Principles of Explanatory Debugging to Personalize Interactive Machine Learning

Autores

Todd Kulesza (Oregon State University)

Margaret Burnett (Oregon State University)

Weng-Keen Wong (Oregon State University)

Simone Stumpf (City University London)

Presentadores

Camilo Ruiz-Tagle Molina

Víctor Gálvez Yanjarí

Contenido

1. Introducción
2. Principios del Explanatory Debugging
3. EluciDebug
4. Principios en EluciDebug
5. Evaluación
6. Resultados
7. Conclusiones

Introducción

Introducción

· El paper presenta la **depuración explicativa:**

enfoque en que el **sistema explica a los usuarios** cómo hizo las predicciones, y el **usuario explica al sistema** las correcciones necesarias.

· Principios del enfoque y un prototipo de instanciación.

· En experimento:

- **Aumentó el entendimiento** de los participantes del sistema de aprendizaje en un 52%.
- Permitió que feedback fuera el **doble de eficiente** que un sistema de aprendizaje tradicional.

Introducción

Motivación:

¿cómo pueden los usuarios, eficientemente y efectivamente personalizar las predicciones o recomendaciones que los sistemas de aprendizaje hacen en su nombre?

Introducción

Ejemplo: almacenamiento automático de correos electrónicos en carpetas temáticas en función del contenido del mail. ¿Cómo entrenar al sistema?

Solución propuesta:

· Explanatory Debugging, que permite personalizar de manera efectiva y eficiente los sistemas de aprendizaje automático.

· ¿Depuración?: usuario está tratando de ejercer un control detallado sobre el comportamiento aprendido del sistema.

Introducción

Explanatory Debugging:

1. *El sistema explica al usuario las razones de sus predicciones.*
2. *El usuario explica las correcciones al sistema.*

En el ejemplo:

1. *El sistema muestra al usuario todas las palabras que utiliza para identificar un grupo temático de mail.*
2. *El usuario hace las correcciones al sistema, agregando o removiendo palabras que son irrelevantes.*

Introducción

- **EluciDebug**: primer sistema interactivo diseñado para ayudar a los usuarios finales a construir modelos mentales útiles, donde los usuarios pueden explicar las correcciones de vuelta al sistema.

- *Sistemas de caja blanca, porque ayudan a los usuarios a **entender cómo funciona el sistema**, para que pueda ser personalizado de mejor forma.*

Principios de Explanatory Debugging

Principio 1: Explicabilidad

Explicar con precisión las razones del sistema de aprendizaje para cada predicción al usuario final.

1: Be Sound.

- Las explicaciones no deben simplificarse explicando el modelo como si fuera menos complejo de lo que realmente es.

2: Be Complete.

- Una explicación completa no omite información importante sobre el modelo

Principio 1: Explicabilidad

Explicar con precisión las razones del sistema de aprendizaje para cada predicción al usuario final.

3: But Don’t Overwhelm.

- Necesidad de seguir siendo comprensibles y de atraer la atención del usuario.

Principio 2: Correctability

Permite a los usuarios explicar las correcciones al sistema de aprendizaje.

1: Be Actionable.

- Los usuarios finales ignorarán las explicaciones cuando no esté claro el beneficio de atenderlas.

2: Be Reversible.

- Ser capaz de revertir fácilmente una acción dañina.

Principio 2: Correctability

Permite a los usuarios explicar las correcciones al sistema de aprendizaje.

3: Always Honor User Feedback.

- Un sistema que parece ignorar los comentarios de los usuarios disuade a los usuarios de continuar proporcionando comentarios.

4: Incremental Changes Matter.

- Los usuarios puedan ver cambios incrementales en el razonamiento del sistema de aprendizaje después de cada interacción

EluciDebug

EluciDebug

Tarea: **clasificación de texto**. ¿Por qué?

(1) **muchos sistemas del mundo real lo requieren** (por ejemplo, filtrado de SPAM, recomendación de noticias, publicación de anuncios relevantes, clasificación de resultados de búsqueda, etc.)

(2) se puede evaluar con **documentos sobre temas comunes** (por ejemplo, deportes populares), permitiendo una gran cantidad de participantes para la evaluación.

EluciDebug

Diseño del prototipo: similar a programa de correo electrónico con múltiples carpetas. Estructura:

(A) Lista de carpetas.

(B) Lista de correos en la carpeta seleccionada.

(C) El mensaje seleccionado.

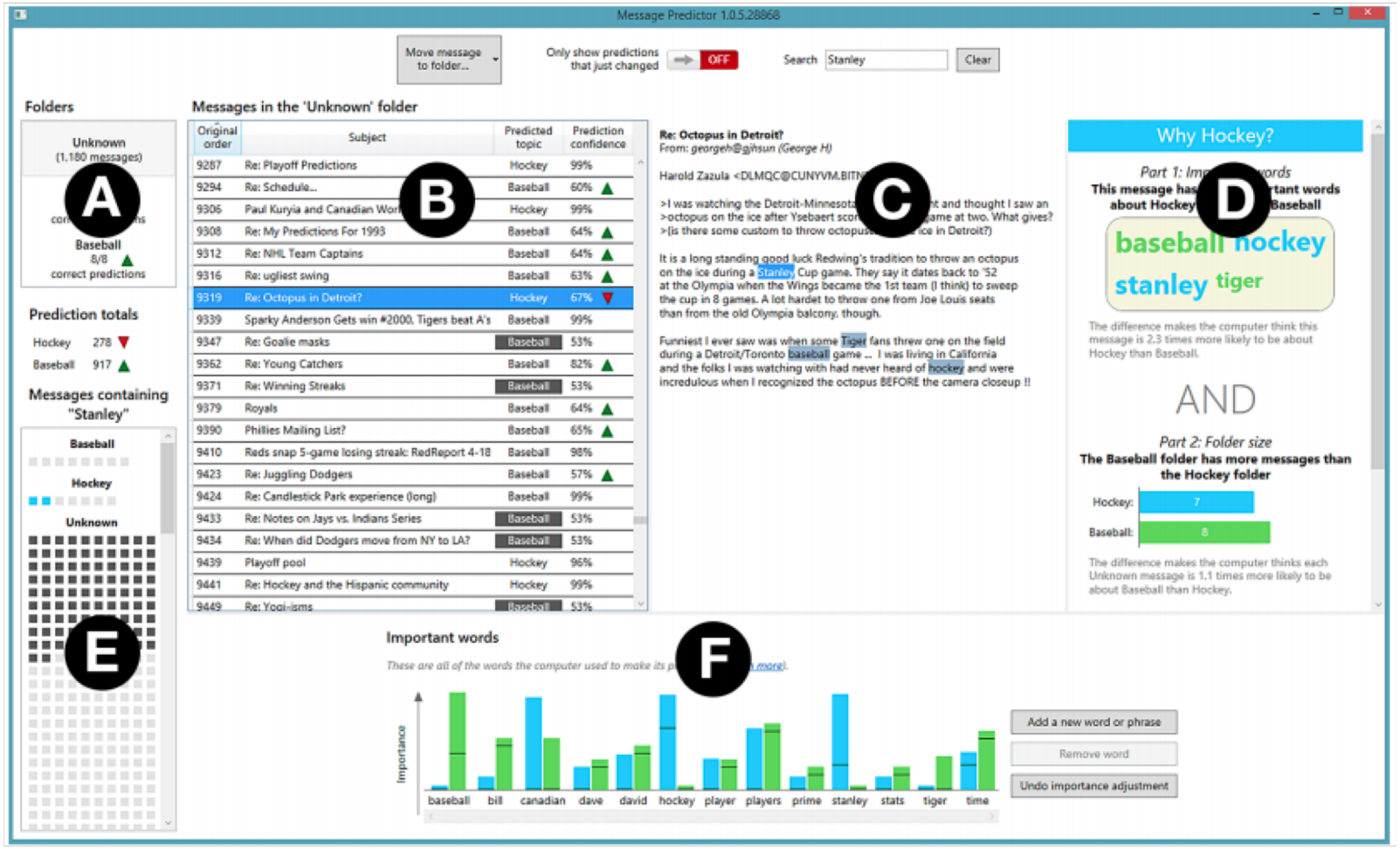
(D) Explicación de los mensajes predichos de la carpeta.

(E) Vista de cuáles mensajes contienen la palabra seleccionada.

(F) Lista completa de palabras que el sistema usa para hacer las predicciones.

EluciDebug

Prototipo EluciDebug



Clasificador
Multinomial Naive Bayes

Clasificador MNB

$$\Pr(c|d_i) = \frac{\Pr(c)\Pr(d_i|c)}{\Pr(d_i)} \qquad (1)$$

- Probabilidad de que un documento que se está clasificando, pertenezca a una salida (etiquetas disponibles).
- c : clase individual en la colección de posibles clases de salida.
- d_i : documento individual para clasificar.

Clasificador MNB

$$\Pr(c|d_i) = \frac{\Pr(c)\Pr(d_i|c)}{\Pr(d_i)} \qquad (1)$$

- La salida con **probabilidad más alta "gana"** y se convierte en la etiqueta predicha para la entrada.
- **Ejemplo:** si MNB calcula que un documento tiene un 70% de probabilidad de ser correo no deseado y un 30% de probabilidad de no ser correo no deseado, el documento se etiquetará como correo no deseado.

Clasificador MNB

$$\Pr(d_i|c) = \prod_n \Pr(w_n|c)^{f_{ni}} \qquad (2)$$

- Pr (c): probabilidad de que cualquier documento dado pertenezca a la clase c. Se puede estimar dividiendo el número de documentos en c por el número total de documentos en el conjunto de training set.
- Pr (d_i | c) representa la probabilidad de documentar la clase c dada.
- f_{ni} es el número de instancias de la palabra n en el documento d_i.
- Pr (w_n | c) es la probabilidad de la palabra n dada clase C.

Clasificador MNB

$$\Pr(w_n|c) = \frac{p_{nc} + F_{nc}}{\sum_{x=1}^{N} p_{xc} + \sum_{x=1}^{N} F_{xc}} \qquad (3)$$

- Pr (w_n | c) es la probabilidad de la palabra n dada clase C
- N es el número de palabras únicas en los documentos de training para todas las clases.
- p_{nc} es un término de smoothing (generalmente 1) para evitar que la ecuación arroje 0 si no hay documentos de la clase c que contengan la palabra w_n.

Principios en EluciDebug

Principios en EluciDebug

I. Explicabilidad

a. Be iterative

b. Be sound

c. Be complete

d. Not overhelming

II. Correctability

a. Be actionable

b. Be reversible

c. Honor user feedback

d. Incremental changes matter

Principio: Explicabilidad

a. Be iterative

- 2 estrategias para construir modelos mentales en forma iterativa:
  - Cada explicación se enfoca en un aspecto individual del learning system.
  - Explicaciones por capas disponibles para el usuario.
- Se puede obtener la explicación de **porqué** una palabra está asociada con otra.

Principio: Explicabilidad

a. Be iterative

- Con el tiempo, el usuario puede aprender sobre el funcionamiento general mediante la observación de puntos en común en las razones para estas predicciones específicas.

Principio: Explicabilidad

b. Be Sound.

- Todo lo que dice una explicación es **verdadero**.
- Se revelan **features y forma** en que estas afectan la predicción del clasificador.
- EluciDebug's **Why explanation**: responsable de la comunicación de gran parte de esta información.

Principio: Explicabilidad

Principio: Explicabilidad

c. Be Complete.

- Explicar **cada término** de la ecuación y todas las f**uentes de información.**
- **Diagramación** de la explicación está pensada para la entrega de información.
- Contenido del modelo en explicación *why*.
- El fin es **impactar en modelo mental** de usuarios.

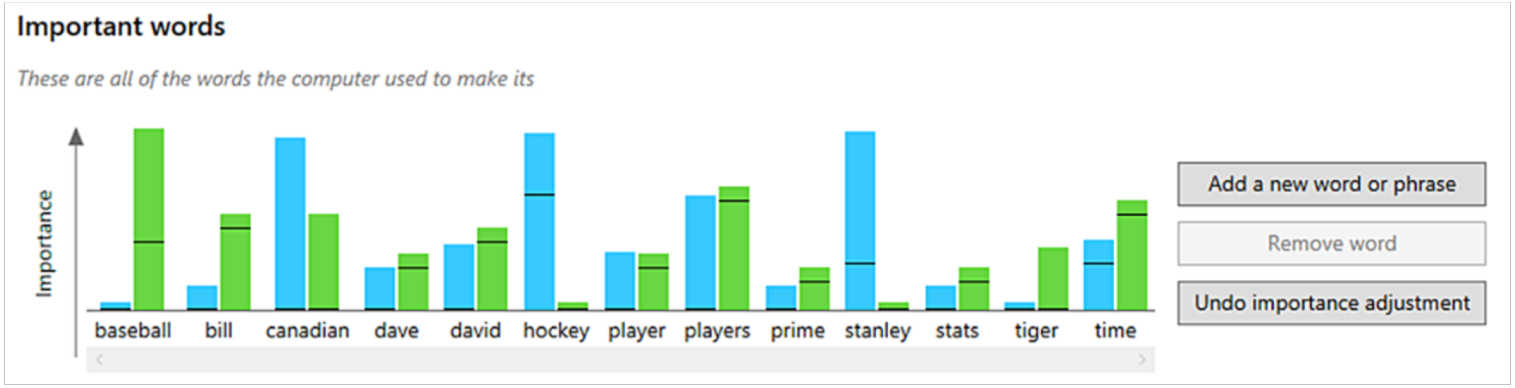
Principio: Explicabilidad

d. Not overhelming.

- EluciDebug limita el conjunto inicial de features disponibles para el clasificador utilizando **information gain**.
- El clasificador selecciona automáticamente las **10 features con mayor information gain.**
- El usuario puede especificar lo contrario **agregando o eliminando** features.

Principio: Correctability

a. Actionability



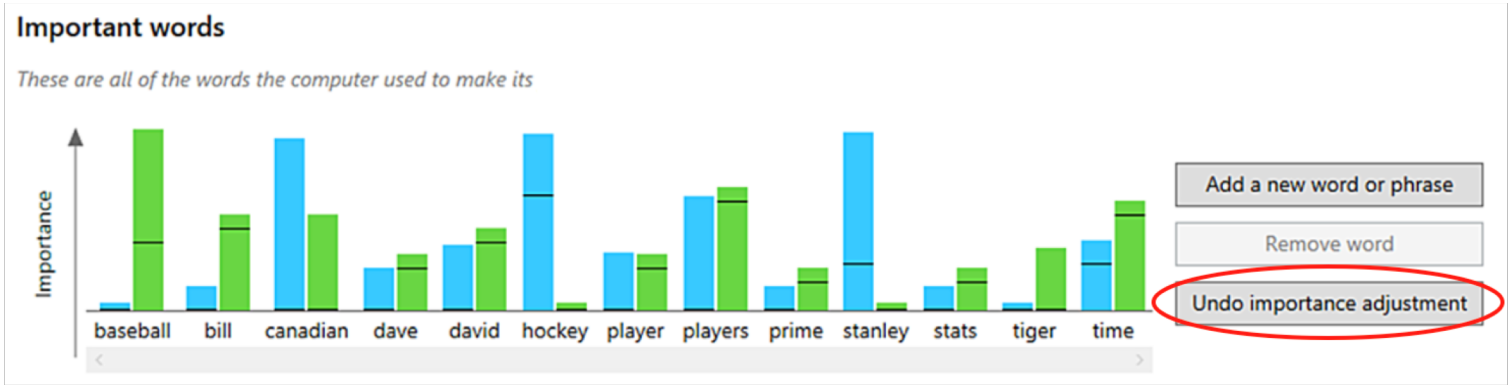
· Única sección accionable.

· Permite eliminar o agregar palabras al feature set.

· Permite ajustar la importancia de cada palabra.

Principio: Correctability

b. Reversibility



· Siempre se puede volver a un estado inicial.

Principio: Correctability

c. Honor User Feedback

· **Instance-based feedback**:

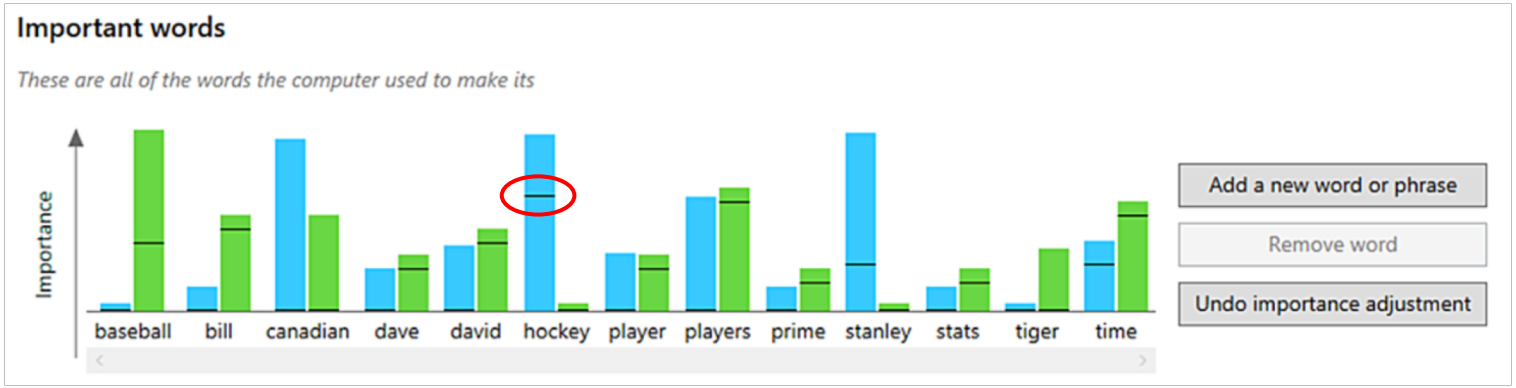
- Se etiqueta un ítem completo.
- Etiquetado por usuario, se va a training set.

Principio: Correctability

c. Honor User Feedback

· **Feature-based feedback**:

- Se etiqueta un ítem siguiendo una forma determinada.



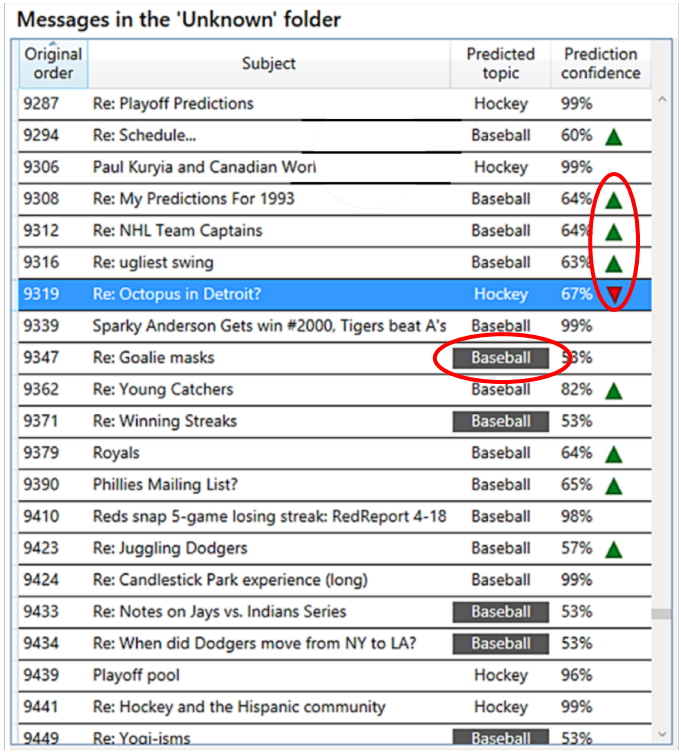
· Se varía la probabilidad de que la palabra pertenezca a una clase (p_{nc})

Principio: Correctability

d. Revelar cambios incrementales

· Flechas indican variación en la confianza de la predicción.

· Fondo gris indica que cambió la clasificación.



Messages in the 'Unknown' folder

Original order	Subject	Predicted topic	Prediction confidence
9287	Re: Playoff Predictions	Hockey	99%
9294	Re: Schedule...	Baseball	60% ▲
9306	Paul Kuryia and Canadian Worl	Hockey	99%
9308	Re: My Predictions For 1993	Baseball	64% ▲
9312	Re: NHL Team Captains	Baseball	64% ▲
9316	Re: ugliest swing	Baseball	63% ▲
9319	Re: Octopus in Detroit?	Hockey	67% ▼
9339	Sparky Anderson Gets win #2000, Tigers beat A's	Baseball	99%
9347	Re: Goalie masks	Baseball	53%
9362	Re: Young Catchers	Baseball	82% ▲
9371	Re: Winning Streaks	Baseball	53%
9379	Royals	Baseball	64% ▲
9390	Phillies Mailing List?	Baseball	65% ▲
9410	Reds snap 5-game losing streak: RedReport 4-18	Baseball	98%
9423	Re: Juggling Dodgers	Baseball	57% ▲
9424	Re: Candlestick Park experience (long)	Baseball	99%
9433	Re: Notes on Jays vs. Indians Series	Baseball	53%
9434	Re: When did Dodgers move from NY to LA?	Baseball	53%
9439	Playoff pool	Hockey	96%
9441	Re: Hockey and the Hispanic community	Hockey	99%
9449	Re: Yogi-isms	Baseball	53%

Evaluación de EluciDebug

Preguntas de Investigación

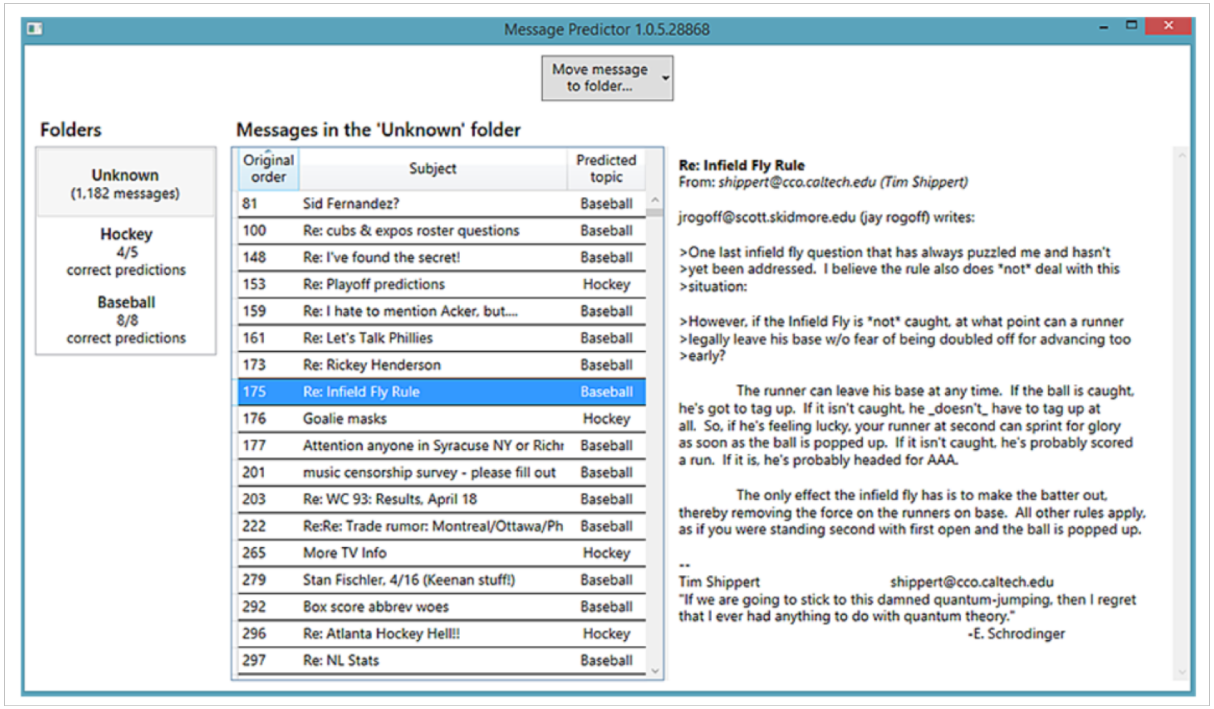
1. ¿Explanatory Debugging ayuda a los usuarios a personalizar un clasificador más eficiente que el *instance labeling*? (***efficient***)

2. ¿Explanatory Debugging ayuda a los usuarios a personalizar un clasificador más preciso que el *instance labeling*? (***accurate***)

3. ¿Explanatory Debugging ayuda a usuarios a construir ***mejores modelos mentales*** que las clásicas cajas negras?

Diseño de experimento

- **2 grupos**:
  - **Control**: sin feature-based feedback.



  - **Tratamiento**: usan EluciDebug.

Diseño de experimento

· **Dataset**: hockey y baseball, extraído de 20 Newsgroups.

· Inicia con **10 mensajes**: 5 en carpeta Hockey y 5 en carpeta Baseball.

· **77 participantes**: 37 grupo control y 40 grupo tratamiento.

· Instrucción: Haga las predicciones del computador **lo más precisa posible** en 30 minutos.

· **Cuestionario** para evaluar modelo mental, features del prototipo y carga de trabajo.

Diseño de experimento

- **Experimento offline:**

  Uso de HighIG features and Comprehensive features.

  - **Grupo control**: recálculo de HighIG y se mantienen las 10 features con HighIG.

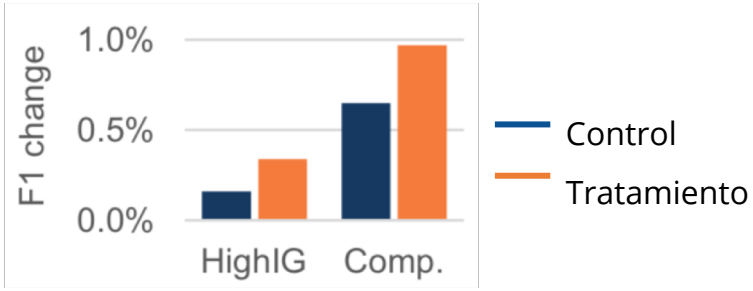
  - **Grupo tratamiento**: no hay recálculo de HighIG, usuario varía las features.

Resultados

Resultados

· Grupo tratamiento **da menos feedback** que grupo control.

· Grupo control suelen examinar **más mensajes**.

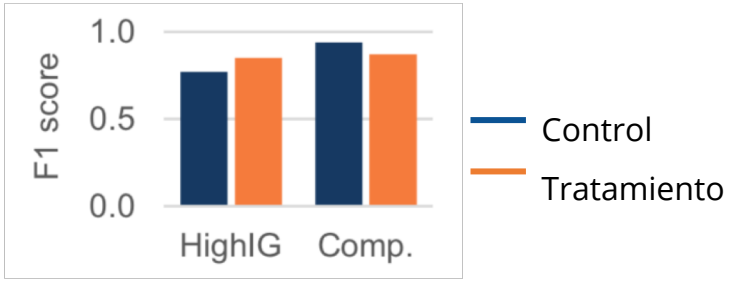


· Acciones de grupo tratamiento son más eficientes que las del grupo control (más aún en Comprehensive features set)

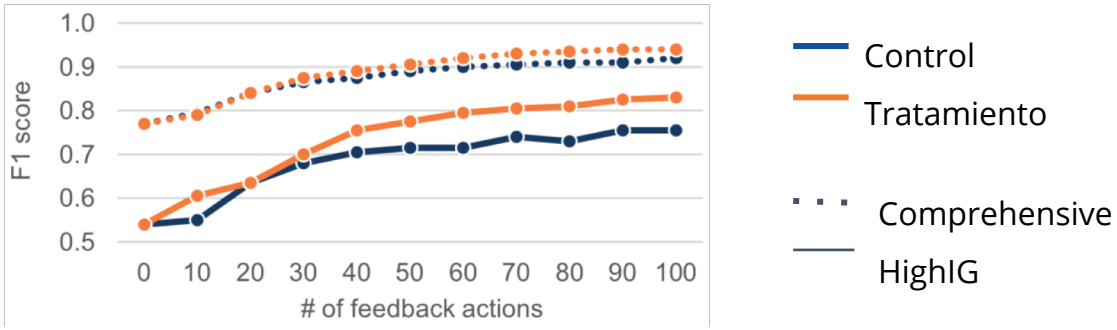
Resultados

· Grupo tratamiento hizo clasificadores **10% más precisos**.

· **Mayor F1 en grupo control**, al usar Comprehensive Feature Set.



· Grupo control **nunca supera** al grupo tratamiento.



Resultados

· Grupo **tratamiento:**

- satisfecho con su variante.
- no sintió que fuera más trabajoso.
- mejores modelos mentales.

· **Comentarios** grupo control:

"Sería más fácil si supiera cómo esto hace las predicciones."

"Fueron unos largos 30 minutos."

· **Comentarios** grupo tratamiento:

"Fue muy fácil aumentar la precisión."

"Es tan simple que mis padres podrían usarlo."

Resultados

· 1ª y 2ª pregunta de investigación:

- Explanatory Debugging **es eficiente**, pero no siempre el más preciso
- **Ventajas** de Explanatory Debugging:
    a. No necesita un set grande para entrenarse.
    b. Rápidas mejoras de precisión.

· 3ª pregunta de investigación:

- Explanatory Debugging construye **modelos mentales útiles**, sin aumentar la carga de trabajo.

Conclusiones

Conclusiones

· Uso de Explanatory Debugging permite entender cómo opera el modelo.

· A los usuarios les gustó EluciDebug.

· Trabajo futuro:

- Impacto de interacciones de usuario en grandes períodos de tiempo.

- ¿usuarios pueden transferir el conocimiento de un learning system para personalizar rápidamente sistemas similares?

Principles of Explanatory Debugging to Personalize Interactive Machine Learning

Autores

Todd Kulesza (Oregon State University)

Margaret Burnett (Oregon State University)

Weng-Keen Wong (Oregon State University)

Simone Stumpf (City University London)

Presentadores

Camilo Ruiz-Tagle Molina

Víctor Gálvez Yanjarí