Universidad de Buenos Aires

Facultad de Ciencias Exactas y Naturales

Especialización en Exploración de Datos

y Descubrimiento del Conocimiento

Selección de técnicas estadísticas para la vectorización de discursos políticos referidos a la reglamentación del acceso al aborto

Autora:
Fernández Urquiza, Macarena

Fecha: 28 – 10 – 2024

Índice

Resumen

En este trabajo se aborda la selección y evaluación de rasgos para el entrenamiento de modelos de aprendizaje automatizado aplicado sobre datos textuales. El *corpus* con el que se trabaja pertenece a la transcripción taquigráfica de la sesión del Senado de la Nación Argentina para la reglamentación del acceso al aborto, la cual fue obtenida mediante la técnica de *scraping*. Durante el procesamiento de los datos se recurre a una lematización automatizada sobre la que luego se realiza una corrección manual. Para la vectorización se emplean métodos estadísticos previamente probados entre los que se encuentran la diferencia de frecuencias absolutas, el *ratio* de *log-odds* y el cálculo de *TF-IDF*, entre otros. Tras evaluar los distintos vectorizadores respecto de su rendimiento para el entrenamiento de un modelo de regresión logística, se utiliza el mejor vectorizador hallado para el entrenamiento final.

1. Introducción

1.1. Antecedentes

El "Encuentro nacional de mujeres" consiste en un evento de alcance nacional en el territorio argentino que se realiza de manera anual desde 1986 y se propone auto-convocado, democrático, pluralista y federal. Como fruto de este evento surge, en 2005, la "Campaña nacional por el derecho al aborto legal, seguro y gratuito", una alianza de organizaciones que promueve y lucha por la educación sexual como pilar fundamental en la decisión sobre la gestación, el acceso a métodos anticonceptions que permitan la adecuada prevención de embarazos no deseados, y el aborto legal, seguro y grauito que le posibilite a la persona gestante discontinuar con un embarazo si no desea atravesarlo. Hacia 2007, la Campaña redacta y presenta el "Proyecto de ley de interrupción voluntaria del embarazo" como iniciativa social para que sea considerado por los legisladores y, así, llegue a ser tratado en el Congreso de la Nación. Pero no es sino hasta junio de 2018, y luego de reiteradas presentaciones, que logran contar con el aval de entre 20 y 70 diputados. Recién entonces el proyecto llega a ser debatido en la Cámara de dichos legisladores, donde logra media sanción. Posteriormente, la Cámara de Senadores lo rechaza. En diciembre de 2020, el proyecto se vuelve a tratar y, esta vez, logra la aprobación del Congreso. La ley 27.610, sobre el "Acceso a la interrupción voluntaria del embarazo", entró en vigencia en la Argentina el 24 de enero de 2021 y regula el acceso a la interrupción voluntaria del embarazo y a los cuidados postaborto para todas las personas con capacidad de gestar.[1] [2]

Por otra parte, en su publicación Acceso a la información pública, el Ministerio de Justicia (s.f.) difunde que la ley 27.275 garantiza el acceso a la información pública y permite su búsqueda, acceso y solicitud, como así también su análisis, procesamiento, uso y distribución. Se considera información pública a todos aquellos "datos que generan, obtienen, transforman, controlan o cuidan los organismos del Estado y empresas indicados en la ley ". Estos datos, a su vez, "tienen que estar contenidos en documentos de cualquier formato o soporte: pueden estar en papel, en archivos digitales, etc.". Entre los organismos del Estado mencionados, se encuentra el Poder Legislativo de la Nación, conformado por la Cámara de Diputados y Senadores, los cuales Luego de sus sesiones, ambas cámaras disponibilizan en formato digital la transcripción taquigráfica de la reunión, de modo que cualquier ciudadano con acceso a una computadora e internet pueda acceder a ella.

Jurafsky (2000)[3] reseña una serie de aspectos relacionados con la semántica léxica, la cual se ocupa del estudio lingüístico del significado de las palabras, para argumentar que "un modelo del significado de las palabras nos debería permitir realizar inferencias para abordar tareas relacionadas al significado". Entre dichos aspectos, menciona la noción de similitud. De ella dice que "mientras que

[1] Campaña nacional por el derecho al aborto legal, seguro y gratuito (s.f.).
[2] Fundación Huésped (s.f.).
[3] Las citas incluidas a continuación fueron traducidas del inglés por la autora de este trabajo.

las palabras no tienen muchos sinónimos, la mayoría de ellas sí tiene muchas palabras similares. [...] Al movernos de la sinonimia a la similitud, es útil pasar de hablar de relaciones entre sentidos de las palabras (como en la sinonimia) a hablar de relaciones entre las palabras (como en la similitud). Lidiar con palabras nos evita tener que comprometernos con una representación particular del sentido de las palabras, lo cual simplifica nuestra tarea. [...] Conocer cuán similares son dos palabras puede ayudar a computar cuán similar es el significado de dos frases u oraciones [...]". Del mismo modo se detiene en otro tipo de relación entre las palabras como es el campo semántico, el cual "refiere a un conjunto de palabras que cubren un dominio semántico particular y mantienen relaciones estructurales entre sí", y también en la connotación o el sentio afectivo de las palabras, el cual indica "los aspectos del significado de una palabra que están relacionados con las emociones, sentimientos, opiniones y evaluaciones de un escritor o lector". En particular, el análisis de sentimientos permite identificar "el lenguaje usado para evaluaciones positivas o negativas". Luego de enunciar estas y otras cuestiones de interés para el análisis semántico, Jurafsky resalta que la "la semántica vectoial es el método tradicional de repressentar el significado de las palabras en el PLN[4], ayudándonos a modelar muchos de los aspectos del significado de las palabras", y menciona que "en el modelo de *TF-IDF*, un modelo base de importante, el significado de una palabra es definifo por una simple función de cálculo de las palabras cercanas [...] este método resulta en vectores muy extensos que además son ralos[5]".

Así, es la vectorización la que le permite al científico de datos convertir, mediante algún cálculo pre-establecido, el *corpus* que desea utilizar en una secuencia de números capaz de ser procesada por una computadora siguiendo cierto procedimiento. Los textos, transformados en vectores, pueden ser agrupados, analizados para la obtención de información o utilizados como conjunto de entrenamiento y testeo de modelos predictivos.

Por su parte, Monroe *et al.* (2008) exponen una serie de técnicas que podrían ser útiles a la hora de capturar palabras con contenido partidario en el discurso político y los aplican sobre datos discursivos tomados del Senado de los Estados Unidos, disponibilizados de manera pública y en formato digital. Sus objetivos se centran en la selección y evaluación de rasgos, a través de los cuales esperan distinguir qué palabras indican la adopción de una u otra postura y, de ahí, que puedan usarse como *features* confiables a la hora de entrenar un modelo. A su vez, respecto de la evaluación, pretenden no solo comprender qué palabras caracterizan a determinado partido sino, además, en qué medida o grado lo caractirzan, cuáles son las palabras más partidarias o representativas de ciertas posturas.

1.2. Objetivos

En este trabajo y a partir del acceso a los datos de la Cámara de Senadores, se retoman las técnicas estadísticas detalladas por Monroe *et al.* y se las utiliza

[4]Nota de traducción: la sigla PLN refiere a Procesamiento de Lenguaje Natural.
[5]Nota de traducción: se traduce como ralo el término del inglés *sparse*.

para vectorizar los discursos emitidos por los legisldores en la vigésimo tercera sesión, en la cual se discutió la ley para el acceso a la interrupción voluntaria del embarazo. Se intenta, con esto, evaluar alternativas posibles al tradicional *TF-IDF* y comparar su desempeño en la selección de rasgos y entrenamiento de modelos.

1.2.1. Objetivos generales

Como objetivos generales, este trabajo persigue:

- Comparar distintos métodos estadísticos que permitan representar discursos en términos numéricos.
- Seleccionar uno o más métodos estadísticos que posibiliten generar una representación vectorial de los discursos a favor y en contra de la ley de la interrupción voluntaria del embarazo y entrenar con ella un modelo predictivo de clasificación.

1.2.2. Objetivos particulares

En términos particulares, los objetivos de este trabajo son:

- Obtener un *corpus* de datos con discursos a favor y en contra de la legislación del aborto sobre el cual se puedan aplicar las técnicas de vectorización y, luego, entrenar un modelo de predicción.
- En los casos necesarios, implementar los métodos estadísticos detallados por Monroe *et al.* (2008) en un código ejecutable de *Python* que luego pueda ser aplicado al *corpus* de discursos.
- Entrenar un modelo predictivo de clasificación, como la regresión logística, que utilice distintos métdos de vectorización para predecir el voto relacionado a cada discurso emitido y, luego, evaluar su rendimiento al utilizar el método de vectoriazión seleccionado.
- A partir de utilizar el voto como verdad fundamental a predecir, comparar el rendimiento de los distintos métodos de vectorización a través de técnicas propias del aprendizaje automático, tales como la validación cruzada, y métricas igualmente acordes, como el *accuracy*, la precisión, la cobertura y el *score F1*.

1.3. Estructura del trabajo

A continuación, el lector podrá encontrar, en el apartado 2, una descripción sobre la descarga y obtención de los datos utilizados en este trabajo, como así también de las características generales que presenta. Las secciones 3 y 4 exponen, respectivamente, los métodos empleados y los resultados obtenidos a partir de su implementación. Ambas poseen la misma estructura: en primer

lugar hacen referencia al proceso de anotación automático con supervisión y corrección manual utilizado para pre-procesar el *corpus*; luego presentan la selección de rasgos para vectorizar los discursos y lo referido a la comparación de estos métodos, y finalmente se detienen en el entrenamiento de un modelo de clasificación (la regresión logística) y su correspondiente evaluación. La sección 5 concluye con algunas reflexiones finales y discusión al respecto. Hacia el final de este trabajo es posible encontrar el Anexo, donde se detallan las pautas de anotación adoptadas y tablas y gráficos adicionales.

2. Datos

2.1. Descarga y preprocesamiento

Para este trabajo se utilizó la versión taquigráfica de la sesión número 23 (reunión 28) celebrada por la Cámara de Senadores, que tuvo lugar entre los días 29 y 30 de diciembre de 2020 y en la que se abordó la regulación del acceso a la interrupción voluntaria del embarazo y a sus posteriores cuidados. Los datos fueron obtenidos de la página del Senado de la Nación Argentina (Senado de la Nación Argentina, 2020) utilizando el método de *scraping*.

Todo el código de programación implementado fue desarrollado en el lenguaje *Python 3*. En primer lugar, se descargó la transcripción en formato *PDF* y luego, por medio de la librería *pdfminer* (Shinyama, 2015), se la convirtió a texto plano de manera que fuese procesable. Dado que la librería empleada en la conversión no arrojó una transcripción limpia y que, además, no todo el texto resultó de relevancia para el presente trabajo, se realizó una tarea de preprocesamiento con el fin de limpiar y organizar los datos de forma útil a los objetivos aquí perseguidos.

Una vez convertida a *.txt*, la transcripción ya no presentó separación de páginas, por lo que los encabezados y pies se convirtieron en líneas de texto que interrumpían los discursos de los participantes del debate y debieron ser removidos. Posteriormente, se extrajo la sección del texto pertinente para este trabajo: la sección 6, dedicada a la "Regulación del acceso a la interrupción voluntaria del embarazo y a la atención postaborto". Otras secciones consistían en el izamineto de la bandera, la convocatoria a la sesión, la lectura de la ley resultante, entre otras cuestiones que no hacían al discurso argumentativos de los asistentes, sino más bien al protocolo, por lo que fueron desestimadas para este análisis.

Con el texto de interés delimitado y mediante el uso de patrones regulares, se identificó a los distintos oradores y sus respectivos discursos, como así también a las secciones que no pertenecían a fragmentos discursivos emitidos durante la discusión, sino a comentarios agregados por el taquígrafo[6]. Siguiendo la metodología descripta por Monroe *et al.* (2008), los datos fueron guardados

[6] "Luego de unos instantes" y "Contenido no inteligible" constituyen ejemplos de estos fragmentos.

en un archivo en formato *.xml*, en el que cada fragmento fue etiquetado con la siguiente información:

- `speech`: etiqueta que indica si el fragmento corresponde propiamente a un discurso emitido durante la discusión, en cuyo caso toma el valor `true`, o si consiste en un comentario del taquígrafo, ante lo cual toma el valor `false`.

- `speaker`: en el caso de que el contenido del texto refiera a un fragmento discursivo, esta etiqueta indica el nombre del orador que lo pronunció. Si el fragmento es un comentario del traquígrafo, la etiqueta recibe el valor `none`.

Adicionalmente, como resultado del preprocesamiento, se extrajo la lista de asistentes a la sesión en formato tabular, con el objetivo de poder enriquecerla con especificaciones como la filiación partidaria y la decisión de voto. En el caso de la filiación partidaria, se utilizó la librería *selenium* (Muthukadan, s.f.) para buscar, en la página del Senado destinada a tal fin (Senado de la Nación Argentina, s.f.), los senadores en ejercicio en la fecha en la que tuvo lugar la sesión. Esta librería permite una interacción dinámica con la página *web* consultada, por lo que posibilitó el acceso al motor de búsqueda del sitio del Senado, la selección del período deseado de forma automática y el procesamiento de los datos de interés, los cuales se encontraban en *.html* en la página pero que, luego de su extracción, fueron guardados en formato *.csv*.

La decisón de voto, en cambio, debió ser transcripta de forma manual. Si bien se encontraba detallada en el documento de la sesión descarcargado, no fue posible convertir la tabla del documento *.pdf* a texto plano sin pérdida de información. Por lo cual, tomando los nombres de los senadores previamente extraídos, se generó un archivo *.csv* en el cual se le agregó a cada uno el voto emitido.

Una vez obtenidos los distintos conjuntos de datos para el estudio en cuestión, fue necesario un paso ulterior de procesamiento a fin de que la información recopilada de diversas fuentes pudiera ser vinculada de manera satisfactoria. El desafío aquí residió en la discrepancia entre dichas fuentes para nombrar a los senadores. Al incio de la versión taquigráfica es posible encontrar una lista de todos los asistentes a la sesión del Senado, incluyendo no solo a los senadores con poder de voto sino también a quienes presidieron y oficiaron en la secretaría. Esta lista, utilizada para la confección de la tabla que vincula a cada senador con el voto emitido, enuncia a cada funcionario recurriendo primero al nombre y luego, al apellido (en los casos en los que la persona tiene más de un nombre y/o apellido, se hizo uso de todos ellos). Esta enunciación, sin embargo, es modificada a lo largo de la transcripción, donde la primera vez que cada interlocutor hace uso de la palabra se lo refiere con apellido y nombre (en ese orden) y luego, solo con el apellido (un solo apellido en los casos que no se prestan a confusión, o más de uno en caso contrario). Así también, utilizando primero el apellido y luego el nombre es como se hace referencia a los senadores en la página destinada a proporcionar su lugar de origen y filiación partidaria. Para lidiar con esta

diversidad en la enunciación, se realizó un proceso de estandarización híbrido en el cual, tomando como base los nombres y apellidos en un conjunto de datos, se exploró exhaustivamente su correspondencia con con aquellos presentes en otro. Para esto, se consideró cada cadena de nombre/s y apellido/s como un conjunto y se evaluó la existencia de coincidencia o de relación de subconjunto entre las distintas cadenas posibles. En la mayoría de los casos, este mecanismo permitió la desambiguación. Los pocos casos que no pudieron ser estandarizados de esta manera por darse relación entre más de dos conjuntos fueron desambiguados manualmente.

2.2. Descripción

La sesión analizada cuenta con un total de 70 senadores, pertenecientes a 27 partidos políticos, distribuidos del siguiente modo (figura 1): solo uno de los partidos (Frente de todos) cuenta con 12 senadores; también uno solo (Alianza frente para la victoria) es representado por 10 senadores; dos partidos (Alianza cambiemos y Juntos por el cambio) cuentan con 7 senadores, y el resto de los partidos tienen entre 3, 2 y un senador. En cuanto a las provincias (24 en total), todas tienen 3 senadores exceptuando a Tucumán y a La Rioja que tienen solo 2.

De los votos emitidos, se observa que 38 son positivos; 29, negativos; 2 son ausencias, y uno es una abstención. Como muestra la figura 2, la intención de voto no guarda una relación unívoca con los partidos a los cuales los senadores representan. A excepción de aquellos que cuentan con un único senador, la mayoría es representado por senadores que votaron a favor y senadores que votaron en contra de la ley para el acceso al aborto.

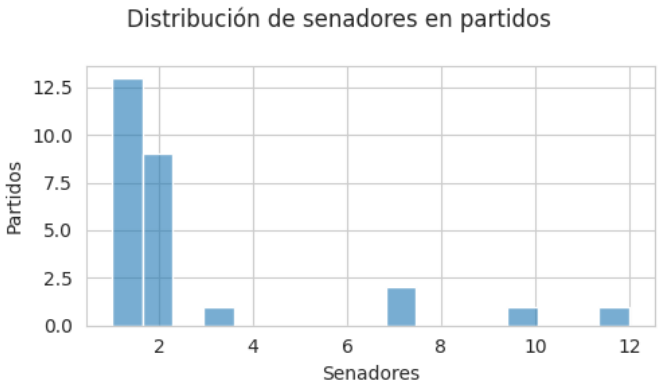

Figura 1: Distribución de senadores en partidos políticos.

Respecto de las intervenciones discursivas, se encuentran 111 discursos por parte de los senadores que votaron a favor de la ley; 88 discursos por parte de quienes votaron en contra; un discurso de quienes estuvieron ausentes durante la votación, y uno de quien se abstuvo de votar. Si se ponen en relación estos

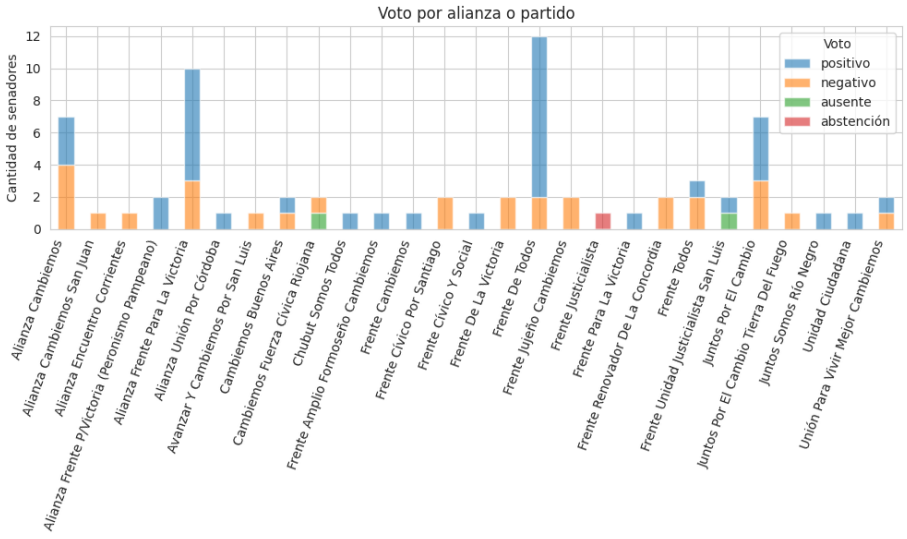

Figura 2: Distribución de votos en los distintos partidos políticos.

datos absolutos con la cantidad de senadores, se ve que, en promedio, se emitieron 2.87 discursos por senador, con un desvío estándar de 6.23 y una mediana de 1, coincidente además con la moda, lo que nos indicaría que se trata de una distribución asimétrica a derecha. Estas medidas de centralidad incluyen a 9 senadores que no intervinieron en la sesión. Al desagregar los datos según elección en la votación, es posible notar que las abstenciones no presentan dispersión y su media se ubica en el valor 1. Esto se debe a que solo un senador se abstuvo de votar y, a su vez, solo emitió un discurso. Una situación similar se da en las ausencias, donde también se observa un único discurso. Pero dado que dos senadores estuvieron ausentes, la media se ubica hacia el valor 0,5 y el desvío, hacia el 0,7. En cuanto a los votos negativos y positivos, ambos presentan una moda de 1, pero los negativos exhiben una media y un desvío estándar mayor que los positivos ($3{,}03 \pm 8{,}43$ *versus* $2{,}92 \pm 4{,}26$, respectivamente). La figura 3 muestra las distribuciones detalladas.

Como señalan Bird *et al.* (2009), las cadenas de caracteres "incluyen muchos detalles en los cuales no estamos interesados tales como espacios en blanco, saltos de línea y líneas blancas [...] Para nuestro procesamiento del lenguaje, queremos dividir estas cadenas en palabras y puntuación [...] A este paso se lo llama **tokenización** y produce una estructura que nos es familiar." [7] Siguiendo esta técnica, se procesaron los discursos y se calculó su longitud considerando solamente las palabras que los consituyen. Se observó que la distribución de la longigud, medida de este modo, varía dependiendo de si se mide los documentos en *tokens* totales o únicos. En promedio, los discursos emitidos tienen 418

[7] Cita traducida del inglés por la autora de este trabajo. El resaltado es del original.

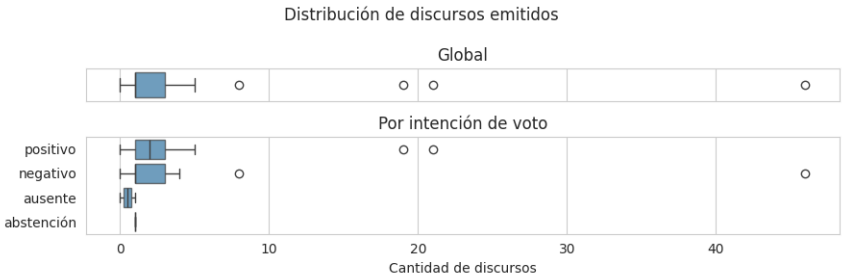

Figura 3: Distribución de votos en los distintos partidos políticos.

palabras, con un desvío estándar de 714; la mediana es de 11 y la moda, de 7. Sin embargo, estas medidas reflejan las palabras totales utilizadas en cada intervención, sin considerar si se repiten o no: cada ocurrencia de una palabra cuenta, sin importar si ya fue pronunciada en el mismo discurso. Es por eso que, a fines comparativos, se tomaron también las medidas de centralidad de los *tokens* únicos, las cuales mostraron una media de 161 palabras por discurso con un desvío de 245, una mediana de 11 y una moda de 2. La figura 4 refleja este contraste. Como es posible notar, si bien en ambos casos los discursos muestran valores atípicos respecto de su longitud, cuando esta se mide en palabras totales se observa una mayor cantidad con valores más extremos que al medirla en *tokens* únicos. En este último caso, un 20% de los registros son valores atípicos, de los cuales solo el 17% constituyen casos extremos, mientras que, al considerar el total de palabras, se observan un 22% y 32% respectivamente.[8]

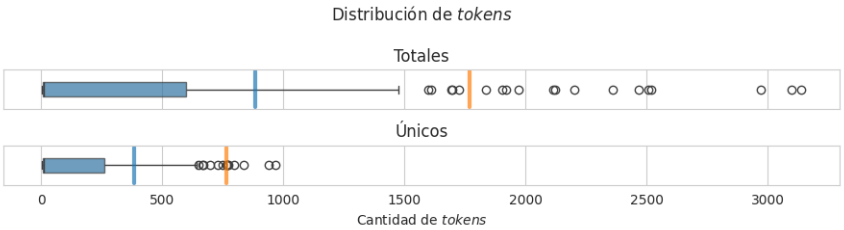

Figura 4: Distribución de *tokens* totales y únicos en los discursos pronunciados. En ambos gráficos, la línea azul indica el límite a partir del cual una observación se considera atípica leve (IQR∗1,5) y la naranja, el límite a partir del cual se la considera atípica extrema (IQR∗3).

[8]Para este análisis se utilizó el test de Tukey, que toma el rango intercuartil o IQR ($Q3$-$Q1$, donde $Q1$ refiere al primer cuartil y $Q3$, al terceo) y considera valores atípicos leves a aquellos que se encuentran entre $Q1 - (1{,}5 * IQR) < x < Q3 - (1{,}5 * IQR)$ y, como extremos a los que están entre $Q1 - (3 * IQR) < x < Q3 - (3 * IQR)$.

3. Metodología

3.1. Anotación

Dado que la intención de este trabajo consiste en utilizar los datos textuales obtenidos de la sesión del Congreso en cuestión para realizar una selección de rasgos a fin de entrenar un modelo de clasificación, resulta crucial procesar dichos datos de forma tal que el proceso de extracción de información por parte de un algoritmo de aprendizaje automático se vea faciliado. En concreto, palabras como 'aceptaron', 'acepta', 'aceptó' se vinculan gramatical y semánticamente: las tres pertenecen a la categoría de los verbos y refieren a la misma acción ('aceptar'). Sin embargo, a la hora de extraer los rasgos a partir de los discursos prounciados, estas palabras serán consideradas como tres rasgos distintos si no se las convierte previamente a una forma única o base.

Como indican Bird *et al.* (2009), "la **lematización** es el proceso que mapea las distintas formas de una palabra (como 'apareció', 'aparece') a su forma canónica o de entrada de diccionario, también conocida como lema". [9] Por otro lado, el ***stemizado*** es el proceso por el cual se le remueven los sufijos a una palabra, persistiendo solamente su primer o primeros morfemas y utilizando esto como forma base.

En este trabajo se exploraron ambas opciones de procesamiento. Cuestiones que se tuvieron en cuenta particularmente fueron la preservación de la marcación de género, puesto que no resulta menor respecto del tema en consideración, y la minimización en la cantidad de formas a las cuales se transformaron los verbos. Para el *stemizado* se utilizó la libería *NLTK* (Bird *et al.*, 2009), la cual proporciona distintos algoritmos posibles. Aquí, se optó por la implementación de *SnowballStemmer*[10], que permite elegir la lengua sobre la cual se quiere trabajar. Respecto del lematizado, se recurrió a la librería *spaCy* (Honnibal *et al.*, 2020), que cuenta con modelos pre-entrenados para la predicción de lemas en distintos idiomas. El modelo aquí utilizado es el *es-core-news-md*, entrenado con textos provenientes de noticias y medios de comunicación[11].

En lo que refiere a la marcación de género, ninguno de los procedimientos se mostró superior por sobre el otro para los propósitos de este trabajo, dado que ambos remueven la marcación de este accidente, ya sea eliminando el sufijo que lo representa ('-o' y '-a' en la mayoría de los casos) como lo hace el *stemizado*, ya transformando la palabra completa a su forma canónica (en masculino singular, en el español) como lo hace el lematizado. En el caso de la conversión de los verbos a una forma base, en cambio, el lematizado permitió una mejor reducción, dado que posibilitó transformar las distintas ocurrencias de un verbo a una única forma: el infinitivo. Al ser un procedimiento que se centra únicamente en la remoción de los sufijos, el *stemizado* no permite lo mismo en el caso de

[9] Cita traducida del inglés por la autora de este trabajo. El resaltado es propio.

[10] Sobre la implementación de la librería, consultar la documentación en https://www.nltk.org/api/nltk.stem.SnowballStemmer.html [último acceso: 15–10–2024]. Y, para más información sobre el algoritmo en español, dirigirse a https://snowballstem.org/algorithms/spanish/stemmer.html [último acceso: 15–10–2024].

[11] https://spacy.io/models/es#es_core_news_md [último acceso: 15–10–2024].

una lengua como el español, que presenta una considerable variedad de verbos irregulares al nivel de la raíz: tras el *stemizado*, las distintas ocurrencias de un mismo verbo resultan reducidas a tantas formas como irregularidades presente su raíz. Ejemplo de esto es el verbo 'aprobar', que fue transformado en 'aprob', para los casos como 'aprobamos' o 'aprobaron' y 'aprueb', para aquellos como 'aprueba', 'apruebe'. Es por esto que, si bien ambos procesamientos requirieron una revisión y correcciómn ulteriores, se prefirió el lematizado por sobre el *stemizado* dado que permitió una transformación de los verbos más clara para el español.

En concreto, el flujo de trabajo consistió en procesar cada discurso emitido con los modelos pre-entrenados de la librería *spaCy*, los cuales permitieron predecir, para cada *token*, su lema y *POS tag*[12]. Luego, esta información fue volcada en una tabla con tres columnas: una para las ocurrencias reales de las palabras, otra para los lemas asignados y una tercera con las etiquetas predichas. En un paso siguiente, estas asignaciones fueron corregidas manualemente, y los lemas y etiquetas obtenidos fueron utilizados para la extracción de rasgos en el entrenamiento de modelos que se detalla a continuación. Los criterios de corrección utilizados pueden encontrarse en el Anexo A.1, y la sección 4.1 proporciona un breve análisis de los resultados de este proceso.

3.2. Selección de rasgos

Tomando como referencia las técnicas estadísticas descriptas por Monroe *et al.* (2008), se implementaron vectorizadores que, a partir de recibir una lista de documentos, devolviesen una matriz donde cada fila representase un documento y cada columna, una palabra. Los valores reflejados en las celdas de esa matriz corresponden a la frecuencia absoluta de la palabra en el documento en cuestión. Lo que diferencia a las distintas matrices es el conjunto de palabras seleccionadas para ser usadas como componentes del vector que caracteriza a los documentos. Cada vectorizador utiliza una técnica estadística determinada para obtener las N palabras más representativas del grupo de votantes a favor y en contra de la legalización del aborto, donde N es el número entero positivo que indica la cantidad de palabras a considerar. Los detalles particulares de cada técnica pueden encontrarse en la sección 3.2.1.

Para la implementación se tomó como base el estimador `CountVectorizer` de la librería *scikit-learn*, de modo que las clases u objetos resultantes contasen con los mismos métodos y, luego, pudieran ser utilizadas en un flujo de trabajo definido con clases y métodos de la misma librería.

En primer lugar, se separaron los datos en un conjunto de entrenamiento y otro de testeo (80% y 20% de los datos, respectivamente). En esta división se tuvo en cuenta la variable objetivo `voto` de modo que los subconjuntos obtenidos

[12] El *POS tag*, por *Part of Speech*, o etiquetado gramatical consiste en la asignación a cada palabra de su categoría gramatical o, en algunos casos, sintáctica. Los modelos de la librería *spaCy* emplean el sistema de etiquetas de *Universal Dependencies*. Para más información, visitar https://spacy.io/usage/linguistic-features#pos-tagging [último acceso: 15–10–2024] y https://universaldependencies.org/u/pos/ [último acceso: 15–10–2024].

reflejasen la misma proporción de votos a favor y en contra. Asimismo, se usó una semilla para garantizar que los resultados fuesen replicables. La variable objetivo, además, fue codificada a valores numéricos para poder ser utilizada por los algoritmos de predicción.

Luego, tomando solamente el conjunto de entrenamiento y vectorizando previamente los datos con cada uno de los vectorizadores desarrollados, se entrenó una regresión logística utilizando los parámetros por defecto de la librería *scikit-learn* [13]. En cada entrenamiento se aplicó el método de validación cruzada con 5 subconjuntos: 4 para entrenamiento y 1 para testeo en cada iteración. Para esto, al igual que en la primera división, se cuidó de realizar una segmentación estratificada, utilizar una semilla y aplicar la misma división en cada vectorizador, de modo que las circunstancias de evaluación fuesen lo más semejantes posible y eso permitiese comparar qué vectorizadores conducen a un mejor entrenamiento. La tabla 1 resume las combinaciones testeadas en la experimentación. En todos los casos se generaron vectores de trescientas dimensiones.

Técnica	*Remoción de stopwords*	*Log*	Suavizado
Frecuencias absolutas	No	No	No
Proporciones	No	No	No
Proporciones	Sí (NLTK)	No	No
Proporciones	Sí (Zipf)	No	No
Ratio de *odds*	No	No	No
Ratio de *odds*	No	Sí (*odds*)	No
Ratio de *odds*	No	Sí (*odds*)	Sí (0,5)
TF-IDF	No	No	No
TF-IDF	No	Sí (*IDF*)	No
Word scores	No	No	No

Tabla 1: Técnicas estadísticas testeadas para la vectorización de los discursos utilizados en el entrenamiento del modelo de Regresión Logísitca.

3.2.1. Métodos estadísticos utilizados en la vectorización de documentos

De los métodos aplicados, dos (*TF-IDF* y *Word Scores*) consisten en algoritmos para la asignación de puntuaciones o *scores* a las palabras dentro de un *corpus*. El resto de las técnicas emplean métricas estadísticas sencillas. Para la aplicación de estos procedimientos se utilizó como *token* las lematizaciones de las palabras incluyendo su categoría gramatical[14].

[13] Entre los más relevantes para este trabajo, destacamos: regularización o `penalty=l2`, intensidad inversa de la regularización o `C=1`, ajuste de *bias* o `fit_intercept=True` y peso de cada clases o `class_weight=None`. Para más información sobre los parámetros por defecto, consultar: https://scikit-learn.org/1.5/modules/generated/sklearn.linear_model.LogisticRegression.html [último acceso: 15–10–2024]

[14] Para esto, todos los documentos fueron pre-procesados de modo tal que cada palabra fue reemplazada por un *token* construido de la siguiente forma: ⟨*lema*⟩_⟨*POS tag*⟩, donde ⟨*lema*⟩

Diferencia de frecuencias absolutas Este método utiliza la diferencia de frecuencias absolutas de los *tokens* entre los discursos de ambos grupos para extraer el conjunto más representativo de cada uno:

$$y_t = y_t^{(P)} - y_t^{(N)} \tag{1}$$

donde $y_t^{(X)}$ refiere a la cantidad de ocurrencias absolutas del *token* t en los discursos que votaron de manera positiva (si $X = P$) y negativa (cuando $X = N$). En los casos en los que y_t sea mayor o igual a 0, el *token* t será característico del discurso de los senadores que votaron a favor de la legalización. En caso contrario, será característico de quienes votaron en contra. De este modo, se seleccionaron las trescientas dimensiones más representativas de ambos grupos cuidando que hubiese igual cantidad de *tokens* para cada uno. Un procedimiento análogo es aplicado en el resto de las técnicas pero utilizando un criterio de contrastación distinto en cada caso.

Diferencia de proporciones El segundo contraste utiliza la proporción de *tokens* en cada conjunto de discursos.

$$f_t = f_t^{(P)} - f_t^{(N)} \tag{2}$$

donde $f_t^{(X)}$ se define como $y_t^{(X)}/n^{(X)}$, siendo $y_t^{(X)}$ la frecuencia absoluta del *token* t en el cojunto los discursos de X y $n^{(X)}$, la cantidad de *tokens* totales en esos discursos.

Remoción de *stopwords* También se intentó calcular la diferencia de proporciones removiendo previamente los *tokens* considerados *stopwords*. Para esto se probaron dos alternativas. Por un lado, se utilizó el conjunto predefinido de la librería *NLTK* [15] y, por otro, la Ley de Zipf. Este último procedimiento consiste en calcular la frecuencia absoluta de aparición de cada *token* y, en virtud de ella, establecer un *ranking* ordenándolos de manera decreciente. Luego, se contrasta dicha frecuencia con el orden asignado y se define un umbral a partir del cual una palabra es considerada *stopword*. Resulta pertinente notar que, dado que la frecuencia absoluta de aparición de una palabra se determina en relación al conjunto de documentos en el cual esta frecuencia es medida, las diferentes iteraciones de la validación cruzada generaron distintos conjuntos de *stopwrods*. En todas las iteraciones se removieron 100 *tokens* seleccionados con este método. Este número fue seleccionado tras realizar un análisis del conjunto de datos completo como primera aproximación general (ver sección A.3.1.).

Ratio* de *Odds Se utilizan *odds* de ambos conjuntos de discursos como técnica de selección. En este caso, los *tokens* más característicos no resultan de una diferencia sino de un cociente:

refiere al lema de la palabra y $\langle POS\ tag \rangle$, a su etiqueta *POS*.

[15] https://raw.githubusercontent.com/nltk/nltk_data/gh-pages/packages/corpora/stopwords.zip [último acceso: 15–10–2024]

$$O_t = \frac{O_t^{(P)}}{O_t^{(N)}} \tag{3}$$

aquí, $O_t^{(X)}$ consiste en $f_t^{(X)}/(1-f_t^{(X)})$. Otra diferencia entre esta técnica y el resto es que aquí el punto de corte para considerar a un *token* como característico de determinado grupo de discursos no es el 0 sino el 1, por la misma definición de la función.

Ratio Log-odds Sobre los *odds*, también se calculó el *ratio log-odds*. Para este cálculo se utilizó el logaritmo natural: $\ln O_t$. Asimismo, se realizó un suavizado sobre las frecuencias en los casos en los que los *tokens* solo se encontraban en uno de los grupos discursivos. Dicho suavizado consistió en agregar un peso predefinido a las frecuencias relativas, $\tilde{f}_w^i = f_w^i + \epsilon$, y tomar este valor en el cálculo del *ratio*. Siguiendo a Monroe *et al.* (2008), se utilizó $\epsilon = 0{,}5$.

TF-IDF Si bien la librería *scikit-learn* cuenta con un vectorizador que permite calcular los *scores* de *TF-IDF* [16], dado que la fórmula propuesta por Monroe *et al.* plantea algunas modificaciones a dicha implementación, se decidió desarrollar un vectorizador propio para obtener un resultado más aproximado al descripto por los autores, quienes indican haber usado dos fórmulas diferentes: ambas emplean la frecuencia de término natural y sin normalizar, pero una con la frecuencia de documentos con logaritmo (fórmula 4) y la otra, con la frecuencia de documentos natural (fórmula 5).

$$tf.idf_t^{(X)}(ntn) = f_t^{(X)} \times \ln\left(\frac{1}{df_t}\right) \tag{4}$$

$$tf.idf_t^{(X)}(nnn) = \frac{f_t^{(X)}}{df_t} \tag{5}$$

donde f_t^X refiere a la proporción del *token* t en el discurso perteneciente a X (con $X \in \{P, N\}$) y df_t, a la cantidad de documentos totales en los que ese *token* aparece.

Lo que los autores no mencionan es qué tipo de comparación realizan entre los *tokens* pertenecientes a los distintos discursos. Siguiendo los procedimientos anteriores, en este trabajo se optó por usar la sustracción, por lo que:

$$tf.idf_t = tf.idf_t^P - tf.idf_t^N$$

[16] Para más información sobre las fórmulas implementadas por *scikit-learn*, visitar: https://scikit-learn.org/stable/modules/feature_extraction.html#tfidf-term-weighting [último acceso: 15–10–2024].

Word Scores Por último, este procedimiento le asigna un peso a los *tokens* a partir del siguiente cálculo:

$$W_t^{*(P-N)} = \frac{y_t^{(P)}/n^{(P)} - y_t^{(N)}/n^{(N)}}{y_t^{(P)}/n^{(P)} + y_t^{(N)}/n^{(N)}} n_t$$

donde $y_t^{(X)}$ refiere a la frecuencia absoluta del grupo X y $n^{(X)}$, a la cantidad de *tokens* totales en X.

3.3. Entrenamiento y evaluación

Una vez elegido el vectorizador a utilizar (*TF-IDF* tomando logaritmo de la frecuencia inversa en documentos), se lo empleó para vectorizar el *corpus* y, con los datos representados numéricamente, se realizo una selección de hiperparámetros para entrenar el modelo de regresión logística[17]. Utilizando el método de búsqueda exhaustiva, se exploraron distintas combinaciones que afectan a al peso asignado a las clases a predecir y la regularización, un procedimiento que permite penalizar el aprendizaje de modo tal que el modelo no resulte sobreajustado a los datos de entrenamiento y ello dificulte las correctas predicciones sobre datos no vistos. Las alternativas evaluadas consisten en:

- Peso de las clases (*class_weight*): sin peso; pesos inversamente proporcionales a los observados en los datos de entrenamiento; pesos fijos calculados a partir de los datos de ese mismo *dataset*.
- Regularización (*penalty*): sin regularización; regularización *L1* o de *Lasso*; regularización *L2* o de *Ridge*.
- Intensidad inversa de la regularización (C)[18]: 0,1; 0,5; 2; 1.

Durante la búsqueda exhaustiva se realizó una validación cruzada del mismo modo que al seleccionar el vectorizador: colocando una semilla para que los resultados fuesen replicables, con una segmentación del conjunto de datos estratificada según la variables objetivo y con cinco *folds* o iteraciones.

Luego, con los mejores hiperparámetros hallados, se entrenó un nuevo modelo con el conjunto de entrenamiento completo, reservando solamente el conjunto de testeo separado al principio del trabajo para realizar la evaluación final.

[17] En este procesamiento solamente se empleó el vectorizador elegido.

[18] La implementación de *scikit-learn* para la regresión logística no emplea el parámetro λ sino el parámetro C, el cual refiere a la intensidad inversa de la regularización y puede calcularse mediante la fórmula $\frac{1}{C}$, por lo que valores más pequeños conllevan a una regularización más fuerte y valores más grandes, a una más débil.

4. Resultados

4.1. Anotación

Luego del proceso de anotación y corrección de lemas y etiquetas gramaticales, se extrajeron algunas métricas de resumen y se realizaron gráficos con el fin de comprender qué distribución presentaban dichas etiquetas utilizadas y cuántos de los datos etiquetdados de forma automática requirieron alguna corrección, ya fuera en el lema asignado, ya en la etiqueta gramatical predicha.

En total, se revisaron 8324 anotaciones[19]. De ellas, se despreciaron 211 cadenas de caracteres (1,86%) [20]. De las palabras que se persistieron para el análisis, el 20,4% requirió una corrección en el lema predicho. De estas correcciones, el 27,8% de los casos corresponden a palabras para las cuales el modelo realizó diferentes predicciones en los distintos contextos de aparición y una de estas predicciones fue acertada. Estos casos se consideraron un error porque, si bien el modelo predijo al menos una vez el lema correcto, no lo hizo de forma consistente. El 72,2% de los errores restantes fueron casos en los que el modelo no logró predecir el lema correcto en ninguna situación. Respecto de las etiquetas *POS* se procedió con un análisis similar. El 12,35% de las palabras etiquetadas de manera automática requirió una corrección. De estos casos de error, el 71% de las palabras constituyen casos en los cuales el modelo predijo distintas etiquetas en los contextos observados y una de ellas resultó ser la adecuada. En el resto de los casos (29%) el modelo no fue capaz de identificar la etiqueta adecuada en ninguna de sus predicciones.

Al final del procedimiento se obtuvo un conjunto de 4889 *tokens* únicos, donde la unicidad hace referencia al lema y a la etiqueta gramatical asignada. En el análisis exploratorio realizado para la descripción de los datos, la unicidad de los *tokens* estaba dada por la igualdad en la secuencia de caracteres en la ocurrencia efectiva de la palabra: allí 'suma' se considera una sola vez, independientemente de si, en sus distintas ocurrencias, refiere al verbo (en la tercera persona singular del verbo 'sumar') o al nombre (*"acción y efecto de sumar"*[21]). Al agrupar los *tokens* por lema y etiqueta gramatical, estas dos ocurrencias son consideradas por separado, dado que se toma en cuenta 'suma (*NOUN*)' y 'sumar (*VERB*)' [22]. La figura 5 refleja el contraste en la longitud de discursos al ser medidos con ambos enfoques. Allí se puede observar que las longitudes medidas considerando lemas y etiquedas *POS* reflejan valores menores[23].

[19] Cada anotación está dada por la ocurrencia *cruda* de la palabra, la etiqueta (o etitquetas, si el modelo predijo más de una posible) de esa palabra en los discursos en los que fue vista y el lema (o lemas, si se predijo más de uno) para esa palabra en tales contextos.

[20] Remitirse a A.1 para un detalle de las palabras despreciadas.

[21] REAL ACADEMIA ESPAÑOLA.

[22] *'Noun'* y *'verb'* refieren a las clases 'nombre' (o 'sustantivo') y 'verbo', respectivamente. Otras clases frecuentes son *'adj'*, por 'adjetivo', y *'adv'*, por 'adverbio'. Para una lista completa de las posibles etiquetas, visitar: https://universaldependencies.org/u/pos/ [último acceso: 15–10–2024].

[23] Es preciso recordar aquí que, si bien el uso de etiquetas *POS* puede llevar a considerar como dos o más palabras lo que, en el análisis previo se consideraba como una palabra única, el uso de los lemas (en contraste con las palabras que reflejan acciedentes morfológicos) reduce

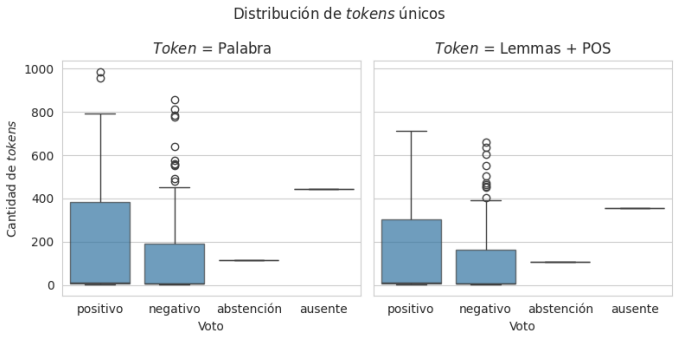

Figura 5: Distribución *tokens* únicos aplicando distintos criterios de unicidad. En la figura de la izquierda los *tokens* se miden en palabras y, en la de la derecha, a lemas con su correspondiente etiqueta *POS*. El eje de ordenadas (eje y) indica la cantidad de *tokens* en cada caso.

4.2. Selección de rasgos

Durante la validación cruzada realizada para evaluar el rendimiento de los distintos vectorizadores se fueron registrando varias métricas de evaluación para cada iteración como así también el tiempo requerido para el entrenamiento en cada caso. La figura 6 muestra el promedio y el desvío estándar de este último en las cinco iteraciones realizadas. Como es posible apreciar, el vectorizado de proporciones con remoción de *stopwords* provenientes de *NLTK* es el que presenta un tiempo de entrenamiento promedio menor, y también un menor desvío. El vectorizador de *TF-IDF* con frecuencia de documentos natural, por otro lado, es el que mayores valores exhibe tanto en la media como en el desvío estándar. La tabla 5 presente en la sección A.2.1 del Anexo detalla una a una las medidas de centralidad.

En cuanto a las métricas de rendimiento, la tabla 2 ofrece el promedio de los valores obtenidos en la validación cruzada para cada vectorizador. Allí se puede ver que, en la mayoría de los casos, fue el vectorizador de *TF-IDF* con logaritmo de la frecuencia de documentos el que presentó mejores resultados. Si bien el vectorizador de proporciones con remoción de *stopwords* obtenidas por el método de Zipf se muestra superior a este en términos de precisión, solo lo hace a expensas de perder cobertura, lo que se conoce como la compensación o el *trade-off* entre ambas métricas. De hecho, esta vectorización es la que presenta la peor cobertura de todos los métodos evaluados. Y algo semejante ocurre con el vectorizador basado en el *ratio* de los *log-odds* con suavizado, que muestra una mejor cobertura que el resto de los vectorizadores, pero una peor precisión, *accuracy* y *f1-macro*.

El vectorizador basado en *TF-IDF (log IDF)* no solo presenta un mayor *accuracy* que el resto de las técnicas de vectorización, sino que también muestra un mejor F_β *score* (con $\beta = 1$). Esta métrica ofrece una representación simétrica de la precisión y la cobertura, lo que nos indica que, si bien dicho vectorizador

en gran medida el vocabulario de los documentos.

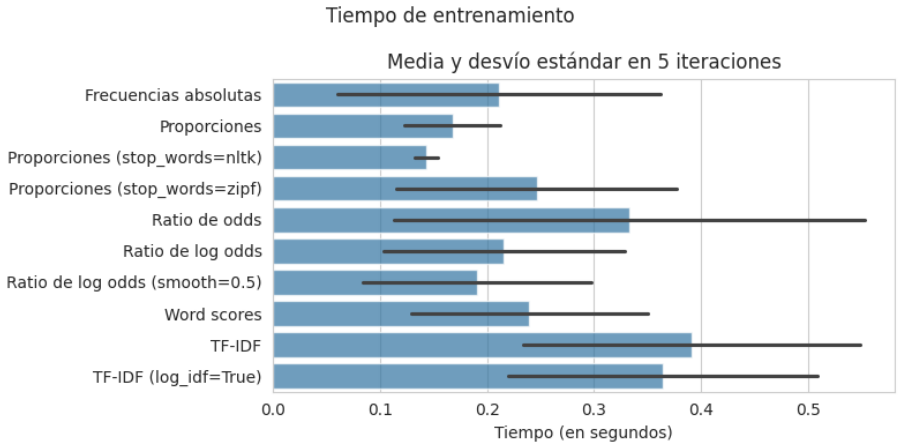

Figura 6: Media y desvió estándar del tiempo requerido (medido en segundos) para el entrenamiento de una regresión logística *baseline* al utilizar distintos vecotrizadores siguiendo una estrategia de validación cruzada con cinco iteraciones.

Vectorizador	*Accuracy*	Precisión	Cobertura	*F1*	*F1-weighted*	*F1-macro*
Frecuencias absolutas	0.610	0.663	0.618	0.635	0.608	0.605
Proporciones	0.623	0.663	0.653	0.654	0.622	0.617
Proporciones ($NLTK$)	0.623	0.663	0.653	0.654	0.622	0.617
Proporciones ($Zipf$)	0.636	0.726	0.576	0.619	0.624	0.625
Ratio de *odds*	0.560	0.587	0.698	0.611	0.518	0.506
Ratio de *log-odds*	0.560	0.587	0.698	0.611	0.451	0.506
Ratio de *log-odds* (suavizado)	0.541	0.555	0.887	0.682	0.518	0.419
TF-IDF	0.572	0.595	0.731	0.630	0.521	0.506
TF-IDF ($logIDF$)	0.667	0.699	0.721	0.702	0.663	0.657
Word scores	0.623	0.655	0.697	0.671	0.618	0.611

Tabla 2: Resultados obtenidos tras evaluar un modelo de regresión logística base utilizando vectorizadores basados en las distintas técnicas estadísticas. Los valores reflejan el rendimiento promedio de las cinco iteraciones de la validación cruzada. Las celdas resaltadas en azul corresponden a la estategia de vectorización que obtuvo un mejor rendimiento promedio en cada métrica de evaluación y las resaltadas en naranja, a la que obtuvo el peor rendimiento.

no es el que mejores valores exhibe en estas métricas, sí es el que mejor maneja la compensación entre ambas. En este trabajo se decidió utilizar $\beta = 1$ porque no se busca privilegiar ninguna de las dos por sobre la otra.

Adicionalmente, se reportan el *F1-macro* y *F1-weighted*. El primero consiste en un promedio del *F1* calculado para ambas clases predichas, lo que da una idea de la compensación de la precisión y la cobertura tomado en cuenta la clase 1 (discursos positivos), por un lado, y la clase 0 (discursos negativos), por otro. No obstante, dado que ambas clases no están balancedas[24], podría ser que, al calcular el promedio, el buen rendimiento en la predicción de una de las clases enmascare la imperfecta predicción de la otra. Es por esto que también se recurre al *F1-weighted*, que multiplica el *F1* de cada clase por la proporción de casos que esta presenta en el conjunto de datos, de modo que el valor resultante es una medida "pesada" en relación a la representación de cada clase. En la sección A.3.2 del Anexo pueden apreciarse los gráficos de estas métricas para cada iteración de la validación cruzada.

4.3. Entrenamiento y evaluación

Para la evaluación y comparación de los modelos entrenados con los distintos hiperparámetros se recurrió a las mismas métricas que las referidas en la sección 4.2 al evaluar los posibles vectorizadores.

En este caso, se encontró que todos los modelos entrenados con $penalty = None$ y $class\_weight = \{1 : 0{,}56, 0 : 0{,}44\}$[25] arrojaron los mejores valores promedios considerando las cinco iteraciones de la validación cruzada, sin importar el valor de C[26]. Las únicas excepciones a esto son el resultado en la métrica de precisión y de cobertura. En la primera, el modelo entrenado con $C = 0{,}1$, $penalty = l2$ y $class\_weight = balanced$ obtiene un mejor resultado. No obstante, este modelo muestra uno de lo valores más bajos de cobertura y *F1*. Es notable que, si bien tiene el mayor valor de precisión observado, esto no llega a compensar su bajo rendimiento en la cobertura y de ahí que también exhiba un valor bajo de *F1*. Y lo inverso sucede con la métrica de cobertura, en la que el modelo entrenado con $C = 0{,}1$, $class\_weight = \{1 : 0{,}56, 0 : 0{,}44\}$ y $penalty = l2$ consigue el mejor rendimiento a costas de tener la peor precición observada. Con valores más moderados, los demás modelos logran un mejor balance entre ambas métricas (ver tabla 3).

Adicionalmente, se graficó la media, el desvío estándar y el valor obtenido en cada iteración de la métrica *F1* para los mejores 4 conjuntos de hiperparámetros (ver figura 6)[27]. Aquí podemos observar que los hiperparámetros $C = 1$, $penalty = None$ y $class\_weight = \{1 : 0{,}56, 0 : 0{,}44\}$ presentan una media mayor y desvío menor que los otros conjuntos, lo cual es consistente con el gráfico

[24] Como se mencionó en la sección 2.2, el conjunto de datos presenta un 56% de discursos a favor y un 44% de discursos en contra.

[25] Los valores reportados en el parámetros `class_weight` corresponden a las proporciones de cada clase halladas en el conjunto de datos.

[26] Lógicamente, el no realizar ninguna penalización, este parámetro no tiene nigún efecto. La clase proporcionada por la librería *scikit-learn*, sin embargo, no permite fijar este valor en `None` en ningún caso. Para un detalle más fiel de los resultados obtenidos, la tabla 3 lista el conjunto completo de hiperparámetros que fueron testeados.

[27] Dado que cuatro modelos presentaron el mismo rendimiento, se graficó uno solo de ellos para facilitar la visualización.

Parámetros	*Accuracy*	Precisión	Cobertura	*F1*	*F1-macro*	*F1-weighted*
$C = 0{,}1, class\_weight = None, penalty = l2$	0.698	0.709	0.786	0.743	0.686	0.693
$C = 0{,}1, class\_weight = balanced, penalty = None$	0.710	0.733	0.762	0.742	0.701	0.706
$C = 0{,}1, class\_weight = balanced, penalty = l2$	0.679	0.786	0.595	0.674	0.678	0.678
$C = 0{,}1, class\_weight = \{1 : 0{,}56, 0 : 0{,}44\}, penalty = None$	0.717	0.728	0.797	0.759	0.706	0.712
$C = 0{,}1, class\_weight = \{1 : 0{,}56, 0 : 0{,}44\}, penalty = l2$	0.616	0.607	0.898	0.723	0.541	0.563
$C = 0{,}5, class\_weight = None, penalty = None$	0.710	0.739	0.752	0.741	0.703	0.708
$C = 0{,}5, class\_weight = None, penalty = l2$	0.648	0.715	0.628	0.665	0.644	0.647
$C = 0{,}5, class\_weight = balanced, penalty = None$	0.710	0.733	0.762	0.742	0.701	0.706
$C = 0{,}5, class\_weight = balanced, penalty = l2$	0.641	0.749	0.549	0.628	0.638	0.637
$C = 0{,}5, class\_weight = \{1 : 0{,}56, 0 : 0{,}44\}, penalty = None$	0.717	0.728	0.797	0.759	0.706	0.712
$C = 0{,}5, class\_weight = \{1 : 0{,}56, 0 : 0{,}44\}, penalty = l2$	0.679	0.682	0.808	0.735	0.657	0.667
$C = 1, class\_weight = None, penalty = None$	0.710	0.739	0.752	0.741	0.703	0.708
$C = 1, class\_weight = None, penalty = l2$	0.622	0.695	0.583	0.631	0.620	0.622
$C = 1, class\_weight = balanced, penalty = None$	0.710	0.733	0.762	0.742	0.701	0.706
$C = 1, class\_weight = balanced, penalty = l2$	0.622	0.739	0.505	0.598	0.620	0.617
$C = 1, class\_weight = \{1 : 0{,}56, 0 : 0{,}44\}, penalty = None$	0.717	0.728	0.797	0.759	0.706	0.712
$C = 1, class\_weight = \{1 : 0{,}56, 0 : 0{,}44\}, penalty = l2$	0.666	0.688	0.740	0.708	0.654	0.661
$C = 2, class\_weight = None, penalty = None$	0.710	0.739	0.752	0.741	0.703	0.708
$C = 2, class\_weight = None, penalty = l2$	0.629	0.717	0.560	0.624	0.626	0.626
$C = 2, class\_weight = balanced, penalty = None$	0.710	0.733	0.762	0.742	0.701	0.706
$C = 2, class\_weight = balanced, penalty = l2$	0.629	0.734	0.527	0.611	0.627	0.625
$C = 2, class\_weight = \{1 : 0{,}56, 0 : 0{,}44\}, penalty = None$	0.717	0.728	0.797	0.759	0.706	0.712
$C = 2, class\_weight = \{1 : 0{,}56, 0 : 0{,}44\}, penalty = l2$	0.647	0.689	0.673	0.678	0.641	0.646

Tabla 3: Resultados obtenidos tras evaluar un modelo de regresión logística con distintos hiperparámetros. Los valores reflejan el rendimiento promedio de las cinco iteraciones de la validación cruzada. Las celdas resaltadas en azul corresponden a conjunto de hiperparámetros que obtuvo un mejor rendimiento promedio en cada métrica de evaluación y las resaltadas en naranja, al que obtuvo el peor rendimiento.

que muestra los valores por iteración: si bien el modelo entrenado con estos hiperparámetros no logra superar en ninguna iteración al resto, tampoco exhibe una variación tan grande y, en ningún caso, posee el peor valor obtenido.

A la vista de los resultados obtenidos y dado que la librería *scikit-learn* toma por defecto $C = 1$, se decide usar el conjunto de hiperparámetros $C = 1$, $penalty = None$ y $class\_weight = \{1 : 0{,}56, 0 : 0{,}44\}$ para entrenar el modelo final. Para esta instancia, se usa el conjunto completo de datos de entrenamiento y luego se lo evalúa con los datos de testeo separados para este fin. La tabla 4 muestra los resultados obtenidos en esta evaluación y la figura 8, la matriz de confusión.

Se observa que el modelo final obtuvo un mejor *accuracy* que sus versiones pre-entrenadas durante la validación cruzada. Posiblemente esto se deba al incremento de los datos durante el entrenamiento. Mientras que, en la selección de hiperparámetros, se entrenó con 127 discursos y se testeó con 32 en cada iteración; en la versión final, el modelo fue entrenado con 159 documentos y evaluado con 40. Este incremento no solo influye en la cantidad y variabilidad de los datos que ve el modelo de regresión sino también en el vectorizador, que se ve provisto de una mayor y más diversa fuente de información para pesar las palabras presentes en los discursos y seleccionar las 300 más representativas de ambas clases a predecir.

Al comparar las métricas de cobertura y precisión en la tabla 4, se puede inferir que el modelo posee una leve tendencia a predecir mayormente discursos 'a favor': la cobertura de esta clase es mayor, pero su precisión es menor; lo que indica que el modelo predice mayormente esta clase aunque no en todos los

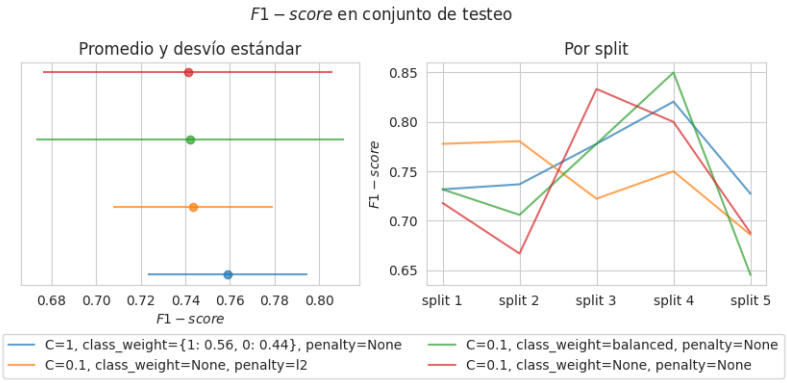

Figura 7: Resultados obtenidos al evaluar los distintos modelos con *F1* durante la validación cruzada. El gráfico a izquierda muestra el promedio y el desvío estándar de las cinco iteraciones para cada conjunto de hiperparámetros y el gráfico de la derecha compara los valores obtenidos en cada iteración.

casos acierta. Lo inverso ocurre con los documentos 'en contra'. Este fenómeno podría estar influido por el hecho de tener clases levemente desbalanceadas y que es la clase 'a favor' la mayoritaria, pese a que entre los hiperparámetros seleccionados, se intentó mitigar este desbalanceo.

Por último, se graficaron las 50 palabras asociadas a los coeficientes positivos y las 50 asociadas a coeficientes negativos con mayor peso, aquellas que más fuertemente conducen a predicciones de la clase 'a favor' y 'en contra' respectivamente. La figura 9 nos permite conjeturar qué *tokens* caracterizan los discursos pertenecientes a ambas clases. Por el lado de los discursos 'a favor' se encuentran con *tokens* como 'mujer_noun', 'embarazo_noun', 'sociedad_noun', 'salud_noun', 'aborto_noun', 'interrupción_noun', 'decidir_verb', 'permitir_voto', 'derecho_noun', 'autonomía_noun', 'respetar_verb', 'integral_adj', 'muerte_noun', 'voluntaria_adj'. Y, por el lado de los documentos 'en contra', se ven 'madre_noun', 'vida_noun', 'ley_noun', 'norma_noun', 'constitución_noun', 'concepción_noun', 'nacer_verb', 'humano_adj', 'problema_noun', 'niña_noun. Esta selección no se pretende exhaustiva sino que intenta recuperar algunas de las palabras con contenido semántico que aparecen asociadas a coeficientes mayores y, de allí, que puedan presumirse representativas de cada clase.

5. Discusión y conclusiones

Como parte de la vida política de una nación, los ciudadanos y agrupaciones sociales tienen la posibilidad de presentar proyectos de ley ante los legisladores. En caso de contar con un aval suficiente, estos proyectos pueden ser abordados en las cámaras de diputados y senadores y llegar a constituirse como ley efectivamente. Acorde a lo estipulado en la ley 27.275, que garantiza el acceso a la información pública, las sesiones de la Cámara de Sendores son transcriptas y

Clase	Precisión	Cobertura	*F1*	*Soporte*
0 ('en contra')	0.93	0.78	0.85	18
1 ('a favor')	0.84	0.95	0.89	22
Accuray	–	–	0.78	40
Promedio *macro*	0.89	0.87	0.88	40
Promedio *weighted*	0.88	0.88	0.87	40

Tabla 4: Métricas de evaluación.

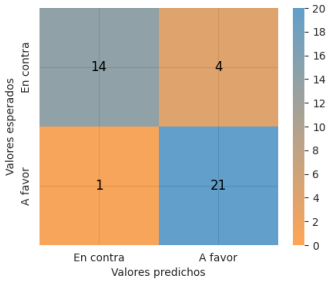

Figura 8: Matriz de confusión.

Resultados obtenidos a partir de entrenar un modelo de regresión logística con los mejores hiperparámetros hallados ($C = 1$, $penalty = None$ y $class\_weight = \{1 : 0{,}56, 0 : 0{,}44\}$) y el conjunto completo de datos de entrenamiento, y luego evaluarlo con el *dataset* de testeo.

disponibilizadas en la página del Senado de la Nación.

En el trabajo llevado a cabo, se utilizó el lenguaje de programación *Python 3* para acceder, mediante la técnica *scraping*, a los datos proporcionados por el Estado. Se los ha pre-procesado de manera que la información proveniente de distintas fuentes pudiese ser relacionada, y se los ha etiquetado siguiendo un procedimiento híbrido que comprendió el uso de modelos pre-entrnados por librerías de código abierto y la correción y anotación manual. Estos datos fueron utilizados luego para entrenar modelos de regresión logística con distintos vectorizadores que se desarrollaron para para este trabajo y cuya implementación estuvo basada en aquellos reseñados por Monroe *et al.* (2008). Dichos vectorizadores fueron contrastados respecto de su rendimiento y el mejor fue utilizado para entrenar un modelo final de regresión logística, para el que previamente fueron seleccionados los mejores hiperparámetros.

Durante el proceso de anotación, el modelo pre-entrenado, utilizado para predecir etiquetas *POS* y lemas, presentó un mejor desempeño en la predicción de las etiquetas gramaticales que de los lemas. Por un lado, fue necesario corregir manualmente una mayor cantidad de lemas que de etiquetas *POS*. Pero además, respecto de las predicciones erróneas observadas en cada caso, se halló un mayor porcentaje de acierto ocasional en la predicción de etiquetas *POS* que en la de los lemas. Aquí el acierto ocasional refiere a la observación de que, en distintos contextos discursivos, el modelo realizó diferentes predicciones, ya de etiqueta gramatical, ya de lema, para una misma palabra y en algunos de esos casos acertó; sin embargo, no fue consistente.

Por limitaciones de tiempo y recursos disponibles para el etiquetado, se decidió simplificar casos como verbos reflexivos y clíticos. En estas situaciones los verbos fueron transcriptos en infnitivos y las marcas reflexivas y de cliticidad fueron omitidas. Una posible mejora a este procedimiento consiste en repensar si esas decisiones fueron las mejores y qué impacto tienen sobre los datos (cuántos verbos se ven afectados y en qué medida). Del mismo modo, podría ser

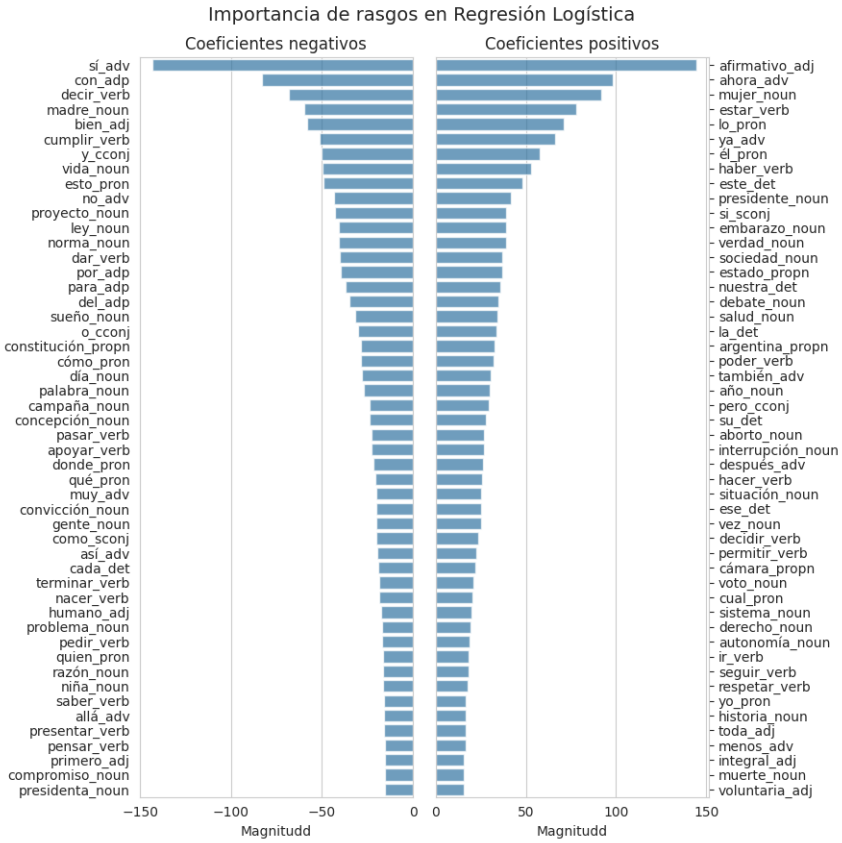

Figura 9: Palabras asociadas a los coeficientes positivos y negativos con mayor peso.

de interés contar con más de una anotación que permitiera validar de forma más robusta la calidad de las etiquetas obtenidas. Para esto, lo óptimo sería contar con anotaciones humanas que siguieran un protocolo estandarizado[28], dado que esto permite un mejor control de los criterios de las etiquetas a utilizar y la fiabilidad del procedimiento. Sin embargo, contar con anotaciones provenientes de distintos algoritmos de etiquetado también puede ser provechoso, y los resultados podrían ser usados para comparar estos algoritmos.

Respecto de la vectorización, la figura 6 muestra que los vectorizadores basados en *ratio* de *odds* y en *TF-IDF* son los que mayor tiempo de entrenamiento requirieron. No obstante, el encontrarse todos los vectorizadores en un rango de tiempo de ajuste muy similar y, además, casi despreciable, este dato no constituyó una variable de peso a la hora de seleccionar el mejor vectorizador. Puesto

[28]Las guías de anotación elaboradas en este trabajo persiguen un propósito semejante. Si bien aquí las anotaciones fueron realizadas por una sola persona, estas guías proporcionaron uns forma de fijar reglas que pudieran ser recordadas en las distintas sesiones de anotación.

que fue la implementación basada en *TF-IDF* con logaritmo sobre la frecuencia inversa en los documentos la que mejores resultados mostró respecto de las métricas de evaluación consideradas, se optó por esta vectorizador para el modelo final. Cabe destacar que, si bien sus resultads respecto de la precisión y la cobertura no fueron los óptimos, sí mostró un mejor balance entre estas dos medidas, tanto en el promedio general, como en el balance por clase (ver tabla 2).

Para la evaluación de los vectorizadores, se fijó en 300 el número de dimensiones a generar, que no es otra cosa que la cantidad de palabras consideradas como relevantes para caracterizar ambos grupos de discursos a predecir. Podría ser se interés contrastar si se observan los mismos resultados al disminuir o incrementar el número de dimensiones. Asimismo, en la implementación desarrollada, las técnicas estadísticas reseñadas por Monroe *et al.* (2008) se emplean solamente para la selección de palabras pero, una vez seleccionadas, todos los vectorizadores siguen el mismo procedimiento: cuentan la frecuencia absoluta de tales palabras en los documentos. En una futura iteración sobre este trabajo, se podrían implementar vectorizadores que no solo utilizasen tales técnicas estadísticas en la selección de palabras sino también en la asignación de pesos a los vectores resultantes.

Previo al entrenamiento del modelo de regresión logística, se realizó una validación cruzada para estimar los mejores hiperparámetros para el ajuste. Se encontró que cualquier conjunto de hiperparámetros que no implicase regularización ($penalty = None$) y utilizase la proporción de la clase objetivo en la estimación ($class\_weight = \{0 : 0{,}56, 1 : 0{,}44\}$) obtuvo, en promedio, los mejores resultados de *accuracy*, *F1-macro* y *F1*-pesado. El parámetro que define la intensidad inversa de la regularización a aplicar es desestimado puesto que no se aplica tal regularización. El modelo final entrenado mostró un mejor rendimiento que aquellos pre-entrenados durante la selección del vectorizador y los hiperparámetros, resultado que puede ser asociado al incremento en los datos durante el entrenamiento, puesto que esta versión fue ajustada utilizando todos los datos de entrenamiento y no con el 75% de los mismos, como se hizo previamente.

La experimentación realizada en este trabajo conduce así a elegir la técnica basada en *TF-IDF* con logaritmo sobre la frecuencia inversa de los documentos como la mejor para realizar una selección de rasgos que puedan ser utilizados al vectorizar los documentos. En futuros trabajos sería de interés contrastar los resultados aquí observados con los que pudieran obtenerse al seguir un experimento similar en otro tipo de modelos. La regresión logística constituye un modelo discriminativo, es decir que aprende rasgos que le permiten diferenciar entre las clases a predecir, aunque estos rasgos no sean necesariamente intrínsecos a las clases mismas. Los modelos generativos como Naïve Bayes, en cambio, estiman un modelo posible que podría haber dado lugar a las clases observadas y luego, ante una nueva observación, calculan las probabilidades de que las distintas clases puedan ser generadas por esa observación. Evaluar las técnicas aquí estudiadas con este tipo de modelos podría conducir a la obtención de resultados más robustos.

Por último, cabe notar que los datos utilizados en este trabajo no han sido voluminosos. Esto se debe a que, tras la descarga de la versión taquigráfica de la sesión, fue necesario un meticuloso procedimiento de limpieza y preprocesamiento que permitiese hacer uso de esos datos. Incluir otras sesiones hubiese implicado una labor mucho mayor y las limitaciones de tiempo y recursos no lo permitieron. Sin embargo, sería deseable aumentar, en trabajos venideros, los datos sobre los cuales se han desarrollados los procedimientos hasta aquí detallados, de modo que los resultados puedan verificarse en *corpora* que no solo sean mayores en tamaño sino que, además, aborden otras temáticas y, en ese sentido, presenten una mayor diversidad. La mejora en los resultados observados en el entrenamiento final constituyen un primer indicio del beneficio que tal aumento podría significar.

6. Bibliografía

Bird, Steven, Ewan Klein, y Edward Loper. 2009. *Natural language processing with python: analyzing text with the natural language toolkit*, capítulo 3. O'Reilly Media, Inc.

Campaña nacional por el derecho al aborto legal, seguro y gratuito. s.f. La lucha por el derecho al aborto. URL https://abortolegal.com.ar/historia/, [último acceso: 15–10–2024].

Fundación Huésped. s.f. Historia del aborto en argentina. URL https://huesped.org.ar/informacion/derechos-sexuales-y-reproductivos/tus-derechos/interrupcion-voluntaria-del-embarazo/historia-del-aborto-en-argentina/, [último acceso: 15–10–2024].

Honnibal, Matthew, Ines Montani, Sofie Van Landeghem, y Adriane Boyd. 2020. spacy: Industrial-strength natural language processing in python, zenodo, 2020.

Jurafsky, Daniel. 2000. *Speech and language processing*, capítulo 6. Prentice-Hall.

Ministerio de Justicia. s.f. Acceso a la información pública. URL https://www.argentina.gob.ar/justicia/derechofacil/leysimple/acceso-la-informacion-publica, [último acceso: 15–10–2024].

Monroe, Burt L., Michael P. Colaresi, y Kevin M. Quinn. 2008. Fightin'words: Lexical feature selection and evaluation for identifying the content of political conflict. *Political Analysis* 16:372–403.

Muthukadan, Baiju. s.f. Documentación oficial de selenium. URL https://selenium-python.readthedocs.io/index.html, [último acceso: 15–10–2024].

REAL ACADEMIA ESPAÑOLA. *Diccionario de la lengua española*. 23.ª edición, [versión 23.7 en línea]. URL https://dle.rae.es, [último acceso: 15–10–2024].

Senado de la Nación Argentina. 2020. *23ª sesión especial*. Período 138º, 28ª reunión. URL https://www.senado.gob.ar/parlamentario/sesiones/29-12-2020/28/downloadTac, [último acceso: 15–10–2024].

Senado de la Nación Argentina. s.f. *Buscador histórico de senadores*. URL https://www.senado.gob.ar/senadores/Historico/Fecha, [último acceso: 15–10–2024].

Shinyama, Yusuke. 2015. Pdfminer: Python pdf parser and analyzer. *Retrieved on* 11.

A. Anexo

A.1. Guía de anotación

Para todas las categorías de palabras se prefiere la lematización. A continuación se detallan las decisiones tomadas en casos particulares.

A.1.1. Conectores

Clases de palabra como preposiciones ('según'), pronombres relativos ('que') o conjunciones ('si') que suelen funcionar como conectores subordinantes son etiquetados con la categoría *SCONJ*.

A.1.2. Nombres

Por cuestiones de relevancia para el estudio, se mantiene la marca de género (femenino o masculino), pero se omite la marca de número (singular o plural). Ejemplo: 'abuelas' → 'abuela'.

A.1.3. Números

Se omiten todos, aquellos que aparecen en dígitos ('1', '10', etc.) y los que aparecen en caracters ('cuarenta').

A.1.4. Puntuación

Se omiten todas los signos etiquetados como signos de puntuación ($PUNCT$). Ejemplos: '¿', '?', ';'.

A.1.5. Verbos

- Se omiten los clíticos ('la', 'lo', 'le' y sus formas plurales). En caso de que el verbo presente una forma con clítico, se lo omite y se lo convierte a su forma infinitiva. Ejemplo: 'consierar*la*' → 'consierar'.
- Se omiten las formas reflexivas ('se'). En caso de que el verbo presente una forma reflexiva, se lo omite y se lo convierte a su forma infinitiva. Ejemplo: 'enfrentar*se*' → 'enfrentar'.

A.1.6. Verbos auxiliares

- La categoría AUX es despreciada.
- Aquellos verbos catalogados con la etiqueta AUX fueron recatalogados como verbos ($VERB$) si pertenecen a ese tipo de palabra, y transformados a la forma en infinitivo. Ejemplo: 'seamos' → 'ser'.

A.1.7. Palabras con tokenizados erróneos

Se omiten las cadenas de caracteres que al ser tokenizadas quedaron unidas a un signo de puntuación. Ejemplos: ‘¡negando’, ‘¿por’.

A.2. Tablas adicionales

A.2.1. Selección de vectorizador

Vectorizador	Media	Mediana	Desvío	Máximo
Frecuencias absolutas	0.190	0.153	0.063	0.284
Proporciones	0.198	0.144	0.128	0.428
Proporciones (*NLTK*)	0.142	0.142	0.009	0.155
Proporciones (*Zipf*)	0.249	0.170	0.147	0.449
Ratio de *log odds*	0.184	0.148	0.088	0.341
Ratio de *log odds* (suavizado)	0.286	0.258	0.141	0.470
Ratio de *odds*	0.189	0.148	0.094	0.355
TF-IDF	0.494	0.316	0.285	0.867
TF-IDF (*log IDF*)	0.422	0.289	0.199	0.644
Word scores	0.304	0.228	0.143	0.484

Tabla 5: Tiempos de ajuste observados tras vectorizar los discursos analizados con las distintas técnicas estadísticas y luego entrenar un modelo de regresión logística *baseline*. Las celdas resaltadas en azul corresponden a la estategia de vectorización que presentó el mejor tiempo de entrenamiento y las resaltadas en naranja, a la que obtuvo el peor tiempo.

A.3. Gráficos adicionales

Aquí pueden encontrarse gráficos adicionales sobre los métodos empleados en este trabajo.

A.3.1. Ley de Zipf

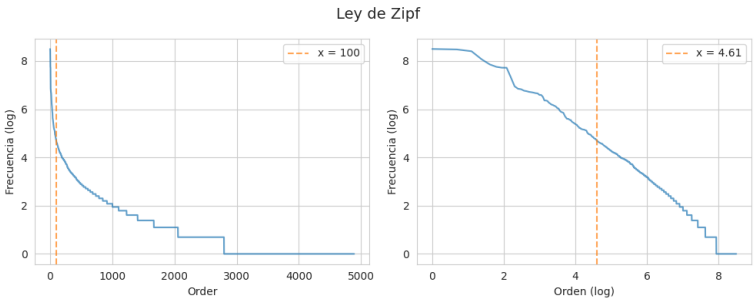

Figura 10: Contraste entre la frecuencia absoluta de cada palabra (en logaritmo) y el *ranking* de esa palabra según su frecuencia. El gráfico de la izquierda exhibe el orden natural de cada palabra y el de la derecha, su logaritmo. En ambos casos, la línea punteada naranja indica el umbral para considerar *stopword* a una palabra.

A.3.2. Selección de vectorizador

Figura 11: *F1*, *F1-macro* y *F1-weighted* por split de validación cruzada para el conjunto de entrenmiento y de testeo, para todos los vectorizadores en evaluación.