



UNIVERSIDAD DE BUENOS AIRES

FACULTAD DE FILOSOFÍA Y LETRAS

DEPARTAMENTO: LETRAS

CÓDIGO N°:

SEMINARIO: ALGORITMOS SUPERVISADOS Y CONVENCIONES DE ANOTACIÓN PARA TAREAS DE PROCESAMIENTO DEL LENGUAJE NATURAL

MODALIDAD DE DICTADO: Ajustado a lo dispuesto por REDEC-2024-2526-UBA-DCT#FFYL

PROFESOR/A: CARRANZA, FERNANDO

1° CUATRIMESTRE 2025

AÑO: 2025

**UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE FILOSOFÍA Y LETRAS
DEPARTAMENTO DE LETRAS
CÓDIGO N°:**

SEMINARIO: ALGORITMOS SUPERVISADOS Y CONVENCIONES DE ANOTACIÓN PARA TAREAS DE PROCESAMIENTO DEL LENGUAJE NATURAL

MODALIDAD DE DICTADO: PRESENCIAL

CARGA HORARIA: 64 HORAS

1° CUATRIMESTRE 2025

PROFESOR/A: CARRANZA, FERNANDO

**EQUIPO DOCENTE COLABORADOR¹:
SCHIAFFINO, FERNANDO**

a. Fundamentación y descripción

El procesamiento del lenguaje natural es una disciplina de la inteligencia artificial que estudia la implementación de desarrollos que permitan resolver computacionalmente tareas que involucran en sentido amplio el uso del lenguaje humano. Hasta mediados de los años noventa, el método estándar de construcción de algoritmos para resolver este tipo de tareas era escribir un código que declarara qué debía hacer la computadora ante cada uno de los estímulos que recibiera. Hacia mediados de los años noventa, sin embargo, empezó a tomar fuerza un paradigma alternativo: el aprendizaje automático. Este enfoque se asienta sobre la idea de que las computadoras no se limiten a hacer exactamente lo que una persona les indicó mediante su código que hicieran, sino que sean capaces de aprender qué hacer a partir de los datos a los que se las expone. Dentro del paradigma del aprendizaje automático, los algoritmos de aprendizaje supervisado nos permiten hacer que las computadoras aprendan a hacer predicciones a partir de datos anotados previamente. Estas anotaciones constituyen, de esta manera, ejemplos de las categorías que queremos que los algoritmos aprendan a predecir. Este tipo de acercamientos ha permitido el desarrollo de diferentes algoritmos diseñados con el objeto de resolver varias tareas de procesamiento del lenguaje natural, como la identificación de clases de palabras, la descomposición morfológica, el análisis sintáctico automático (*parsing*), la identificación de expresiones que refieren a distintos tipos de entidades y sus relaciones y la detección de sentimientos y lenguaje de odio, entre muchas otras. Dado el enorme impacto que este enfoque de aprendizaje automático ha tenido sobre el procesamiento del lenguaje natural y la importancia que esta disciplina ha alcanzado en los últimos años, en este seminario nos proponemos hacer un recorrido por algunos de los algoritmos de aprendizaje supervisado y determinadas convenciones de anotación que se han utilizado para entrenar modelos de procesamiento del lenguaje natural. Para eso, en primer lugar, se revisarán algunos de los algoritmos más importantes, ya sea generativos, como el bayesiano ingenuo, o discriminativos, como la regresión logística y la máquina de soporte vectorial (support vector machine). En segundo lugar, se usarán algunos de los modelos entrenados para resolver tareas de procesamiento del lenguaje natural que pueden encontrarse en librerías de acceso público como NLTK, Spacy y Stanza, entre otras. Por último, también se revisarán algunos corpus anotados y se harán prácticas de anotación con estándares como las etiquetas morfológicas de BNC, de Universal Dependencies y de EAGLES, así como la información morfosintáctica de CONLL-U. El seminario pretende acercar al estudiantado nociones teóricas y herramientas prácticas para su introducción en uno

¹ Los/as docentes interinos/as están sujetos a la designación que apruebe el Consejo Directivo para el ciclo lectivo correspondiente.

de los enfoques más importantes de procesamiento del lenguaje natural en la actualidad, con impacto y productividad tanto en el ámbito académico como en el laboral. Además, pretende brindar los conocimientos básicos que constituyen el punto de partida de los Grandes Modelos de Lenguaje o LLMs, actual enfoque más importante en este campo

b. Objetivos:

Con este curso, se pretende que quienes asistan alcancen los siguientes objetivos:

- Familiarización con las tareas de procesamiento del lenguaje natural existentes.
- Conocer los grandes tipos de algoritmos supervisados que existen, su funcionamiento y su aplicación a tareas de procesamiento del lenguaje natural.
- Alcanzar la capacidad de anotar textos utilizando diferentes sistemas de anotación.
- Reconocer las distintas decisiones teóricas que subyacen a los sistemas de anotación existentes y qué decisiones alternativas podrían tomarse.
- Manejo básico de librerías de Python y su documentación asociada para el entrenamiento de algoritmos supervisados y para su uso en tareas de procesamiento del lenguaje natural.
- Poder correr y entender código básico en Python para interactuar con algunos modelos entrenados previamente a partir de textos anotados.

c. Contenidos:

Unidad 1: Introducción

i) la inteligencia artificial, ii) programación clásica vs. el aprendizaje automático (machine learning). iii) Procesamiento del lenguaje natural: definición, tareas y enfoques; iv) lenguaje de programación Python en Google Colab, v) Exploración de algunos recursos de libre acceso relevantes: Kaggle y Huggingface, GitHub.

Duración estimada: 2 semanas

Unidad 2: Los algoritmos supervisados

i) Aprendizaje supervisado y no supervisado. ii) Métricas usuales para medir el rendimiento de modelos de clasificación (accuracy, precisión y cobertura). iii) Anotación como tarea a resolver por un modelo predictivo. iv) Datos estructurados y no estructurados: manejo de estructuras de almacenamiento de datos (json, csv). v) Vectorización (conversión de un texto en tanto dato no estructurado en un arreglo numérico estructurado; CountVectorizer, TfidfVectorizer). vi) Modelos de clasificación: Bayesiano ingenuo, Regresión Logística, Máquina de soporte vectorial (*Support Vector Machines*).

Duración estimada: 3 semanas

Unidad 3: Anotación morfológica y de clase de palabra

a. Análisis morfológico

i) nociones básicas de morfología: raíz, lema, tema y forma de palabra, morfología flexiva y derivativa; ii) stemmatizer y lemmatizer de NLTK; iii) lemmatizer de Spacy basado en reglas y basado en aprendizaje automático; iv) etiquetamiento en CONLL-U: columnas de ID, de palabra y de lema.

Duración estimada: 2 semana

b. Clases de palabras

i) etiquetas de BNC y de Universal. Práctica de anotación. ii) etiquetamiento en CONLL-U de clase de palabra con EAGLES y de rasgos morfológicos. iii) Postag NLTK con tagsets BNC y Universal y Postag de Spacy y Stanza.

Duración estimada: 2 semanas

Unidad 4: Anotación sintáctica

i) Análisis basado en constituyentes y gramáticas basadas en dependencias. ii) Parser basado en constituyentes BLLIP y parsers basados en dependencias Spacy y Stanza. iii) Penn Treebank. Anotación sintáctica basada en constituyentes. iv) Análisis sintáctico basado en dependencias y su anotación en CONLL-U.

Duración estimada: 2 semanas y ½

Unidad 5: Anotación para propósitos específicos

a. Análisis de sentimiento

i) el análisis de sentimiento. Modelos disponibles en NLTK y Stanza. ii) Representación del texto como bolsa de palabras. iii) Representación del texto como arreglo numérico: matrices ralas (término-término, término-documento) y matrices densas (embeddings). iv) Entrenamiento de modelos de clasificación para análisis de sentimiento (Bayesiano Ingenuo, Regresión Logística y Máquina de soporte vectorial) y comparación de su rendimiento. v) Detección de lenguaje de odio. Pysentimiento.

b. Reconocimiento de entidades nombradas (NER)

i) Sistemas de anotación (BOU, BILOU). ii) NER de Spacy y Stanza y extracción de relaciones. iii) Extracción de información en reportes médicos. iv) Detección de relaciones en informes médicos.

c. Implementación de identificador de similitud semántica

i) armado de dataset con oraciones. ii) métodos de medición de similitud semántica (similitud coseno, similitud por presencia de token, word2vec, FastText)

d. Entendimiento del lenguaje natural (Natural Language Understanding) en la era de los grandes modelos de lenguaje.

Duración estimada: 4 semanas ½

d. Bibliografía, filmografía y/o discografía obligatoria, complementaria y fuentes, si correspondiera:

Nota importante: Una parte de la bibliografía está en inglés. Esto obedece a que la temática abordada no tiene una tradición robusta, asentada y actualizada en español. Traducir la bibliografía es un trabajo arduo que excede la dedicación que se asigna para el dictado de un seminario de grado y reemplazar textos en inglés por textos en español implica atenerse a bibliografía mayormente desactualizada y no

siempre de la calidad esperada, a la vez que quita a los estudiantes la posibilidad de leer acompañados bibliografía en inglés en un campo en el que la lectura en esa lengua es, lamentablemente, un requisito. Por este motivo, preferimos conservar bibliografía en inglés y, como se ha hecho en otros seminarios en los que participaron miembros del equipo docente, se acompañará y se brindarán recursos para ayudar a abordar la bibliografía en inglés a quienes manifiesten dificultades al respecto. Esperamos que del dictado del seminario puedan surgir recursos en español que hagan este campo más accesible a la comunidad hispanohablante.

Unidad 1

Bibliografía obligatoria

- Álvarez Vega, Marianella, Laura María Quirós Mora y Mónica Valeria Cortés Badilla “Inteligencia artificial y aprendizaje automático en medicina”. *Revista Médica Sinergia*. Vol.5 Num.8, Agosto 2020, e557. Disponible en <https://www.revistamedicasinergia.com/index.php/rms/article/download/557/923> (para puntos i y ii)
- Domínguez Burgos, Andrés. “Lingüística computacional. Un esbozo.” BOLETÍN DE LINGÜÍSTICA, Vol. 18 / 2002: 104 -119. Disponible en <https://www.redalyc.org/pdf/347/34701805.pdf> (para punto iii)
- Downey, Allen, Jeffrey Elkner y Chris Meyers. 2002. “El camino del programa”. *Aprenda a Pensar Como un Programador con Python*. Massachusetts: Green Tea Press. Capítulo 1: 1-9. Disponible en <https://argentinaenpython.com/quiero-aprender-python/aprenda-a-pensar-como-un-programador-con-python.pdf>. (para punto iv)
- Severance, Charles. 2009. “Variables, expresiones y sentencias”. *Python para todos Explorando la información con Python 3*. Capítulo 2: 19-32. https://do1.dr-chuck.com/pythonlearn/ES_es/pythonlearn.pdf.
- Severance, Charles. 2009. “Funciones”. *Python para todos Explorando la información con Python 3*, Capítulo 4: 45-57. https://do1.dr-chuck.com/pythonlearn/ES_es/pythonlearn.pdf.

Bibliografía complementaria

- Turing, Alan. 1950. “Computing machinery and Intelligence”. *Mind* LIX(236): 433-460. Hay versión en español en <http://xamanek.izt.uam.mx/map/cursos/Turing-Pensar.pdf> (para punto i).
- Chollet, F. 2018. “What is deep learning”. *Deep learning with Python*. Capítulo 1. (para puntos i y ii)
- Chollet, F. 2018. “Fundamentals of machine learning”. *Deep learning with Python*. Capítulo 4. (para puntos i y ii)
- Downey, Allen, Jeffrey Elkner y Chris Meyers. 2002. “Variables, expresiones y sentencias”. *Aprenda a Pensar Como un Programador con Python*. Massachusetts: Green Tea Press. Capítulo 2: 11-19. Disponible en <https://argentinaenpython.com/quiero-aprender-python/aprenda-a-pensar-como-un-programador-con-python.pdf>. (para punto iv)
- Downey, Allen, Jeffrey Elkner y Chris Meyers. 2002. “Funciones”. *Aprenda a Pensar Como un Programador con Python*. Massachusetts: Green Tea Press. Capítulo 3: 21-32. Disponible en <https://argentinaenpython.com/quiero-aprender-python/aprenda-a-pensar-como-un-programador-con-python.pdf>. (para punto iv)
- Severance, Charles. 2009. “¿Por qué deberías aprender a escribir programas?”. *Python para todos Explorando la información con Python 3*. Capítulo 1: 1-18. https://do1.dr-chuck.com/pythonlearn/ES_es/pythonlearn.pdf.

Recursos computacionales

- Github: <https://github.com/> (para punto v)
- Google Colab: <https://colab.google/> (para punto iv)
- Huggingface: <https://huggingface.co/> (para punto v)
- Jupyter notebook: <https://jupyter.org/> (para punto iv)
- Kaggle: <https://www.kaggle.com/> (para punto v)
- Python: <https://www.python.org> (para iv)

Unidad 2

Bibliografía obligatoria

- Jurafsky, Daniel y James H. Martin. 2024. “Naive Bayes, Text Classification, and Sentiment”. Speech and Language Processing. Versión no final de febrero de 2024 (para puntos vi).
- Jurafsky, Daniel y James H. Martin. 2024. “Logistic regression”. Speech and Language Processing. Versión no final de Febrero de 2024 (para puntos vi).
- Tan, Pang-Ning, Michael Steinbach y Vipin Kumar. 2006. “Classification: Alternative Techniques”. Introduction to data mining. Pearson, pp. 208-326 (para punto vi).

Bibliografía complementaria

- Moreno Sandoval, Antonio. 1998. “Modelos probabilísticos”. Lingüística computacional. Síntesis, pp. 153-210 (para punto vi).

Recursos computacionales

- Algoritmos supervisados de clasificación de ScikitLearn: https://scikit-learn.org/stable/supervised_learning.html (para punto vi)
- Bayesiano ingenuo de ScikitLearn: https://scikit-learn.org/stable/modules/naive_bayes.html (para punto vi)
- Bayesiano ingenuo de NLTK <https://www.nltk.org/api/nltk.classify.naivebayes.html> (para punto vi)
- Extracción de rasgos de ScikitLearn: https://scikit-learn.org/1.5/api/sklearn.feature_extraction.html (para punto v)
- Librería Pandas para manipulación de bases de datos: <https://pandas.pydata.org/> (para punto iv)
- Máquina de soporte vectorial de ScikitLearn: <https://scikit-learn.org/stable/modules/svm.html> (para punto vi)
- Regresión logística en ScikitLearn: https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression (para punto vi)

Unidad 3

Bibliografía obligatoria

- Bird, S, E. Klein y E. Loper. “Categorizing and Tagging Words”. Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit. Capítulo 5. Disponible en <https://www.nltk.org/book/ch05.html> (para puntos a.ii y a.i)
- Real Academia Española (RAE) y Asociación de Academias de la Lengua Española (ASALE). 2009. “Partes de la gramática. Unidades fundamentales del análisis gramatical”. *Nueva gramática de la lengua española*. Madrid, Espasa Calpe. Cap. 1 (para puntos a.i y b, se retoma también en la unidad 4).

- Universitat politècnica de Catalunya. “Introducción a las etiquetas de EAGLES”. Disponible en <https://www.cs.upc.edu/~nlp/tools/parole-sp.html> (Para punto b.ii).

Bibliografía complementaria

- Feliú Arquiola, Elena. 2009. “Palabras con estructura interna”. Cap. 2 en E. De Miguel (ed). Panorama de la lexicología. Ariel. (para punto b)
- Giammatteo, Mabel. e Hilda Albano. 2006. *¿Cómo se clasifican las palabras?* Biblos. (para punto b).
- Radford et al. 2000. “Construyendo palabras”. *Introducción a la lingüística*. Cambridge University Press. pp. 231-252. (para punto b).

Recursos computacionales

- NLTK: <https://www.nltk.org/>
- Lematizador de Spacy: <https://spacy.io/api/lemmatizer>
- Lematizador de Stanza: <https://stanfordnlp.github.io/stanza/lemma.html>
- Tagger de Spacy: <https://spacy.io/api/tagger>
- Spanish Inflections: https://github.com/mathigatti/spanish_inflections
- Analizador morfológico y de clase de palabra de Stanza: <https://stanfordnlp.github.io/stanza/pos.html>
- Analizador morfológico de Spacy: <https://spacy.io/api/morphologizer>

Unidad 4

Bibliografía obligatoria

- Carnie, Andrew. 2007. “Dependency and Constituency”. *Constituent Structure*. Oxford University Press, pp. 168-188 (para punto i).
- Universal Dependencies. Documentación sobre el formato CONLL-U. Disponible en <https://universaldependencies.org/format.html> (para punto iv)
- Marcus, Mitchell, Mary Ann Marcinkiewicz y Beatrice Santorini. 1993. “Building a Large Annotated Corpus of English: The Penn Treebank “. Disponible en <https://aclanthology.org/J93-2004.pdf> (para punto iii)
- Charniak, Eugene y Mark Johnson. 2005. “Coarse-to-fine n-best parsing and maxent discriminative reranking”. En: *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL’05)*, 173-180). Disponible en <https://aclanthology.org/P05-1022.pdf> (para punto ii)

Bibliografía complementaria:

- Real Academia Española. 2009. *Nueva gramática de la lengua española*. Madrid, Espasa Calpe. Cap. 33-36 y 39 (para punto i y iii).
- Roark, B. y R. Sproat. 2007. “Basic Context-free approaches to Syntax”. *Computational Approaches to Morphology and Syntax*. Capítulo 7. Oxford University Press. (para puntos i y ii)
- Roark, B. y R. Sproat. 2007. “Enriched Context-Free Approaches to Syntax”. *Computational Approaches to Morphology and Syntax*. Capítulo 8. Oxford University Press. (para puntos i y ii)

Recursos computacionales

- ArboratorGrew: <https://arboratorgrew.elizia.net/#/> (para punto iv)
- Blliparser: <https://pypi.org/project/blliparser/> (para punto ii)
- Corpus anotado Ancora: https://github.com/UniversalDependencies/UD_Spanish-AnCora (para

- punto iv)
- DgAnnotator: <http://medialab.di.unipi.it/Project/QA/Parser/DgAnnotator/> (para punto iv)
- Identificación de tipos de cláusulas: https://github.com/johnosbb/NLP/blob/main/Spacy/clause_types.md (para punto ii)
- Parser de dependencias de Spacy: <https://spacy.io/api/dependencyparser> (para punto ii)
- Parser de dependencias de Stanza: <https://stanfordnlp.github.io/stanza/depparse.html> (para punto ii)
- Parser de constituyentes de Stanza: <https://stanfordnlp.github.io/stanza/constituency.html>
- UPENN: <https://catalog ldc.upenn.edu/docs/LDC95T7/cl93.html> (para punto iii)

Unidad 5

Bibliografía obligatoria

- Bender, E. M., y A. Koller, A. 2020. “Climbing towards NLU: On meaning, form, and understanding in the age of data”. En: Proceedings of the 58th annual meeting of the association for computational linguistics, pp. 5185-5198. Disponible en <https://aclanthology.org/2020.acl-main.463.pdf> (para punto d)
- Cotik, Viviana, Laura Alonso Alemany, Darío Filippo, Franco Luque, Roland Roller, Jorge Vivaldi, Ammer Ayach, Fernando Carranza, Lucas Defrancesca, Antonella Dellanzo y Macarena Fernández Urquiza. 2021. “Overview of CLEF eHealth Task 1 - SpRadIE: A challenge on information extraction from Spanish Radiology Reports”. CEUR Workshop Proceedings de CLEF 2021 – Conference and Labs of the Evaluation Forum, September 21–24, 2021, Bucharest, Romania: 1-19 (para punto b.3).
- Jurafsky, Daniel y James H. Martin. 2024. “Vector semantics and embeddings”. Speech and Language Processing. Draft of February 3, 2024 (para puntos a.ii, a.iii).
- Kedia, A y M. Rasu. 2020. “Identifying Patterns in Text Using Machine Learning”. *Hands-On Python Natural Language Processing*. Birmingham: Packt. Capítulo 7: 147-168 (para punto a.iv)
- Lance A. Ramshaw, Mitchell P. Marcus. 1995. Text Chunking using Transformation-Based Learning. <https://arxiv.org/abs/cmp-lg/9505040> (para punto b.i).
- Minces Müller, Javier. 2020. “Resumen”. *Detección de relaciones en informes médicos escritos en español*. Tesis de Licenciatura en Ciencias de la Computación: Universidad de Buenos Aires, p 4. Disponible en https://gestion.dc.uba.ar/media/academic/grade/thesis/Tesis_Javier_Minces.pdf (para punto b.iv)
- Pérez, Juan Manuel. 2022. “Resumen”. Técnicas y recursos para la detección automática de lenguaje discriminatorio en redes sociales. Tesis de doctorado: UBA, pp 1-2. Disponible en https://bibliotecadigital.exactas.uba.ar/download/tesis/tesis_n7305_Perez.pdf (para punto a.v)

Bibliografía complementaria

- Polignano, Marco, Marco de Gemmis y Giovanni Semeraro. “Comparing Transformer-based NER approaches for analysing textual medical diagnoses”. Disponible en <https://ceur-ws.org/Vol-2936/paper-68.pdf> (para punto b.iii)
- Severance, Charles. 2009. “Expresiones regulares”. *Python para todos Explorando la información con Python 3*, Capítulo 11: 137-149. https://do1.dr-chuck.com/pythonlearn/ES_es/pythonlearn.pdf.
- Suominen, Hanna, Lorraine Goeriot, Liadh Kelly, Laura Alonso Alemany, Elias Bassani, et al. 2021. “Overview of the CLEF eHealth Evaluation Lab 2021. Experimental IR Meets

Multilinguality, Multimodality, and Interaction”, Disponible en https://hal.science/hal-03369846/file/CLEF_eHealth_21_LNCS_Overview.pdf (para punto b.iii)

Recursos computacionales:

- Analizador de sentimiento de NLTK: <https://www.nltk.org/howto/sentiment.html> (para punto a)
- Analizador de sentimiento de Stanza: <https://stanfordnlp.github.io/stanza/sentiment.html> (para punto b.ii)
- Fasttext: <https://fasttext.cc/> (para punto c.ii)
- Labelstudio: <https://labelstud.io/> (para punto b.i)
- Numpy: <https://numpy.org/> (para punto c.ii)
- Pysentimiento: <https://pypi.org/project/pysentimiento/0.5.2rc3/> (para punto a.iv)
- Reconocedor de entidades de Spacy: <https://spacy.io/usage/linguistic-features#named-entities>, <https://spacy.io/api/entityrecognizer> (para punto b.ii)
- Reconocedor de entidades de Stanza: <https://stanfordnlp.github.io/stanza/ner.html> (para punto b.ii)
- Reconocedor de relaciones entre entidades de Spacy: <https://spacy.io/api/entitylinker> (para punto b.ii)
- Word2vec: <https://www.tensorflow.org/text/tutorials/word2vec> (para punto c.ii)

Bibliografía general

- Altinok, Duygu. 2021. *Mastering spaCy: An end-to-end practical guide to implementing NLP applications using the Python ecosystem*. Packt.
- Báez San José, Valerio. 1988. *Fundamentos críticos de la Gramática de Dependencias*. Madrid: Síntesis.
- Bender, Emily. 2013. *Linguistic Fundamentals for Natural Language Processing: 100 Essentials from Morphology and Syntax*. Toronto: Morgan & Claypol.
- Bender, Emily y Alex Lascarides. 2020. *Linguistic Fundamentals for Natural Language Processing: 100 Essentials from Semantics and Pragmatics*. Toronto: Morgan & Claypol.
- Bird, Steven, Ewan Klein y Edward Loper. 2009. *Natural Language Processing with Python*. Sebastopol: O'Reily, pp. 291-326.
- Carnie, Andrew. 2010. *Constituent Structure*. Oxford: Oxford University Press.
- Clark, Alexander, Chris Fox y Shalom Lappin (eds). *The Handbook of Computational Linguistics and Natural Language Processing*. Massachusetts: Willey Blackwell.
- Cotik, Viviana. Una propuesta conexionista para el el reconocimiento y predicción de promotores en secuencias de ADN de procariotas. Tesis: Universidad de Buenos Aires. Disponible en <https://gestion.dc.uba.ar/media/academic/grade/thesis/tesisEntregadaCotikLicenciatura.pdf>
- Downey, Allen, Jeffrey Elkner y Chris Meyers. 2002. *Aprenda a Pensar Como un Programador con Python*. Massachusetts: Green Tea Press. Disponible en <https://argentinaenpython.com/quiero-aprender-python/aprenda-a-pensar-como-un-programador-con-python.pdf>
- Gerón, Aurélien. 2020. *Aprende Machine Learning con Scikit-Learn, Keras y TensorFlow*. O'Realy.
- Gómez-Rodríguez, Carlos y Joakim Nivre. 2010. “A Transition-Based for 2-Planar Dependency Structures”. En: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 1492-1501.
- González Duque, Raúl. Python para todos. Disponible en

<https://persoal.citius.usc.es/eva.cernadas/informaticaparacientificos/material/libros/Python%20para%20todos.pdf>

- Guibon, Gaël, Marine Courtin, Kim Gerdes y Bruno Guillaume. 2020. “When Collaborative Treebank Curation Meets Graph Grammars”. En: *Proceedings of the Twelfth Language Resources and Evaluation Conference*. Disponible en <https://aclanthology.org/2020.lrec-1.651/>
- Hays, David. 1964. “Dependency Theory: A Formalism and Some Observations”. *Language* 40. 511-525.
- Hudson, Richard. 1984. *Word Grammar*. Oxford: Basil Blackwell.
- Hudson, Richard. 2010. *An introduction to Word Grammar*. Cambridge: Cambridge University Press.
- Kedia, Aman y Mayank Rasu. 2020. *Hands-On Python Natural Language Processing*. Birmingham: Packt.
- Mel’čuk, Igor A. 1988. *Dependency Syntax: Theory and Practice*. Albany: State University of New York Press.
- Minces Müller, Javier. 2020. *Detección de relaciones en informes médicos escritos en español*. Tesis de Licenciatura en Ciencias de la Computación: Universidad de Buenos Aires. Disponible en https://gestion.dc.uba.ar/media/academic/grade/thesis/Tesis_Javier_Minces.pdf
- Moreno Sandoval, Antonio. 1998. *Lingüística computacional: Introducción a los modelos simbólicos, estadísticos y biológicos*. Madrid: Síntesis
- Nivre, Joakim, Laura Rimell, Ryan McDonald y Carlos Gómez-Rodríguez. 2010. “Evaluation of Dependency Parser of Unbounded Dependencies”. En: *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*. Beijing, 833-841.
- Nivre, Joakim. 2003. “An Efficient Algorithm for projective dependency parsing”. En: *Proceedings of the 8th International Workshop on parsing Technologies (IWPT 03)*. Nancy, Francia. Pp. 149-160.
- Nivre, Joakim, Johan Hall y Jens Nilsson. 2004. “Memory-based Dependency Parsing”. En: H. T. Ng y E. Riloff (eds.). *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL)*, 6 y 7 de Mayo de 2004, Boston, Massachusetts, pp. 49-56.
- Pérez, Juan Manuel. 2022. *Técnicas y recursos para la detección automática de lenguaje discriminatorio en redes sociales*. Tesis de doctorado: Universidad de Buenos Aires. Disponible en https://bibliotecadigital.exactas.uba.ar/download/tesis/tesis_n7305_Perez.pdf (para punto iv)
- Roark, Brian, y Richard Sproat. 2007. *Computational approaches to morphology and syntax*. Oxford: Oxford University Press.
- Severance, Charles. 2009. *Python para todos. Explorando la información con Python 3*. https://do1.dr-chuck.com/pythonlearn/ES_es/pythonlearn.pdf
- Stricker, Vanesa. 2016. *Detección de Negaciones en Informes Radiológicos escritos en Español*. Tesis de licenciatura en Ciencias de la Computación: Universidad de Buenos Aires. Disponible en <https://gestion.dc.uba.ar/media/academic/grade/thesis/stricker.pdf>
- Tesnière, Lucien. 1959. *Éléments de syntaxe structurale*. Paris: Klincksieck.

Recursos computacionales:

- Brat (herramienta de anotación): <https://brat.nlplab.org>, <https://github.com/nlplab/brat>
- GitHub (repositorio): <https://github.com/>
- Huggingface (repositorio): <https://huggingface.co/>
- Kaggle (repositorio): <https://www.kaggle.com/>
- Keras (librería de Python): <https://keras.io/>
- Labelstudio (herramienta de anotación): <https://labelstud.io/>
- NLTK (librería de Python): <https://www.nltk.org/>

- Numpy (librería de Python): <https://numpy.org/>
- Pandas (librería de Python): <https://pandas.pydata.org/>
- Pysentimiento (librería de Python): <https://pypi.org/project/pysentimiento/0.5.2rc3/>
- Pytorch (librería de Python): <https://pytorch.org/>
- Scikitlearn (librería de Python): <https://scikit-learn.org/stable/>
- Spacy (librería de Python): <https://spacy.io/>
- Stanza (librería de Python): <https://stanfordnlp.github.io/stanza/>
- TensorFlow (librería de Python): <https://www.tensorflow.org/?hl=es>
- TextBlob (librería de Python): <https://github.com/sloria/TextBlob>

e. **Organización del dictado de seminario**

El seminario / proyecto se dicta atendiendo a lo dispuesto por REDEC-2024-2526-UBA-DCT#FFYL la cual establece pautas complementarias para el dictado de las asignaturas de grado durante el Ciclo Lectivo 2025.

Seminario cuatrimestral (presencial)

El seminario se dictará completamente en modalidad presencial en un aula y horario predefinidos.

El seminario se llevará a cabo mediante clases presenciales que se realizarán con el acompañamiento de materiales y guías de clase para facilitar el seguimiento. Estas guías incluirán, según corresponda, presentaciones de diapositivas, handouts, Google Colabs, scripts y tutoriales, que quedarán disponibles para los estudiantes durante la cursada.

Dada la naturaleza del seminario, en el mismo habrá exposiciones teóricas y demostraciones con computadora, en las que se intercalarán instancias prácticas. Para conservar la regularidad, el equipo docente proveerá una serie de trabajos prácticos por unidad que deberán ser resueltos en tiempo y forma en instancias asincrónicas. Del desempeño en estos trabajos prácticos se extraerá la nota de la cursada.

Carga Horaria:

Seminario cuatrimestral

La carga horaria mínima es de 64 horas (sesenta y cuatro) y comprenden un mínimo de 4 (cuatro) y un máximo de 6 (seis) horas semanales de dictado de clases.

f. **Organización de la evaluación**

El sistema de regularidad y aprobación del seminario se rige por el Reglamento Académico (Res. (CD) N° 4428/17):

Regularización del seminario:

Es condición para alcanzar la regularidad de los seminarios:

- i. asistir al 80% de las reuniones y prácticas dentro del horario obligatorio fijado para la cursada;
- ii. aprobar una evaluación con un mínimo de 4 (cuatro) la cursada. Para ello el/la Docente a cargo

dispondrá de un dispositivo durante la cursada.

Aprobación del seminario:

Los/as estudiantes que cumplan el requisito mencionado podrán presentar el trabajo final integrador que será calificado con otra nota. La calificación final resultará del promedio de la nota de cursada y del trabajo final integrador.

Si el trabajo final integrador fuera rechazado, los/as interesados/as tendrán la opción de presentarlo nuevamente antes de la finalización del plazo de vigencia de la regularidad. El/la estudiante que no presente su trabajo dentro del plazo fijado, no podrá ser considerado/a para la aprobación del seminario.

VIGENCIA DE LA REGULARIDAD: El plazo de presentación del trabajo final de los seminarios es de 4 (cuatro) años posteriores a su finalización.

RÉGIMEN TRANSITORIO DE ASISTENCIA, REGULARIDAD Y MODALIDADES DE EVALUACIÓN DE MATERIAS: El cumplimiento de los requisitos de regularidad en los casos de estudiantes que se encuentren cursando bajo el Régimen Transitorio de Asistencia, Regularidad y Modalidades de Evaluación de Materias (RTARMEM) aprobado por Res. (CD) N° 1117/10 quedará sujeto al análisis conjunto entre el Programa de Orientación de la SEUBE, los Departamentos docentes y los/las Profesores a cargo del seminario.

g. Recomendaciones

Además de Gramática y Lingüística, se recomienda asimismo haber cursado Sintaxis, Fonología y Morfología o alguna materia afín y Modelos Formales No Transformacionales. El seminario es de incumbencia para todos aquellos interesados en la lingüística formal, la gramática, la lingüística computacional, el procesamiento del lenguaje natural y/o las ciencias de la computación. Si bien se llevarán a cabo algunas implementaciones computacionales de los modelos trabajados, fundamentalmente en Python, no se presuponen conocimientos de programación previos. Se recomienda que los estudiantes sean competentes en la lectura de textos académicos en inglés.

Dr. Fernando Martín Carranza

MIGUEL VEDDA
DIRECTOR DEL DEPARTAMENTO DE

LETRAS