Programación Matemática y Software (2010)
Vol.2. Num. 1. ISSN: 2007-3283

Recibido: 31 de Agosto del 2009 Aceptado: 28 de Diciembre del 2009
Publicado en línea: 30 de Junio del 2010

Proposición de un modelo para la acentuación automática de palabras ambiguas del español, utilizando etiquetado de texto

Raymundo Montiel,* Blanca E. Pedroza,* María Guadalupe Medina,* Carlos Pérez*

*Instituto Tecnológico de Apizaco
División de Estudios de Posgrado e Investigación
Av. Instituto Tecnológico s.n., Apizaco, Tlaxcala, C.P. 90300, México.

mlirary@hotmail.com, thelismedina@hotmail.com

Resumen. La acentuación de palabras cuando se escribe un texto en español es un problema de ambigüedad, debido a que muchas palabras llevan acento o no dependiendo del contexto de la frase. El problema de la ambigüedad está relacionado con la asignación de etiquetas o categorías gramaticales a las palabras dentro de una frase, es decir, cuando se indica si se trata de un verbo, un sustantivo, etcétera. En el presente artículo se propone un modelo que ayuda a determinar en forma automática si una palabra con acento diacrítico debe llevar o no acento ortográfico, con base en la asignación de etiquetas y mediante la aplicación de métodos híbridos —algoritmos supervisado y no supervisado. Posteriormente, el método se aplica en el diseño de un programa de cómputo cuya función es de apoyo en la enseñanza de las reglas de acentuación y con el cual se realiza la acentuación automática de palabras ambiguas. Este método podría ser una herramienta en un procesador de palabras.

Palabras clave: desambiguación del sentido de la palabra (DSA)

Abstract. The process of accentuating words when writing a text in Spanish, is a problem of ambiguity, because there are many words that can be accentuated or not depending on the context of the sentence. The problem of ambiguity of words is related to the process of assigning tags or grammatical categories to the words in the sentence, that is, indicating wheter a Word is a verb , noun, etc. In this work, we propose a model, which helps to automatically determine wheter a word with diacritic accent should carry or not orthografic accent based on the tagging by means of application of hibryd methods (supervised and unsupervised algorithms). After this method is applied to design a software that will support in teaching the rules of acentuation and for automatic acentuation of ambiguous words; which can be implemented as a tool at a word processor.

Keywords: word sense disambiguation, accent restoration, part of speech tagger.

1 Introducción

En computación, una de las tareas más difíciles —y que ha suscitado mucho interés en el ámbito del procesamiento del lenguaje natural (PLN)— se produce cuando una palabra tiene varios sentidos o significados. Este fenómeno lingüístico se conoce como “polisemia” y a él se le asocia lo que en el procesamiento del lenguaje natural se conoce como desambiguación del sentido de la palabra (DSA), tarea que consiste en identificar el sentido correcto de una palabra en un contexto.[4] La polisemia está muy relacionada con el problema de la asignación de categorías gramaticales, el cual consiste en decir si una palabra es un verbo, un artículo, un adjetivo o un sustantivo,[3] dependiendo del significado que le corresponde en el contexto de la oración.

El proceso de acentuar palabras es también un problema de ambigüedad, debido a que en el español existen muchas palabras que pueden estar acentuadas o no, dependiendo de su contexto, del tiempo de la acción en la oración, etcétera. El acento diacrítico distingue palabras formalmente idénticas, es decir, escritas con las mismas letras, pero que pertenecen a categorías gramaticales diferentes.

Aunque los procesadores de texto de las grandes compañías de *software* poseen funciones muy sofisticadas —como, por ejemplo, un diccionario para corregir la ortografía—, tienen un límite.

Estos correctores no comprenden el significado de las palabras, sino que se limitan a ir comparando cada palabra del texto con las existentes en su diccionario, de modo que si encuentra la palabra en cuestión la dará por válida si está escrita correctamente, en caso contrario la marca para indicar un error.[1]

Sin embargo, cuando aparecen palabras que pueden llevar acento ortográfico o no, el procesador no las marca como errores si el usuario no les pone acento, y tampoco le indica que puede estar mal escrita. Entonces, un usuario que no conozca bien el idioma se quedaría con la idea de que la palabra está escrita correctamente.

La falta de acentos en algunas palabras de las oraciones es un problema de ambigüedad. Las ambigüedades más comunes en la acentuación se dan entre las palabras con terminación “o”, como en “completo” y “completó”. Se trata de los tiempos presente y pretérito de los verbos que terminan en “ar”. Otras ambigüedades son semánticas, como sucede con algunos sustantivos; por ejemplo, “secretaria” —persona encargada de escribir la correspondencia— y “secretaría” —sección de un organismo, institución o empresa.[8]

En este artículo se describirá el modelo que se usó para diseñar una herramienta informática de desambiguación semántica para el idioma español, cuya utilidad es la acentuación correcta de las

palabras en los textos escritos, basándose en el etiquetado de las oraciones. Para llevar a cabo el modelo se aplicó un método híbrido, esto es, mediante un algoritmo supervisado y uno no supervisado.

2 Solución

El objetivo de este trabajo es crear un sistema que ayude a determinar si una palabra con acento diacrítico debe o no llevar acento ortográfico, lo cual está determinado por el contexto en el que se esté ocupando, con ayuda de las etiquetas asignadas. El modelo general que se propone para la solución del problema se ilustra en la figura 1. La primera tarea del modelo es el análisis léxico, el cual consiste en la eliminación de los acentos de las palabras de la frase de entrada, ya que para el cálculo de los parámetros del modelo se requieren palabras sin acentos. Además, el analizador léxico identifica y separa los signos de puntuación de las palabras. Posteriormente, se establece si existen palabras ambiguas en la oración de entrada, y al mismo tiempo, el modelo indica la posición en la que éstas se encuentran. Esto se realiza comparando cada una de las palabras de las oraciones con las palabras cuyo acento es diacrítico, en el diccionario previamente construido.

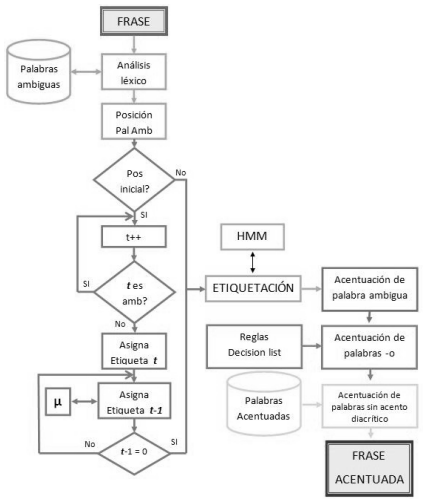


Figura 1. Arquitectura del modelo de solución.

En este método de solución se utilizan los modelos ocultos de Markov (HMM), los cuales se definen siguiendo la notación presentada por Rabiner,[9] como la una 5-tupla $\mu = (Q, V, \pi, A, B)$, donde:

1. Q es el conjunto de estados del modelo. Aunque los estados permanecen ocultos, para la mayoría de las aplicaciones prácticas se conocen *a priori*. En nuestro caso de etiquetado de palabras, cada etiqueta sería un estado. En general, los estados están conectados de tal modo que cualquiera de ellos se alcanza desde cualquier otro estado en un solo paso. Los estados se etiquetan como {1, 2,…, N}, y el estado actual en el instante de tiempo t se denota como q_t. Para el etiquetado de palabras no hablaremos de los instantes de tiempo, sino de las posiciones de cada palabra en la frase.

2. *V* es el conjunto de los distintos sucesos que se observan en cada uno de los estados. Cada uno de los símbolos individuales que un estado emite se denota como $\{v_1, v_2, \ldots, v_M\}$. En el etiquetado de palabras, *M* es el tamaño del diccionario y cada $v_k, 1 \leq k \leq M$ es una palabra distinta.
3. π = $\{\pi_i\}$ es la distribución de probabilidad del estado inicial. Por lo tanto,

$$\pi_i = P(q_1 = i), \qquad \pi_i \geq 0, \qquad 1 \leq i \leq N, \qquad \sum_{i=1}^{N} \pi_i = 1 \tag{1}$$

4. ***A*** = $\{a_{ij}\}$ es la distribución de probabilidad de las transiciones entre estados, es decir,

$$a_{ij} = P(q_t = j | q_{t-1} = i) = P(j|i), \qquad 1 \leq i, j \leq N, \qquad 1 \leq t \leq T, \qquad \sum_{j=1}^{N} a_{ij} = 1, \qquad \forall\ i. \tag{2}$$

5. ***B***= $\{b_j(v_k)\}$ es la distribución de probabilidades de los sucesos observables, es decir,

$$b_j(v_k) = P(o_t = v_k | q_t = j) = P(v_k|j), \qquad b_j(v_k) \geq 0, \qquad 1 \leq j \leq N, \qquad 1 \leq k \leq M, \qquad 1 \leq t \leq T \qquad \sum_{k=1}^{M} b_j(v_k) = 1, \qquad \forall i. \tag{3}$$

Éste es un método estocástico que necesita parámetros previamente definidos, los cuales se calculan mediante métodos de entrenamiento supervisado y no supervisado.

Para la fase del entrenamiento supervisado se utilizó un cuerpo de texto etiquetado conocido como CONLL, el cual es una colección de artículos noticiosos del año 2000 ofrecida por la Agencia de Noticias EFE. Los parámetros del modelo se estiman por "máxima verosimilitud",[5] a partir de las frecuencias relativas de aparición de los eventos del texto etiquetado (véase la figura 2).

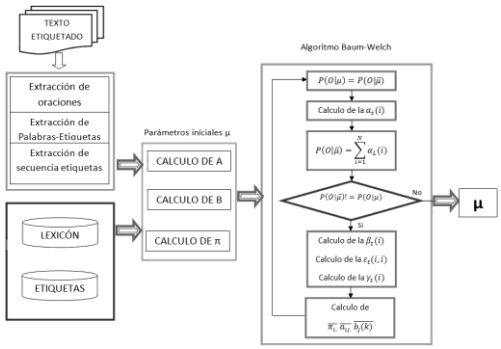


Figura 2. Entrenamiento supervisado y no supervisado.

Las probabilidades de transición a_{ij} se obtienen contando las veces que se transita del estado s_i al estado s_j y dividiendo el resultado por las veces que se transita por el estado s_i.

$$a_{ij} \approx \hat{P}(q_t = s_j | q_{t-1} = s_i) = \frac{f(q_{t-1} = s_i, q_t = s_j)}{f(q_{t-1} = s_i)} \tag{1}$$

Las probabilidades de emisión se obtienen contando las veces que un símbolo (v_k) ha sido emitido en un estado

(s_j) y dividiendo el resultado por las veces que se ha transitado por ese estado:

$$b_j(v_k) = P(o_t = s_i, |q_t = s_j) = \frac{f(o_{t-1} = v_k, q_t = s_j)}{f(q_t = s_j)} \quad (2)$$

Una vez que tenemos los parámetros iníciales de μ, se aplica el algoritmo Baum-Welch,[6] lo que incrementa la probabilidad de las transiciones entre los estados y sus símbolos, y se mejora en consecuencia la probabilidad de la secuencia de observaciones dada.

En el modelo de solución (véase la figura 1) se maneja una fase de etiquetado que utiliza el algoritmo de Viterbi, con algunas modificaciones, ya que no se consideran todos los estados posibles, es decir, todas las etiquetas del juego de etiquetas utilizado, sino sólo las etiquetas candidatas asignadas a cada una de las palabras ambiguas.

Para ejemplificar el funcionamiento del modelo, consideremos la oración “El jugo frío está sobre la mesa”, en la cual existen varias palabras ambiguas. Además, como existen palabras ambiguas al principio de la oración, se recorre *t* palabras hasta encontrar una palabra no ambigua, en este caso, “frío”. Una vez encontrada la primera palabra no ambigua, se le asigna la etiqueta más probable. Después de esto, se asigna la etiqueta más probable a *t-1* a partir de la etiqueta (*t*), ya conocida (véase la figura 3), hasta llegar a t=1.

Una vez obtenidas las primeras etiquetas de la oración, el cálculo para obtener las etiquetas de las palabras ambiguas se realiza sólo sobre las etiquetas asignadas manualmente a estas palabras. En el cuadro 1 se muestra la oración junto con las etiquetas asignadas.

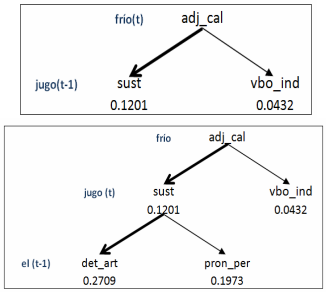


Figura 3. Asignación de etiquetas.

Cuadro 1. Resultado del etiquetado

El	**jugo**	**frío**	**está**	**sobre**	**la**	**mesa**
det_art	sust	adj_cal	vbo_aux	prep	det_art	sust

Después de haber obtenido las etiquetas de cada una de las palabras de la oración, se determina si las palabras ambiguas deberán o no llevar acento, con base en nuestro diccionario de palabras ambiguas etiquetadas (véase el cuadro 2).

Cuadro 2. Ejemplo de palabras ambiguas con sus etiquetas

PALABRA AMBIGUA	ETIQUETA	PALABRA
EL	**det_art**	**el**
	pron_per	él
ESTÁ	det_dem	esta
	vbo_aux	**está**
JUGO	**sust**	**jugo**
	vbo_ind	jugó
	sust	sobre
SOBRE	**prep**	**sobre**
	vbo_sub	sobre
	vbo_ind	sobré

Por último, se hace una comparación de las palabras de la oración, con la etiqueta asignada, y se determina si la palabra debe o no llevar acento. El resultado de nuestro ejemplo se muestra en el cuadro 3.

Cuadro 3. Resultado de ejemplo

El	jugo	frío	está	sobre	la	mesa

Uno de los problemas que no se resolvió con la asignación de categorías gramaticales fue el de una palabra con acento diacrítico etiquetada con la misma categoría gramatical, como se muestra en el cuadro 4.

Cuadro 4. Palabras con terminación “o”

Palabra	Etiqueta
abrazo	sust
abrazo	**vbo_ind**
abrazó	**vbo_ind**

Las dos últimas palabras corresponden a la conjugación en modo indicativo del verbo “abrazar”; sin embargo, la primera corresponde al tiempo presente en primera persona y la segunda al tiempo pasado en tercera persona. Para solucionar este problema se creó un conjunto de reglas utilizando “listas de decisiones” similares a las utilizadas por Yarowsky.[2]

En el cuadro 5 se presenta un extracto de las reglas obtenidas, ordenadas de acuerdo con su aparición en el corpus de entrenamiento, en el cual también se indica el orden en que se han de ir verificando cada una de las reglas cuando se obtengan ejemplos nuevos para identificar el patrón que les corresponde.

Cuadro 5. Extracto del conjunto de reglas

%	Regla		Clasificación
19.63	**YO** -o	»	-o
19	**SE** -o	»	-ó
16.6	-o **DE**	»	-o
14.6	-o **det_art**	»	-ó
10.83	**sust** -o	»	-ó
10.6	-o **CON**	»	-o
9.09	**adv** -o	»	-o
8.33	-o **det_ind**	»	-ó

Además de acentuar las palabras ambiguas, se trata de acentuar las palabras que siempre llevan acento, es decir, aquellas palabras que tienen acento, pero no diacrítico. Para esto, se hizo una base de datos de palabras con acento.

3 Resultados

Para la evaluación del modelo, se reunió una colección de 63 artículos de distintos contextos —deportes, ciencia, salud, etcétera—, obtenidos del periódico digital *El Universal*. Los artículos fueron evaluados por el modelo a partir de frases completas, en el cuadro 6 se muestra un resumen de los resultado obtenidos.

Cuadro 6. Resumen de los resultados obtenidos

1	2	3	4	5	6	7	8	9
1	164	26	26	26	0	0	12	12
7	166	24	22	22	2	2	14	8
10	214	30	26	26	4	3	22	21
19	376	54	49	49	5	3	26	23
24	1180	215	191	170	22	19	125	101
47	322	69	67	66	2	2	26	23
48	384	87	79	78	8	7	23	20
53	898	187	164	162	23	21	78	74
58	114	78	68	63	10	8	34	30
63	165	34	28	26	6	6	21	21
RESUL	22883	4238	3769	3611	472	397	1721	1526

1) Número del artículo. 2) Palabras por artículo. 3) Con acento diacrítico. 4) Con acento diacrítico, sin acento. 5) Con acento diacrítico, sin acentuar. 6) Con acento diacrítico, con acento. 7) Con acento diacrítico, acentuadas. 8) Con acento no diacrítico. 9) Con acento no diacrítico, acentuadas.

Con estos datos nos damos cuenta de que:

$$palabras\ ambiguas = \frac{4238\ x\ 100}{22883} = 18.52\ \%$$

$$palabras\ acentuadas\ (acento\ no\ diacrítico) = \frac{1526\ x\ 100}{1721} = 88.66\ \%$$

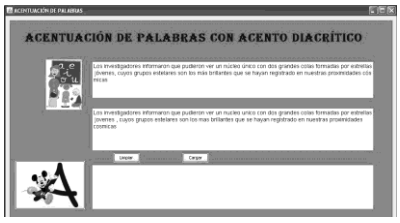
A continuación, los datos que aparecen en el cuadro 7 son los resultados obtenidos utilizando las métricas de efectividad de las tasas de recuperación (r), precisión (p), error (e), exactitud (a), y F_B. En la primera parte del cuadro 7 se muestran los resultados obtenidos a partir de las palabras acentuadas por nuestro modelo, donde la "precisión" —total de palabras acentuadas entre el número total de palabras con acento diacrítico— tiene como resultado 84.11%. Uno de los problemas que se han identificado está en el etiquetado de la palabra "el", pues siempre que el sistema encuentra esta palabra al comienzo de la frase, le asigna la etiqueta "pronombre personal" y la acentúa; sin embargo, también puede tomar la etiqueta de "artículo", en cuyo caso no lleva acento.

Cuadro 7. Resultados

Con base en palabras acentuadas		**Con base en palabras no acentuadas**
r = .7153	**e = 0.0549**	r = .9796
p = .8411	**a = .945**	p = .9580
f_{-B} = .7731		f_{-B} = .9687

Pero, como ya se ha mencionado, este modelo también determina cuándo una palabra con acento diacrítico no debe llevar acento ortográfico. En la segunda parte del cuadro 7 se muestran los resultados obtenidos a partir de las palabras que no deben llevar acento ortográfico. En este caso, la “precisión” —total de palabras que el modelo decide no acentuar, entre el número total de palabras con acento diacrítico— tiene como resultado 95.8%. Sin embargo, el resultado de “exactitud” —total de decisiones correctas tomadas por el modelo— es de 94.5%, con un error de 5.49%.

Este modelo se programó en Java y se está trabajando en la creación de un programa de cómputo que sirva como auxiliar en la enseñanza de las reglas de acentuación y para acentuar palabras ambiguas que por lo regular los editores de texto no acentúan. En la figura 4 se muestran algunas de las pantallas del sistema.



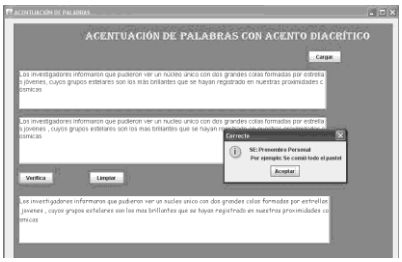


Figura 4. Pantallas del sistema para la acentuación de textos en español.

4 Conclusiones

El etiquetado de palabras es una técnica que ayuda a identificar la ambigüedad en la acentuación de palabras de un texto determinado y a producir herramientas computacionales que revisen y corrijan la acentuación correcta de textos en español, ya sea para hacer más fácil la escritura de los documentos o para utilizarlas en la enseñanza de las reglas gramaticales del español.

Aunque el problema de la ambigüedad en la acentuación de las palabras no es sencillo de resolver, aún se puede seguir analizando la gramática del idioma para encontrar patrones de comportamiento en las palabras, de acuerdo con las características del contexto en el cual se encuentren las oraciones. Además, las teorías y algoritmos ya existentes para el manejo de las gramáticas son factibles de aplicación, así como lo son las teorías del área de los sistemas inteligentes, como el aprendizaje automático. Por lo tanto,

todavía hay mucho que investigar en este tema, si se quieren producir sistemas de cómputo que ayuden al usuario a corregir sus errores de acentuación y otros errores ortográficos.

Se propone que en el futuro se trabaje en los siguiente:

1) El aumento de las palabras con acento diacrítico, las palabras con acento no diacrítico y en las categorías gramaticales para disminuir la ambigüedad en el etiquetado. 2) La creación de un módulo, dentro del modelo, para el entrenamiento de los casos no resueltos. 3) El diseño de la interfaz para la enseñanza de las reglas gramaticales, y en especial, del uso de las palabras con acento diacrítico. 4) La aplicación del algoritmo de desambiguación semántica, para mejorar resultados. 5) El análisis sintáctico previo para reducir ambigüedad en la frase. 6) La creación de un sistema global de apoyo para la enseñanza o reafirmación de las reglas ortográficas, particularmente, de la acentuación. 7) La identificación de un patrón de ambigüedad para los casos similares al de la palabra "el".

Referencias

1. Pascual, F., *Domine Microsoft© Office XP Profesional, versión 2002*, edición especial, Alfaomega, México, 2002.
2. Yarowsky, D., "Decision List for Lexical Ambiguity Resolution: Application to Accent Restoration in Spanish and French", en *Proceeding of the XXXII Annual Meeting of the Association for Computational Linguistics*, 1994, pp. 88-95.
3. Perea, J., *Etiquetado de textos y su aplicación a la traducción*, University of Granada (investigación inédita), 2005.
4. Stevenson, M. y Y. Wilks, "Combining Independent Knowledge Sources for Word Sense Disambiguation", en R. Mitkov (ed.), *Recent Advances in Natural Language Processing*, John Benjamins Publisher, 2000.
5. Dempster A., N. Laird *et al*., "Maximum Likelihood from Incomplete Data Via the EM Algorithm", *Journal of Royal Statistical Society* (Series B, Metodological), vol. 39, núm. 1, 1977, pp. 1-38.
6. Baum, L., "Statistical Inference for Probabilistic Functions Finite State Markov Chains", *Annual Mathematic Statistical*, vol. 37, 1966, pp. 1554-1563.
7. Yarowsky, D., "A Comparison of Corpus-Based Techniques for Restoring Accents in Spanish and French Text", *Proceedings of the II Annual Wordshop on Very Large Text Corpora*, Kyoto, 1994 (en prensa).
8. Real Academia Española, *Banco de datos (CREA). Corpus de referencia del español actual*, en Internet (http://www.rae.es), página consultada el 29 de abril de 2009.
9. Rabiner, L. R., "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", *IEEE*, vol. 77, núm. 2, 1989, pp. 257-286.

10. Wagacha, P., G. De Pauw *et al*., “A Grapheme-Based Approach for Accent Restoration in Gikuyu”, en *Proceedings of the V International Conference on Language Resources and Evaluation*, 2006, pp. 1937-1940.
11. Bobiceva, V., “O altã metodã de restabilire a semnelor diacritice”, en I. Pistol, D. Cristea y D. Tufiş (eds.), *Resurse Lingvistice şi Instrumente pentru Prelucrarea Limbii Române*, 2008, pp. 179-188.
12. De Pauw, G., P. W. Wagacha y De Schryver, G. M., “Automatic Diacritic Restoration for Resource-Scarce Language”, en V. Matousek y P. Mautner (eds.), *TSD 2007, LNAI 4629*, 2007, pp. 170-179.

Acerca de los autores.

M.C. Raymundo Montiel Lira. Obtuvo el grado de Licenciado en Informática en el año 2006 en el Instituto Tecnológico de Apizaco, Tlaxcala y el de Maestro en Sistemas Computacionales en el año 2009 en el mismo instituto, actualmente es docente e investigador en el Instituto Tecnológico Superior de San Martín Texmelucan, Puebla. Su área de investigación es Procesamiento de Lenguaje Natural.

M.C. Blanca Estela Pedroza Méndez. Estudió la licenciatura en Matemáticas Aplicadas en la Universidad Autónoma de Tlaxcala. Posteriormente se graduó como Maestro en Ciencias Computacionales en la Benemérita Universidad Autónoma de Puebla. Es profesora de tiempo completo de la División de Estudios de Posgrado e Investigación del Instituto Tecnológico de Apizaco. Actualmente se encuentra desarrollando investigaciones en el área de Procesamiento de Lenguaje Natural.

M. en C. María Guadalupe Medina Barrera. Estudió la Maestría en Ciencias en Ciencias Computacionales en el área de Sistemas Basados en Conocimiento, por el Centro Nacional de Investigación y Desarrollo Tecnológico (cenidet). Sus áreas de interés son: Visión Artificial, Procesamiento Digital de Imágenes, Reconomiento de Patrones, Graficación y Animación Digital. Actualmente es Jefa de la División de Estudios de Posgrado e Investigación del Instituto Tecnológico de Apizaco, donde también es docente, impartiendo cátedra a nivel Licenciatura y Posgrado desde el 2002, siendo directora de tesis y revisora de diversos proyectos de investigación y desarrollo tecnológico.

M.C. Carlos Pérez Corona. Profesor-Investigador en la facultad de Ciencias Básicas, Ingeniería y Tecnología de la Universidad Autónoma de Tlaxcala. Profesor de Tiempo Parcial de la División de Estudios de Posgrado e Investigación del Instituto Tecnológico de Apizaco. Estudió la Licenciatura en Informática en el Instituto Tecnológico de Apizaco (1992). Tiene una especialidad en

Simulación y Control de Procesos de Ingeniería Química, en la Facultad de Ciencias Básicas, Ingeniería y Tecnología de la Universidad Autónoma de Tlaxcala(1995) y una Maestría en Inteligencia Artificial, en conjunto LANIA-Universidad Veracruzana (1999). Sus áreas de Interés son: Redes Neuronales, Redes Bayesianas, Minería de Datos, Sistemas Distribuidos, Sistemas Multiagentes y Redes de Computadoras.

Autónoma de Puebla. Es profesora de tiempo completo de la División de Estudios de Posgrado e Investigación del Instituto Tecnológico de Apizaco. Actualmente se encuentra desarrollando investigaciones en el área de Procesamiento de Lenguaje Natural.