

# Reconocimiento de patrones para la clasificación de componentes argumentales en textos académicos en español

Kenia Nieto-Benitez<sup>1</sup>, Noé Alejandro Castro-Sánchez<sup>1</sup>,  
Héctor Jiménez-Salazar<sup>2</sup>

<sup>1</sup> Tecnológico Nacional de México, CENIDET, Cuernavaca, Morelos,  
México

<sup>2</sup> Universidad Autónoma Metropolitana, Unidad Cuajimalpa, Ciudad de México,  
México

{kenianieto, ncastro}@cenidet.edu.mx, hgimenezs@gmail.com

**Resumen.** En este artículo se implementa y experimenta con un método de reconocimiento de patrones para la clasificación de componentes argumentales en español, sobre un corpus de textos académicos. Experimentamos con el uso de características léxicas, estructurales e indicadores junto con una representación basada en tf-idf y binario de estas características. Entre los modelos de aprendizaje automático implementados, se obtuvo un *F-measure* de 0.84 con la Red Neuronal Perceptrón Multicapa con dos clases, premisa y conclusión, y 0.79 con tres clases, premisa, conclusión y no argumental. Ambos resultados al combinar todas las características. La experimentación realizada muestra la importancia y necesidad de determinar características para distinguir segmentos no argumentales, que hasta el momento no se han considerado en la identificación de argumentos.

**Palabras clave:** Componentes argumentales, reconocimiento de patrones, procesamiento del lenguaje natural, redes neuronales artificiales, minería de argumentos.

## Pattern Recognition for Argumentative Components Classification in Academic Spanish Texts

**Abstract.** This article we implements and experiments with a pattern recognition method for the argumentative components classification in Spanish language, it on an academic texts corpus. We experimented with the use of lexical, structural, and indicator characteristics along with a representation based on tf-idf and binary of these characteristics. Among the machine learning models implemented, an F-measure of 0.84 was obtained with Multilayer Perceptron Neural Network with two classes, premise and conclusion, and 0.79 with three classes, premise, conclusion

and non-argumentative. Both results when combining all the characteristics. The experimentation carried out shows the importance and the need for establish characteristics to distinguish non-argumentative segments, which until now have not been considered in the argument identification.

**Keywords:** Argumentative components, pattern recognition, natural language processing, artificial neural networks, argument mining.

## 1. Introducción

Las investigaciones existentes en Minería de Argumentos han implementado modelos que identifiquen de forma automática los componentes y relaciones argumentales en diversos tipos de textos, mediante el diseño e implementación de modelos de aprendizaje automático y procesamiento de lenguaje natural [1]. Sin embargo, se han desarrollado principalmente para el idioma inglés y aplicado en corpus de ensayos persuasivos. Algunas investigaciones han creado nuevos corpus utilizando otras fuentes de información como periódicos en línea, redes sociales, noticias en la Web, entre otras. También han optado por procesar otros corpus que contienen argumentos en diferentes temas, como *Araucaria*, *IBM corpus* o *LiveJournal*.

En el idioma español, hasta ahora se cuenta con un corpus a partir de textos académicos [2]. Sobre este corpus, en su primera versión, se han identificado los componentes argumentales [3] y determinado algunas características para su identificación. Si bien, presenta ciertos tipos de características, sabemos que la forma de representar el texto para la implementación de los modelos puede variar, así como también la selección de características para obtener buenos resultados.

Este trabajo presenta un método de clasificación de componentes argumentales en idioma español, basado en reconocimiento de patrones que incluye dos etapas: extracción de características y reconocimiento de clases. En la primera etapa, experimentamos con características léxicas, estructurales e indicadores. También, exploramos con una representación del texto distinta a [3], basada en valores binarios y tf-idf. En la segunda, la clasificación de componentes se realizó con tres modelos de aprendizaje automático usando un enfoque de clasificación para dos y tres clases.

El artículo está estructurado de la siguiente manera. En la sección 2, presentamos cómo se ha abordado la clasificación de componentes argumentales y los resultados obtenidos. En la sección 3, presentamos el corpus utilizado para los experimentos. En la sección 4, describimos el método implementado basado en reconocimiento de patrones, así como las características y modelos utilizados. En la sección 5, presentados y discutimos los resultados obtenidos en las pruebas con los tres modelos, asimismo la comparación entre los modelos y características. Finalmente, concluimos nuestro trabajo en la sección 6 junto al trabajo futuro.

## 2. Trabajos relacionados

Los sistemas desarrollados en Minería de Argumentos se han dedicado a implementar modelos que identifiquen de forma automática componentes y relaciones argumentales en diversos tipos de textos [1]. En este trabajo, nos hemos centrado en la identificación de componentes argumentales en textos académicos.

En [4], realizaron la segmentación de textos en unidades de argumento: conclusión, premisa, anécdota y suposición. En el caso de las dos primeras unidades, corresponden a los componentes argumentales. Abordaron esta segmentación como una tarea de clasificación mediante dos modelos de aprendizaje automático y uno de aprendizaje profundo: *Support Vector Machine*, *Conditional Random Fields* y *Bi Long Short-Term Memory* respectivamente, junto con la extracción de características semánticas, sintácticas, estructurales y pragmáticas. Reportaron mejores resultados con la red *Bi Long Short-Term Memory*. Obtuvieron un *F-scores* de 88.54 al utilizar todas las características sobre un conjunto de 402 ensayos académicos [5].

En el caso de [6], identificaron y clasificaron los componentes en tres clases, conclusión principal, conclusión y premisa, empleando el mismo corpus de ensayos que [4], pero en su versión reducida [7] con un total de 90 ensayos. Implementaron un clasificador *Support Vector Machine* con características contextuales, estructurales, léxicas, sintácticas e indicadores. Como resultado obtuvieron un F1 de 0.726.

En [8] también utilizaron el corpus presentado en [5], en este caso, para abordar la identificación automática de componentes y relaciones argumentales juntamente. Emplearon características estructurales, léxicas, sintácticas e indicadores junto a cuatro clasificadores: *Logistic Regression*, *Random Forest*, *Decision Tree* y *Support Vector Machine*. Lograron mejores resultados con *Logistic Regression* al obtener una medida F1 de 0.6081.

En trabajos anteriores, [1] presentó un conjunto de investigaciones que abordaron la identificación de componentes y relaciones argumentales desde 2011 a 2015. En su mayoría, los trabajos citados en [1] se desarrollaron principalmente para el idioma inglés, a excepción de [9] que analizaron y procesaron datos en Alemán y en griego [10,11]. Hasta el momento, se ha encontrado un solo trabajo para la clasificación de componentes argumentales sobre un conjunto de datos en español [3], en el cual se basa el presente artículo.

En [3], clasificaron los componentes en conclusión, premisa y ninguna, con base en la representación del texto *Document Occurrence Representation* (DOR). Utilizaron características léxicas, sintácticas, semánticas e indicadores con modelos de aprendizaje automático como *Support Vector Machine*, *Naive Bayes* y *Random Forest*. Obtuvieron un mejor rendimiento al combinar todas las características sobre el modelo *Support Vector Machine*.

En contraste con el enfoque presentado en [3], en este trabajo, la experimentación se centra en la clasificación de componentes mediante *Support Vector Machine* y dos clasificadores adicionales: *Logistic Regression* y *Multilayer Perceptron*, los tres basados en Reconocimiento de Patrones. En los trabajos mencionados anteriormente utilizaron diversas características, aquí hemos reducido las caracte-

terísticas a indicadores, a características léxicas y a características estructurales. De acuerdo con [12] el elemento clave para obtener buenos resultados es la selección de características y no el modelo de clasificación. También, hemos explorado con una representación del texto distinta basada en valores binarios y tf-idf.

### 3. Datos

Para nuestros experimentos, utilizamos un corpus de 468 tesis y propuestas de investigación compilado por [2]. El corpus contiene anotaciones de componentes argumentales, nivel del argumento, tipo de argumento y tipo de relación entre los componentes. En particular, incluye anotaciones de conclusiones y premisas, las cuales están vinculadas mediante relaciones argumentales de soporte y ataque. El nivel del argumento está relacionado con el número de componentes presentes en cada argumento. Los tipos de argumentos encontrados fueron causal, ejemplos y de autoridad.

De los 468 documentos, seleccionaron 300 secciones pertenecientes al planteamiento del problema, justificación y conclusión. En total, de las 300 secciones obtuvieron 901 párrafos que fueron etiquetados por dos anotadores. Obtuvieron un *inter-annotator agreement* (IAA) de 0.867 en los párrafos pertenecientes al planteamiento del problema; 0.935 pertenecientes a justificación; y 0.866 para conclusión. Posteriormente, seleccionaron solo los párrafos donde los anotadores estuvieron de acuerdo, esto redujo el número de párrafos a 856. En total, el corpus comprende de 856 párrafos (de estos, 584 contienen argumentos y 272 no presentan argumentos) y 2,104 componentes.

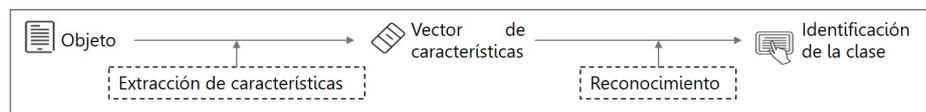
```
<seg type="conclusion"> consideramos que este software será de gran aplicabilidad, </seg>  
<seg type="premise"> ya que facilita a los profesores la enseñanza de estos tópicos de manera  
amena pero con un alto índice de confiabilidad; </seg>  
<seg type="premise"> además, sirve de base para la creación de un software hecho en México  
que se utilice en la creación de modelos tridimensionales de proteínas y su proceso de síntesis,  
con un amplio contenido técnico que puede ser utilizado tanto en docencia como en la  
investigación y la industria. </seg>
```

**Fig. 1.** Etiquetado de componentes argumentales sobre un texto académico. Obtenido de [2].

De acuerdo con los elementos del corpus, nos centramos únicamente en la identificación de componentes del argumento (ver Fig. 1). Estos se encuentran clasificados en tres clases: conclusión, premisa y ninguno. De los 2,104 componentes identificados, 562 son conclusiones, 1060 premisas y 482 componentes no argumentales.

## 4. Método

El reconocimiento de patrones se ocupa de la clasificación automática de un objeto dado en un número de categorías o clases. El objetivo del reconocimiento de patrones es el desarrollo de un sistema de clasificación automática que, dándole un objeto desconocido pueda decir a cuál clase pertenece el objeto. Para tomar la decisión analizará las características (variables) de los objetos de las diferentes clases [13].



**Fig. 2.** Método de un sistema de reconocimiento de patrones.

Un sistema de reconocimiento de patrones toma un vector de características como entrada, donde cada objeto es representado por medio de estos vectores obteniendo un patrón  $P = \{x, y\}$ , donde  $x$  es un conjunto de  $n$  características  $[x_1, x_2, x_3, \dots, x_n]$  e  $y$  la pertenencia, es decir, a la clase que pertenece el objeto.

Al final, la identificación de la clase se realiza con base en los vectores de características que se procesan mediante modelos de aprendizaje automático (ver Fig. 2). Por tanto, dado un conjunto de datos  $D = \{t_1, t_2, t_3, \dots, x_m\}$  y un conjunto de clases  $C = \{C_1, C_2, C_3, \dots, C_o\}$  se emplea una función

$$f : D \rightarrow C, \quad (1)$$

tal que cada  $t_i$  sea asignada a una clase  $C_j$ . En este trabajo, hemos implementado tres clasificadores para la ecuación (1) y tres tipos de características que nos han permitido encontrar la pertenencia  $C_j$  para cada objeto  $t_i$ .

### 4.1. Características

La selección de las características que se utilizaron para representar un objeto fueron con base en [6,3]. Por cada argumento, las características extraídas fueron indicadores, léxicas y estructurales:

- Indicadores.** Los marcadores del discurso son palabras o conjunto de palabras que permiten identificar los componentes de un argumento y se encuentran divididos en dos categorías: indicadores de premisa e indicadores de conclusión. Por ejemplo, las conclusiones se introducen frecuentemente con 'por lo tanto', 'por ende', 'en consecuencia' o 'por consiguiente', mientras que las premisas tienen marcadores como 'ya que', 'dado que', 'puesto que' o 'viendo que' [3]. Creamos el conjunto de indicadores basados en [14,15] citados por [3] que presentan diferentes marcadores argumentales en español.

- **Léxicas.** Este tipo de característica está basada en los pronombres personales. A menudo, los pronombres personales se introducen con expresiones que hacen referencia a la postura del autor [6]. Seleccionamos los pronombres 'nosotros', 'nosotras', 'nos' y 'se' con el supuesto que este tipo de pronombres son utilizados en el ambiente académico.
- **Estructurales.** Esta característica aporta información sobre el tamaño del componente, que de acuerdo con [16] citado por [6], las premisa son más amplias en promedio que las conclusiones. El número de palabras presentes en cada segmento del argumento fue dividido entre 1000 para obtener un valor menor a 1.

**Representación de características** Al trabajar con texto, las características léxicas e indicadores son categóricas y los algoritmos, dada su forma matemática, no procesan este tipo de características y deben convertirse a números. Uno de los métodos más comunes es la representación basada en los modelos *Bag of Words* o *n-gram* [17]. Si bien, actualmente existen modelos como *word2vec* y *GloVe* que representan las palabras como vectores, decidimos explorar con una representación basada en tf-idf y números binarios.

La primera representación consiste en obtener la relevancia de las palabras mediante la frecuencia del término y la frecuencia inversa del documento con base en

$$tfidf(t, d) = tf(t, d) \times idf(t, d), \quad (2)$$

donde  $tf(t, d)$  es la frecuencia del término y se obtiene a partir del número de veces que un término  $t$  aparece en un documento  $d$ .  $idf(t, d)$  es la frecuencia inversa del documento y se calcula del siguiente modo:

$$idf(t, d) = \log \frac{n_d}{1 + df(d, t)}. \quad (3)$$

En este caso  $n_d$  es el número total de documentos y  $df(d, t)$  es el número de documentos  $d$  que contiene el término  $t$ . Para obtener los valores de  $tfidf$  utilizamos la clase *TfidfTransformer* de la librería *scikit-learn*. Esta clase normaliza los valores obtenidos  $v$  a partir de (2) aplicando la normalización L2 mediante la ecuación (4) para obtener los valores finales de  $tfidf$ :

$$v_{norm} = \frac{v}{(\sum_{i=1}^n v_i^2)^{\frac{1}{2}}}. \quad (4)$$

Por último, en la segunda representación utilizamos valores binarios para representar las características léxicas, indicadores y estructurales, como una función booleana que indica si un componente argumental contiene un marcador o un pronombre. Para esta representación utilizamos el modelo *n-gram* y *Stanford Part-Of-Speech Tagger* como apoyo para obtener los vectores de características.

## 4.2. Modelos

Para la clasificación de los componentes argumentales implementamos tres modelos. Dos frecuentemente utilizados en el área de Minería de Argumentos:

Máquinas de Vectores de Soporte y Regresión Logística, y la Red Neuronal Perceptrón Multicapa. Si bien, actualmente se utilizan Redes Recurrentes o Convolucionales, ésta red no se ha utilizado para abordar problemas en Minería de Argumentos como la clasificación de componentes argumentales.

Los modelos se implementaron con apoyo de *scikit-learn Toolkit*, conjunto de librerías para aprendizaje automático en *Python*. A continuación presentamos de manera general cómo trabajan estos modelos y cómo se pueden extender, en el caso de Máquinas de Vectores de Soporte y Regresión Logística, para abordar problemas de clasificación multiclase.

**Red Neuronal Perceptrón Multicapa** Perceptrón Muticapa (*MLP*) es un algoritmo de aprendizaje supervisado para abordar problemas de clasificación no lineal o multiclase. A diferencia del Perceptrón emplea una función

$$f : \mathbb{R}^m \rightarrow \mathbb{R}^o, \quad (5)$$

donde  $m$  es el número de dimensiones de entrada y  $o$  el número de dimensiones de salida. Además, entre la capa de entra y salida puede haber una o más capas ocultas.

La capa de entrada tiene un conjunto de neuronas, que representa los datos a procesar  $X = \{x_1, x_2, x_3, \dots, x_n\}$ . Tanto la capa de entrada, oculta y salida tienen un conjunto de coeficientes  $W = \{w_1, w_2, w_3, \dots, w_m\}$  distintos para cada capa de acuerdo al número de neuronas establecidas por capa.

La clasificación se puede realizar mediante la función logística (sigmoide):

$$\phi(z) = \frac{1}{1 + e^{-z}}, \quad (6)$$

donde  $z$  es la suma lineal ponderada de los valores de entrada y coeficientes, como se especifica en (7):

$$z = \sum w_{nm} \times x_{nm} + b. \quad (7)$$

Para una clasificación multiclase, en la capa de salida se implementa una función de activación *Softmax*:

$$\phi(z) = \frac{e^{Z_i}}{\sum_{k=1}^M e^{Z_k}}, \quad (8)$$

donde  $z$  es la salida de las capas ocultas de acuerdo con (7) y  $M$  el número de funciones lineales en el modelo. Entonces, la salida  $\hat{y}$  de la red será el  $\phi(z)$  con mayor probabilidad.

MLP en la etapa de entrenamiento actualiza los coeficientes  $w$  incorporando el descenso del gradiente  $\phi(z)(1 - \phi(z))$ , que permite ajustar y obtener los valores óptimos de  $w$ , junto con una tasa de aprendizaje  $\eta$  que controla el tamaño de la actualización de  $w$ . Para evitar el sobreajuste del modelo se añade una función de regularización L2:  $\|w\|^2$  y una tasa de regularización  $\lambda$ .

**Máquinas de vectores de soporte** Una máquina de vector soporte (MVS), es un algoritmo de aprendizaje supervisado estadístico para problemas de clasificación lineal de dos grupos de datos como se especifica en (9):

$$f : \mathbb{R}^m \rightarrow \mathbb{R}. \quad (9)$$

Dado un conjunto  $S = \{(x_1, y_1), \dots, (x_n, y_n)\}$ , donde  $x_i \in \mathbb{R}^m$  e  $y_i \in \{+1, -1\}$ , mediante una función lineal:

$$f(x) = w^T \times x + b = \sum_{i=1}^m w_i \times x_i + b, \quad (10)$$

se busca un hiperplano en el espacio de variables que separe las dos clases con un margen de separación óptimo  $M$  dado por

$$M = \frac{2}{\|w\|}. \quad (11)$$

El objetivo principal de MVS es maximizar  $M$  mediante la minimización de la función objetivo:

$$\begin{aligned} & \text{mín}_{w,b,\zeta} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \zeta_i \\ & \text{s.a. } y_i(w^T x_i + b) \geq 1 - \zeta_i, \quad \zeta_i \geq 0 \end{aligned}, \quad (12)$$

donde  $\zeta_i$  son variables de holgura, número de datos no clasificados, y  $C$  es un parámetro de regularización para regular el sobreajuste y la proporción del número de datos mal clasificados.

En el caso de un problema no lineal, MVS emplea métodos kernel para proyectar las características hacia un espacio de mayor dimensionalidad, donde este espacio sea un problema lineal. Uno de los kernel más utilizados en la Función de base radial, habitualmente simplificada como:

$$K(x_i, x_j) = \exp(-\gamma \|x_i, x_j\|^2), \gamma > 0, \quad (13)$$

que puede ser interpretada como una función de similitud entre un par de muestras  $(x_i, x_j)$  en una matriz  $K$  de  $n \times n$ . El parámetro  $\gamma$  es un parámetro de influencia o alcance de las muestras y juega un papel importante en el sobreajuste del modelo. Por otro lado, MVS puede ampliarse para abordar un problema multiclase mediante las técnicas *OneVsRest* (OVR) o *OneVsOne* (OVO) que permiten construir clasificadores binarios a partir de las clases. OVR transforma un problema de  $c$  clases en  $c$  problemas binarios. En contraste, OVO construye un clasificador para cada par de clases, transformando un problema de  $c$  clases en  $c(c-1)/2$  problemas binarios.

**Regresión logística (RL)** Éste es un modelo lineal para clasificación binaria que también puede ampliarse a multiclase mediante OVR. El objetivo es predecir

la probabilidad de que una determinada muestra  $x_i$  pertenezca a una clase concreta  $\hat{y}$ . Para determinar  $\hat{y}$ , primero se emplea una función logística (sigmoide) dada en (6) para obtener la probabilidad, a diferencia que  $z$  se obtiene como:

$$z = \sum_{i=1}^m w_i \times x_i + b \quad (14)$$

Posteriormente, la probabilidad predicha se convierte en un resultado binario mediante una función umbral para determinar  $\hat{y}$ :

$$\hat{y} = \begin{cases} 1 & \text{si } \phi(z) \geq 0,5 \\ 0 & \text{cualquier otro caso} \end{cases} \quad (15)$$

Por otro parte, así como en otros modelos se tiene el riesgo de un sobreajuste en la etapa de entrenamiento. Para abordar este sobre ajuste, se incorpora una función de regularización L2:  $\|w\|^2$  y una tasa de regularización  $\lambda$  en la actualización de los coeficientes  $w$ .

## 5. Experimentación y resultados

La clasificación de componentes argumentales se abordó como una tarea de clasificación binaria/bipolar y multiclase utilizando el corpus detallado en la sección 3. En ambas clasificaciones, se experimentó con dos grupos de características. En el primero, agrupamos las características estructurales e indicadores. En el segundo, consideramos las tres características, léxicas, estructurales e indicadores. Con base en lo anterior, implementamos los modelos SVM, RL y MLP empleando un 70% de los datos para la etapa de entrenamiento y 30% en pruebas.

Realizamos cuatro pruebas para cada tipo de clasificación de acuerdo con la representación binaria y tf-idf de las características junto a la combinación de estas características en dos grupos. La Tabla 1 presenta los resultados obtenidos en las dos primeras pruebas pertenecientes a la clasificación binaria/bipolar. Primero, se combinaron las características estructurales e indicadores y se obtuvo un mejor resultado con el modelo MLP junto con la representación binaria. Posteriormente, agregamos la característica léxica como valor discriminante para la clase conclusión. Al agregar esta característica, incrementó el resultado obtenido, a un *F-measure* de 0.84, con respecto a la prueba anterior.

Las dos siguientes pruebas siguieron el mismo enfoque que las anteriores, a diferencia que se introdujo la tercer clase, ninguno. Esta clase agrupa los segmentos no argumentales. Como se observa en la Tabla 2, los resultados obtenidos en cada modelo disminuyeron. Aquí, se obtuvo un mejor resultado, *F-measure* de 0.79, al combinar todas las características con el modelo MLP junto con la representación binaria.

Las premisas del argumento tienden a ser más amplias que la conclusión. Sin embargo, los segmentos no argumentales presentes en el corpus, en su mayoría,

**Tabla 1.** F-measure para la combinación de características con dos clases.

Grupo de características	SVM	RL	MLP	Representación
Indicadores + Estructurales	0.80	0.79	<b>0.81</b>	Binaria
	0.69	0.70	0.69	TFIDF
Indicadores + Estructurales + Léxicas	0.80	0.81	<b>0.84</b>	Binaria
	0.76	0.71	0.78	TFIDF

**Tabla 2.** F-measure para la combinación de características con tres clases.

Grupo de características	SVM	RL	MLP	Representación
Indicadores + Estructurales	<b>0.67</b>	0.65	0.65	Binaria
	0.53	0.52	0.53	TFIDF
Indicadores + Estructurales + Léxicas	0.78	0.76	<b>0.79</b>	Binaria
	0.61	0.62	0.64	TFIDF

también son muy amplios. Entonces, la disminución del rendimiento de los modelos se debe al tamaño (número de palabras) de los segmentos no argumentales. A esto se añade, la influencia de los marcadores del discurso para distinguir premisas de conclusiones. Algunos segmentos no argumentales tienen marcadores como ‘por lo que’, ‘así que’, ‘por otra parte’ o ‘ya que’.

El corpus utilizado en [3] contiene menos componentes argumentales en comparación a [2] y no se encuentra disponible al público. Por ello, utilizamos el corpus presentado en [2] para los experimentos. Con este corpus, a diferencia de [3], obtuvimos un *F-measure* de 0.78 con SVM y 0.79 con la red MLP al utilizar una representación binaria. Si bien, hubo una diferencia a favor en los resultados, es necesario considerar el tamaño del corpus aunque se trate del mismo corpus, pero con menos componentes argumentales. Además, observamos que la representación binaria de las características favorece a la clase conclusión representada como 0. Dado el número de ceros que pueden tener los vectores de características con dicha representación.

Los resultados indican la importancia de la selección de características para distinguir y clasificar los componentes argumentales, al igual que en [3]. En este trabajo nos enfocamos en las características estructurales, indicadores y léxicas para distinguir premisas de conclusiones. Sin embargo, dejamos de lado las características para distinguir segmentos no argumentales, así como en los trabajos relacionados. Por lo tanto, consideramos que es conveniente determinar algunas características para distinguir estos segmentos.

## 6. Conclusiones y trabajo futuro

Presentamos un método basado en el reconocimiento de patrones para la clasificación automática de componentes argumentales en español. A diferencia de trabajos previos que emplearon ensayos, periódicos y discursos en idioma

inglés y noticias así como comentarios en blogs y redes sociales en idioma griego, en este trabajo utilizamos un corpus de 468 tesis y propuestas de investigación en español. Experimentamos con características léxicas, indicadores junto con una representación del texto basada en tf-idf y binaria. Además, utilizamos la Red Neuronal Perceptrón Multica, no empleada en trabajos previos, y dos clasificadores frecuentemente utilizados para abordar problemas existentes en Minería de Argumentos, Máquinas de vectores de soporte y Regresión logística.

La clasificación de los componentes se abordó como una clasificación de dos y tres clases. Con el método presentado, se obtuvieron mejores resultados al combinar todas las características mediante la Red Neuronal Perceptrón Multica con un *F-measure* de 0.84 en la clasificación con dos clases y 0.79 con tres clases. Estos resultados preliminares indican la importancia y necesidad de determinar algunas características para distinguir segmentos no argumentales, que hasta el momento no se han considerado en la identificación de argumentos.

Como trabajo futuro, planeamos experimentar con representaciones de texto actuales como *Word2Vec* y *GloVe*. Ampliar el conjunto de características incorporando características sintácticas y semánticas. Y por último, planeamos implementar otros modelos como Redes Neuronales Recurrentes.

**Agradecimientos.** Este trabajo fue apoyado parcialmente por el gobierno de México (beca CONACYT, SNI, TecNM 7346.19-P).

## Referencias

1. Lippi, M., Torrioni, P.: Argumentation mining: State of the art and emerging trends. *ACM Transactions on Internet Technology (TOIT)* 16(2), pp. 1–25 (2016)
2. Garcia-Gorrostieta, J.M., López-López, A.: A corpus for argument analysis of academic writing: argumentative paragraph detection. *Journal of Intelligent and Fuzzy Systems* 36(5), pp. 4565–4577 (2019)
3. Garcia-Gorrostieta, J.M., Lopez-Lopez, A.: Argument component classification in academic writings. *Journal of Intelligent and Fuzzy Systems* 34(5), pp. 3037–3047 (2018)
4. Ajjour, Y., Chen, W.F., Kiesel, J., Wachsmuth, H., Stein, B.: Unit segmentation of argumentative texts. In: *Proceedings of the 4th Workshop on Argument Mining*, pp. 118–128 (September 2017)
5. Stab, C., Gurevych, I.: Parsing argumentation structures in persuasive essays. *Computational Linguistics* 43(3), pp. 619–659 (2017)
6. Stab, C., Gurevych, I.: Identifying argumentative discourse structures in persuasive essays. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pp. 46–56 (October 2014)
7. Stab, C., Gurevych, I.: Annotating argument components and relations in persuasive essays. In: *Proceedings of COLING 2014, the 25th international conference on computational linguistics: Technical papers*. pp. 1501–1510 (August 2014)
8. Du, Y., Li, M., Li, M.: Joint extraction of argument components and relations. In: *2017 International Conference on Asian Language Processing (IALP)*. pp. 1–4 (December 2017)

9. Eckle-Kohler, J., Kluge, R., Gurevych, I.: On the role of discourse markers for discriminating claims and premises in argumentative discourse. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 2236–2242 (September 2015)
10. Sardanios, C., Katakis, I. M., Petasis, G., Karkaletsis, V.: Argument extraction from news. In: Proceedings of the 2nd Workshop on Argumentation Mining, pp. 56–66 (June 2015)
11. Goudas, T., Louizos, C., Petasis, G., Karkaletsis, V.: Argument Extraction from News, Blogs, and Social Media. In: Likas A., Blekas K., Kalles D. (eds.) Artificial Intelligence: Methods and Applications, LNCS, vol. 8445, pp. 287–29. Springer, Cham (2014). [https://doi.org/10,1007/978-3-319-07064-3\\_23](https://doi.org/10,1007/978-3-319-07064-3_23)
12. Wells, G.: *MInería de falacias en el discurso político*. Universidad de Barcelona, Barcelona (2018)
13. Theodoridis, S., Koutroumbas, K.: Pattern recognition and neural networks. In: Paliouras G., Karkaletsis V., Spyropoulos C.D. (eds.) Machine Learning and Its Applications, ACAI 1999, vol. 2049, pp. 169–195. Springer, Berlin, Heidelberg (2001). [https://doi.org/10,1007/3-540-44673-7\\_8](https://doi.org/10,1007/3-540-44673-7_8)
14. Fuentes, C.: *Diccionario de conectores y operadores del español*. Arco/libros, Madrid (2009)
15. Avendaño, C.S.: Los conectores discursivos: su empleo en redacciones de estudiantes universitarios costarricenses. *Revista de Filología y Lingüística de la Universidad de Costa Rica* 31(2), pp. 169–199 (2005)
16. Biran, O., Rambow, O.: Identifying justifications in written dialogs by classifying text as argumentative. *International Journal of Semantic Computing*, 5(04), pp. 363–381 (2011)
17. Skansi, S.: Machine Learning Basics. In: *Introduction to Deep Learning. Undergraduate Topics in Computer Science*. Springer, Cham (2018) <https://doi.org/10,1007/978-3-319-73004-2>