



UNIVERSIDAD AUTÓNOMA DE  
**CHIHUAHUA**

Clave: 08MSU0017H

**UNIDAD ACADÉMICA  
INGENIERÍA**

**PROGRAMA DE LA UNIDAD DE  
APRENDIZAJE:**

**PROCESAMIENTO DE  
LENGUAJE NATURAL**

<b>DES:</b>	Ingeniería
<b>Programa Educativo:</b>	Maestría en Ingeniería en Computación
<b>Tipo de materia (Obligatoria/Optativa):</b>	Optativa
<b>Clave de la materia:</b>	MICOP222
<b>Semestre:</b>	2, 3, 4
<b>Área en plan de estudios</b>	Inteligencia Computacional
<b>Créditos</b>	6
<b>Total de horas por semana:</b>	6
<i>Teoría: Presencial o Virtual</i>	4
<i>Laboratorio o Taller:</i>	0
<i>Prácticas:</i>	2
<i>Trabajo extra-clase:</i>	0
<b>Créditos Totales:</b>	6
<b>Total de horas semestre (x 16 sem):</b>	96
<b>Fecha de actualización:</b>	6 de mayo de 2021
<b>Prerrequisito (s):</b>	Ninguna

**Propósito del curso:**

El curso describe conceptos fundamentales y avanzados del procesamiento de lenguaje natural, así como también aporta al alumno las competencias necesarias para que pueda diseñar y desarrollar aplicaciones útiles como traducción de idiomas, reconocimiento de entidades nombradas, análisis de sentimientos y modelos de lenguaje entre otras.

COMPETENCIAS	DOMINIOS COGNITIVOS	RESULTADOS DE APRENDIZAJE
<p><b>Aplicación de Ciencias de la Computación: <i>Aplica con ética las ciencias computacionales para resolver problemas en los sectores: industrial, gubernamental, académico y social bajo esquemas de colaboración inter, multi y transdisciplinaria</i></b></p> <p><b>1. Soluciona problemas en diversas áreas del conocimiento aplicando las ciencias computacionales</b></p>	<p><b>1. Representaciones textuales y espacios vectoriales.</b></p> <p>1.1. Representaciones dispersas.</p> <p>1.1.1. Extracción de características.</p> <p>1.1.2. Preprocesamiento</p> <p>1.1.3. Sentiment Analysis: implementación de un ejemplo usando Logistic regression.</p> <p>1.2. Probabilidades en Natural Language Processing (NLP)</p> <p>1.2.1. Probabilidades condicionales (Regla de Bayes)</p> <p>1.2.2. Naïve Bayes para Sentiment Analysis.</p> <p>1.2.3. Laplacian Smoothing para evitar probabilidades 0</p> <p>1.2.4. Función Log-likelihood para evitar valores pequeños de probabilidades (underflow)</p> <p>1.2.5. Ejemplos de aplicaciones: ataques adversarios, detección de sarcasmo, ironía y eufemismos.</p> <p>1.3. Modelos espacio-vectoriales</p>	<p>Aplica los fundamentos de ingeniería y de ingeniería especializada en la identificación, formulación, análisis y resolución de problemas complejos con el fin de alcanzar conclusiones fundamentadas.</p> <p>Realiza investigaciones de problemas complejos por métodos que incluyen experimentos apropiados, análisis e interpretación de datos y síntesis de la información con el fin de llegar a conclusiones válidas.</p>

	<ul style="list-style-type: none"> <li>1.3.1. Representaciones vectoriales de palabras y documentos</li> <li>1.3.2. Distancia euclidiana y similitud de coseno como medidas de significado.</li> <li>1.3.3. Visualización de vectores n-dimensionales: Principal Component Analysis (PCA), t-distributed stochastic neighbor embedding (t-SNE)</li> <li>1.4. Traducción con vectores de palabras. <ul style="list-style-type: none"> <li>1.4.1. La norma Frobenius.</li> <li>1.4.2. Hash tables.</li> <li>1.4.3. Local Sensitive Hashing (LSH).</li> </ul> </li> </ul> <p><b>2. Modelos probabilísticos.</b></p> <ul style="list-style-type: none"> <li>2.1. Creando un modelo para autocorrección de palabras <ul style="list-style-type: none"> <li>2.1.1. Probabilidades de palabras</li> <li>2.1.2. Planteando el problema como alineación de secuencias.</li> <li>2.1.3. Solución como programación dinámica.</li> </ul> </li> <li>2.2. Part of Speech Tagging (POS). <ul style="list-style-type: none"> <li>2.2.1. Cadenas de Markov.</li> <li>2.2.2. Modelos ocultos de Markov</li> <li>2.2.3. El algoritmo de Viterbi.</li> </ul> </li> <li>2.3. Modelos de Lenguaje <ul style="list-style-type: none"> <li>2.3.1. N-gramas</li> <li>2.3.2. Probabilidad condicional de secuencias de palabras.</li> <li>2.3.3. Probabilidad logarítmica.</li> <li>2.3.4. Un modelo de lenguaje generativo.</li> <li>2.3.5. Perplexity y su variante logarítmica como medida de evaluación de modelos de lenguaje.</li> <li>2.3.6. Palabras Out of Vocabulary (OOV).</li> </ul> </li> <li>2.4. Word Embeddings. <ul style="list-style-type: none"> <li>2.4.1. Representaciones por medio de enteros</li> <li>2.4.2. One-Hot vectors.</li> <li>2.4.3. Significado n-vectorial embebido.</li> <li>2.4.4. Métodos de Word Embeddings. <ul style="list-style-type: none"> <li>2.4.4.1. No-contextuales. <ul style="list-style-type: none"> <li>2.4.4.1.1. Word2vec, GloVe, fastText.</li> </ul> </li> <li>2.4.4.2. Contextuales. <ul style="list-style-type: none"> <li>2.4.4.2.1. ELMo, BERT.</li> </ul> </li> <li>2.4.4.3. EL método Skipgram a fondo.</li> </ul> </li> </ul> </li> </ul> <p><b>3. Modelos secuenciales.</b></p> <ul style="list-style-type: none"> <li>3.1. Redes Neuronales en NLP.</li> <li>3.2. Frameworks para redes neuronales profundas: Tensorflow, Pythorch, TRAX.</li> <li>3.3. Sentiment Analysis con redes neuronales.</li> <li>3.4. Redes Neuronales Recurrentes (RNNs). <ul style="list-style-type: none"> <li>3.4.1. Arquitecturas one-to-one, one-to-many, many-to-many y many-to-one.</li> </ul> </li> </ul>	
--	---	--

	<ul style="list-style-type: none"> <li>3.4.2. Vanilla RNN</li> <li>3.4.3. Gated Recurrent Unit GRU RNNs.</li> <li>3.4.4. Bi-Directional RNNs.</li> <li>3.4.5. Deep RNNs.</li> <li>3.4.6. RNNs y el vanishing gradient.</li> <li>3.4.7. Long Short-Term Memory LSTM RNNs.</li> <li>3.4.8. Named Entity Recognition (NER) con LSTMs.</li> <li>3.5. Redes Siamesas</li> </ul> <p><b>4. Modelos de Atención.</b></p> <ul style="list-style-type: none"> <li>4.1. Neural Machine Translation (NMT).</li> <li>4.2. Modelos sequence-to-sequence. <ul style="list-style-type: none"> <li>4.2.1. Encoders-Decoders y el cuello de botella de información.</li> <li>4.2.2. Alineación de palabras y atención.</li> <li>4.2.3. Keys, Queries y Values para recuperación de información en la capa de atención.</li> <li>4.2.4. Beam search para mejores decoders.</li> </ul> </li> <li>4.3. Bilingual Evaluation Understudy (BLEU score) como medida de evaluación de calidad en traducción de textos.</li> <li>4.4. Recall-Oriented Understudy for Gisting Evaluation (ROUGE score) como medida de evaluación de calidad en textos generados automáticamente.</li> <li>4.5. Los Transformers. <ul style="list-style-type: none"> <li>4.5.1. Aplicaciones: Sumarización de textos, auto-complete, NER, question answering, traducción, chatbots, sentiment analysis, text classification, spell-checking.</li> <li>4.5.2. Arquitectura básica.</li> <li>4.5.3. Modelos de lenguaje pre-entrenados basados en Transformers: BERT, GPT-3, T5.</li> <li>4.5.4. Self attention y las matemáticas detrás.</li> <li>4.5.5. Multi-head attention.</li> <li>4.5.6. Transfer learning.</li> <li>4.5.7. Fine-tuning.</li> <li>4.5.8. Algunas aplicaciones. <ul style="list-style-type: none"> <li>4.5.8.1. Question answering.</li> <li>4.5.8.2. Chat-bots</li> </ul> </li> <li>4.5.9. Limitantes de los Transformers <ul style="list-style-type: none"> <li>4.5.9.1. Secuencias largas.</li> <li>4.5.9.2. Complejidad computacional cuadrática de la capa de atención.</li> <li>4.5.9.3. LSH attention.</li> <li>4.5.9.4. Reversible Layers.</li> <li>4.5.9.5. El Reformer, un Transformer mas eficiente.</li> </ul> </li> </ul> </li> </ul>	
--	---	--

OBJETO DE APRENDIZAJE	METODOLOGIA	EVIDENCIAS DE APRENDIZAJE
1. Representaciones textuales y espacios vectoriales. 2. Modelos probabilísticos. 3. Modelos secuenciales. 4. Modelos de Atención.	1. Para cada unidad, se presenta una introducción por parte del maestro. 2. Para cada unidad, el maestro deja una tarea donde se aplican los conceptos vistos en clase para la resolución de problemas. La tarea requiere que el alumno revise las técnicas y concepto vistos en clase, aclare dudas y aplique las técnicas ya sea manualmente o las implemente utilizando un lenguaje de programación. 3. La discusión y el análisis se propician a partir del planteamiento de una situación problemática, donde el estudiante aporte alternativas de solución o resolver un ejercicio en el que aplique conceptos ya analizados. 4. En algunas unidades el maestro muestra directamente en una computadora, posiblemente con la ayuda de un proyector, cómo se implementan las técnicas vistas en clase usando un lenguaje de programación.  Material de Apoyo didáctico: Recursos <ul style="list-style-type: none"> <li>• Talleres para realizar ejercicios</li> <li>• Materiales gráficos: artículos, libros, diccionarios, etc.</li> <li>• Cañón</li> <li>• Rotafolio</li> <li>• Pizarrón, pintarrones</li> <li>• Proyector de acetatos</li> </ul> * Plataforma	<ul style="list-style-type: none"> <li>• Tareas escritas</li> <li>• Reportes de investigación</li> <li>• Evaluaciones parciales</li> <li>• Trabajo final integrador</li> </ul>

FUENTES DE INFORMACIÓN	EVALUACIÓN DE LOS APRENDIZAJES
Deep Learning for NLP and Speech Recognition. Uday Kamath, John Liu, James Whitaker. Springer.  Natural Language Processing with Python. Steven Bird, Ewan Klein, Edward Loper. O'reilly.  Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow. Aurélien Géron. O'reilly.  Natural Language Processing with PyTorch. Delip Rao, Brian McMahan. O'reilly.	<ul style="list-style-type: none"> <li>• Tareas escritas 20%</li> <li>• Reportes de investigación 20%</li> <li>• Evaluaciones parciales 20%</li> <li>• Trabajo final integrador 40%</li> </ul>

### Cronograma del Avance Programático

UNIDADES DE APRENDIZAJE	SEMANAS															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1. Representaciones textuales y espacios vectoriales.	■	■	■	■												
2. Modelos probabilísticos.					■	■	■	■								
3. Modelos secuenciales.									■	■	■	■				
4. Modelos de Atención.													■	■	■	■