

Minería de argumentos en ambientes de aprendizaje colaborativo soportado por computadoras

Ariel Monteserin¹, Marcelo Armentano¹, Nicolás García¹, Damián Palavecino¹

¹ISISTAN (CONICET-UNCPBA) - Campus Universitario, Tandil, Bs.As. Argentina

{ariel.monteserin,marcelo.armentano}@isistan.unicen.edu.ar

Abstract. *Poder determinar automáticamente si una sentencia expresada en lenguaje natural es un argumento bien conformado es una tarea clave en sistemas de aprendizaje colaborativo, especialmente cuando se pretende analizar la calidad de los argumentos elaborados por los estudiantes. Por esta razón, en este trabajo presentamos un enfoque de minería de argumentos que se basa en el procesamiento del lenguaje natural de una sentencia, particularmente mediante un análisis de su estructura sintáctica, y el posterior entrenamiento de un modelo de clasificación. La evaluación experimental fue realizada con estudiantes reales y mostró resultados prometedores. En particular, el enfoque basado en conectores argumentativos con etiquetas diferenciadas para indicadores de premisas y conclusiones fue el que obtuvo un mejor desempeño.*

1. Introducción

Diversos trabajos han señalado que el aprendizaje es particularmente efectivo cuando los estudiantes encuentran conflictos y los resuelven a través del debate para generar una solución compartida [Veerman 2003]. La clave para que dicho debate sea correctamente conducido radica en cómo los estudiantes construyen sus argumentos. En el contexto de los ambientes de aprendizaje colaborativo soportado por computadoras, es vital poder determinar automáticamente si una sentencia es o no es un argumento. Esta automatización permite entre otras cosas analizar el nivel de argumentos intercambiados por un grupo de estudiantes durante la resolución de un problema o detectar fallas en la construcción de sus argumentos, entre otras aplicaciones.

Para poder identificar argumentos automáticamente se debe, en primer lugar, analizar cuál es el formato que poseen dichos argumentos. Tradicionalmente un argumento es definido como un conjunto de premisas al cual sigue una conclusión [Palau and Moens 2009]. Sin embargo, cuando hablamos de argumentos expresados en lenguaje natural y especialmente en sistemas de discusión *online* dentro de un ambiente de aprendizaje colaborativo, mayormente informales, este formato puede verse desdibujado. En primera medida, las partes que conforman un argumento no siguen ningún orden preestablecido. Se puede presuponer que todas las premisas se encuentran al inicio del argumento y la conclusión al final del mismo. Sin embargo, un argumento bien constituido puede contener ambas partes entrelazadas, sumando dificultades al momento de determinar una estructura que lo defina. Otro de los desafíos en la detección automática de argumentos es la presencia de premisas tácitas, debido a presunciones por parte de los autores de los argumentos. Es decir, basándose en conocimientos previos, las personas omiten detalles que a su parecer son obvios. Por otra parte, un nuevo desafío que se presenta al momento de identificar argumentos es la independencia contextual del tópico de la discusión, es decir, que la solución presentada no esté supeditada a una única temática.

En este contexto, proponemos un enfoque de minería de argumentos mediante el análisis de la estructura sintáctica del texto y su posterior clasificación. La primera tarea es llevada a cabo por un módulo de procesamiento de lenguaje natural el cual permite etiquetar los textos de entrada con el fin de centrar el análisis en la estructura sintáctica de los argumentos, sin que el aspecto semántico sea influyente. Con esto se determina la función de cada palabra o frase dentro de las sentencias (verbo, sustantivo, adverbio, etc.) y se logra independizar el modelo del dominio en el cual se está trabajando. En este punto, se proponen dos tipos de procesamiento: enfoque basado en longitud de etiqueta y enfoque basado en conectores argumentativos. La primera trata a todas las palabras o frases por igual y varía la cantidad de información sintáctica almacenada en cada etiqueta. En cambio, la segunda procesa de forma particular aquellas palabras o frases fuertemente identificadas con la estructura de los argumentos. Una vez realizado este procesamiento, un clasificador es utilizado para determinar si el texto procesado es un argumento o no. Durante la experimentación distintos algoritmos fueron evaluados, entre los que se encuentran *Support Vector Machines* (SVM), *Naives Bayes*, árboles de decisión (J48) y k-vecinos más cercanos.

Finalmente, la validación de este enfoque fue realizada con estudiantes universitarios quienes debieron expresar, en un sistema de aprendizaje colaborativo soportado por computadoras, argumentos a favor o en contra de un conjunto de técnicas de inteligencia artificial. Las sentencias ingresadas por los estudiantes fueron analizadas por un experto para determinar si efectivamente eran argumentos o no y se utilizaron para entrenar el modelo propuesto. Los resultados comparan distintas combinaciones de procesamiento de texto y distintos clasificadores, mostrando un efectividad máxima del 76,24 % en la detección de argumentos. Dicha efectividad fue obtenida mediante la utilización del enfoque basado en conectores argumentativos y el clasificador basado en arboles de decisión.

El artículo se estructura de la siguiente forma. La sección 2 presenta los trabajos relacionados presentes en la literatura. En la sección 3, presentamos el enfoque de minería de argumentos propuesto en este trabajo. La sección 4 muestra los resultados obtenidos en la evaluación experimental. Finalmente, en la sección 5, presentamos las conclusiones y los trabajos futuros.

2. Trabajos relacionados

Un argumento se puede definir como una serie de aseveraciones, sea dentro de una oración o dentro de un grupo de oraciones, que individualmente o en su conjunto dicen “apoyar”, “demostrar” o “dar prueba de” otra aseveración. La aseveración que resulta apoyada, demostrada o probada se denomina conclusión. Hay una sola conclusión para cada argumento, pero puede haber una serie de aseveraciones de apoyo denominadas premisas.

Algunos trabajos han sido desarrollados con el objetivo de minar argumentos en diversos textos. Estos trabajos han sido en su mayoría desarrollados para el idioma inglés, no encontrándose hasta el momento trabajos realizados para el idioma Castellano. Asimismo, el principal dominio de aplicación de estos trabajos es en el análisis de textos legales, siendo nula su aplicación en el contexto de educación. Particularmente, en [Palau and Moens 2009] centraron su trabajo en este tipo de textos, brindando una herramienta a los abogados para facilitar el procesamiento de dichos textos. Por un lado, se enfocaron en clasificar si un determinado texto era o no un argumento, teniendo como

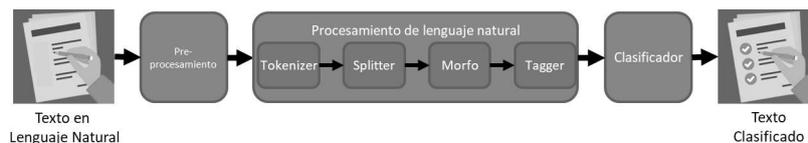


Figura 1. Enfoque de minería de argumentos propuesto.

entrada textos en idioma anglosajón que contienen hechos, antecedentes y explicaciones. Por otro lado, se plantearon como desafío descomponer dichos argumentos en los componentes que lo conforman: premisas y conclusiones. Cada sentencia fue representada como un vector de características, entre las que se pueden mencionar: unigramas, bigramas, trigramas, adverbios, verbos, auxiliares modales, entre otras. Teniendo en cuenta estas características se entrenaron dos clasificadores, *Naives Bayes* Multinomial y Modelo de Máxima Entropía, usando como conjunto de datos el Corpus Araucaria.

Por otra parte, en [Chow 2016] se desarrollaron varios métodos para la detección de argumentos orientados a corpus compuestos por textos de editoriales chinas, asumiendo que estos son textos ideales, a pesar de que una única editorial pueden contener tanto textos argumentativos como no argumentativos. Estos trabajos se basan principalmente en un enfoque de frecuencia de palabras, tomando como base la idea de que las palabras que aparecen más frecuentemente en textos de editoriales que en textos de otros ámbitos pueden ser palabras que soportan argumentos. Adicionalmente, Chow propone un enfoque basado en la creación de una lista de palabras argumentativas y la posterior puntuación de su cantidad de ocurrencias.

3. Enfoque de minería de argumentos

La Figura 1 muestra gráficamente los componentes del enfoque de minería de argumentos propuesto. Como podemos ver, nuestro enfoque toma como punto de partida texto en lenguaje natural y determina si dicho texto representa o no a un argumento. Para realizar esta clasificación, nuestro enfoque está compuesto de tres componentes principales: un módulo de pre-procesamiento, un módulo de procesamiento de lenguaje natural, y un modelo de clasificación. En las subsecciones siguientes, explicaremos cada uno de estos componentes en detalle.

3.1. Pre-procesamiento

El enfoque propuesto está ideado para procesar texto en lenguaje natural que puede proceder tanto de foros de discusión como de *chats* o documentos *online* que estén embebidos en un sistema de aprendizaje colaborativo soportado por computadoras. En este ámbito predomina el lenguaje informal, con una estructura lingüística poco clara y abreviaciones conocidas en ese contexto, pero no tan claras al momento de analizarlas automáticamente. Con el objetivo de que el “ruido” generado por estas variantes no influya significativamente en el modelo generado, se decidió realizar en primer lugar un pre-procesamiento de los datos de entrada. Este pre-procesamiento consta de dos pasos principales. En primer lugar, con el objetivo de facilitar la delimitación de los argumentos, se considera que cada argumento debe estar contenido en un único párrafo. En caso de que un argumento candidato contenga un salto de línea, este es eliminado.

Finalmente, con el objetivo de reducir el ruido que pudiera generar la presencia de errores ortográficos, se añadió un módulo de corrección ortográfica. El mismo consta de una interfaz visual en la cual se muestra el conjunto de sentencias resaltando los errores

encontrados. Para cada uno de ellos se presenta una lista de posibles correcciones. Dichas correcciones pueden ser aplicadas automáticamente o manualmente por un experto.

3.2. Procesamiento de lenguaje natural

Si bien el preprocesamiento es clave para permitir obtener una correcta precisión durante la clasificación, el principal componente de nuestro modelo es el módulo de procesamiento de lenguaje natural. Dado que no existe una única estructura sintáctica para expresar un argumento, se enfocó el análisis no tanto en el aspecto semántico, sino en la función que cumple cada palabra en una sentencia. El procesamiento de lenguaje natural es un proceso modo *pipeline*, compuesto por diversos módulos con funciones específicas, que tienen una entrada y una salida previamente definidas. Cada módulo tiene como entrada la información procesada por un módulo anterior, y como resultado se obtiene información enriquecida útil para los módulos que continúan el *pipeline*. La herramienta seleccionada para llevar a cabo esta tarea fue *FreeLing*¹. En la Figura 1 se detalla los submódulos que componen el *pipeline* comprendido en el módulo de procesamiento de lenguaje natural (PLN): *tokenizer*, *splitter*, *morfo* y *tagger*. A continuación se describen cada uno de los módulos utilizados:

- *Tokenizer*. Es la etapa inicial del PLN, en donde se tiene como entrada el texto plano y devuelve una lista de objetos tipificados. Entre los elementos lingüísticos que son reconocidos en este módulos se encuentran las *urls*, las direcciones de *e-mails*, abreviaturas, horas, puntos suspensivos, entre otras.
- *Splitter*. Dado el conjunto de objetos generado por el submódulo *Tokenizer* retorna una lista de sentencias. En otras palabras, es el módulo que delimita el inicio y el fin de cada oración.
- *Morfo*. Este módulo aplica una cascada de analizadores especializados, entre los que se puede mencionar la detección de números, fechas, locuciones y multipalabras, entre otros. En este módulo se hizo hincapié puntualmente en el analizador orientado a la detección de palabras compuestas, el cual asigna etiquetas especiales, basándose en un archivo de configuración. De esta manera, se puede especificar palabras compuestas que en un determinado análisis tienen peso o es importante su detección, y que en otro contexto no lo son tanto.
- *Tagger*. Recibe una lista de sentencias y desambigua la categoría morfosintáctica de cada palabra en las frases de la lista. El módulo utilizado está basado en modelos ocultos de Markov y como resultado agrega una categoría a cada *token* generado por las etapas anteriores. En caso de que a un *token* se le haya asignado una etiqueta especial en la etapa anterior, esta es mantenida y no se vuelve a etiquetar.

3.2.1. Enfoque basado en longitud de etiquetas

Tomando como base el procesamiento descrito en la sección anterior, en este trabajo planteamos distintas alternativas que surgen de variar algunos de sus parámetros. La primera variante se basó en la salida pura brindada por el módulo de PLN. Con este procesamiento se obtuvieron 150 atributos distintos, cada uno vinculado con al menos una

¹<http://nlp.lsi.upc.edu/freeling/node/1>

FORMA	LEMA	ETIQUETA	FORMA	LEMA	ETIQUETA
chico	chico	NCMS000	oyente	oyente	NCCS000
chicas	chico	NCFP000	oyentes	oyente	NCCP000
gatito	gato	NCMS00D	Barcelona	barcelona	NP000G0
tesis	tesis	NCFN000	Pedro	pedro	NP000P0

Cuadro 1. Ejemplo de etiquetas.

por lo tanto	por estas razones	por consiguiente	lo que quiere decir que	consecuentemente	por esta razón	de este modo
se sigue que	podemos inferir que	concluyo que	lo que lleva a la conclusión	lo que implica que	como resultado	así que
de ahí que	lo que muestra que	en consecuencia	lo que permite inferir que	lo que conlleva a	prueba que	luego

Cuadro 2. Lista de indicadores de conclusión.

ocurrencia dentro de alguna sentencia contenida en los datos de entrenamiento. Las etiquetas definidas por EAGLES² y adoptadas por *FreeLing* comienzan con una clasificación genérica en el primer carácter que define qué tipo de palabra es (por ejemplo, adjetivo, adverbio, verbo, sustantivo, etc.), y las restantes denotan características tales como género, número, entre otros. Teniendo en cuenta esta información, se pensó que no todos los caracteres eran determinantes al momento de detectar un argumento.

En el cuadro 1 se puede visualizar que para las palabras “chico” y “chicas”, la etiqueta difiere aun cuando el lema es el mismo. Se puede ver que los primeros dos caracteres coinciden y se diferencian al momento de indicar género y número en la tercera y cuarta posición respectivamente. Por este motivo se pensó que un sustantivo sea singular o plural es irrelevante al momento de realizar la clasificación. Basándonos en este principio, se evaluaron dos enfoques teniendo en cuenta la longitud de etiquetas. Dichos enfoques consideraron etiqueta completa y etiqueta de longitud 2 (es decir, únicamente con las 2 primeras características).

3.2.2. Enfoque basado en conectores argumentativos

Un análisis preliminar de los argumentos candidatos nos confirmó que no existe un orden establecido en el cual deban aparecer tanto premisas como conclusiones dentro de un argumento. Por este motivo, se propone un conjunto de enfoques alternativos cuyo objetivo consiste en diferenciar aquellas palabras o frases que indican la presencia de componentes argumentativos. De esta forma, algunas palabras o frases normalmente sirven para introducir la conclusión de un argumento y son, por lo tanto, indicadores de conclusión. Los argumentos con indicadores de conclusión tienen, por ejemplo, la siguiente estructura: [premisas], [indicador de conclusión], [conclusión]. En el cuadro 2 se muestra una lista parcial de los posibles indicadores de conclusión.

Otras palabras o frases normalmente sirven para señalar las premisas de un argumento y, por lo tanto, las llamamos indicadores de premisas. Comúnmente, aunque no siempre, lo que sigue a cualquiera de estos indicadores es la premisa del argumento. Los argumentos con indicadores de premisas tienen, por ejemplo, esta estructura: [conclusión], [indicador de premisa], [premise]. Algunos posibles indicadores de premisas se muestran en el cuadro 3.

²<http://www.lsi.upc.es/~nlp/tools/parole-sp.html>

puesto que	como lo indica	porque	puede derivarse de	como lo muestra
la razón es que	por la razón de que	ya que	en vista del hecho de que	dado que
se sigue de	puede inferirse de	como	puede deducirse de	en base a

Cuadro 3. Lista de indicadores de premisa.

Definidos los posibles conectores argumentativos, tanto para premisas como para conclusiones, se presentan a continuación diversas transformaciones aplicadas a los mismos:

- Conectores argumentativos sin etiquetar: Se toma la lista de conectores y se etiqueta su etiquetado.
- Conectores argumentativos con etiqueta unificada: como en el ítem anterior, se tienen en cuenta la totalidad de los conectores, sin discriminar entre los orientados a preceder premisas o conclusiones. Estos conectores fueron etiquetados con una única etiqueta especial.
- Conectores argumentativos con etiqueta diferenciada: en esta transformación se etiqueta los conectores distinguiendo entre los indicadores de premisas y los indicadores de conclusión. Siempre tomando como base la lista de conectores mencionada previamente, se rotuló con las etiquetas “tag-pre” y “tag-con” aquellos indicadores de premisa y conclusión, respectivamente. El objetivo de este análisis fue identificar el comienzo de premisas o conclusiones basándonos en la presencia de un conector y no por las palabras específicas que lo componen.

3.3. Construcción del clasificador

Una vez realizado el pre-procesamiento y el procesamiento de lenguaje natural, el objetivo de esta etapa consiste en encontrar el modelo que maximice la tasa de detección de argumentos. En los trabajos relacionados mencionados anteriormente, fue constante la utilización de SVM y Naives Bayes como algoritmos típicos al momento de generar un modelo que reconozca y clasifique argumentos. A pesar de eso, se decidió evaluar otros algoritmos. Los modelos de clasificación evaluados son: Naives Bayes, SVM, arboles de decisión, y k-vecinos más cercanos. En este punto, se utilizó la herramienta Weka³ para entrenar y evaluar los distintos modelos generados. A continuación, se detallan brevemente los modelos de clasificación utilizados:

- Naive Bayes (*NB*): un clasificador NB es un modelo bayesiano que simplifica el aprendizaje al asumir que los atributos (variables observables) son independientes dada la clase (variable inferida). A pesar de esta afirmación irreal, los clasificadores obtenidos son muy exitosos en la práctica, compitiendo frecuentemente con técnicas mucho más sofisticadas [Rish 2001].
- Arboles de decisión (*J48*): El modelo producido por este algoritmo es un árbol [Quinlan 1993], donde cada nodo corresponde a un atributo, cada arco del nodo corresponde a un posible valor del atributo nodo, y cada hoja final es la clase inferida.
- Support Vector Machines (*SVM*): los modelos basados en SMV [Cristianini and Shawe-Taylor 2000] intentan separar los ejemplos, basándose en su categoría, en el espacio de n dimensiones siendo n el número total de atributos o características, mediante hiperplanos.

³<http://www.cs.waikato.ac.nz/ml/weka/>

- **K-vecinos más cercanos (KNN):** KNN es un algoritmo basado en memoria con la idea subyacente de que las experiencias pasadas pueden ayudar a resolver las presentes mediante analogía [Cunningham and Delany 2007]. Considera a cada ejemplo como un vector de n componentes, siendo nuevamente n el número de atributos o características. Para inferir la clase de un ejemplo desconocido hasta el momento, el algoritmo compara ese ejemplo con todos los ejemplos de entrenamiento calculando la distancia entre ellos. A continuación, la clase mayoritaria de entre los K ejemplos más similares al de entrada es la categoría inferida para el mismo. La medida de distancia empleada es la distancia Euclídea entre dos vectores.

4. Evaluación experimental

La evaluación experimental se realizó con alumnos de la materia Inteligencia Artificial de la carrera de Ingeniería de Sistemas de la UNCPBA. Durante la experimentación, los alumnos tuvieron que resolver distintos trabajos prácticos sobre diversas técnicas de inteligencia artificial en un sistema experimental de aprendizaje colaborativo soportado por computadoras. Particularmente, los alumnos resolvieron dos trabajos prácticos uno sobre reglas de asociación y otro sobre técnicas de clasificación. Luego, cada alumno debió elaborar una serie de argumentos a favor, en contra o comparando las técnicas estudiadas en los trabajos prácticos previos. En total, durante los experimentos participaron 37 estudiantes, los cuales generaron un total de 264 sentencias (nos referimos a sentencias pues no necesariamente todas fueron argumentos bien formados). A continuación, se evaluó cuáles sentencias eran efectivamente un argumento y cuáles no. Para ello, se etiquetó como argumentos aquellas sentencias conformadas por una o varias premisas y una conclusión. Tras esta tarea, fueron identificados 101 argumentos (38,26 % del total). Para mantener, el conjunto de datos balanceado se seleccionó al azar, a su vez, 101 sentencias no argumentos conformando así un total de 202 sentencias de entrenamiento.

Una vez, conformado el conjunto de datos de entrenamiento, la evaluación experimental consistió en evaluar distintos escenarios de acuerdo a las distintas alternativas planteadas en la sección 3. Así, se realizaron experimentos en 6 escenarios. En primer lugar, se llevaron a cabo experimentos con el conjunto de datos sin etiquetar, es decir sin aplicar ningún procesamiento de lenguaje natural. Este escenario sirvió principalmente como base para la comparación de resultados obtenidos por el enfoque propuesto. El resto de los escenarios evaluados consistieron en el enfoque basado en longitud de etiqueta (dos escenarios correspondientes a etiqueta completa y longitud de etiqueta igual a 3) y el enfoque basado en conectores argumentativos (tres escenarios correspondientes a conectores sin etiquetar, con etiqueta unificada y con etiqueta diferenciada).

Para cada uno de los escenarios, se aplicó el procesamiento correspondiente y a continuación se construyeron los modelos de clasificación previamente descritos (Sección 3.3). Finalmente, distintas métricas fueron computadas y comparadas para determinar cuál de los enfoques obtuvo un mejor desempeño durante la clasificación de los argumentos. Las métricas utilizadas fueron:

- **Efectividad (EF):** mide el número de sentencias correctamente clasificadas respecto al total de sentencias del conjunto de datos, siendo su ecuación $EF = (VP + FP) / (VP + FP + FN + VN)$ donde VP corresponde al total de verdaderos positivos, FP es el total de falsos positivos, FN es el total de falsos negativos y VN es el total de verdaderos negativos.

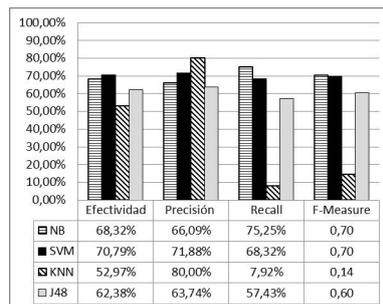


Figura 2. Resultados obtenidos con el enfoque base (sin etiquetar).

- **Precisión (P):** mide el número de sentencias correctamente reconocidas respecto al total de sentencias predichas, sean éstas ciertas o no. $P = VP / (VP + FP)$.
- **Recall (R):** mide la proporción de sentencias correctamente reconocidas respecto al total de sentencias reales. Cuando este valor se acerca al 1, nos indica que el modelo se encuentra más cerca de detectar la totalidad de los términos a clasificar. $R = VP / (VP + VN)$.
- **F-Measure (F):** es una métrica que permite caracterizar con un único valor la bondad de un clasificador o algoritmo. $F = 2 * (Precision * Recall) / (Precision + Recall)$.

A continuación, se detalla escenario a escenario los resultados obtenidos.

4.1. Resultados del enfoque sin etiquetar

Con la idea de tener un modelo base que nos sirva como punto de partida para evaluar los resultados obtenidos, se decidió generar modelos cuya entrada no fuera procesada por la herramienta de PLN. La figura 2 muestra los resultados obtenidos para el enfoque sin etiquetar con los distintos modelos de clasificación. Como se puede observar, entre los resultados obtenidos a partir de los modelos generados resaltaron los algoritmos NB y SVM. En el caso de NB se destaca por tener el mejor *Recall*, es decir una mejor tasa de acierto en la detección de argumentos etiquetados como tales. Sin embargo, SVM demostró tener tanto mayor efectividad como mejor precisión. En un cuanto al *F-Measure* ambos modelos son similares por lo que la efectividad termina por definir cuál será la mejor opción. En conclusión, el mejor resultado se obtuvo con SVM teniendo una efectividad del 70,79 % y un *F-Measure* de 0,7. Estos resultados fueron considerados como valores de referencia a mejorar por medio de las diversas alternativas del enfoque propuesto.

4.2. Resultados del enfoque basado en longitud de etiquetas

Para esta alternativa se incluyó el módulo de procesamiento de lenguaje natural y se analizaron los resultados teniendo en cuenta la longitud de las etiquetas almacenadas. Como se explicó anteriormente en la primera variante se mantiene la etiqueta con la longitud original que, dependiendo de la función semántica, puede variar entre 2 y 7 caracteres. Para la variante restante se acota el tamaño máximo de las etiquetas pudiendo éstas tener una longitud igual a 2. La figura 3.a muestra los resultados obtenidos por el enfoque de longitud de etiqueta completa, mientras que la figura 3.b muestra los resultados del enfoque de longitud de etiqueta igual a 2.

Si tenemos en cuenta el modelo generado a partir de la etiqueta completa, se puede ver que NB es el algoritmo que mejor resultados arroja con un *F-Measure* de 0,67 y una

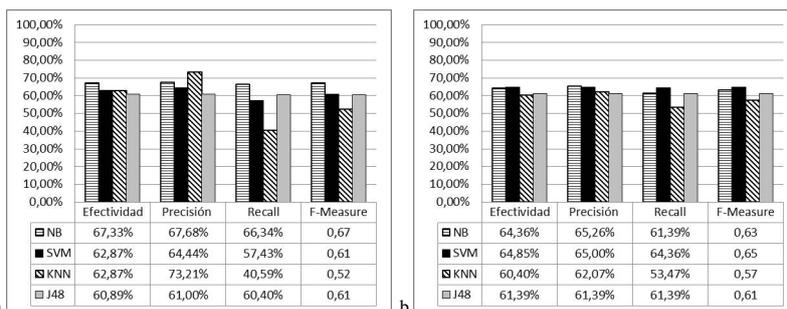


Figura 3. Resultados obtenidos por el enfoque de longitud de etiqueta completa (a) y longitud 2 (b).

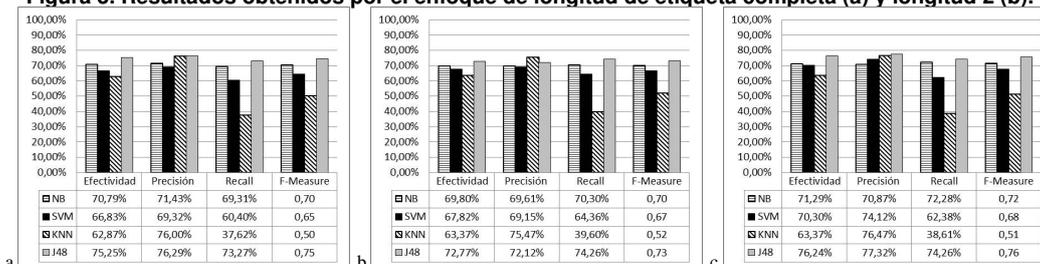


Figura 4. Resultados obtenidos por el enfoque basado en conectores argumentativos en sus distintas variantes.

efectividad del 67,33 %. En el modelo generado a partir de una transformación que limita la longitud de etiqueta a 2, el *F-Measure* obtenido es de 0,65 y la efectividad de 64,85 % utilizando SVM como algoritmo de clasificación. Los resultados muestran que ninguna variante mejora los resultados obtenidos por el enfoque base. Esto puede deberse a que la utilización de etiquetas disminuye la cantidad de atributos que componen el conjunto de datos de entrenamiento provocando así la pérdida de características esenciales a la hora de realizar la detección. Por otra parte, las particularidades en la redacción de los argumentos candidatos pueden provocar deficiencias al momento de determinar la función semántica de algunas palabras provocando “ruido” en el modelo. Finalmente, es importante destacar que ninguno de los clasificadores del enfoque de longitud 2 mejora el mejor resultado obtenido por el clasificador NB del enfoque de etiqueta completa.

4.3. Resultados del enfoque basado en conectores argumentativos

Este enfoque fue basado en la diferenciación de conectores argumentativos y las diferentes transformaciones que se le pueden aplicar al conjunto de datos previos a la generación del modelo. La primera alternativa no altera dichos conectores quedando tal como fueron escritos originalmente mientras que las dos restantes asignan un rotulo particular a cada una de ellas. La figura 4.a muestra los resultados obtenidos por el enfoque de conectores argumentativos sin etiquetar. Por otra parte, las figuras 4.b y 4.c muestran los resultados de los enfoques basados en conectores argumentativos con etiqueta unificada y diferenciada, respectivamente.

Cuando analizamos los resultados de la alternativa que utiliza los conectores en su formato original, se observó que J48 es el modelo de clasificación que mejor se adapta, obteniendo una efectividad del 75,25 % y un *F-Measure* de 0,75. Dicho modelo sigue siendo el mejor para el enfoque con etiqueta unificada. No obstante, las métricas obtenidas por este enfoque empeoran con respecto al anterior. Esto se debe a que la etiqueta unificada no permite determinar exitosamente si una sentencia posee tanto premisas como conclusiones, condiciones necesarias para formar un argumento.

Finalmente, si diferenciamos entre premisas y conclusiones al momento de etiquetar los conectores, obtenemos un *F-Measure* de 0,76 y una efectividad del 76,24 %

utilizando el algoritmo J48. Este modelo presenta la mejor tasa de acierto al momento de detectar sentencias argumentativas, siendo estas métricas las más optimas entre todos los modelos generados. Cabe destacar que todas las variantes evidencian una mejora en cada una de las métricas analizadas al momento de compararlas con el enfoque base. Esto se debe a que si diferenciamos los conectores argumentativos, estos resaltan por sobre el conjunto de etiquetas ayudando a la correcta clasificación de la sentencia.

5. Conclusiones y trabajos futuros

En este trabajo, hemos presentado un enfoque para determinar si sentencias expresadas en lenguaje natural son o no argumentos. Dicho enfoque está destinado al análisis automático de texto en sistemas de aprendizaje colaborativo soportado por computadoras. Los resultados experimentales mostraron que el enfoque basado en conectores argumentativos con etiqueta diferenciada obtuvo el mejor desempeño. Esto se debe a que dicho enfoque permite determinar la presencia de los componentes básicos de un argumento: premisas y conclusión.

Los trabajos futuros se centrarán principalmente en la aplicación de este enfoque en el seguimiento de estudiantes mientras desempeñan actividades en sistemas de aprendizaje colaborativo. Este seguimiento incluirá, por ejemplo, el análisis del volumen de argumentos intercambiados por los estudiantes durante sesiones de trabajo grupal. Esto permitirá entre otras cosas analizar cuál es la mejor forma de armar grupos teniendo como objetivo favorecer el debate durante la resolución de trabajos colaborativos. Por otra parte, nos enfocaremos en el análisis de las sentencias no argumentativas con el objetivo de analizar las causas por las cuales el estudiante falló en su elaboración. Esto nos permitirá asistir a los estudiantes en la elaboración de sus argumentos.

Agradecimientos. Este trabajo fue parcialmente financiado por los proyectos PICT 2014-2750 y PIP 2015-2017 número 11220150100030CO.

Referencias

- Chow, M. (2016). Argument identification in chinese editorials. In *Proceedings of NAACL-HLT 2016*, pages 16–21.
- Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines: And Other Kernel-based Learning Methods*. Cambridge University Press, New York, NY, USA.
- Cunningham, P. and Delany, S. J. (2007). k-nearest neighbour classifiers. *Multiple Classifier Systems*, 34:1–17.
- Palau, R. M. and Moens, M.-F. (2009). Argumentation mining: The detection, classification and structure of arguments in text. In *Proc. of the 12th Int. Conf. on Artificial Intelligence and Law, ICAIL '09*, pages 98–107, New York, NY, USA. ACM.
- Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- Rish, I. (2001). An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46. IBM New York.
- Veerman, A. (2003). Constructive discussions through electronic dialogue. *Arguing to learn: Confronting cognitions in CSCL environments*, 1:117–143.