Trabajo Fin de Máster

Modelos para el soporte de la evaluación de competencias de argumentación

Ángel Rodríguez Marco

Directora: Ángeles Manjarrés Riesco

UNED

E.T.S. de Ingeniería Informática

Máster en Inteligencia Artificial Avanzada: Fundamentos, Métodos y Aplicaciones

Índice general

Resumen

Este trabajo es una aportación al uso de técnicas de Inteligencia Artificial como herramienta educativa en el desarrollo de la competencia en argumentación. Con este fin, y centrados únicamente en ensayos persuasivos, se ha identificado una serie de atributos del texto argumentativo que se pueden asociar parcialmente a indicadores de la competencia en argumentación. Se ha utilizado un corpus de ensayos para el aprendizaje de estructura y probabilidades condicionadas de un modelo bayesiano de estos indicadores, así como para la definición de un algoritmo de resumen y visualización de argumentos utilizados en un grupo de ensayos persuasivos. Como apoyo a estas tareas, se han entrenado y validado varios modelos de clasificación para extraer algunos de los atributos utilizados (coherencia, estilo) y ayudar a resumir argumentos (paráfrasis).

Los resultados de validación del modelo bayesiano de indicadores y los clasificadores son buenos, y el algoritmo de resumen y visualización presenta resultados prometedores, aunque hay limitaciones y puntos abiertos que discutimos en el estudio. Estos modelos y algoritmo podrían utilizarse como base de una sistema recomendador educativo, que, a partir de ensayos argumentativos escritos por un grupo de estudiantes, recomiende, por un lado, acciones individuales de mejora de las competencias de argumentación y, por otro, proporcione soporte a una revisión crítica dentro del grupo de estudiantes e instructor de los argumentos más utilizados. El estudio y desarrollo de esta aplicación se deja para un trabajo futuro.

1. Introducción

1.1. Contexto y motivación

En un debate, diferentes personas argumentan utilizando lenguaje natural: intercambian ideas, las atacan o defienden e intentan persuadir al grupo de que su propuesta es la correcta o la más adecuada. Los argumentos pueden incluir inferencias lógicas, junto con apelaciones a emociones y percepciones colectivas, así como recursos dialécticos, retóricos y estilísticos (incluyendo y sin limitarse a ironía, sarcasmo, ataques personales y falacias).

Cuando se estudia la argumentación, se puede considerar el debate, intercambio de argumentos entre varios actores, o una parte más reducida, la expresión de argumentos por parte de unos de los actores. Dada la extensión y complejidad del tema, en este trabajo se considera exclusivamente la exposición de argumentos por parte de una persona por escrito en un ensayo persuasivo o argumentativo, sin considerar que pueda formar parte de un debate más completo.

Desde el punto de vista de la Inteligencia Artificial (IA) y el Procesamiento de Lenguaje Natural (PLN), estudiar modelos de debate y argumentación, que son funciones de alto nivel del lenguaje humano, tiene un interés evidente (como ejemplo, muestras recientes [131] y [5]), y tareas directamente relacionadas, como la minería de argumentos y el Procesamiento de Lógica Natural, pueden tener aplicaciones directas en la educación, estudios de opinión y mercados y otras áreas.

Dentro del área de la educación, el desarrollo y evaluación de competencias de debate y argumentación tiene una gran importancia. El proyecto Tuning [13], cuyo objetivo era armonizar las estructuras educativas universitarias de Europa para la adopción de un sistema de titulaciones fácilmente reconocibles y comparables, estableció las competencias genéricas y las específicas de diferentes disciplinas para definir puntos de referencia para la elaboración y evaluación de planes de estudio. Tanto en las competencias genéricas como específicas definidas en Tuning, se incluyen competencias transversales de argumentación así como otras competencias relacionadas: capacidad de comunicación oral y escrita, capacidad de abstracción, análisis y síntesis, capacidad crítica y autocrítica y capacidad de razonamiento lógico, entre otras, junto con competencias ético-cívicas que se desarrollan con la práctica de debates: habilidades interpersonales, valoración y respeto por la diversidad y multiculturalidad...

También en el campo de la educación, la comunidad académica muestra interés en identificar las características que indican la competencia en argumentación y

debate, y en definir rúbricas para su evaluación. Por ejemplo, en [82], las autoras proponen diferentes criterios de alto nivel para entender cómo un estudiante utiliza la evidencia para construir argumentos, cómo entiende la metodología científica (por ejemplo, observación y estudio estadístico de datos) para elegir posiciones, así como su inclinación a considerar ideas alternativas.

Dentro del área de la tecnología educativa aplicada al desarrollo de competencias de debate y argumentación, se han creado sistemas informáticos y metodologías para ayudar a los estudiantes a, por un lado, desarrollar argumentos más persuasivos, con bases objetivas y menos puntos abiertos que se puedan atacar, así como a estimar la validez de ideas y argumentos presentados por otras personas. Gran parte de ellos son aplicaciones de soporte al debate o la construcción de argumentos en grupos de estudiantes, y no utilizan (o, por lo menos, no de manera significativa) técnicas de IA. Aquí podríamos destacar Carneades ([45]) o el entorno DUNES[17].

Hay también sistemas sofisticados, basados en diferentes estudios teóricos sobre la argumentación que ayudan a construir buenos mapas de argumentación de forma manual (por ejemplo, Araucaria, OVA [100]), o que permiten evaluar la fuerza de un argumento y, en un debate, quien tiene la carga de la prueba (Carneades [45], orientado a argumentos legales). Estos sistemas pueden ser eficientes y tener un buen fundamento teórico, pero, como los anteriores, se limitan a dar soporte a las entradas manuales por parte de una o varias personas.

Además de estos sistemas, se han creado ontologías o taxonomías de argumentación (por ejemplo, AIF [95], DILIGENT, [117], esta última creada para apoyar debates sobre definición de ontologías en la web semántica) que pueden utilizarse como una base para caracterizar una argumentación.

Al revisar el estado del arte, no se han encontrado referencias a la utilización en sistemas informáticos educativos de tecnologías que, de forma automática, puedan caracterizar el texto de un debate, inferir los argumentos utilizados y ofrecer pautas para evaluarlos y mejorarlos. Opinamos que el principal obstáculo hasta ahora era la complejidad de identificar sentencias argumentativas en texto real, así como las relaciones entre estas sentencias (esto es una tarea de minería de argumentos). Otro problema relacionada es la estimación de relaciones de inferencia entre argumentos (relacionada con el Procesamiento de Lógica Natural) o la definición de mejores modelos y algoritmos para describir la semántica de los argumentos. Aunque importante, en nuestra opinión este problema es menos crítico que el anterior a la hora de crear sistemas de IA que se puedan utilizar en aplicaciones educativas o de otro tipo.

Sin embargo, en los últimos años ha habido avances en minería de argumentos (véase [111],[78]) que permiten hacer algunas propuestas de implementación de sistemas de aprendizaje automático que potencialmente se pueden aplicar al desarrollo de herramientas educativas en competencias de argumentación. Así, estas herramientas podrían ofrecer a estudiantes e instructores información casi en tiempo real sobre los argumentos utilizados y recomendaciones para mejorarlos. Este tipo de guía posible-

mente no puede ser prescriptiva, pero ayudaría a dirigir de forma más efectiva una discusión o actividad educativa, particularmente en escenarios de enseñanza online (véase por ejemplo [54]). El presente trabajo se centra en los modelos de aprendizaje automático, pero no discutiremos, debido a la complejidad y extensión del tema, su adaptación a un sistema tecnológico de soporte educativo para la competencia en argumentación.

Hacemos notar también que en el área de Lógica Natural ha habido progresos recientes (véase [18], [85], y, para más detalle, las referencias listadas en StanfordNLP Natural Logic), pero no se aplicarán en este estudio, aunque pueden ser interesantes en trabajos futuros.

1.2. Hipótesis y Objetivos

1.2.1. Objetivos

Este trabajo tiene como objetivo principal generar un modelo de indicadores de la competencia en argumentación de ensayos persuasivos, basado en aprendizaje automático a partir de texto y que potencialmente pueda ayudar a estudiantes e instructores a evaluar y entender la calidad de un ensayo y obtener recomendaciones para mejorar las competencias en argumentación.

Como objetivo secundario, se hará una primera aproximación al problema de definir un algoritmo de aprendizaje no supervisado para resumir y visualizar los argumentos utilizados en varios ensayos y que pueda servir de guía para una discusión en grupo de estudiantes e instructor sobre los argumentos utilizados.

El modelo de indicadores de la competencia en argumentación ha de ser lo suficientemente expresivo como para poder proporcionar información nueva y relevante a los usuarios. La respuesta del modelo debería ser fácil de entender para estudiantes e instructores, con el fin de que puedan discutir estas respuestas o crear planes de acción a partir de ellas. De forma similar, el algoritmo de visualización de argumentos debe ayudar a que emerja información relevante y permitir articular una discusión sobre calidad de los argumentos y diferentes técnicas de argumentación que se han utilizado o podrían utilizarse.

Los objetivos parciales para alcanzar el objetivo general son:

1. Identificar una serie de atributos del ensayo argumentativo que se pueden asociar a indicadores de la competencia en argumentación o competencias relacionadas. Estos indicadores se pueden relacionar con conceptos manejados frecuentemente en el dominio educativo, de manera que se puede esperar que un instructor esté familiarizado con ellos.
    a) Una guía para elegir atributos es encontrar aproximaciones a los criterios estándar de pensamiento crítico de Elder y Paul [35]. Una asunción gene-

ral que hacemos es que las competencias en pensamiento crítico engloban en general a la competencia en argumentación.

2. Identificar datos de entrenamiento. El corpus de ensayos argumentativos no necesita una anotación detallada, solo una o varias evaluaciones de la calidad del ensayo que se integrarán en el modelo de indicadores de la competencia en argumentación.
3. Entrenar y validar una red bayesiana como modelo de indicadores de la competencia en argumentación a partir de los atributos del primer punto y entrenando sobre textos de ensayos reales que incluirán evaluaciones por parte de instructores humanos. La red aprenderá la estructura y probabilidades condicionadas a partir de los atributos generados en texto real y las calificaciones anotadas por instructores. El modelo no necesita ajustes manuales.
4. Definir un algoritmo no supervisado para resumir los argumentos más frecuentes en un corpus, una medida de su popularidad y la fuerza esperada de las relaciones entre cláusulas argumentativas. Se utilizará un grafo para representar las estructuras argumentales más comunes y nos referiremos a este grafo como grafo de argumentación colaborativo, ya que resume la aportación de varios autores. Como primera aproximación, este algoritmo se validará únicamente de forma cualitativa y se indicarán puntos de mejora para aumentar su efectividad.
5. Mostrar cómo se puede extraer información útil para estudiantes e instructores a partir del modelo bayesiano y el grafo resumen de argumentos utilizados:
   a) El modelo de indicadores de la competencia en argumentación es inteligible: una persona puede entender cuáles son los factores relevantes.
   b) Ayuda a estimar qué características de un ensayo puede ser más eficiente mejorar para mejorar su calificación.
   c) Permite visualizar la estructura de argumentación más frecuente en un conjunto de ensayos, así como la fuerza de cada uno de ellos.

Se debe destacar que, aunque durante este trabajo se realiza inferencia de calificaciones o evaluaciones de los ensayos, desarrollar un sistema de calificación automática («Automated Essay Scoring» AES) no es parte de los objetivos. El objetivo general es crear una serie de modelos y algoritmos que potencialmente se puedan utilizar en un sistema educativo y de recomendaciones para el desarrollo de competencias en argumentación. Un sistema puro AES presenta una serie de problemas éticos que no se pueden ignorar (véase, por ejemplo,[15]).

1.2.2. Hipótesis

El estudio asume las siguientes hipótesis, que se verificarán a partir de los cálculos de este trabajo.

Hipótesis 1 Se puede construir un número relativamente pequeño de atributos para caracterizar parcialmente las competencias de argumentación. Estos atributos incluyen métricas definidas «a priori» que se pueden extraer del texto argumentativo e indicadores definidos implícitamente por anotaciones de evaluadores humanos en el corpus de entrenamiento. Estos atributos se pueden utilizar en el aprendizaje de una red bayesiana que permite inferir el valor de estos indicadores a partir de datos no utilizados en el entrenamiento. La capacidad de generalizar de este modelo justifica «a posteriori» la validez de los atributos.

Los atributos no cubren todas las posibles características de la competencia en argumentación, debido al tamaño y complejidad del problema que supone su aprendizaje por parte de un modelo de aprendizaje automático, pero al incluir una calificación por parte de evaluadores humanos, la red bayesiana permite generalizar criterios puramente humanos de evaluación de ensayos y extraer información más allá de los indicadores definidos «a priori».

Hipótesis 2 Al aplicar un marco retórico específico, en nuestro caso RST («Rethorical Structure Theory» [65]), se puede dividir un texto en fragmentos con una función retórica bien definida (EDU, «Elementary Discourse Units») y definir una serie de relaciones entre ellas. Siguiendo las ideas de la referencia [62], podemos medir la coherencia de un texto utilizando la frecuencia de ocurrencia de las diferentes relaciones retóricas. Esta métrica es útil como indicador de una característica de la competencia en argumentación.

Hipótesis 3 Se pueden definir una o varias métricas para caracterizar la diferencia de estilos en diferentes ensayos. A partir de la referencia [41] se aprenderá un modelo para estimar un indicador único del estilo de un ensayo.

Hipótesis 4 Es posible resumir los argumentos utilizados en un grupo de ensayos, utilizando la paráfrasis (expresión de las mismas ideas y conceptos de un texto utilizando una redacción diferente) como criterio para seleccionar representantes de cada cláusula argumentativa.

1.2.3. Principales aportaciones de este trabajo

En relación al estado del arte en el área de aplicaciones educativas para ayudar al desarrollo de competencias de la argumentación, consideramos que este trabajo hace una serie de aportaciones novedosas, apoyadas parcialmente en resultados previos:

- Aunque las redes bayesianas se han utilizado anteriormente para modelar competencias educativas en general (véase [60]), no se han encontrado referencias en las que se entrene la estructura de las redes a partir de datos exclusivamente, tal como se hace en este trabajo. La aproximación habitual es utilizar

conocimiento de expertos en el área para definir relaciones entre las variables del modelo. La ventaja del enfoque utilizado en este estudio es que el conocimiento experto se integra en los datos de entrenamiento directamente como atributos de los ensayos, y el modelo se aprende sobre atributos y anotaciones de expertos. Consideramos que de esta manera:

  - Se reduce el sesgo de predefinir manualmente la estructura del modelo.
  - Es más fácil incorporar conocimiento experto, ya que el evaluador o anotador no necesita entender las correlaciones o causalidad entre los atributos utilizados, sino que una calificación o evaluación es suficiente para poder incorporarla al entrenamiento.

- Se ha propuesto un algoritmo no supervisado de resumen y visualización de argumentos utilizados en un conjunto de ensayos, basado en un modelo entrenado para detectar paráfrasis (expresión en un texto de las ideas de otro texto, posiblemente en una forma diferente) y la generación de un grafo con los argumentos más frecuentes. Los nodos corresponden a grupos de cláusulas argumentativas que consideramos paráfrasis. Los nodos están conectados si sus cláusulas están enlazadas en alguno de los ensayos considerados. La proporción de enlaces entre representantes indica la fuerza del enlace entre nodos. Este algoritmo se utiliza para resumir y representar gráficamente los argumentos más utilizados en una muestra de ensayos. Aunque hay métodos de representación y evaluación gráfica de argumentos (por ejemplo Carneades, [45]) no se han encontrado referencias a representaciones gráficas basadas en agrupación o resumen de argumentos utilizando algoritmos no supervisados.

  - Respecto a la estimación de la fuerza de la relación entre grupos de cláusulas argumentativas, en la referencia [113], Stegmann, Weinberger y Fischer definen una métrica en un experimento para validar la mejora en competencias de argumentación para un grupo de estudiantes en dos experimentos controlados. Sin embargo, una finalidad en el presente trabajo es expresar de forma general la fuerza de un argumento aprendido sin supervisión, mientras que los autores en [113] miden una mejora de un grupo de ensayos respecto a una estructura argumental predefinida.
  - Se ha observado que, cualitativamente, el algoritmo de visualización proporciona información interesante, pero como punto para un trabajo futuro queda validarlo de forma más rigurosa, así como mejorar los métodos para agrupar cláusulas argumentativas.

- Utilizando una metodología similar a la usada en diferentes tareas de la estilometría ([76], [41]) se ha entrenado un modelo para estimar una indicador del estilo de un texto. La «estilometría» se centra en problemas de atribución de texto a autores, pero no se han podido encontrar referencias en las que se utilicen sus métodos para construir rasgos de un texto.
- Para calcular un indicador de coherencia de un texto este estudio ha seguido de cerca el método utilizado en la referencia [62]. Sin embargo, a diferencia

de esta referencia, en la que utilizaban un marco retórico PDTB [137], hemos utilizado el modelo RST [65] con buenos resultados.

1.2.4. Limitaciones al alcance del trabajo y asunciones

Al tener en cuenta los recursos actualmente disponibles (corpus y modelos pre-entrenados) y la complejidad o esfuerzo requerido por las tareas a completar, se limitó el alcance de este trabajo. Las limitaciones más relevantes se listan a continuación:

- El estudio se centra en ensayos argumentativos: la presentación por escrito (en general breve) de una serie de ideas con argumentos a favor y en contra. No se consideran debates o intercambio de argumentos entre personas.
- No se consideran reservas o ataques a argumentos que se quieran refutar, tanto en el modelo bayesiano como en el algoritmo de resumen y visualización. Esto debería ser posible con los modelos actuales [78], re-entrenando cuidadosamente en el corpus AAEC [109, 110], por ejemplo.
- Consideraremos únicamente texto en inglés, que tiene disponibles más modelos y corpus que otros idiomas, como el español. En principio sería posible re-entrenar algunos de los modelos disponibles en inglés para adaptarlos al español u otros lenguajes, pero para ello se necesita un corpus de texto argumentativo en español semejante al AAEC («Argument Annotated Essays Corpus») [109, 110] o CDCP («Cornell eRulemaking Corpus»), [78].
- No se tienen en cuenta posibles variaciones regionales del idioma. Los corpus que utilizamos tienen muestras de inglés británico, estadounidense y otros países.
- Los atributos utilizados en el modelo bayesiano solo asignan indicadores a parte de las características de la competencia en argumentación. Desde el principio se está reduciendo la capacidad del modelo bayesiano para aprender y expresar características de alto nivel, como por ejemplo la madurez en el razonamiento del estudiante, así como la granularidad de esta información.
- El modelo bayesiano de indicadores de la competencia en argumentación aprenderá a evaluar la calidad de un argumento a partir de una o dos calificaciones asignadas a cada ensayo por un grupo de instructores humanos. Aunque durante la generación de atributos se utiliza un modelo pre-entrenado de minería de argumentos, y este modelo se basa en un modelo conceptual de la argumentación ([86]), utilizaremos indicadores descriptivos de los argumentos encontrados (tipo y número de cláusulas argumentativas utilizadas en el ensayo), sin intentar definir la calidad de un argumento a partir de este modelo conceptual. Por ejemplo, no intentamos definir un atributo calidad de argumento a partir por ejemplo, del número de cláusulas argumentativas conectadas.

- El algoritmo de resumen y visualización infiere la fuerza de la relación entre algunos grupos de cláusulas argumentativas. Sin embargo está orientado a una discusión cualitativa en grupo, no a medir la fuerza de argumentos individuales.

Una asunción básica en este trabajo, que atañe a su utilidad en el desarrollo de sistemas educativos, es que los atributos seleccionados, el modelo bayesiano de indicadores de competencia en argumentación y el algoritmo de resumen y visualización se pueden utilizar en diferentes tipos de ensayo argumentativo, dentro de las limitaciones indicadas arriba. Esta asunción se apoya en que los atributos y modelos no dependen directamente de los datos de entrenamiento (por ejemplo, temas tratados o un vocabulario específico de los textos de entrenamiento), pero se pueden esperar limitaciones en la efectividad de los modelos para, por ejemplo, ensayos procedentes de grupos de estudiantes muy diferentes a los analizados. De cualquier manera, el proceso de generación de atributos y modelo se puede replicar, en principio, en cualquier grupo de ensayos persuasivos. Estudiar la validez de esta asunción está fuera del alcance de este estudio.

1.3. Estado del arte

1.3.1. Introducción

Esta sección revisa algunos conceptos claves del estudio de la argumentación y describe el estado del arte de varias áreas relevantes para nuestros objetivos:

- Modelado de competencias de estudiantes. Este es un campo relevante porque una parte muy importante de nuestros objetivos es generar un modelo de competencia en argumentación.

- Sistemas educativos para el desarrollo de competencias de argumentación: En las últimas décadas ha habido progresos en el área de Technology Enhanced Learning (TEL) y Computer Supported Collaborative Learning (CSCL) aplicada a la enseñanza de competencia en argumentación. Todos estos sistemas combinan procesos manuales y automáticos y definen un contexto a tener en cuenta en este trabajo.

- Sistemas para el procesamiento automático del discurso y la argumentación. Típicamente tareas de PLN y minería de argumentos para extraer la estructura retórica y argumental de un texto.

1.3.2. Algunos conceptos clave sobre la argumentación

1.3.2.1. Discurso, argumentación y debate

A efectos prácticos, el discurso lo podemos entender como un acto comunicación (hablada o escrita) entre dos o más personas o agentes (véase [43] para una revisión más profunda)

La argumentación se puede definir ([80], [123]) como un discurso en el que las personas con diferentes posturas:

- Justifican sus posiciones con argumentos (basados en razonamientos lógicos, datos, creencias, prejuicios, sensibilidad) e intentan persuadir a los demás participantes.
- Determinan la validez de las justificaciones (argumentos) ofrecidas.

La argumentación puede tener tres funciones ([80]):

- Dialéctica: los proponentes de diferentes posturas evalúan la validez de argumentos y contra-argumentos.
- Retórica: persuasión a través de argumentos basados en prejuicios o sensibilidad de la audiencia. En el ámbito educativo, se da más peso a la función dialéctica, pero en otros ámbitos, como la minería de opiniones, la función retórica puede ser interesante ([89]).
- Didáctica: Se alcanza certidumbre absoluta basándose en evidencia apodíctica, esto es argumentos presentados como verdad categórica.

Un debate puede considerarse un discurso en que los participantes sostienen diferentes posiciones e intercambian argumentos. La finalidad de un debate puede ser alcanzar una posición común.

1.3.2.2. Marcos de argumentación

Si se intenta crear un sistema automático de extracción, identificación o evaluación de argumentos, en principio se necesita un modelo conceptual o un marco formal que describa el discurso argumentativo para así poder empezar a identificar las tareas necesarias. Uno de los modelos clásicos es el de Toulmin ([63], [118], [10]), que define una estructura para la argumentación y razonamiento humano basado en seis categorías:

- Evidencia que es la base de una afirmación («claim»).
- Una regla de inferencia que une evidencia y afirmación (justificación, «warrant»).
- Cualificadores («qualifiers»): elementos que indican nuestra certidumbre sobre la afirmación.

- Reserva o ataques («rebuttals»): elementos que imponen condiciones para que la afirmación sea cierta.
- Respaldo o soporte («backing»): justificación de la regla de inferencia.

Este modelo conceptual, o alternativas tales como el formalismo pragma-dialéctico [124], que enfoca la argumentación como un proceso de comunicación e interacción en el que la argumentación forma parte, idealmente, de un debate, o marcos conceptuales con orientación práctica, como IBIS ([58], que describe la organización de temas en la resolución de problemas utilizando grupos de discusión) o la aplicación de técnicas de lógica informal ([39]), necesitan una representación de la argumentación que se adapte al dominio del discurso (legal, científico, etc.) y a las técnicas empleadas.

En el campo de la IA se consideran tres tipos de representación de la argumentación ([63]):

- Retórica, centrada en la representación de audiencias y el efecto persuasivo de la argumentación.
- Dialógica: Conexión entre argumentos durante el discurso. Por ejemplo, interesa describir cómo un argumento ataca a otro argumento anterior.
- Monológica: Estructura interna del argumento y relaciones entre sus partes. Por ejemplo, descripción del grado de justificación entre evidencia y afirmación en un argumento.

Un modelo dialógico puede considerar un argumento como entidad fundamental (sin estructura interna), frente a un modelo monológico, que considera una estructura interna. A la hora de extraer argumentos a partir de texto libre (minería de argumentos), se suele partir de un modelo monológico para poder definir subtareas más sencillas (por ejemplo, identificación de componentes de un argumento y su función como evidencia o afirmación).

1.3.2.3. Modelos formales de representación de argumentos

Modelo de Walton Uno de los modelos más populares lo definió Walton ([63], [130]), caracterizando argumentos como un conjunto de premisas, una conclusión y y una aplicación (inferencia) entre premisas y conclusión. Hay otras variaciones más elaboradas que permiten una representación mono y dialógica.

Diagramas de argumentación Tal como se indica en [89], a partir del trabajo de Toulmin ([118]) se desarrollaron diferentes esquemas de anotación gráfica, con extensiones que permitían recursividad en la aplicación de tipos de argumentos, así como una primera representación dialógica. Estos métodos a veces no tenían una especificación precisa, lo que introducía ambigüedad en el uso.

Freeman ([89],[40, 39]) estudió la macroestructura de los argumentos (como premisas y conclusiones se organizan en estructuras más complejas) y modeló la argumentación como un proceso dinámico, un diálogo entre un agente que presenta y defiende afirmaciones y un agente que las ataca. De esta manera, el estudio de un texto argumentativo se reduce a identificar las preguntas correspondientes del agente atacante. Freeman elimina la diferencia entre evidencia y justificación, y las considera ambas como premisas, lo que facilita la conexión serial de premisas y argumentos.

En la referencia [89] proponen una extensión del modelo gráfico de Freeman, que permite representar sin ambigüedad patrones complejos de relaciones entre premisa y afirmación, así como ataques y contra-ataques. Ademas permite indicar si uno de los agentes considera uno de los elementos insuficiente.

Estructura retórica como representación de argumentos La estructura retórica de un texto modela la coherencia de un texto, considerando cómo los diferentes fragmentos del texto se relacionan entre sí para formar un discurso único. Existen varios modelos (SDRT, LDM D-LTAG [89]), pero destacamos:

1. RST (Rethorical Structure Theory, [65]), que representa el texto como un árbol de relaciones retóricas en el que las hojas son unidades elementales del discurso (EDU. «Elementary Discourse Unit»). Cada nodo es núcleo o satélite en la relación correspondiente.
2. PDTB (Penn Discourse Tree Bank [71]), que indica conexiones explícitas e implícitas entre fragmentos del discurso.

En principio, la diferenciación entre núcleo/satélite en RST puede ayudar a representar premisas y conclusiones, pero la utilización de esquemas tipo RST tiene serias limitaciones para representar [89]:

- Relaciones a distancia (premisas y argumentos separados por varias EDUs).
- Patrones tipo reserva y refutación de la reserva.

Por lo que se ha podido ver en la literatura, las relaciones retóricas se utilizan más como atributos de un discurso para ayudar a encontrar una estructura argumental (por ejemplo, PDTB en el trabajo de Niculae, Park, Cardie [78], o la propuesta de usar RST como soporte para el uso de DILIGENT [93]), que como un modelo de representación en sí.

Argument Interchange Format (AIF) [95] AIF es un modelo para describir la estructura de un argumento. En este sentido, su propósito es puramente monológico. La especificación básica es:

- Dos tipos de nodo, que pueden formar redes de argumentos:
  - Información (nodo I). Representan proposiciones contenidas en argumentos.

    - Esquema, (nodos S) representan el tipo o esquema de argumentación (patrón de razonamiento), limitados a regla de inferencia (RA), conflicto (CA) y preferencia (PA).
- AIF introduce predicados como enlaces con los que pueden definir de manera formal patrones de argumentación a partir de redes de argumentos.

AIF tiene una representación como RDF (Resource Description Framework), por lo que se puede utilizar en una web semántica. AIF también permite definir una ontología [95] con enunciados (nodos I), patrones de argumentación (representados como nodos S) y autores.

Otro modelo monológico de interés Introducimos el modelo monológico de representación de argumentos propuesto por Park, Blake y Cardie [86] ya que es el utilizado en el sistema de minería de argumentos creado por Niculae, Park y Cardie [78] que hemos utilizado en este trabajo. En este modelo los autores consideran que las cláusulas que participan en un argumento pueden ser de varios tipos:

- Proposiciones, con subtipo «Policy» (llamamiento a la acción), «Value» (creencia u opinión), «Fact» (hecho, idea sobre la que hay acuerdo y se puede dar por sentada) y «Testimony» (testimonio, relato personal de una experiencia).
- Evidencia, con subtipo testimonio o referencia («Reference», cita a una fuente considerada de evidencia objetiva).

Los autores definen una aplicación *Type* entre las cláusulas de un argumento y su subtipo. Un argumento es el conjunto de conclusión (una Proposición), una serie de razones (Proposiciones) que explican por qué la conclusión es cierta y evidencia que confirman la conclusión. Se pueden definir sub-argumentos de manera recursiva (unión de subconjuntos del argumento tal que *Type(conclusión, sub-argumentos)*, sea del subtipo Proposición).

A partir del concepto de argumento y sub-argumento, los autores definen un argumento evaluable imponiendo ciertas restricciones a las conclusiones y razones.

Modelo de Dung Dung [34] sigue una aproximación dialógica y postula un sistema formal de argumentos y una relación binaria de ataque/reserva («rebuttal»). Para un conjunto de argumentos, define la aceptabilidad de un argumento (A) respecto a un subconjunto de argumentos (S): A es aceptable respecto a S si cualquier argumento que ataca a B es atacado por S. Esta idea encaja bien con los métodos de la Lógica Natural ([11]), que se centra en estudiar cómo el lenguaje natural expresa inferencias lógicas.

En una línea similar Cabrio y Villata [22] propusieron, basándose en el formalismo de Dung utilizar la técnicas de RTE («Recognize Text Entailment», tarea del PLN, en la que se evalúa hasta qué punto un fragmento de texto, o su contradicción, es

consecuencia lógica de otro fragmento) para describir la argumentación utilizada en un texto como una serie de cláusulas que se apoyan o atacan entre sí.

Es interesante observar que Dung incorpora en su formalismo la aceptabilidad de un argumento. El resto de sistemas monológicos, describen posibles estructuras, pero típicamente necesitan criterios adicionales para definir que argumentos son aceptables o ciertos.

1.3.2.4. Competencia en argumentación y «scaffolding»

En el ámbito educativo ([80]), la competencia se puede definir de manera muy simplista como la capacidad de ejecutar las tareas y actividades básicas en una profesión De forma menos restrictiva, la competencia en argumentación sería la capacidad de participar en una discusión para:

- Construir, considerar y sopesar argumentos.
- Apoyar una posición, aceptar o rechazar contra-argumentos utilizando evidencias razonable.
- Clarificar incertidumbres, malentendidos y, en general llegar a entender de manera racional el asunto en discusión.

Se puede entender un argumento de manera informal como una estructura de afirmaciones, alegaciones o postulados con relaciones definidas por cómo una afirmación se infiere o es evidencia de otra (por ejemplo [125]). Identificar y evaluar las validez de estas relaciones de evidencia o inferencia, así como la de las afirmaciones o postulados es una parte clave del aprendizaje de la competencia en argumentación. Un diagrama que muestre estas relaciones de forma explícita es un mapa o diagrama de argumentos, y es una herramienta utilizada en muchos sistemas educativos en esta competencia.

El andamiaje ("scaffolding", véase [135] y edglossary.org) es una serie de técnicas educativas basadas en proporcionar una serie de apoyos temporales a los estudiantes que se van retirando según mejora su competencia. El aprendizaje colaborativo se puede definir de manera general como una situación en la que un grupo de estudiantes intenta aprender en una actividad colectiva (véase [31] para una discusión sobre el concepto).

Tal como se explicó en el contexto de este trabajo, la competencia en argumentación es importante y ha adquirido relevancia a la hora de diseñar programas de estudio. Aparte de las razones clásicas de desarrollo profesional (legal o científico, por ejemplo), en el mundo actual hay cada vez más problemas complejos, con mucha incertidumbre, y en los que se cruzan intereses de todo tipo. En estos casos, la argumentación ayuda a los participantes a entender de manera racional la situación y tomar mejores decisiones.

Aunque ahora hay muchas más oportunidades para practicar la argumentación que en el pasado (debido a las redes sociales y foros de discusión), redes y foros no suelen

favorecer una mejora en la competencia, debido sobre todo a la falta de incentivo para proporcionar buenos argumentos y considerar de manera racional los contra-argumentos.

En el mundo educativo, específicamente,hay varios obstáculos para la enseñanza de las competencias de argumentación [80]:

- Estas competencias no pertenecen en exclusiva a una única área clásica.
- No hay un método estándar de evaluar la competencia. Hay diferentes dimensiones reconocidas: meta-cognitiva (estructura y valor epistémico del argumento, esto es, el motivo del rechazo o aceptación), meta-estratégica (presencia o ausencia de elementos de la argumentación y tipos de discurso utilizados) y epistemológicas (calidad y objetivo de la argumentación) Hay evidencias de que el comportamiento de un estudiante durante la argumentación no llega a reflejar su conocimiento teórico. Esto apoya al idea de que para medir la competencia en argumentación de un estudiante se debe medir sobre todo el desempeño práctico.
- El desempeño en el discurso está también afectado por cuestiones psicológicas, emocionales, de motivación y sociales (presión del grupo, por ejemplo).

1.3.3. Representación de competencias y visualización de resultados del aprendizaje

Desde los últimos 15 años se está produciendo un cambio de los objetivos del sistema educativo para no centrarse únicamente en la adquisición de conocimientos y dar énfasis al aprendizaje de competencias (ver por ejemplo el proyecto Tuning, [13]). Como consecuencia, el área de sistemas educativos para ofrecer herramientas de soporte a la educación en competencias ha crecido. En este trabajo no pretendemos ofrecer una revisión completa del estado del arte, ya que abarca un campo mucho más amplio que la competencia en argumentación, sino centrarnos en unos pocos puntos que son relevantes para nuestro estudio.

Un punto clave en los sistemas educativos es modelar el perfil de los estudiantes [26] (por ejemplo, conocimientos, comportamiento). Para ello se han utilizado diferentes algoritmos y modelos (arboles de decisión, redes neuronales y sistemas de lógica borrosa entre otros). A la hora de modelar las competencias de los estudiantes, se han utilizado ontologías o modelos psicológicos formales como la Teoría del Espacio de Conocimiento («Competence-based Knowledge Space Theory») además de redes bayesianas [60, 20]. Tras revisar el uso de una red bayesiana en la referencia [60], opinamos que la ventaja de este tipo de modelo es que proporciona una manera relativamente sencilla de visualizar correlaciones entre diferentes variables, que en algunos casos se pueden identificar con relaciones de causalidad o de diagnóstico. Una red bayesiana se puede construir manualmente, incorporando conocimiento experto o a partir de datos, y al describir una distribución de probabilidad conjunta, permite

realizar inferencia sobre todas las variables implicadas, por lo que se puede extraer información variada.

En la referencia [60], los autores experimentan con una red bayesiana para describir un área de conocimiento: Las variables son por un lado competencias conocidas en un área, una serie de eventos (problemas a resolver) y conocimientos adquiridos. La estructura de la red se establece manualmente y las probabilidades conjuntas se estiman inicialmente y se pueden actualizar incorporando más datos.

En el área del análisis y visualización del aprendizaje, se utilizan también modelos del estudiante. Siguiendo la referencia [20], se puede hablar de modelos del estudiante abiertos («Open Learner Models»), en los que los rasgos o perfil del estudiante le son accesibles a este. Es común la utilización de visualizaciones de competencias como diagramas de barras o como grafos, en las que cada nodo indica una competencia ([20], página 151).

1.3.4. Sistemas educativos para el desarrollo de competencia en argumentación

En esta sección se hará una revisión de los sistema educativos computacionales utilizados para el desarrollo de competencia en argumentación, destacando aquellos que utilizan alguna técnica de IA. Dada la extensión del tema, esta sección se centra en los aspectos más relevantes y en el apéndice A.2 se incluye un listado detallado de los sistemas que se han identificado. Véase las tablas 1.1, 1.2,1.3 para una lista de todos los sistemas considerados.

Hay dos observaciones que se pueden realizar:

- La mayoría de los sistemas aquí incluidos se limitan a proporcionar una plataforma tecnológica para las actividades de aprendizaje, sin aplicar métodos de IA.
- Los sistemas educativos que aplican técnicas de IA e implementan algún tipo de aprendizaje a partir de entradas de estudiantes son del siglo XX. Los sistemas más modernos utilizan una ontología o modelo formal pre-definido como ayuda en un sistema educativo que requiere entradas manuales.

Creemos posible que, debido a la complejidad técnica de las tareas asociadas a la minería de argumentos y el Procesamiento de Lógica Natural, no haya habido avances apreciables en la aplicación de técnicas de aprendizaje automático en el desarrollo de sistemas educativos.

1.3.4.1. Tipos de sistema según su objetivo

Siguiendo a [80], los sistemas de software para el desarrollo de competencias de argumentación son de dos tipos, según sus objetivos.

Orientados a guiar el discurso Dirigen al estudiante en su discurso argumentativo, incentivando las interacciones y técnicas argumentativas que se consideran deseables (participación, ceñirse la tema de discusión. Estos sistemas se usan sobre todo para organizar y dirigir la interacción entre estudiantes, fomentando un aprendizaje colaborativo sobre aspectos de los temas discutidos. Ayudar al estudiante a practicar y evaluar su estilo de argumentación suele ser un aspecto secundario en este tipo de diseños.

Estos sistemas suelen tratar directamente con texto natural, y las salidas suelen estar abiertas a interpretación. Un ejemplo sería CoLLeGe [99], que, a través de un diálogo dirigido con un estudiante, parsea las respuestas del estudiante en cada turno y utiliza un motor de razonamiento basado en reglas para redirigir el diálogo, con el objetivo final de optimizar el nivel de refinamiento del conocimiento mostrado por el estudiante.

Orientados a la construcción de argumentos Son sistemas orientados a dar soporte al estudiante para construir, estructurar, clasificar, analizar y evaluar argumentos. Siguiendo a [104] Este proceso puede ser:

- Exploratorio y dirigido por el usuario, por ejemplo, a través de la creación y validación de un argumento.
- Encuadrado en una tarea (típicamente resolución de un problema), como por ejemplo traducir un texto a un diagrama de los argumentos.

Estos sistemas suelen definir una semántica de los argumentos y tener una salida bien definida (por ejemplo, patrones que indican errores en la argumentación). Hay ejemplos como Carneades [45] o Online Visualization of Argument(OVA) que implementan modelos formales de representación de argumentos u ontologías para ayudar a los estudiantes a confeccionar argumentos.

1.3.4.2. Tipos de sistema según su funcionalidad

Siguiendo de nuevo a [80], podemos clasificar los sistemas para el desarrollo de competencias de argumentación en cuatro grandes grupos según su funcionalidad o el tipo de proceso detallado que implementan.

Sistemas de representación de argumentos Aplicaciones utilizadas por los estudiantes para construir, examinar y manipular argumentos y representaciones del conocimiento. El sistema provee primitivas a partir de las que se puede construir una representación de hipótesis, hechos, etc. y el usuario construirá artefactos (por ejemplo, diagramas de conceptos utilizando estas primitivas) [114]. La implementación típicamente utiliza una semántica para primitivas y representaciones.

Las tipos de representación comúnmente utilizados son:

- Representaciones gráficas: Por ejemplo, Carneades [45]. Presentado como un modelo formal de discurso argumentativo legal, centrado en determinar de forma precisa dentro del mapa de argumentación de un debate o intercambio de argumentos en qué nodo recae el peso de la prueba («burden of proof», [45]). Se puede combinar con una herramienta para representar mapas de argumentación gráficamente ([104], Carneades - Implementación en Go).
- Representación como tabla o texto.

Diferentes estudios indican que los dos tipos de herramientas, gráficas y textuales o tabla, mejoran la calidad de los resultados del aprendizaje ([80]). Sin embargo se observan algunas diferencias:

1. Los estudiantes que utilizaban diagramas o mapas de argumentos se centraban más en encontrar un equilibrio entre argumentos.
2. Los estudiantes que utilizaban tablas se centraban en los cambios que se habían ido acumulando durante la actividad.
3. Otros estudios sugieren que, en el caso de diagramas, la calidad del soporte de los argumentos, de la presentación por escrito y de la representación del conocimiento es mejor. No se observa esa diferencia para otro tipo de actividades de la argumentación (discusión de argumentos y contra-argumentos, por ejemplo).

Colaboración guiada por software/»Micro-scripting» Estos sistemas proporcionan una guía paso a paso de la secuencia de actividades en una discusión. En general, el «micro-scripting» se implementa a partir de líneas de comando ("prompts"), comienzos de frases o raíces de preguntas. Un ejemplo interesante es el experimento descrito en la referencia [113], en la que Stegmann, Weinberger y Fischer presentan una metodología para medir la efectividad del aprendizaje tomando como referencia inicial una estructura ideal del argumento y en una primera ronda del experimento medir la frecuencia con la que los argumentos de los estudiantes siguen esta estructura ideal. Los cambios en estas frecuencias al volver a generar argumentos en ciertas condiciones proporcionan una métrica de efectividad de una sesión de aprendizaje.

Los sistemas de "micro-scripting" ([80]) pueden ayudar a que los estudiantes se centren en la construcción de sus argumentos. Se pueden diseñar para incentivar que los estudiantes sigan determinadas secuencias de argumentación o se centren, por ejemplo, en analizar los argumentos de sus compañeros.

Guión de actividades de aprendizaje/»macro-scripting» Este término se refiere a planes de trabajo para grupos de aprendizaje de competencia en argumentación en los que se asignan roles, tareas y actividades a los estudiantes. Suelen utilizar estos tres tipos de planificación:

- Secuencial («traversal»): los estudiantes siguen la misma serie de actividades secuencialmente.

- Rotación: Se anima a los estudiantes a participar en cada actividad cambiando el orden de sus elementos.
- Los estudiantes participan en tareas de aprendizaje con un apoyo que va creciendo o decreciendo gradualmente ("fading").

En general, hay dos fases durante la actividad:

- Coordinación de tareas (por ejemplo, asignando roles específicos como moderador, investigador, etc. a los estudiantes).
- Discusiones preparadas ("conflict scheme/seeded discussions)": Los estudiantes se organizan de acuerdo a sus puntos de vista diferentes sobe el tema a argumentar. Estas diferencias se utilizan como punto de partida en las discusiones. Hay observaciones de que se obtienen mejores resultados del aprendizaje cuando se parte de semillas proporcionadas por expertos.

1.3.4.3. Juegos de diálogo digitales

Los tres tipos de herramientas anteriores encuentran dificultades para implantarse en la vida real por varias razones: La argumentación no encaja claramente en una materias tradicional (incorpora aspectos de varias materias diferentes), por lo que no hay una metodología de aprendizaje clara. Además, barreras psicológicas y sociales como la falta de contexto en las herramientas disminuye el interés de los estudiantes.

Inicialmente, el término juegos de diálogo («Dialog Game») ([99]) se utilizaba en el área de las ciencias cognitivas para describir patrones observados en el diálogo humano (por ejemplo, »Helping, Seeking-Information, Information-Probing, Instructing»). En el campo del aprendizaje, un juego de diálogo pasó a designar una actividad de grupo en la que se asignan roles, objetivos, y las reglas que definen qué tipo de movimientos son aceptables así como los turnos para jugar si es necesario.

El uso de juegos de diálogo en la enseñanza de competencias de argumentación se puede ver como una aplicación del argumento socio-constructivista de que los estudiantes aprenderán mejor los puntos esenciales de la argumentación practicándolos en lugar de leer/pensar sobre ellos ([25]). Estas actividades motivan al estudiante a que se forme sus propios conceptos y colabore con sus compañeros para refinarlos.

Como ejemplo, destacamos Computer-Based Lab for Language Games in Education (**CoLLeGE**) ([99]). Implementa un modelo de aprendizaje como refinamiento del conocimiento. Está orientado a un único jugador, que introduce sus argumentos por turno. El sistema gestiona:

1. Un modelo del mundo (parseado a partir de las entradas del estudiante) y un motor de razonamiento basado en reglas.
2. Un gestor de diálogo, que determina la táctica a utilizar.

3. Estos dos sistemas van generando una agenda táctica y una agenda de razonamiento que se pueden ir vaciando según la respuesta del estudiante. El juego acaba cuando las dos agendas están vacías.

Aunque el uso de juegos de diálogo parece tener potencial, faltan estudios en escenarios reales ([80]). También faltan pautas para diseñar juegos que den buenos resultados en el desarrollo de la competencia en argumentación.

1.3.5. Minería de argumentos

1.3.5.1. Consideraciones sobre metodología

La minería de argumentos es un área del procesamiento de lenguajes natural (PLN) y la lingüística computacional que intenta identificar de manera automática los argumentos incluidos en texto libre (por ejemplo, en un ensayo o en un debate). Es una tarea compleja que, en general, requiere atacar tres problemas diferentes [63]:

- Identificar los fragmentos de texto que contienen cláusulas o proposiciones argumentativas.
- Identificar los límites de las cláusulas argumentativas en cada fragmento de texto. En bastante común que se realicen asunciones simplificadoras para unificar estas dos primeras tareas en un único problema de clasificación
- Identificar la estructura del argumento. Esto puede incluir identificar la función de cada cláusula argumentativa (evidencias, afirmaciones, ataques, etc.) y las cláusulas conectadas funcionalmente (como se hace, por ejemplo, en las referencias [111], [78], o identificar únicamente las relaciones de soporte y ataque entre cláusulas, basándonos sobre todo en estimar hasta que punto una frase implica otra frase o su contraria, o RTE [22]).

La minería de argumentos comparte parcialmente metodología y objetivos con la minería de opiniones (véase por ejemplo [128]) o el análisis de sentimientos ([63]), pero, aparte de los diferentes dominios que tradicionalmente han tratado estas áreas, la inferencia de la estructura suele ser menos importante en estos casos, lo que limita el tipo de algoritmos o corpus que se pueden compartir.

Normalmente, los sistemas de minería de argumentos no intentan evaluar la validez de los argumentos. Una excepción es el sistema propuesto por Cabrio y Villata [22], que, basado en el modelo formal de Dung, permite definir de forma natural la aceptabilidad de un argumento.

En general, la minería de argumentos se ataca como un problema supervisado de aprendizaje automático, utilizando corpus anotados y diferentes algoritmos para las tareas de búsqueda de cláusulas argumentativas y de la estructura del argumento. Dos puntos de interés:

- Los sistemas existentes se suelen basar en un conjunto de atributos de texto muy elaborados, incluyendo estadísticas, rasgos léxicos, semánticos, sintácticos, diccionarios de conectores relevantes, y criterios de información mutua entre atributos de fragmentos de texto contiguos.
- Aunque ha aumentado el número de recursos disponibles, muchos sistemas se han entrenado en dominios bastante específicos (legal, educación), por lo que consideramos que la capacidad de transferencia a otros dominios diferentes es todavía un punto abierto.

1.3.5.2. Corpus disponibles

Uno de los principales obstáculos para desarrollar sistemas de minería de argumentos era la falta de corpus anotados para la identificación de argumentos. En los últimos diez años han aparecido corpus en inglés que han permitido un progreso relevante en el área. Siguiendo principalmente a Lippi y Torroni ([63]), destacamos:

1. La versión original de AraucariaDB [100] era una colección más de 600 textos cortos legales, discusiones en foros y fragmentos de artículos de prensa, anotado según el modelo de esquemas de argumentación ([101]). La nueva versión ha perdido esta anotación y sólo contiene el texto original y mapas de argumentación utilizando el modelo AIF.
2. NoDe, con unos 800 pares de cláusulas con relaciones de soporte/ataques entre ellas. Las fuentes son Debatepedia, los foros de discusión de revisiones en WikiPedia y el guión de una obra teatral que representa un juicio («Twelve Angry Men»). Este corpus solo contiene cláusulas argumentativas.
3. Corpus en el dominio legal, como las disposiciones del ECHR (Tribunal Europeo de Derechos Humanos, 45 resoluciones inicialmente anotado utilizando como un árbol que siguen de cerca el marco pragma-dialéctico [83], [124]) o el corpus del proyecto V/IP (Vaccine/Injury Project) (la referencia original se puede encontrar en [63]).
4. Argument Annotated Essays Corpus (AAEC) versiones 1.0 [109] y 2.0 [110], con 90 y 402 ensayos argumentativos creados por estudiantes. Se anotan las cláusulas argumentativas y las relaciones de apoyo y ataque siguiendo una estructura de árbol con ciertas restricciones en la estructura: se define una afirmación principal y varias secundarias, de las que dependen el resto de cláusulas.
5. El corpus de IBM para detección de afirmaciones y evidencias, [1]: con más de 2000 argumentos sobre 33 temas, extraídos de la Wikipedia. Se anotan cláusulas que apoyan o atacan uno de los temas seleccionados. En principio, presenta una utilidad limitada para modelos que intenten aprender estructuras de argumentación compleja, pero puede ser aplicable a la tarea de detección de proposiciones argumentativas.

6. Corpus de comentarios en foros web en inglés [47], anotados con una variación del modelo formal de Toulmin.
7. Cornell eRulemaking Corpus - CDCP, [78] : 1462 textos que recogen la discusión sobre nuevas propuestas de reglamentación relativa a la morosidad, eRulemaking. Los argumentos están anotados con un modelo que permite una estructura de argumentación más general que un bosque. No distingue entre apoyo o ataque a afirmaciones, pero admite más granularidad en el tipo de apoyo y afirmaciones (por ejemplo distingue expresión de hechos, juicios de valor, opiniones y propuestas de acción).

1.3.5.3. Algunos sistemas de minería de argumentos

En la referencia [63] hay una revisión de sistemas de minería de argumentos hasta el año 2016. Destacamos:

1. Mochales y Moens [73] entrenan una serie de modelos en los corpus Araucaria y ECHR para segmentar el texto en cláusulas argumentativas, detectar su tipo (afirmación, soporte,...) y encontrar la estructura del argumento (estas dos últimas tareas únicamente para el corpus ECHR). Los resultados son buenos, pero el modelo está muy adaptado al texto legal, por lo que no podemos esperar que se pueda aplicar directamente a otros dominios.
   a) En las dos primeras tareas utilizan clasificadores de entropía máxima (regresión logística) y máquina de soporte vectorial (SVM) sobre una serie de atributos diferentes para cada tarea. Las autoras reportan una métrica F1 mayor de 70 % para afirmaciones y cercana al 70 % para evidencias.
   b) Para aprender la estructura de los argumentos, se creó manualmente una gramática libre de contexto (CFG). Al parsear el texto con esta gramática se genera una estructura de argumentación. Se reporta un F1 cercano al 70 %.
2. Cabrio y Villata [22] utilizaron texto extraído de Debatepedia (sentencias a favor o en contra de una serie de 20 temas) para entrenar un modelo de RTE y definir un argumento para cada uno de los tema seleccionados siguiendo un modelo inspirado en las ideas de Dung [34]. Como métricas de validación las autoras utilizan:
   a) Precisión y exhaustividad («recall») de la tarea RTE, alineada con los resultados típicos del modelo utilizado (algo menor del 70 %).
   b) Impacto de la precisión y exhaustividad del sistema RTE en la aceptabilidad del cada argumento. Obtienen una precisión y exhaustividad por encima del 70 %.

   A diferencia del sistema de Mochales y Moens, este modelo sólo aprende las relaciones entre cláusulas argumentativas, sin intentar detectarlas en el texto,

y depende muy estrechamente del sistema utilizado en la tarea de RTE (las autoras utilizaron EDITS en 2012). RTE es una tarea PLN que ha avanzado rápidamente desde que se hiciera público el corpus Stanford Natural Language Inference Corpus en 2015 ([18], con algunos grupos reportando precisiones cercanas al 90 %. Un ejemplo del tipo de métodos utilizado se puede encontrar en la referencia [85].

Más recientemente se ha avanzado en sistemas que pueden realizar la detección de cláusula argumentativa y estructura del argumento sin una dependencia tan fuerte del tema del texto que tienen sistemas como el descrito en [73].

1. Stab y Gurevych ([111]) utilizan el corpus AAEC v2 para entrenar un modelo que identifica cláusulas argumentativas y estructura (en este caso, árboles):
   a) Los atributos utilizados incluyen, entre otros, rasgos léxicos, sintácticos, retóricos y probabilistas.
   b) Utilizan un modelo de «Conditional Random Field» (CRF, un modelo gráfico probabilista de máxima entropía apropiado para problemas en red en los que hay dependencia con los nodos vecinos) para segmentar el texto e identificar las cláusulas argumentativas.
   c) Utilizan una máquina de soporte vectorial (SVM) para identificar la estructura argumental.
   d) Combinan los resultados de los dos clasificadores como un problema de programación lineal, en el que se necesita optimizar una función objetivo que reproduce la estructura de los argumentos de cada ensayo.
   e) El modelo obtiene una métrica F1 de más del 80 % en la segmentación, 75 % en la identificación de relaciones y cerca del 70 % en la diferenciación de soporte y ataque.
2. Niculae, Park y Cardie ([78]) siguen parcialmente a Stab y Gurevych [111] que combina la segmentación en cláusulas argumentativas y el aprendizaje de la estructura del argumento como un problema de aprendizaje estructural:
   a) En el aprendizaje estructural se busca optimizar una función ganancia definida en los atributos del texto y enlaces (los autores siguen de cerca los utilizados por Stab y Gurevych), la estructura a buscar y los parámetros del modelo a utilizar. La estructura a buscar se modela como un «factor graph» (grafo bipartito que representa la factorización de la distribución de probabilidad sobre la estructura del argumento), de modo que al considerar los potenciales asociados al «factor graph», el problema se puede atacar como el cálculo de los parámetros que nos proporcional el máximo «a posteriori» (MAP) de la función ganancia.
      1) Utilizar un grafo permite considerar arboles o estructuras más generales, y permite incluir restricciones «ad hoc» de forma relativamente fácil.

b) El punto técnico clave en la implementación es la parametrización de los potenciales asociados a la estructura. Los autores prueban:

1) Máquina de soporte vectorial, SVM, estructuradas [120] [75] (generalización del algoritmo SVM, en el que la etiqueta a separar es un grafo y la función de riesgo a minimizar incluye una distancia definida sobre los grafos a considerar).

2) Redes neuronales de diferentes arquitecturas (LSTM, Long-term and Short-term Memory, [50, 46], y MLP o perceptrones multi-capa).

c) Los autores reproducen los resultados de Stab y Gurevych en el corpus AAEC v2. Para el corpus CDCP, obtienen valores de F1 mayores del 70 % en la identificación del tipo de cláusula argumentativa y del 50 % en la detección de enlaces.

3. Stab, Miller y Gurevych ([112]) generan un nuevo corpus anotado con una relación simplificada de soporte/ataque entre cláusula argumentativa y tema. El texto se vectoriza utilizando un modelo de «word embeddings» ([69],[90]), y junto con una medida sencilla de similaridad con el tema, entrenan una red neuronal (LSTM bidireccional) e incluyen un término de importancia de los atributos del texto como parámetros a aprender (estos modelos se denominan de atención interna, «inner-attention»). Los autores encuentran valores de F1 entre el 60 y el 80 % dependiendo de los temas considerados, pero, en nuestra opinión, el mayor interés de este trabajo es:

a) Proponen un modelo de anotación sencillo, que no necesita especialistas, por lo que es más fácil obtener conjuntos de datos grandes.

b) Los atributos del texto son muy sencillos, sobre todo en comparación con los dos modelos previos que hemos revisado.

En el último par de años, con la disponibilidad de mejores sistemas de minería de argumentos, hay más trabajos que utilizan los argumentos de un texto para mejorar tareas. Por ejemplo, Nguyen y Litman desarrollan un sistema de extracción de argumentos basado en [111] para definir nuevos atributos que mejoran el rendimiento de sistemas automáticos de evaluación de ensayos persuasivos [77]. En este trabajo, aunque utilizamos uno de los corpus analizados por Nguyen y Litman (2.2), el objetivo es diferente al de evaluar automáticamente ensayos.

En este estudio utilizaremos el software y modelo de minería de argumentos publicado por Niculae, Park y Cardie ([78]) para extraer los tipos de cláusula argumentativa y las relaciones entre estas.

Funcionalidad	Sistema	Técnica de IA utilizada	Implementado	Año de publicación
Representación de argumentos	CATO	Razonamiento basado en casos	Si	1997
Representación de argumentos	Convince Me	Algoritmo conexionista para estimar la aceptabilidad de argumentos (ECHO).	Si	1998
Representación de argumentos	Carneades	Modelo formal de representación de argumentos	Si	2007
Representación de argumentos	Zeno	Modelo formal de representación de argumentos	No	1997
Representación de argumentos	DUNES: Digalo y Argunaut	Utiliza una ontología de la argumentación	Si	2007
Representación de argumentos	DREW y DREWLITE	Utiliza una taxonomía de los resultados del aprendizaje (SOLO)	Si	2002
Representación de argumentos	Online Visualization of Argument/Araucaria	Modelo formal de representación de argumentos	Si	2009 (ontología AIF)
Juegos de diálogo	CLARISSA	Simulación de un diálogo a través de una máquina de estados	Si	2000
Juegos de diálogo digitales	Computer-Based Lab for Language Games in Education (CoLLeGe)	Modelo basado en reglas para gestionar el diálogo con un estudiante, parseando sus respuestas como entrada de un proceso en el que se busca optimizar el nivel de refinamiento del conocimiento.	Si	2000

Cuadro 1.1.: Sistemas educativos para el desarrollo de competencia en argumentación que utilizan técnicas de IA o relacionadas. Estamos incluyendo sistemas que utilizan modelos formales u ontologías predefinidas, pero sin aprendizaje basado en los datos o entradas de los estudiantes (Carneades, Zeno, DUNES, DREW y DREWLITE, OVA). Indicamos que el sistema está implementado si ha habido en algún momento una aplicación de software que lo implemente. En algunos casos, los enlaces al software están inactivos, y en otros, solo hemos encontrado referencias del uso del sistema en experimentos controlados.

Funcionalidad	Sistema	Implementado	Descripción
Macro-scripting	Rashi	Si	Entorno que simula el proceso de enunciado, discusión y verificación de hipótesis en algunos dominios científicos
Representación de argumentos	Belvédère	Si	Sistema gráfico de representación de argumentos para aprendizaje colaborativo
Representación de argumentos	Reason!Able	Si	Sistema de construcción de mapas o diagramas de argumentos. Incluye un sistema de evaluación de argumentos que necesita entradas manuales
Representación de argumentos	LARGO	Si	Modelo de enseñanza de evaluación de discurso argumentativo legal, centrado en identificación de puntos débiles del argumento y basado en el filtrado colaborativo
Representación de argumentos: Tablas	TC3	Si	Plantillas compartidas para elaborar un discurso argumentativo de forma lineal
Micro-scripting	C-CHENE	Si	Aplicación gráfica para proporcionar un soporte básico a alumnos trabajando en problemas de Física
Micro-scripting	Scripted Cooperation, Ask to Think/Tel-Why, Structured Academic Controversy, Reciprocal Teaching	Si	Guiones de actividades colaborativas de argumentación.

Cuadro 1.2.: Sistemas educativos para el desarrollo de competencia en argumentación que no utilizan técnicas de IA

Funcionalidad	Sistema	Implementado	Descripción
Micro-scripting y juegos de diálogo digitales	AcademicTalk	Si	Sistema de chat con opciones de frase de apertura estándar
Micro-scripting y juegos de diálogo digitales	InterLoc	Si	Sistema de chat con opciones de frase de apertura estándar
Macro-scripting	ArgueGraph	Si	Plan para que los estudiantes generen un mapa de opiniones y discutan sobre estas de manera organizada
Macro-scripting	ConceptGrid	Si	Los estudiantes crean y discuten una tabla de conceptos de forma colaborativa
Macro-scripting	WiSim	Si	Aplicación móvil para dar soporte a discusiones sobre problemas de Física
Macro-scripting	UniverSanté	Si	Herramienta gráfica para facilitar la interacción entre estudiantes e instructores en diferentes localizaciones. Utilizado en una experiencia de educación en sanidad comunitaria en la que se busca contrastar diferentes puntos de vista
Macro-scripting	WISE	Si	Sistema que integra simuladores y paneles de discusión. Orientado a la enseñanza de Física
Juegos de diálogo digitales	DIALAB	Si	Sistema de práctica de diálogo en base a sentencias primitivas sencillas. Implementa un modelo propio de diálogo

Cuadro 1.3.: Continuación de la lista de sistemas educativos para el desarrollo de competencia en argumentación que no utilizan técnicas de IA

2. Metodología

2.1. Introducción

Para llevar a cabo los objetivos de este trabajo, se van a realizar las siguientes tareas:

- Proponer una lista de atributos del texto argumentativo que consideramos se asocian a indicadores de la competencia en argumentación o la competencia lingüística en general.
- Especificar el método de generar estos atributos a partir de texto. En dos casos, coherencia y estilo, se necesita entrenar modelos de aprendizaje automático para obtener estos atributos.
    - Identificar los corpus de texto que se utilizan en el entrenamiento de estos modelos.
- Entrenar y validar una red bayesiana como modelo de los indicadores de competencias utilizando los atributos seleccionados.
    - Identificar el corpus de ensayos argumentativos a utilizar en este entrenamiento.
- Mostrar como la red bayesiana puede generar unas recomendaciones básicas.
- Especificar y aplicar un algoritmo de resumen y visualización de argumentos para obtener el grafo de argumentación colaborativo de un subconjunto de ensayos del corpus utilizado en el entrenamiento de la red bayesiana.
    - Entrenar y validar un modelo de clasificación para la identificación de paráfrasis. Esta es una herramienta necesaria para el algoritmo de resumen y visualización.

Al utilizar atributos asociados a indicadores de una competencia en el modelo bayesiano, se consigue que este modelo y sus resultados sea más fácil de interpretar por parte de instructores y estudiantes. Otra ventaja de un modelo bayesiano es que se puede re-entrenar fácilmente si añadimos nuevos datos. No se ha experimentado con el re-entrenamiento en esta primera aproximación al problema.

Como se indicó en la introducción, el estudio se centra únicamente en texto en inglés, para el que hay más recursos para PLN (Procesamiento de Lenguaje Natural), tanto modelos como corpus. Para aplicar la metodología a otros lenguajes, se requeriría una nueva compilación de texto anotado y la adaptación de algunos de los modelos que hemos utilizado para extraer atributos lingüísticos (por ejemplo, estructura retórica).

A recalcar que el objetivo de este trabajo no es la calificación automática de ensayos. Se introducen como variables del modelo bayesiano la calificación de evaluadores humanos, y predecir esta clasificación nos permite validar el modelo. Sin embargo, el objetivo final de nuestra red bayesiana es expresar la interrelación de indicadores de competencia predefinidos junto con los definidos únicamente en los datos de entrenamiento (calificaciones de evaluadores humanos), y así poder responder preguntas sobre el efecto de los diferentes indicadores de competencia entre sí.

2.2. Datos utilizados

Al revisar los corpus existentes con texto argumentativo disponibles públicamente, los más relevantes para este trabajo son:

Argument Annotated Essays Corpus (AAEC) versión 2.0 402 ensayos argumentativos [109, 110] obtenidos en https://essayforum.com/ (un portal web en el que estudiantes cuelgan texto para recibir correcciones). Los autores del corpus anotaron la estructura de los argumentos utilizados siguiendo un modelo argumentativo propio. Este modelo describe la estructura de un argumento como un conjunto de árboles, en los que el nodo raíz es un una afirmación principal ("major claim"), de la que descienden varias afirmaciones secundarias ("claims") que pueden apoyar o atacar la afirmación principal y respaldos que apoyan estas afirmaciones secundarias.

Araucaria La versión actualmente disponible de Araucaria [100] (AraucariaDB) consta de unos 650 textos argumentativos, principalmente en inglés, anotados siguiendo el modelo AIF [95]. Las fuentes son textos periodísticos, foros de discusión, registros parlamentarios y textos legales. Las versiones públicas actualmente sólo contienen un mapa de argumentos (incluyendo argumentos implícitos que no se encuentran en el texto), y el texto original, por lo que la correspondencia directa entre argumento y texto se tiene que inferir. Esto limita la utilidad de este corpus para minería de argumentos, pero es útil si se quiere estudiar texto argumentativo de diferentes dominios .

Kaggle ASAP Phase 1 Los Automated Student Assessment Prize (ASAP) Phase 1 Kaggle Competition Datasets, Kaggle ASAP Phase 1, son una serie de ensayos cortos (150 a 550 palabras) creados por estudiantes de entre 12 y 15 años sobre 8 temas predefinidos. Cada texto está calificado por dos o tres expertos en uno o dos dominios.

- Kaggle es un portal en el que se proponen problemas que pueden admitir soluciones utilizando aprendizaje automático o IA (Inteligencia Artificial), y en el que diferentes equipos compiten por encontrar una solución con la mayor

precisión. La competición ASAP fue propuesta por la Hewlett Foundation, con el propósito de desarrollar sistemas de calificación automática, pero los datos presentados se han utilizado en diferentes estudios sobre minería de argumentos ([77]) .

Se utilizó el corpus Kaggle ASAP Phase 1 para entrenar y validar el modelo bayesiano de indicadores de competencia en argumentación. Para acotar el experimento, utilizaremos únicamente 531 documentos del tema 2, en el que dos expertos proporcionan una nota por ensayo en dos áreas: efectividad en la expresión («Writing Application», o dominio 1) y dominio del lenguaje («Language Convention» o dominio 2). Nótese que la documentación del corpus incluye rúbricas más detalladas para estas áreas o competencias.

Cada área tiene una nota agregada por ensayo. Los textos están creados por estudiantes de unos 15 a 16 años (nivel 10 en el sistema educativo de los Estados Unidos de América).

Una de las limitaciones de este estudio es la edad de los estudiantes que crearon los textos (13 a 15 años). Se puede esperar que, el nivel de madurez sea un factor relevante y, que al entrenar el modelo a partir de datos directamente, el modelo no se pueda transferir directamente (sin re-entrenar) a otro tipo de estudiantes. Sin embargo, consideramos que, aceptando esta limitación, la metodología es valida. Sería interesante re-entrenar el mismo modelo con otros conjuntos de datos para otros rangos de edades/nivel de formación, pero esto sería tema para un futuro trabajo.

Microsoft Research Paraphrase Corpus Se ha utilizado el corpus Microsoft Research Paraphrase Corpus [129] para entrenar el modelo de identificación de paráfrasis que se ha aplicado en el algoritmo de resumen y visualización de argumentos. Este corpus contiene 5801 pares de frases en inglés, etiquetadas con uno si una frase se considera paráfrasis de la otra o cero en cualquier otro caso, y está anotado semi-automáticamente con ayuda de un modelo clasificador.

En diferentes tareas se han utilizado partes del corpus de Brown [38] (editoriales y reseñas) y varios corpus distribuidos con la librería de Python NLTK (Natural Language Toolkit, [14]), incluidos WordNet [121, 70] y SentiWordNet ([6]) .

2.3. Generación de atributos del texto argumentativo

2.3.1. Consideraciones generales

En este estudio se propone una serie de atributos o rasgos del texto argumentativo que consideramos que se pueden asociar aproximadamente a indicadores de la com-

petencia en argumentación y competencias relacionadas con el fin de entrenar un modelo bayesiano de estos indicadores. Una guía que se ha elegido para la selección de atributos es utilizar indicadores que aproximen criterios estándar de pensamiento crítico enunciados por Elder y Paul [35], especialmente las competencias centradas en el razonamiento y estándares intelectuales («Elements of reasoning and intelectual standards as they relate to the elements»), la revisión critica del pensamiento («Assessing Thinking») las habilidades necesarias para el aprendizaje, como la expresión escrita («Skill of Substantive Writing») y las barreras al pensamiento racional («Barriers to the Development of the rational thought»).

La definición de estas competencias y las utilizada en rúbricas sobre la capacidad de argumentación ([82, 23]) son de muy alto nivel, lo que hace que tratar su caracterización como una tarea de aprendizaje automático sea un problema abierto muy complejo que no se abordará. En este estudio definiremos unos atributos sencillos que se pueden asociar directamente a conceptos inteligibles para un instructor y que sea razonable esperar que aproximan las características de estas competencias de alto nivel. Uno de los puntos claves es la inclusión como atributos a utilizar en el modelo bayesiano de indicadores de competencia de la argumentación calificaciones realizadas por instructores humanos en los ensayos utilizados en el entrenamiento. De esta manera, el modelo bayesiano aprenderá criterios humanos de corrección de ensayos persuasivos más allá de los atributos propuestos inicialmente. En el presente trabajo, al utilizar el corpus Kaggle ASAP Phase 1, estos criterios son la efectividad en la expresión y el dominio del lenguaje, que son difícil de caracterizar completamente «a priori».

Teniendo en cuenta los recursos publicados disponibles, se propondrán indicadores de la riqueza de vocabulario, la precisión de términos utilizada y la claridad, entendida como legibilidad y coherencia del texto. Se definirán atributos más genéricos, como una métrica de estilo y una medida de la expresión de sentimientos en el texto, así como métricas de los argumentos detectados en el texto. Para algunas de estas métricas, como la legibilidad y argumentos utilizados existen modelos ya entrenados y código disponible públicamente. Para otros atributos, se extraen las métricas directamente a partir de otros atributos gramaticales, sintácticos y semánticos del texto (por ejemplo, PoS, «Part of Speech» o parte de la oración, estructura de dependencias sintácticas, raíz semántica) o entrenamos modelos de aprendizaje automático sobre estos a rasgos para generar atributos del ensayo argumentativo.

Durante el estudio realizado se ha considerado utilizar modelos entrenados en tareas de «Recognize Text Entailment», RTE (estimar si un fragmento de texto implica o contradice otro fragmento), para generar atributos más ricos del discurso argumentativo, que tuvieran en cuenta inferencias lógicas entre cláusulas argumentativas. Esta idea se abandonó, por un lado, porque requería utilizar algoritmos adicionales para segmentar el texto en posibles cláusulas argumentativas, y por otro, porque los primeros resultados que obtuvimos con algunos modelos RTE disponibles (el modelo de [85] implementado en la libraría AllenNLP) no eran muy prometedores.

El modelo bayesiano de indicadores de competencia en argumentación se validará estimando la capacidad del modelo de generalizar mejor, por un lado, la distribución de valores de los atributos y, por otro, el valor de las calificaciones de instructores humanos. Para ello se medirán el rendimiento del modelo al inferir una distribución de valores y los valores de las calificaciones para ensayos no utilizados en el entrenamiento (véase por ejemplo, el capítulo 7 de [49] para una discusión detallada sobre la capacidad de generalización de modelos en el aprendizaje estadístico). Consideramos que la validez del modelo bayesiano es un buen indicador de la validez de los atributos elegidos, pero de todas maneras, se propondrá un procedimiento de validación alternativo.

Hay que hacer notar, que al utilizar un número reducido de variables, algunas de ellas generadas por modelos con un margen de error relevante, podemos esperar que el modelo bayesiano de indicadores de la competencia en argumentación tenga menos precisión que modelos enfocados directamente a inferencia de calificaciones (por ejemplo, el estudiado en la referencia [77]). En este sentido, estamos admitiendo un sesgo a cambio de obtener un modelo más inteligible.

La elección de atributos hecha en este estudio no es la única posible. En esta primera aproximación al problema, no se han comparado resultados con diferentes conjuntos de atributos, pero puede ser un tema a explorar en el futuro.

2.3.2. Atributos lingüísticos y retóricos del ensayo argumentativo

A continuación se describirán los atributos definidos para cada ensayo argumentativo. Durante la extracción de estos atributos se han utilizado diferentes librerías públicamente disponibles para algunas tareas de PLN:

A lo largo del trabajo, las tareas de segmentación en palabras y símbolos ("tokenization", tokenización), etiquetado PoS ("Part-of-Speech"), lematización (extracción de la raíz léxica y morfológica de una palabra), análisis de dependencias sintácticas e identificación de entidades (nombres propios, organizaciones), se han realizado en Python 3, utilizando la librería SpaCy[52, 51] junto con el modelo ya entrenado en_core_web_lg (creado y publicado por Explosion AI estudio responsable del desarrollo de SpaCy). Estos modelo se construyen a partir de un corpus muy amplio (OntoNotes 5 [53]) para el que se construyen los vectores GloVe de cada palabra [90]. Este algoritmo de vectorización o «embedding» (de los que hay otras variedades, como word2vec [69]) capturan en espacios de dimensión más reducida que en los métodos clásicos de vectorización («bag-of-words», tf-idf, LSA) información contextual y semántica relevante. En estos modelos, se define la similaridad entre palabras o fragmentos de texto como la similaridad coseno del vector correspondiente, que para un fragmento sería el promedio de vectores de las palabras.

En principio, se podría estimar qué valores de los diferentes atributos serían más deseables, basándonos en teoría de la argumentación y los criterios del pensamiento

crítico [35], y utilizar el modelo bayesiano de indicadores entrenado en ensayos como validación. Este paso está fuera del alcance de este trabajo, en el que se aprenderá un modelo bayesiano de indicadores a partir de datos y se generarán recomendaciones de cómo debe cambiar el valor de los atributos para aumentar las calificaciones en los dominios considerados.

2.3.2.1. Fracción de palabras y palabras muy comunes ("stop words")

Se cuenta el número de palabras muy comunes en inglés («stop words») encontradas al tokenizar el ensayo, y se calcula la fracción respecto al número total de palabras y símbolos. Se ha utilizado la lista de «stop words» incluido en el Natural Language Toolkit (NLTK, [14]).

El número de «stop words» es una métrica general del texto que se ha considerado interesante. En otros estudios, por ejemplo, se ha utilizado para estimar el perfil de diferentes autores ([76][87]). Al utilizar la fracción se obtiene una métrica que no depende directamente de la longitud del texto, lo que en nuestra opinión, facilita la comparación de valores.

2.3.2.2. Fracción de signos de puntuación

Calculada como el número de signos de puntuación encontrados al tokenizar el ensayo dividido por el número total de palabras y símbolos. Es una métrica general que, al igual que la anterior, se ha utilizado en estudios estilométricos ([76]).

2.3.2.3. Fracción de raíces morfológicas diferentes

Calculada como el número de lemas (raíces léxicas y morfológicas) de nombres, adjetivos, verbos y adverbios encontrados al tokenizar el texto, dividido por el número total de palabras y símbolos. Con esta medida se intenta capturar parcialmente la riqueza de vocabulario, entendida como el número de términos diferentes utilizados (este tipo de medida de riqueza léxica se utiliza también en estilometría [76]). La lematización ayuda a descontar el efecto de la flexión de palabras (por ejemplo, conjugación verbal).

2.3.2.4. Fracción de entidades (nombres propios de personas, lugares, organizaciones)

Se utiliza la funcionalidad de identificar nombres de personas, lugares, organizaciones y otras entidades específicas implementada en SpaCy para identificar palabras o secuencias de palabras que especifiquen una entidad. En el caso del corpus Kaggle ASAP Phase 1, la mayoría de estas entidades están anotadas con una arroba

en la primera posición, pero aún así utilizamos la funcionalidad mencionada, para reducir el riesgo de no identificar alguna referencia. Como en las métricas anteriores, dividimos el número de entidades encontradas por el número de total de palabras y símbolos del ensayo.

Consideramos que esta métrica es interesante porque al hacer referencia a entidades específicas en el texto, el estudiante muestra un intento de ofrecer referencias o ejemplos, lo que potencialmente puede ayudar en la formación de argumentos. Por ejemplo, en la referencia [84] se utilizan indicadores de referencias a artículos legales encontradas en el texto para clasificar cláusulas argumentativas en texto legal, y las entidades correspondientes forman parte de la gramática de contexto libre que las autoras utilizan para parsear el argumento.

2.3.2.5. Fracción de palabras fuera de vocabulario

Se calcula como el número de términos que al tokenizar el ensayo no se reconocen como parte del vocabulario del modelo ni como entidad (personas, lugares, organizaciones), dividido por el número total de palabras y símbolos del ensayo.

Se utiliza esta métrica como una medida de la corrección en la escritura. El modelo utilizado tiene unas 685.000 palabras y símbolos, por lo que se asume que una palabra fuera de vocabulario es probablemente incorrecta.

Una validación de esta métrica como medida específica de incorrección ortográfica está fuera del alcance de este estudio. Sin embargo, creemos que es plausible, y algunas pruebas muy reducidas que se han realizado sugieren que esta métrica tiene buena precisión pero una exhaustividad baja, esto es, si marca una palabra como incorrecta, hemos observado que suele serlo, pero no marca como incorrectas algunas que en principio lo son. Este último resultado no tiene validez estadística.

2.3.2.6. Número de palabras y signos de puntuación

Se calcula como el número de términos obtenidos al tokenizar el texto. Es una estadística básica que se suele utilizar en otros trabajos ([84, 76]) y que en este estudio añadimos además para dar cuenta del efecto directo que el tamaño de un ensayo puede tener, ya que los atributos anteriores son fracciones independientes, en principio, de la longitud del texto.

2.3.2.7. Legibilidad

Se utiliza SpaCy [52, 51] junto con la extensión pará cálculo de métricas de legibilidad elaborada por Michael Holtzscher. La utilización de métricas de legibilidad es común, por ejemplo, en tareas estilométricas ([76]). En este estudio se utilizarán tres indicadores:

- Dale-Chall [28]: Fórmula empírica que tiene en cuenta la proporción de palabras difíciles (definidas en un vocabulario estándar definido por los autores) y la proporción de palabras por frase. Los valores están ajustados para que, en términos de nivel escolar en los Estados Unidos de América, estudiantes en niveles 7 y 8 entiendan fácilmente textos con índices entre 6 y 6.9, y estudiantes de niveles 9 y 10, textos con índices entre 7 y 7.9.
- Flesch-Kincaid - Legibilidad [28, 57]: Fórmula empírica que tiene en cuenta el número de palabras, frases y sílabas del texto. En términos de los niveles escolares, estudiantes en niveles 8 y 9 entiendan fácilmente texto con un índice de 70 a 60, y estudiantes de niveles 10 a 12, textos con índices entre 60 y 50. A menor, valor, más dificultad en la lectura.
- Flesch-Kincaid - Nivel («grade level») [28, 57]: Variación de la fórmula empírica anterior, en la que a cada texto se le asigna un nivel escolar apropiado. A mayor valor, más dificultad en la lectura.

2.3.2.8. Expresión de sentimientos

Se utilizará como atributo una medida de la polaridad del ensayo. La polaridad se puede entender como una estimación de los sentimientos y posición respecto a un tema del autor de un texto (véase [30] para una revisión reciente). Se utilizará un indicador numérico con valores mayores o menores que cero, para indicar un sentimiento positivo o negativo. Los valores más extremos indican una expresión de sentimiento o posición más acusada. La polaridad se ha utilizado en otros estudios, por ejemplo, como atributo para clasificar la función de una cláusula dentro de un argumento [74].

Aunque hay disponibles modelos bastante avanzados para estimar la polaridad de un fragmento de texto o frase (véase por ejemplo Stanford NLP - Sentiment Analysis), en este trabajo sólo se utilizará una medida agregada de hasta que punto el texto intenta expresar o causar una reacción emocional o expresar un estado de ánimo. Para ello, utilizaremos SentiWordNet [6] como un léxico, basado en WordNet, anotado con una estimación numérica de polaridad (positiva, negativa u objetiva).

Tras tokenizar el texto del ensayo y, para las palabras que aparecen en SentiWordnet calculamos:

1. El promedio de la polaridad anotada en SentiWordNet de todas las palabras del texto encontradas en el léxico. Considerando el valor de polaridad negativa y positiva en el rango [-1,1], obtenemos un único atributo numérico.
2. La diferencia entre el valor más extremo de la polaridad respecto al promedio calculado arriba. Con este indicador cuantificamos posibles picos de polaridad aislados.

Las palabras del ensayo que no aparecen en el léxico se ignoran.

2.3.2.9. Precisión promedio del texto

Se calculará la precisión promedio de un ensayo a partir de una medida de la precisión de los términos utilizados. Para obtener esta métrica, de forma sencilla se parte de WordNet ([121, 70]), un léxico que recoge definiciones de palabras y relaciones de sinonimia entre ellas. En WordNet, palabras sinónimas están agrupadas en grupos, que siguiendo la nomenclatura de WordNet llamamos «synset». Se acuerdo con [121], WordNet incluye relaciones de hiperonimia (términos de significado más general, por ejemplo, ser vivo es hiperónimo de planta) e hiponimia (término más específico, por ejemplo, cardo es hipónimo de planta) entre synsets. En este estudio, se considera más preciso un lema cuyo synset tenga menos hipónimos que hiperónimos, esto es, hay menos sysnsets más específicos que synsets menos específicos (así, se espera que cardo tenga menos hipónimos que planta).

El atributo de precisión para un ensayo será el promedio del número de hiperónimos dividido por la suma del número de hiperónimos e hipónimos calculado cada uno de los nombres, adjetivos, verbos y adverbios del ensayo cuyo lema se encuentra también en WordNet.

Como rasgos adicionales de la precisión en la expresión utilizada en un ensayo, utilizaremos las siguientes métricas:

- Fracción de palabras que se refieren a números.
- Fracción de palabras que se refieren a URLs.

En nuestra opinión, el uso de números y URLs indican un intento de establecer hechos o dar alguna referencia, y por tanto acotar el mensaje.

2.3.2.10. Coherencia del texto

Se sigue la referencia [62], en la que los autores proponen definir la coherencia de un texto como una calificación numérica o ranking entre textos que indique que un texto es más o menos coherente que otros. La contribución clave de los autores es utilizar la estructura retórica del texto para definir este ranking.

Como se mencionó en la introducción 1.3.2.3, se han propuesto diferentes modelos que describen como los fragmentos de un discurso ("Elementary Discourse Units", EDU) se relacionan en un discurso para implementar su función retórica (por ejemplo, explicar, convencer o persuadir). Los autores [62] utilizan el modelo PDTB [137] para definir una matriz documento-rol retórico (detalles en 2.1), a partir de la que calculan el número de transiciones entre roles, especificando un número máximo de roles. Se incluyen transiciones a/desde "no rol" para construir un vector de transiciones de dimensión fija para todos los documentos y adicionalmente separan las transiciones que incluyen términos que aparecen por encima (salientes) o por debajo de un valor umbral. Los autores proponen esta separación siguiendo trabajos

anteriores en medida de coherencia de textos ya que puede ofrecer mejores resultados. El vector de transiciones retóricas salientes y no salientes de un documento se normaliza para poder interpretarlo como un vector de probabilidades.

Los autores [62] parten de un corpus (artículos del Wall Street Journal) y generan una serie de textos que se consideran menos coherentes cambiando aleatoriamente el orden de las frases. Entrenan un modelo de máquina de soporte vectorial para aprender un ranking en el que un documento original es más coherente que su versión desordenada. Básicamente este tipo de aprendizaje se puede intentar reducir ([55]) a entrenar un SVM de núcleo lineal como clasificador binario sobre pares formados por documentos con una relación de rango conocida (en nuestro caso, un documento y su versión desordenada), utilizando como atributos la diferencia de los vectores de cada documento.

Al entrenar un clasificador SVM de núcleo lineal binario, se especifica un parámetro de margen C, y se resuelve el siguiente sistema de ecuaciones para un vector normal al hiperplano separador de menor riesgo w y la intersección del hiperplano con cero, b. Si $x_i, y_i, i = 1, ..., n$ son los vectores diferencia entre pares de documentos y la etiqueta (1 o -1) para n instancias de entrenamiento respectivamente:

$$\min_{w,b,\zeta} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \zeta_i$$

con la restricciones $y_i(w^T x_i + b) \geq 1 - \zeta_i$ y $\zeta_i \geq 0 \forall i = i, ...n$

El producto escalar $w^T x_i$ define la proyección del vector diferencia entre pares de documentos a la normal del hiperplano separador entre documentos conocidos de mayor y menor coherencia. Se utilizará este producto escalar como grado de coherencia de un documento. Cuanto más alto, más coherente es un documento.

Se ha seguido de cerca el trabajo [62], con varias diferencias:

1. Se ha utilizado el esquema RST de estructura retórica [65]. Consideramos de interés variar el modelo retórico y comprobar si obtenemos precisiones similares en nuestro experimento.
   a) El modelo RST estructura el discurso como un árbol, en el que cada fragmento de texto hace el papel de núcleo o satélite (véase la figura 2.1).
2. Se han utilizado el código Python y el modelo entrenado hechos públicos por los autores de las referencias [134],[133] para parsear la estructura RST de cada ensayo. Los autores entrenaron un modelo de segmentación y búsqueda de árboles en el corpus RST Discourse Treebank, y reportan métricas F1 del 85.7 % para la segmentación en EDUs, 71 % para estimar la nuclearidad (esto es, si el EDU juega el papel de núcleo o satélite en la relación retórica y un F1 promedio alrededor del 35 %.

	Lema 1	Lema 2	Lema 3	Lema 4
Frase 1	No aparece	Rol retórico 4	Rol retórico 4	No aparece
Frase 2	Rol retórico1	No aparece	Rol retórico 1	No aparece
Frase 3	Rol retórico1, Rol retórico2	No aparece	Rol retórico 1	Rol retórico 1
Frase 4	No aparece	No aparece	Rol retórico 3	Rol retórico 3

no->Rol1	Rol1->Rol1	Rol1->Rol2	Rol1->no	Rol2->no	Rol1->Rol3	Rol4->no	Rol4-> Rol1
2	2	1	1	1	2	1	1

Cuadro 2.1.: Ejemplo de matriz documento-rol: Una fila por frase del documento, y una columna para cada término de interés, que siguiendo de cerca [62] serán nombres, adjetivos, verbos y adverbios lematizados. Cada celda contiene el rol retórico correspondiente, entendiendo que en una frase puede haber una o varios EDUs. A partir de esta matriz se calcula el número de lemas que efectúan una transición de rol en el documento al pasar de una frase a la siguiente. Mostramos como ejemplo las transiciones de longitud dos que no son cero

3. Se han generado datos sintéticos a partir de una unión de los corpus Araucaria, AAEC, unos 30 artículos y editoriales del corpus de Brown [38], junto con unos 20 artículos de dos ensayistas conocidos. La razón es buscar textos que incluyan ensayos persuasivos de estudiantes, junto con artículos de prensa y texto de buena calidad con el fin de cubrir una gama de casos amplia. Esperamos que esta elección ayude a transferir el modelo a otro tipo de ensayos más fácilmente.

4. Se han estandarizado los documentos vectorizados para que la desviación estándar de cada atributo sea 1. Este es un paso clave para que el clasificador binario proporcione buenos resultados y es un escalado directamente aplicable a otros documentos no utilizados en el entrenamiento o test.

5. Se ha utilizado la librería **scikit-learn** [88] para entrenar una maquina de soporte vectorial con núcleo lineal que separe los documentos originales de su versión desordenada. Para ello, se entrena como clasificador binario sobre el conjunto de vectores diferencia entre el vector del texto original y el vector del texto desordenado, con etiquetas 1 y -1 según el primer vector corresponda al documento ordenado o no.

6. Salvamos el vector normal del SVM y la escala utilizada en la estandarización durante el entrenamiento para calcular el grado de coherencia en nuevos documentos.

2.3.2.11. Estilo

Se busca una caracterización muy simplificada del estilo, entendido como «personalidad literaria» ([67]). Se sigue la referencia [41], en la que el autor estudia diferencias

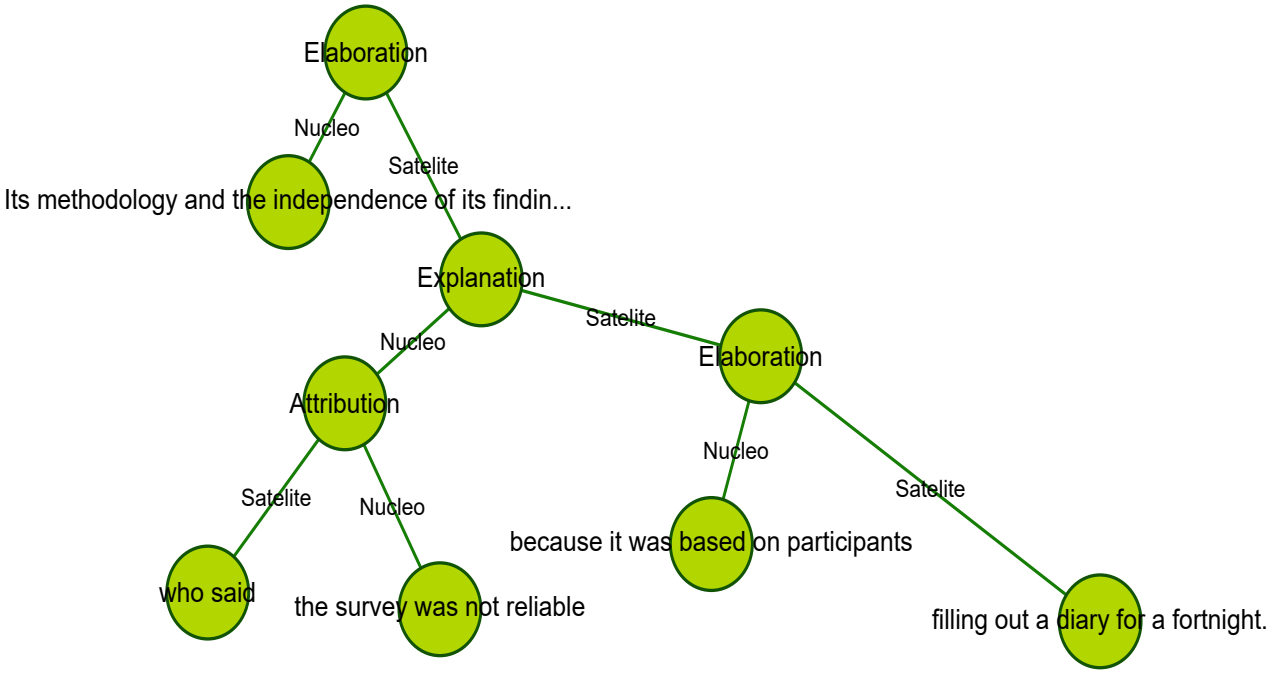

Figura 2.1.: Estructura RST del fragmento de texto "Its methodology and the independence of its findings were questioned by InterChurch Gambling Taskforce spokesman Reverend Tim Costello, who said the survey was not reliable because it was based on participants filling out a diary for a fortnight.", extraído de nodeset11, AraucariaDB

estilísticas en escritores utilizando un clasificador sobre una serie de atributos sintácticos, gramaticales y léxicos. El autor vectoriza fragmentos de texto de dos autores diferentes contando las frecuencias de etiquetas PoS, tri-gramas de PoS, las diferentes producciones sintácticas (sustituyendo nodos terminales por PoS) y relaciones semánticas, y utiliza una máquina de soporte vectorial, SVM, como clasificador binario para inferir la autoría de cada texto. El autor reporta una métrica F1 mayor del 95 % para textos largos y alrededor del 85 % para textos cortos (5 frases). Una ventaja de este método es que no hay dependencia directa del vocabulario utilizado, con lo que es posible que funcione correctamente para textos de diferentes temas.

En este estudio, en lugar de identificar la autoría de cada texto, vamos a definir un marco de referencia para el estilo de escritura de un ensayo:

1. Se vuelve a utilizar el corpus con el que hemos entrenado el modelo de coherencia 2.3.2.10, pero se han excluido los documentos de Araucaria para tener dos tipos de texto: ensayos de estudiantes y ensayos o editoriales más profesionales.
2. Se han extraído los atributos de los documentos del corpus siguiendo [41], con algunas variaciones:
   - Tri-gramas PoS ("Part of Speech", parte de la oración), esto es, como se presentan las diferentes etiquetas PoS en el texto, agrupadas en segmentos

de tres etiquetas. Se excluyen símbolos, palabras desconocidas, símbolos de puntuación y espacios.

- El esquema de anotación que sigue el modelo utilizado está descrito en PoS Annotation, y es una variación del utilizado en el corpus Penn Treebank.
- Ejemplo: "I-PRP am-VBP the-DT one-NN that-WDT drives-VBZ" genera los tri-gramas (PRP, VBP, DT), (VBP, DT, NN), (DT, NN, WDT) y (NN,WDT, VBZ).
  - PRP, VBP, DT, NN, WDT, VBZ son ejemplos etiquetas PoS en la versión OntoNotes 5 [53] del sistema de etiquetas del corpus Penn Treebank ([66]). Corresponden a pronombre personal, verbo (no en tercera persona) singular presente, determinante, nombre, determinante de tipo «Wh», y verbo en tercera persona singular presente respectivamente.

- El número de veces que se presenta cada etiqueta PoS en el texto.
- Las relaciones de dependencia sintáctica de un solo nivel. Se indica la etiqueta PoS del padre y del descendiente, junto con la relación de dependencia detectada. Nótese que el modelo utilizado realiza un análisis sintáctico utilizando una gramática de dependencias en lugar de un árbol sintáctico ("constituency tree"). La referencia [41] utiliza relaciones semánticas, pero por simplificar, consideraremos las dependencias sintácticas como una aproximación que se puede justificar si nuestro modelo de estilo nos permite separar de manera razonable ensayos de estudiantes respecto a textos periodísticos/ensayos más profesionales.
  - Ejemplo: $VBP \overset{nsubj}{\longrightarrow} PRP$ (relación verbo, sujeto, pronombre personal).
- Las producciones sintácticas: padres y todos sus descendientes a un único nivel. No se tiene en cuenta el tipo de dependencia y, como antes, solo consideramos las etiquetas PoS.
  - Ejemplo, $VBP \longrightarrow PRP, NN$ (se tiene en cuenta el orden el que se presenta cada parte de la oración).

3. Se vectorizan los atributos usando la frecuencia con la que tri-gramas, relaciones de dependencia y producciones sintácticas se presentan en cada documento, junto con la frecuencia de cada etiqueta PoS. Estas frecuencias se normalizan por separado para cada documento.
4. Se entrena un clasificador binario en los documentos elegidos que asigne etiquetas a diferentes ensayos de estudiantes y textos profesionales. Se utiliza como modelo clasificador un bosque aleatorio (Random Forest, RF [19]) implementado en **scikit-learn**. Un RF puede usarse como un clasificador robusto que suele dar buenos resultados en diferentes escenarios. Se basa en utilizar

un conjunto de árboles de decisión, que pueden tratar diferentes conjuntos de atributos y cuyos resultados se combinan para dar un resultado final.

5. Un RF entrenado como clasificador puede estimar una probabilidad de que una instancia pertenezca a una clase. Para ello se calcula una probabilidad por árbol (fracción de instancias de la misma clase en una hoja) y se promedia. Adicionalmente, un RF proporciona un ranking de relevancia de cada atributo: en un árboles de decisión, los atributos que aparecen más cerca de la raíz deciden sobre más instancias de datos. En un RF se estima la fracción de muestras en las que cada atributo decide (en cualquiera de los árboles del bosque) para estimar su importancia.
6. Una vez entrenado el bosque aleatorio, para cada nuevo documento, se calcula la probabilidad de pertenecer a la clase de ensayos profesionales como un medidor de estilo.

En este estudio se utiliza un único indicador de estilo, pero la definición puede ampliarse a más dimensiones si se introducen grupos adicionales de documentos con estilos que consideramos bien definidos (por ejemplo, texto científico). En este caso se podrían utilizar un bosque aleatorio multi-clase y la métrica F1 ponderada, para tener en cuenta la descompensación en el número de instancias de cada clase disponible en el entrenamiento, para estimar hasta que punto son discriminativos estos indicadores de estilo.

2.3.2.12. Rasgos de la argumentación

Se definen una serie de indicadores sencillos de los argumentos utilizados en el ensayo. Para ello es necesario identificar las cláusulas argumentativas y la estructura que forman, a través de las relaciones entre cláusulas. En nuestro caso se ha utilizado directamente el código Python y modelos pre-entrenados de minería de argumentos descritos en la referencia [78]. Este código, descrito en la introducción, 2, se basa en un modelo monológico de representación de argumentos publicado en la referencia [86].

En nuestro estudio se utilizó el modelo que parametriza los potenciales del grafo de argumentación con una máquina de soporte vectorial. El modelo devuelve las cláusulas argumentativas, etiquetadas como :

1. Hecho (“Fact”). Uno o varios conceptos para los que hay acuerdo en dar por sentados.
2. Testimonio (“Testimony”). El relato de una persona, posiblemente de un hecho.
3. Opinión (“Value”). Básicamente una creencia u opinión de una persona.
4. Llamamiento a la acción (“Policy”)
5. Estas cláusulas se pueden defender con evidencia, que serán testimonios o referencias (“Reference”)

El modelo define enlaces entre proposiciones o entre proposiciones y evidencia para definir argumentos. En el modelo implementado no hay restricciones al tipo de grafo que se forma, aunque se añaden pesos para reducir la incidencia de enlaces que los autores consideran improbables ([78]).

En nuestro caso, utilizaremos unas métricas básicas y fácilmente inteligibles: Número de hechos, testimonios, opiniones, llamamientos a la acción, referencias y enlaces que se han encontrado en el ensayo. Estos atributos son más simples que los utilizados en la referencia [77]. En esta referencia los autores buscan mejorar resultados de predicción de calificaciones de ensayos persuasivos, utilizando, entre otros, el corpus Kaggle ASAP Phase 1 que se ha utilizado en este trabajo. Sin embargo, nuestro estudio busca aprender un modelo bayesiano de indicadores de la competencia en argumentación que permita explicar la calidad de ensayos persuasivos a estudiantes e instructores, en lugar de centrarse en predicción de calificaciones. En este caso, consideramos que la inteligibilidad es una ventaja que compensa un posible peor rendimiento en tareas más específicas de predicción.

2.3.3. Validación de atributos

Con el fin de validar objetivamente que los atributos elegidos son suficientemente expresivos para la tarea a realizar, se utilizan para entrenar un modelo de regresión sobre los textos del corpus Kaggle ASAP Phase 1, utilizado para construir el modelo bayesiano de indicadores de la competencia en argumentación. Utilizando los atributos descritos en la sección 2.3.2 como variables independientes, se ajustarán dos modelos de regresión diferentes, uno para cada una de las dos calificaciones anotadas en el corpus (agregada para todos los evaluadores). Para el primer dominio (efectividad en la expresión), es un número entre 1 y 6, y para el segundo (dominio del lenguaje), entre 1 y 4. Aunque las calificaciones anotadas son enteras y el modelo de regresión produce números reales, el resultado sigue siendo útil.

Como regresor se utilizará un bosque aleatorio (RF, véase la referencia [19]). De manera semejante a su uso como clasificador, un RF va añadiendo árboles que generan etiquetas numéricas en lugar de categóricas, y combina los resultados. En este trabajo se utilizará el error cuadrático medio (respecto al número de documentos) como medida de hasta que punto podemos esperar que, a partir de los atributos elegidos, un modelo pueda aprender a evaluar un ensayo argumentativo en uno de los dominios considerados (efectividad en la expresión y dominio del lenguaje). Como resultado añadido, un bosque aleatorio permite estimar la relevancia de cada atributo de la misma manera que cuando se utiliza como clasificador (2.3.2.11).

2.4. Modelo bayesiano de indicadores de la competencia en argumentación

En este estudio se utiliza una red bayesiana para modelar la distribución de probabilidad de los atributos o variables de un ensayo argumentativo corto junto con la evaluación sobre uno o varios dominios. Aunque este tipo de modelos pueden no ser los más precisos a la hora de clasificar o efectuar regresiones, suelen ser robustos y además, la estructura de grafo acíclico dirigido nos permite visualizar la correlación entre variables muy fácilmente, y ayuda a los usuarios a discutir las posibles relaciones causa-efecto que la red pueda describir.

El modelo se entrenará en dos pasos a partir de un subconjunto de ensayos del corpus Kaggle ASAP Phase 1 tema 2, vectorizados con los atributos descritos en 2.3.2 junto con las calificaciones para efectividad en la expresión (dominio 1) y dominio del lenguaje (dominio 2) agregadas por evaluador. Utilizaremos el algoritmo "Max-Min Hill Climbing" (MMHC, [119]) para aprender, primero, la estructura de la red bayesiana. Luego se entrenará esta red bayesiana como un modelo paramétrico en el que las probabilidades condicionadas son gaussianas, mediante un proceso de inferencia en el que, a partir de la distribución "a priori" de los parámetros del modelo, generaremos una distribución "a posteriori" de los mismos parámetros al tener en cuenta los datos que se quieren explicar.

El aprendizaje de la estructura de una red bayesiana a partir de datos, es, en general, un problema complejo, en el que probablemente se pueden encontrar diferentes soluciones óptimas locales diferentes. Se podrían intentar aplicar reglas heurísticas para reducir la dificultad de la búsqueda, como, por ejemplo, imponer que las calificaciones agregadas no tengan descendientes o que algunos atributos básicos no tengan padres. Sin embargo, consideramos que es mucho más interesante aprovechar esta oportunidad para reducir nuestro sesgo y generar una red bayesiana únicamente a partir de los datos disponibles.

Teniendo esto en cuenta, una posible mejora futura sería generar diferentes modelos de indicadores (con diferentes estructuras) y utilizar un sistema de votación para combinar resultados.

2.4.1. Aprendizaje de la estructura de red bayesiana

El algoritmo MMHC tiene dos pasos principales [119]:

1. MMPC (Max-Min Parent Children): Intenta construir un grafo no dirigido con todas las variables del problema que preserve las relaciones de independencia condicional de los datos:
   a) Se define una función que define la asociación entre dos variables aleatorias X, Y respecto al conjunto de variables Z, de modo que esta sea cero si $X \perp Y \mid Z \iff P(XY|Z = z) = P(X|Z = z)P(Y|Z = z)$.

b) Para cada variable X se parte de un conjunto de candidatos a padre e hijo (CPC) vacío, y se buscan las variables con máxima asociación a X respecto a CPC para añadirlas a CPC si su asociación no es cero, en un proceso iterativo.

c) En un segundo paso se eliminan de CPC las variables A que cumplan $X \perp A \mid S, S \subseteq CPC$.

d) Una vez obtenidos los candidatos a padre o hijo de cada variable, se sigue un proceso de limpieza de falsos positivos, en el que, para cada variable X se revisan sus candidatos $A \in CPC$ y si $X \notin CPC(A)$, se elimina esta variable del conjunto de candidatos de X.

2. MMHC (Algoritmo de escalada simple/"Greedy Hill-Climbing"):

a) Ejecutar el paso MMPC para todas las variables para calcular el grafo no-dirigido máximo.

b) Definir una métrica para evaluar la calidad con la que una red bayesiana describe los datos.

c) Partiendo de un grafo vacío, utilizar las operaciones añadir-enlace dirigido, eliminar enlace-dirigido e invertir-dirección de enlace para buscar redes que mejoren la métrica:

1) Solo se pueden añadir enlaces que unan nodos enlazados en el grafo no-dirigido obtenido a partir de MMPC.

2) A partir de la red inicial vacía, aceptar la primera operación que mejore la métrica de evaluación de la red bayesiana.

Se ha implementado el algoritmo en Python 3. Teniendo en cuenta que en la práctica la mayoría de los atributos o variables son continuas, los puntos más complicados son:

1. Definir una medida de asociación en el paso MMPC que sea cero en caso de independencia condicional: Para variables continuas, el cálculo no es trivial, pero se ha utilizado una librería de software pre-existente (Tigramite). Esta libraría implementa el algoritmo descrito en [102] para calcular un test estadístico de independencia condicional para grupos de variables continuas, basado en un criterio de información condicional mutua de dos grupos de variables aleatorias X, Y respecto a un tercer grupo Z, $I(XY|Z)$. El punto clave de este nuevo algoritmo es que en el caso continuo no necesita asumir relaciones lineales entre variables, lo que lo hace más versátil.

a) La salida de este algoritmo es el p-valor para la hipótesis nula $I(X, Y|Z) = 0 \iff X \perp Y \mid Z$. Este p-valor es la probabilidad de que, bajo la condición de que la hipótesis nula sea cierta, el valor de una estadística sea mayor que el valor observado. Si este p-valor es pequeño (en nuestro caso menor que 0.05) se puede rechazar la hipótesis nula con una probabilidad del 0.95.

b) En el código que se ha creado se utiliza este p-valor para definir un grado de asociación. Si es mayor que 0.05, se asume que no hay asociación, esto es, no se puede rechazar la hipótesis nula de independencia

2. En el paso de escalada simple, para evaluar la métrica de calidad de una red bayesiana en cada paso:

   a) Se modela una red bayesiana paramétrica lineal. $\mathcal{N}$ será una distribución de probabilidad normal:

   $$P(X|padres(X)) = \mathcal{N}(X, \mu = \mu_{0x} + \sum_{p \in padres(X)} \mu_{px} X_p, \sigma_x)$$

   a) Se definen probabilidades a priori para los parámetros μ_{xy}, σ_x. Se asume que son variables independientes y se utiliza una distribución normal para μ_{xy}, con media cero y desviación relativamente grande. Para σ_x se utiliza la distribución de Cauchy con dominio positivo. De esta manera, aceptamos que los valores medios puedan ser positivos o negativos, en un rango razonable, y las desviaciones σ_x serán siempre positivas, con mayor probabilidad de tener valores grandes que si hubiéramos impuesto una distribución normal con dominio positivo.

   b) A partir de los atributos de los ensayos utilizados en el entrenamiento, se realiza un proceso de inferencia numérica para, a partir del teorema de Bayes, obtener las probabilidades posteriores de los parámetros μ_{xy}, σ_x condicionadas a los datos observados X_{obs}. La probabilidad conjunta de observar los datos es una constante irrelevante en la presente tarea:

   $$P(\mu, \sigma|X_{obs}) = P(X_{obs}|\mu\sigma)P_{prior}(\mu\sigma)/P(X_{obs}) \approx P(X_{obs}|\mu\sigma)P_{prior}(\mu\sigma)$$

   c) Se necesita una métrica de calidad del modelo que permita comparar las diferentes redes que generamos durante el algoritmo de escalada simple, con el fin de decidir si seleccionamos la nueva red bayesiana generada o continuamos con al red bayesiana salida del paso anterior.

      1) En la literatura se proponen diferentes medidas de calidad del modelo:

         - BIC, Criterio de Información Bayesiano, DIC, Criterio de desviación de la información o AIC, Criterio de Información de Akaike [108], que estiman la información perdida por el modelo (con un factor penalizador de la complejidad) a partir de los parámetros que proporcionan la máxima verosimilitud $P(X_{obs}|\mu_{max}\sigma_{max})$ ("maximum likelihood") y las dimensiones estimadas del modelo. Si realizamos inferencia bayesiana, calcular la máxima verosimilitud o estimar el DIC puede requerir bastantes cálculos adicionales. Además, desde el punto de vista metodológico, estos cálculos se basan en estimaciones de la verosimilitud para un valor dado

de los parámetros del modelo, por lo que se pierde toda la información sobre su distribución «a posteriori» aprendida durante la inferencia.

- Recientemente se han propuesto métodos eficientes para estimar métricas que no requieren calcular una máxima verosimilitud, y por lo tanto se ajustan más al paradigma bayesiano. Siguiendo la referencia [126]:
  - Se parte de un modelo bayesiano paramétrico $P(X|\theta)$ para las variables aleatorias X con parámetros θ, y el resultado de la inferencia $P(\theta|X_0)$, donde X_0 son datos ya observados. Una métrica de calidad del modelo es la densidad puntual predictiva esperada («pointwise») para un conjunto de nuevas observaciones $Y = \{y_1, ..., y_n\}$, que es el valor esperado (respecto a la distribución de probabilidad real para los datos, que es desconocida) de la probabilidad (dada por el modelo) de que se observen estos nuevos datos Y si ya se ha observado X_0. Cuanto más alta sea esta densidad, *lpd*, más poder predictivo tiene la probabilidad modelada.

$$elpd = \sum_{i=i}^{n} \int P_{real}(y_i) \ln P(y_i|X_0) dy_i$$

  - Se sabe que $P(y_i|X_0) = \int P(y_i|\theta)P(\theta|X_0)d\theta$. Estos valores se pueden estimar para cada y_i como el promedio de la función $P(y_i|\theta)$ en un muestreo «a posteriori» de θ generado por $P(\theta|X_0)$. El valor esperado sobre P_{real} a su vez se estima como el promedio de $\ln P(y_i|X_0)$ utilizando valores de $Y = \{y_1, ..., y_n\}$ no utilizados durante el entrenamiento (por ejemplo, se puede utilizar validación cruzada o reservar datos para la validación).
  - En la referencia [126] proponen métodos eficientes para estimar medidas de la densidad puntual predictiva durante el proceso de inferencia. Estas métricas serían la validación cruzada con un único registro excluido ("Leave One-Out cross validation", LOO) o utilizando el "Widely Available Information Criteria/Watanabe Akaiki Information Criteria". Entre otros métodos, los autores implementan un algoritmo para estimar LOO utilizando muestreo de importancia suavizado utilizando una distribución Pareto ("Pareto smoothed Importance Sample"). Estos algoritmos requieren pocos cálculos adicionales y están implementados en diferentes paquetes o librerías enfocados a la estadística (R [94], Stan, PYMC3).
  - Sin embargo, estos algoritmos son sensibles a la calidad del modelo o la presencia de muestras aisladas («outliers»), y, a la

hora de optimizar diferentes redes bayesianas, es fácil que den valores poco fiables. Por esa razón, utilizaremos como métrica la densidad predictiva logarítmica estimada sobre un conjunto de datos de test, no utilizados en el entrenamiento, tal como recomienda [126]. Para ello, se generaran los atributos de discurso para un conjunto nuevo de ensayos anotados $\{y_1, ..., y_n\}$. Para cada red bayesiana entrenada por inferencia, generaremos N_s muestras de los parámetros $\theta_s, s = 1..., N_s$ utilizando la distribución de probabilidad posterior $P(\theta|X_0)$ y estimamos la densidad predictiva logarítmica como

$$lpd = \sum_{i=1}^{n} \ln(\frac{1}{N_s} \sum_{s=1}^{N_s} P(y_i|\theta_s))$$

- En esta aproximación, $elpd = lpd/n$. Como n, el número de documentos utilizados en la validación, es fijo a lo largo de la experimentación, se utilizará la métrica *lpd* para comparar documentos. Aunque requiere más cálculos, esta métrica debería proporcionar valores más robustos que LOO o WAIC [126].

Para realizar la inferencia y calcular la densidad predictiva logarítmica se utiliza la librería PYMC3 de Python [103]. Con esta libraría se pueden crear modelos usando el paradigma de programación probabilista, realizar inferencia con diversos algoritmos (Markov Chain Monte Carlo - MCMC, No-U-Turn Sampler - NUTS [29]), calcular métricas de validación y generar muestreos a partir de las probabilidades «a posteriori» inferidas.

Como se indica en [126], la densidad predictiva no es la única métrica posible. Se podría utilizar, por ejemplo, una métrica de clasificación, pero la métrica seleccionada (densidad predictiva) se adapta mejor al propósito del trabajo, que es modelar la densidad de probabilidad conjunta de nuestros atributos

2.4.2. Aprendizaje de los parámetros de la red bayesiana

Una vez completado el paso anterior, se re-entrena la red bayesiana como un modelo gaussiano lineal con los datos utilizados durante los pasos de escalada simple para el entrenamiento del modelo bayesiano. Se aumenta el número de iteraciones de "burn in" para aumentar la probabilidad de conseguir una convergencia correcta del proceso. Se reutilizan también los datos de test en tareas de validación.

Una vez alcanzada la convergencia, se realiza un muestro utilizando las probabilidades a posteriori y se chequea si la distribución de la media para cada una de las variables muestreadas se centra en la media de los datos de test. Este criterio cualitativo indica que el modelo entrenado aproxima la distribución real de datos.

Los parámetros finales del modelo serían los valores medios de su distribución a posteriori. Inspeccionando esa distribución y revisando si hay anormalidades (distribuciones multi-modales, divergencias) se puede realizar una validación adicional de que el proceso de inferencia ha acabado con éxito.

2.4.3. Aplicación del modelo bayesiano de indicadores de la competencia en argumentación

Una vez completado el proceso de aprendizaje, se ha encontrado una distribución de probabilidad conjunta para los atributos del discurso más las evaluaciones agregadas para los dos dominios. A partir de la densidad de probabilidad conjunta, se puede estimar el valor de un atributo de un nuevo documento a partir del valor de los otros atributos. Para ello se utiliza la formula siguiente, que corresponde al máximo «a posteriori» (MAP) del atributo una vez se han observado el resto de datos:

$$\text{Predicción}(E) = \arg\max_{E} P(E|X) = \frac{P(E,X)}{P(X)}$$

La densidad de probabilidad conjunta se obtiene a partir de la distribución de probabilidad «a posteriori» de los parámetros de la red una vez observados los datos de entrenamiento X_{entr}:

$$P(E,X) = \int P(E,X|\theta)P(\theta|X_{entr})d\theta$$

Esta distribución de probabilidad se estima mediante un muestreo $\theta_s, s = 1..., N_s$ de las distribuciones «a posteriori» $P(\theta|X_{entr})$ como:

$$P(E,X) = \frac{1}{N_s}\sum_{s=1}^{N_s} P(E,X|\theta_s)$$

En este estudio se utilizan las predicciones de las calificaciones agregadas para la efectividad de la expresión (dominio 1) y el dominio del lenguaje (dominio 2) con el fin de validar el modelo obtenido.

Otra aplicación del modelo es la generación de recomendaciones básicas: Si $x_i, i = 1, ...$ son valores de los atributos del discurso, sin incluir la calificación para un dominio, e, podemos calcular el valor esperado de esta calificación dados los atributos anteriores:

$$\mathbb{E}[e]_{X=x} = \int eP(e|X=x)de$$

Interesa recomendar acciones que aumenten este valor esperado $\mathbb{E}[e]_{X=x}$. Si calculamos el gradiente respecto a los atributos del ensayo

$$\nabla_x \mathbb{E}[e]_{X=x} = [\partial_{X_1}\mathbb{E}[e]_{X=x}, \partial_{X_2}\mathbb{E}[e]_{X=x}, \ldots]$$

los componentes positivos mayores corresponden a los indicadores que, al crecer, hacen aumentar más rápidamente el valor esperado de las calificación. Los componentes negativos del gradiente indican atributos cuyo valor debería reducirse. Para densidades de probabilidad generales, esta es una aproximación local, válida cuando el valor de los atributos es cercano a los valores usados en el cálculo del gradiente. Como se tratará durante la discusión de resultados, en el modelo gaussiano considerado en este trabajo este gradiente es constante, por lo que solo generará recomendaciones genéricas.

Para obtener recomendaciones adaptadas a cada uno de los ensayos, habría que recurrir a técnicas de recomendación más generales [16]. Por ejemplo, se podrían aplicar técnicas basadas en buscar ensayos similares con mejor calificación, o considerar otro tipo de información adicional, tales como preferencia de estudiantes o repositorios de ensayos estándar para aplicar algoritmos de recomendación basados en contenido, colaborativos o híbridos. Estás recomendaciones más generales están fuera del alcance de nuestro trabajo.

2.5. Grafo de argumentación colaborativo como resumen de los argumentos utilizados en varios ensayos

Durante la generación de atributos de los ensayos analizados, se ha extraído la estructura argumental utilizando el modelo de minería de argumentos [78]. El modelo permite identificar los fragmentos de cada texto que hacen el papel de hecho (“Fact”), testimonio (“Testimony”), opinión (“Value”), llamamiento a la acción (“Policy”) y referencia (“Reference”).

Por otro lado, es razonable pensar que en un grupo de ensayos sobre el mismo tema habrá argumentaciones muy similares. Si se agrupan cláusulas argumentales que expresan la misma idea (esto, es que son paráfrasis una de otra), se consigue una herramienta para resumir y ayudar a visualizar los argumentos utilizados por el grupo de estudiantes. En este estudio se propone crear un grafo (grafo de argumentación colaborativo) para esta visualización, lo que ayudaría captar rápidamente los argumentos más utilizados, así como su popularidad. Una visión de las relaciones entre argumentos podría ser una herramienta para dirigir una discusión sobre la calidad de estos y como podría mejorarse.

En principio se puede aplicar esta idea sin necesidad de recurrir a la paráfrasis, utilizando una medida de similaridad de texto. Sin embargo, al utilizar una medida de similaridad basada en palabras o lemas comunes es fácil perder detalles como, por ejemplo, si en una frase se apoya o ataca una idea. Hay ideas alternativas (RTE, o directamente algoritmos de identificación de lógica natural), pero para este trabajo se considera que la identificación de paráfrasis es una tarea que se puede tratar de

una manera no demasiado complicada, y que resulta útil a la hora de construir un grafo de argumentación colaborativo.

2.5.1. Aprendizaje de la paráfrasis

Se ha entrenado un bosque aleatorio ("random forest") como clasificador binario para identificar si un par de frases están relacionadas por paráfrasis o no. Para el entrenamiento se ha utilizado el Microsoft Research Paraphrase Corpus [129].

1. El corpus tiene parejas de frases etiquetadas con 0 (no hay paráfrasis) o 1 (hay paráfrasis). El corpus tiene documentados los criterios de anotación.
2. Para cada sentencia argumentativa se ha definido una serie de atributos siguiendo de cerca la referencia [107]. Cada atributo se define para un par de fragmentos de texto, A y B.
   a) Similaridad entre A y B. Calculada como el coseno del angulo que forman el vector GloVe de A y B (evaluado como el promedio de los vectores para cada término del texto). El vector GloVe se obtiene con la librería SpaCy [51, 52] y el modelo en_core_web_lg.
   b) Si $L(A)$ y $L(B)$ son los lemas (raíces morfológicas) diferentes en A y B, se construyen los atributos $|A|, |B|, |A - B|, |A - B|/|B|, |B - A|, |B - A|/|A|, |A \cup B|, |A \cap B|$, donde $|A|$ indica la cardinalidad de A.
   c) Longitud de la subcadena común más larga. Se consideran únicamente lemas del texto.
   d) Índice de Jaccard, $\text{Jaccard}(S_1, S_2) = |S_1 \cap S_2|/|S_1 \cup S_2|$ para los siguientes conjuntos evaluados en A y B:
      1) Bi, tri y tetra-gramas de caracteres.
      2) Mono, bi y tri-gramas de "tokens" (palabras, incluyendo símbolos de puntuación, símbolos).
      3) Mono, bi y tri-gramas de etiquetas PoS.
      4) Mono, bi y tri-gramas de lemas (exceptuando pronombres).
   e) Diferentes medidas de similaridad semántica. Para ello, se vectorizan los fragmentos A y B construyendo un vocabulario $V = lemma(A) \cup lemma(B)$ a partir de todos los lemas en cualquiera de los dos fragmentos que sean también lemas en WordNet ([70, 121], excluyendo "stop words" y se calculan dos tipos de vectores, wo, sem_{ic} de dimensión $|V|$ para cada uno de los dos fragmentos de texto:
      1) Vector orden de palabras ("Word Order"): Posición en A de los lemas en el vocabulario. Para lemas del vocabulario no presentes en el texto, se indica la máxima similaridad de este lema con $V \cap lemma(A)$. Se

usa la similaridad Leacock-Chodorow sobre WordNet (longitud del camino más corto uniendo dos palabras en la taxonomía, con un factor que tiene en cuenta la profundidad de los sentidos de cada palabra).

2) Vector semántico ("Semantic vector"): Para cada palabra del vocabulario, si está en A, se indica un uno en la posición correspondiente, y si no de nuevo se anota la máxima similaridad entre la palabra del vocabulario y los lemas de A. Este vector se pesa con el producto del contenido de información (IC) de la palabra del vocabulario y los lemas de A considerados. El contenido de información se calculará como $IC(w) = 1 - \ln(n+1)/\ln(N+1)$ ([107]), con n el número de veces que el lema w aparece en WordNet y N el número de lemas en este corpus.

3) Dados los dos tipos de vectores $wo(A),\ wo(B),\ sem_{ic}(A),\ sem_{ic}(B)$, utilizaremos como atributo las siguientes similaridades:

$$Sim(wo) = 1 - \frac{\|wo(A) - wo(B)\|}{\|wo(A) + wo(B)\|}$$

$$Sim(wo)_{cos} = \frac{wo(A) \cdot wo(B)}{\|wo(A)\|\|wo(B)\|}$$

$$Sim(sem_{ic})_{cos} = \frac{sem_{ic}(A) \cdot sem_{ic}(B)}{\|sem_{ic}(A)\|\|sem_{ic}(B)\|}$$

f) Alineamiento de partes de la frase (Phrase-Entity Alignment): Se sigue el algoritmo descrito en [107], aunque ampliando el tipo de partes de frase que utilizamos y con medidas de similaridad diferentes.

1) Para A y B se consideran como "entidad" los términos (y su etiqueta PoS) que forman parte de una oración nominal ("noun chunk") o tienen dependencia sintáctica directa con un verbo.

2) Se construye una matriz $M_{AB}(n_A, n_B)$ donde cada posición indica la similaridad entre una entidad de A y otra de B (n indica el número de entidades y las dimensiones de la matriz inicial). Se define la similaridad entre dos entidades, $Sim_{pea}(a|b)$ a partir del promedio del máxima similaridad (calculada como el coseno de los vectores GloVe) entre cada token de una entidad a y todos los términos («tokens») con la misma función gramatical de la entidad b. La similaridad final se calcula como:

$$Sim_{pea}(A, B) = (Sim_{pea}(A|B) + Sim_{pea}(B|A))/2$$

3) Se define el paso $T : M_{AB}(n_A, n_B) \longrightarrow M_{AB}(n_A - 1, n_B - 1)$ a través de la eliminación de la fila y columna de la posición de mayor similitud $max(M_{AB}(n_A, n_B)$. Aplicamos T iterativamente hasta eliminar

todos los elementos de M. La similaridad final que consideraremos es:

$$Sim_{pea}(A,B) = \sum_{i=0,...,min(n_A,n_B)} max(M_{AB}(n_A - i, n_B - i))/max(n_A, n_B)$$

2.5.2. Construcción del grafo de argumentación colaborativo

Se parte de una selección de ensayos y las sentencias argumentativas encontradas utilizando el método descrito en 2.3.2.12. El proceso de construcción del grafo tiene dos pasos.

Agrupación en clases de equivalencia A partir de una muestra de ensayos y sus cláusulas argumentales:

1. Se consideran las combinaciones en parejas de las clausulas, independientemente de si pertenecen al mismo o diferentes ensayos. Cada par de frases se procesa secuencialmente.
2. Inicialmente, cada frase pertenece a su propia clase.
3. Se calcula la similaridad entre fragmentos de texto utilizando la libraría SpaCy [52, 51], y se aplica el modelo de identificación de paráfrasis sobre pares de frases suficientemente similares (hemos utilizado un corte de 0,9). Este valor umbral de la similaridad es necesario para reducir el número de pares de los que se evalúa la paráfrasis, que es el paso más costoso. La selección del valor de corte se basa en una observación del total de grupos obtenidos y la similaridad entre pares de frases en el corpus de entrenamiento del modelo (normalmente superior a 0,75).
4. Si las dos frases están relacionadas por paráfrasis, las clases de cada frase se fusionan.
5. Se continúa el proceso hasta agotar todas las parejas de frases.

Un par de detalles relevantes:

1. Se asume que la paráfrasis define una relación e equivalencia:
   a) A es paráfrasis de A (reflexividad).
   b) Si A es paráfrasis de B, B lo es de A (simetría).
   c) Si A es paráfrasis de B y B lo es de C, A es paráfrasis de C (transitividad).
2. El algoritmo compara todas las parejas de frase que superan el corte en similaridad. En ese sentido, se puede decir que hace «backtracking», lo que sin duda hace que su tiempo de ejecución sea mayor.

Construcción del grafo Partiendo del conjunto de clases anterior:

1. Se selecciona un representante para cada clase y tipo de argumento y se crean los nodos que representan las cláusulas argumentales utilizadas en la muestra de ensayos.
2. Cada nodo tiene asignada como calificación la suma de la evaluación para cada dominio de todos los documentos que aportan una frase a la clase. Se asigna también un tamaño de la clase, que es simplemente el número de cláusulas.
3. Si dos frases en dos clases diferentes (A y B) están enlazadas, se crea un enlace direccional A $\longrightarrow$ B si A contiene más frases enlazadas con frases de B de las que B tiene con A. En caso de empate se asigna una dirección aleatoria. El enlace tiene un peso final dado por la fracción de frases conectadas entre A y B y el tamaño total de las dos clases.

El resultado es un grafo dirigido cuyos nodos y enlaces representan argumentos típicos en la muestra de ensayos elegida. El peso de los enlaces indica la popularidad de este tipo de apoyo entre fragmentos del argumento. Cada nodo (clase o cláusula argumental típica) tiene asociado un texto, que se elige de entre las frases de la clase pertenecientes al ensayo con mejor calificación agregada. Se espera que haya nodos desconectados, debido al margen de error del modelo de minería argumental [78] que utilizamos y eventuales incoherencias en los argumentos expuestos en los ensayos, entre otros posibles motivos.

El grafo resultante se puede visualizar con aplicaciones específicas, como por ejemplo, Gephi [9]. Se puede denotar este grafo resumen como grafo colaborativo, ya que está construido a partir de las aportaciones de los estudiantes. A partir de esta visualización, se puede dirigir una discusión sobre la validez de los argumentos conectados encontrados y posibles conexiones con otros argumentos detectados o ausentes.

3. Discusión de resultados y conclusiones

Los resultados de este trabajo se basan en cinco experimentos numéricos:

- Entrenamiento y validación de algoritmos de aprendizaje automático para generar los atributos de estilo y coherencia de un ensayo argumentativo.
- Utilización de estos algoritmos de aprendizaje automático ya entrenados y técnicas de procesamiento de texto natural para generar los atributos de un corpus de ensayos argumentativos anotados. En este proceso validaremos hasta qué punto los atributos seleccionados son suficientemente descriptivos.
- Entrenamiento y validación de una red bayesiana sobre el corpus anterior como modelo de indicadores de la competencia en argumentación.
- Entrenamiento y validación de un algoritmo de aprendizaje automático para detectar la paráfrasis entre dos frases.
- Generación de un grafo de argumentación colaborativo para una selección de documentos del corpus.

Los métodos utilizados están descritos en el capitulo anterior. En este capitulo se presentan los resultados detallados y una discusión de estos.

3.1. Detalles de los experimentos realizados

3.1.1. Entrenamiento y validación del modelo de coherencia de un texto

A continuación se indican los detalles del proceso seguido para entrenar el modelo descrito en la sección 2.3.2.10.

Corpus utilizado:

- 402 ensayos persuasivos sin anotar procedentes del corpus AAEC v2. A estos textos se les ha eliminado el encabezado original, que indica la propuesta de tesis del ensayo.
- 615 documentos sin anotar del corpus Araucaria (última versión), [100].

- 44 documentos sin anotar del corpus de Brown [38], referencias cb01-27y cc01-17, correspondientes a editoriales y reseñas de prensa (más detalles en Brown Corpus Details).
- 10 artículos Anne Lamott y 9 artículos de Martin Amis.

El proceso de vectorización del corpus siguió estos pasos:

1. Generación del árbol RST utilizando el código Python 2.7 y el modelo pre-entrenado disponible en gCRF Suite ([134, 133]).
2. Tokenización de los documentos (con un script en Python3, la librería SpaCy y el modelo en_core_web_lg) y cálculo del número de tokens que participan en transiciones de rol retórico en hasta tres fragmentos del discurso (EDU) consecutivos. De esta manera se representa cada documento como un vector de dimensión 88264, en el que cada componente indica la probabilidad de que un término participe en EDUs con una relación retórica entre ellas. Se calculan dos probabilidades diferentes para términos salientes (aparecen más de dos veces) y no salientes (aparecen una sola vez). Nótese que muchas de estas componentes serán cero para todos los documentos.
   a) El número de atributos se reduce a 583 al eliminar los que son cero para todo el corpus.
3. Generación de una copia de los documentos anteriores, pero con cada frase ordenada aleatoriamente. Se repiten los pasos 1 y 2 para vectorizar los documentos desordenados pero se conservan únicamente las mismas 583 dimensiones obtenidas para el corpus sin desordenar. Se asume que estos son los únicos atributos relevantes para modelar la coherencia de un texto.
   a) Véase en la figura 3.1 la distribución de documentos vectorizados en las dos primeras componentes principales
4. Cálculo de un conjunto de datos de entrenamiento y test asociando a cada documento la diferencia de su vector con el vector del documento desordenado. Etiquetamos cada registro como 1.
5. Añadimos al conjunto de datos la diferencia entre el vector del documento desordenado y el vector del documento original, y lo etiquetamos como -1.
6. El conjunto de datos final contiene las 583 componentes del vector diferencia y la etiqueta.

Se utilizó la librería **scikit-learn** [**88**] de Python para entrenar una maquina de soporte vectorial (SVM) de núcleo lineal como clasificador sobre estos datos. Con el fin de buscar el modelo que tenga más probabilidades de generalizar mejor al predecir sobre datos nuevos, se buscó el valor del parámetro de margen C que maximiza el valor de la métrica F1 (media armónica de la precisión y exhaustividad) calculada sobre documentos no utilizados en el entrenamiento. El proceso se implementó como una validación cruzada estratificada con tres iteraciones ("Stratified k-fold) (3.1). Seleccionamos el modelo de mayor valor F1.

C	F1 micro - Entrenamiento	F1 macro - Test
0.1	0.998214	0.847518
0.1	0.998221	0.839286
0.1	1	0.828571
0.5	1	0.851064
0.5	1	0.828571
0.5	1	0.85
1	1	0.851064
1	1	0.828571
1	1	0.85
5	1	0.836879
5	1	0.828571
5	1	0.857143
10	**1**	**0.886525**
10	1	0.835714
10	1	0.871429
50	1	0.858156
50	1	0.864286
50	1	0.864286
100	1	0.843972
100	1	0.821429
100	1	0.857143

Cuadro 3.1.: Métricas obtenidas durante el entrenamiento de la maquina de soporte vectorial de núcleo lineal utilizadas para modelar la coherencia de texto. En negrita los resultados para el modelo utilizado

3.1.2. Entrenamiento y validación del modelo de estilo de un texto

Para implementar el método descrito en 2.3.2.11, se utilizó el corpus de la sección anterior, sin los documentos de Araucaria y con un par de ensayos adicionales. La idea es conseguir una representación simplista, pero útil, del estilo de un texto, comparando textos de un estilo razonablemente diferenciado: ensayos de estudiantes y texto periodístico o de ensayistas reconocidos.

- En un primer paso se generaron rasgos para 467 documentos siguiendo la referencia [41], aunque con algunas diferencias ya mencionadas en el capítulo 2. De nuevo utilizamos un script de Python 3, SpaCy y el modelo en_core_web_lg para extraer las etiquetas PoS ("Part of Speech", parte de la oración)y las dependencias sintácticas en el texto y construir los atributos detallados en 2.3.2.11.
- Se generaron vectores para todos los documentos del corpus utilizando las frecuencias de aparición de cada tri-grama PoS, producción sintáctica y relación

de dependencia, junto con la frecuencia de cada etiqueta PoS. Cada tipo de frecuencia se normaliza por separado en cada documento.

- Se excluyeron símbolos, palabras desconocidas, espacios y puntuación.
- Cada documento se representa inicialmente con un vector de dimensión 30628.

- Se etiquetaron los ensayos de estudiantes como 0 y textos periodísticos o literarios como 1.

Se utilizó la librería **scikit-learn** [**88**] de Python para entrenar un bosque aleatorio ("Random Forest", RF) como clasificador binario sobre este conjunto de datos. Al contrario que en el estudio de la coherencia, los datos tienen una descompensación de clases importante (86 % de los documentos son ensayos), lo que potencialmente puede viciar los resultados del clasificador (esto es, simplemente asignando la clase mayoritaria de forma ingenua, lograríamos una precisión alta). Para mitigar este efecto, el modelo se entrenó con diferentes métodos de ponderación de cada instancia (dando más peso a la clase infrarrepresentada), que está disponible como opción en **scikit-learn**, y se utilizará una métrica F1 ponderada para elegir el mejor modelo.

A partir de una validación cruzada estratificada con tres iteraciones ("Stratified k-fold), se entrenó un RF en cada una de las tres particiones con diferentes valores del número de estimadores (árboles de decisión considerados en el modelo) y métodos de asignar pesos a cada instancia (el algoritmo da más peso a la clase menos representada). En cada partición calculamos la métrica F1 ponderada en la parte de los datos que no se utilizaron en el entrenamiento. De nuevo, seleccionamos el modelo de mayor valor F1. Presentamos también (3.23.3) los resultados de precisión y exhaustividad de la clase minoritaria, con el fin de entender hasta qué punto podemos considerar que el modelo entrenado generaliza bien. En la tabla 3.4 se listan los rasgos más relevantes en el bosque aleatorio.

La figura 3.2 indica la distribución de documentos vectorizados utilizando las tres primeras componentes principales.

3.1.3. Entrenamiento y validación del modelo de detección de paráfrasis

El objetivo es entrenar y validar un modelo que permita estimar si dos fragmentos de texto expresan las mismas ideas utilizando, posiblemente palabras o estructura diferente (paráfrasis). Tal como se explicó en una sección 2.5, se entrenó un clasificador binario (bosque aleatorio o "Random Forest") en el Microsoft Research Paraphrase Corpus [129].

Este corpus consta de 4076 pares de frases para entrenamiento y 1725 de test. Todas las frases están anotadas con una etiqueta 0 o 1 según la pareja se considera que presenta paráfrasis o no.

Ponderado de clases	Número de estimadores	F1 ponderado entrenamiento	F1 ponderado test
Ninguno	10	1	0.958923
Ninguno	10	0.996769	0.980172
Ninguno	10	0.996779	0.958488
Ninguno	50	1	0.986922
Ninguno	50	1	0.980172
Ninguno	50	1	0.973025
Ninguno	100	1	0.980172
Ninguno	100	1	0.966188
Ninguno	100	1	0.973025
Ninguno	150	1	0.966188
Ninguno	150	1	0.980172
Ninguno	150	1	0.993482
Ninguno	200	1	0.973265
Ninguno	200	1	0.980172
Ninguno	200	1	0.980008
balanced	10	1	0.951455
balanced	10	1	0.986922
balanced	10	0.996779	0.980008
balanced	50	1	0.980172
balanced	50	1	0.993527
balanced	50	1	0.980008
balanced	**100**	**1**	**1**
balanced	100	1	0.966188
balanced	100	1	0.965858
balanced	150	1	0.986922
balanced	150	1	0.980172
balanced	150	1	0.973025
balanced	200	1	0.958923
balanced	200	1	0.986922
balanced	200	1	1
balanced_subsample	10	0.996769	0.986922
balanced_subsample	10	0.996769	0.980172
balanced_subsample	10	0.996779	0.986822
balanced_subsample	50	1	0.993527
balanced_subsample	50	1	0.980172
balanced_subsample	50	1	0.980008
balanced_subsample	100	1	0.993527
balanced_subsample	100	1	0.958923
balanced_subsample	100	1	0.980008
balanced_subsample	150	1	0.966188
balanced_subsample	150	1	0.980172
balanced_subsample	150	1	0.986822
balanced_subsample	200	1	0.986922
balanced_subsample	200	1	0.973265
balanced_subsample	200	1	0.980008

Cuadro 3.2.: Métricas de test obtenidas al entrenar una serie de bosques aleatorios para caracterizar el estilo de un texto. Detalles en 2.3.2.11. Estimadores es el número de árboles de decisión en el bosque. El ponderado de clases «balanced» pesa cada instancia con el inverso de la frecuencia de su clase. El método «Balanced subsample» utiliza el inverso de la frecuencia en la muestra de datos utilizada por el RF cuando añade un árbol nuevo

Para este experimento se ha seguido de cerca la referencia [107]:

- En un primer paso se generan los atributos descritos en 2.5, utilizando un script de Python3 y la librería NLTK [14] para extracción de n-gramas, referencia de “stop words” en inglés, y como interfase con el corpus WordNet [121, 70]. Se vuelve a utilizar SpaCy junto con el modelo en_core_web_lg para tokenizar y lematizar el texto, y detectar oraciones nominales y palabras con dependencias sintácticas de un verbo. Para calcular la subcadena común más larga, hemos utilizado directamente el código encontrado en Longest_common_subsequence_code.

    - Al vectorizar los pares de frases de entrenamiento y test del corpus el resultado es un vector de dimensión 18 para cada pareja de frases.

Ponderado de clases	Estimadores	Precisión clase mayoritaria	Exhaustividad clase mayoritaria	Precisión clase minoritaria	Exhaustividad clase minoritaria
Ninguno	10	0.957143	1	1	0.727273
Ninguno	10	0.978102	1	1	0.863636
Ninguno	10	0.957143	1	1	0.714286
Ninguno	50	0.985294	1	1	0.909091
Ninguno	50	0.978102	1	1	0.863636
Ninguno	50	0.971014	1	1	0.809524
Ninguno	100	0.978102	1	1	0.863636
Ninguno	100	0.964029	1	1	0.772727
Ninguno	100	0.971014	1	1	0.809524
Ninguno	150	0.964029	1	1	0.772727
Ninguno	150	0.978102	1	1	0.863636
Ninguno	150	0.992593	1	1	0.952381
Ninguno	200	0.971014	1	1	0.818182
Ninguno	200	0.978102	1	1	0.863636
Ninguno	200	0.978102	1	1	0.857143
balanced	10	0.950355	1	1	0.681818
balanced	10	0.985294	1	1	0.909091
balanced	10	0.978102	1	1	0.857143
balanced	50	0.978102	1	1	0.863636
balanced	50	0.992593	1	1	0.954545
balanced	50	0.978102	1	1	0.857143
balanced	**100**	**1**	**1**	**1**	**1**
balanced	100	0.964029	1	1	0.772727
balanced	100	0.964029	1	1	0.761905
balanced	150	0.985294	1	1	0.909091
balanced	150	0.978102	1	1	0.863636
balanced	150	0.971014	1	1	0.809524
balanced	200	0.957143	1	1	0.727273
balanced	200	0.985294	1	1	0.909091
balanced	200	1	1	1	1
balanced_subsample	10	0.985294	1	1	0.909091
balanced_subsample	10	0.978102	1	1	0.863636
balanced_subsample	10	0.985294	1	1	0.904762
balanced_subsample	50	0.992593	1	1	0.954545
balanced_subsample	50	0.978102	1	1	0.863636
balanced_subsample	50	0.978102	1	1	0.857143
balanced_subsample	100	0.992593	1	1	0.954545
balanced_subsample	100	0.957143	1	1	0.727273
balanced_subsample	100	0.978102	1	1	0.857143
balanced_subsample	150	0.964029	1	1	0.772727
balanced_subsample	150	0.978102	1	1	0.863636
balanced_subsample	150	0.985294	1	1	0.904762
balanced_subsample	200	0.985294	1	1	0.909091
balanced_subsample	200	0.971014	1	1	0.818182
balanced_subsample	200	0.978102	1	1	0.857143

Cuadro 3.3.: Precisión y exhaustividad en los datos de validación obtenidos durante el entrenamiento del modelo de caracterización de estilo. Aunque el modelo elegido presenta una precisión y exhaustividad perfectas, hay bastante variabilidad, por lo que se espera que en otros conjuntos de datos su precisión sea más baja que uno

- De nuevo se recurre a la librería **scikit-learn** [**88**] de Python para entrenar un bosque aleatorio ("Random Forest", RF) como clasificador binario sobre este conjunto de datos. Se respetó la división entre textos de entrenamiento (4076 pares) y validación (1725) realizada por los autores del corpus. En este caso entrenaremos sobre los 4074 pares marcados para entrenar, y el resto se utilizará como validación.
- Hay una descompensación de clases (en entrenamiento, 67.6 % de pares están etiquetados con 1, y en validación, el 66.4 %). Al igual que para el modelo de estilo, se mitiga el riesgo para la validez del clasificador utilizando diferentes técnicas de pesado de instancias en el entrenamiento. Elegimos el modelo que presente mejor métrica F1 ponderada en los datos de validación.
    - Además del número de estimadores, hemos entrenado el bosque aleatorio

Atributo de estilo	Importancia
dep_IN-pobj-NNP	0.0218456
pos3_NNP-CC-NNP	0.0212608
dep_NNP-det-DT	0.0174486
synpro_NNP->POS	0.0164486
dep_NN-poss-NNP	0.0160289
dep_NNP-compound-NNP	0.0153788
pos3_NNP-POS-NN	0.0125015
pos3_IN-DT-NNP	0.0113138
dep_NNP-conj-NNP	0.0110874
pos3_NNP-IN-DT	0.0110726
dep_NNP-amod-JJ	0.0103585
synpro_NNP->NNP	0.00987288
synpro_NNP->DT	0.00976966
dep_VBD-nsubj-NNP	0.00955429
dep_VBD-nsubj-WDT	0.0085188
dep_VBD-advcl-VBD	0.00839776
dep_VBD-nsubj-NN	0.00835621
dep_VB-dobj-NNP	0.00811883
dep_NNPS-det-DT	0.00811053
synpro_IN->NNP	0.00797402

Cuadro 3.4.: Los veinte atributos de estilo más relevantes. La importancia es estimada por el bosque aleatorio durante el entrenamiento

con diferentes profundidades máximas de cada árbol de decisión y diferentes métodos para limitar el número de atributos que cada árbol de decisión procesa.

- Véanse los resultados del entrenamiento y validación en las tablas 3.5 y 3.6. La tabla 3.7 presenta los atributos del modelo ordenados por relevancia.

- En general, las métricas respecto a la clase mayoritaria (paráfrasis) no están lejos de los resultados reportados por [3] en el mismo corpus en 2009. Se han buscado resultados más recientes, pero sólo se ha encontrado la referencia [116], sin mejoras aparentes en la precisión.
- La precisión y exhaustividad respecto a la clase minoritaria (pares de frases sin paráfrasis) son peores. Es posible que el corpus utilizado tenga un sesgo apreciable ([3]). En este trabajo no se intentará mitigar este efecto, pero se tendrá en cuenta a la hora de discutir los resultados.
- Los resultados de todos los ciclos de entrenamiento se pueden encontrar en el apéndice, tablas A.2, A.3, A.4 y A.5. En las tablas 3.5, 3.6 y 3.7se puede

encontrar una muestra parcial

Máxima profundidad de árbol	Máximo número de atributos por árbol	Ponderado de clases	Número de estimadores	F1 ponderado entrenamiento	F1 ponderado test
Sin límite	auto	Ninguno	10	0.989602	0.729696
Sin límite	auto	Ninguno	50	0.999492	0.753441
Sin límite	auto	Ninguno	100	1	0.745425
Sin límite	auto	Ninguno	150	1	0.745947
Sin límite	auto	Ninguno	200	1	0.750464
Sin límite	**auto**	**Ninguno**	**250**	**1**	**0.758064**
Sin límite	auto	balanced	10	0.991884	0.724556

Cuadro 3.5.: Resultados de la validación del bosque aleatorio clasificador de paráfrasis sobre los datos de validación. Resaltamos los resultados para el modelo elegido. Estimadores es el número de árboles de decisión en el bosque. El ponderado de clases «balanced» y «balanced subsample» dan más importancia en el entrenamiento a instancias minoritarias.

Máxima profundidad de árbol	Máximo número de atributos por árbol	Ponderado de clases	Número de estimadores	Precisión clase mayoritaria	Exhaustividad clase mayoritaria	Precisión clase minoritaria	Exhaustividad clase minoritaria
Sin límite	auto	Ninguno	10	0.798701	0.79136	0.595365	0.606171
Sin límite	auto	Ninguno	50	0.792262	0.865809	0.675556	0.551724
Sin límite	auto	Ninguno	100	0.782072	0.874081	0.676123	0.519056
Sin límite	auto	Ninguno	150	0.781046	0.878676	0.681928	0.513612
Sin límite	auto	Ninguno	200	0.787854	0.870404	0.677346	0.537205
Sin límite	**auto**	**Ninguno**	**250**	**0.791563**	**0.879596**	**0.695349**	**0.54265**

Cuadro 3.6.: Precisión y exhaustividad de cada clase calculada. La clase mayoritaria son los pares de frase con paráfrasis

3.1.4. Extracción de atributos del discurso y validación

El corpus en el que se entrenó el modelo de discurso argumentativo y el grado de argumentación es una muestra de 527 ensayos extraídos del corpus Kaggle ASAP Phase 1 tema 2. Este corpus es interesante por contener más de mil ensayos persuasivos con calificaciones en un par de dominios (efectividad en la expresión y dominio del lenguaje). Al integrar estas calificaciones como variables aleatoria de nuestro dominio podemos ir más allá de intentar inferirlas a partir de atributos conocidos, y entender como pueden influir en los otros atributos.

La extracción de atributos para un conjunto de textos tiene dos pasos principales:

1. Entrenar los modelos de coherencia, estilo y tal como se ha descrito en la sección 2.3.
2. Generar los atributos del discurso 3.8.

Los pasos para generar los atributos son:

1. Generar los atributos lingüísticos generales detallados en 2.3.2.

Atributo de estilo	Importancia
PEA	0.103846
jaccard_char_ngrams	0.10372
similarity_global	0.0944087
wos	0.0647729
jaccard_lemma_ngrams	0.0620008
jaccard_word_ngrams	0.0607625
size_set1-2_2	0.0569252
size_set2-1_1	0.0547112
wos_cos	0.0537759
lcs	0.0534903
jaccard_POS_ngrams	0.0530902
weight_ic	0.0519823
size_set2-1	0.0385824
size_set1-2	0.0335624
size_set_1+2	0.0316576
size_set1	0.0284414
size_set2	0.0277115
size_set_1_int_2	0.0265581

Cuadro 3.7.: Importancia relativa de todos los atributos utilizados para similaridad semántica

   a) Además de la librería de Python SpaCy.io junto con el modelo preentrenado en_core_web_lg, se han utilizado las librerías para calcular los indices de legibilidad spacy_readability.

2. Vectorizar documentos para calcular la métrica de estilo (2.3.2.11).
3. Procesar los documentos con el código y modelo entrenado descritos en [134, 133] para obtener el árbol de estructura retórica RST ([65])
4. Obtener los atributos de coherencia (2.3.2.10) de cada documento a partir de la estructura RST y calcular la métrica de coherencia.
5. Procesar el corpus para generar los atributos léxicos, sintácticos, semánticos y de contenido de información indicados en la sección 2.5.
6. Utilizar el código y modelo entrenado descritos 2.3.2.12, que implementan el sistema de minería de argumentos propuesto en [78] para:
   a) Vectorizar los textos.
   b) Obtener la descomposición en proposiciones argumentativas así como sus relaciones.
7. Eliminar variables constantes (en este caso la frecuencia de URLs, que es cero en nuestra muestra).

Las figuras 3.3,3.4, 3.5, 3.6, 3.7, 3.8 representan la distribución de valores de los atributos calculados, así como de las calificaciones anotadas por instructores.

Una vez vectorizado el corpus de ensayos a analizar, validar hasta que punto los atributos calculados son descriptivos, entrenando un modelo de regresión de las calificaciones para cada dominio.

Atributo/Etiqueta	Desscripción
n_tokens	Número de tokens: palabras y símbolos
f_stopwords	Fraccion de stop words en el texto respecto a tokens
f_punct	Fraccion de símbolos de puntuación en el texto respecto a tokens
f_diff_lemmas	Fraccion de stop words en el texto respecto a tokens
f_ents	Fraccion de entidades el texto respecto a tokens
f_oov	Fraccion de palabras no reconocidas en el texto respecto a tokens
fk_reading_ease	Índice de legibilidad de Flesh-Kincaid
fk_grade_level	Nivel de legibilidad de Flesch-Kincaid
dale_chall_readability	Índice de legibilidad de Dale-Chall
f_numeric_tokens	Fraccion de expresiones numéricas en el texto respecto a tokens
f_uri_tokens	Fraccion de referencias estilo URI en el texto respecto a tokens
precision	Grado de precisión del texto, utilizando WordNet
polarity_average	Promedio de polaridad en le texto, utilizando SentiWordNet
polarity_peak	Picos de polaridad en le texto, utilizando SentiWordNet
style	Probabilidad de que el estilo se acerque al de los ensayos del corpus AAEC
coherence	Medida de coherencia del texto
n_fact	Número de proposiciones argumentativas que se refieren a un hecho aceptado
n_testimony	Número de proposiciones argumentativas que exponen un testimonio
n_value	Número de proposiciones argumentativas que expresan una opinión
n_policy	Número de proposiciones argumentativas que piden una llamada a la acción
n_reference	Número de proposiciones argumentativas en las que se indica una referencia: URL, cita
n_links	Número de relaciones detectadas entre proposiciones argumentativas
rater1_domain1	Calificación del primer evaluador en el dominio 1. Anotada en el corpus ASAP, no utilizada
rater2_domain1	Calificación del segundo evaluador en el dominio 1. Anotada en el corpus ASAP, no utilizada
domain1_score	Calificación agregada en el dominio 1, efectividad de la expresión. Anotada en el corpus ASAP
rater1_domain2	Calificación del primer evaluador en el dominio 2. Anotada en el corpus ASAP, no utilizada
rater2_domain2	Calificación del segundo evaluador en el dominio 2. Anotada en el corpus ASAP, no utilizada
domain2_score	Calificación agregada en el dominio 2, dominio del lenguaje. Anotada en el corpus ASAP

Cuadro 3.8.: Atributos del discurso considerados para el modelo bayesiano de indicadores de la competencia en argumentación. Los últimos seis campos son las etiquetas anotadas en el corpus. Las calificaciones agregadas en el primer y segundo dominio, efectividad de la expresión y dominio del lenguaje respectivamente, se utilizaron durante el entrenamiento del modelo, y para texto nuevo, se infieren del resto de atributos. Solo se utilizan las calificaciones agregadas para los dos evaluadores.

3.1.4.1. Modelo de regresión para los dos dominios en el corpus Kaggle ASAP Phase 1 tema 2

A partir de los 527 ensayos vectorizados se crearon dos conjuntos de datos, en el que la calificación en uno de los dos dominios es la etiqueta numérica a aprender por el modelo de regresión. De nuevo se utilizó la librería **scikit-learn** [88] de Python para entrenar un par de bosques aleatorio ("Random Forest", RF) como regresores sobre cada conjunto de datos.

Se utilizó validación cruzada con tres iteraciones («k-fold cross validation») para elegir el número de árboles estimadores más adecuado para el modelo. Se entrena el modelo sobre una partición de datos y se calcula el error cuadrático promedio por instancia en la partición restante. Esta métrica nos indica en promedio el cuadrado

del error que podemos esperar al predecir la calificación de un ensayo nuevo. Cada árbol del bosque tratará todos los atributos, ya que son un número reducido.

Los resultados se muestran en las tablas 3.9 y 3.10. Se incluyen los atributos ordenados por relevancia en estos modelos de regresión en la tabla 3.11.

Número de estimadores	R_2 Entrenamiento	Error cuadrático medio Entrenamiento	R_2 Validación	Error cuadrático Validación
20	0.91144	0.0250709	0.354167	0.0450855
20	0.900938	0.0256737	0.44409	0.0446827
20	0.918167	0.0236177	0.360372	0.0476244
50	0.930858	0.0224256	0.360626	0.0435435
50	0.909342	0.0231966	0.451638	0.048613
50	0.917085	0.0247063	0.451966	0.0405297
100	0.927646	0.0222056	0.349656	0.0468706
100	0.924803	0.0224453	0.481017	0.0428411
100	0.920563	0.0235803	0.501111	0.0403035
150	0.933554	0.021141	0.330544	0.0485345
150	0.919247	0.0236055	0.500356	0.040804
150	0.924443	0.0228872	0.450163	0.0433249
200	0.923463	0.0225219	0.473149	0.0434859
200	0.92454	0.0224879	0.481076	0.0428742
200	0.928167	0.0227725	0.405152	0.0429447
250	0.929215	0.0213564	0.416861	0.0470773
250	0.924069	0.0231423	0.411155	0.0432819
250	**0.919617**	**0.0238282**	**0.542718**	**0.0387253**

Cuadro 3.9.: Resultados del modelo de regresión de la calificación para el dominio 1 del corpus ASAP utilizado. Como variables independientes se utilizan los atributos calculados previamente. Resaltamos el modelo que podemos esperar que generalice mejor

Número de estimadores	R_2 Entrenamiento	Error cuadrático medio Entrenamiento	R_2 Validación	Error cuadrático Validación
20	0.896745	0.0244252	0.304959	0.0466255
20	0.886168	0.0258616	0.373548	0.0435679
20	0.888478	0.026357	0.272224	0.0449726
50	0.905659	0.0232163	0.337561	0.0459493
50	0.901767	0.0245249	0.280424	0.044738
50	0.89724	0.024908	0.381066	0.0427784
100	0.919296	0.0215105	0.255699	0.048571
100	0.896859	0.0243783	0.450313	0.0414973
100	0.908772	0.0241168	0.264143	0.0437869
150	**0.902201**	**0.0248501**	**0.393751**	**0.0392518**
150	0.922724	0.0212022	0.234702	0.0485567
150	0.904445	0.0233792	0.411843	0.0438537
200	0.902443	0.0243456	0.409188	0.0408363
200	0.912358	0.0228879	0.28764	0.0454815
200	0.917255	0.0219314	0.276768	0.0480374
250	0.913473	0.0233019	0.287147	0.0430033
250	0.913272	0.0230607	0.290728	0.0436766
250	0.898205	0.0233317	0.367037	0.0465533

Cuadro 3.10.: Resultados del modelo de regresión de la calificación para el dominio 2 del corpus ASAP utilizado. Resaltamos el modelo que podemos esperar que generalice mejor

3.1.5. Entrenamiento y evaluación del modelo bayesiano de indicadores de la competencia en argumentación

Una vez vez generados los 21 atributos para los 527 ensayos extraídos el corpus Kaggle ASAP Phase 1 tema 2 (se ha eliminado la fracción de URLs, que es cero para todos los documentos), junto con las dos calificaciones agregadas por evaluador para los dominios 1 y 2, se utiliza una red bayesiana para representar la distribución

Atributo	Importancia Dominio 1	Atributo	Importancia Dominio 2
n_tokens	0.457981	n_tokens	0.312188
n_value	0.0663751	n_value	0.145895
f_punct	0.0559463	f_punct	0.0864617
precision	0.0438963	f_oov	0.0436358
dale_chall_readability	0.035346	fk_reading_ease	0.0428661
fk_reading_ease	0.0340282	polarity_peak	0.0400497
coherence	0.0324534	f_diff_lemmas	0.037983
f_stopwords	0.0312338	precision	0.0349798
f_diff_lemmas	0.0292791	dale_chall_readability	0.0297468
f_oov	0.0284197	coherence	0.0292289
polarity_peak	0.025916	f_stopwords	0.0291405
fk_grade_level	0.0249553	f_ents	0.0279015
style	0.0220653	fk_grade_level	0.0257289
polarity_average	0.021585	polarity_average	0.0243936
f_ents	0.0212923	f_numeric_tokens	0.0239406
f_numeric_tokens	0.0167679	n_links	0.0197956
n_testimony	0.0164694	style	0.0158468
n_links	0.014269	n_policy	0.0102154
n_policy	0.00929985	n_reference	0.00877964
n_fact	0.00785732	n_testimony	0.00618652
n_reference	0.00456367	n_fact	0.00503566

Cuadro 3.11.: Relevancia de atributos a la hora de aprender la calificación en dominios 1 y 2 para los documentos de ASAP.

de probabilidad conjunta de estas 23 variables. Para ello se desarrolló un script de Python 3 para implementar el algoritmo MMHC detallado en la referencia [119] y explicado en 2.4.1.

Como datos de validación, se utiliza un conjunto de 498 documentos diferentes del mismo corpus. Estos documentos se vectorizaron siguiendo el mismo proceso descrito anteriormente.

- El primer paso (Max-Min Parent Children) aprende la estructura de un grafo no dirigido que representa las relaciones de independencia condicional encontradas en los datos. Para testear la independencia condicional se utilizó la librería Python Tigramite, que implementa el algoritmo descrito en [102]. La figura 3.9 representa el grafo obtenido.
- El segundo paso (escalada simple o «Greedy Hill-Climbing») busca enlaces dirigidos entre las 23 variables que estén contenidos en el grafo no dirigido obtenido anteriormente. Partiendo de una red sin enlaces, en cada paso añade o elimina un enlace y se genera un modelo diferente de red bayesiana normal (o gaussiano) paramétrica lineal para el que se realiza un proceso de inferencia de sus parámetros a partir de los datos de entrenamiento. Si la densidad predictiva logarítmica del nuevo modelo calculada sobre los datos de validación es más alta que en el paso anterior, se acepta el cambio. El proceso acaba si no se puede encontrar un paso que mejore esta métrica.
    - Como se indicó en la descripción de la metodología, los parámetros libres del modelo son las desviaciones estándar y los coeficientes de las combinaciones lineales que definen la media de cada distribución normal. Para estos coeficientes se utiliza una distribución «a priori» normal, con media cero y desviación estándar 100 y para las desviaciones estándar, una distribución de Cauchy de dominio positivo con parámetro $\beta = 5$ (este parámetro indica la anchura de la distribución).

    - En este y el siguiente paso, el proceso de inferencia y cálculo de métricas utiliza la librería PYMC3, [103]. Para cada red bayesiana se construye un modelo PYMC3 programáticamente. La inferencia se realiza con el algoritmo No-U-Turn Sampler -NUTS, con 2000 pasos de inicialización («burn-in») para evitar que haya divergencias en algunos pasos
    - La figura 3.10 representa el grafo obtenido en este paso.
- Una vez encontrado una red bayesiana gaussiana lineal en el paso anterior, realizamos un nuevo proceso de inferencia de parámetros para recoger las métricas de validación definitivas para el modelo.
    - La red bayesiana tiene 83 variables independientes para parametrizar valor medio y desviación estándar de las distribuciones normales asociadas a cada uno de los 23 nodos de la red
    - En este proceso se realizaron 2000 pasos de inicialización («burn-in») y otros mil pasos adicionales de inferencia para estimar la distribución a posteriori de las variables independientes
    - Se pueden encontrar parte de trazas del proceso de inferencia en el apéndice A.4. Unas primeras indicaciones de que la inferencia ha convergido con éxito son:
        - La ausencia de divergencias reportadas por el algoritmo de muestreo
        - Las estadísticas del test Gelman-Rubin son cercanas a uno. El test de Gelman-Rubin [42] compara la varianza entre diferentes cadenas del proceso de inferencia (Markov Chain Monte Carlo o No-U-Turn) y la varianza dentro de cada cadena
- La métrica de validación usada es el error cuadrático medio (la suma del error cuadrático total dividido por el número de ensayos) de la predicción para cada una de las dos calificaciones de cada ensayo. Esta métrica se calcula sobre los datos de validación, no utilizados durante el entrenamiento. Los resultados están tabulados en 3.12.
    - Véase la sección 2.4.3 para más detalles sobre como utilizamos la red bayesiana para predicción de valores de variables una vez conocidos los valores del resto de variables.
- Una validación adicional es la comparación de media, desviación estándar y asimetría entre el corpus de ensayos de test (no utilizados en el entrenamiento) y atributos muestreados a partir de la distribución de probabilidad del modelo «a posteriori» (tras la inferencia a partir del corpus de entrenamiento). Véase la tabla 3.13 con los resultados obtenidos para mil muestras.
    - Hemos utilizado la librería de Python Scipy [56] para calcular la asimetría del conjunto de datos, definida como $g_1 = m_3/m_2^{3/2}$, donde $m_r = \sum_{i=1,\dots N}(x_i - \bar{x})^r$ es el r-momento de las observaciones $\{x_i\}_{i=1}^N$[138].

Las métricas de convergencia del proceso se obtuvieron directamente con PYMC3. Esta librería permite realizar muestreos a posteriori que se utilizaron para el cálculo de métricas de validación. Para el cálculo de la densidad predictiva, predicciones de la calificación de cada documento y el gradiente del valor esperado condicionado de cada calificación se definieron funciones específicas utilizando Theano, una librería de cálculo simbólico utilizada por PYMC3 para ejecutar cálculos.

Atributo	Error cuadrático medio	Desviación estándar observada
Domain1_score	0.02384269736393971	0.5778053844875372
Domain2_score	0.022947979391388415	0.5309830052749068

Cuadro 3.12.: Error cuadrático medio de validación para el modelo bayesiano de indicadores de la competencia en argumentación. Aunque el valor medio del error es mejor que en el modelo de regresión visto anteriormente, la desviación estándar que observamos es alta por lo que no podemos afirmar categóricamente que la red bayesiana predice mejor las calificaciones que un modelo de regresión. De todas maneras, el modelo de regresión no incluía la calificación de ningún dominio, mientras que en este modelo mantenemos una de ellas, por lo que si se podría esperar un mejor resultado

3.1.6. Extracción de un ejemplo de grafo de argumentación colaborativo

A partir de dos muestras de 61 y 23 ensayos extraídos del Kaggle ASAP Phase 1, tema 2, y las cláusulas argumentativas detectadas durante el proceso de extracción de atributos descrito en 3.1.4, se ha generado un par de grafos de argumentación colaborativos siguiendo los pasos descritos en 2.5.2. Cada nodo del grafo (una clase o cláusula argumental típica) tiene un texto asociado, que es una de las frases de la clase, elegida del ensayo con calificación agregada más alta.

La primera muestra se eligió aleatoriamente. La siguiente muestra incluye ensayos para los que hemos detectado el mayor número de enlaces entre cláusulas argumentativas. Las dos muestras no se solapan. A la hora de crear los grafos para la visualización, no hemos considerado clases con un único elemento.

Se ha escrito un script de Python 3 para procesar los ensayos y los modelos de minería de argumentos y de detección de paráfrasis, generar las agrupaciones y construir los grafos finales.

La similaridad de textos se ha calculado con SpaCy y el modelo en_core_web_lg. Para construir los grafos utilizamos la librería networkx [48]. La tabla 3.14 indica las métricas básicas de las agrupaciones encontradas.

.

Variable	Media observada	Media del muestreo	Desviación estándar observada	Desviación estándar del muestreo	Asimetría observada	Asimetría del muestreo
n_tokens	433.026	427.838	185.943	148.806	0.688056	-0.00158024
f_stopwords	0.48214	0.48412	0.0422274	0.0372889	0.178414	0.00592103
f_punct	0.0991162	0.0978853	0.0284867	0.0251154	0.487347	0.00265237
f_diff_lemmas	0.252555	0.248565	0.0503052	0.0487278	0.704491	-0.0032019
f_ents	0.0179832	0.0171119	0.0140688	0.0152035	1.37306	-0.00330824
f_oov	0.0152547	0.0154967	0.0139228	0.0144821	2.54677	-0.00199961
fk_reading_ease	71.6556	71.5836	9.65958	8.8973	0.011599	-0.000691442
fk_grade_level	7.73571	7.8317	2.18365	2.06335	0.909674	-9.38671e-05
dale_chall_readability	8.04453	8.04311	0.843694	0.710044	-0.746512	0.00252819
f_numeric_tokens	0.00360312	0.00339146	0.00448248	0.00415186	2.73953	0.00187517
precision	0.353096	0.351371	0.0467004	0.0436871	-0.0913424	-0.000221561
polarity_average	0.0005719	0.000167295	0.00569808	0.00516951	-0.179717	0.000629278
polarity_peak	-0.0695563	-0.0916895	0.521933	0.504697	0.181623	-0.000354694
style	0.0697189	0.0649736	0.06058	0.0567937	1.61822	-0.00171996
coherence	0.150577	0.10219	1.58691	1.61543	0.206594	0.00360717
n_fact	0.556225	0.453321	0.859233	0.7411	1.88286	9.51415e-05
n_testimony	0.74498	0.895922	1.5056	2.08233	3.53326	-0.00218393
n_value	16.6767	15.9298	9.55856	8.62766	1.27498	0.00093728
n_policy	2.85944	2.86964	2.16807	2.18468	0.955461	-0.00496226
n_reference	2.85944	2.86367	2.16807	2.18716	0.955461	0.0019784
n_links	2.5	2.53198	2.80216	2.97708	1.66059	-0.00567408
domain1_score	3.44779	3.37399	0.778409	0.778609	-0.223174	-0.00739487
domain2_score	3.33735	3.33872	0.741586	0.708399	-0.897511	-0.00110609

Cuadro 3.13.: Comparación de media, desviación estándar y asimetría («skewness») observadas en el corpus de validación y en atributos muestreados a partir de la distribución de probabilidad «a posteriori». Esta última distribución captura razonablemente bien media y desviación estándar, pero, como era de esperar en un modelo gaussiano, no puede representar la asimetría de la distribución real de datos

	Muestra de 61 ensayos	Muestra de 23 ensayos
Número total de cláusulas argumentativa	1323	393
Grupos totales	64	14
Grupos con más de una cláusula argumentativa	4	4
Porcentaje de llamadas a la acción	13.76 %	23.16 %
Porcentaje de opiniones	79.89 %	72.01 %
Porcentaje de testimonios	4.31 %	2.04 %
Porcentaje de hechos	2.04 %	2.8 %

Cuadro 3.14.: Métricas básicas de las dos muestras de ensayos analizadas para las que se extrajo el grafo colaborativo de argumentación

3.2. Discusión de resultados

3.2.1. Definición y cálculo de atributos

En este trabajo se ha propuesto un conjunto de atributos o rasgos del ensayo argumentativo, y se han calculado estos rasgos para un corpus concreto (Kaggle ASAP Phase 1 tema 2). Los métodos de cálculo se pueden utilizar en principio en otros ensayos en inglés y, como característica importante, no dependen explícitamente del dominio o el tema tratado en los ensayos.

En general, la extracción de estos atributos requieren utilizar modelos entrenados de aprendizaje automático y técnicas de Procesamiento de Lenguaje Natural (PLN). Por un lado hemos podido utilizar modelos ya entrenados y publicados (SpaCy.io y en_core_web_lg, el modelo de Wei Feng y Hirst para parseo de estructura retórica,

[134][133] y el modelo de Niculae, Park y Cardie [78] para minería de argumentos). Por otro lado, hemos tenido que definir y entrenar dos modelos para definir la coherencia y estilo como un atributo de un ensayo argumentativo.

Al extraer métricas con modelos pre-existentes (todas menos estilo y coherencia) y revisar su distribución en el corpus Kaggle ASAP Phase 1 tema 2, observamos que en general, varían sobre un rango más o menos amplio, por lo que es razonable esperar que puedan ser relevantes a la hora de caracterizar ensayos argumentativos. Una vez se haya aprendido el modelo bayesiano de indicadores de la competencia en argumentación, se puede comprobar qué atributos son irrelevantes, en el sentido de no estar correlacionados con el resto.

A priori, se observó que la frecuencia de URLs era cero en todo el corpus, por lo que se excluyó del estudio por irrelevante. La fracción de términos numéricos en el texto tiene mucho peso en cero, pero no la hemos excluido, ya que hay un número apreciable de ensayos que presentan alguno, por lo que es posible que sea relevante.

3.2.1.1. Coherencia del texto

Para definir la coherencía seguimos de cerca la referencia [62]. A partir de un corpus específico de ensayos, texto periodístico y de otros tipos se ha generado un corpus paralelo de texto que se considera incoherente (generado al desordenar aleatoriamente sus frases), para con los dos formar un corpus sintético. Los documentos de este corpus sintético se han vectorizando estimando la probabilidad de que una palabra participe en hasta tres tipos de relaciones retóricas consecutivas. Se ha utilizado una máquina de soporte vectorial de núcleo lineal para aprender el ranking de coherencia en estos vectores. El algoritmo elegido es clave para poder definir la métrica de coherencia, que está basada en la proyección sobre el vector normal a un hiperplano separado de documentos ordenados y desordenados.

Comparando los resultados de este trabajo con los de Lin, Tou Ng y Kan [62], se observa:

- Lin, Tou Ng y Kan utilizan como corpus artículos del Wall Street Journal obtenidos del corpus Penn Treebank sobre dos temas específicos. Utilizan 1040 para el entrenamiento y 1079 para la validación y ejecutan un preprocesado cuidadoso para eliminar expresiones típicas en ciertas secciones de artículos. Generan cerca de 20 versiones desordenadas de un documento.
    - En nuestro trabajo utilizamos un corpus con 1071 documentos y realizamos validación cruzada estratificada para estimar la capacidad de generalización. La única limpieza realizada es el encabezamiento de los documentos del corpus AAEC, que indica el tema del ensayo [110]. Se ha considerado que los textos no tienen estructuras adicionales que puedan perjudicar el rendimiento del modelo de coherencia.

    - Se ha generado únicamente un documento aleatorio por ensayo para reducir el tiempo total de procesamiento. Para reducir el riesgo que supone utilizar una muestra de datos más pequeña, hemos utilizado una validación cruzada con 3 particiones, para por un lado, reducir el riesgo de sobreajuste y que, por otro, las métricas de validación se calculen sobre más datos y podamos esperar que sean más precisas. Si los resultados de validación son buenos, el riesgo de haber perdido datos de interés al usar menos datos de entrenamiento («underfitting») es más pequeño.

- El modelo de Lin, Tou Ng y Kan se basa en el modelo formal PDTB de estructura retórica de textos. En este trabajo se ha utilizado el modelo RST («Rethorical Structure Theory»). Hemos utilizado directamente los dos parámetros del algoritmo de los autores (frecuencia de términos para considerar un término como saliente y número de relaciones retóricas consecutivas a considerar). Aunque se podría haber intentado hacer un ajuste propio, los resultados obtenidos son buenos y no se consideró prioritario un refinamiento del modelo.
- El número inicial de dimensiones es muy alto, pero muchas de las relaciones retóricas no se presentan de manera consecutiva, por lo que el número total de dimensiones se redujo (de más de ochenta mil a menos de seiscientos). En cualquier caso, una máquina de soporte vectorial puede dar buenos resultados cuando el número de atributos es muy elevado en comparación con el número de instancias de entrenamiento. Es importante indicar que los atributos se seleccionan sobre los documentos ordenados y se reutilizan sobre los documentos desordenados, para eliminar combinaciones inusuales de relaciones retóricas.
- Lin, Tou Ng y Kan reportan precisiones oscilando entre 0,85 y 0,89, dependiendo de diferentes variaciones de su algoritmo y el tema de los artículos. Para una combinación de su algoritmo con el de Barzilay y Lapata [8], reportan entre 0,89 y 0,92. En este trabajo encontramos una métrica F1 máxima calculada en datos de validación de de 0.88, con poca variabilidad (el mínimo es 0.82). Aunque la metodología en el presente estudio tiene diferencias en datos y metodología, se puede decir que el rendimiento del modelo de coherencia generado es cercano al reportado en [62].

La métrica de coherencia (véase en 3.5 la distribución de esta métrica en el corpus Kaggle ASAP Phase 1) es razonablemente suave sobre un rango, por lo que es razonable que sea relevante a la hora de describir las observaciones. Esto es un criterio necesario a la hora de definir rasgos de ensayos argumentativos que se puedan utilizar en el aprendizaje de modelos.

El modelo de coherencia no depende directamente de temas tratados o el vocabulario utilizado. El corpus utilizado en este trabajo tiene más variedad temática y estilística que el de Lin, Tou Ng y Kan, aunque sin duda tiene más ruido. En este trabajo hemos dado más prioridad a que un modelo generalice mejor en textos diferentes respecto a obtener resultados más precisos. Sin embargo, no hemos validado la capacidad de generalización en un corpus adicional, al considerarlo fuera del alcance de este

trabajo.

3.2.1.2. Estilo del texto

La estilometría es un área de estudio activa en lingüística computacional, centrada en diferentes tareas [76]: atribución de texto, verificación de autoría, categorización del autor usando perfiles generales o «author profiling», evolución del estilo de un autor con el tiempo («stylochronometry») y estudio de alteraciones en la atribución de textos (típicamente vía imitación, traducción u ofuscación). En este trabajo sin embargo, se busca una caracterización general de estilo de un texto que permita asignar uno o más rasgos de estilo a un texto, con la finalidad no tanto de estudiar la autoría sino para caracterizar cómo influye el estilo (entendido como «personalidad literaria», o una combinación de tendencias personales y la voluntad por expresarse de una forma determinada [67]) en la evaluación y caracterización de un ensayo argumentativo. El atributo de estilo que buscamos debería ser independiente del tema o del vocabulario utilizado, para aumentar la capacidad de generalizar a nuevos textos.

En principio este objetivo es muy general, por lo que se han tomado dos pasos clave para convertirlo en un problema tratable:

1. Utilizar una serie de rasgos basados en la función gramatical y la estructura sintáctica para realizar análisis de autoría de texto siguiendo de cerca la referencia [41]. Gamon extrae estos atributos a partir de fragmentos de texto para atribuirlos a autores que se puede asumir tendrá estilos literarios cercanos. En este trabajo se consideran interesantes por ser independientes del vocabulario o tema del texto y por la capacidad discriminadora que Gamon muestra en su artículo.
2. Simplificar el concepto de estilo asumiendo que para ensayos persuasivos nos basta con poder caracterizar lo próximo que está el texto de un estudiante a un ensayo o artículo más profesional. Tal como ya se indicó, creamos un corpus pensando en estas dos referencias básicas (estudiantes versus texto periodístico y ensayos de autores reconocidos) y entrenamos un clasificador binario que nos permita asignar probabilidades de pertenencia a un clase. Esta probabilidad se interpretará como el atributo de estilo buscado.

A la hora de elegir un algoritmo clasificador, un bosque aleatorio es una buena opción, por ser robusto (se espera que en principio tenga poca variabilidad y un sesgo más pequeño que el de un árbol de decisión) y está pensado para manejar muchos atributos (en nuestro caso, mas de treinta mil, y muy pocas constantes para todos los datos de entrenamiento). Por otro lado, no hay una motivación geométrica, como si la había en el modelo de coherencia, que nos sugiera otro tipo de algoritmo.

De nuevo se ha utilizado validación cruzada estratificada con tres particiones para entrenar y validar el modelo usado para caracterizar el estilo de un texto. Esperamos

que el menor tamaño de datos de entrenamiento reduzca el riesgo de sobreajuste y, con métricas de validación más precisas, podamos reducir el riesgo de haber perdido patrones de interés durante el entrenamiento. Otro factor a tener en cuenta es la descompensación de categorías (ensayo de estudiante versus profesional) en el corpus utilizado, que puede aumentar artificialmente la métricas agregadas.

Así, los resultados muestran una precisión y exhaustividad máximos de uno (entrenamiento y validación). Si observamos la precisión y exhaustividad para la clase minoritaria, seguimos observando una precisión y exhaustividad perfectas, pero con mayor variabilidad. A partir de estas observaciones, consideraremos que el bosque aleatorio entrenado como clasificador va a tener una alta precisión sobre texto nuevo (en principio cercana a 1), pero una exhaustividad mucho más variable (entre 0,72 y 1).

Si se observa la distribución de la métrica de estilo, definida como la probabilidad de que el estilo del texto sea periodístico/profesional, (véase la figura 3.5 para el corpus Kaggle ASAP Phase 1), aunque hay discontinuidades es aproximadamente suave sobre un intervalo, por lo que, de nuevo, es razonable pensar que es relevante para caracterizar ensayos argumentativos.

Esta definición de estilo se puede generalizar añadiendo diferentes referencias de estilo al corpus de entrenamiento. Dependiendo del número de referencias se podría intentar utilizar un clasificador multi-clase para asignar una métrica de estilo, que en este caso tendría varias dimensiones. Esta generalización está fuera del alcance de este trabajo.

3.2.2. Entrenamiento de la red bayesiana para evaluación del ensayo argumentativo

En este trabajo se ha aprendido la estructura y probabilidades condicionales de una red bayesiana a partir de un corpus de ensayos argumentativos anotados con evaluaciones en dos dominios, efectividad de la expresión y dominio del lenguaje. La red tiene un total de 23 variables aleatorias (véase 3.8), que incluye las evaluaciones agregadas para los dos dominios.

Como se comentó en el capítulo 2, al utilizar una red bayesiana para modelar la distribución de probabilidad conjunta de estos 23 atributos, tenemos, por un lado, flexibilidad en las posibles aplicaciones, ya que al conocer la distribución de probabilidad conjunta, podemos marginalizar y realizar diferentes tipos de inferencia (utilizando cualquiera de los algoritmos de inferencia en red bayesiana existentes, o vía integración numérica), y por otro, tenemos una representación gráfica del conocimiento que es mucho más fácil de visualizar y utilizar para discutir posibles relaciones de causalidad.

La red bayesiana se ha construido como un modelo parametrizado, en el que la probabilidad condicionada de cada variable se modela con una distribución normal cuya

media es combinación lineal de los valores de sus padres en la red. Los coeficientes de esta combinación lineal y la desviación estándar son parámetros libres para los que se define una distribución «a priori» fija. Estas distribuciones se han elegido para permitir explorar un rango amplio de valores durante la inferencia del modelo (por ejemplo, usando una desviación estándar grande para los «priors» normales).

Al utilizar un modelo normal o gaussiano y lineal estamos limitando la expresividad del modelo. La elección de «priors» para los parámetros libres nos permite reducir el sesgo, pero no aumenta el tipo de observaciones que se pueden reproducir. Por ejemplo, algunas de las distribuciones observadas para los atributos de los ensayos en el corpus Kaggle ASAP Phase 1 no son simétricas, y algunos de estos rasgos son valores acotados o incluso discretos, por lo que nuestra red bayesiana no podrá reflejar esta información (véase el apéndice A.3 para más detalles). A cambio, el proceso de inferencia bayesiana con el que se aprende el modelo es mucho más robusto en el caso gaussiano que al intentar utilizar distribuciones que permitan, por ejemplo, asimetría o tengan un soporte acotado o discreto. Adicionalmente, tampoco tenemos información que nos sugiera una distribución concreta, por lo que una distribución normal, bastante común en la literatura, es razonable, reconociendo sus límites.

Como parte del experimento se han inferido las calificaciones de un conjunto de ensayos obtenido del mismo corpus de entrenamiento, pero apartados durante este. En principio se podría realizar la inferencia a partir del valor esperado de la variable calificación (E) respecto a la distribución de probabilidad condicionada $P(E|X = x)$. Sin embargo, seguiremos el criterio típico del aprendizaje bayesiano ([72], pg. 159) y utilizaremos el máximo «a posteriori» (MAP) de E una vez observados $X = x$. El valor esperado describe mejor el promedio de calificaciones que observaríamos en diferentes observaciones, pero el MAP es más fácil de calcular. Los resultados (error cuadrático medio) son comparables a los de modelos regresores específicos descritos en la sección 3.1.4.1.

Como validación adicional se ha generado un muestreo de atributos a partir de la distribución de probabilidad aprendida y se han comprobado varias estadísticas básicas entre cada atributo por separado. La media y desviación estándar de observación y muestreo son relativamente cercanas (diferencias cercanas al 10 %), pero la red bayesiana falla al capturar la asimetría, tal como se esperaba (ver apéndice A.3)

Observando la red bayesiana obtenida a partir del entrenamiento, se observa que el atributo «n_policy» (número de «llamadas a la acción» encontradas en el texto), que es parámetro directamente relacionado con los argumentos construidos en el texto, es independiente del resto de atributos, por lo que no contribuye a evaluar el discurso ni explicar el comportamiento de otras variables. Esta observación está en desacuerdo con la idea de que la relación entre partes de un argumento es importante. Hay dos factores que podrían explicar este resultado:

1. Los modelos utilizados en la extracción de atributos tienen un margen de error todavía elevado. Para la extración de argumentos, Niculae, Park y Cardie

reportan métricas F1 entre 0,38 («Facts») y 0,79 («Policy» o llamada a la acción) y F1 de 0,24 para identificación de enlaces. Es posible que al utilizar directamente el número de sentencias argumentativas y enlaces, el modelo sea más sensible a estos errores que en el caso, por ejemplo de la coherencia. Para calcular esta métrica se utiliza un modelo de parseo de estructura retórica con un error relevante (F1 de 0,86 para identificación de unidades elementales de discurso, pero menos de 0,4 en la detección de relaciones retóricas), pero este modelo no se utiliza directamente, sino que alimenta un segundo modelo que aprende la coherencia de un texto basándose en sus resultados. Se podría decir que este segundo entrenamiento aprende y asume implícitamente el margen de error si logramos que las métricas de validación sean lo suficientemente buenas.

2. Algún otro atributo que no se ha considerado podría estar correlacionados con «n_policy» y el resto de atributos.

El segundo punto es difícil de tratar, y posiblemente necesitaría una revisión exhaustiva de otro tipo de atributos del texto, o incluso una aproximación diferente a la de este trabajo. Por ejemplo, dentro del paradigma del aprendizaje profundo, modelos de atributos de alto nivel que se aprenden directamente de atributos lingüísticos de bajo nivel, y que fueran la entrada para el aprendizaje de un modelo de la estructura del ensayo. Véase [64][61] para un proyecto semejante aplicado al parseo de la estructura retórica. Sin embargo, por las referencias que hemos podido encontrar, la utilización de estas técnicas en la minería de argumentos es todavía un punto abierto, y además la interpretación de los atributos generados a partir de datos no está clara en principio.

Para el primer punto, podría ser interesante definir atributos de la argumentación a partir de modelos aprendidos de las métricas que estamos utilizando en este trabajo. Al igual que con la coherencia, podríamos esperar que el error del primer modelo se asuma implícitamente en el segundo modelo.

Estos dos últimos puntos están fuera del alcance de este trabajo. Respecto a la asunción de que la red bayesiana que se ha aprendido esta asumiendo el margen de error en el cálculo de atributos, nos limitamos a evaluar las métricas de evaluación ya indicadas e intentar localizar problemas de falta de expresividad.

Tal como se describe en la sección 2.4.3, se ha generado una recomendación muy básica para mejorar la evaluación de ambos dominios basado en el gradiente del valor esperado de cada calificación respecto a la probabilidad condicionada a un valor fijo del resto de atributos, incluyendo la calificación restante (véase las figuras 3.11 y 3.12). Como ya se comentó, en una red bayesiana gaussiana normal, este valor esperado es una función lineal (véase apéndice A.3), por lo que el gradiente es constante y la recomendación es independiente del valor de los atributos.

Es interesante observar que la calificación en el dominio 2, dominio del lenguaje, está correlacionado directamente con la calificación del dominio 1, efectividad en la expresión. Esto se observa tanto en el grafo dirigido de la red bayesiana (3.10) como en los resultados de la recomendación para el segundo dominio.

3.2.3. Grafo de argumentación colaborativo

El algoritmo descrito en el capítulo 2 (2.5.2) se puede considerar un algoritmo no supervisado de resumen de argumentos utilizados en un corpus de ensayos. El objetivo de generar grafos de argumentación colaborativos es proporcionar una visualización sencilla de los argumentos utilizados, lo que puede utilizarse para dirigir una discusión sobre su calidad y cómo mejorarlos o complementarlos.

En principio podemos esperar una serie de limitaciones al grafo que obtenemos:

- Precisión limitada en el modelo de minería de argumentos e identificación de paráfrasis utilizados.
- El algoritmo busca formar grupos de cláusulas expresando la misma idea (paráfrasis), pero el criterio de agrupación es estricto, por lo que quedan muchas clases con una única frase.
- El modelo de minería de argumentos no contempla ataques a posiciones, lo que limita su expresividad.

Como se observa en los resultados obtenidos al analizar dos conjuntos relativamente reducidos de ensayos (61 y 23 documentos respectivos), aunque el número de grupos (o cláusulas típicas) con un única frase es alto, se han podido encontrar argumentos estructurados que son susceptibles de discusión en un grupo, ya que presentan una llamada la acción y testimonios, hechos u opiniones que pueden apoyar esta llamada (3.13,3.14). De hecho, tras filtrar grupos con una única cláusula, no hay clases desconectadas, por lo que cada grafo representa realmente un único argumento. Sin embargo, no se espera que esto suceda con todas las muestras de documentos. En general, deberíamos esperar grupos desconectados y uno o más argumentos total o parcialmente estructurados.

Este buen resultado cualitativo apoya la idea de que utilizar la paráfrasis como criterio para resumir da buenos resultados. Sin embargo hay que notar que, si se revisa una muestra de fragmentos de texto que correspondan a los argumentos indicados en los dos grafos (véase 3.15, 3.16), aparte de paráfrasis se observan proposiciones de clara intención discordante. Esto es una indicación de que el modelo de paráfrasis entrenado generaliza peor de lo esperado en el tipo de texto utilizado. Aunque no podemos sacar conclusiones definitivas a partir de una muestra tan pequeña, es una muestra clara de que es probable que el grafo colaborativo está sobrestimando algunas clases, por lo que no se puede usar para evaluar cuantitativamente los argumentos utilizados.

Pese a estas limitaciones, a partir de un grafo como el obtenido en las figuras 3.13,3.14, se puede plantear guiar una discusión a través de una serie de preguntas dirigidas a criticar el argumento y buscar mejorarlo:

- ¿Es inteligible el contenido de cada una de las cláusulas argumentativas tipo? Aunque es texto real, cabe la posibilidad de que el representante elegido no

sea el mejor. Para ello, en una aplicación real, el usuario debería ser capaz de cambiar el texto (manualmente o eligiendo otro representante).

- ¿Son correctas las relaciones entre cláusulas así como la fuerza de cada apoyo? Es posible que la dirección de una inferencia no sea correcta, o que un apoyo corresponda a un ataque. En una aplicación real, nodos y relaciones deberían poder etiquetarse.
- ¿Hay alguna premisa o ataque que no aparezca en el argumento conectado? En una aplicación real se necesitaría la posibilidad de revisar clases desconectadas o añadir grupos nuevos.
- ¿Qué otros argumentos se pueden formar, ya sea a partir de alguna de las clases existentes o algún nuevo grupo?

Si se plantean los aspectos positivos del análisis (obtención de grafos de argumentos de manera no supervisada) con los problemas encontrados (el modelo de paráfrasis tiene posiblemente menos precisión y exhaustividad de la que estimamos) y el riesgo (no concretado) del impacto que el margen de error del modelo de minería de argumentación presenta, podemos plantear una serie de mejoras para reducir estos efectos sin perder las ventajas de utilizar un algoritmo de aprendizaje automático:

1. Pasar de un modelo no supervisado a uno supervisado con entrenamiento en línea (esto es, en el que se tenga en cuenta las entradas y cambios propuestos por usuarios para mejorarlos)
2. Revisar el método de agrupación para que, además de la paráfrasis, considere otros criterios, como, por ejemplo, hasta qué punto los fragmentos de texto considerados representan una inferencia lógica (RTE, «Recognize Text Entailment»). En la referencia [106] plantean un método interesante para estimar esta relación de RTE basándose en un grafo de relaciones entre conceptos extraído de WordNet.
3. Intentar la mejora del modelo de paráfrasis, re-entrenando en un corpus anotado más amplio para poder encontrar mejores resultados para texto nuevo. Crear nuevos corpus requiere un esfuerzo considerable, por lo que podemos intentar alternativas como reutilizar corpus con argumentos anotados (por ejemplo AAEC o CDCP):
   a) El objetivo es aprender un modelo de paráfrasis que optimice dos objetivos. El primer objetivo es la métrica de validación en un corpus de paráfrasis (Micrososft Research Paraphrase Corpus, por ejemplo)
   b) El segundo objetivo serían métricas de evaluación en un corpus de argumentación, siguiendo estos pasos:
      1) Se extraen las clases de argumentos típicos del corpus de argumentación.
      2) Se definen enlaces entre grupos utilizando parte del corpus de argumentación.

3) Estos enlaces de grupo permiten estimar enlaces en el corpus de argumentación no utilizado en el paso anterior. Al comparar con los enlaces anotados se obtiene la segunda métrica de validación.

Hay otras opciones, como utilizar modelos más sofisticados para estimación de la paráfrasis.

A añadir también que el algoritmo propuesto escala cuadráticamente con el número de clausulas argumentales, por lo que en esta primera versión solo puede manejar un número pequeño de ensayos en un tiempo razonable. Todos estos puntos abiertos que se han descrito están fuera del alcance de este trabajo.

3.3. Conclusiones

Como se mencionó en el capítulo 1, la competencia en argumentación y el desarrollo del pensamiento crítico son cruciales en los planes educativos del siglo XXI. El progreso en sistemas computacionales que apoyen el desarrollo de estas competencias, aunque ha atraído interés por parte de numerosos grupos, ha tenido un desarrollo desigual a la hora de aplicar técnicas de Inteligencia Artificial (IA). Eso es debido, a nuestro entender, a las dificultades que entraña el problema de caracterizar estas competencias. Teniendo en cuenta este contexto, consideramos que nuestro trabajo hace una aportación relevante al estudio de la aplicabilidad de la IA en sistemas educativos orientados al soporte del aprendizaje de la competencia en argumentación.

Al revisar los objetivos de este trabajo junto con los resultados obtenidos, podemos concluir:

1. Se ha entrenado y validado un modelo bayesiano de indicadores de la competencia en argumentación. Este modelo es una red bayesiana paramétrica, cuya estructura y probabilidades condicionadas se han aprendido directamente de un corpus de ensayos, y que aproxima la distribución de probabilidad de una serie de atributos indicadores de diferentes competencias argumentales.

   a) Pese a las simplificaciones realizadas (utilizamos un modelo gaussiano lineal), el modelo tiene un buen poder predictivo, pero no permite, por ejemplo, describir la asimetría observada en la distribución de algunos de los atributos del ensayo.

   b) El modelo permite efectuar recomendaciones de mejora de competencias argumentales utilizando un criterio bien fundamentado. Al ser un modelo gaussiano lineal, estas recomendaciones son globales para cualquier ensayo. Sin embargo con otro tipo de parametrización que tenga más poder predictivo, se podrían hacer recomendaciones específicas dependientes de los atributos del ensayo.

2. Se ha definido un algoritmo de visualización de argumentos típicos en un corpus de ensayos en inglés. Este es un algoritmo no supervisado, basado en un modelo de paráfrasis (frases de estructura y forma diferentes que expresan ideas muy similares), que hemos entrenado aparte y que nos permite definir un grafo de argumentación colaborativo que resume los argumentos más utilizados en el corpus. Aunque no se ha realizado una validación exhaustiva, se han analizado un par de muestras de ensayos y se ha observado que el algoritmo puede identificar estructuras de argumentos, aunque la precisión y exhaustividad son un punto a fortalecer.
    a) En nuestra opinión, los grafos generados, con la ayuda de las preguntas adecuadas, pueden ser un buen soporte para guiar una discusión sobre la calidad de los argumentos detectados, posibles premisas ausentes y otros aspectos de técnica argumentativa.

Para definir el modelo bayesiano de indicadores de competencia en argumentación, hemos definido una serie de atributos del texto que pueden ser indicadores de diferentes competencias argumentales. Hay varias argumentos para realizar esta afirmación:

- Los atributos elegidos están parcialmente alineados con criterios estándar de pensamiento crítico incluidos en el formalismo de pensamiento crítico de Elder y Paul [35]. En nuestra opinión, es razonable asociar ciertas estadísticas (proporción de palabras diferentes, precisión de las palabras utilizadas considerando hiperónimos e hipónimos) con competencias lingüísticas (precisión en la expresión, riqueza de vocabulario).
    - Aunque esta asociación entre atributos y criterios es necesariamente una aproximación muy simplificada, al añadir al modelo bayesiano de indicadores de competencia criterios más generales aprendidos directamente de anotaciones (efectividad de la expresión y dominio del lenguaje), estamos mejorando la expresividad del modelo en la caracterización de competencias en argumentación. En la documentación adjunta al corpus Kaggle ASAP Phase 1 se puede encontrar una rúbrica detallada para estos dos criterios.
- La conexión entre atributo y competencia lingüística se ha fundamentado más fuertemente para el estilo, la coherencia y las estructura de los argumentos. Para los dos primeros, basados en referencias previas, se han entrenado y validado modelos de aprendizaje automático específicos. No se han encontrado referencias sobre la utilización de clasificadores para caracterizar el estilo de un texto, por lo que en principio, es un aportación original.
    - Aunque el modelo de coherencia de texto sigue de cerca la metodología de la referencia [62], el utilizar un modelo de estructura retórica RST es una aportación que no hemos encontrado en otras referencias, por lo que es, en principio de nuevo, una aportación original.
    - En el caso de la estructura argumental se ha utilizado un modelo de minería de argumentos [78] basado en un modelo formal de la argumentación

([86]).

- Los atributos elegidos han permitido definir un par de modelos de aprendizaje automático con una capacidad predictiva razonable: por un lado, el modelo bayesiano de indicadores de competencia en argumentación, y por otro un modelo de regresión basado en los atributos definidos.

Uno de los puntos claves en la selección de los atributos utilizados en el modelo bayesiano es que no dependen directamente de un vocabulario o temática específica, por lo es razonable pensar que estos modelos se pueden transferir fácilmente a ensayos argumentativos de temas diferentes. También podemos pensar que la principal limitación para reutilizar el modelo bayesiano es el corpus de ensayos utilizados, que en este trabajo estaba centrado en un grupo de edad específico (unos 15 años). Sin embargo, utilizando los mismos atributos, el modelo podría ser reutilizado tras ser re-entrenado en un corpus nuevo.

Para definir el algoritmo de extracción de un grafo de argumentación colaborativo a partir de ensayos, se ha entrenado y validado un modelo de aprendizaje automático de detección de paráfrasis. Esto nos ha permitido agrupar las sentencias argumentativas detectadas en un subconjunto relativamente pequeño de ensayos en clases de equivalencia con relaciones de apoyo entre ellas. En este trabajo se considera que estas clases y las relaciones entre ellas se pueden considerar un resumen de los argumentos más utilizados en los ensayos analizados. Este grafo se denomina colaborativo ya que se forma a partir de ensayos de diferentes estudiantes.

Aunque se han manejado muestras pequeñas de documentos, sobre todo porque el algoritmo escala con el cuadrado del número de cláusulas argumentativas, los resultados de visualización son prometedores, pero hay indicaciones de que la precisión y exhaustividad del modelo de predicción de paráfrasis en el corpus de ensayos es más baja que la estimada al entrenar el modelo. Consideramos que esto no invalida el algoritmo, ya que el grafo generado puede ser útil como una primera aproximación para soportar la discusión de los argumentos utilizados, su calidad y posibles premisas ausentes.

En principio, no se han encontrado referencias previas a la utilización de métodos de agrupamiento similares para detectar argumentos compartidos en varios textos, por lo que este algoritmo sería una aportación original.

En resumen:

- Se ha podido definir y validar un modelo bayesiano de indicadores de la competencia en argumentación para ensayos persuasivos, junto con diferentes atributos de este tipo de ensayos que se pueden identificar con competencias argumentativas. Para generar estos atributos se han utilizado modelos ya existentes o se han entrenado y validado modelos de aprendizaje automático, incluyendo algunas aportaciones originales.
- Se ha podido definir un algoritmo no supervisado para generar grafos argumentativos colaborativos. No se ha validado completamente, pero los resultados

parciales en muestras pequeñas son prometedores

- Han quedado puntos abiertos respecto al modelo bayesiano de indicadores de la competencia en argumentación y el algoritmo de extracción de grafo de argumentación colaborativo. Estos puntos, o se excluyeron inicialmente del alcance del trabajo o no se han abordado por su complejidad:
  - Estrategia de mitigación del impacto que tiene el margen de error de los modelos de minería argumental y detección de paráfrasis. Los modelos de estilo y coherencia presentan errores más bajos, por lo que tratarlos es menos prioritario.
  - Validación del algoritmo de generación de grafos colaborativos más allá del análisis realizado.
  - Metodología de aplicación y validación de los modelos y algoritmos desarrollados en la educación en competencias argumentativas. En este trabajo se han dado indicaciones breves de cómo se podrían utilizar, pero es necesario un desarrollo formal antes de poder aplicarlos a casos reales.

3.4. Trabajo futuro

Junto a los puntos abiertos mencionados en las conclusiones, este trabajo señala diferentes líneas de trabajo que permitirían extender los resultados y mejorar la aplicabilidad de los modelos y algoritmos de aprendizaje automático propuestos:

- Considerar un conjunto más amplio de atributos del ensayo argumentativo y, al igual que para el caso del estilo y la coherencia, utilizar modelos de aprendizaje automático para generar una interpretación. De esta manera se podrían realizar aproximaciones más sistemáticas a la caracterización de competencias.
- Investigar más a fondo el modelo bayesiano de indicadores de la competencia en argumentación: Posibles estructuras alternativas de la red bayesiana y considerar un modelo con varias redes bayesianas y un método de votación para proporcionar resultados.
- Extensión del estudio a ensayos en español. El principal obstáculo sería encontrar corpus de entrenamiento y validación adecuados, sobre todo a la hora de entrenar modelos de minería argumental, paráfrasis, y coherencia.
- Evaluar la transferencia de los modelos a otro tipo de ensayos, por ejemplo, escritos por estudiantes en diferentes rangos de edad al considerado en el presente trabajo.
- Mejora de la capacidad de generalización del modelo de identificación de paráfrasis, para así aumentar la precisión de los grafos colaborativos. Sería interesante obtener un algoritmo de resumen más escalable que el presentado en este trabajo.

- Introducir un estudio experimental con un grupo de estudiantes. Requeriría una formulación detallada de la metodología de uso de los modelos en un escenario educativo así como de evaluación de los resultados.
- Desarrollar un sistema recomendador basado en el modelo bayesiano de indicadores de la competencia en argumentación y, posiblemente, otras técnicas como filtrado colaborativo [16], además un catálogo de recursos educativos.

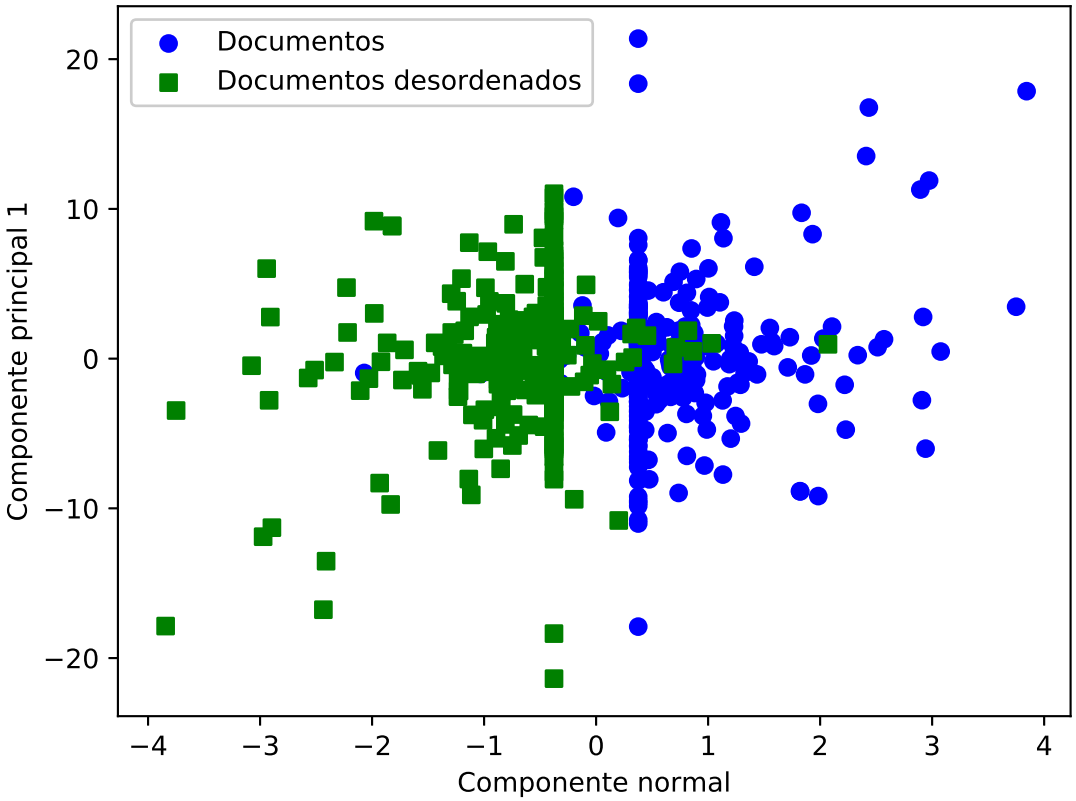

Figura 3.1.: Representación de los vectores diferencia entre documentos y documentos desordenados. Los puntos de color diferente están relacionados por un cambio de signo, que se observa en la simetría respecto al centro entre los dos subconjuntos. El eje X es la componente normal al hiperplano separador de menor riesgo. La componente Y es la primera componente principal de los vectores proyectados en el hiperplano separador. Los dos conjuntos están claramente separados, aunque hay un solape directamente relacionado con la precisión de nuestro modelo

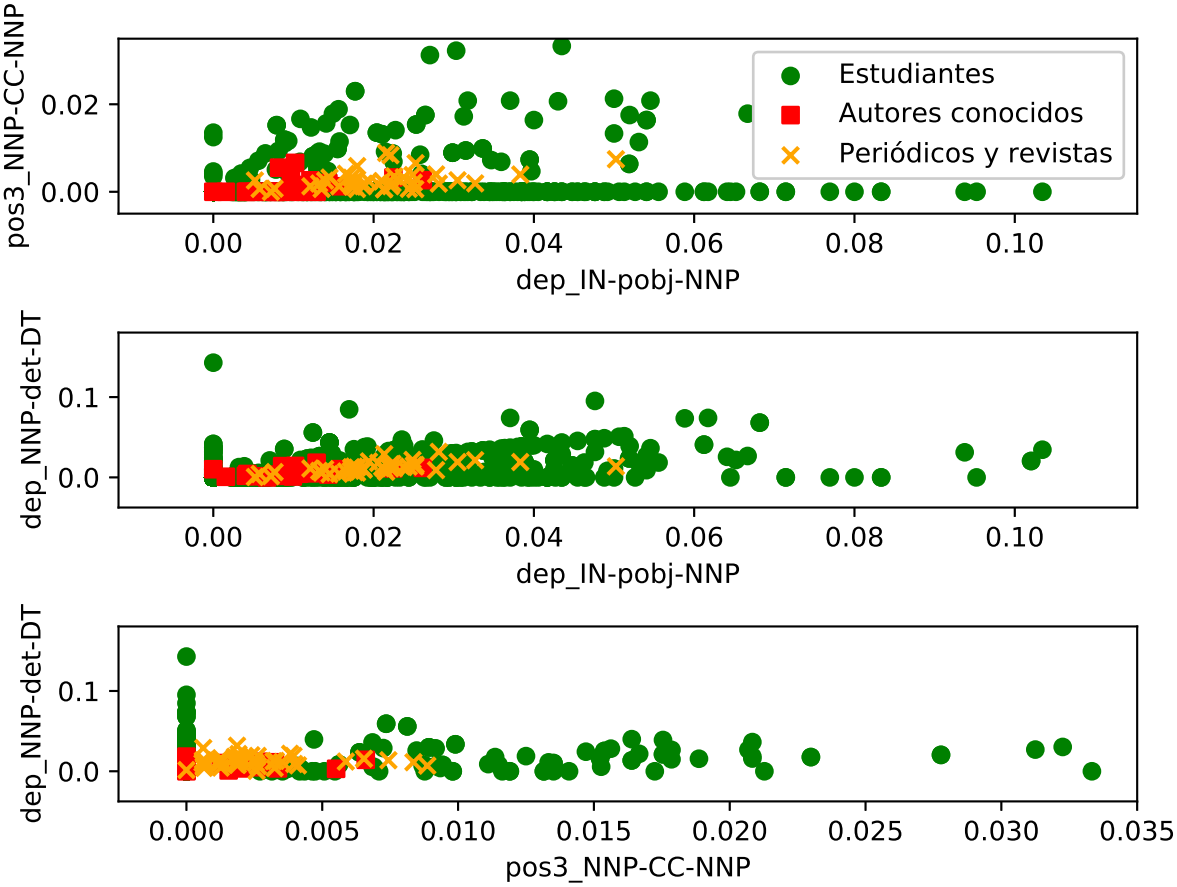

Figura 3.2.: Documentos utilizados en el entrenamiento y test del modelo de estilo, proyectados en las tres primeras dimensiones más relevantes

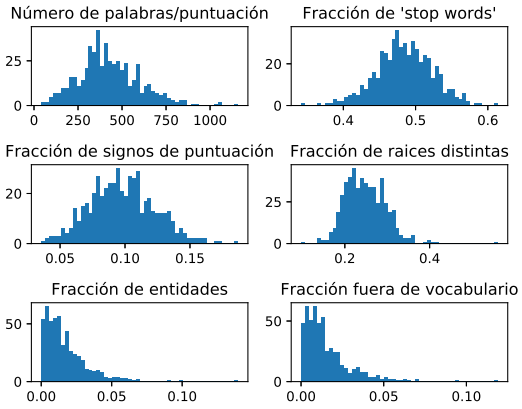

Figura 3.3.: Distribución de métricas para los datos Kaggle ASAP Phase 1 tema 2: Métricas básicas

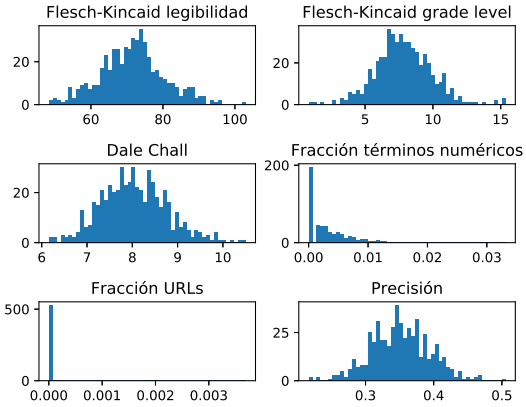

Figura 3.4.: Distribución de métricas para los datos ASAP Phase 1 tema 2: Legibilidad y precisión

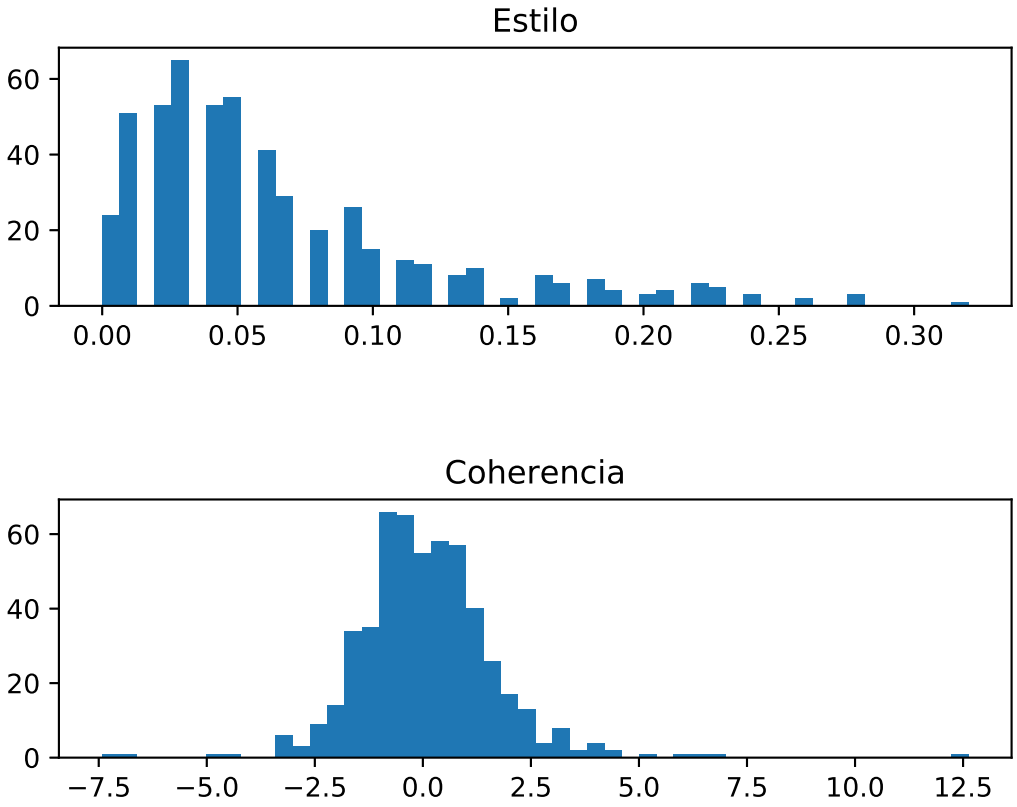

Figura 3.5.: Distribución de métricas para los datos Kaggle ASAP Phase 1 tema 2: Estilo y coherencia

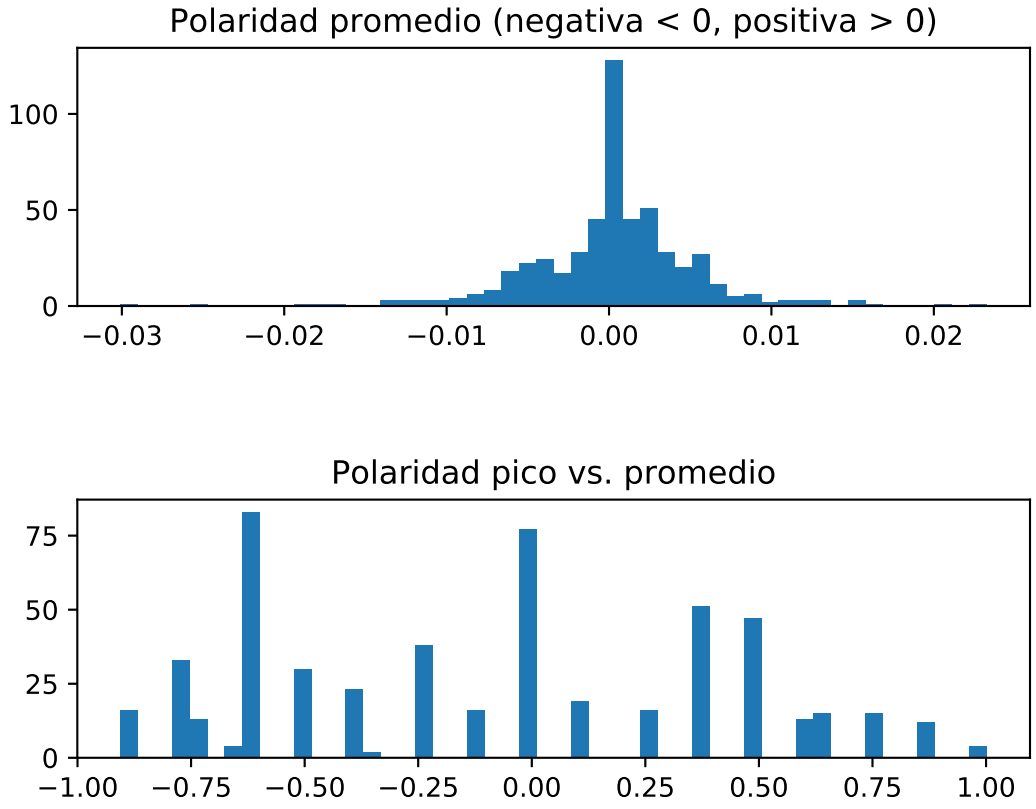

Figura 3.6.: Distribución de métricas para los datos Kaggle ASAP Phase 1 tema 2: Polaridad/sentimientos

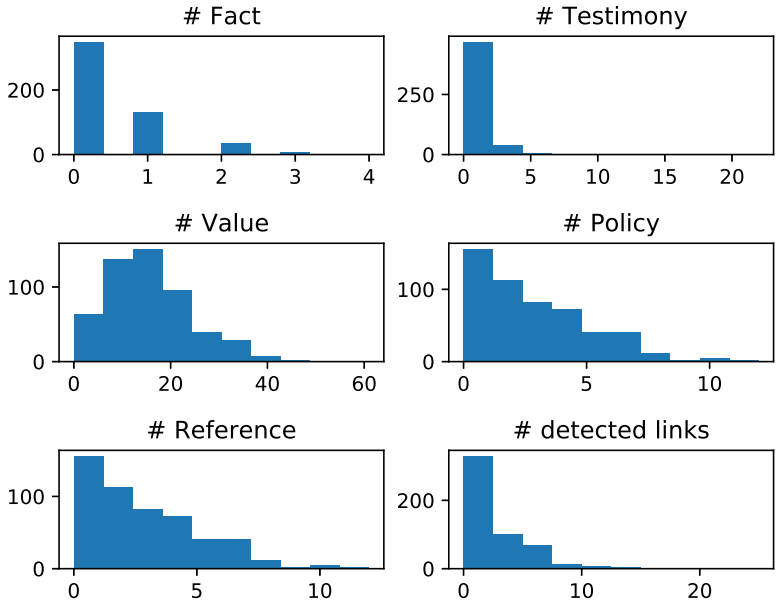

Figura 3.7.: Distribución de métricas para los datos Kaggle ASAP Phase 1 tema 2: Argumentación

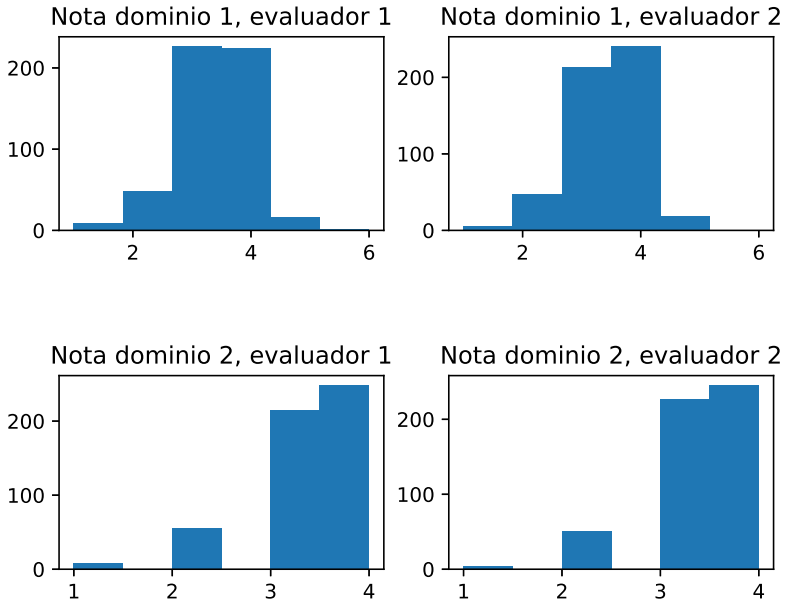

Figura 3.8.: Distribución de métricas para los datos Kaggle ASAP Phase 1 tema 2: Evaluaciones por dominio y examinador. El dominio 1 corresponde a la efectividad en la expresión, y el dominio 2 al dominio del lenguaje

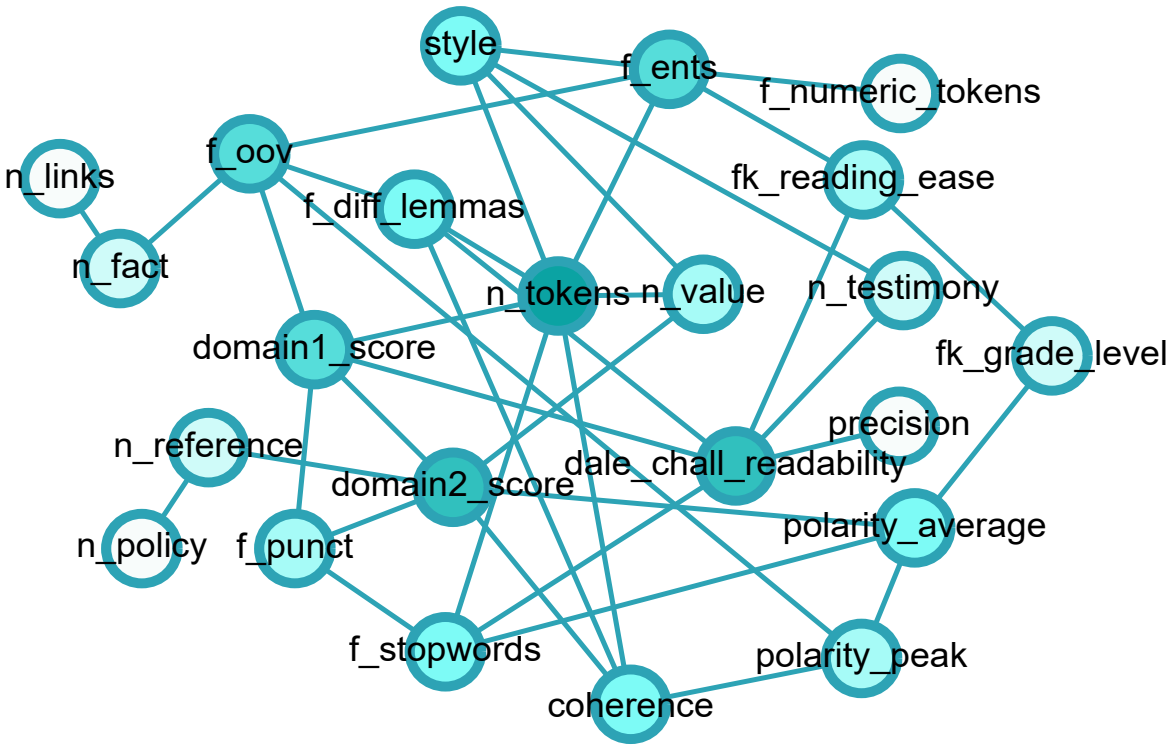

Figura 3.9.: Grafo indirecto que representa la independencia condicional de los documentos del corpus Kaggle ASAP Phase 1 tema 2 seleccionados para el entrenamiento del modelo de indicadores de la competencia en argumentación. Resultado del paso MMPC

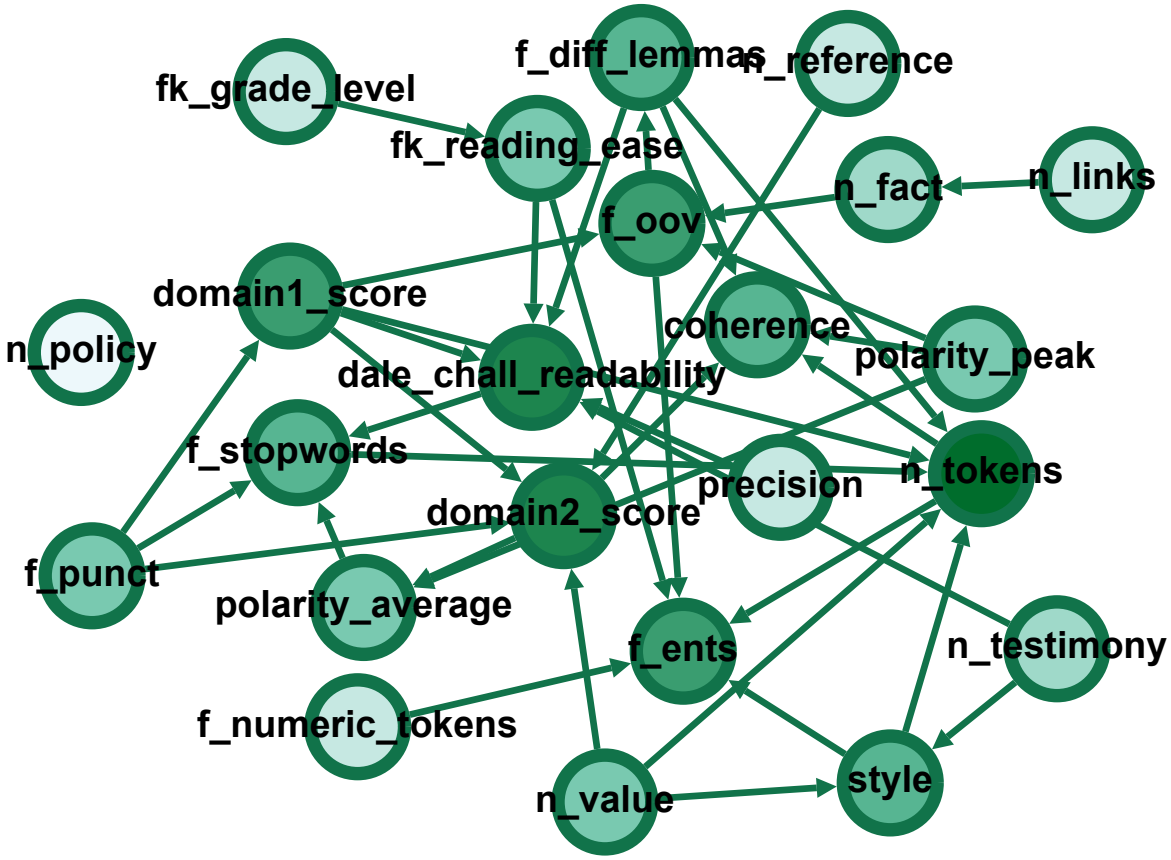

Figura 3.10.: Grafo dirigido acíclico de la red bayesiana entrenada sobre el corpus Kaggle ASAP Phase 1 tema 2. El número de llamadas a la acción (n_policy) se puede considerar independiente

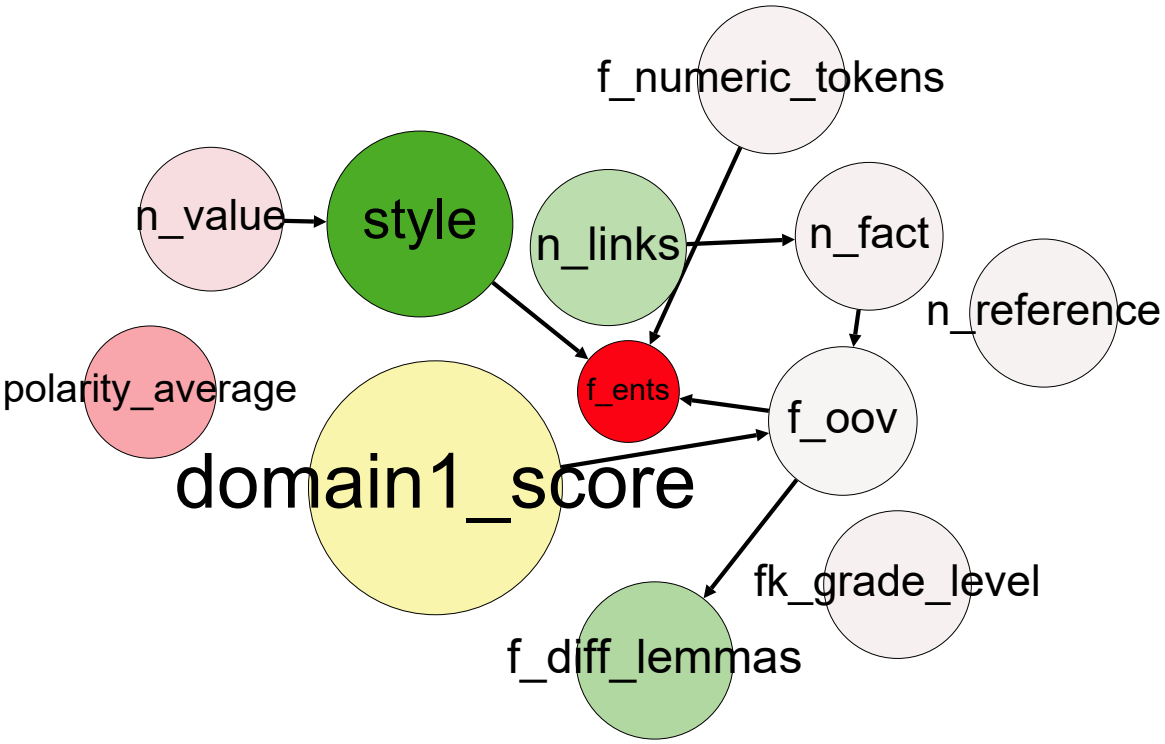

Figura 3.11.: Recomendación global para mejorar la calificación en el dominio 1, efectividad en la expresión. El gradiente de colores es proporcional al gradiente calculado (verde positivo, rojo negativo). Se observa que la recomendación general es utilizar un estilo más profesional y, con menos énfasis, utilizar más variedad de palabras e intentar expresar claramente las relaciones entre frases argumentativas, «n_links» (por ejemplo, opiniones o testimonios relacionadas como parte de un argumento). Se recomienda con énfasis disminuir el número de entidades (nombres de personas, lugares, organizaciones) en el texto. Cambien conviene reducir el número de palabras que expresan un sentimiento fuerte, así como el número de opiniones. El resto de atributos mostrados no tienen una influencia clara. El grafo indica únicamente componentes del gradiente con un valor apreciable. Las fechas indican las relación padre-hijo en la red bayesiana original. Los nodos pueden aparecer desconectados si el camino a las variables representadas incluyen nodos en los que el gradiente es muy bajo

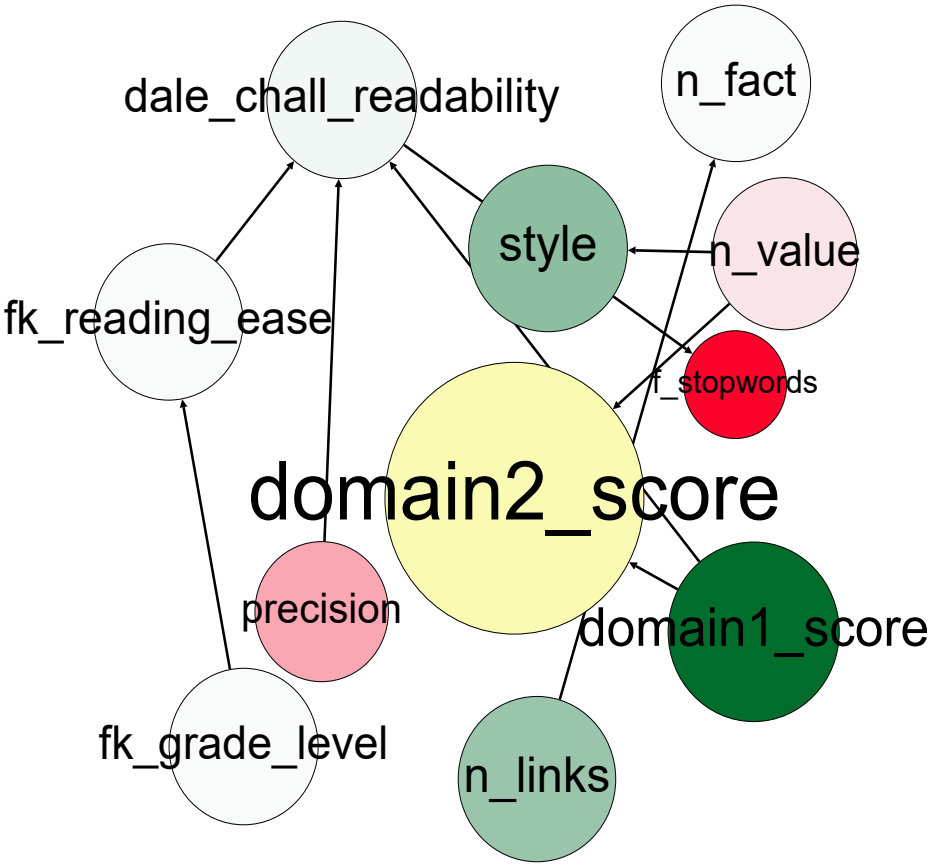

Figura 3.12.: Recomendación global para mejorar la calificación en el dominio 2, dominio del lenguaje. El verde más vivo indica atributos cuyo crecimiento ayuda a subir la calificación. El rojo indica atributos cuyo valor tiene que disminuir. Se observa una dependencia muy fuerte con la evaluación del dominio 2. De nuevo hay énfasis en mejorar el estilo y expresar claramente las relaciones entre frases argumentativas. Se recomienda reducir el número de opiniones («n_value»), la precisión media de las palabras («precision») y, con énfasis, el número de «stop words» (palabras sin contenido semántico real)

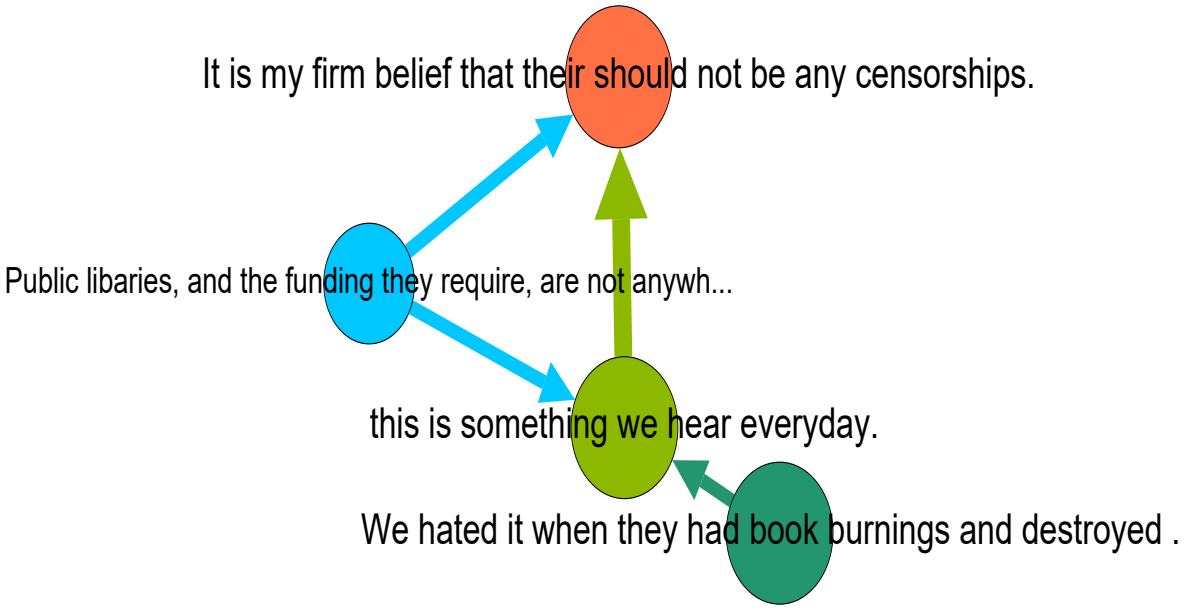

Figura 3.13.: Grafo colaborativo para 61 ensayos extraídos el corpus Kaggle ASAP Phase 1. El tamaño del enlace es proporcional a su peso o fortaleza. Las llamadas a la acción están representadas en color naranja, los hechos en color azul, opiniones en verde claro, testimonios en verde oscuro. Se han filtrado los grupos con una única cláusula argumentativa

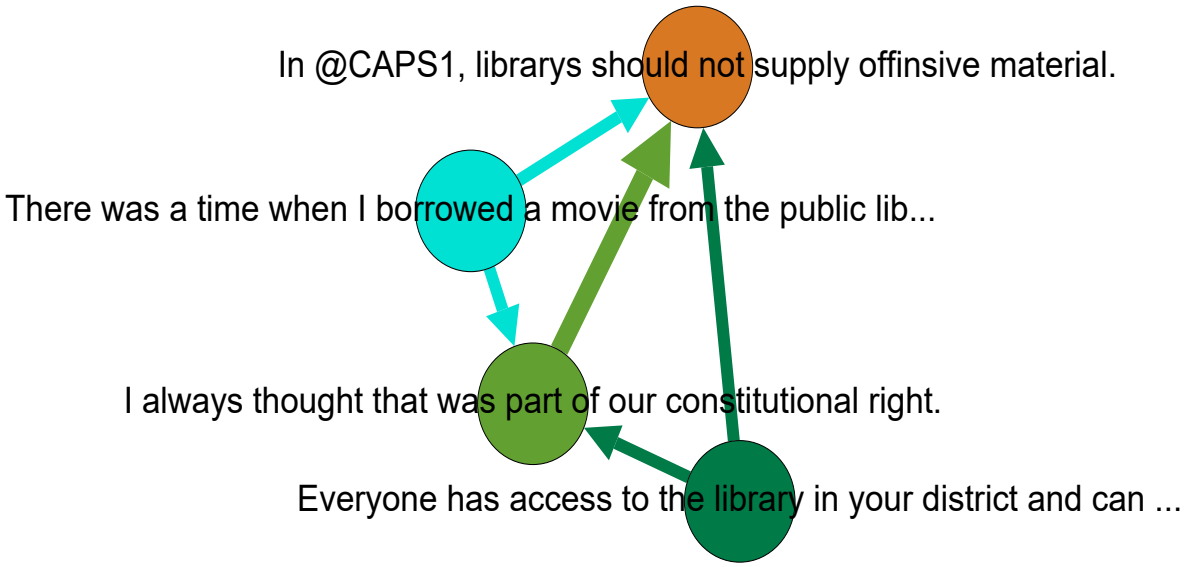

Figura 3.14.: Grafo colaborativo para 23 ensayos extraídos el corpus Kaggle ASAP Phase 1

cláusulas típicas	Representantes de la cláusula típica
Llamada a la acción	Although some of the racism and cencorship that should be cencored are on shelves, I think it should be in a separete room so only older people can check it out or buy it, instead of the young kids getting a hold of it.
Llamada a la acción	At the @ORGANIZATION1 leave again I think they should be, kids that age shouldn't be reading @ORGANIZATION2.
Llamada a la acción	I think they could put stuff in there about drug or alcohol perhaps but not go in depth, but keep it at a bare minimum.
Llamada a la acción	There are some magazines and movies I feel shouldn't be in librarys, and those are as follows: pornography, non-educational movies, @CAPS6 @CAPS7, and social networks.
Llamada a la acción	All libraries should have the right to have the content they please to have on thier shelves.
Llamada a la acción	I believe that all the uncensored books, movies, music, and magazines in a library should be taken off the shelves and put into a single room in the library and be labeled as the @CAPS1 @CAPS2.
Llamada a la acción	I personally dont think there should be any censorship in books coming into a @CAPS1.
Llamada a la acción	Should I be allowed to remove it from shelves, leaving teenagers without their music, denying artists' freedom of speech, and ultimately, leaving them unemployed?
Llamada a la acción	For example, a person must be the age of seventeen or older to rent a '@CAPS3' rated movie.
Llamada a la acción	At most it should be placed in a less accessable manner.
Opinión	People have different oppions, of whats good and whats bad.
Opinión	@CAPS1 are some books, music, movies, and magazines out @CAPS1 that are offensive.
Opinión	Yet we still want to read, listen, watch, and look at them.
Opinión	Why should we have to remove a book if just some people think its offensive?
Opinión	Ask yourself the question again, '@CAPS2 do you know it's a bad book when you haven't even given it a chance?'
Opinión	@CAPS3't judge a book by what you hear.
Opinión	I believe that certain materials such as books, music,movies,ect.
Opinión	Now lets just think about this for a moment.What if a sexist book fell into the hands of an inocent childs hands that could just read enough to get through the book?
Opinión	That child could end up living that kind of life that the book protays.
Testimonio	We hated it when they had book burnings and destroyed the custums, art, and music that the government felt was wrong for their people.
Testimonio	@PERSON1, an author, once said; 'All of use can think of a book that we hope none of our children or any other children have taken off the shelf.
Testimonio	I have read and seen a lot of books in my life time.
Testimonio	And then we have no books left on the shelf for any of us.'
Testimonio	I think that when Katherine Paterson said, 'Children or any other children have taken off the shelf' that she was making a big mistake.
Testimonio	I once was watching a movie with my two year old that we got from the library.
Testimonio	I went to my friends houses and listed to explicit music and watched rated @CAPS1 movies.
Testimonio	I also have had a few bad reading experiences.
Testimonio	More than half the time, they heard that language from their parents.
Testimonio	I don't drink or do drugs, because my parents were there if I had a question about something I didn't understand in a book or media.
Hecho	Music, magazines, books, and movies.
Hecho	There are also things that don't say sex directly, but imply it.
Hecho	Sometimes when parents try to pull their kids closer they actually push them away.
Hecho	Public libaries, and the funding they require, are not anywhere within the @ORGANIZATION1.
Hecho	In libraries there are book that have alot of racism in them.
Hecho	Today, it is almost a requirement for cities to have public libraries to be considered as real cities.
Hecho	Yes there are some youth that take somethings too seriously but thats because the parents are not there to explain.
Hecho	People here have freedom of speech and desirve the right to not have there work censored because a small group finds it offensive.
Hecho	Today libraries are not focused on the materials that they have on their shelves.
Hecho	Even in our country today, the government applies this which in some cases violates the constitution because people have the freedom of speech, but to a certain extent only.

Cuadro 3.15.: Representantes de las cláusulas típicas mostradas el el grafo colaborativo para la muestra de 61 ensayos.

cláusulas típicas	Representantes de la cláusula típica
Llamada a la acción	How do you act, and what should you do?
Llamada a la acción	If something causes someone else harm we should not do it.
Llamada a la acción	In @CAPS1, librarys should not supply offinsive material.
Llamada a la acción	In my personal opinion no I don't think that any book should be banned from the public library.
Llamada a la acción	Things should not be left around if they are offensivce in any way.
Llamada a la acción	So that's one reason why certain books, music, and magazines should be romved from shelves.
Llamada a la acción	Their are a few reasons why I think censorship shouldn't be allowed in libraries or out in public.
Llamada a la acción	People should not take this censorship ability out of hand.
Llamada a la acción	Should certain books be banned or shouldn't certain books be banned?
Llamada a la acción	Book @CAPS1 @CAPS2 I think about removing certain materials such as books, music, movies, and-or magazines is that it should be your choice.
Opinión	If they are found offensive to the public.
Opinión	I think it offends lots of people.
Opinión	The economy is bad enough the way it is.
Opinión	There is a new generation of people who are being raised at the moment, and they should't have to look back at their childhood and have memories of seeing offensive movies, or reading offensive books and magazines.
Opinión	I think music is what influences a person the most.
Opinión	If they are constantly listening to foul language and what not, that is going to just stay into a persons mind.
Opinión	I know this from experience.
Opinión	I also believe that movies influence a peron lot too.
Opinión	For example, a kid goes to see a movie that has a lot of violence in it.
Opinión	Well after he is done watching the movie he thinks to himself, 'well it must be okay to kill someone because i saw someone kill another person because they got away with it and never got into trouble.'
Hecho	There is something offensive to people in every work that has been published, and the buyer is fully aware of that fact before they go and buy something.
Hecho	Censorship should not be allowed in libraries because it shows a form of disobedience.
Hecho	Everyone has access to the library in your district and can look at anything that is there.
Hecho	Most of the time the child will have an adult with them but sometimes they can pick up something before you can stop them.
Hecho	Most people do not go to libraries because of that.
Hecho	Throught the libaries of the world, there are two general sections, adult, kid, and entertaminet sectioons.
Hecho	Typically when you walk in to a library there are books, magazines, movies, and cds everywhere.
Hecho	Furthermore, children also have access to the public library.
Testimonio	There was a time when I borrowed a movie from the public library.
Testimonio	I have @NUM3 little brothers, and they are @NUM4, @NUM5, @NUM6.
Testimonio	They realae towhere I have been, or where I @MONTH1 be going in my life.
Testimonio	I would also have my own room devoted for music only.
Testimonio	In my personal opinion I never remove anything from a library because I know their are alot of people with many interests.
Testimonio	I am here to tell you today my views and opinions on cenership at the library.
Testimonio	Like I said many people have many intersts in many things and they deservie to enjoy them in the library.

Cuadro 3.16.: Representantes de las cláusulas típicas mostradas el el grafo colaborativo para la muestra de 23 ensayos.

Agradecimientos

Agradezco a mi directora, Ángeles Manjarrés Riesco, por un lado la sugerencia de un tema muy interesante de estudio, así como todo el trabajo de revisión, comentarios e indicaciones detallados, que me han permitido organizar y presentar los resultados de una forma mucho más clara.

Y agradezco a mi familia su apoyo constante durante todo el proceso, sin el cual este trabajo no hubiera sido posible.

A. Apéndice I

A.1. Algunos conceptos y términos utilizados en el estudio

A lo largo de este trabajo se utilizan frecuentemente una serie de términos comunes en diferentes áreas de la Inteligencia Artificial (IA) o la Educación, y para los que conviene matizar su significado en el ámbito de este estudio::

1. Competencia en argumentación es la habilidad de una persona para elaborar argumentos.
    a) Por argumento en un texto nos referimos en general al fragmento de discurso que apoya o ataca una o varias posiciones. Esto incluye las cláusulas argumentativas (frases que expresan apoyos o ataques a ideas y otros recursos), junto con las relaciones entre estas cláusulas. En la sección 1.3.2.1, se incluyen definiciones más completas.
2. Características de la argumentación son propiedades que describen en detalle la competencia en argumentación de una persona. Por ejemplo, precisión en la expresión, capacidad de crear argumentos que muestren una inferencia lógica clara.
3. Un indicador es una métrica de una característica. Por ejemplo, alta precisión en la expresión.
4. Atributos o rasgos de un texto argumentativo son una representación de un indicador. En este trabajo son simples valores numéricos, pero se podrían considerar categorías discretas o estructuras más complejas como grafos.

El aprendizaje automático es un área de la IA que estudia como un sistema computacional puede aprender. Siguiendo a [59], una definición de aprendizaje apropiada en IA es la de un proceso de mejora del rendimiento en un medio ambiente a partir de la adquisición de conocimiento basado en la experiencia en este medio ambiente. Uno de los puntos claves es el de la representación de la experiencia y del conocimiento. En el ámbito de este estudio, un modelo es una representación precisa de este medio ambiente para la que se puede definir un rendimiento y extraer conocimiento inteligible a los seres humanos o útil en otras tareas. Un modelo típicamente relaciona atributos observados y variables y se basa en una representación lógica, probabilista o de otro tipo (red neuronal, por ejemplo).

Entrenamiento es un término alternativo para el aprendizaje de un modelo a partir de experiencia. Un algoritmo es un proceso bien definido para completar una tarea, que en este ámbito suele ser aprendizaje de modelos o extracción de información a partir de un modelo entrenado.

Dentro de este marco, se puede considerar que el modelo bayesiano de indicadores de la competencia en argumentación entrenado sobre ensayos anotados y el grafo argumentativo colaborativo son una representación artificial del conocimiento. Sin embargo, todos los atributos considerados (salvo las anotaciones por parte de evaluadores humanos) se obtienen a partir de otros modelos (por ejemplo los modelos de PLN utilizados para toquenización, extracción de PoS, modelos de coherencia y estilo) que representan conocimiento aprendido en otro medio ambiente (o corpus en nuestro caso). En nuestro caso la experiencia se representa en texto natural y los atributos se obtienen a partir de conocimiento artificial previo.

A.2. Revisión detallada de sistemas educativos para el desarrollo de competencia en argumentación según su funcionalidad

En este apéndice damos un listado detallado de sistemas educativos, organizados según su funcionalidad. Esta sección está pensada como información complementaria a la revisión del estado del arte 1.3.4.

A.2.1. Representaciones gráficas del discurso argumentativo

1. **CATO** ([2]) es un entorno de aprendizaje para estudiantes de Derecho centrado en validación de teorías sobre un conjunto de casos prácticos y en la creación de argumentos para clasificar un caso en base a precedentes. Las principales funciones del sistema son:

   a) Proporciona acceso a consultar una base de datos de casos y genera también casos relevantes a las cuestiones de interés del estudiante de forma dinámica. Uno de los puntos más interesantes es que el sistema aplica Razonamiento Basado en Casos (una variación del aprendizaje automático basado en instancias, [72] pg 231) para definir similaridades parciales entre casos utiliza conocimiento previo, organizado de forma jerárquica.

   b) Genera diagramas de estructura de argumentos mostrando la relación de estos con el uso de los casos generados.

2. **Belvédère** ([114]): Concebido inicialmente como un sistema gráfico que utilizaría diagramas para capturar la interacción del estudiante con un sistema

de enseñanza automático, terminó utilizando los diagramas como recurso (estímulo y guía) para representar las relaciones entre datos e hipótesis en un entorno de aprendizaje colaborativo de materias científicas

3. **Convince Me** ([96]): Es un sistema que:
   a) Permite la construcción y revisión de argumentos, incluyendo una interfaz gráfica para visualizar un diagrama de hipótesis y evidencia.
   b) Proporciona una evaluación automática y sin supervisión de la coherencia de los argumentos utilizando un modelo conexionista (ECHO).
      1) En este modelo, evidencias e hipótesis se organizan en un grafo. Cada nodo tiene un valor de activación de partida que representa nuestra creencia de que la evidencia o hipótesis es aceptable.
      2) Los enlaces entre nodos son simétricos e incluyen tres parámetros (excitación entre hipótesis que solapan, inhibición entre hipótesis que no solapan, y excitación de datos para indicar evidencia).
      3) Los valores de activación se actualizan en un proceso recursivo para que la relación entre nodos vecinos sea coherente.
   c) Permite a los usuarios modificar o añadir hipótesis y evidencia tras recibir una evaluación automática de su coherencia. Este ciclo de revisiones permite una mejora de la calidad en la construcción de argumentos versus estudiantes que simplemente exponían sus argumentos por escrito.
4. **Reason!Able** ([125]): Es un sistema para construir y modificar mapas de argumentos manualmente, junto con un sistema de evaluación de la argumentación.
   a) Soporta tres tipos de evaluaciones, que el estudiante tiene que parametrizar en el sistema.
      1) Fuerza de los argumentos a favor y en contra.
      2) Grado de confianza en la certeza de las afirmaciones.
      3) Bases (independientes de los argumentos presentados) para aceptar una afirmación como verdadera (por ejemplo, conocimiento común, opinión de experto).
   b) Tiene un sistema de guía durante la construcción y evaluación de argumentos (lo que supone una ayuda al andamiaje del proceso de enseñanza).
   c) Se ha utilizado en la enseñanza de competencia en argumentación como un sistema, con un andamiaje muy marcado, para mejorar la calidad de los argumentos del estudiante y su capacidad de evaluar los de los demás estudiantes.
5. **LARGO** ([92]): Es un modelo de enseñanza de evaluación de discurso argumentativo legal basado en el filtrado colaborativo («collaborative filtering»):

a) Se centra en identificar tres tipos de debilidades del argumento: Contextual, estructural y de contenido. Evita utilizar técnicas de PLN para detectar estas debilidades debido a la complejidad de la tarea y su alto grado de error.

b) Los estudiantes crean pruebas para validar para las argumentos.

c) Se pide que cada estudiante evalúe una muestra de otras respuestas e indique cuáles son las más similares a la suya. En un segundo paso, selecciona de entre un conjunto de respuestas tipo (buenas y malas) más respuestas de sus compañeros, a las que considera al menos tan buenas como la suya.

d) Con estos resultados se calcula una métrica que se utiliza para seleccionar los mejores resultados.

6. **Carneades**, del que se han presentado más detalles en la introducción de este trabajo 1.3.4.2.
7. **Zeno** ([44]), un modelo formal de argumentación, que incluye una sintaxis para grafos de argumentos y una semántica de etiquetado que soporta un tipo de inferencia. Se utilizó para dar soporte a foros de discusión públicos en el proyecto GeoMed.
8. Entorno DUNES: Sistema **Digalo**.

   a) Los argumentos se representan en un diagrama, soportado por una ontología propia que especifica etiquetas y roles a jugar durante la sesión de aprendizaje **Argunaut** es una extensión que permite la participación de un moderador.

   b) Adicionalmente, el entorno **DUNES** ofrecía **OASIS** (un portal y sistema de gestión del aprendizaje) y **PASEO**, un sistema de comunicación ([17]) No se ha encontrado un enlace activo a **Digalo**, aunque hay evidencia de su uso reciente ([105]).
9. **DREW** y **DREWLITE** ([27]) es una plataforma gráfica con sistema de mensajería en la que los estudiantes, de forma colaborativa van construyendo una representación de una tarea a realizar y argumentan sobre ella.

   a) Tal como se indica en [79], se realizaron experimentos en los que se evaluó la calidad de la representación de conocimiento del estudiante utilizando la taxonomía **SOLO** (una representación jerárquica de los resultados del aprendizaje) y una representación de la calidad del resultado basado en relevancia, corrección, amplitud, profundidad justificación y razonamiento en el discurso.
10. Learning to Argue: Generalized Support Across Domains (LASAD, LASAD-alt) es un proyecto para crear una plataforma de soporte a la enseñanza de la argumentación: una ontología de objetos relacionados con la argumentación

y una serie de componentes gráficos, analíticos y pedagógicos que se pueden combinar para crear aplicaciones para un dominio específico (por ejemplo, argumentación legal o científica).

11. **Online Visualization of Argument** (OVA), es una herramienta web que permite construir mapas de argumentos siguiendo la ontología Argument Interchange Format, AIF ([95]). Se puede considerar una evolución de **ARAUCARIA**, una aplicación de escritorio desarrollada en la Universidad de Dundee (Center for Argument Technology).
    a) Un recurso complementario es **AIFdb**, un buscador web de argumentos publicado por el Center for Argument Technology de la Universidad de Dundee que permite acceder y visualizar a diferentes corpus de diagramas de argumentos, implementados como AIF u otras ontologías.

A.2.2. Representación como tablas o texto del discurso argumentativo

En general son sistemas técnicamente sencillos, y, al igual que las herramientas gráficas, se han utilizado con algún tipo de herramienta de comunicación para fomentar la discusión y colaboración entre estudiantes (por ejemplo, mensajería instantánea).

Algunos ejemplos:

1. Plantillas compartidas que ayudan a organizar los argumentos según un esquema lineal. Por ejemplo, **TC3**, utilizada en el estudio descrito en [36].
2. Representación de hechos e hipótesis, junto con su relaciones (evidencia o inferencia) en una tabla ([115]).
3. **ArguMed** ([127]) proporciona una serie de plantillas para que los usuarios creen un discurso argumentativo escrito a partir de tres tipos de pasos básicos (hacer una afirmación, añadir una razón y su conclusión e indicar una excepción que bloquea la conexión entre razón y conclusión). El sistema utiliza una versión del modelo CumulA de argumentación ([127]). El sistema está orientado hacia la argumentación legal.

A.2.3. Colaboración guiada por software/»Micro-scripting»

Estos sistemas proporcionan una guía paso a paso de la secuencia de actividades en una discusión. En general, el «micro-scripting» se implementa a partir de líneas de comando (“prompts”), comienzos de frases o raíces de preguntas:

1. **C-CHENE** fue un sistema utilizado para que estudiantes colaboraran en resolver problemas de modelado en Física, que manejaba las interacciones entre estudiantes con una interfase gráfica muy sencilla: Un panel para cada actividad básica y botones para indicar una actividad o nuevo argumento ([7]).

2. Otros sistemas anteriores al año 2000 que dirigen el discurso con un guión ([113]): **Scripted Cooperation**, **Ask to Think/Tel-Why, Structured Academic Controversy, Reciprocal Teaching.**

3. **AcademicTalk** ([68]) es un sistema tipo "chat" (basado en Internet Relay Chat, IRC) con funcionalidades para utilizar frases de apertura estándar en cada entrada y la posibilidad de visualizar entradas sobre el mismo tópico en la discusión (en contraste con un foro estándar).

4. Foros de discusión en Microsoft Office SharePoint adaptados para seguir un guión de argumentación ([81]).

5. La referencia [132] utiliza el sistema Accountable Talk ClassRoom Facilitation (una serie de pasos o tácticas para mejorar la implicación del estudiante, [122]) para mejorar la eficacia de «micro-scripting» en el campo de los cursos «online» masivos (MOOC).

6. En la referencia [37] los autores proponen un marco teórico para los sistemas de aprendizaje colaborativo guiados.

7. **Rashi** ([136], Rashi) es un entorno de investigación para estudiantes, específico para algunos dominios científicos, en el que se plantea un problema, se pide a los estudiantes que hagan observaciones, propongan hipótesis y las contrasten con las hipótesis. Para ello el entorno proporciona varias herramientas: Cuaderno de Investigación, Editor de Hipótesis, Aula de Examen y Herramienta de Entrevistas o Guía de Campo (dependiendo del dominio). La plataforma permite ejecutar simulaciones y hace un seguimiento detallado de hipótesis y argumentos a favor utilizados.

A.2.4. Guión de actividades de aprendizaje/»macro-scripting»

Ejemplos de sistemas que implementan o apoyan al "macro-scripting":

1. **ArgueGraph** ([33]):

   a) En una primera fase los estudiantes responden a un cuestionario online para recoger diferentes puntos de vista. El sistema produce un mapa de opiniones (representación de cada opinión como un punto en dos dimensiones. Cada dimensión indica una progresión entre dos posiciones considerada relevante. Por ejemplo descubrimiento/enseñado, sistema/estudiante).

   b) En una segunda fase, estudiantes con respuestas contrapuestas forman parejas y responden de nuevo al cuestionario. El sistema da acceso a las respuestas de cada uno En una tercera fase el grupo, con ayuda del instructor, reformula, estructura y sintetiza los argumentos y justificaciones utilizados por los estudiantes.

c) En una última fase cada estudiante resume los argumentos de una pregunta del cuestionario original utilizando los conceptos discutidos en la tercera fase.

2. **ConceptGrid** ([33]):
   a) Los estudiantes se reparten roles y estudian cada uno un artículo (escogido para su rol).
   b) El grupo escoge los conceptos a definir y los reparte entre los estudiantes para que escriban una definición corta El grupo utiliza **ConceptGrid** para construir una tabla de conceptos en la que conceptos vecinos se pueden relacionar con muy pocas frases.
   c) El grupo sintetiza y estructura en un marco teórico los conceptos y relaciones entre ellos.
      1) **ConceptGrid** utiliza el patrón JIGSAW, en el que el conocimiento se distribuye entre miembros del grupo de manera que cada estudiante depende del otro para completar con éxito la actividad ([4]).
3. **WiSim** ([33]) es un aplicación móvil para dar soporte a actividades de aprendizaje colaborativo aparentemente en dominios muy específicos (Física).
   a) Tras una introducción al fenómeno a simular, los estudiantes forman diferentes grupos, negocian los parámetros de simulación a utilizar y utilizan WiSim en su móvil para enviar estos parámetros.
   b) Cada estudiante recibe una representación diferente de los resultados de la simulación.
   c) El grupo compara cada resultado. Tras varias iteraciones, el grupo sintetiza los resultados de la simulación dentro de un marco teórico.
4. **UniverSanté** ([12]) fue un proyecto de CSCL orientado a la enseñanza de sanidad comunitaria en países muy diferentes (Suiza, Túnez, Camerún, Líbano), utilizando una aproximación centrada en fases y roles y buscando contrastar los diferentes puntos de vista locales para estimular la dinámica social.
   a) Utilizaban una herramienta gráfica para regular la interacción entre estudiantes e instructores, que no estaban en diferentes localizaciones. Esta herramienta era básicamente estática, con un panel para mostrar datos y hechos, y un segundo panel con información sobre estudiantes, instructores y un foro de colaboración.
5. Se han publicado diferentes sistemas para generar guiones ("scripts") adaptados a otros dominios de conocimiento (**CeLS** y **LAMS**, [33]). **ArgueGraph** y **ConceptGrid** están integrados en una plataforma que permite generar nuevos guiones.
6. En la referencia [24] utilizan el sistema **WISE** (Web-based Integrated Science Environment) que integra simuladores y paneles de discusión para guiar a

un grupo de estudio en una actividad de aprendizaje de Física. Los estudiantes ejecutan experimentos virtuales y escriben argumentos estructurados para explicar los resultados utilizando una plantilla. Estos argumentos se utilizan como semillas de una discusión online, en la que se orienta a los estudiantes para mejorar la calidad de sus argumentos.

A.2.5. Juegos de diálogo digitales

Ejemplos de juegos de diálogo basados en software:

1. **DIALAB** (referencia original [91], revisión en [32]) es un entorno en el que los estudiantes pueden practicar un diálogo basado en sentencias primitivas sencillas. La disponibilidad de las primitivas está controlada por un modelo de diálogo.

2. **CLARISSA** ([21]) es un sistema que simula un diálogo, considerado como una máquina de estados. El sistema consiste en un módulo de diálogo y un módulo cognitivo. El módulo cognitivo genera objetivos en paralelo a partir de un modelo del dominio, y el sistema de diálogo gestiona estos objetivos como locuciones, con un mecanismo para gestionar el foco del diálogo.

    a) Este sistema se ha utilizado en simulaciones de diálogo, con un modelo de dominio similar al de C-CHENE y con varios agentes funcionando en paralelo como estudiantes y profesores.

3. Computer-Based Lab for Language Games in Education (**CoLLeGE**) ([99]), descrito en la introducción 1.3.4.3.

4. Sistemas de mediación como **AcademicTalk** (mencionado anteriormente como una herramienta de «micro-scripting») e **InterLoc** ([98]) que es un sistema de mensajería tipo IM o chat, en el que se ofrecen locuciones para abrir un mensaje y se estructura la secuencia de locuciones disponibles para orientar al estudiante a seguir un modelo de argumentación definido.

    a) **AcademicTalk**([97]) e **InterLoc** pueden dar soporte a «micro-scripting», pero también se han utilizado como plataforma de juegos de diálogo. Por ejemplo, se asignan roles a los estudiantes (gestor de Aprendizaje, Facilitador y Jugador) y a partir de unas preguntas semillas se desarrolla un diálogo interactivo controlado.

Aunque el uso de jugos de diálogo parece tener potencial, faltan estudios en escenarios reales ([80]). También faltan pautas para diseñar juegos que den buenos resultados en el desarrollo de competencias de argumentación.

A.3. Algunas propiedades de una red bayesiana gaussiana lineal

En una red bayesiana gaussiana paramétrica lineal de n variables aleatorias $\{X_1, ...X_n\}$, la distribución de probabilidad conjunta se puede expresar como

$$P(X_1, ...X_n) = \prod_{i=1}^{n} P(X_i|parents(X_i))|_{X=x} = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp(-\frac{1}{2\sigma_i^2}(x_i - w_{ii} - \sum_{j \in parents(X_i)} w_{ij}x_j\ )^2)$$

Para una variable concreta X_a nos interesa poder marginalizar la distribución anterior. En el caso gaussiano esto se puede hacer de manera analítica. A partir de la definición anterior de la distribución de probabilidad conjunta, se obtiene:

$$P(X_1, ..., X_n)\ |_{X=x} = \frac{1}{\sqrt{2\pi\sigma_a^2}} \exp(-\frac{1}{2\sigma_a^2}(x_a - w_{aa} - \sum_{j \in parents(X_a)} w_{aj}x_j\ )^2) \times$$

$$\prod_{i \in children(X_a)} \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp(-\frac{1}{2\sigma_i^2}(x_i - w_{ii} - w_{ia}x_a - \sum_{\substack{j \in parent(X_i) \\ j \neq a}} w_{ij}x_j\ )^2) \Pi(x_1, ...x_{a-1}, x_{a+1}, ...x_n)$$

donde $\Pi(x_1, ...x_{a-1}, x_{a+1}, ...x_n)$ son los factores sin dependencia de x_a. Los factores que dependen de x_a se pueden reagrupar en un único término gaussiano:

$$P(X_1, ..., X_n)\ |_{X=x} = \Pi(x_1, ...x_{a-1}, x_{a+1}, ...x_n) \frac{1}{\sqrt{2\pi\sigma_a^2}} \prod_{i \in children(X_a)} \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-Ax_a^2 + 2x_aB - C\right) =$$

$$\Pi(x_1, ...x_{a-1}, x_{a+1}, ...x_n) \frac{1}{\sqrt{2\pi\sigma_a^2}} \prod_{i \in children(X_a)} \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-A(x_a - \frac{B}{A})^2 + \frac{B^2}{A} - C\right) =$$

$$\Pi(x_1, ...x_{a-1}, x_{a+1}, ...x_n) \frac{1}{\sqrt{2\pi\sigma_a^2}} \prod_{i \in children(X_a)} \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-A(x_a - \frac{B}{A})^2\right) \exp\left(\frac{B^2}{A} - C\right)$$

Donde hemos definido una constante A, que depende únicamente de los parámetros de la distribución de probabilidad, una función B lineal en los valores de las variables aleatorias, salvo constante aditiva y una función cuadrática C.

$$A = \frac{1}{2\sigma_a^2} + \sum_{i \in children(X_a)} \frac{w_{ia}^2}{2\sigma_i^2}$$

$$B(x_i, ...x_{a-1}, x_{a+1}...x_n) = \frac{1}{2\sigma_a^2}(w_{aa} + \sum_{j \in parents(X_a)} w_{aj}x_j\ ) + \sum_{i \in children(X_a)} \frac{1}{2\sigma_i^2} w_{ia}(x_i - w_{ii} - \sum_{\substack{j \in parent(X_i) \\ j \neq a}} w_{ij}x_j\ )$$

$$C(x_i, ...x_{a-1}, x_{a+1}...x_n) = \frac{1}{2\sigma_a^2}(w_{aa} + \sum_{j \in parents(X_a)} w_{aj}x_j\ )^2 + \sum_{i \in children(X_a)} \frac{1}{2\sigma_i^2}(x_i - w_{ii} - \sum_{\substack{j \in parent(X_i) \\ j \neq a}} w_{ij}x_j\ )^2$$

A partir de aquí es fácil calcular la probabilidad marginal para X_a que resulta ser una distribución normal:

$$P_{marginal}(X_1, ...X_{a-1}, X_{a+1}, ...X_n)\ |_{X=x} = \int_{-\infty}^{\infty} dx_a P(X_1, ..., X)\ |_{X=x} =$$

$$\Pi(x_1, ...x_{a-1}, x_{a+1}, ...x_n) \frac{1}{\sqrt{2\pi\sigma_a^2}} \prod_{i \in children(X_a)} \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(\frac{B^2}{A} - C\right) \int_{-\infty}^{\infty} dx_a \exp\left(-A(x_a - \frac{B}{A})^2\right) =$$

$$\Pi(x_1, ...x_{a-1}, x_{a+1}, ...x_n) \frac{1}{\sqrt{2\pi\sigma_a^2}} \prod_{i \in children(X_a)} \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(\frac{B^2}{A} - C\right) \sqrt{\frac{\pi}{A}}$$

La probabilidad condicionada es otra distribución normal:

$$P(X_a | X_1 = x_1, ...X_{a-1} = x_{a-1}, X_{a+1} = x_{a+1}, ...X_n = x_n) =$$

$$\frac{P(X_1, ...X_{a-1}, X_{a+1}, ...X_n, X_a)\ |_{X=x}}{P_{marginal}(X_1, ...X_{a-1}, X_{a+1}, ...X_n)\ |_{X=x}} =$$

$$\sqrt{\frac{A}{\pi}} \exp\left(-A(x_a - \frac{B}{A})^2\right)$$

A X fija, salvo X_a, el máximo de esta función verosimilitud cumple:

$$\frac{d}{dx_a} P(X_a = x_a | X_1 = x_1, ...X_{a-1} = x_{a-1}, X_{a+1} = x_{a+1}, ...X_n = x_n)\ |_{X_a = x_a} = 0$$

Esto implica que el valor de X_a para el que la verosimilitud es máxima debe cumplir:

$$-2A(x_a - \frac{B}{A})\sqrt{\frac{A}{\pi}} \exp\left(-A(x_a - \frac{B}{A})^2\right) = 0$$

lo que implica:

$$\arg\min P(X_a = x_a | X_1 = x_1, ...X_{a-1} = x_{a-1}, X_{a+1} = x_{a+1}, ...X_n = x_n) = \frac{B}{A}$$

que es una función lineal, excepto constante aditiva, de los valores observados $x_1, ...x_{a-1}, x_{a+1}, ...x_n$.

Si $\mathbb{E}[X_a]_{X=x}$ es el valor esperado de X_a respecto a $P(X_a|X_1 = x_1, ..., X_{a-1} = x_{a-1}, X_{a+1} = x_{a+1}, ..., X_n = x_n)$, se puede calcular como:

$$\mathbb{E}[X_a]_{X=x} = \int_{-\infty}^{\infty} dx_a x_a P(X_a = x_a | X_1 = x_1, ...X_{a-1} = x_{a-1}, X_{a+1} = x_{a+1}, ...X_n = x_n) =$$

$$\sqrt{\frac{A}{\pi}} \int_{-\infty}^{\infty} dx_a x_a \exp\left(-A(x_a - \frac{B}{A})^2\right) =$$

$$-\sqrt{\frac{A}{\pi}} \int_{-\infty}^{\infty} dx_a \left(\frac{B}{A} \exp\left(-A(x_a - \frac{B}{A})^2\right) - \frac{1}{2A}\frac{d}{dx_a} \exp\left(-A(x_a - \frac{B}{A})^2\right)\right) =$$

$$\sqrt{\frac{A}{\pi}} \left(-\frac{1}{2A} \exp\left(-A(x_a - \frac{B}{A})^2\right) |_{-\infty}^{\infty} + \frac{B}{A} \int_{-\infty}^{\infty} dx_a \exp\left(-A(x_a - \frac{B}{A})^2\right)\right) = \frac{B}{A}$$

que de nuevo es una función lineal, salvo constante aditiva, y que coincide, como era de esperar debido a la normalidad de la distribución, con el máximo de la verosimilitud. Por tanto, $\nabla_x \mathbb{E}[X_a]_{X=x}$ es una constante.

A.4. Algunas trazas del proceso de inferencia de la red bayesiana para el modelo bayesiano de indicadores de la competencia en argumentación y métricas completas del entrenamiento del modelo de identificación de paráfrasis

		Variables independientes del modelo
BNE_w_n_value	BNE_sigma_fk_reading_ease_log___	BNE_w_dale_chall_readability_domain1_score
BNE_sigma_n_value_log___	BNE_w_fk_reading_ease_fk_grade_level	BNE_w_dale_chall_readability_n_testimony
BNE_w_f_punct	BNE_w_domain2_score	BNE_w_f_stopwords
BNE_sigma_f_punct_log___	BNE_sigma_domain2_score_log___	BNE_sigma_f_stopwords_log___
BNE_w_fk_grade_level	BNE_w_domain2_score_n_reference	BNE_w_f_stopwords_f_punct
BNE_sigma_fk_grade_level_log___	BNE_w_domain2_score_n_value	BNE_w_f_stopwords_polarity_average
BNE_w_f_numeric_tokens	BNE_w_domain2_score_f_punct	BNE_w_f_stopwords_dale_chall_readability
BNE_sigma_f_numeric_tokens_log___	BNE_w_domain2_score_domain1_score	BNE_w_n_tokens
BNE_w_precision	BNE_w_n_fact	BNE_sigma_n_tokens_log___
BNE_sigma_precision_log___	BNE_sigma_n_fact_log___	BNE_w_n_tokens_n_value
BNE_w_polarity_peak	BNE_w_n_fact_n_links	BNE_w_n_tokens_style
BNE_sigma_polarity_peak_log___	BNE_w_f_oov	BNE_w_n_tokens_f_diff_lemmas
BNE_w_n_testimony	BNE_sigma_f_oov_log___	BNE_w_n_tokens_f_stopwords
BNE_sigma_n_testimony_log___	BNE_w_f_oov_n_fact	BNE_w_n_tokens_domain1_score
BNE_w_n_policy	BNE_w_f_oov_polarity_peak	BNE_w_f_ents
BNE_sigma_n_policy_log___	BNE_w_f_oov_domain1_score	BNE_sigma_f_ents_log___
BNE_w_n_reference	BNE_w_polarity_average	BNE_w_f_ents_f_numeric_tokens
BNE_sigma_n_reference_log___	BNE_sigma_polarity_average_log___	BNE_w_f_ents_fk_reading_ease
BNE_w_n_links	BNE_w_polarity_average_polarity_peak	BNE_w_f_ents_n_tokens
BNE_sigma_n_links_log___	BNE_w_polarity_average_domain2_score	BNE_w_f_ents_f_oov
BNE_w_domain1_score	BNE_w_f_diff_lemmas	BNE_w_f_ents_style
BNE_sigma_domain1_score_log___	BNE_sigma_f_diff_lemmas_log___	BNE_w_coherence
BNE_w_domain1_score_f_punct	BNE_w_f_diff_lemmas_f_oov	BNE_sigma_coherence_log___
BNE_w_style	BNE_w_dale_chall_readability	BNE_w_coherence_n_tokens
BNE_sigma_style_log___	BNE_sigma_dale_chall_readability_log___	BNE_w_coherence_polarity_peak
BNE_w_style_n_testimony	BNE_w_dale_chall_readability_precision	BNE_w_coherence_domain2_score
BNE_w_style_n_value	BNE_w_dale_chall_readability_fk_reading_ease	BNE_w_coherence_f_diff_lemmas
BNE_w_fk_reading_ease	BNE_w_dale_chall_readability_f_diff_lemmas	

Cuadro A.1.: Listado de las 83 variables independientes de la red bayesiana. Las variables tipo BNE_sigma_*variable* corresponden a la desviación estándar de la distribución normal que modela la variable correspondiente. Las variables BNE_w_*variable* y BNE_w_*variable_variable1* se utilizan para construir el valor medio de esta misma distribución, según la fórmula $\mu_{\text{variable}} = \text{BNE\_w\_variable} + \text{BNE\_w\_variable\_variable1} \times \text{variable1}$

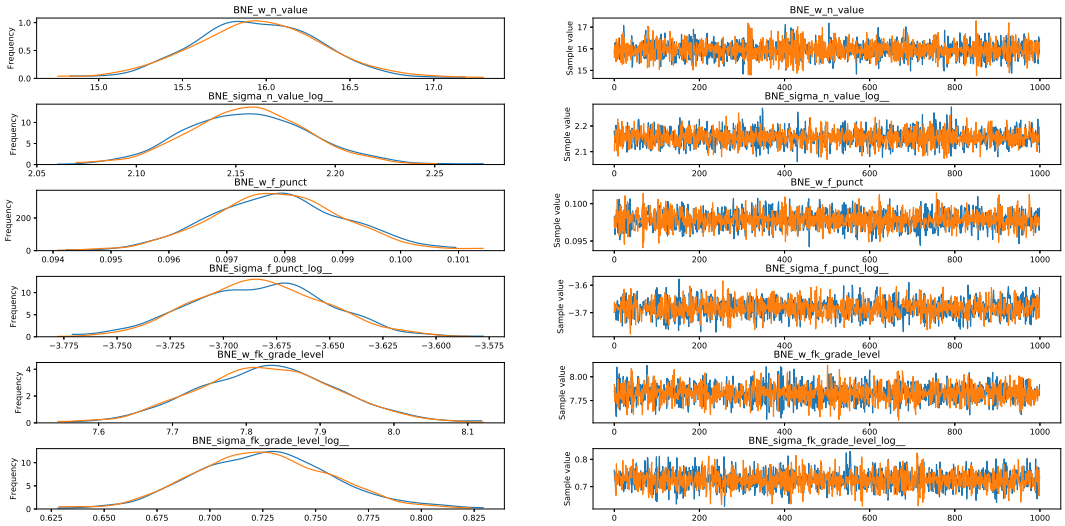

Figura A.1.: Traza del proceso de inferencia para algunas las variables independientes de la red bayesiana. Se muestran por un lado (azul) el muestreo realizado durante la inferencia para cada variable y su distribución de frecuencias, y por otro (naranja) una estimación de muestreo y distribución (usando KDE, "Kernel Density Estimates"). Dos observaciones claves sobre la calidad del proceso de inferencia: Las muestras no deberían concentrarse en un intervalo de valores pequeño de forma apreciable (esto indicaría que la exploración está atrapada en una región pequeña del espacio de parámetros), y la distribución de frecuencias estimada no debería mostrar una multimodalidad clara

Máxima profundidad de árbol	Máximo número de atributos por árbol	Ponderado de clases	Número de estimadores	F1 ponderado entrenamiento	F1 ponderado test
Sin límite	auto	Ninguno	10	0.989602	0.729696
Sin límite	auto	Ninguno	50	0.999492	0.753441
Sin límite	auto	Ninguno	100	1	0.745425
Sin límite	auto	Ninguno	150	1	0.745947
Sin límite	auto	Ninguno	200	1	0.750464
Sin límite	**auto**	**Ninguno**	**250**	**1**	**0.758064**
Sin límite	auto	balanced	10	0.991884	0.724556
Sin límite	auto	balanced	50	0.999492	0.733178
Sin límite	auto	balanced	100	1	0.73928
Sin límite	auto	balanced	150	1	0.740423
Sin límite	auto	balanced	200	1	0.74002
Sin límite	auto	balanced	250	1	0.741671
Sin límite	auto	balanced_subsample	10	0.989598	0.71475
Sin límite	auto	balanced_subsample	50	0.999746	0.74757
Sin límite	auto	balanced_subsample	100	1	0.736724
Sin límite	auto	balanced_subsample	150	1	0.745915
Sin límite	auto	balanced_subsample	200	1	0.739817
Sin límite	auto	balanced_subsample	250	1	0.746801
Sin límite	Todos	Ninguno	10	0.989086	0.726473
Sin límite	Todos	Ninguno	50	0.999746	0.741776
Sin límite	Todos	Ninguno	100	1	0.749406
Sin límite	Todos	Ninguno	150	1	0.744889
Sin límite	Todos	Ninguno	200	1	0.745246
Sin límite	Todos	Ninguno	250	1	0.748854
Sin límite	Todos	balanced	10	0.991618	0.71378
Sin límite	Todos	balanced	50	1	0.736462
Sin límite	Todos	balanced	100	1	0.733916
Sin límite	Todos	balanced	150	1	0.745374
Sin límite	Todos	balanced	200	1	0.743042
Sin límite	Todos	balanced	250	1	0.733633
Sin límite	Todos	balanced_subsample	10	0.987563	0.723876
Sin límite	Todos	balanced_subsample	50	0.999746	0.739631
Sin límite	Todos	balanced_subsample	100	1	0.744446
Sin límite	Todos	balanced_subsample	150	1	0.745026
Sin límite	Todos	balanced_subsample	200	1	0.732469
Sin límite	Todos	balanced_subsample	250	1	0.740423
5	auto	Ninguno	10	0.772887	0.736175
5	auto	Ninguno	50	0.769742	0.74076
5	auto	Ninguno	100	0.774593	0.742836
5	auto	Ninguno	150	0.772617	0.741761
5	auto	Ninguno	200	0.770949	0.743653
5	auto	Ninguno	250	0.773299	0.743247
5	auto	balanced	10	0.74168	0.700886
5	auto	balanced	50	0.751783	0.709256
5	auto	balanced	100	0.750491	0.702155
5	auto	balanced	150	0.754101	0.70862
5	auto	balanced	200	0.752459	0.706883
5	auto	balanced	250	0.752275	0.705119
5	auto	balanced_subsample	10	0.746614	0.701564
5	auto	balanced_subsample	50	0.756822	0.715173
5	auto	balanced_subsample	100	0.755855	0.706311
5	auto	balanced_subsample	150	0.757798	0.707472
5	auto	balanced_subsample	200	0.75588	0.709853
5	auto	balanced_subsample	250	0.752209	0.707487
5	Todos	Ninguno	10	0.777251	0.745868
5	Todos	Ninguno	50	0.784023	0.751324
5	Todos	Ninguno	100	0.778316	0.745196
5	Todos	Ninguno	150	0.779376	0.744657
5	Todos	Ninguno	200	0.783192	0.751004
5	Todos	Ninguno	250	0.782574	0.74979
5	Todos	balanced	10	0.761662	0.710638
5	Todos	balanced	50	0.765041	0.713977
5	Todos	balanced	100	0.77124	0.720874
5	Todos	balanced	150	0.768333	0.71982
5	Todos	balanced	200	0.770293	0.719184
5	Todos	balanced	250	0.774223	0.718616
5	Todos	balanced_subsample	10	0.770999	0.714374
5	Todos	balanced_subsample	50	0.765958	0.715729
5	Todos	balanced_subsample	100	0.769621	0.71677
5	Todos	balanced_subsample	150	0.770321	0.718633
5	Todos	balanced_subsample	200	0.77007	0.716259
5	Todos	balanced_subsample	250	0.770804	0.718581

Cuadro A.2.: Primera parte de los resultados de validación del bosque aleatorio clasificador de paráfrasis sobre los datos de validación. Resaltamos los resultados para el modelo elegido. Estimadores es el número de árboles de decisión en el bosque. El ponderado de clases «balanced» y «balanced subsample» dan más importancia en el entrenamiento a instancias minoritarias.

Máxima profundidad de árbol	Máximo número de atributos por árbol	Ponderado de clases	Número de estimadores	F1 ponderado entrenamiento	F1 ponderado test
10	auto	Ninguno	10	0.909648	0.740377
10	auto	Ninguno	50	0.945209	0.740558
10	auto	Ninguno	100	0.944653	0.744295
10	auto	Ninguno	150	0.949475	0.752657
10	auto	Ninguno	200	0.944026	0.753203
10	auto	Ninguno	250	0.949223	0.753013
10	auto	balanced	10	0.905167	0.731605
10	auto	balanced	50	0.920582	0.748431
10	auto	balanced	100	0.92732	0.751794
10	auto	balanced	150	0.929239	0.751414
10	auto	balanced	200	0.930494	0.748907
10	auto	balanced	250	0.930259	0.751141
10	auto	balanced_subsample	10	0.913943	0.736997
10	auto	balanced_subsample	50	0.927525	0.74439
10	auto	balanced_subsample	100	0.927249	0.747138
10	auto	balanced_subsample	150	0.926759	0.750489
10	auto	balanced_subsample	200	0.929974	0.757844
10	auto	balanced_subsample	250	0.930769	0.75347
10	Todos	Ninguno	10	0.927645	0.733365
10	Todos	Ninguno	50	0.952627	0.750191
10	Todos	Ninguno	100	0.959119	0.74738
10	Todos	Ninguno	150	0.954229	0.748111
10	Todos	Ninguno	200	0.957073	0.749216
10	Todos	Ninguno	250	0.958353	0.748076
10	Todos	balanced	10	0.909556	0.723524
10	Todos	balanced	50	0.931494	0.742306
10	Todos	balanced	100	0.937941	0.758007
10	Todos	balanced	150	0.938995	0.750395
10	Todos	balanced	200	0.935958	0.75283
10	Todos	balanced	250	0.93946	0.754604
10	Todos	balanced_subsample	10	0.91438	0.736903
10	Todos	balanced_subsample	50	0.938231	0.739843
10	Todos	balanced_subsample	100	0.934541	0.749546
10	Todos	balanced_subsample	150	0.934249	0.753491
10	Todos	balanced_subsample	200	0.936949	0.745847
10	Todos	balanced_subsample	250	0.936049	0.752483
20	auto	Ninguno	10	0.98859	0.728512
20	auto	Ninguno	50	1	0.749377
20	auto	Ninguno	100	1	0.747401
20	auto	Ninguno	150	1	0.751577
20	auto	Ninguno	200	1	0.754291
20	auto	Ninguno	250	1	0.755927
20	auto	balanced	10	0.987313	0.739955
20	auto	balanced	50	1	0.738002
20	auto	balanced	100	1	0.734785
20	auto	balanced	150	1	0.737706
20	auto	balanced	200	1	0.746801
20	auto	balanced	250	1	0.737796
20	auto	balanced_subsample	10	0.991119	0.73065
20	auto	balanced_subsample	50	0.999492	0.726392
20	auto	balanced_subsample	100	1	0.731075
20	auto	balanced_subsample	150	1	0.740558
20	auto	balanced_subsample	200	1	0.74076
20	auto	balanced_subsample	250	1	0.736596
20	Todos	Ninguno	10	0.988081	0.716344
20	Todos	Ninguno	50	0.998984	0.742318
20	Todos	Ninguno	100	1	0.755625
20	Todos	Ninguno	150	1	0.750431
20	Todos	Ninguno	200	1	0.741211
20	Todos	Ninguno	250	1	0.748678
20	Todos	balanced	10	0.990352	0.716804
20	Todos	balanced	50	0.999492	0.734988
20	Todos	balanced	100	0.999746	0.740622
20	Todos	balanced	150	1	0.748617
20	Todos	balanced	200	1	0.744052
20	Todos	balanced	250	1	0.743451
20	Todos	balanced_subsample	10	0.988829	0.713193
20	Todos	balanced_subsample	50	0.999492	0.737126
20	Todos	balanced_subsample	100	1	0.74524
20	Todos	balanced_subsample	150	1	0.739942
20	Todos	balanced_subsample	200	1	0.743513
20	Todos	balanced_subsample	250	1	0.735321

Cuadro A.3.: Segunda parte de los resultados de la validación del bosque aleatorio clasificador de paráfrasis sobre los datos de validación. Resaltamos los resultados para el modelo elegido. Estimadores es el número de árboles de decisión en el bosque. El ponderado de clases «balanced» y «balanced subsample» dan más importancia en el entrenamiento a instancias minoritarias.

Máxima profundidad de árbol	Máximo número de atributos por árbol	Ponderado de clases	Número de estimadores	Precisión clase mayoritaria	Exhaustividad clase mayoritaria	Precisión clase minoritaria	Exhaustividad clase minoritaria
Sin límite	auto	Ninguno	10	0.798701	0.79136	0.595365	0.606171
Sin límite	auto	Ninguno	50	0.792262	0.865809	0.675556	0.551724
Sin límite	auto	Ninguno	100	0.782072	0.874081	0.676123	0.519056
Sin límite	auto	Ninguno	150	0.781046	0.878676	0.681928	0.513612
Sin límite	auto	Ninguno	200	0.787854	0.870404	0.677346	0.537205
Sin límite	**auto**	**Ninguno**	**250**	**0.791563**	**0.879596**	**0.695349**	**0.54265**
Sin límite	auto	balanced	10	0.785714	0.808824	0.599229	0.564428
Sin límite	auto	balanced	50	0.77173	0.873162	0.661765	0.490018
Sin límite	auto	balanced	100	0.774194	0.882353	0.679198	0.491833
Sin límite	auto	balanced	150	0.775709	0.880515	0.678218	0.497278
Sin límite	auto	balanced	200	0.774818	0.882353	0.68	0.493648
Sin límite	auto	balanced	250	0.773525	0.891544	0.693506	0.484574
Sin límite	auto	balanced_subsample	10	0.780378	0.796875	0.581439	0.557169
Sin límite	auto	balanced_subsample	50	0.781581	0.881434	0.686893	0.513612
Sin límite	auto	balanced_subsample	100	0.772581	0.880515	0.674185	0.488203
Sin límite	auto	balanced_subsample	150	0.779854	0.882353	0.686275	0.508167
Sin límite	auto	balanced_subsample	200	0.774376	0.883272	0.680905	0.491833
Sin límite	auto	balanced_subsample	250	0.779757	0.88511	0.690594	0.506352
Sin límite	Todos	Ninguno	10	0.789189	0.805147	0.599244	0.575318
Sin límite	Todos	Ninguno	50	0.784992	0.855699	0.653422	0.537205
Sin límite	Todos	Ninguno	100	0.785008	0.875919	0.682353	0.526316
Sin límite	Todos	Ninguno	150	0.783113	0.869485	0.670534	0.524501
Sin límite	Todos	Ninguno	200	0.784053	0.867647	0.668966	0.528131
Sin límite	Todos	Ninguno	250	0.78607	0.871324	0.676674	0.53176
Sin límite	Todos	balanced	10	0.778674	0.798713	0.581262	0.551724
Sin límite	Todos	balanced	50	0.773984	0.875	0.667482	0.495463
Sin límite	Todos	balanced	100	0.772358	0.873162	0.662592	0.491833
Sin límite	Todos	balanced	150	0.779675	0.881434	0.684597	0.508167
Sin límite	Todos	balanced	200	0.775461	0.888787	0.691327	0.491833
Sin límite	Todos	balanced	250	0.770783	0.877757	0.6675	0.484574
Sin límite	Todos	balanced_subsample	10	0.788758	0.799632	0.593284	0.577132
Sin límite	Todos	balanced_subsample	50	0.778143	0.870404	0.665877	0.509982
Sin límite	Todos	balanced_subsample	100	0.778589	0.882353	0.684729	0.504537
Sin límite	Todos	balanced_subsample	150	0.779951	0.879596	0.682039	0.509982
Sin límite	Todos	balanced_subsample	200	0.769293	0.879596	0.668354	0.479129
Sin límite	Todos	balanced_subsample	250	0.775709	0.880515	0.678218	0.497278
5	auto	Ninguno	10	0.77541	0.869485	0.661098	0.502722
5	auto	Ninguno	50	0.775444	0.882353	0.680798	0.495463
5	auto	Ninguno	100	0.77502	0.889706	0.692308	0.490018
5	auto	Ninguno	150	0.774659	0.887868	0.688776	0.490018
5	auto	Ninguno	200	0.776793	0.886029	0.688442	0.497278
5	auto	Ninguno	250	0.775904	0.887868	0.690355	0.493648
5	auto	balanced	10	0.841121	0.661765	0.530013	0.753176
5	auto	balanced	50	0.854265	0.662684	0.538365	0.77677
5	auto	balanced	100	0.849642	0.654412	0.530587	0.771325
5	auto	balanced	150	0.849941	0.66636	0.538168	0.767695
5	auto	balanced	200	0.851896	0.660846	0.535849	0.77314
5	auto	balanced	250	0.852205	0.657169	0.53375	0.774955
5	auto	balanced_subsample	10	0.849462	0.653493	0.529925	0.771325
5	auto	balanced_subsample	50	0.858491	0.669118	0.54488	0.782214
5	auto	balanced_subsample	100	0.854241	0.657169	0.534913	0.778584
5	auto	balanced_subsample	150	0.852071	0.661765	0.536524	0.77314
5	auto	balanced_subsample	200	0.855279	0.662684	0.538945	0.778584
5	auto	balanced_subsample	250	0.853746	0.659926	0.536341	0.77677
5	Todos	Ninguno	10	0.778675	0.886029	0.690773	0.502722
5	Todos	Ninguno	50	0.781628	0.891544	0.703518	0.508167
5	Todos	Ninguno	100	0.776179	0.892463	0.698454	0.491833
5	Todos	Ninguno	150	0.776	0.891544	0.696658	0.491833
5	Todos	Ninguno	200	0.778839	0.899816	0.71466	0.495463
5	Todos	Ninguno	250	0.7792	0.895221	0.706941	0.499093
5	Todos	balanced	10	0.83802	0.684743	0.542667	0.738657
5	Todos	balanced	50	0.856471	0.669118	0.543726	0.778584
5	Todos	balanced	100	0.851936	0.6875	0.553219	0.764065
5	Todos	balanced	150	0.856481	0.680147	0.550968	0.774955
5	Todos	balanced	200	0.853855	0.681985	0.550649	0.76951
5	Todos	balanced	250	0.854503	0.680147	0.549806	0.771325
5	Todos	balanced_subsample	10	0.846857	0.681066	0.545812	0.756806
5	Todos	balanced_subsample	50	0.85614	0.672794	0.545918	0.77677
5	Todos	balanced_subsample	100	0.849943	0.681985	0.548303	0.76225
5	Todos	balanced_subsample	150	0.855324	0.679228	0.549677	0.77314
5	Todos	balanced_subsample	200	0.853009	0.67739	0.547097	0.76951
5	Todos	balanced_subsample	250	0.852874	0.681985	0.550065	0.767695

Cuadro A.4.: Primera parte de los resultados de precisión y exhaustividad de cada clase cálculada en los datos de validación. La clase mayoritaria son los pares de frase con paráfrasis

Máxima profundidad de árbol	Máximo número de atributos por árbol	Ponderado de clases	Número de estimadores	Precisión clase mayoritaria	Exhaustividad clase mayoritaria	Precisión clase minoritaria	Exhaustividad clase minoritaria
10	auto	Ninguno	10	0.779983	0.866728	0.662791	0.517241
10	auto	Ninguno	50	0.775	0.883272	0.681704	0.493648
10	auto	Ninguno	100	0.779316	0.879596	0.681265	0.508167
10	auto	Ninguno	150	0.784841	0.88511	0.696602	0.520871
10	auto	Ninguno	200	0.787479	0.878676	0.689412	0.53176
10	auto	Ninguno	250	0.784553	0.886949	0.699267	0.519056
10	auto	balanced	10	0.809021	0.774816	0.589615	0.638838
10	auto	balanced	50	0.818354	0.795037	0.616838	0.651543
10	auto	balanced	100	0.822857	0.794118	0.619694	0.662432
10	auto	balanced	150	0.820416	0.797794	0.621343	0.655172
10	auto	balanced	200	0.821394	0.790441	0.614865	0.660617
10	auto	balanced	250	0.822074	0.794118	0.619048	0.660617
10	auto	balanced_subsample	10	0.806935	0.79136	0.603147	0.626134
10	auto	balanced_subsample	50	0.813146	0.795956	0.61324	0.638838
10	auto	balanced_subsample	100	0.820268	0.788603	0.612142	0.658802
10	auto	balanced_subsample	150	0.821293	0.794118	0.618399	0.658802
10	auto	balanced_subsample	200	0.824083	0.805147	0.631944	0.660617
10	auto	balanced_subsample	250	0.823362	0.796875	0.622867	0.662432
10	Todos	Ninguno	10	0.778806	0.851103	0.64	0.522686
10	Todos	Ninguno	50	0.780096	0.893382	0.704835	0.502722
10	Todos	Ninguno	100	0.781123	0.882353	0.687805	0.511797
10	Todos	Ninguno	150	0.781759	0.882353	0.688564	0.513612
10	Todos	Ninguno	200	0.78332	0.880515	0.6875	0.519056
10	Todos	Ninguno	250	0.780567	0.886029	0.693069	0.508167
10	Todos	balanced	10	0.798677	0.776654	0.581756	0.61343
10	Todos	balanced	50	0.81028	0.796875	0.611599	0.631579
10	Todos	balanced	100	0.821828	0.809743	0.634921	0.653358
10	Todos	balanced	150	0.814471	0.806985	0.625668	0.637024
10	Todos	balanced	200	0.816327	0.808824	0.629234	0.640653
10	Todos	balanced	250	0.817424	0.810662	0.632143	0.642468
10	Todos	balanced_subsample	10	0.81244	0.780331	0.597643	0.644283
10	Todos	balanced_subsample	50	0.811143	0.789522	0.605172	0.637024
10	Todos	balanced_subsample	100	0.818697	0.796875	0.618966	0.651543
10	Todos	balanced_subsample	150	0.817084	0.808824	0.629893	0.642468
10	Todos	balanced_subsample	200	0.8125	0.800551	0.617284	0.635209
10	Todos	balanced_subsample	250	0.817335	0.806066	0.627208	0.644283
20	auto	Ninguno	10	0.791328	0.805147	0.601504	0.580762
20	auto	Ninguno	50	0.7875	0.868566	0.67426	0.537205
20	auto	Ninguno	100	0.786012	0.867647	0.671233	0.533575
20	auto	Ninguno	150	0.785714	0.879596	0.688836	0.526316
20	auto	Ninguno	200	0.787829	0.880515	0.692671	0.53176
20	auto	Ninguno	250	0.787113	0.886949	0.702179	0.526316
20	auto	balanced	10	0.791449	0.83364	0.63286	0.566243
20	auto	balanced	50	0.773387	0.881434	0.676692	0.490018
20	auto	balanced	100	0.772285	0.875919	0.666667	0.490018
20	auto	balanced	150	0.771817	0.886029	0.682051	0.482759
20	auto	balanced	200	0.779757	0.88511	0.690594	0.506352
20	auto	balanced	250	0.772947	0.882353	0.677582	0.488203
20	auto	balanced_subsample	10	0.786467	0.82261	0.61477	0.558984
20	auto	balanced_subsample	50	0.766802	0.870404	0.65099	0.477314
20	auto	balanced_subsample	100	0.767386	0.882353	0.670103	0.471869
20	auto	balanced_subsample	150	0.775	0.883272	0.681704	0.493648
20	auto	balanced_subsample	200	0.775444	0.882353	0.680798	0.495463
20	auto	balanced_subsample	250	0.773279	0.877757	0.670792	0.491833
20	Todos	Ninguno	10	0.779163	0.804228	0.587209	0.549909
20	Todos	Ninguno	50	0.785173	0.856618	0.654867	0.537205
20	Todos	Ninguno	100	0.792959	0.869485	0.681614	0.551724
20	Todos	Ninguno	150	0.789121	0.866728	0.673423	0.54265
20	Todos	Ninguno	200	0.777506	0.876838	0.674757	0.504537
20	Todos	Ninguno	250	0.785596	0.872243	0.677494	0.529946
20	Todos	balanced	10	0.774587	0.818015	0.595918	0.529946
20	Todos	balanced	50	0.772727	0.875	0.665848	0.491833
20	Todos	balanced	100	0.776156	0.879596	0.67734	0.499093
20	Todos	balanced	150	0.780744	0.886949	0.694789	0.508167
20	Todos	balanced	200	0.777688	0.884191	0.686567	0.500907
20	Todos	balanced	250	0.776348	0.886949	0.689394	0.495463
20	Todos	balanced_subsample	10	0.77669	0.80239	0.582524	0.544465
20	Todos	balanced_subsample	50	0.777686	0.86489	0.657343	0.511797
20	Todos	balanced_subsample	100	0.781609	0.875	0.67696	0.517241
20	Todos	balanced_subsample	150	0.773676	0.886029	0.684478	0.488203
20	Todos	balanced_subsample	200	0.777508	0.883272	0.684864	0.500907
20	Todos	balanced_subsample	250	0.77247	0.876838	0.668317	0.490018

Cuadro A.5.: Precisión y exhaustividad de cada clase cálculada cálculada en los datos de validación. La clase mayoritaria son los pares de frase con paráfrasis

Bibliografía

[1] Ehud Aharoni, Anatoly Polnarov, Tamar Lavee, Daniel Hershcovich, Ran Levy, Ruty Rinott, Dan Gutfreund, and Noam Slonim. A benchmark dataset for automatic detection of claims and evidence in the context of controversial topics. In *Proceedings of the First Workshop on Argumentation Mining*, pages 64–68, 2014.

[2] Vincent Aleven. *Teaching Case-Based Argumentation Through a Model and Examples*. PhD thesis, University of Pittsburgh, 1997.

[3] Ion Androutsopoulos and Prodromos Malakasiotis. A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research*, 38, 12 2009.

[4] E. Aronson, N. Blaney, J. Sikes, G. Stephan, and M. Snapp. *The jigsaw classroom*. Beverly Hills, CA: Sage, 1978.

[5] Katie Atkinson, Pietro Baroni, Massimiliano Giacomin, Anthony Hunter, Henry Prakken, Chris Reed, Guillermo Simari, Matthias Thimm, and Serena Villata. Toward artificial argumentation. *AI Magazine*, 38(3):25–36, 9 2017. Copyright 2017, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

[6] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *in Proc. of LREC*, 2010.

[7] Michael Baker and Kristine Lund. Promoting reflective interactions in a computer-supported collaborative learning environmen. *Journal of Computer Assisted Learning*, pages 175–193, 1997.

[8] Regina Barzilay and Mirella Lapata. Modeling local coherence: An entity-based approach. volume 34, 01 2005.

[9] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: An open source software for exploring and manipulating networks, 2009.

[10] Luisa Isabel Rodríguez Bello. El modelo argumentativo de Toulmin en la escritura de artículos de investigación educativa. *Revista Digital Universitaria*, 2004.

[11] Johan van Benthem. A brief history of natural logic. Technical report, ILLC Anmsterdam & Stanford, 2008.

[12] A. Berger, R. Moretti, P. Chastonay, P. Dillenbourg, A. Bchir, R. Baddoura, C. Bengondo, D. Scherly, P. Ndumbe, P. Farah, and B. Kayser. Teaching community health by exploiting international socio-cultural and economical differences. In P. Dillenbourg, A. Eurelings, and K. Hakkarainen, editors, *Proceedings of the first European Conference on Computer Supported Collaborative Learning*, pages 97–105, 2001.

[13] María Bezanilla, Robert Wagenaar, and Julia González Ferreras. *Tuning Educational Structures In Europe. Final Report. Pilot project-Phase 1. Learning outcomes: Competences.* 01 2003.

[14] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly Media, 2009.

[15] Ian A. Blood. Automated Essay Scoring: A literature review.

[16] J Bobadilla, Fernando Ortega, A Hernando, and A Gutiérrez. Recommender systems survey. *Knowledge-Based Systems*, 46:109â132, 07 2013.

[17] Yannis Bouyias, Stavros Demetriadis, and Ioannis Tsoukalas. Scripting argumentation in technology enhanced learning: a proposed system architecture. In *Balkan Conference in Informatics (BCI2007)*, pages 337–348, 2007.

[18] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2015.

[19] Leo Breiman. Random forests. *Machine Learning*, (45):5–32, 2001.

[20] Susan Bull and Barbara Wasson. Competence visualisation: Making sense of data from 21st-century technologies in language learning. *ReCALL : the Journal of EUROCALL*, 28(2):147–165, 05 2016.

[21] M. Burton, P. Brna, and R. Pilkington. Clarissa: a laboratory for the modelling of collaboration. *International Journal of Artificial Intelligence in Education*, 11:79–105, 2000.

[22] Elena Cabrio and Serena Villata. Combining textual entailmentand argumentation theory for supporting online debates interactions. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 208–212. Association for Computational Linguistics, 2012.

[23] Yunuen Ixchel Guzmán Cedillo, Rosa del Carmen Flores-Macías, and Felipe Tirado Segura. Rubrics for argumentative competence assessment in online discussion forums. *Innovación Educativa*, 12:17–40, 12 2012.

[24] D. B. Clark and V. Sampson. Personally-seeded discussions to scaffold online argumentation. *International Journal of Science Education*, 29(3):253–277, 2007.

[25] C. Coffin and O'Halloran K. Researching argumentation in educational contexts: New directions, new methods. *International Journal of Research and Method in Education*, 31(3):219–227, 2008.

[26] Khalid Colchester, Hani Hagras, Daniyal Alghazzawi, and Ghadah Aldabbagh. A survey of artificial intelligence techniques employed for adaptive educational systems within e-learning platforms. *Journal of Artificial Intelligence and Soft Computing Research*, 7(1, pages: 47 - 64), 2017.

[27] A. Corbel, P. Jaillon, X. Serpaggi, M. Baker, M. Quignard, K. Lund, and et al. *EIAH2003 Environnements Informatiques pour l'Apprentissage Humains*, chapter DREW: Un outil internet pour créer situations d

appretissage coopérant, pages 109–113. Paris: INRP, 2002.

[28] Scott A. Crossley, Stephen Skalicky, Mihai Dascalu, Danielle S. McNamara, and Kristopher Kyle. Predicting text comprehension, processing, and familiarity in adult readers: New approaches to readability formulas. *Discourse Processes*, 54(5-6):340 – 359, 2017.

[29] Matthew D. Hoffman and Andrew Gelman. The No-U-Turn Sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15, 11 2011.

[30] Chedia Dhaoui, Cynthia M. Webster, and Lay P. Tan. Social media sentiment analysis: lexicon versus machine learning. *The Journal of Consumer Marketing*, 34(6):480–488, 2017.

[31] Pierre Dillenbourg. *Collaborative-learning: Cognitive and Computational Approaches*, chapter What do you mean by collaborative learning?, pages 1–19. Oxford: Elsevier, 1999.

[32] Pierre Dillenbourg. *Three worlds of CSCL. Can we support CSCL?*, chapter Over-scripting CSCL: The risks of blending collaborative learning with instructional design, pages 61–91. Open Universiteit Nederland, 2002.

[33] Pierre Dillenbourg and Fabrice Hong. The mechanics of CSCL macro scripts. *International Journal of Computer-Supported Collaborative Learning*, 3(1):5–23, 2008.

[34] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, (77):321–357, 1995.

[35] L. Elder and R. Paul. *A Guide for Educators to Critical Thinking Competency Standards*. 2005.

[36] G. Erkens, J. Jaspers, M. Prangsma, and G. Kanselaar. Coordination processes in computer supported collaborative writing. *Computers in Human Behaviour*, 21(3):463–486, 2005.

[37] Frank Fischer, Ingo Kollar, Karsten Stegmann, and Christof Wecker. Toward a script theory of guidance in computer-supported collaborative learning. *Educational Psychologist*, 48(1):56–66, 2013.

[38] W. N. Francis and H. Kucera. The Brown corpus of standard american english.

[39] James B. Freeman. *Dialectics and the macrostructure of arguments: A theory of argument structure.* FORIS PUBLICATIONS, 1991.

[40] James B. Freeman. *Argument Structure: Representation and Theory.* Springer, 2011.

[41] Michael Gamon. Linguistic correlates of style: Authorship classification with deep linguistic analysis features. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 611–617. Association for Computational Linguistics, 01 2004.

[42] Andrew Gelman and Donald B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4):457–511, 1992.

[43] Jason Glynos, David Howarth, Aletta Norval, and Ewen Speed. Discourse analysis: Varieties and methods. Technical report, ESRC National Centre for Research Methods, 2009.

[44] T. F. Gordon and N. Karacapilidis. The Zeno argumentation framework. In *Proceedings of the 6th Intl. Conf. on Artificial Intelligence and Law (ICAIL 1997)*, pages 10–18, 1997.

[45] T. F. Gordon, H. Prakken, and D. Walton. The Carneades model of argument and burden of proof. *Artificial Intelligence*, 171(10-15):875–896, 2007.

[46] Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník, Bas R. Steunebrink, and Jürgen Schmidhuber. LSTM: A search space odyssey. *CoRR*, abs/1503.04069, 2015.

[47] Ivan Habernal, Judith Eckle-Kohler, and Iryna Gurevych. Argumentation mining on the web from information seeking perspective. In Elena Cabrio, Serena Villata, and Adam Wyne, editors, *Proceedings of the Workshop on Frontiers and Connections between Argumentation Theory and Natural Language Processing*, volume 1341, 2014.

[48] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In Gäel Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference (SciPy2008)*, pages 11–15, 2008.

[49] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning.* Springer, second edition edition, 2008.

[50] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.

[51] Matthew Honnibal and Mark Johnson. An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1373–1378, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[52] Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. 2019.

[53] Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. OntoNotes: The 90In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, NAACL-Short '06, pages 57–60, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[54] Nurul Islam, Martin Beer, and Frances Slack. E-learning challenges faced by academics in higher education: A literature review. *Journal of Education and Training Studies*, 3, 07 2015.

[55] Thorsten Joachims. Optimizing search engines using clickthrough data. In *SIGKDD 02*, 2002.

[56] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001-2019.

[57] J. Peter Kincaid, Robert P. Fishburne, Richard L. Rogers, and Brad S. Chissom. Derivation of new readability formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy enlisted personnel, 1975.

[58] Werner Kunz and Horst WJ Rittel. Issues as elements of information systems. institute of urban and regional development. *Institute of Urban and Regional Development*, 131, 1970.

[59] Pat Langley. *Elements of Machine Learning*. Morgan Kaufmann, 1996.

[60] Nguyen-Thinh Le and Niels Pinkwart. Bayesian networks for competence-based student modeling. In K. Seta and T. Watanabe, editors, *Proceedings of the 11th International Conference on Knowledge Management*, 2015.

[61] Qi Li, Tianshi Li, and Baobao Chang. Discourse parsing with attention-based hierarchical neural networks. pages 362–371, 01 2016.

[62] Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. Automatically evaluating text coherence using discourse relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 997–1006, 2011.

[63] M. Lippi and P. Torroni. Argumentation mining: State of the art and emerging trends. *ACM Transactions on Internet Technology*, 16(2):1–25, 2016.

[64] Yang Liu and Mirella Lapata. Learning contextually informed representations for linear-time discourse parsing. pages 1289–1298, 01 2017.

[65] W. C. Mann and S. A. Thompson. Rhetorical structure theory: Towards a functional theory of text organizatio. *TEXT*, 8:243–281, 1988.

[66] Marcus and Mitchell et al. Treebank-3, 1999.

[67] Juan Marichal. *La voluntad de estilo*. Seix Barral, 1957.

[68] S. McAlister, A. Ravenscroft, and E. Scanlon. Combining interaction and context design to support collaborative argumentation using a tool for synchronous cmc. *Journal of Computer Assisted Learning*, 20(3):194â204, 2004.

[69] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

[70] George A. Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41, 1995.

[71] Eleni Miltsakaki, Rashmi Prasad, Aravind Joshi, and Bonnie Webber. The Penn Discourse Treebank. In *Proceedings of the Language and Resourcesand Evaluation Conference*, 04 2004.

[72] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[73] Raquel Mochales and Marie-Francine Moens. Argumentation mining. *Artificial Intelligence and Law*, 19(1):1–22, 2011.

[74] D. Chandra Mohan, Dipankar Das, and Sivaji Bandyopadhyay. Emotion argumentation. In *2015 IEEE 2nd International Conference on Recent Trends in Information Systems (ReTIS)*, 2015.

[75] Andreas C. Müller and Sven Behnke. pystruct - learning structured prediction in Python. *Journal of Machine Learning Research*, 15:2055–2060, 2014.

[76] Tempestt Neal, Kalaivani Sundararajan, Aneez Fatima, Yiming Yan, Yingfei Xiang, and Damon Woodard. Surveying stylometry techniques and applications. *ACM Computing Surveys*, 50(6):1–36, November 2017.

[77] Huy V. Nguyen and Diane J. Litman. Argument mining for improving the automated scoring of persuasive essays. In *AAAI*, 2018.

[78] Vlad Niculae, Joonsuk Park, and Claire Cardie. Argument mining with structured svms and rnns. In *ACL*, 2017.

[79] O. Noroozi, H. J. A. Biemans, M. C. Busstra, M. Mulder, and M. Chizari. Differences in learning processes between successful and less successful students in computer-supported collaborative learningin the field of human nutrition and health. *Computers in Human Behaviour*, 27(1):309–318, 2011.

[80] O. Noroozi and S. McAlister. *Competence-based Vocational and Professional Education*, chapter Software Tools for Scaffolding Argumentation Competence Development, pages 819–839. Springer, 2017.

[81] O. Noroozi, A. Weinberger, H. J. A. Biemans, M. Mulder, and M. Chizari. Facilitating argumentative knowledge construction through a transactive discussion script in CSCL. *Computers and Education*, 61(2):59–76, 2013.

[82] Helena P. Osana and Jennifer R. Seymour. Critical thinking in preservice teachers: A rubric for evaluating argumentation and statistical reasoning. *Educational Research and Evaluation*, 2004.

[83] Raquel Mochales Palau and Aagje Ieven. Creating an argumentation corpus: do theories apply to real arguments? a case study on the legal argumentation of the echr. In *ICAIL-2009*, 2009.

[84] Raquel Mochales Palau and Marie-Francine Moens. Argumentation mining: the detection, classification and structure of arguments in text. In *ICAIL-2009*, 2009.

[85] Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. In *EMNLP*, 2016.

[86] J Park, C. Blake, and C. Cardie. Toward machine-assisted participation in eRulemaking: An argumentation model of evaluability. In *ICAIL15*, 2015.

[87] Braja Patra, Somnath Banerjee, Dipankar Das, Tanik Saikh, and Sivaji Bandyopadhyay. Automatic author profiling based on linguistic and stylistic features. In *Proceedings of CLEF 2013 Evaluation Labs*, 09 2013.

[88] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[89] Andreas Peldszus and Manfred Stede. From argument diagrams to argumentation mining in texts: a survey. *IJCINI*, 7(1):1–31, 2013.

[90] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[91] R.M. Pilkington, J.R. Hartley, D. Hintze, and D. Moore. Learning to argue and arguing to learn: An interface for computer-based dialogue games. *International Journal of Artificial Intelligence in Education*, 3(3):275–285, 1992.

[92] N. Pinkwart, V. Aleven, K. Ashley, and C. Lynch. Toward legal argument instruction with graph grammars and collaboraive filtering techniques. In *Proceedings of the 8th international conference on Intelligent Tutoring Systems (ITS 2006)*, pages 227–236, 2006.

[93] H. S. Pinto, C. Tempich, and S. Staab. Diligent: Towards a fine grained methodology for distributed, loosely controlled and evolving engineering of

ontologies. In *Proceeedings of the 16th European Conference on Artificial Intelligence*, 2004.

[94] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013. ISBN 3-900051-07-0.

[95] Iyadh Rahwan and Chris Reed. *Argumentation in Artificial Intelligence*, chapter The Argument Interchange Format. Springer, 2009.

[96] M. Ranney and P. Schank. *Connectionist models of social reasoning and social behavior*, chapter Toward an integration of the social and the scientific: Observing, modeling, and promoting the explanatory coherence of reasoning, pages 245–274. Mahwah, NJ: Lawrence Erlbaum, 1998.

[97] A. Ravenscroft. Promoting thinking and conceptual change with digital dialogue games. *Journal of Computer Assisted Learning*, 23(6):453–465, 2007.

[98] A. Ravenscroft, S. McAlister, and M. Sagar. *Educational Technologies for Teaching Argumentation Skills*, chapter Digital Dialogue Games and InterLoc: A Deep Leaning Design for Collaborative Argumentation on the Web, pages 1–21. Bentham Science E-Books, 2010.

[99] A. Ravenscroft and R. M. Pilkington. Investigation by design: Developing dialogue models to support reasoning and conceptual change. *International Journal of Artificial Intelligence in Education*, 11(1):273–298, 2000.

[100] C. Reed. *Linguistics in the Twenty First Century*, chapter Preliminary Results from an Argument Corpus, pages 185–196. Cambridge Scholars Press, 2006.

[101] C. Reed and D. Walton. Towards a formal and implemented model of argumentation schemes in agent communication. *Autonomous Agents and Multi-Agents Systems*, (00):1–16, 2005.

[102] Jakob Runge. Conditional independence testing based on a nearest-neighbor estimator of conditional mutual information. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2018.

[103] J. Salvatier, T. V. Wiecki, and C. Fonnesbeck. Probabilistic programming in python using PyMC3. *PeerJ Computer Science*, 2016.

[104] O. Scheuer, B. M. McLaren, F. Loll, and N. Pinkwart. *Educational technologies for teaching argumentation skills*, chapter Automated analysis and feedback techniques to support and teach argumentation: A survey, pages 71–124. Bentham Science, 2012.

[105] B. B. Schwarz and N. Shahar. Combining the dialogic and the dialectic: Putting argumentation into practice in classroom talk. *Learning, Culture and Social Interaction*, 2017.

[106] Vivian Dos Santos Silva, Siegfried Handschuh, and André Freitas. Recognizing and justifying text entailment through distributional navigation on definition graphs. In *AAAI*, 2018.

[107] Vangapelli Sowmya, Bulusu Vishnu Vardhan, and Mantena S.V.S. Bhadri Raju. Improving semantic textual similarity with phrase entity alignment. *International Journal of Intelligent Engineering & Systems*, 10(4):193–204, 2017.

[108] David J. Spiegelhalter, Nicola G. Best, Bradley P. Carlin, and Angelika van der Linde. Bayesian measures of model complexity and fit. *J. R. Statist. Soc. B*, (64):583–639, 2002.

[109] Christian Stab and Iryna Gurevych. Annotating argument components and relations in persuasive essays. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics*, pages 1501–1510, 2014.

[110] Christian Stab and Iryna Gurevych. Parsing argumentation structure in persuasive essays. *arXiv:1604.07370v2*, 2016.

[111] Christian Stab and Iryna Gurevych. Parsing argumentation structures in persuasive essays. *Computational Linguistics*, 43, 04 2016.

[112] Christian Stab and Iryna Gurevych. Cross-topic argument mining from heterogeneous sources using attention-based neural networks. preprint arXiv:1802.05758, 2018, February 2018.

[113] K. Stegmann, A. Weinberger, and F. Fischer. Facilitating argumentative knowledge construction with computer-supported collaboration scripts. *International Journal of Computer- Supported Collaborative Learning*, 2(4):421–447, 2007.

[114] D. Suthers. Towards a systematic study of representational guidance for collaborative learning discourse. *Journal of Universal Computer Science*, 7(3):254–277, 2001.

[115] D. D. Suthers and C. D. Hundhausen. The effects of representation on students' elaborations in collaborative inquiry. In *Proceedings of CSCL 2002*, pages 472–480, 2002.

[116] Yui Suzuki, Tomoyuki Kajiwara, and Mamoru Komachi. Building a non-trivial paraphrase corpus using multiple machine translation systems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics-Student Research Workshop*, pages 36–42, 2017.

[117] Christoph Tempich, H. Sofia Pinto, York Sure, and Steffen Staab. *An Argumentation Ontology for DIstributed, Loosely-controlled and evolvInG Engineering processes of oNTologies (DILIGENT)*, chapter An Argumentation Ontology for DIstributed, Loosely-controlled and evolvInG Engineering processes of oNTologies (DILIGENT). Springer, 2005.

[118] Stephen Edelston Toulmin. *The Uses of Argument*. Cambridge University Press, 1958.

[119] Ioannis Tsamardinos, Laura E. Brown, and Constantin F. Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. In *Machine Learning*, 2006.

[120] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *International Conference on Machine Learning (ICML)*, pages 104–112, 2004.

[121] Princeton University. About WordNet. Online, 2010.

[122] Chiel van der Veen, Femke van der Wilt, Claudia van Kruistum, Bert van Oers, and Sarah Michaels. MODEL2TALK: An intervention to promote productive classroom talk. *The Reading Teacher*, 70(6):689–700, 2017.

[123] Frans H. van Eemeren and Rob Grootendorst. Developments in argumentation theory.

[124] Frans H. van Eemeren and Rob Grootendorst. *A Systematic Theory of Argumentation. The pragma-dialectical approach.* Cambridge University Press, 2004.

[125] T. Van Gelder. Argument mapping with Reason!Able. *The American Philosophical Association Newsletter on Philosophy and Computers*, 2002.

[126] Aki Vehtari, Andrew Gelman, and Jonah Gabry. Practical bayesian model evaluation using leave-one-out cross-validation and waic. *J. Stat Comput*, 2017.

[127] B. Verheij. Argumed: a template argument-mediation system for lawyers. In Gerard Noodt, editor, *Legal Knowledge-Based Systems. Jurix: The 11th Conference*, pages 113–130, 1998.

[128] Maria Paz Garcia Villalba and Patrick Saint-Dizier. Some facets of argument mining for opinion analysis. In Bart Verheij, Stefan Szeider, and Stefan Woltran, editors, *COMMA 2012*, volume 245, pages 23–24, 2012.

[129] W. B. W. B. Dolan and C. Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proc. of the 3rd Int. Workshop on Paraphrasing*, pages 9–16, 2005.

[130] Douglas Walton. *Argumentation Theory: A Very Short Introduction*, pages 1–22. Springer, 2009.

[131] Douglas Walton. Some artificial intelligence tools for argument evaluation: An introduction. *Argumentation*, 2016.

[132] Xu Wang, Miaomiao Wen, and Carolyn Rosé. Contrasting explicit and implicit scaffolding for transactive exchange in team oriented project based learning. In *CSCL 2017 Proceedings*, pages 25–32, 2017.

[133] V. Wei Feng and Graeme Hirst. A linear-time bottom-up discourse parser with constraints and post-editing. In *Proceedings of the 52th Annual Meeting of*

the Association for Computational Linguistics: Human Language Technologies, 2014.

[134] V. Wei Feng and Graeme Hirst. Two-pass discourse segmentation with pairing and global features. arXiv:1407.8215v1, 2014.

[135] Kate Wilson and Linda Devereux. Scaffolding theory: High challenge, high support in academic language and learning (all) contexts. *Journal of Academic Language & Learning*, 8(3):91–100, 2014.

[136] B.P. Woolf, T. Murray, D. Marshall, T. Dragon, K. Kohler, M. Mattingly, M. Bruno, D. Murray, and J. Sammons. Critical thinking environments for science education. In *Proceedings of the 12th International Conference on AI and Education,*, 2005.

[137] Hwee Ziheng Lin, Tou Ng, and Min-Yen Kan. A PDTB-styled end-to-end discourse parser. *Natural Language Engineering*, 20(2):151–184, 2014.

[138] D. Zwillinger and S. Kokoska. *CRC Standard Probability and Statistics Tables and Formulae*. Chapman & Hall, 2000.