la existencia de los atributos más relevantes en la instancia por clasificar. El no comprometer la ocurrencia de rasgos no relevantes de una instancia para concluir su pertenencia a una clase, permite incrementar la flexibilidad para clasificar diversas instancias.

La mayor ventaja del método basado en memoria es que conserva, en las instancias de entrenamiento, las regularidades y las excepciones. La flexibilidad de comparar instancias que no sean exactamente iguales, en todos sus rasgos, ofrece también una ventaja en situaciones que ocurren con frecuencia en el procesamiento del lenguaje natural [Daelemans, van-den-Bosch, et. al. 98]. Este tipo de empatamiento lo tiene, en cierta medida, el método de síntesis, pero solamente en los atributos que se generalizaron.

Aunque la búsqueda exhaustiva del vecino más cercano en el método basado en memoria es costosa, respecto al uso de reglas, el trabajo puede aligerarse representando la colección de entrenamiento en un árbol *trie*, mismo que permite efectuar podas en el cálculo de la distancia. Esta implantación del método basado en memoria se conoce como *IGTree* [Bosch, Daelemans, et. al. 96].

Con la elección hecha en esta sección estamos en posibilidades de llevar a cabo el desarrollo de otros sistemas [Daelemans, van-den-Bosch, et. al. 98] para el PLN, como el análisis sintáctico parcial.

Pero más aún, la aplicación de los métodos de aprendizaje automático abarcan un espectro amplio de problemas en el campo del PLN, ya que la clasificación es una forma de desambiguación.

Referencias

[Chan, Wong, 91] Chan, K. & Wong, A.: A statistical technique for extracting classificatory knowledge from databases, Knowledge discovery in databases, Patietsky-Shapiro, G. & Frawley, W. (Eds.), *American Association of Artificial Intelligence*, AAAI Press, pp 107-24, 1991.

[Daelemans, van-den-Bosch, et. al. 98] Daelemans, Walter; van-den-Bosch, Antal; Zavrel, Jakub; Veenstra, Jorn; Buchholz, Sabine & Busser, Bertjan: Rapid development of NLP modules with memory-based learning, *Proc. of ELSNET in Wonderland*, pp 105-113, 1998.

[Lara, Ham-Chande, et. al. 79] Lara, Luis Fernando; Ham-Chande, Roberto & García-Hidalgo, Ma. Isabel: *Investigaciones lingüísticas en lexicografía*, Jornadas 89, El Colegio de México, 1979.

[Oflazer, Tür, 97] Oflazer, Kemal & Tür, Gökhan: Morphological disambiguation by voting constraints, http://lib-www.lanl.gov/cmp-lg/9704011, 1997.

[Quinlan, 93] Quinlan, Ross: *C4.5: programs for machine learning*, Morgan Kaufmann, California, 1993.

[Stanfill, Waltz, 86] Stanfill, C. & Waltz, D.: Towards memory-based reasoning. In *Communications of the Association for Computational Machinery*, V.29 (12), pp 1213-1228, 1986.

[Bosch, Daelemans, et. al. 96] van-den Bosch, Antal; Daelemans, Walter; Weijters, Ton: Morphological analysis as classification: an inductive-learning approach, http://lib-www.lanl.gov/cmp-lg/9607021, 1996.