PUCE Virtual

CLASE

7

Procesamiento de Lenguaje Natural

Validación semántica y ética

QS WORLD UNIVERSITY RANKINGS

Educación de calidad

INTRODUCCIÓN DE LA CLASE

GRADO / POSGRADO / TECNOLOGÍAS

Estudia a tu tiempo
son interrupciones

EDUCACIÓN EN LÍNEA DE CALIDAD

En la Semana 6 el curso avanzó desde la clasificación superficial de textos hacia el análisis del significado, la intención comunicativa y las señales de ingeniería social. Sin embargo, construir un modelo semántico no basta: es necesario demostrar que realmente funciona frente a mensajes ambiguos, reformulados o diseñados para evadir filtros automáticos. Por esta razón, la Semana 7 se centra en la validación del sistema semántico, es decir, en cómo evaluar si el modelo identifica correctamente amenazas que no dependen únicamente de palabras clave explícitas.

Al mismo tiempo, esta semana introduce un componente fundamental en sistemas de PLN aplicados a ciberseguridad: la ética. Un modelo capaz de interpretar intención, detectar anomalías o inferir manipulación puede ser muy útil, pero también implica riesgos relacionados con sesgos, privacidad, sobreinterpretación y opacidad de decisión. Por ello, la validación del sistema no debe limitarse a medir aciertos, sino también a analizar cómo toma decisiones, qué tan explicables son sus resultados y cuáles son los límites éticos de interpretar el lenguaje humano en contextos sensibles.

RDA 3: Evaluar sistemas de procesamiento de lenguaje natural para apoyar la toma de decisiones en ciberseguridad, a través de técnicas y métricas de precisión, eficacia, explicabilidad e impacto ético de los modelos.

Reto 3

Validación semántica y ética.

Evaluación cualitativa de casos de evasión.

¿Qué significa validar semánticamente un sistema?

Validar semánticamente un sistema significa comprobar si el modelo puede reconocer amenazas reformuladas, mensajes ambiguos o variantes lingüísticas que no repiten exactamente las palabras vistas en entrenamiento. Más que responder a: ¿clasifica bien?, es:

¿entiende suficientemente el significado como para detectar evasión?

En “Cuál es la diferencia entre VULNERABILIDAD, AMENAZA y RIESGO en la CIBERSEGURIDAD? | Quanti Guides” https://www.youtube.com/watch?v=5OLeuzwEeng se hace una distinción.

¿Qué es un caso de evasión?

Un caso de evasión es un mensaje que intenta evitar la detección automática modificando su forma lingüística sin cambiar su intención. Por ejemplo: “Verifique su cuenta inmediatamente para evitar suspensión.” Su variante evasiva es: “Se recomienda actualizar su acceso a la plataforma para mantener disponibilidad del servicio.”

Aunque las palabras son diferentes, la intención puede seguir siendo:

- inducir acción,
- crear presión,
- obtener información sensible.

¿Por qué la evaluación cualitativa es necesaria?

Las métricas cuantitativas como Precision o Recall son útiles, pero no explican cómo falla o acierta un sistema semántico. La evaluación cualitativa permite analizar:

- qué tipo de reformulaciones detecta,
- qué señales semánticas reconoce,
- qué casos le generan confusión,
- qué mensajes parecen legítimos pero no lo son.

Esto es especialmente importante en el Reto 3, donde el objetivo no es solo clasificar, sino identificar intención maliciosa más allá de palabras clave.

Tipos de evasión

- Sustitución léxica: Cambiar palabras directas por sinónimos o expresiones más neutrales. Ejemplos: “verifique” → “valide”; “suspensión”→”restricción temporal”.
- Reformulación pragmática: Convertir una orden explícita en una recomendación o sugerencia. Ejemplos: “Debe actualizar sus credenciales”; “Se recomienda actualizar su acceso”.
- Evasión por tono: Reducir urgencia explícita y adoptar un tono institucional o amable.
- Evasión por contexto: Insertar el mensaje malicioso dentro de un contexto aparentemente técnico o administrativo.

Metodología de evaluación cualitativa

La evaluación cualitativa puede estructurarse en una tabla de casos (Eisenstein,2019):

Texto	Tipo	Predicción	Interpretación
“Valide su acceso para mantener operatividad”	Sustitución léxica	Ataque	Detectó intención
“Revise el informe cuando tenga tiempo”	Legítimo	Legítimo	Decisión correcta

Criterios cualitativos de análisis

Al revisar cada caso, se debe responder:

- ¿Qué señal semántica conserva el mensaje?
- ¿Qué cambió respecto al ataque explícito?
- ¿El modelo detectó la intención o solo coincidencias léxicas?
- ¿La decisión parece justificable?

De esta manera, es posible distinguir entre:

- detección superficial
- comprensión semántica real

Ejemplo Python

Tomando como base el recurso de profundización de la semana 6 (sistema híbrido), se construye el siguiente conjunto de casos de evasión (Ilustración 1):

```
casos = [
    "Valide su acceso para mantener continuidad del servicio",
    "Actualice su información ahora para evitar restricción",
    "Reunión del equipo a las 10 de la mañana",
    "Por favor revise el documento adjunto cuando pueda"
]

for caso in casos:
    print(caso, "->", sistema_hibrido(caso))
```

Ilustración 1. Evaluación cualitativa de evasión (creación de autor Rafael Melgarejo)

Con el resultado, se debe discutir:

- ¿Por qué se acertó?
- ¿Por qué falló?
- ¿Quá ptrón

Error comun

Un error frecuente es asumir que si el modelo detecta casos explícitos, también detecta reformulaciones. Esto no ocurre necesariamente, pues muchos sistemas funcionan bien con ejemplos prototípicos, y fallan con lenguaje diplomático, técnico o ambiguo. Por eso es crucia la validación (discusión).

Un sistema semántico robusto no solo reconoce ataques típicos, sino también sus disfraces lingüísticos.

Explicabilidad del modelo.

En sistemas de PLN aplicados a ciberseguridad no es suficiente que el modelo diga: "ALERTA: posible ataque"; sino que es necesario responder (Eisenstein,2019):

¿Por qué el sistema tomó esa decisión?

La explicabilidad del modelo se refiere a la capacidad de entender, interpretar y justificar las decisiones del sistema. En un SOC (Security Operations Center), esto es crítico porque:

- los analistas deben confiar en el sistema,
- necesitan justificar decisiones operativas,
- deben poder auditar errores o falsos positivos.

Un sistema que no puede explicar sus decisiones se considera una “caja negra”, lo cual es problemático en entornos críticos.

Tipos de explicabilidad

Existen dos niveles principales:

a) **Explicabilidad global:** Describe cómo funciona el modelo en general. Ejemplos:

- qué variables influyen más,
- qué patrones detecta,
- cómo toma decisiones en promedio.

b) **Explicabilidad local**: Explica por qué el modelo tomó una decisión específica. Ejemplo: "Este mensaje fue clasificado como ataque porque contiene urgencia, acción y referencia a autoridad"

En ciberseguridad, la explicabilidad local es la más importante.

Explicabilidad en modelos de PLN

Dependiendo del modelo, la explicabilidad puede variar (Jurafsky,2023):

Modelo	NIVEL DE EXPLICABILIDAD
REGLAS HEURÍSTICAS	Alta
NAIVE BAYES TF-IDF	Media
EMBEDDINGS / DEEP LEARNING	Baja

Esto genera un trade-off:

Más precisión ≠ más explicabilidad

Ejemplo: en el mensaje: "El equipo de seguridad solicita validar su cuenta inmediatamente", la salida del sistema puede ser: "ALERTA: posible ataque". La posible explicación sería:

- detecta autoridad → "equipo de seguridad"
- detecta acción → "validar"
- detecta urgencia → "inmediatamente"

Esto es: una combinación de señales de ingeniería social

Métodos de explicabilidad

a) **Basado en reglas**: El más simple y directo (Ilustración 2):

```
if urgencia and accion:
    alerta = True
```

Ilustración 2. Explicabilidad basada en reglas (contenido de autor Rafael Melgarejo)

Explicación: "Se detectaron patrones de urgencia y acción"

b) **Basado en pesos (TF-IDF):** Se pueden identificar las palabras más relevantes, viendo qué terminos influyeron en la decisión. Ilustración 3 muestra código de ejemplo:

```
feature_names = vectorizer.get_feature_names_out()
```

Ilustración 3. Explicación vectorizada basada en pesos (creación de autor Rafael Melgarejo)

c) **Basado en similitud semántica:** Explicación: "Este mensaje es similar a otros ataques conocidos".

d) **Métodos avanzados (conceptual):** LIME y SHAP permiten explicar modelos complejos, aunque no siempre son necesarios en este curso.

Ejemplo Python

Ilustración 4 muestra una explicación simple usando código Python

```
def explicar_decision(texto):
    texto = texto.lower()

    razones = []

    if "inmediatamente" in texto:
        razones.append("urgencia")
    if "seguridad" in texto:
        razones.append("autoridad")
    if "verifique" in texto or "valide" in texto:
        razones.append("acción")

    return razones


texto = "Valide su cuenta inmediatamente"
print(explicar_decision(texto))
```

Ilustración 4. Explicar decisión (creación de autor Rafael Melgarejo)

De esta manera, se amplía el sistema del recurso de profundización de la semana 6, con el código de la Ilustración 5:

```
resultado = sistema_hibrido(texto)
explicacion = explicar_decision(texto)
```

Ilustración 5. Sistema integrado: híbrido y explicación (creación de autor Rafael Melgarejo)

Así, el sistema se transforma en:

clasificador + justificador

Buenas prácticas

Un buen sistema además de detectar amenazas también puede explicar por qué las detecta. Si el sistema no es explicable:

- los analistas pueden ignorar las alertas,
- no se pueden corregir los errores fácilmente,
- aumenta el riesgo de decisiones incorrectas,
- se pierde trazabilidad del sistema

Por ello, se recomienda las siguientes buenas prácticas

- siempre acompañar predicciones con explicación,
- usar modelos interpretables cuando sea posible,
- documentar criterios de decisión,
- permitir auditoría del sistema.

Para conocer más revise el video “Explicabilidad en Procesamiento de Lenguaje Natural” de ACHIRP, https://www.youtube.com/watch?v=pPoeMzJwLTw.

Niveles de Explicabilidad en PLN para ciberseguridad (Floridi et al., 2018)

La explicabilidad en PLN opera en tres niveles integrados

a) Local (caso específico): ¿por qué este mensaje fue marcado?

b) Global (modelo): ¿cómo decide el sistema en general?

c) Operativo (sistema): ¿Cómo se documenta y audita la decisión?

A nivel de diseño técnico, un sistema PLN explicable en ciberseguridad debe incluir:

- Capa de logging explicativo, con la información mínima del JSON (Ilustración 6).

```
#JSON

 {
  "input": "mensaje...",
  "prediccion": "malicioso",
  "confianza": 0.91,
  "tokens_clave": ["credenciales", "urgente", "enlace"],
  "metodo_explicacion": "SHAP"
}
```

Ilustración 6. JSON con información explicativa (contenido de autor Rafael Melgarejo)

- Interfaz de auditoría: visualización de tokens relevantes e historial de decisiones.
- Política de umbrales explicables: justificación del por qué un score (digamos >0.85) implica bloqueo.

Límites éticos del análisis semántico.

A medida que los sistemas de PLN avanzan desde la detección de palabras clave hacia la interpretación de intención, se vuelven más poderosos, pero también más sensibles. Un modelo capaz de inferir si un mensaje es manipulador, urgente o sospechoso está, en cierto sentido, interpretando el comportamiento humano. Esto introduce riesgos importantes si no se establecen límites claros.

En contextos de ciberseguridad, donde las decisiones pueden afectar accesos, bloqueos o investigaciones internas, es fundamental garantizar que los sistemas no solo sean precisos, sino también justos, transparentes y responsables. La ética no es un complemento: es una condición necesaria para la implementación real de estos sistemas.

Principales riesgos éticos

a) **Sesgos en el modelo**: Los modelos aprenden de datos. Si los datos contienen sesgos, el sistema los aprende y los replica. Ejemplo: Si muchos ataques en el

dataset están escritos en cierto tono o idioma, el sistema puede asociar incorrectamente ese patrón de amenaza. El riesgo son los falsos positivos sistemáticos sobre ciertos tipos de comunicación.

b) **Sobreinterpretación** de intención: Los modelos semánticos pueden inferir intención donde no existe. Ejemplo: "Sería bueno revisar esto pronto", puede ser: una sugerencia legítima, o interpretado como urgencia sospechosa. Riesgo: el sistema "lee demasiado" en el lenguaje.

c) **Falsos positivos con impacto operativo**: Un mensaje legítimo puede ser marcado como ataque, con las consecuencias correspondientes:
   - bloqueo de comunicaciones reales
   - interrupción de procesos internos
   - pérdida de confianza en el sistema

d) **Privacidad y datos sensibles:** Los textos analizados pueden contener:
   - información personal
   - datos confidenciales
   - comunicaciones internas

   Riesgo: uso indebido, almacenamiento o exposición de datos.

e) **Opacidad (falta de explicabilidad):** Si el sistema no explica sus decisiones:
   - no se puede auditar,
   - no se puede corregir,
   - no se puede justificar ante usuarios.

Principios éticos aplicados al sistema

Un sistema de Procesamiento de Lenguaje Natural PLN en ciberseguridad debe cumplir al menos con:

a) **Transparencia (Explainability & Interpretability):** el sistema debe explicar sus decisiones. La transparencia implica que el sistema no opere como una "caja negra", sino que proporcione mecanismos de explicabilidad (explainability) y trazabilidad de decisiones. En el contexto de PLN aplicado a ciberseguridad, esto se traduce en:
   - Generación de justificaciones interpretables sobre por qué un mensaje fue clasificado como malicioso (por ejemplo, identificación de patrones semánticos, embeddings o tokens críticos).
   - Uso de técnicas como model-agnostic explanations (LIME, SHAP) o arquitecturas intrínsecamente interpretables.
   - Registro de decisiones en logs estructurados, accesibles para auditoría.

Desde un enfoque normativo, la transparencia se vincula con el derecho a la explicación en sistemas automatizados (Floridi et al., 2018; European Commission, 2021).

b) **Responsabilidad (Accountability & Human Oversight):** las decisiones finales deben poderse auditar por seres humanos. La responsabilidad implica que las decisiones del sistema sean atribuibles, auditables y reversibles, evitando la delegación absoluta en la automatización.
    - Implementación de un modelo human-in-the-loop (HITL) o human-on-the-loop (HOTL), donde un operador pueda validar o corregir decisiones críticas.
    - Definición de líneas claras de responsabilidad institucional: quién responde ante errores del sistema.
    - Mecanismos de auditoría periódica del modelo (evaluación de sesgos, precisión, tasas de error).
    - Versionamiento del modelo y de los datasets para garantizar reproducibilidad.

    Este principio se alinea con marcos como el AI Act de la Unión Europea (2024), que exige supervisión humana en sistemas de alto riesgo.

c) **Minimización de daño (Non-maleficence & Fairness):** evitar bloquear o penalizar usuarios legítimos sin justificación. El sistema debe priorizar la reducción de impactos negativos, especialmente en contextos donde errores pueden afectar derechos digitales.
    - Minimización de falsos positivos (usuarios legítimos clasificados como amenazas), mediante calibración de umbrales y validación cruzada.
    - Evaluación de sesgos algorítmicos (por idioma, jerga, contexto cultural), especialmente relevante en PLN multilingüe o regional.
    - Implementación de políticas de gradualidad en la respuesta: advertencias antes de bloqueos definitivos.
    - Uso de métricas éticas como Equal Opportunity o False Positive Rate Balance.

    Este principio se relaciona con el enfoque de AI for Social Good y con la noción bioética de no maleficencia (Jobin, Ienca & Vayena, 2019).

d) **Protección de datos (Privacy & Data Governance):** no almacenar más información de la necesaria. La protección de datos implica cumplir con principios de privacidad desde el diseño (Privacy by Design) y minimización de datos.

- Recolección únicamente de datos estrictamente necesarios para la tarea (principio de data minimization).
- Anonimización o seudonimización de datos sensibles antes del procesamiento.
- Implementación de técnicas como differential privacy o federated learning cuando sea posible.
- Definición de políticas claras de retención y eliminación de datos.
- Cumplimiento de normativas como GDPR o equivalentes locales.

Este principio es central en la gobernanza de sistemas de IA y en la protección de derechos fundamentales (Cavoukian, 2010).

e) **Robustez y seguridad (Robustness & Adversarial Resistance):** Resistencia frente a ataques adversariales (ej. prompt injection, evasión semántica). Evaluación continua ante inputs manipulados.

f) **Justicia y no discriminación (Fairness):** Evitar discriminación indirecta por lenguaje, dialecto o contexto sociocultural. Validación en datasets diversos.

g) **Gobernanza y mejora continua: Monitoreo post-despliegue (model drift, concept drift).** Actualización periódica del modelo y sus políticas éticas.

Buenas prácticas técnicas

Un sistema inteligente a más de detectar amenazas, reduce errores injustificados y respeta el contexto humano. Para reducir riesgos éticos, es recomendable seguir los siguientes consejos:

- Usar datasets balanceados y diversos,
- Validar con casos reales y ambiguos
- Incluir explicabilidad en cada decisión
- Ajustar umbrales para reducir falsos positivos
- Mantener supervisión humana (human-in-the-loop).

En resumen, en esta semana 7, se comprende que el sistema además de detectar amenazas, sabemos:

- Cuándo confiar en él
- Cuándo cuestionarlo
- Cómo mejorarlo.

En esta semana el pipeline se complementa con lo aprendido en las semanas anteriores:

detección → interpretación → validación → responsabilidad

Ejemplo de control ético usando código Python:

Ilustración 7 es un ejemplo para evitar decisiones automáticas débiles. Nota: necesita las funciones definidas en el recurso de profundización de la semana 6.

```python
def decision_final(texto):
    resultado = sistema_hibrido(texto)
    explicacion = explicar_decision(texto)

    if resultado == "posible_ataque" and len(explicacion) < 2:
        return "requiere_revision_humana"

    return resultado
```

Ilustración 7. Control ético en Python (creación de autor Rafael Melgarejo)

Glosario:

- **Eisenstein, J. (2019).** Introduction to natural language processing. MIT Press. https://cdn.jsdelivr.net/gh/it-ebooks-0/it-ebooks-2018-04to07/Natural%20Language%20Processing%20%28Jacob%20Eisenstein%29.pdf
- **Jurafsky, D., & Martin, J. H. (2023).** Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition (3rd ed., draft). Stanford University. https://web.stanford.edu/~jurafsky/slp3/.
- **Cavoukian, A. (2010).** Privacy by Design: The 7 Foundational Principles. Information and Privacy Commissioner of Ontario.
- **European Commission. (2021).** Ethics Guidelines for Trustworthy AI.
- **European Union. (2024).** Artificial Intelligence Act.
- **Floridi, L., et al. (2018).** AI4People—An ethical framework for a good AI society. Minds and Machines, 28(4), 689–707.
- **Jobin, A., Ienca, M., & Vayena, E. (2019).** The global landscape of AI ethics guidelines. Nature Machine Intelligence, 1, 389–399.
- **Mittelstadt, B. D. (2019).** Principles alone cannot guarantee ethical AI. Nature Machine Intelligence, 1, 501–507.

PUCE Virtual

síguenos en:

www.pucevirtual.puce.edu.ec