
Universidad Complutense de Madrid
Facultad de Informática

# Trabajo de Fin de Grado

*Minería de procesos aplicada al estudio de wikis*

Ignacio García Sánchez-Migallón

Director:
Javier Arroyo Gallardo

Madrid, Mayo, 2019

*A Javier Arroyo Gallardo, por su esfuerzo y dedicación, por hacer que este proyecto haya sido posible.*
*A mi familia, amigos, y a Helena, por su incansable apoyo.*

# Resumen

La creación colaborativa de conocimiento siempre ha sido uno de los pilares de internet desde la web 2.0. A priori, los intentos de crear contenido mediante la acción colectiva de diferentes individuos sin coordinación ni lucración personal debería ser inutil pues se produce un fenómeno conocido como *la tragedia de los comunes*. *La tragedia de los comunes* es una situación donde un sistema compuesto por usuarios que actúan independientemente para lograr su beneficio personal tienden a tener comportamientos contrarios al interés común. Sin embargo, se ha demostrado que esto no tiene porque ser cierto: la colaboración desinteresada y sin organización entre los diferentes usuarios ha hecho posible la existencia de comunidades cuyo único propósito es la difusión del conocimiento: las wikis.
Las wikis, compuestas por diferentes artículos, están siendo ampliamente estudiadas. Sin embargo, los procesos que determinan la evolución de su contenido e inherentes a su propio funcionamiento y aquelllos seguidos por los propios usuarios en su actividad no son del todo conocidos. En este proyecto se propone y aplica una serie de técnicas conocidas como (i) minería de procesos para descubrir y analizar estos procesos existentes en la labor de la escritura colaborativa tanto a nivel artículo como a nivel usuario así como (ii) técnicas de minería social para visualizar las estructuras de colaboración existentes entre los propios usuarios. Para esto se hará uso de la Wikipedia Española como referencia.
Con el objetivo de realizar este estudio se hará uso de los historiales de revisión con el que cuenta cada artículo de Wikipedia y una taxonomía de intenciones semánticas tras cada revisión compuesta de 13 categorías como contra vandalismo, refactorización o elaboración. Haciendo uso de un conjunto de datos dotado 5684 revisiones y sus intenciones semánticas se desarrolla un modelo predictivo que alcanza un valor de F1 micro de 0.64. Con este modelo y dichos historiales de revisiones se genera un corpus compuesto de diferentes *artículos destacados* y las intenciones tras cada una de sus revisiones. Con las revisiones en combinación con sus intenciones y el uso de minería social y de procesos se observa la estructura colaborativa de los usuarios, los procesos seguidos por los artículos así como los procesos seguidos por los propios usuarios en sus sesiones de edición.
Los resultados muestran que, aunque no existe un proceso unificado en la evolución de los artículos, se puede ver como las diferentes maneras de trabajar de los editores en etapas tempranas del artículo tiene influencia en el desarrollo del mismo. Además, los procesos seguidos por los propios usuarios siguen patrones que permiten clasificarlos dentro de una taxonomía de roles de trabajo, verificando los hallazgos obtenidos en otros estudios. Por último, aunque generalmente no existe colaboración explícita entre los usuarios, se observan colaboraciones organizadas en momentos puntuales.

**Palabras clave**: Minería de procesos, Mineria social, Aprendizaje automático, ProM, Red de petri, Edición, Producción colaborativa de conocimiento

# Abstract

Collaborative writing has always been one of the pillars of the internet since the web 2.0. Usually, the attempts to create content collaboratively with individuals without organization or benefit are useless, resulting in a phenomenon called *the tragedy of the commons*. *The tragedy of the commons*, is a situation where a system composed of independent users that pursue their own goals behave against the common good. However, the selfless collaboration between different, unorganized users made possible the existence of communities whose only purpose is the diffusion of knowledge: the wikis.
The wikis, formed by a corpus of different articles, are currently under extensive study. Despite that, the processes that determine the evolution of its content and the processes followed by the users in its activity are not totally known. In this project, a series of techniques are proposed and applied: (i) process mining to discover and analyze the existent processes in the task of collaborative writing from an article and user point of view, and (ii) social mining to visualize the collaboration structures among the different users. The object of the study, will be the Spanish Wikipedia.
Every article in Wikipedia has a record of all the editions made to it. Those records along with a taxonomy of 13 semantic intention behind each revision (e.g re-factoring, elaboration or counter vandalism) will be used to meet the goals of this study. With a data set of 5684 revisions and its semantic intentions, a predictive model with a F1 micro of 0.64 is created. Combining this model with the records of revisions of some *featured articles* generates the different semantic intentions behind each revision. These predicted semantic intentions and its revisions constitute the input of the techniques of social mining and process mining. Such techniques allows the observation of the processes followed by the article, the processes followed by their users in their activity and the collaborative structure between users.
Results show that, even though there is no such thing as a unified process in the evolution of the articles, the different behaviours of the users in the initial stages of an article have an influence in its development. Furthermore, the processes followed by the users in their activity follow patterns that verify the findings of previous studies in this topic. Generally, there is no organized collaboration among the users. However, the results imply that sometimes the collaboration between users is explicitly organized.

**Keywords**: Process mining, Social mining, Machine learning, ProM, Petri Net, Collaborative production of knowledge, Edition

# Índice general

**Índice general** v

**Índice de figuras** viii

**Índice de tablas** 1

**1. Introducción** 2
1.1. Motivación . . . 3
1.2. Objetivos . . . 4
1.3. Metodología . . . 4
1.4. Estructura . . . 5

**2. Introduction** 7
2.1. Motivation . . . 8
2.2. Objectives . . . 9
2.3. Methodology . . . 9
2.4. Structure . . . 10

**3. Fundamentos teóricos** 11
3.1. Trabajo relacionado . . . 11
3.1.1. Identificando intenciones semánticas de las revisiones de Wikipedia . . . 11
3.1.2. Estabilidad turbulenta de los roles emergentes . . . 13
3.2. Aprendizaje automático . . . 14
3.2.1. Aprendizaje supervisado . . . 14
3.2.2. Métricas . . . 14
3.2.3. Validación cruzada de la clasificación . . . 15
3.2.4. Algoritmos de clasificación . . . 16
3.2.5. Ingeniería de características . . . 17
3.2.6. Sobremuestreo . . . 17
3.3. ¿Qué es la minería de procesos? . . . 18
3.3.1. Punto de comienzo: el formato XES . . . 19
3.3.2. Descubriendo los procesos . . . 21
3.3.3. Redes de Petri . . . 22
3.3.4. De la minería de procesos a la minería social . . . 23

**4. Tecnologías y herramientas** **25**
4.1. Tecnologías . . . 25
4.2. Herramientas . . . 27
4.2.1. ProM tools . . . 27

**5. Procesamiento de los historiales de revisión** **29**
5.1. Descarga de datos . . . 30
5.2. Extracción de información . . . 31
5.3. Obtención de características . . . 31
5.4. Cambio de formato . . . 32
5.5. Generación y análisis de modelos predictivos . . . 32
5.5.1. Datos iniciales . . . 33
5.5.2. Métricas de evaluación . . . 35
5.5.3. Clasificación binaria . . . 35
5.5.4. Clasificación multi-etiqueta . . . 43
5.5.5. Conclusiones . . . 47

**6. Análisis con minería de procesos** **49**
6.1. Obtención de datos . . . 50
6.2. Transformación a XES . . . 53
6.3. Análisis exploratorio de los datos . . . 53
6.4. Análisis a nivel artículo . . . 58
6.4.1. Análisis de la red de Petri descompuesta 1 . . . 61
6.4.2. Análisis de la red de Petri descompuesta 2 . . . 63
6.4.3. Análisis del conjunto de mini redes de Petri descompuestas 3 . . . 63
6.5. Análisis a nivel editor . . . 71
6.5.1. Editores de actividad baja . . . 71
6.5.2. Editores de actividad intermedia . . . 75
6.5.3. Editores de actividad alta . . . 80

**7. Análisis con minería social** **86**
7.1. Obtención de datos . . . 88
7.2. Handover of Work . . . 88
7.2.1. Artículo Tierra . . . 89
7.2.2. Artículo Ácido desoxirribonucleico . . . 90
7.3. Subcontracting . . . 92
7.3.1. Artículo Tierra . . . 92
7.3.2. Artículo Ácido desoxirribonucleico . . . 94

**8. Conclusiones** **96**
8.1. Conclusiones minería de procesos a nivel artículo . . . 96
8.2. Conclusiones minería de procesos a nivel editor . . . 96
8.3. Conclusiones minería social . . . 97
8.4. Conclusiones globales . . . 98

**9. Conclusions** **100**
9.1. Process mining applied to the article: conclusions . . . . . 100
9.2. Process mining applied to the editor: conclusions . . . . . 100
9.3. Social mining conclusions . . . . . 101
9.4. Global conclusions . . . . . 102

**10. Trabajo futuro** **103**

**11. Código** **104**

# Índice de figuras


3.1. Ejemplo de matriz de confusión (fuente: https://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html) . . . . . 15
3.2. Validación cruzada de 4 iteraciones (fuente: https://en.wikipedia.org/wiki/Cross-validation_(statistics)) . . . . . 16
3.3. Árbol de decisión . . . . . 16
3.4. Estructura de la minería de procesos (fuente: [4]) . . . . . 18
3.5. Ejemplo de log de eventos . . . . . 19
3.6. Diagrama de la estrúctura del formato XES (fuente: http://www.xes-standard.org/_media/xes/xes_standard_proposal.pdf) . . . . . 20
3.7. Ejemplo de evento en XML . . . . . 21
3.8. Descubrimiento de los procesos dentro de un log de eventos . . . . . 21
3.9. Ejemplo de una red de petri . . . . . 23
3.10. Estructura de la minería social en ProM . . . . . 24

4.1. Interfaz de ProM tools 6.8 . . . . . 27
4.2. Visor de grafos de ProM tools 6.8 . . . . . 28

5.1. Proceso seguido durante el procesamiento de los historiales de revisión de Wikipedia . . . . . 30
5.2. Distribución de intenciones en el conjunto de datos . . . . . 33
5.3. Revisiones con múltiples intenciones . . . . . 34
5.4. Distribución de las intenciones en porcentajes . . . . . 34
5.5. Proporcion de positivos y negativos por intencion . . . . . 36
5.6. Resultados por intencion del clasificador de prueba . . . . . 37
5.7. Matriz de confusión del clasificador de prueba binario . . . . . 38
5.8. Resultados de los bosques aleatorios . . . . . 39
5.9. Matriz de confusión de los bosques aleatorios . . . . . 40
5.10. Resultados por intencion de la máquina de soporte de vectores . . . . . 41
5.11. Matriz de confusión de la máquina de soporte de vectores . . . . . 42
5.12. Resultados bosques aleatorios trasaplicar el sobremuestreo . . . . . 43
5.13. Matriz de confusión tras realizar el sobremuestreo en los bosques aleatorios . 44
5.14. Resultados con ingeniería de características y sobremuestreo en los bosques aleatorios . . . . . 44
5.15. Matriz de confusión con ingeniería de características y sobremuestreo de los bosques aleatorios . . . . . 45
5.16. Resultados de de los bosques aleatorios finales . . . . . 45

5.17. Matriz de confusión de los bosques aleatorios finales . . . . . 46
5.18. Gráfica con resultados de modelos finales . . . . . 47

6.1. Proceso seguido para la descarga y preparación de los datos para realizar minería de procesos . . . . . 50
6.2. Esquema del filtrado del corpus para el análisis a nivel editor . . . . . 52
6.3. Representación de los atributos del formato CSV tras la conversión XES . . . 53
6.4. Corpus en formato XES . . . . . 54
6.5. Revisiones a lo largo del tiempo por artículo . . . . . 55
6.6. Intenciones a lo largo del tiempo en el corpus . . . . . 56
6.7. Representación de las distintas intenciones . . . . . 57
6.8. Número de revisiones por editor . . . . . 58
6.9. Número de editores según cantidad de ediciones realizadas . . . . . 58
6.10. Petri net obtenida con Minero Heurístico . . . . . 60
6.11. Petri net del proceso de edición tras su descomposición 1 . . . . . 61
6.12. Petri net del proceso de edición tras su descomposición 2 . . . . . 64
6.13. Redes de petri del proceso de edición tras su descomposición 3 . . . . . 65
6.14. Inicio de la red de petri del proceso de edición tras su descomposición 1 . . . 66
6.15. Sección intermedia de la red de petri del proceso de edición tras su descomposición 1 . . . . . 67
6.16. Final de la red de petri del proceso de edición tras su descomposición 1 . . . 68
6.17. Inicio de la red de petri del proceso de edición tras su descomposición 2 . . . 69
6.18. Final de la red de petri del proceso de edición tras su descomposición 2 . . . 70
6.19. 1º Petri net del proceso seguido por los editores de baja actividad . . . . . 72
6.20. 2º Petri net del proceso seguido por los editores de baja actividad . . . . . 72
6.21. 3º Petri net del proceso seguido por los editores de actividad baja . . . . . 73
6.22. 4º Petri net del proceso seguido por los editores de actividad baja . . . . . 74
6.23. Sección inferior de la 4º Petri net del proceso seguido por los editores de actividad baja . . . . . 75
6.24. Sección superior de la 4º Petri net del proceso seguido por los editores de actividad baja . . . . . 76
6.25. 5º Petri net del proceso seguido por los editores de actividad baja . . . . . 76
6.26. 1º Petri net del proceso seguido por los editores de actividad intermedia . . . 77
6.27. 2º Petri net del proceso seguido por los editores de actividad intermedia . . . 78
6.28. 3º Petri net del proceso seguido por los editores de actividad intermedia . . . 79
6.29. 4º Petri net del proceso seguido por los editores de actividad intermedia . . . 80
6.30. 1º Petri net del proceso seguido por los editores de actividad alta . . . . . 83
6.31. 2º Petri net del proceso seguido por los editores de actividad alta . . . . . 84
6.32. 3º Petri net del proceso seguido por los editores de actividad alta . . . . . 85

7.1. Estructura de la minería social en ProM . . . . . 87
7.2. Proceso a seguir para la obtención de los datos usados en la minería social . . 88
7.3. Grafo handover of work del artículo Tierra . . . . . 89
7.4. Zoom grupo central del grafo handover of work del artículo Tierra . . . . . 90
7.5. Grafo handover of work del artículo Ácido desoxirribonucleico . . . . . 91
7.6. Zoom grupo central del grafo handover of work del artículo Ácido desoxirribonucleico . . . . . 91

7.7. Grafo subcontracting del artículo Tierra . . . . . 92
7.8. Zoom grupo central del grafo subcontracting del artículo Tierra . . . . . 93
7.9. Grafo subcontracting del artículo Ácido desoxirribonucleico . . . . . 94
7.10. Zoom grupo central del grafo Subcontracting del artículo Ácido desoxirribonucleico . . . . . 95

# Índice de tablas

3.1. Intenciones según la taxonomía usada y su descripción (fuente: [24]) . . . . . 12
3.2. Taxonomía de roles según actividades realizadas en cada revisión . . . . . 13

5.1. Vistazo general de los datos . . . . . 33
5.2. Micro media de los resultados de los tres algoritmos . . . . . 38
5.3. Micro-media de los resultados de los diferentes bosques aleatorios creados . . 43
5.4. Bosque aleatorio final multi-etiqueta . . . . . 46
5.5. Micro media de los resultados de los modelos finales de cada enfoque . . . . . 47

# Capítulo 1

# Introducción

La colaboración es uno de los pilares de internet desde la web 2.0. Desde su inicio, su objetivo era poder eliminar las barreras existentes en la comunicación. Así, en cuestión de unas decadas, el volumen de datos existente en la red ha aumentado exponencialmente, llegando a influir en numerosos aspectos de nuestras vidas diarias. Uno de sus logros más destacables es la facilidad de la difusión de la información, haciendola más accesible y abundante que nunca.

Un fenómeno conocido como la tragedia de los comunes describe como en un sistema compuesto por usuarios que actúan independientemente, estos buscan el beneficio personal dando lugar a comportamientos contrarios al bien común. Sorprendentemente, en internet esto no siempre se cumple. Prueba de ello es la creación y éxito de las wikis: páginas de conocimiento basadas en la escritura colaborativa. En las wikis los usuarios son totalmente voluntarios y no existe beneficio alguno derivado de sus actividades [21]. A día de hoy existen miles de wikis diferentes pero de entre todas, destaca Wikipedia. Wikipedia es la enciclopedia libre más grande del mundo con 5.853.387 artículos, 36.262.835 editores y más de 18 mil millones de visitas anuales poniendo el conocimiento a disponibilidad del mundo entero. Además, Wikipedia cuenta con uno tipo de artículos denominados *artículos destacados*. Estos artículos han sido catalogados como referente de calidad, es decir, de 'los mejores artículos de Wikipedia'. En la Wikipedia Española, esto supone sólo 1127 artículos, el 0.07 % del total.

Cada uno de estos artículos cuenta con un historial de revisiones público al que cualquiera puede acceder facilmente. Gracias a esto las wikis pueden ser y están siendo ampliamente estudiadas (especialmente Wikipedia).
Un ejemplo de investigaciones ya realizadas, es el estudio de los diferentes roles que pueden adquirir los usuarios en función de su manera de trabajar en 'Estabilidad turbulenta de los roles emergentes' ([5]). En la investigación, se determina un conjunto de roles en función de las actividades que realizan los editores (por ejemplo: vigilantes, que supervisan el la integridad del artículo; editores todo terreno que hacen un poco de todo; vándalos que generan prejuicios a los artículos intencionadamente)
Otro caso, es el estudio de la supervivencia de los usuarios en función de las intenciones encontradas tras sus ediciones en 'Identificando intenciones semánticas en las revisiones de Wikipedia' ([24]). Aquí se desarrolla una taxonomía de 13 intenciones semánticas como wikificación, relacionada con motivos de formato de Wikipedia; elaboración, cuyo objetivo es añadir contenido, etc... y en base a ella se estudia la supervivencia de los usuarios tras sus

primeras revisiones, determinando que existe una relación.

Sin embargo, los flujos de trabajo en Wikipedia inherentes a su propio funcionamiento en los artículos y en los propios editores no son tan conocidos. Este flujo de trabajo puede ser denominado **proceso**. Un ejemplo de proceso sería el conjunto de pasos seguido por un artículo desde su creación hasta la actualidad. Partiendo de las investigaciones anteriores y los historiales de revisiones de los artículos de Wikipedia se plantea el estudio de los procesos que siguen los artículos en caso de que estos existan, así como los seguidos por los usuarios durante sus ediciones y las relaciones de colaboración existentes entre los mismos. Para esto se hará uso de la minería de procesos y la minería social.

La minería de procesos se compone por un conjunto de técnicas de minería que permiten extraer información para descubrir, monitorizar y mejorar procesos [4]. Esta información puede ser analizada para tomar forma de decisiones de negocio o estratégicas para optimizar los procesos o simplemente para conocer el propio funcionamiento de los mismos. Sin embargo, sus aplicaciones para incrementar la eficiencia de un proceso no es lo que lo hace interesante de cara al estudio objeto de este proyecto, sino el propio descubrimiento de los procesos inherentes a la edición y evolución de los *artículos destacados* y de los editores en comunidades de conocimiento colaborativo abiertas como la Wikipedia española.

Por otro lado, la minería social hace uso de técnicas de sociometría y análisis de redes sociales [3] para observar y conocer las posibles relaciones existentes entre los usuarios. Es decir, se trata una rama de la minería de procesos donde el foco se pone en las interacciones entre los usuarios presentes en el proceso. De esta manera, se realizará un estudio de las relaciones existentes entre los editores de un artículo de Wikipedia.

Aunque estas técnicas de minería de procesos y minería social son relativamente nuevas, ya han sido applicadas con éxito en ámbitos como la educación ([6]) para estudiar los procesos seguidos por escuelas o academias o en la escritura de conocimiento colaborativa dentro de trabajos realizados por estudiantes en un contexto similar al de este proyecto ([19]).

## 1.1. Motivación

La motivación de este proyecto es la falta de respuesta ante las siguientes preguntas:

- ¿Existen procesos determinados que siguan los artículos durante su desarrollo?
- ¿Qué procesos, en caso de existir, siguen los usuarios durante su historial de ediciones?
- ¿De qué manera colaboran entre sí los usuarios?

Tanto la primera como la segunda pregunta no consta que hayan sido resueltas. Para resolverlas se hará uso de la minería de procesos. Un enfoque totalmente novedoso, pues además de ser un área de conocimiento relativamente nueva, nunca ha sido aplicada en este contexto. Este enfoque seleccionado además de novedoso es prometedor pues estudiar la estructura o estructuras que puedan seguirse en Wikipedia aporta una perspectiva completamente nueva

y más general que pretende ver si realmente Wikipedia sigue un proceso anárquico de evolución en sus artículos o si existe cierta organización. Del mismo modo, estudiar los procesos de los usuarios desde la perspectiva del conjunto de sus acciones, puede permitir reforzar los resultados de los roles de usuario de estudios como el previamente mencionado 'Estabilidad turbulenta de los roles emergentes' ([5]).
Por otro lado, la tercera pregunta aunque sí estudiada nunca se ha estudiado mediante la minería social derivada de la perspectiva de proceso. Además, permite complementar los resultados obtenidos para responder la segunda pregunta.

## 1.2. Objetivos

Los objetivos de este proyecto son básicamente las preguntas previamente formuladas. Mediante el novedoso enfoque de la minería de procesos, se tratará de descubrir los procesos internos seguidos por un conjunto aleatorio seleccionado de *artículos destacados* de la Wikipedia española. Además se descubrirán también los procesos seguidos por los propios usuarios en su comportamiento habitual de edición dentro de esos mismos artículos y las estructuras de colaboración que puedan formarse entre los usuarios.

## 1.3. Metodología

El punto de partida del proyecto es el historial de revisiones de cada artículo, la Wikipedia Española y las investigaciones previamente mencionadas ( 'Estabilidad turbulenta de los roles emergentes' [24] e 'Indentificando intenciones semánticas en las revisiones de Wikipedia' [5]).

En primer lugar se seleccionará un conjunto de *artículos destacados* aleatorios pero cada uno de diferente temática. Los historiales de edición de cada artículo serán descargados y se extraerá información útil de los mismos como el nombre del editor y la fecha en la que se realizó la revisión.

De estos historiales, queremos obtener las intenciones tras cada una de sus revisiones. En este punto, entra en juego la investigación que desarrolla una taxonomía de 13 categorías que representan las distintas intenciones semánticas tras cada revisión ([24]) relacionadas con motivos de edición de texto o formato, supervisión de la integridad del artículo o corrección de errores. Siguiendo esta taxonomía desarrollamos un modelo predictivo que alcanza valores de micro F1 de 0.64 capaz de etiquetar las intenciones de los historiales de revisión descargados.

Una vez que contamos con los historiales de revisión etiquetados en base a su intencionalidad, se puede comenzar el estudio para responder a las preguntas objetivo de este trabajo.

La pregunta *¿Existen procesos determinados que siguan los artículos durante su desarrollo?* será contestada mediante la aplicación de la minería de procesos bajo la perspectiva de artículo. Es decir, se analizará el conjunto de artículos en base al flujo de intenciones seguido en cada una de las revisiones realizadas en los mismos.

En cuanto a la siguiente pregunta u objetivo, *¿Qué procesos, en caso de existir, siguen los usuarios durante su historial de ediciones?*, se hará uso también de la minería de procesos. Sin embargo, la perspectiva será otra: el foco se pondrá en el usuario. De esta manera, lo

que se hará será estudiar el conjunto de usuarios y sus acciones en cada revisión determinada por su intencionalidad en lugar de cada artículo. Para obtener una perspectiva mas compartimentalizada, los usuarios serán agrupados en función de diferentes niveles de actividad y cada nivel se estudiará por separado. Así, los resultados obtenidos se interpretarán junto con la taxonomía de roles de editor mencionada previamente ([5]).

Por último la cuestión *¿De qué manera colaboran entre sí los usuarios?* será contestada gracias a la aplicación de la minería social. Se estudiarán las relaciones entre los usuarios en base a distintas métricas como handover of work, que mide el relevo de unos usuarios a otros en las labores de edición o subcontracting que mide la cantidad de veces que un usuario edita entre dos revisiones de otro. Este análisis requiere un nivel de detalle mayor por lo que en lugar de aplicar estas técnicas al conjunto de historiales de revisión previamente descargados, se aplicarán individualmente a solamente dos artículos del conjunto.

## 1.4. Estructura

La estructura del documento es la siguiente:

1. Introducción: La introducción plantea el problema inicial, la motivación para resolverlo y el proceso seguido para ello. Este capítulo además se encuentra tanto en inglés como en español.

2. Fundamentos teóricos: Introduce los fundamentos teóricos necesarios para la correcta comprensión del texto. Desarrolla los pilares teóricos de investigaciones previas en los que se apoya el proyecto y explica en detalle en qué consiste la minería de procesos y qué utilidad tiene. Además, se explica en qué consiste la minería social y su relación con la minería de procesos.

3. Tecnologías y herramientas: Este capítulo se centra en la explicación de las diferentes tecnologías usadas a lo largo del proyecto.

4. Procesamiento de los historiales de revisión: Su objetivo es explicar todo el proceso seguido desde la descarga de los datos iniciales hasta la obtención de un conjunto de datos listo para ser analizado mediante la minería de procesos. Incluyendo por tanto la elaboración y uso de un modelo predictivo basando en la taxonomía de intenciones semánticas desarrollada en 'Identificando intenciones semánticas tras las revisiones de Wikipedia' [24].

5. Análisis con minería de procesos: Explica en detalle todo el trabajo realizado con técnicas de minería de procesos desde dos perspectivas diferentes: usuario y artículo para obtener información acerca del proceso seguido por los artículos en su evolución y el seguido por los editores en sus sesiones de edición.

6. Análisis con minería social: En este capítulo comentamos en detalle los resultados de aplicar un análisis de minería social para descubrir las posibles colaboraciones entre los editores de Wikipedia usando un variado conjunto de datos y diferentes métricas.

7. Conclusiones: Clara y breve descripción de todos los hallazgos hechos a lo largo del proyecto con la aplicación de la minería social y de procesos.

8. Trabajo futuro: Capítulo que habla del trabajo que aún no ha sido realizado, los motivos de esto y posibles mejoras que se puedan añadir.

9. Código: Qué scripts han sido utilizados durante el proyecto así como dónde se encuentran localizados y su autoría.

# Capítulo 2

# Introduction

Collaborative writing is one of the pillars of the Internet since the 2.0 web. In the beginning of the internet its objective was to eliminate the existing communication barriers. In just a couple of decades the volume of data existing on the net has grown exponentially, reaching a point of direct influence on our daily lives. One of Internet's biggest achievement is its diffusion power making knowledge more available and freer than ever before.

A phenomenon known as *the tragedy of the commons* describe a system composed of independent users that pursue their own goals behave against the common goal. However, this is not always true. The selfless collaboration between different, unorganized users made possible the existence of communities whose only purpose is the diffusion of knowledge: the wikis. In a wiki, the users are completely voluntary and there is no benefit whatsoever [21]. Nowadays, there is thousands of different wikis but among all of them there is an special case called Wikipedia. Wikipedia is the biggest free encyclopedia in the world with 5.853.387 of articles, 36.262.835 editors and more than 18 billions of annual visits. Furthermore, Wikipedia has different categories of articles based on their quality. The top quality ones are called *featured articles* and they are scarce. As an example, in the Spanish Wikipedia the featured articles only represent the 0.07 % of articles: only 1127 articles.

Each and every of the available articles in Wikipedia has its own public historic of revisions. Thanks to this, the wikis are currently being under extensive study.
One of the researches already published creates a profiling for the editors based on their activities ('Turbulent stability of emergent roles' [5]). A taxonomy of roles is created (e.g watchdogs that ensures the integrity of the article, all-round-contributors who do every kind of task, vandals who damage the article on purpose...) in order to perform this profiling.
Another case of study is the survival of the new editors based on the intentions behind their revisions in 'Identifying semantic edit intentions from revisions in Wikipedia' ([24]). In this paper a taxonomy of 13 categories of semantic intentions behind each revision is created (e.g wikification, related with formatting of Wikipedia; elaboration, based on the addition of content...). Based on this taxonomy, the survival of the new users is studied during their first session of editions. Findings show that there is a relation between the intentions done and their survival, implying that some editions are not suited for beginners.

However, the work flows in Wikipedia inherent to its own activity in the articles and in the

editors are not known. These work flows are also known as **processes**. An example of process would be the orderly combination of steps since the creation of an article to the present day. Using the results obtained in the the previous researches and the historic of revisions from the Wikipedia articles, this project proposes the study of the processes followed by articles, in case they exists, and the processes of the users in their edition sessions. Also, the relationships between the users will be studied. In order to attain those goals, a new approach is proposed: process mining and social mining.

Process mining is composed by different mining techniques with the purpose of extracting information, discovering processes and improving them [4]. This information is analyzed to transform it into business or strategic decisions to improve performance. However, its applications to improve performance are not useful in the context of this study. In this document the discovery of the processes themselves is the goal. Using process mining the processes followed by the articles and the users in their edit sessions will be unveiled.

On the other hand social mining use a combination of sociometry techniques with social network analysis [3] to observe and discover the existing relationship among the users. Social mining derives from process mining. However here the focus is the interactions in the users present in the processes. With such techniques an study of the social aspects of the users will be performed to obtain a clear picture of the relationship formed between the editors in a Wikipedia article.

Despite this techniques being relatively new, they have already been applied with success in different fields like education to study the processes followed by schools or academies ([6]) and also in the collaborative writing trying to discover the processes followed by students writing a project in groups ([19]).

## 2.1. Motivation

The motivation behind this project is the lack of answer to the following questions:

- Is there any process followed by an article in its evolution?
- Which processes do a users follow during its activity?
- How do the different editors collaborate with each other?

The first and the second answer have not been answered previously. In order to answer them, process mining will be used. This approach is completely new in this context and provides new tools to broaden our knowledge of Wikipedia. Such techniques enable us to study the inner processes of Wikipedia's articles and users to determine if there is a 'path' in the evolution of articles and users or if it is an anarchic evolution.
On the other hand, the third question has already been studied. However, it has not been studied yet with the social mining approach given by the process mining. Furthermore, this information could potentially complement the results from the previous question.

## 2.2. Objectives

The goals of this project are the questions presented above. Through process mining, we will try to discover the inner processes followed by a randomly selected corpus of *featured articles* from the Spanish Wikipedia. Also, the processes followed by the uses will be discovered as well as the collaboration structures that may happen naturally within the editors in these communities.

## 2.3. Methodology

The starting point of this project is the historic of revisions from each article, the Spanish Wikipedia and the researches previously mentioned ([24], [5]).

First of all a corpus of random featured articles from different topics will be chosen. Each historic of revisions is downloaded and the useful information like editor name and time-stamp extracted.

The next step is obtaining the intentions behind each revision of the historic. The taxonomy of 13 categories of semantic intentions behind a revision is perfect for this task. This intentions (developed in [24]) are related with motives like formatting, edition, grammar changes... To apply this taxonomy to our historic of revisions we developed a predictive model achieving a micro F1 score of 0.64.

Once the historic of revisions are labeled based on the semantic intentions, it is possible to try and answer the main questions of this project.

The question *Is there any process followed by an article in its evolution?* will be answered via the application of process mining with the focus on the article itself. In other words, the corpus of articles will be studied by observing the flow of intentions in each article.

The second question, *Which processes do a users follow during its activity?* will also be answered with process mining. However, the perspective is different. Now, the focus is on the users: the users will be studied by discovering the processes followed by them in their edit sessions, considering the intentions each user have behind its own edits. In order to obtain more information the users are grouped in different categories based on their edit count in the corpus articles. This allows for an individual analysis for each category to observe difference in behaviour and processes followed, depending on the activity of the editors. Furthermore, this results can be interpreted jointly with the role taxonomy previously mentioned ([5]).

Lastly, the question *How do they different editors collaborate with each other?* requires the use of social mining. The relationships of the users will be studied using different metrics (e.g handover of work, which measures the handing of work from one editor to the other; subcontracting, which measure if an editor performs an edit between two editions of another user). This analysis requires a bigger level of detail as each article may have its own 'community' of editors. Such detail will be obtained by studying specific articles individually.

## 2.4. Structure

The structure of this document is as follows:

- Introduction: The introduction states the initial problem, motivation to solve it and the approach used for solving it.
- Theoretical background: Introduces the theoretical background needed for the correct comprehension of this document. It explains related work and provides a detailed introduction of the process mining and the social mining field.
- Technologies and tools: This chapter focus on the explanation of the different technologies and tools used throughout the project.
- Processing of the historic of revisions: Focus on all the work to get a corpus ready for process mining. Starting with the download of the initial data to the elaboration of the model to predict the intentions based on the 13-category taxonomy.
- Process mining analysis: Explains in detail all the work done with the process mining techniques regarding this project from two perspectives: article and editor.
- Social mining analysis: This chapter describes the use of social mining to unveil the collaborations between the different editors, using a rich set of data and different social metrics.
- Conclusions: Clear and brief explanation of all the finding done throughout this project.
- Future Work: Work that hasn't been done yet as well as possible improvements for the actual work.
- Code: Chapter explaining the different code used as well as its location and authorship.

# Capítulo 3

# Fundamentos teóricos

El objetivo de este capítulo es explicar la base teórica necesaria para la correcta comprensión de este documento.

Primero de todo, se explicarán aquellos trabajos existentes que sirven de fundamento para este proyecto. Después, se explicarán los conceptos claves utilizados durante el capítulo 5 para realizar un modelo predictivo mediante el uso del aprendizaje automático. Por último se introducirá y explicará brevemente las bases de la minería de procesos y de la minería social así como la relación entre ambas.

## 3.1. Trabajo relacionado

Durante la introducción de este documento se han mencionado dos investigaciones previas clave para este proyecto:

- La primera investigación estudiaba la supervivencia de los usuarios en Wikipedia en función del tipo de edición que realizan mediante una taxonomía de intenciones semánticas tras cada revisión ([24]) titulada *identificando intenciones semánticas de las revisiones de Wikipedia*
- La segunda realizaba un estudio sobre los diferentes roles que pueden adquirir los usuarios de una comunidad como Wikipedia y sus variaciones en el tiempo ([5]) y toma el nombre *estabilidad turbulenta de los roles emergentes.*

### 3.1.1. Identificando intenciones semánticas de las revisiones de Wikipedia

Esta investigación supone una parte crucial de este proyecto pues su existencia posibilita este estudio. Realizado por Diyi Yang, Aaron Halfaker, Robert Kraut y Eduard Hovy trata de crear una taxonomía de 13 intenciones semánticas tras cada revisión de un artículo de Wikipedia [24].

Con esta taxonomía clasifican un conjunto de datos que les permite estudiar la supervivencia de los usuarios. Este estudio lo realizan midiendo la 'efectividad' de una revisión en base a su intención. Es decir, si la edición es revertida o no. Con esto, obtienen la conclusión de que el tipo de intención realizado en la primera sesión de edición de un usuario afecta a la

supervivencia del mismo en la comunidad y que diferentes etapas de desarrollo de un artículo necesitan diferentes tipos de edición.

| **Intención** | **Descripción** |
|---|---|
| Clarification | Especificar o explicar un hecho existente o un significado mediante un ejemplo o discusión sin añadir información nueva |
| Copy editing | Editar oraciones, mejorar gramática o sintaxis. |
| Counter-Vandalism | Eliminar vandalismo |
| Disambiguation | Actualizar el enlace de una página desambiguada a una específica |
| Elaboration | Extiende/Añade una cantidad substancial de contenido nuevo. Añade hechos o nuevos datos importantes. |
| Fact update | Actualiza datos, fechas, puntuaciones, estado, etc... basándose en nueva información disponible |
| Point of view | Re-escribir utilizando un tono neutral propio de enclicopedias. Elimina valoraciones de carácter personal |
| Process | Comenzar/Continuar un flujo de trabajo en la wiki como marcar un artículo con noticias references a su limpieza, borrado, etc. |
| Refactoring | Re-estructurar el artículo. Mover y re-escribir contenido sin cambiar el significado del mismo |
| Simplification | Reduce la complejidad del artículo; puede eliminar contenido. |
| Vandalism | Intenta deliberadamente dañar el artículo |
| Verification | Añade/modifica referencias/citaciones; elimina texto que no pueda ser verificado |
| Wikification | Cuestiones de formato para cumplir con las guías de estilo |
| Other | Ninguna de las anteriores |

Tabla 3.1: Intenciones según la taxonomía usada y su descripción (fuente: [24])

Esta taxonomía de intenciones es de especial relevancia para el proyecto pues es la forma en la que se caracterizarán las revisiones. Cada revisión dotada de su intencionalidad permite estudiar la evolución y el flujo de trabajo seguido en Wikipedia tanto por los artículos como por los propios editores, proporcionando un contexto.

### 3.1.2. Estabilidad turbulenta de los roles emergentes

Publicada en 2016 esta investigación desarrollada por Ofer Arazy, Johannes Daxenberger, Hila Lifshitz-Assaf, Oded Nov e Iryna Gurevych con el objetivo de estudiar las dinámicas existentes en los roles emergentes tanto a nivel individual como organizacional [5].

Haciendo uso de un conjunto de mil artículos de Wikipedia con un alto número de diferentes editores se estudió cada revisión realizada y se desarrolló un conjunto de posibles acciones en ellas tales como mover o crear un nuevo artículo, añadir nuevo contenido, añadir referencias... En base a esto, se perfiló a cada usuario generando una taxonomía de roles de editor:

| **Rol** | **Descripción** |
|---|---|
| All-round contributor | Realizan tareas de todo tipo |
| Quick-and-dirty | Añaden contenido a los artículos, aunque por el estilo de edición a veces incurren en vandalismo por no ajustarse al formato de Wikipedia |
| Copy editors | Arreglan errores de texto y gramáticales |
| Content shapers | Reorganiza el texto y adecúan el formato de Wikipedia |
| Layout shapers | Sólo adecúan los artículos al formato de Wikipedia |
| Watchdogs | Eliminan vandalismo de los artículos |
| Vandals | Intentan de forma deliverada dañar el artículo |

Tabla 3.2: Taxonomía de roles según actividades realizadas en cada revisión

En base a esta taxonomía se estudia la estabilidad de los roles por usuario y se observa que existen variaciones en el rol que toma un usuario a lo largo de su actividad en Wikipedia. Sin embargo, lo que más interesa de cara a este proyecto es la taxonomía de roles presentada.

Previamente se han introducido los objetivos de este documento siendo uno de ellos la siguiente pregunta : *¿Qué procesos, en caso de existir, siguen los usuarios durante su historial de ediciones?* Gracias a los diferentes roles de esta taxonomía podemos determinar el perfil de cada editor. Haciendo uso de la taxonomía de intenciones semánticas tenemos información al respecto de que se ha hecho en cada edición, lo que permite tener información suficiente para tratar de relacionar estos procesos existentes en el trabajo de los usuarios con los roles aquí presentados.

De esta manera, los resultados aportados por la minería de procesos pueden ser contrastados con estudios existentes y quizá, complementarlos.

## 3.2. Aprendizaje automático

El aprendizaje automático es un campo de la informática cuyo objetivo es la solución de problemas mediante la obtención de un conjunto de datos y un modelo estadístico basado en estos datos [8]. Es decir, se dedica al estudio de los algoritmos y modelos estadísticos que hacen posible que un computador realize una tarea específica sin instrucciones explícitas sobre cómo hacerlo.

Este aprendizaje puede ser supervisado, sin supervisión, reforzado o semi-supervisado. En este proyecto específicamente el interés está en el aprendizaje supervisado.

### 3.2.1. Aprendizaje supervisado

El aprendizaje supervisado se basa en el modelado de la relación entre un conjunto de atributos y una etiqueta asociada a los mismos [20].

El conjunto de datos de entrada se compone de ejemplos etiquetados en base a diferentes categorías como podría ser $Verdadero$ o $Falso$.

De esta manera, los algoritmos de aprendizaje automático hacen uso de unos datos etiquetados para inferir una función que sea capaz de distinguir entre las diferentes etiquetas posibles y poder etiquetar automáticamente nuevos casos en el futuro con la necesidad, solamente, de los atributos.

Un ejemplo podría ser un algoritmo que detecte spam en la bandeja de entrada del correo electrónico, catalogando cada correo entre {*spam, no spam*}.

El aprendizaje supervisado, puede subdividirse en tareas de *clasificación* o *regresión*. La diferencia se basa en el tipo de etiqueta: en la regresión la etiqueta es una cantidad continua como $[0, 10]$ mientras que en la clasificación se trata de una categoría discreta como {*spam, no spam*} [20]. En este proyecto, las tareas realizadas en aprendizaje automático supervisado son de *clasificación*.

Para poder evaluar estas clasificaciones, se han desarrollado diferentes métricas.

### 3.2.2. Métricas

Para evaluar el rendimiento de un modelo clasificador, existen diferentes métricas calculadas en base a las siguientes variables:

- Positivos verdaderos (PV) : ejemplos clasificados como verdaderos correctamente.
- Positivos falsos (PF) : ejemplos clasificados como verdaderos incorrectamente.
- Negativos verdaderos (NV) : ejemplos clasificados como negativos correctamente.
- Negativos falsos (NF): ejemplos clasificados como negativos correctamente.

Estas variables permiten el cálculo de las siguientes métricas:

- Precisión: Representa el número de ejemplos positivos asignados correctamente clasificados dividido por el total de ejemplos positivos asignados clasificados con ese valor [18]. $P = \frac{PV}{PV+PF}$
- Sensibilidad: Representa el número de ejemplos positivos correctamente clasificados dividido entre el número de positivos en el conjunto de datos [18]. $S = \frac{PV}{PV+NF}$
- F1: Se trata de una combinación de precisión y sensibilidad. $F1 = \frac{2 \cdot precisión \cdot sensibilidad}{precisión + sensibilidad}$
- Matriz de confusión: Sirve para describir el rendimiento de un modelo predictivo mostrando visualmente los PV, PF, NV y NF tal y como se observa en la figura 3.1. En la figura de ejemplo, se ve como todas las instancias de setosa y virginica han sido clasificadas correctamente, mientras que 6 instancias de versicolor han sido clasificadas como virginica incorrectamente.

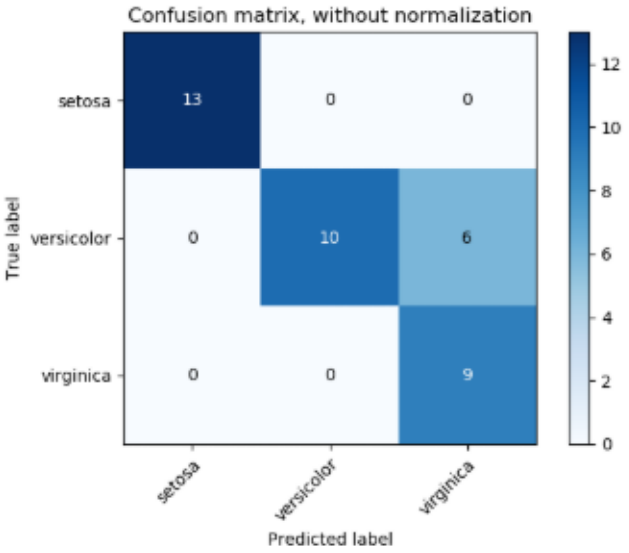



Figura 3.1: Ejemplo de matriz de confusión (fuente: `https://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html`)

### 3.2.3. Validación cruzada de la clasificación

La validación de un modelo es sencilla: tras escoger un modelo y sus parámetros, estimamos con las métricas anteriores su efectividad mediante datos de entrenamiento, comparando la predicción con los datos conocidos [20].

Si el conjunto de datos del que se dispone no es particularmente grande, la porción de los datos que se utilizan para entrenar el modelo puede no contener todos los casos necesarios para una adecuada clasificación. Para evitar esto, existe la validación cruzada.

La validación cruzada se basa, por tanto, en realizar una secuencia de entrenamientos del modelo con diferentes sub-grupos del conjunto de datos que actúan tanto como entrenamiento como de validación [20]. Su funcionamiento puede ser observado en la figura 3.2 donde se realizan 4 iteraciones.

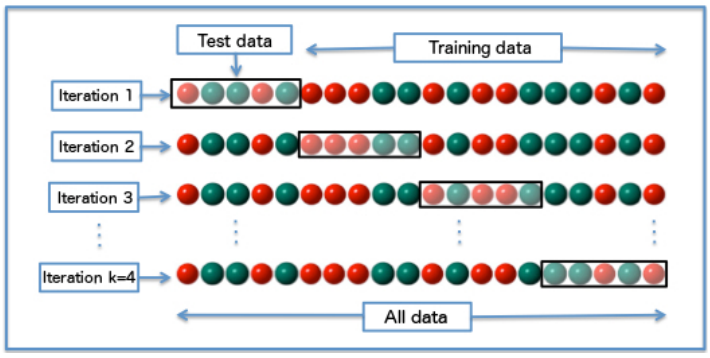



Figura 3.2: Validación cruzada de 4 iteraciones (fuente: `https://en.wikipedia.org/wiki/Cross-validation_(statistics)`)

### 3.2.4. Algoritmos de clasificación

#### Árboles de decisión

Un árbol de decisión es un grafo acíclico que puede ser utilizado para tomar decisiones [8]. En nodo rama del grafo, un atributo específico se evalúa. Si el valor del atributo se encuentra debajo de un valor específico, se sigue la rama izquierda; en el caso contrario, se sigue la rama derecha. La decisión a tomar se encuentra alojada en los nodos hoja, por lo que al llegar al final del árbol, hemos obtenido la etiqueta.

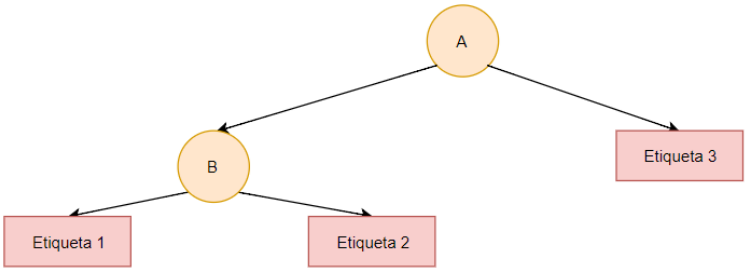



Figura 3.3: Árbol de decisión

La ventaja de los árboles de decisión es que son extremadamente intuitivos y eficientes pues cada decisión tomada reduce las posibilidades a la mitad. Sin embargo, son muy propensos al *overfitting* o sobre-ajustamiento [20]. Este fenónemo sucede cuando los algoritmos se ajustan a las particularidades específicas de los datos de entrenamiento en lugar de las peculiaridades generales que permitan generalizar esos datos.

#### Bosques aleatorios

Una de las soluciones para reducir el sobre-ajustamiento de los árboles de decisión es aplicar una técnica llamada *embolsado* (*bagging*).
El embolsado, se basa en combinar modelos que sobre-ajusten para reducir los propios efectos del sobre-ajustamiento. Con los resultados de cada modelo se hace una media para encontrar

la mejor clasificación posible [20].

De esta manera, surgen los bosques aleatorios. Es decir, un bosque aleatorio, no es más es un conjunto de árboles de decisión aleatorios de modo que el problema del sobre-ajustamiento se soluciona en cierta medida.

#### Máquina de soporte vectorial

Las máquinas de soporte vectorial son algoritmos fléxibles que permiten tanto hacer tareas de clasificación como de regresión.

Su funcionamiento es simple, en lugar de modelar cada posible etiqueta, se busca una línea o curva (en espacios 2-Dimensionales) o planos e hiperplanos (en espacios N-dimensionales) que separe a las diferentes etiquetas en secciones [20].

Entre sus ventajas, destacan la rapidez en la predicción y su buen rendimiento con datos de alta dimensionalidad como es el caso de los datos de este proyecto [20].
Sin embargo, también existen desventajas: el tiempo de entrenamiento suele ser alto y son modelos muy dependientes de un minucioso ajuste de parámetros [20].

#### $k$-vecinos más próximos

El algoritmo $k$-vecinos más proximos es un algoritmo simple.
Para generar una predicción en un nuevo dato, encuentra $k$ ejemplos que tengan atributos similares y busca cual es la etiqueta que más aparece en los ejemplos seleccionados. Una vez que encuentra la etiqueta, se la asigna al nuevo dato [12].

De esta manera este algoritmo es uno de los más simples y rápidos que hay. Sin embargo, esta simpleza tiene sus consecuencias: se le considera un *aprendiz vago* (*lazy learner*), es decir, no aprende nada de los datos de entrenamiento, solamente los utiliza para clasificar [12].

### 3.2.5. Ingeniería de características

La ingeniería de características se compone de un conjunto de técnicas cuyo objetivo es mejorar el rendimiento de los modelos predictivos [20]. En este caso, las técnicas de ingeniería de características que se utilizan en el documento tienen un enfoque basado en el modelo.

El funcionamiento es simple, en base a un umbral de importancia dado, el propio modelo predictivo decide la importancia de cada atributo y descarta todos aquellos con un nivel de importancia menor. Así, todos aquellos atributos que solo añaden incertidumbre o ruido para la tarea de clasificación son eliminados.

### 3.2.6. Sobremuestreo

En algunas ocasiones, las datos se encuentran distribuidas de manera poco equitativa. Esto provoca que la proporción en la cual se encuentran las diferentes etiquetas sea muy desigual. Un ejemplo de datos poco balanceados: un conjunto de 1000 datos formado por las

visitas a una tienda on-line donde sólo 10 casos han realizado una compra. De esta manera, el modelo tiene muy pocos ejemplos para aprender a clasificar correctamente aquellos casos en los cuales se realizó la compra.

Una solución a este problema es el sobremuestreo, aplicado en este proyecto. En el sobremuestreo se generan casos de la clase en minoría para reducir la desigualdad.
Esta generación de casos se realiza mediante la 'perturbación' de casos ya existentes. Creando nuevos casos que son similares pero no iguales, el modelo tiene más oportunidades para aprender a clasificarlos correctamente [12]. Hay diferentes técnicas para ello, la aplicada en este proyecto se denomina *ADASYN*.

ADASYN hace uso de una distribución con pesos para cada etiqueta en minoría, así, genera ejemplos de esa clase mejorando los resultados en la clasficación reduciendo la parcialidad [11].

## 3.3. ¿Qué es la minería de procesos?

La minería de procesos se compone por un conjunto de técnicas de minería de datos que permiten extraer información de logs de eventos para descubrir, monitorizar y mejorar procesos [4]. Esta información puede ser analizada para tomar forma de decisiones de negocio o estratégicas para optimizar los procesos o simplemente para conocer el propio funcionamiento de los mismos. Sin embargo esto genera las siguientes preguntas ¿qué es un proceso y en qué consiste la minería de datos?

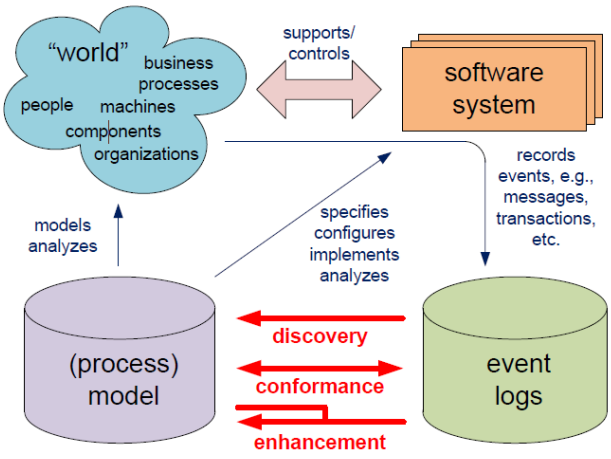



Figura 3.4: Estructura de la minería de procesos (fuente: [4])

Por un lado, un proceso es un conjunto estructurado de actividades o tareas realizados por un actor con un objetivo particular. Un ejemplo de proceso, sería el flujo de trabajo seguido durante la escritura colaborativa de un artículo en Wikipedia o la cadena de tareas realizadas en un restaurante desde que se sienta el usuario hasta que su comida ha sido servida.
Por otro lado, la minería de datos consiste en un conjunto de aprendizaje automático y es-

tadística cuyo objetivo es el descubrimiento de patrones en grandes cantidades de datos [9]. Por lo que en este caso, la minería de procesos se basa en los mismos principios pero aplicado a un nivel organizativo mayor: el proceso en sí mismo.

Así, la minería de procesos tiene multitud de aplicaciones en diferentes áreas, pero principalmente sirve para unir la brecha entre la minería de datos y el Business Intelligence [4]. Algunos ejemplos de empresas que han aplicado la minería de procesos con notable éxito son Walmart o Vodafone tal y como expone Michal Rosik en [15]. Walmart aplicó la minería de procesos para descubrir y hallar ineficiencias en el proceso de compra, hallando que el checkout no era lo eficiente que debería lo que les llevó a poder hallar estrategias para reducir el tiempo que los usuarios perdían en el proceso. Vodafone por otro lado ha conseguido aumentar el número de procesos que funcionan satisfactoriamente sin intervención humana en un 20 % en tan solo dos años mediante el uso de minería de procesos según expone el artículo de Rosik.

Además de en la industria, la minería de procesos se ha aplicado en diferentes investigaciones con éxito en ámbitos como la educación ([6]) para estudiar los procesos seguidos por escuelas o academias o en la escritura de conocimiento colaborativa dentro de trabajos realizados por estudiantes en un contexto muy similar al de este proyecto ([19]).

Los requisitos de cara a poder realizar minería de procesos son pocos, sólo necesitamos un log de eventos. Un log de eventos no es más que un fichero de texto donde almacenamos información proveniente de bases de datos, transacciones... Es decir, son colecciones de secuencias de eventos.

| | page_id | page_title | page_ns | revision_id | timestamp | contributor_id | org:resource | bytes | intentionality |
|---|---|---|---|---|---|---|---|---|---|
| | Filtro | Filtro | Filtro | Filtro | Filtro | Filtro | Filtro | Filtro | Filtro |
| 1 | 3825 | \|Leche\| | 0 | 7778 | 2002-11-20T0... | 1 | \|AstroNomo\| | 466 | wikification |
| 2 | 3825 | \|Leche\| | 0 | 7883 | 2002-11-20T0... | 1 | \|AstroNomo\| | 497 | |
| 3 | 3825 | \|Leche\| | 0 | 22507 | 2002-11-24T0... | 7 | \|Maveric149\| | 509 | wikification |
| 4 | 3825 | \|Leche\| | 0 | 38407 | 2003-08-09T1... | 5 | \|Andre Engels\| | 574 | copy-editing |
| 5 | 3825 | \|Leche\| | 0 | 42101 | 2003-10-06T0... | 2121 | \|Moriel\| | 592 | wikification |

Figura 3.5: Ejemplo de log de eventos

Un ejemplo de un conjunto de datos que podría ser considerado un log de eventos puede ser observado en la figura 3.5. Como vemos se compone de diversos atributos habituales en ficheros de datos: id, timestamp, título... Sin embargo, para poder aplicar técnicas de minería de procesos es necesario un formato específico para almacenar los datos: el formato XES.

### 3.3.1. Punto de comienzo: el formato XES

XES es el formato estándar de la IEEE Task Force en minería de procesos [4]. El nombre proviene de eXtensible Event Stream y se trata de un estándar que define una gramática para un lenguaje de marcado cuyo objetivo es proveer a los diseñadores de los sistemas de información de una metodología unificada para capturar el comportamiento de un sistema mediante log de eventos y torrentes de eventos [23]. Este lenguaje de marcado sobre el que se basa es el XML.

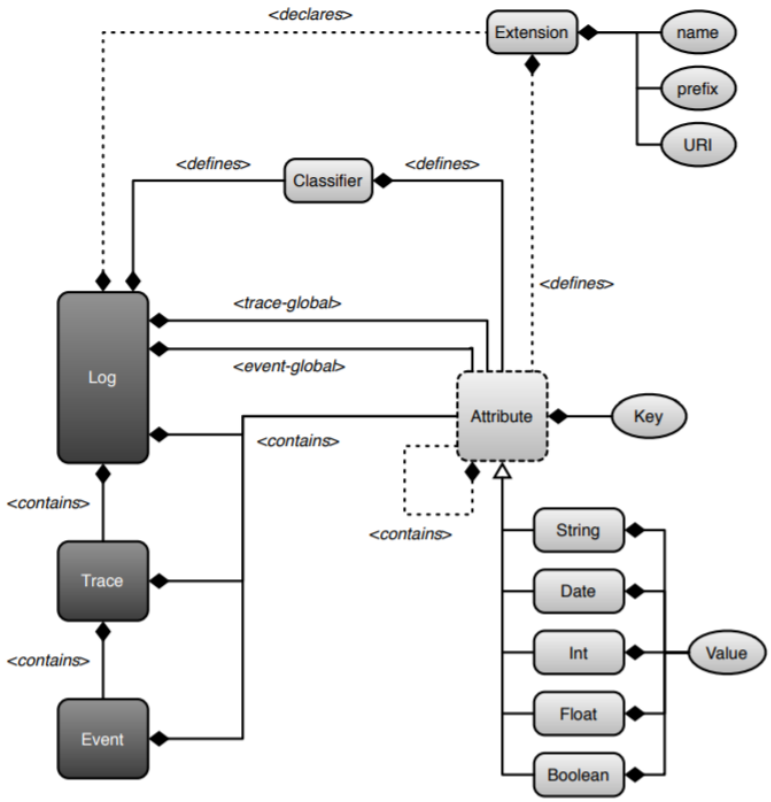



Figura 3.6: Diagrama de la estrúctura del formato XES (fuente: `http://www.xes-standard.org/_media/xes/xes_standard_proposal.pdf`)

El XML o *lenguaje de marcado extensible* no es más que un conjunto de reglas de codificación para hacer un documento legible tanto para un humano como para una máquina. Fue desarrollado por el World Wide Web Consortium (W3C) con foco en la simplicidad y la generalidad [7].

Algunas de las aplicaciones que hacen uso de este tipo de archivos son por ejemplo ProM, detallada en el capítulo 4, Disco desarrollada por Fluxicon con el foco en la creación de visualizaciones muy cuidadas de los procesos encontrados en los datos o QPR ProcessAnalyzer un programa de nivel empresarial para aplicar técnicas de minería de procesos de manera sencilla y rápida.

La estructura de de los archivos XES puede verse en el esquema 3.6. La base del formato XES y de los logs de eventos es asumir que es posible guardar secuencialmente eventos de modo que cada evento se refiera a una actividad y esté asociado a un caso particular (traza) [4]. Además, los logs de eventos pueden guardar otra información extra en forma de atributos como el actor del evento iniciando o finalizando la actividad, la marca de tiempo o timestamp o datos relativos al evento.

- Las trazas representan cada caso del proceso a analizar. Por ejemplo, en un log de

eventos compuesto por las revisiones realizadas en diferentes artículos de Wikipedia, cada traza podría ser representada por cada artículo.

- Los eventos reflejan cada acción/actividad dentro de cada caso del proceso (traza). Continuando con el ejemplo anterior, los eventos estarían representados por cada revisión realizada a cada artículo. Dentro de un log de eventos, el atributo que determina cada evento toma el nombre de concept:name.
- De entre los posibles atributos que puedan tener una traza o evento es importante la existencia de una marca de tiempo o timestamp para cada evento. En el caso del ejemplo anterior, el timestamp sería la hora a la que se realizó una revisión. En un log de eventos el atributo que representa el timestamp es llamado time:timestamp.
- Otro de los posibles atributos existentes en cada traza o evento es el actor que realiza el evento. Siguiendo la misma linea que anteriormente, esto estaría representado por el nombre o id del editor que realiza una revisión. En un log de eventos esto es denominado org:resource.

```
<event>
    <int key="page_id" value="45307"/>
    <string key="org:resource" value="|JorgeGG|"/>
    <string key="concept:name" value="wikification,elaboration"/>
    <int key="bytes" value="1323"/>
    <string key="lifecycle:transition" value="complete"/>
    <date key="time:timestamp" value="2004-05-08T01:01:44.000+02:00"/>
    <int key="page_ns" value="0"/>
    <int key="revision_id" value="145370"/>
    <string key="contributor_id" value="2906"/>
</event>
```

Figura 3.7: Ejemplo de evento en XML

La figura 3.7 representa en XML el aspecto que tendría un evento dentro de un log de eventos. Como vemos, nos encontramos con org:resource, concept:name y time:timestamp además de otros atributos adicionales. Lifecycle:transition representa si el evento está comenzando o finalizando, en el caso del ejemplo, finalizando pues indica que ha sido completado.

### 3.3.2. Descubriendo los procesos


Figura 3.8: Descubrimiento de los procesos dentro de un log de eventos

La principal técnica y la más utilizada dentro de la minería procesos es el descubrimiento. El descubrimiento consiste en la generación de un modelo que represente los procesos existentes dentro del log de eventos. Las técnicas de descubrimiento parten de un log de eventos para generar un modelo sin ninguna información a piori [4]. Estas técnicas se basan en diferentes

algoritmos de minería que hacen uso de estadística y aprendizaje automático para buscar estos patrones existentes en los datos que representan un proceso y generar una salida que representa los procesos existentes en el log de eventos mediante una red de petri.

Dentro de la multitud de algoritmos de minería existentes dentro de la minería de procesos destacan:

- Alpha Miner: Fue el primer algoritmo de minería de procesos en desarrollarse y como tal tiene ciertos problemas graves. El principal problema es que es demasiado simple para representar procesos seguidos en log reales por lo que su interés es principalmente teórico. Examina las relaciones entre los diferentes eventos generando un modelo donde cada transición representa una tarea observada.
- Minero heurístico: Se trata de una mejora respecto al algoritmo Alpha Miner. El minero heurístico se centra en el flujo de control considerando solo el orden de los eventos dentro de cada evento [22]. Por lo cual un log de eventos con timestamp es necesario de cara a hacer uso de este algoritmo. Además, tiene en cuenta las frecuencias de aparición pudiendo filtrar comportamiento infrequente y permite saltarse las actividades individuales.
- Minero inductivo: El minero inductivo garantiza la generación de modelos 'sound' que traducido de modo literal implica modelos 'buenos'. 'Soundness' es una característica que implica que todo el comportamiento observado en el log de eventos puede ser reproducido por el modelo generado. Para esto, el algoritmo genera un arbol que representa el proceso, lo cual logra diviendo en log del modo más optimo posible hasta generar el árbol/es. Sin embargo, esto puede dar lugar a modelos dificiles de interpretar. Al tratar de generar modelos que representen todo el comportamiento observable en el log de eventos como mínimo, el minero inductivo puede recurrir a modelos en forma de flor. Un modelo en forma de flor es aquel que permite cualquier tipo de comportamiento en base a un tipo dado de actividades.

### 3.3.3. Redes de Petri

Las redes de petri son un modelo abstracto y formal de mostrar un flujo de información [13]. Permiten mostrar el flujo que sige un proceso de principio a fin representando así los proceso descubiertos en un log de eventos.

**Definición de una red de petri:** *Una red de petri es una tupla N = (P, T, F) donde P es el conjunto de lugares, T de transiciones,* $P \cap T = 0$ *y* $F \subseteq (PxT) \cup (TxP)$ *la relación del flujo* [1]

Con esto en cuenta, en la imagen 3.9 podemos ver una red de petri (P, T, F) donde *P = {Start, P1, end}, T = {T1, T2, T3} y F = {(Start, T1), (Start, T2), (T2, P1), (P1, T3), (T3, End), (T1, end)}*. Así, la red comienza en el lugar Start y finaliza en End pasando por las diferentes transiciones. El flujo podría ser o bien de Start a T1 y de ahí a End o bien de Start a T2 y hasta llegar a End para finalizar el proceso. Esto sería traducido en que el proceso representado por la red de la figura escenifica dos posibles caminos, o bien el proceso sigue el camino de la acción determinada en la transición T1 o bien sigue el camino

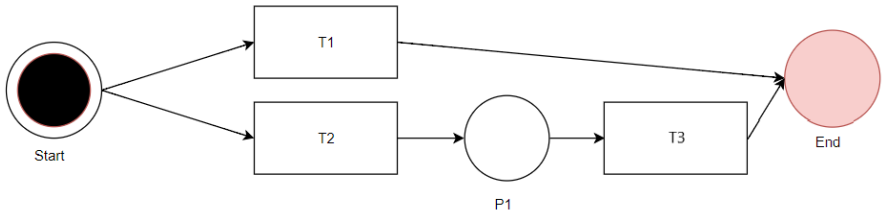



Figura 3.9: Ejemplo de una red de petri

compuesto por las transiciones T2 y T3 pero en ningún caso ambos caminos simultaneamente.

No obstante, las redes de petri generadas no suelen representar el posible proceso existente al 100 % incluso en el caso del minero inductivo. Es por esto que existe una métrica llamada fitness la cual mide en que medida el log de eventos puede ser reproducido en el modelo representado por la red de petri. Es decir, que porcentaje de todos los eventos y casos observados en el log son representados y pueden ser reproducidos en la red. Un valor de 0 implicaría que la red generada no representa en absoluto el log de eventos y de 1 que todo comportamiento observado en el log de eventos puede reproducirse en la red de petri obtenida.

### 3.3.4. De la minería de procesos a la minería social

Para complementar un análisis realizado mediante minería de procesos para descubrir los procesos existentes dentro de un log de eventos, se pueden aplicar técnicas de minería social para descubrir también las relaciones existentes entre los diferentes actores del log.

Lás técnicas de minería social hacen uso de técnicas de sociometría y de análisis de redes sociales [3]. La utilidad que tienen son la posibilidad de observar y conocer las posibles relaciones existentes entre los diferentes actores (org:resource) que aparecen a lo largo de un log de eventos. Estos algoritmos son implementados en ProM mediante la librería basada en las métricas establecidas por Wil M.P. van der Aalst [2]. Representan las relaciones existentes en forma de grafo donde cada cada es un nodo y cada arista una relación, dependiendo el peso de la intensidad de esta relación.

Se pueden hacer uso de diferentes métricas para obtener las relaciones entre los diferentes autores en un log:

1. Handover of Work: Se define como Handover Of Work como el relevo en el trabajo de un individuo $i$ a un individuo $j$ si hay dos actividades subsecuentes entre ellos dentro de un log de eventos. [2] Es decir, si después de editar $i$ edita $j$ se establece una relación entre ellos. Estos individuos $i$ y $j$ son representados mediante nodos y su conexión mediante aristas, variando el peso en función de lo fuerte que sea la relación entre ellos.

2. Subcontracting: Subcontracting cuenta el número de veces que un individuo $j$ ejecuta una actividad entre dos actividades ejecutadas por el individuo $i$ [2]. Es decir, si $i$ edita, $j$ edita e $i$ vuelve a editar, se establece una relación de $i$ a $j$. Así, siendo $i$ y $j$ representado como nodos su relación se establece mediante una arista común cuyo peso dependerá de las veces que suceda la relación.

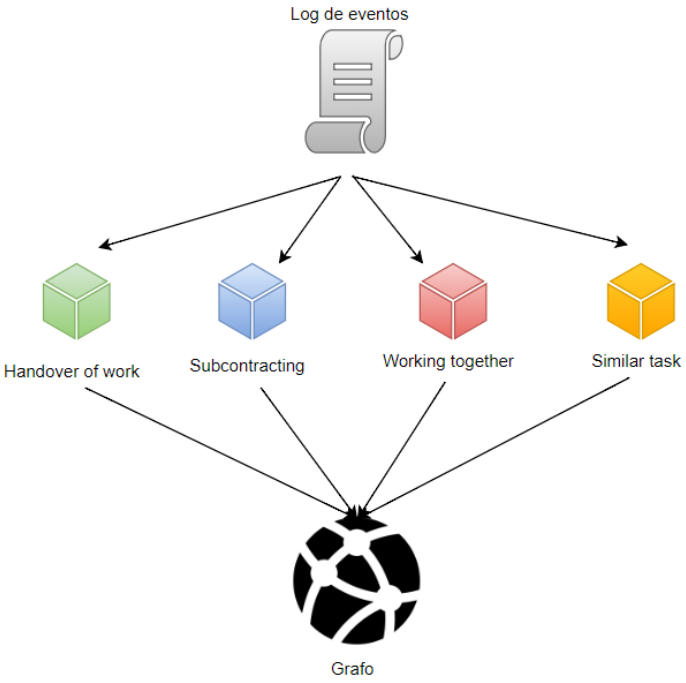



Figura 3.10: Estructura de la minería social en ProM

3. Working-Together: Working Together simplemente tiene en cuenta la frecuencia con la cual dos actores realizan actividades en el mismo caso dentro del log de eventos [2]. De este modo, dos actores que trabajen en el mismo caso estarán relacionados y el peso de su arista dependerá de la frecuencia con la que suceda siendo representados por nodos.

4. Similar Task: Similar Task se centra en el tipo de actividades que realizan los diferentes actores. Así, se establecerá una relación entre actores que estén realizando un tipo similar de actividades, el peso de su arista al representar los actores mediante nodos dependerá de la similud de las tareas realizadas. La similitud entre las tareas realizadas por los diferentes actores se puede calcular mediante diferentes métricas como la tradicional distancia euclidea.

Con toda esta información la base teórica necesaria para aplicar la minería social y de procesos es suficiente. De esta manera en los próximos capitulos se definirá y aplicará la minería social y de procesos contextualizandola dentro de los objetivos de este proyecto.

# Capítulo 4

# Tecnologías y herramientas

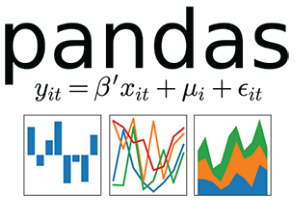
A continuación, se establecen las tecnologías y herramientas usadas en este proyecto así como la motivación tras su elección.

## 4.1. Tecnologías

Python ha sido escogido como lenguaje de programación para los diferentes scripts necesarios a lo largo del proyecto por su facilidad de uso y la enorme cantidad de librerías y recursos existentes para manejo de datos así como para técnicas de minería de datos.

Anaconda ha sido escogido para servir de base para hacer uso de Python y manejar las diferentes librerías por su simplicidad y facilidad a la hora de gestionar las librerias y versiones de python así como sus multiples utilidades como la opción de hacer uso de Jupyter.


Para realizar tareas aprendizaje automático se ha utilizado Jupyter en conjunto con Anaconda. Los Jupyter Notebook dan lugar a un entorno de programación donde además se puede añadir texto generando un código interactivo ejecutable sección a sección aumentando mucho la legibilidad del código y favoreciendo la posibilidad de realizar un análisis al mismo tiempo que se programa.




Por otro lado, se ha hecho uso en los diferentes scripts del proyecto (11) de las librerías Matplotlib para las visualizaciones, Pandas para el manejo de datos así como Numpy y Scikit-learn para aplicar machine learning. Estas 4 librerías, dan un vuelco a python y hacen del machine learning y el tratamiento de datos una tarea fácil.

- Matplotlib permite generar gráficas de alta calidad con tan sólo unas lineas de código, simplificando en gran parte la tarea.
- Numpy se trata de un paquete fundamental de python para la computación científica. Aporta un tipo de vectores N-dimensionales con muchas funcionalidades además de contar con muchas funciones de álgebra lineal útiles, entre otras cosas.
- Pandas es una librería centrada en el manejo de datos. Se encuentra construida sobre Numpy y Matplotlib y hace de las tareas de visualización y manipulación de datos algo sencillo.
- Scikit-learn es la librería por excelencia para aplicar algoritmos de machine learning. Aporta herramientas simples y eficientes para aplicaciones de data mining y es de código abierto. Está desarrollada sobre NumPy, SciPy y Matplotlib.

.

Para almacenar y organizar el código en un servicio externo se ha hecho uso de Github. Github aporta facilidades para realizar gestión de versiones y mantenimiento del código además de salvaguarda en caso de necesitar revertir cambios.


Aunque no se menciona a lo largo del proyecto, pues solo ha sido utilizado para guardar los datos, se ha hecho uso de una base de datos SQL de SQLite por su ligereza y facilidad de uso a la hora de importar archivos y realizar consultas rápidas.

## 4.2. Herramientas

### 4.2.1. ProM tools


Figura 4.1: Interfaz de ProM tools 6.8

Para aplicar técnicas de minería de procesos se hace uso de la herramienta ProM tools. ProM es la herramienta lider en minería de procesos. Se trata de un framework que permite una alta variedad de técnicas de minería de procesos implementado en Java y disponible en `www.processmining.org` [10]. La versión 6.8 ha sido escogida en lugar de la última disponible pues el foco de audiencia de esta versión es la investigación al garantizar que no se realizarán

cambios que puedan afectar a los resultados que ya hayan sido publicados u obtenidos. [14].

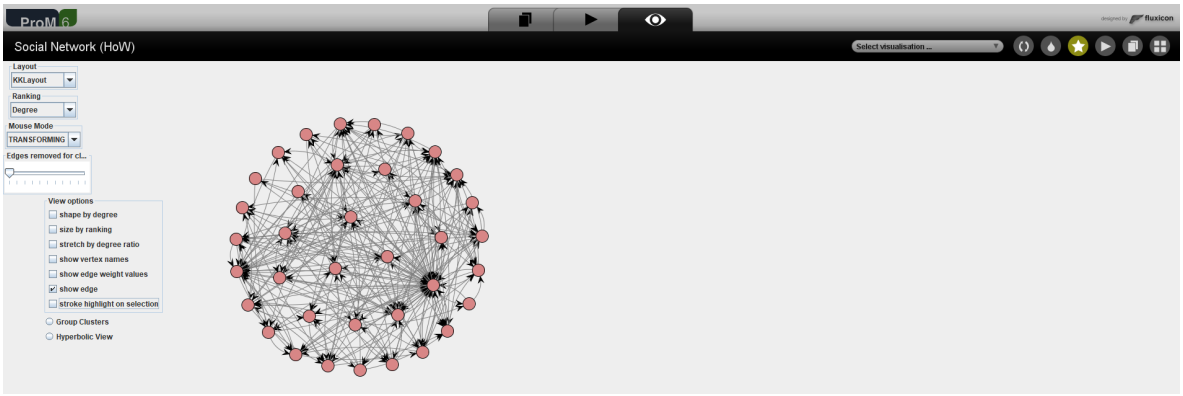



Figura 4.2: Visor de grafos de ProM tools 6.8

ProM es un programa de sencillo manejo con una interfaz bástante básica, tal y como vemos en la imagen 4.1 nos encontramos con una pestaña izquierda donde seleccionamos los datos de entrada, una pestaña central donde seleccionamos la técnica o algoritmo que queremos aplicar y la pestaña derecha donde aparecerán los ficheros de salida tras aplicar la técnica/algoritmo seleccionado. Nos permite tanto importar como exportar ficheros para poder utilizarlos en otro momento y cuenta con un visor integrado para poder ver gráficamente las redes de petri generadas o las relaciones entre los actores. Sin embargo, este visor cuenta con limitaciones como pocas opciones de organización y en el caso de la visualización de grafos no poder regular el tamaño y grosor de los nodos y aristas manuel supone una desventaja.

# Capítulo 5

# Procesamiento de los historiales de revisión

Durante este capítulo, se detallará el proceso seguido desde la descarga del historial de revisiones de un artículo en Wikipedia, hasta la generación de los datos a analizar mediante la minería de procesos. Así, el proceso consta de un número reducido de pasos, representados en la figura 5.1:

- Descarga de datos: descarga inicial del conjunto de datos para el desarrollo del proyecto de Wikipedia: los historiales de revisión.
- Extracción de información: Se extrae solamente información útil de los historiales de revisión.
- Obtención de características: Se comparan las diferentes revisiones en cada artículo, generando un conjunto de atributos que servirán para determinar las intenciónes tras cada revisión (recuadrado en rojo en el esquema 5.1).
- Cambio de formato: Este paso es necesario para adecuar el formato de los archivos resultado del paso previo al requerido en el próximo punto (recuadrado en verde en el esquema 5.1).
- Generación y análisis del modelo predictivo: Generación de diferentes modelos predictivos en busca de aquel con el rendimiento más óptimo en la tarea de predicción de intenciones.

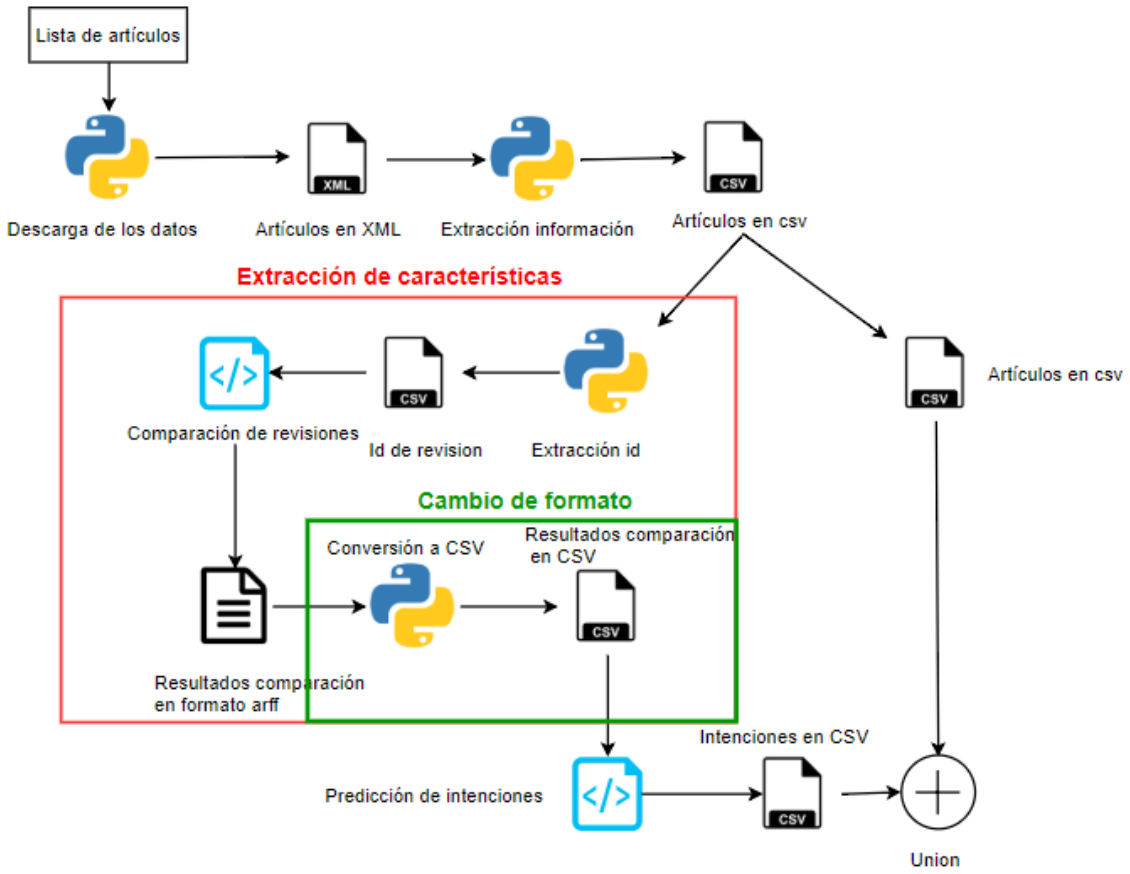



Figura 5.1: Proceso seguido durante el procesamiento de los historiales de revisión de Wikipedia

## 5.1. Descarga de datos

El primer paso de todos es la descarga de los datos iniciales. Estos datos, pueden ser obtenidos de Wikipedia ya sea en forma de un artículo específico o un conjunto de artículos en un archivo de texto plano. Para esto, se hace uso del script wiki_dump_downloader.py (11). Este script ha sido desarrollado en base a `https://phabricator.wikimedia.org/diffusion/PWBC/browse/master/scripts/maintenance/download_dump.py`. La diferencia principal con el código en el cual el script está basado, es la eliminación de las dependencias con la libreria PyWiki, además de la adición de numerosas utilidades, como la selección del idioma de la Wiki escogida, la posibilidad de pasar por parámetro una lista de artículos a descargar, y la unión de las diferentes partes descargadas (pues los artículos son descargados separados en diferentes fragmentos de menor tamaño) en una sola.

A priori, este script sirve, por tanto, para descargar el historial de revisiones de un artículo específico o de una lista de artículos en una Wikipedia de un lenguaje específico (e.j Wikipedia española). El archivo descargado tendrá formato XML y será un archivo de elevado peso. Es

por esto, que se necesita convertir el formato del archivo en otro formato más facil de manejar, lo que conduce al siguiente paso de este proceso: la extracción de información de estos archivos XML obtenidos.

## 5.2. Extracción de información

Para extrar información del XML descargado se utiliza el analizador desarrollado por Abel Serrano Juste, wiki_dump_parser.py (11) como parte de un conjunto de scripts para Wikipedia. Este script recibe como entrada el conjunto histórico de revisiones de un artículo en formato XML y lo convierte en un CSV legible con información útil. Esta información está compuesta por los siguientes atributos:

1. Id de artículo.
2. Título de artículo.
3. Id de revisión.
4. Timestamp.
5. Id editor.
6. Nombre editor (nombre de usuario).
7. Bytes del artículo tras la edición.

## 5.3. Obtención de características

El objetivo es obtener las diferentes características de cada revisión en comparación con la revisión previa como por ejemplo, que palabras han sido eliminadas/añadidas o de que manera el formato ha sido editado. El flujo de trabajo a seguir se puede observar en el recuadro rojo del esquema 5.1.

Para ello, hacemos uso del proceso seguido en la investigación 'Identificando intenciones semánticas en las revisiones de Wikipedia' ([24]). En la investigación, se crea una taxonomía de intenciones existentes tras cada revisión. Estas intenciones, se pueden determinar gracias los cambios realziados en cada revisión.

Para lograr estas características, hacen uso de los resultados obtenidos mediante la aplicación de un comparador online de la API de Wikipedia, en combinación con un conjunto de scripts realizados por los autores (11).

Sin embargo, ha habido que realizar modificaciones para conseguir adaptar su flujo de trabajo a este proyecto. En primer lugar, ha habido que actualizar librerías obsoletas y adaptar los parámetros de entrada para poder añadir lenguaje de la Wikipedia utilizada.

Gracias a esto, es posible hacer una comparación entre las diferentes revisiones generando un fichero de salida que cuenta con 208 atributos. Estos 208 atributos se encuentran estructurados en tres grupos. El primero se compone de atributos asociadas al propio editor del

artículo, el segundo se compone de 16 atributos en base al comentario escrito por el editor respecto a su revisión, y el tercer grupo consta de los restantes 198 atributos y consiste de detalles de la comparación de Wikipedia [24]. La entrada de este conjunto de scripts requiere de un conjunto de parejas de id's de revisión e intenciones asociadas. Dado que no se cuenta con esas intenciones asociadas en nuevos artículos que se acaben de descargar, se asigna siempre 0 en la intención por defecto.

Así, una vez se tiene el fichero resultado del analizador previamente mencionado[1], se utiliza el script llamado revision_id_extractor.py (11). Su función es extraer el id de revisión de cada revisión de los artículos seleccionados generando un nuevo archivo, compuesto por el id de revisión y una etiqueta de 0 asignada automáticamente como intención. Con este archivo, se llama al conjunto de scripts para generar las 207 características en base a las diferencias entre revisiones.

Es importante mencionar que debido a que el comparador de Wikipedia se realiza online en sus propios servidores, este proceso conlleva un elevado número de horas para un conjunto grande de revisiones, siendo una limitación real de cara a analizar comunidades muy grandes.

## 5.4. Cambio de formato

Debido a que la salida del comparador utilizado tiene formato arff, es necesario realizar un cambio de formato a csv por motivos de simplicidad de cara a futuras secciones. Para ello, se hace uso del script arffToCsv.py (11). Se trata de un script ligero y simple que cumple con su función.

El siguiente y último paso es la elaboración de un modelo predictivo.

## 5.5. Generación y análisis de modelos predictivos

El objetivo principal de esta subsección es explicar el proceso seguido en la generación de un modelo predictivo para asignar una o varias intenciones a cada revisión realizada al artículo objeto del estudio, en base a la taxonomía semántica de intenciones previamente introducida. Para esto, se cuenta con un conjunto de revisiones elaborado para la citada investigación ([24]) que será utilizado como entrenamiento a la hora de determinar el mejor modelo más eficiente.

Con el objetivo de obtener un modelo predictivo que sea capaz de generar los resultados más precisos posibles, se entrenarán diferentes modelos basados en variados algoritmos de clasificación tales como el bosques aleatorios o máquinas de soporte de vectores o $k$-vecinos más cercanos. Además, se aplicarán diferentes técnicas para optimizar los resultados como la ingeniería de características, el sobremuestreo, o la normalización de los datos.

Una vez creado el mejor modelo predictor posible, este es exportado de cara a automatizar el proceso para futuros artículos que se quieran analizar.

[1] wiki_dump_parser.py (11)

| feats_0 | feats_1 | feat_2 | feats_3 | feats_4 | feats_5 | ... | other | wikification | vandalism | simplification | elaboration | verifiability | process | clarification | disambiguation | point-of-view |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 741692138 | 0.0 | 0.0 | -1.0 | 0.0 | 555.0 | ... | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 710764506 | 0.0 | 0.0 | -1.0 | 0.0 | 3.0 | ... | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 711588802 | 0.0 | 0.0 | -1.0 | 0.0 | 9.0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 709526386 | 0.0 | 0.0 | -1.0 | 0.0 | 326.0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 713098731 | 0.0 | 0.0 | -1.0 | 0.0 | 190.0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Tabla 5.1: Vistazo general de los datos

### 5.5.1. Datos iniciales

Inicialmente contamos con el conjunto de datos generado para la investigación *'Identificando intenciones semánticas en las revisiones de Wikipedia'*. Cuenta con 5684 revisiones de artículos de Wikipedia con intenciones asignadas a mano [24].

El conjunto de revisiones, como se puede observar en la tabla 5.1 está formado por 208 atributos y 14 etiquetas binarias diferentes, una por cada intención.

Los atributos destacan por su dificil interpretabilidad. Son de tipo escalar y con nombres que no aportan ninguna información a priori. Además, para facilidad en el futuro uso de las predicciones generadas, el id de revisión ha sido añadido como feats_0. Por ultimo, todos los valores de los atributos son correctos y no hay valores perdidos.

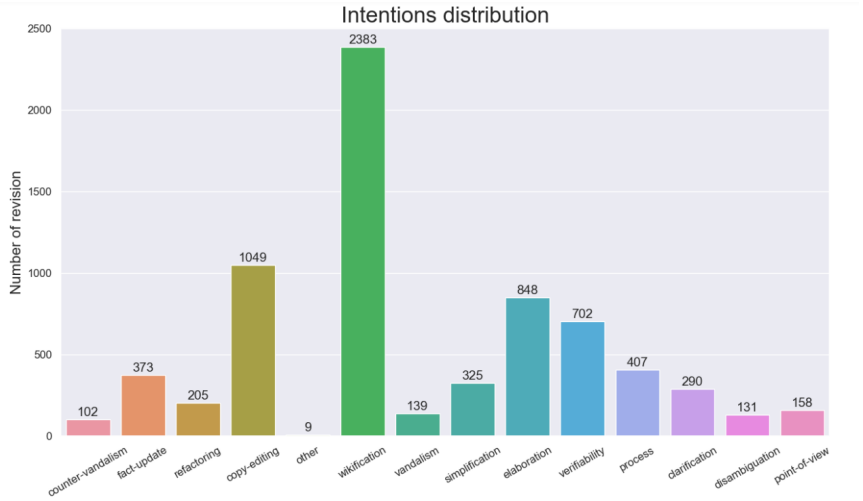



Figura 5.2: Distribución de intenciones en el conjunto de datos

Una vez que se conoce el formato de los datos, el próximo paso es realizar un análisis exploratorio de los mismos. En la figura 5.2, podemos ver la distribución de las intenciones en el conjunto de datos.
Se observa que no hay una distribución equitativa de intenciones. Intenciones como other, counter vandalism, disambiguation, point of view o vandalism se encuentran en proporciones inferiores al 3 % mientras que Wikification aparece en el 41 % de las revisiones. El caso más serio se da en la intencion other. El objetivo de esta intención es determinar que en esa revision se ha hecho algo diferente a todo lo demás. Sin embargo, dada su frecuencia de aparición, podría no ser considerada.

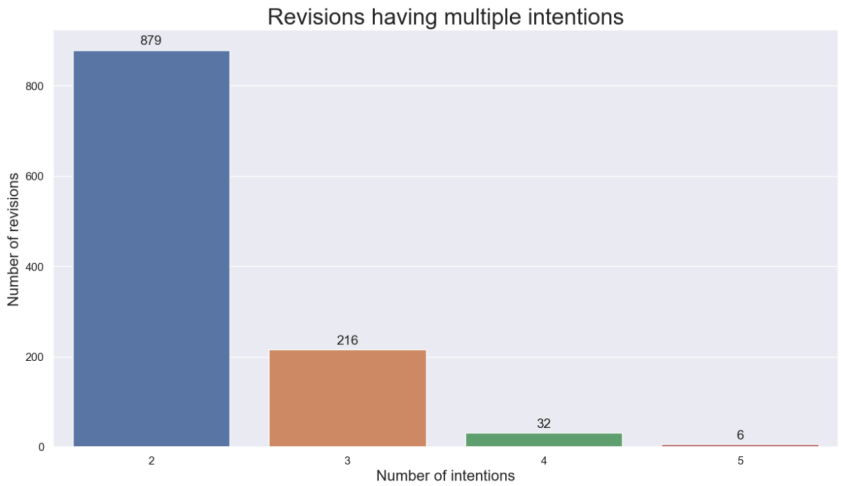



Figura 5.3: Revisiones con múltiples intenciones

| | intention | count | % of appearance |
|---|---|---|---|
| 0 | counter-vandalism | 102 | 1.792619 |
| 1 | fact-update | 373 | 6.555360 |
| 2 | refactoring | 205 | 3.602812 |
| 3 | copy-editing | 1049 | 18.435852 |
| 4 | other | 9 | 0.158172 |
| 5 | wikification | 2383 | 41.880492 |
| 6 | vandalism | 139 | 2.442882 |
| 7 | simplification | 325 | 5.711775 |
| 8 | elaboration | 848 | 14.903339 |
| 9 | verifiability | 702 | 12.337434 |
| 10 | process | 407 | 7.152900 |
| 11 | clarification | 290 | 5.096661 |
| 12 | disambiguation | 131 | 2.302285 |
| 13 | point-of-view | 158 | 2.776801 |

Figura 5.4: Distribución de las intenciones en porcentajes

Por otro lado, en lo que a revisiones con multiples intenciones se refiere, nos encontramos con que la figura 5.3 muestra una cantidad relativamente alta de ediciones con más de una intención. Sin embargo, según aumenta el número de intenciones por revisión, decrece exponencialmente la frecuencia a la que esto sucede. Sólo un 0,1 % de las revisiones tienen cinco intenciones, alcanzando un 15 % revisiones con dos intenciones. Así, el 80 % de todas las revisiones del dataset solo poseen una intencionalidad.

En base a estos datos obtenidos, de cara a obtener el modelo predictor más optimo posible, se han aplicado dos tipos de clasificación:

1. Clasificación binaria: La clasificación binaria es aquella en la que la etiqueta toma los valores discretos 0 o 1. De esta manera, el objetivo es la generación de un modelo clasificador a medida para cada intención. Es decir: un modelo para predecir wikification, otro para vandalism, etc... El resultado final constaría de trece modelos predictivos diferentes donde la etiqueta, en cada intención, toma el valor 0 en caso de no encontrar esa intención en la revisión o 1 en caso positivo.

2. Clasificación multi-etiqueta: La clasificación multi-etiqueta es aquella en la que la etiqueta puede tomar un conjunto discreto de valores (por ejemplo: $['pajaro, pez, reptil']$) que pueden aparecer combinados. En este caso, un solo modelo puede predecir todas las intenciones de la taxonomía en una revisión al mismo tiempo, siendo las etiquetas, el conjunto de intenciones.

Esencialmente, el objetivo es realizar una clasificación donde se obtengan las posibles diferentes intenciones que suceden naturalmente en cada revisión.

Para poder evaluar cada modelo y compararlos en justas condiciones, hay que sentar unas métricas de base.

### 5.5.2. Métricas de evaluación

Las métricas ha utilizar serán aquellas introducidas durante el capítulo 3 de este documento en la sección de aprendizaje automático:

- Precisión: Representa el número positivos verdaderos dividido entre la suma de los positivos verdaderos y los positivos falsos.
- Sensibilidad: Representa el número de positivos verdaderos entre los positivos verdaderos y los negativos falsos.
- F1: Se trata de una combinación de precisión y sensibilidad.
- Matriz de confusión: Sirve para describir el rendimiento de un modelo predictivo mostrando visualmente los PV, PF, NV y NF. Las matrices de confusión mostradas durante este capítulo, **se encuentran limitadas** en su capacidad expresiva. La librería que permite representarlas no permite que por cada ejemplo pueda haber más de una etiqueta. Como hemos visto, hay revisiones con más de una intención. Esto hace que solo se escoja una sola intención por revisión, eliminando, por tanto, parte de los resultados. Por este motivo, solo sirven a modo de orientación y las métricas que muestran ( % de clasificaciones erróneas) son incorrectas al no tener en cuenta todos los datos y pueden no corresponder con lo observado en precisión, sensibilidad y F1.

### 5.5.3. Clasificación binaria

La clasificación binaria es aquella en la que la etiqueta toma los valores discretos 0 o 1. De esta manera, el objetivo es la generación de un modelo clasificador a medida para cada intención. Es decir: un modelo para predecir wikification, otro para vandalism, etc... El resultado final constaría de trece modelos predictivos diferentes donde la etiqueta, en cada intención, toma el valor 0 en caso de no encontrar esa intención en la revisión o 1 en caso positivo.

Como consideraciones iniciales, se ha eliminado la intencion other al no aportar información extra y se ha eliminado la aleatoriedad en la generación de muestras y modelos para facilitar la reproducibilidad.

Para validar los diferentes modelos se hace uso de la validación cruzada con cuatro iteraciones.

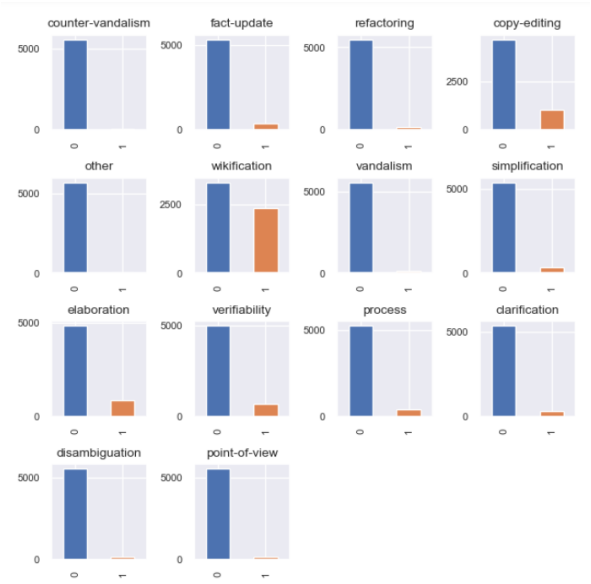



Figura 5.5: Proporcion de positivos y negativos por intencion

Así, el conjunto de datos ha de ser divido por el número de intenciones diferentes. En este caso se generan trece etiquetas diferentes, uno por cada intención. De esta manera quedaría la proporción de positivos y negativos en cada intención observada en la figura 5.5.

Las técnicas de aprendizaje automático y los modelos que se han aplicado han sido los siguientes:

1. Clasificadores de prueba: Mediante un clasificador que se base en reglas simples prefijadas, obtenemos unas métricas que establencen un rendimiento mínimo a superar por el resto de clasificadores.

2. Bosques aleatorios: Se aplicarán algoritmos de bosque aleatorio que hace uso de n árboles de decisión para generar las predicciones.

3. Máquinas de soporte de vectores: Se modelarán máquinas de soporte de vectores.

4. Sobremuestreo: El objetivo es conseguir un conjunto de datos donde las intenciones se encuentren divididas equitativamente.

5. Ingeniería de características: Se tratará de optimizar los atributos del conjunto de datos para facilitar la tarea de predicción a los diferentes modelos.

### Clasificadores de prueba

El motivo de la creación de un modelo con un clasificador de prueba es el de establecer unos mínimos valores de rendimiento para los demás clasificadores. Así, el clasificador seleccionado es el clasificador de prueba de Scikit-learn, el cual hace predicciones basándose en reglas simples [17]. En este caso, la estrategia que sigue es llamada 'estratificación' que predice basándose en la proporción de las etiquetas. Los resultados obtenidos son los observables en

| | Intention | Precision | Recall | F1 |
|---|---|---|---|---|
| 0 | counter-vandalism | 0.010864 | 0.009995 | 0.010411 |
| 1 | fact-update | 0.034535 | 0.029457 | 0.031793 |
| 2 | refactoring | 0.041667 | 0.039027 | 0.040303 |
| 3 | copy-editing | 0.185020 | 0.160165 | 0.171698 |
| 4 | wikification | 0.413157 | 0.395302 | 0.404032 |
| 5 | vandalism | 0.007577 | 0.007144 | 0.007354 |
| 6 | simplification | 0.052085 | 0.046147 | 0.048936 |
| 7 | elaboration | 0.129831 | 0.110846 | 0.119590 |
| 8 | verifiability | 0.147573 | 0.121075 | 0.133017 |
| 9 | process | 0.078488 | 0.066346 | 0.071908 |
| 10 | clarification | 0.034611 | 0.031055 | 0.032736 |
| 11 | disambiguation | 0.000000 | 0.000000 | 0.000000 |
| 12 | point-of-view | 0.013506 | 0.012814 | 0.013151 |

Figura 5.6: Resultados por intencion del clasificador de prueba

la tabla de la imagen 5.6.

En general, son valores muy pobres. Las intenciones con menor frecuencia de aparición son las más perjudicadas, como se puede ver en los valores de precision y sensibilidad de point-of-view, disambiguation o counter-vandalism. Observando la matriz de confusión, se ve como efectivamente la matriz principal se encuentra compuesta de valores muy reducidos, explicando los valores encontrados en la tabla.

### Bosques aleatorios

Los resultados por intención de los bosques aleatorios se pueden consultar en la figura 5.9.

En general, se observa que los resultados son mejores que los obtenidos por los clasificadores de prueba. Los valores de precisión son altos. Sin embargo, la sensibilidad es generalmente baja, con excepciones. Se puede ver claramente como aquellas intenciones en minoría son las que poseen valores de sensibilidad muy bajos. Esto, indica que el clasificador está asignando más negativos de los que debería, mientras que hay un valor muy bajo de positivos falsos. En lineas generales, son modelos con una capacidad predictora débil. Observando su matriz de confusión vemos como su diagonal principal obtiene tonos más oscuros que el clasificador previo.

### Máquinas de soporte de vectores

Se han utilizado máquinas de soporte de vectores lineales. Se ha aplicado un escalado de los datos los datos de modo que sus valores se encuentren más uniformemente repartidos. Los resultados pueden consultarse en la tabla de la imagen 5.10.

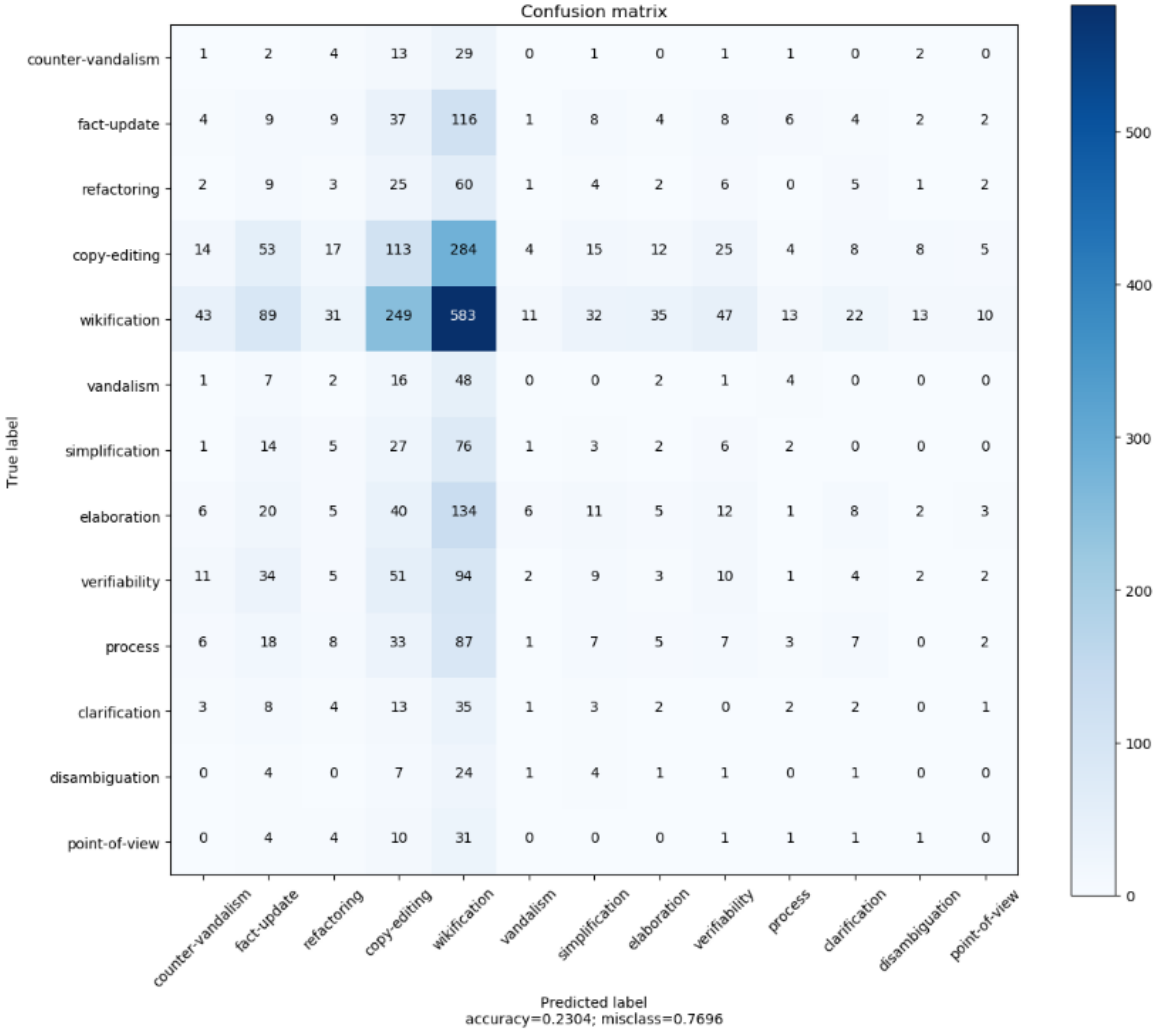



Figura 5.7: Matriz de confusión del clasificador de prueba binario

A simple vista, puede verse que los clasificadores son similares a los bosques aleatorios: altos valores de precision en ciertas intenciones pero con valores de sensibilidad generalmente bajos. En su matriz de confusión, se ve que, efectivamente, la diagonal principal es ligeramente más debil que la del bosque aleatorio.

| Algoritmo | Precision | Sensibilidad | F1 micro |
|---|---|---|---|
| Algoritmo de prueba | 0.213536 | 0.191648 | 0.202001 |
| Bosque aleatorio | 0.742346 | 0.467098 | 0.573401 |
| Máquina soporte vectores | 0.648503 | 0.371485 | 0.472376 |

Tabla 5.2: Micro media de los resultados de los tres algoritmos

En lineas generales ninguno de los tres algoritmos ha tenido un gran rendimiento. Mientras que son precisos, sus bajos valores de sensibilidad los convierten en clasificadores no fiables

| | Intention | Precision | Recall | F1 |
|---|---|---|---|---|
| 0 | counter-vandalism | 0.868101 | 0.343092 | 0.485771 |
| 1 | fact-update | 0.754571 | 0.259906 | 0.377850 |
| 2 | refactoring | 0.694371 | 0.185773 | 0.285627 |
| 3 | copy-editing | 0.722279 | 0.421365 | 0.531374 |
| 4 | wikification | 0.771595 | 0.623140 | 0.686864 |
| 5 | vandalism | 0.796960 | 0.330679 | 0.466420 |
| 6 | simplification | 0.593725 | 0.289645 | 0.359494 |
| 7 | elaboration | 0.694357 | 0.520044 | 0.594568 |
| 8 | verifiability | 0.786536 | 0.626778 | 0.696530 |
| 9 | process | 0.832994 | 0.343964 | 0.485214 |
| 10 | clarification | 0.675531 | 0.069187 | 0.107507 |
| 11 | disambiguation | 0.883366 | 0.260859 | 0.373689 |
| 12 | point-of-view | 0.427582 | 0.063601 | 0.102670 |

Figura 5.8: Resultados de los bosques aleatorios

pues asignan un gran número de negativos falsos. En la tabla 5.2 vemos los valores calculados mediante una micro-media para cada conjunto de clasificadores. Claramente, el que mejor rendimiento ha tenido es el conjunto de bosques aleatorios, superando a los clasificadores de prueba y a las máquinas de soporte de vectores en todas las métricas.

No obstante, los valores de sensibilidad de los bosques aleatorios son alarmantemente bajos. Sin embargo, y debido a que en comparación tienen el mejor rendimiento, será el algoritmo utilizado a partir de ahora en la clasificación en búsqueda de su optimización.

Llegados a este punto, está claro que la distribución poco equitativa de las intenciones en los datos está generando un gran perjuicio, dificultando mucho la tarea de predicción. Por ello, se van a aplicar técnicas de sobremuestreo en búsqueda de mejores resultados de los bosques aleatorios.

### Sobremuestreo

Para solventar la falta de equilibrio, hay numerosas técnicas de muestreo. Tal y como se ha introducido en el capítulo 3 el sobremuestreo aumenta la frecuencia de aparición de las etiquetas minoritarias. En este caso debido al tamaño de los datos, se ha decidido hacer uso de una técnica de sobremuestreo llamada ADASYN.

ADASYN hacía uso de una distribución con pesos para cada etiqueta en minoría que resulta más dificil de predecir, así, genera ejemplos de esa clase en minoría mejorando los resultados en la clasficación, reduciendo el bias. [11]

Para hacer uso de ADASYN se implementa el sobremuestreo en una tubería proporcionaad por la libería imblearn ya que transforma solo los datos de entrenamiento durante el

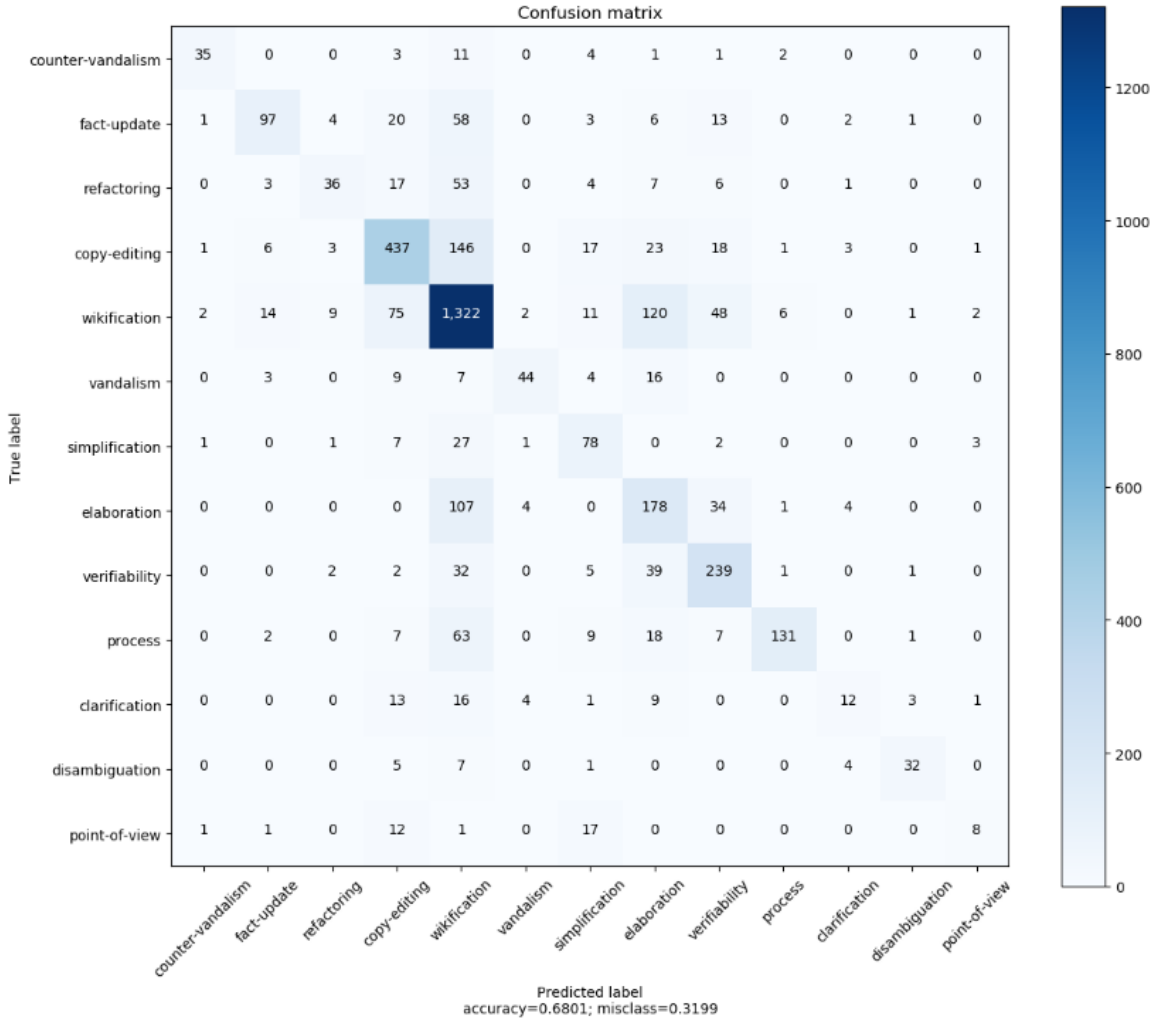



Figura 5.9: Matriz de confusión de los bosques aleatorios

proceso de la validación cruzada, dando lugar a que los datos de prueba estén libre de muestras creadas con esta técnica.

Los resultados obtenidos por los bosques aleatorios con sobremuestreo pueden verse en la tabla de la figura 5.12.

Vemos como sus valores de precisión se han reducido, pero al mismo tiempo sensibilidad ha aumentado así como F1. La bajada de precision en sí, no es positiva. Sin embargo, la proporción en la que ha bajado en contraposición con la subida obtenida en sensibilidad hace que el cambio sea positivo. Esto es debido a que, ahora, el conjunto de entrenamiento se encuentra equilibrado. Gracias a esto, el clasificador asigna más positivos mientras que anteriormente asignaba negativos con mayor facilidad.
Esto, también puede verse en la matriz de confusión, cuya diagonal principal se refuerza ligeramente, obteniendo valores más altos en todas las intenciones.

| | Intention | Precision | Recall | F1 |
|---|---|---|---|---|
| 0 | counter-vandalism | 0.586231 | 0.441529 | 0.495271 |
| 1 | fact-update | 0.527069 | 0.198248 | 0.280371 |
| 2 | refactoring | 0.487456 | 0.258820 | 0.335521 |
| 3 | copy-editing | 0.582443 | 0.246944 | 0.328946 |
| 4 | wikification | 0.740611 | 0.542971 | 0.612757 |
| 5 | vandalism | 0.664106 | 0.286994 | 0.380450 |
| 6 | simplification | 0.489223 | 0.178650 | 0.248467 |
| 7 | elaboration | 0.742780 | 0.380903 | 0.501048 |
| 8 | verifiability | 0.731040 | 0.594170 | 0.649355 |
| 9 | process | 0.383056 | 0.081122 | 0.133490 |
| 10 | clarification | 0.234822 | 0.051925 | 0.069583 |
| 11 | disambiguation | 0.416424 | 0.169703 | 0.205362 |
| 12 | point-of-view | 0.127472 | 0.057350 | 0.066970 |

Figura 5.10: Resultados por intencion de la máquina de soporte de vectores

En definitiva, el sobremuestreo ha tenido un efecto positivo por lo que se procederá a intentar mejorar los resultados obtenidos haciendo ingeniería de características y ajustando los hiperparámetros de los bosques aleatorios.

### Ingeniería de características

La ingeniería de características se compone de un conjunto de técnicas cuyo objetivo es mejorar el rendimiento de los modelos predictivos [20]. En este caso, las técnicas de ingeniería de características que se utilizan en el documento tienen un enfoque basado en el modelo.

El funcionamiento es simple, en base a un umbral de importancia dado, los propios bosques aleatorios decide la importancia de cada atributo y descartan todos aquellos con un nivel de importancia menor. El umbral suele ser un valor en base a variaciones de la media de importancia del conjunto de datos, como por ejemplo $0{,}75 * media$.Los resultados obtenidos pueden verse en la tabla de la imagen 5.15.

Los valores de precisión, sensibilidad y F1 son ligeramente más altos que antes de aplicar la ingeniería de características y se puede ver como el modelo desarrollado para cada intención hace uso de umbrales diferentes en cada caso. Los bosques aleatorios de intenciones como copy-editing o clarification tienen umbrales muy elitistas, donde sólo aquellos con un valor de importancia mayor a la media se mantienen, mientras que otros bosques aleatorios, como el de la intención process, hace uso de todos los atributos del conjunto de datos.

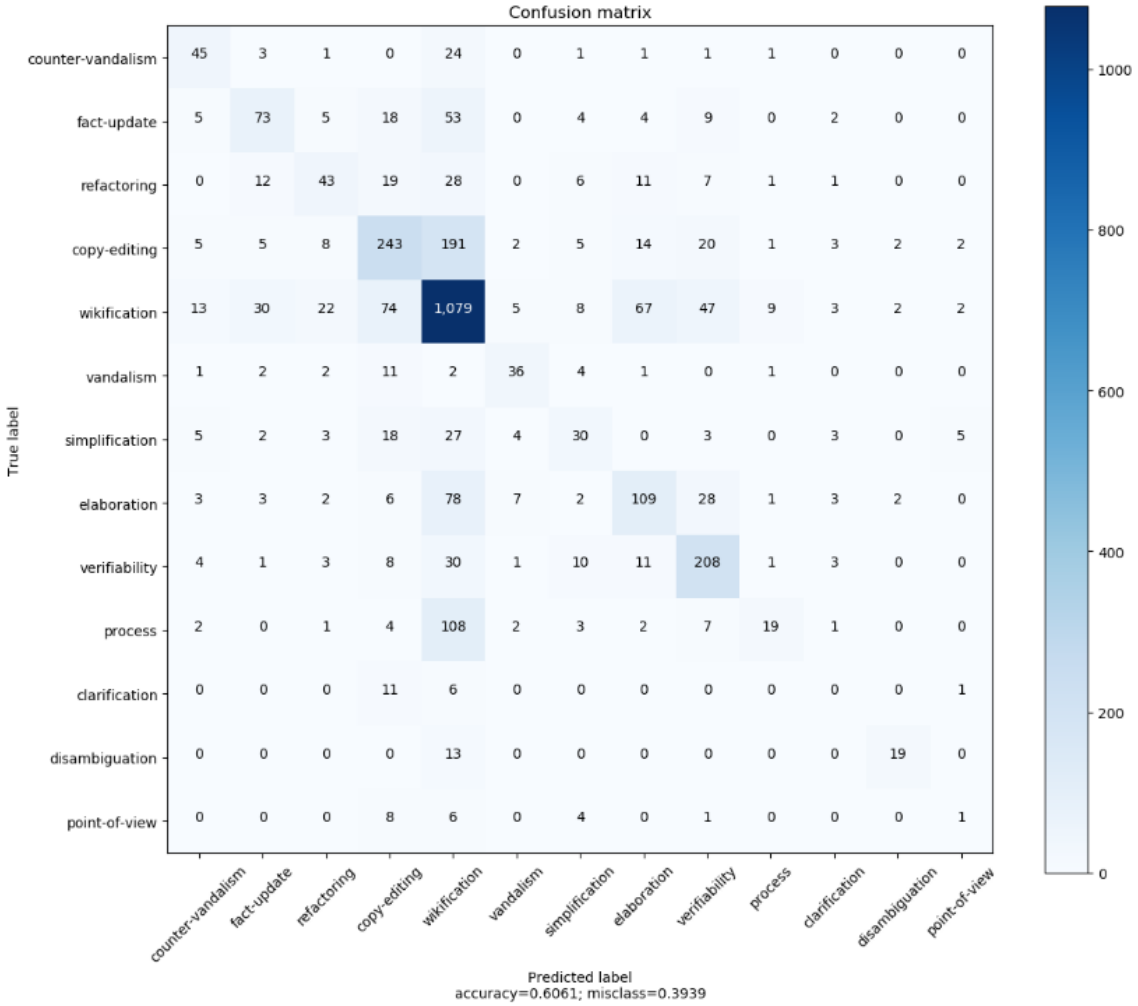



Figura 5.11: Matriz de confusión de la máquina de soporte de vectores

### Ajuste del modelo final

Finalmente, se ha decidido seguir haciendo uso de la ingeniería de características y el sobremuestreo pues toda mejora, aunque pequeña, es bienvenida.

El objetivo aquí, es ajustar los diferentes hiperparámetros posibles de los bosques aleatorios de modo que aumente la calidad de sus predicciones. En este caso, el parámetro a ajustar, además del umbral de la ingeniería de características, es el número de árboles de decisión que compone a cada bosque aleatorio. Las posibilidades propuestas para este valor van entre uno y 512 en potencias de dos principalmente. Los resultados obtenidos son los observables en la tabla de la figura 5.17.

Se aprecia una mejora en los resultados a simple vista, obteniendo valores relativamente altos de todas las métricas. Observando la matriz de confusión, vemos como su diagonal principal muestra valores visiblemente más altos que anteriormente. Claramente, algunas intenciones son clasificadas con mucho más exito que otras. Se puede observar que, en este caso,

| | Intention | Precision | Recall | F1 |
|---|---|---|---|---|
| 0 | counter-vandalism | 0.778281 | 0.489621 | 0.592346 |
| 1 | fact-update | 0.566401 | 0.428844 | 0.466220 |
| 2 | refactoring | 0.486070 | 0.376167 | 0.413354 |
| 3 | copy-editing | 0.611355 | 0.534810 | 0.569848 |
| 4 | wikification | 0.757257 | 0.682323 | 0.717202 |
| 5 | vandalism | 0.625896 | 0.416613 | 0.473872 |
| 6 | simplification | 0.492995 | 0.419452 | 0.376032 |
| 7 | elaboration | 0.648704 | 0.672175 | 0.659301 |
| 8 | verifiability | 0.730142 | 0.735034 | 0.732177 |
| 9 | process | 0.586111 | 0.442403 | 0.500836 |
| 10 | clarification | 0.502816 | 0.304112 | 0.318278 |
| 11 | disambiguation | 0.679155 | 0.398403 | 0.456685 |
| 12 | point-of-view | 0.485245 | 0.228649 | 0.249458 |

Figura 5.12: Resultados bosques aleatorios trasaplicar el sobremuestreo

los umbrales escogidos para la ingeniería de características han variado respecto a la tabla previa, adaptándose al número de árboles de decisión disponibles en cada bosque.

| Algoritmo | Precisión | Sensibilidad | micro F1 |
|---|---|---|---|
| Bosques aleatorios | 0.742346 | 0.467098 | 0.573401 |
| Sobremuestreo | 0.648764 | 0.575787 | 0.610101 |
| Ingeniería de características | 0.618446 | 0.585489 | 0.601517 |
| Bosques aleatorios finales | 0.630408 | 0.645388 | 0.63781 |

Tabla 5.3: Micro-media de los resultados de los diferentes bosques aleatorios creados

En la tabla 5.3 se ve en orden todos los modelos de bosques aleatorios generados hasta el momento. Los valores, aunque modestos, se ve como mejoran progresivamente obteniendo tras el ajuste de los bosques finales obtenemos un valor de F1 micro de 0,64, más alto que los obtenidos durante la investigación 'Identificando intenciones semánticas en las revisiones de Wikipedia' [24].

### 5.5.4. Clasificación multi-etiqueta

La clasificación multi-etiqueta es aquella en la que la etiqueta puede tomar un conjunto discreto de valores (por ejemplo: $[pajaro, pez, reptil]$) que pueden aparecer combinados. En este caso, un solo modelo puede predecir todas las intenciones de la taxonomía en una revisión al mismo tiempo, siendo las etiquetas, el conjunto de intenciones.

El proceso seguido durante esta clasificación ha sido similar a la clasificación binaria, sin embargo, los resultados obtenidos han sido peores y por tanto este enfoque ha sido descartado.

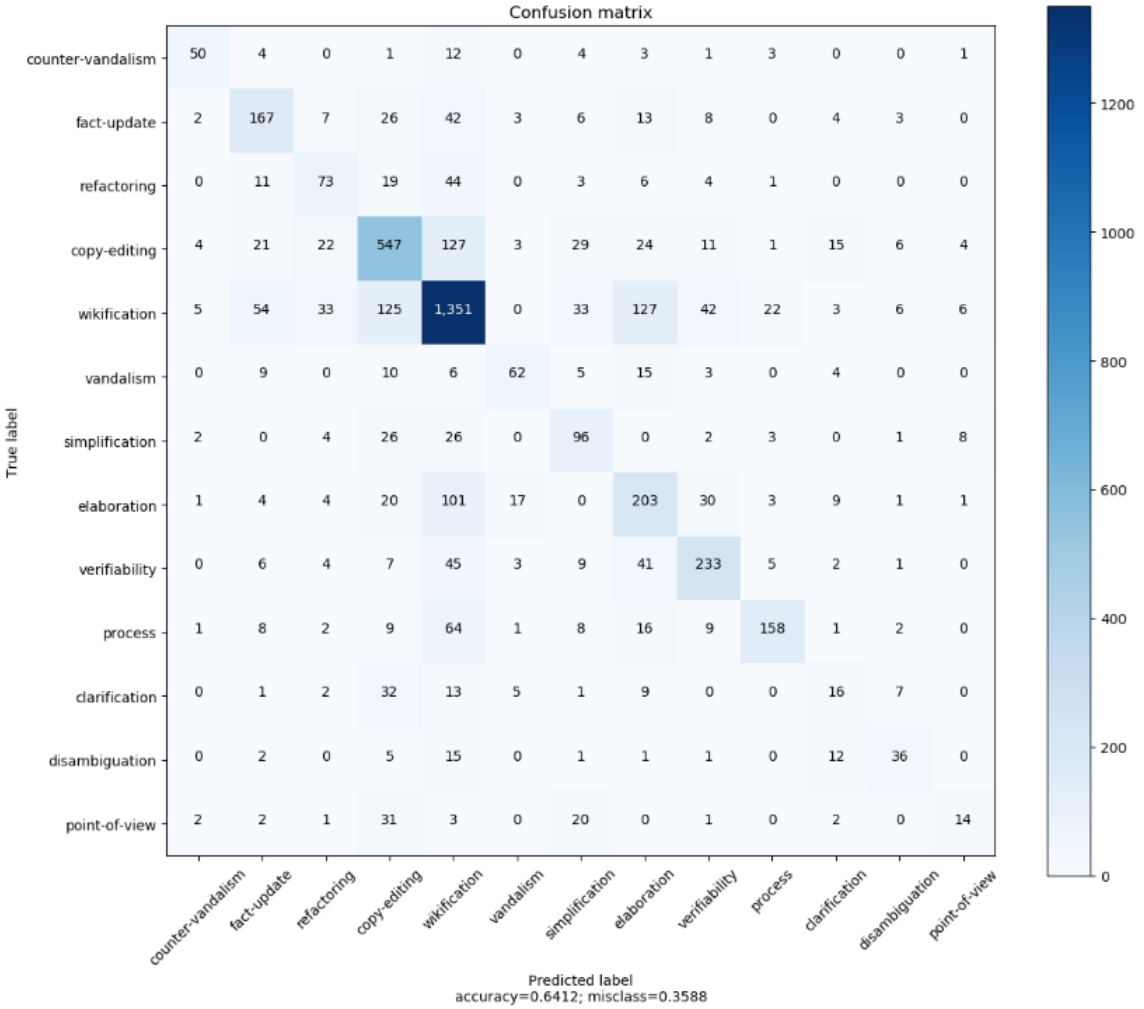



Figura 5.13: Matriz de confusión tras realizar el sobremuestreo en los bosques aleatorios

| | Intention | Precision | Recall | F1 | Feature Engineering Threshold |
|---|---|---|---|---|---|
| 0 | counter-vandalism | 0.780855 | 0.606591 | 0.682489 | mean |
| 1 | fact-update | 0.584234 | 0.476787 | 0.484916 | 0.75*mean |
| 2 | refactoring | 0.501405 | 0.395505 | 0.436443 | 0*mean |
| 3 | copy-editing | 0.623253 | 0.557679 | 0.585363 | 1.25*mean |
| 4 | wikification | 0.768331 | 0.674352 | 0.717534 | 0*mean |
| 5 | vandalism | 0.610224 | 0.560315 | 0.551897 | 0.75*mean |
| 6 | simplification | 0.588868 | 0.515164 | 0.409604 | mean |
| 7 | elaboration | 0.631552 | 0.691032 | 0.658957 | mean |
| 8 | verifiability | 0.746068 | 0.740795 | 0.741321 | 0.75*mean |
| 9 | process | 0.636291 | 0.461902 | 0.534507 | 0*mean |
| 10 | clarification | 0.576469 | 0.418093 | 0.378402 | 1.25*mean |
| 11 | disambiguation | 0.710611 | 0.436289 | 0.473881 | 0.75*mean |
| 12 | point-of-view | 0.491144 | 0.330393 | 0.310440 | 1.25*mean |

Figura 5.14: Resultados con ingeniería de características y sobremuestreo en los bosques aleatorios

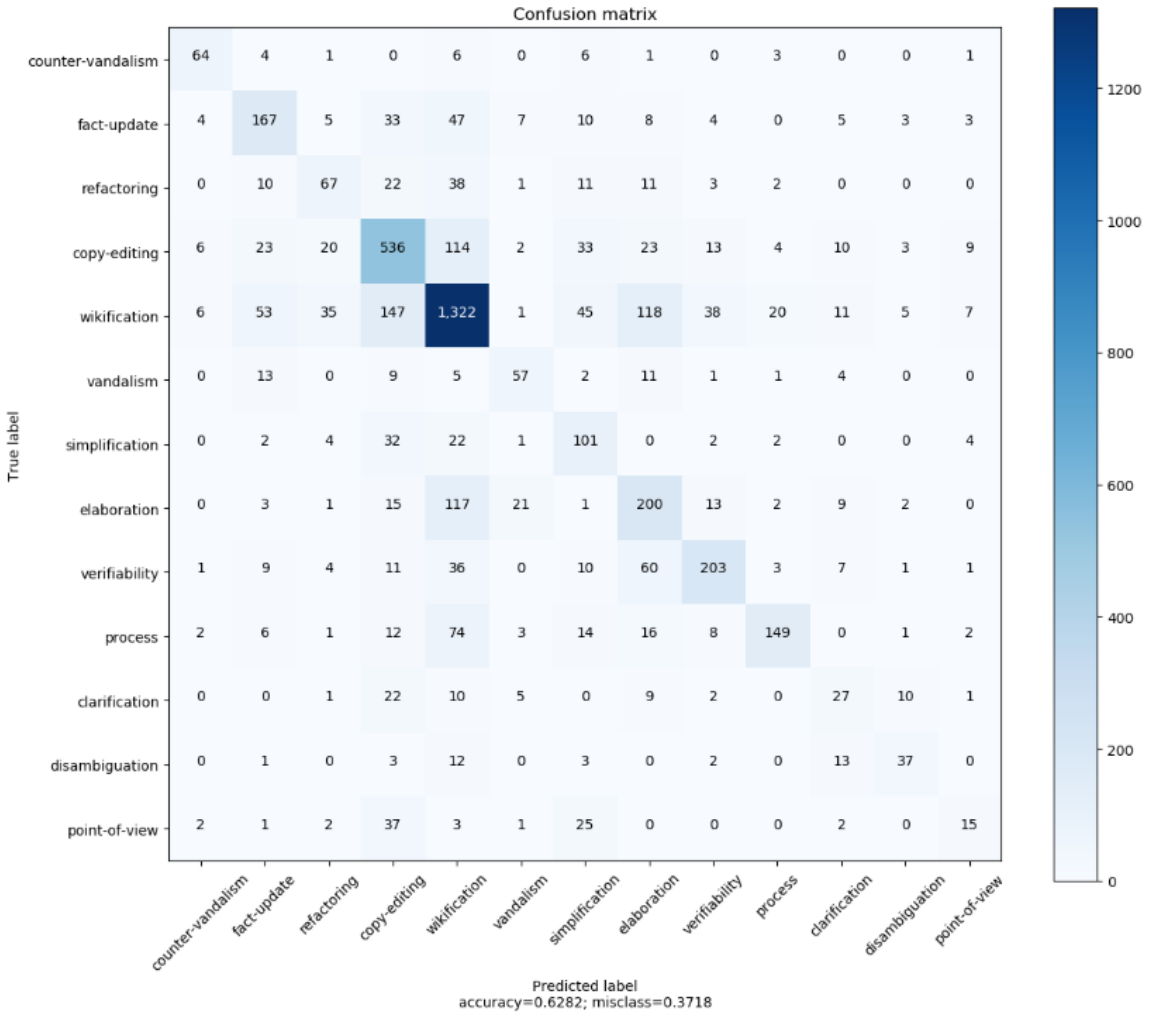



Figura 5.15: Matriz de confusión con ingeniería de características y sobremuestreo de los bosques aleatorios

| | Intention | Precision | Recall | F1 | FE threshold | N_estimators |
|---|---|---|---|---|---|---|
| 0 | counter-vandalism | 0.812393 | 0.675791 | 0.724679 | 1.25*mean | 32 |
| 1 | fact-update | 0.565191 | 0.533272 | 0.539639 | mean | 64 |
| 2 | refactoring | 0.541181 | 0.429990 | 0.469146 | 0*mean | 64 |
| 3 | copy-editing | 0.658105 | 0.625390 | 0.639063 | 0*mean | 385 |
| 4 | wikification | 0.768023 | 0.747786 | 0.756478 | 0.75*mean | 256 |
| 5 | vandalism | 0.674359 | 0.510101 | 0.565078 | 0*mean | 16 |
| 6 | simplification | 0.346733 | 0.566399 | 0.420931 | 0*mean | 1 |
| 7 | elaboration | 0.665594 | 0.742920 | 0.701621 | 0*mean | 192 |
| 8 | verifiability | 0.724446 | 0.789245 | 0.753469 | mean | 128 |
| 9 | process | 0.667541 | 0.489002 | 0.563783 | 0*mean | 64 |
| 10 | clarification | 0.598845 | 0.459437 | 0.424462 | 1.5*mean | 256 |
| 11 | disambiguation | 0.704810 | 0.444097 | 0.487909 | mean | 512 |
| 12 | point-of-view | 0.572345 | 0.368207 | 0.345979 | 1.25*mean | 96 |

Figura 5.16: Resultados de de los bosques aleatorios finales

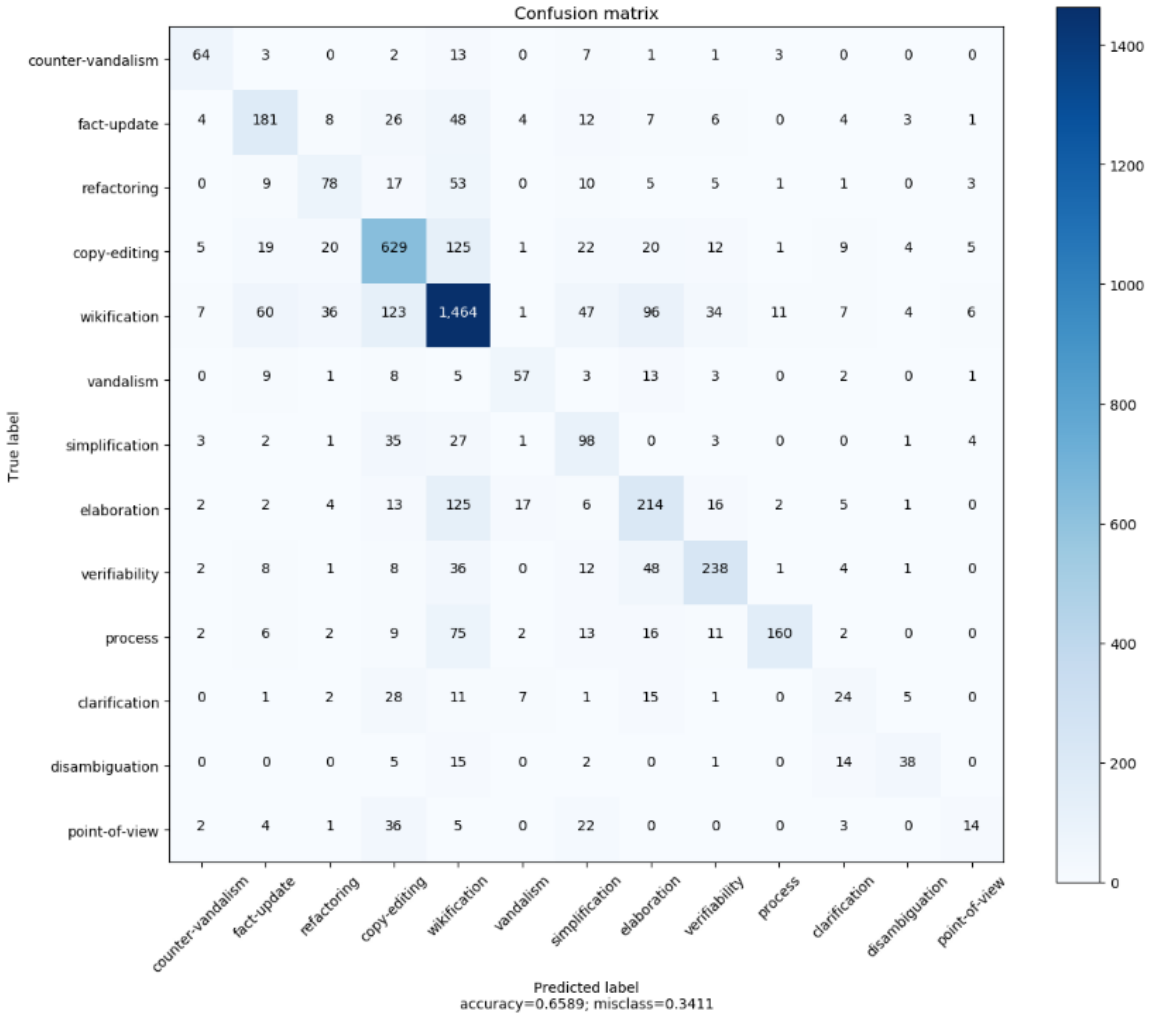



Figura 5.17: Matriz de confusión de los bosques aleatorios finales

| Precisión | Sensibilidad | micro F1 |
|---|---|---|
| 0.760734 | 0.474013 | 0.584657 |

Tabla 5.4: Bosque aleatorio final multi-etiqueta

Los mejores resultados obtenidos durante esta clasificación han sido obtenidos mediante el uso, de nuevo, de un bosque aleatorio con ingeniería de características. En este caso, no se ha utilizado sobremuestreo. Se pueden consultar sus resultados en la tabla 5.4.

Se observa que aunque la precisión es alta, el valor de sensibilidad es muy bajo en comparación, dando lugar a un F1 ligeramente 'engañoso'.

### 5.5.5. Conclusiones

En definitiva, está claro que los clasificadores tienen problemas prediciendo las diferentes intenciones que pueden existir detrás de una revision. A pesar de utilizar dos enfoques diferentes y distintos algoritmos, en ningún caso los resultados muestran un rendimiento alto. Sin embargo, no siempre es posible obtener clasificaciones perfectas y es por esto que el tipo de datos que se esté prediciendo es muy importante.

El rendimiento de estos clasificadores aún puede ser incrementado. Con conocimiento del dominio suficiente, se podría ajustar el conjunto de datos existente para reducir ambiguedades entre las diferentes intenciones o eliminar por completo atributos que no sean necesarios para determinadas intenciones. Sin embargo, eso se escapa al objetivo de este análisis.

| Algoritmo | Precision | sensibilidad | F1 micro |
|---|---|---|---|
| Bosques aleatorios binarios | 0.630408 | 0.645388 | 0.63781 |
| Bosque aleatorio multi-etiqueta | 0.760734 | 0.474013 | 0.584657 |

Tabla 5.5: Micro media de los resultados de los modelos finales de cada enfoque

El conjunto de bosques aleatorios de la clasificación binaria ha tenido un rendimiento mayor que la clasificación multi-etiqueta, aunque en ambos casos, el tipo de algoritmo con mejores resultados haya sido el mismo. Mientras que los bosques aleatorios binarios han obtenido un valor de micro F1 de 0,63781 y una precision y un sensibilidad de 0,630408 y 0,645388 respectivamente, el bosque aleatorio multi-etiqueta tiene un valor de micro F1 de 0,584657 y valores de precision y sensibilidad de 0,760734 y 0,474013 respectivamente.

Por tanto, el enfoque seleccionado para hacer futuras clasificaciones de revisiones será el conjunto de bosques aleatorios utilizados en la clasificación binaria.

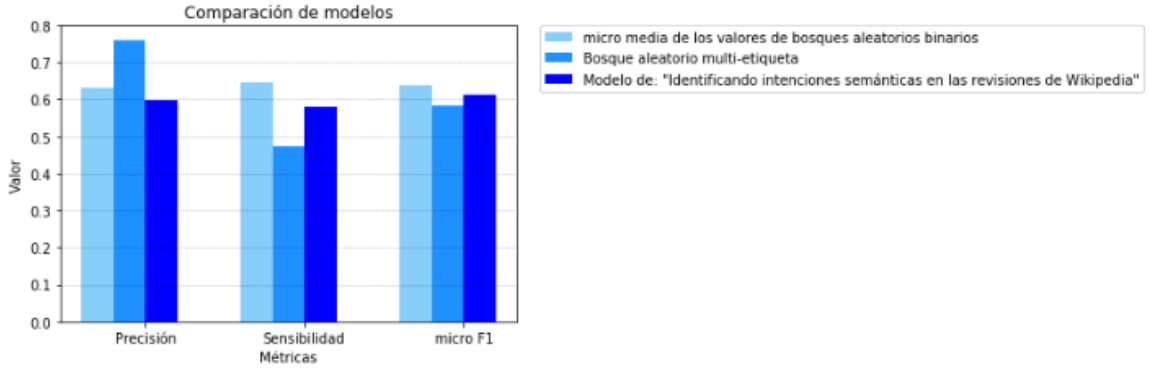



Figura 5.18: Gráfica con resultados de modelos finales

En definitiva y como se puede ver en la figura 5.18, los bosques aleatorios de la clasificación binaria superan a los demás en sensibilidad y micro F1, solo estando ligeramente por debajo en precisión frente al bosque aleatorio multi-etiqueta. Por otro lado, se ha conseguido obtener mejores resultados de precisión, sensibilidad y micro F1 que aquellos obtenidos en

'Identificando intenciones semánticas en las revisiones de Wikipedia' ([24]), con valores de $0,613$ en micro F1, 0,578 en sensibilidad y 0,599 en precisión en su modelo que hace uso del algoritmo $k$-vecinos más cercanos en su versión multi-etiqueta.

# Capítulo 6

# Análisis con minería de procesos

Tal y como se explica en el capítulo 3, la minería de procesos se compone de un conjunto de técnicas de minería que permiten extraer información de logs de eventos para descubrir, monitorizar y mejorar procesos [4]. Esta información puede ser analizada para tomar forma de decisiones de negocio o estratégicas para optimizar los procesos o simplemente para conocer el propio funcionamiento de los mismos. Para ello, se hace uso de logs de eventos. Un log de eventos se compone de información proveniente de bases de datos, transacciones... y pueden considerarse una colección de secuencias de eventos ya que se asume que es posible guardar secuencialmente eventos de modo que, cada evento, denote una actividad (evento) y esté asociado a un caso particular (traza).

Tal y como se ha comentado anteriormente incrementar la eficiencia de un proceso no es lo que hace interesante a la minería de procesos de cara al estudio objeto de este proyecto, si no el propio descubrimiento de los procesos inherentes a la edición y evolución de los *artículos destacados* en comunidades de conocimiento colaborativo abiertas como la Wikipedia española.

De esta manera, el objetivo de usar la minería de procesos es descubrir y analizar el flujo de trabajo que siguen los propios articulos desde su creación hasta la fecha actual en caso de que este exista, así como los posibles procesos existentes dentro del comportamiento de los propios editores.

La herramienta escogida para aplicar las técnicas de minería de procesos es ProM tools 8.6 tal y como ha sido comentado previamente durante el capítulo 4 y 3.

El proceso a seguir será el siguiente:

1. Obtención de datos: En este apartado se explicará el proceso a seguir para obtener el log de eventos de cara a aplicar las técnicas de minería de procesos. Se determinará de manera clara cuales son los log de eventos que serán utilizados durante las demás secciones de este capítulo.

2. Transformación a XES: Se centra en la conversión del log de eventos al formato estándar de la minería de procesos: XES. Determinando de manera clara su estructura y utilidad

en las futuras secciones.

3. Análisis exploratorio de los datos: Antes de aplicar las técnicas de minería se estudiará el log de eventos mediante la realización de diferentes gráficas para poder analizar visualmente el conjunto de datos de partida y así obtener información acerca de la organización y estructura del log.

4. Análisis a nivel artículo: Se aplicará la minería de procesos de cara a descubrir los procesos existentes dentro de la evolución de los artículos de Wikipedia.

5. Análisis a nivel editor: Esencialmente se hará lo mismo que en el análisis a nivel de artículo pero cambiando el foco de estudio. En lugar de descubrir los procesos inherentes a la evolución de los artículos estudiaremos aquellos seguidos por los propios editores en sus sesiones de trabajo.

## 6.1. Obtención de datos

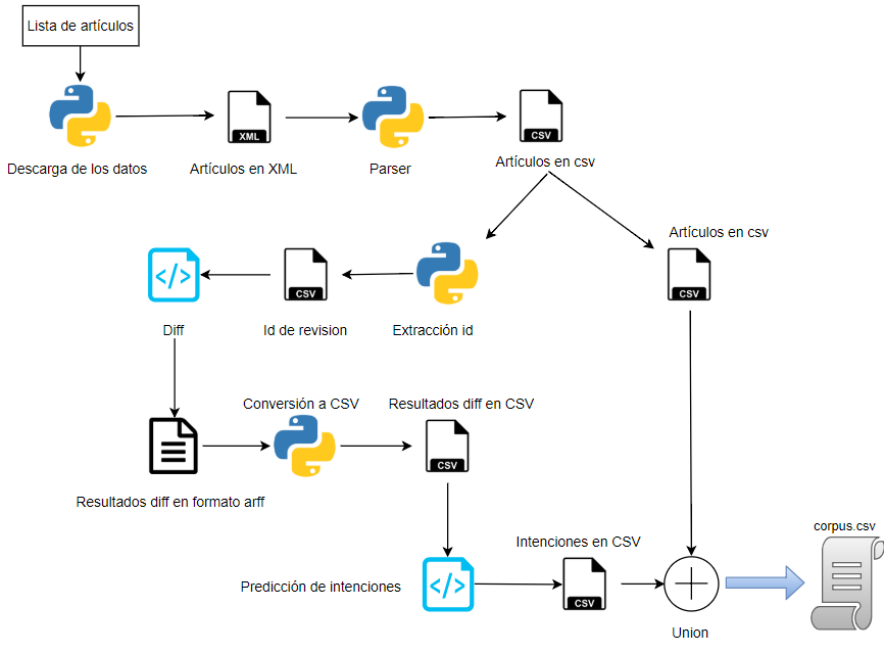



Figura 6.1: Proceso seguido para la descarga y preparación de los datos para realizar minería de procesos

Durante el capítulo 5, se ha explicado el proceso seguido desde la descarga del historial de revisiones de un artículo de Wikipedia hasta la predicción de las intenciones semánticas tras cada revisión. En la figura 6.1 podemos observar un esquema de los pasos a seguir.

1. El proceso comienza con la descarga de un historial de revisiones de un artículo en Wikipedia, realizado mediante el script de descarga[1]. En este caso, se han seleccionado los siguientes *artículos destacados* de diferentes categorías aleatoriamente de entre todos los disponibles en Wikipedia:

   - Leche
   - Odín
   - Homer Simpson
   - Bifaz
   - Angkor Wat
   - Airbus A380
   - Ácido desoxirribonucleico
   - Tierra

   El motivo de la selección de artículos aleatorios de diferentes categorías es tratar de descubrir procesos a nivel general dentro de la escritura colaborativa. Variedad de temas da lugar a variedad de editores y por tanto de estilos.

2. Una vez contamos con los XML de cada historial de revisiones de artículos descargado hacemos uso del parser proporcionado por Abel Serrano Juste[2] que los transforma en archivos CSV dotados de id y título de artículo y revisión, timestamp e id y nombre del editor además del conjunto de bytes afectados en la revisión.

3. Con estos archivos CSV que almacenan la información de los artículos seleccionados extraemos el id de cada revisión[3] y ejecutamos el conjunto de scripts de la investigación base 'Identificando intenciones semánticas de las revisiones de Wikipedia'[4]. Estos scripts realizan una comparación de cada revisión con la anterior en un artículo, generando un archivo en formato arff por artículo compuesto de 207 atributos y el id de revisión. Estos atributos hacen referencia a las diferencias encontradas en cada comparación y son utilizados para la posterior predicción de las intenciones tras cada revisión.

4. Para predecir las intenciones en base a los archivos arff generados por la comparación, primero, debemos cambiar su formato a CSV de nuevo lo cual realizamos mediante el uso de un script de conversión[5].

5. Llegados a este punto, se hace uso del bosque aleatorio ajustado seleccionado en la sección 4.5 del capítulo 5 para predecir las intenciones tras cada revisión del conjunto de

[1] wiki_dump_downloader.py (11)
[2] wiki_dump_parser.py (11)
[3] Haciendo uso del script revision_id_extractor.py (11)
[4] Más en 11 y `https://github.com/diyiy/Wiki_Semantic_Intention`
[5] arffToCsv.py (11)

archivos csv dotados de los 207 atributos obtenidos en la comparación[6]. Estas intenciones, al mismo tiempo que son predecidas, son añadidas a los archivos CSV resultado de hacer uso del parser de Abel durante el punto 2 de esta lista.

Se cuenta por lo tanto con ocho archivos csv, uno por cada artículo seleccionado que contienen id y título de artículo y revisión, timestamp e id y nombre del editor además del conjunto de bytes afectados en la revisión y la intencionalidad tras la misma. El último paso, es unir todos los archivos en uno solo denominado **corpus.csv** tal y como se ve en el paso final del esquema 6.1 y será el fichero base de todo este capitulo.

Para realizar el análisis a nivel artículo se hará uso de este mismo **corpus.csv** en su totalidad pues el objetivo es descubrir los procesos ocultos tras la elaboración de artículos en Wikipedia por lo que un filtrado podría alterar los resultados.

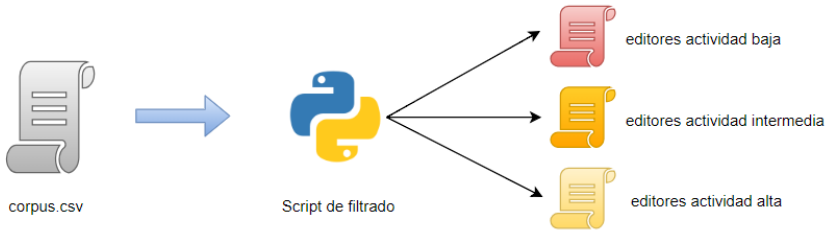



Figura 6.2: Esquema del filtrado del corpus para el análisis a nivel editor

En cambio, en la sección análisis a nivel de autor se van a generar tres ficheros diferentes filtrando por cantidad de revisiones por revisor. Como observamos en el esquema 6.2 mediante un script de filtrado[7] generaremos tres archivos siguiendo la siguiente pauta:

- Editores de actividad baja: engloba solo revisiones realizadas por autores con menos de cinco revisiones.
- Editores de actividad intermedia: contiene exclusivamente revisiones hechas por autores que han realizado entre cinco y cincuenta revisiones.
- Editores de actividad alta: Compuesto solo por aquellos con más de cincuenta revisiones.

De este modo, a lo largo de las sucesivas seciones estos archivos serán referidos de la misma manera: fichero de revisiones de editores con actividad baja/intermedia/alta.

El motivo de este filtrado es el siguiente: durante la sección análisis a nivel de editor el foco está puesto en el editor como individuo, por lo que para analizar un grupo específico de editores es necesario poder filtrar aquel comportamiento que no nos interesa, pues solo añade ruido al análisis. Así, se puede determinar si en función del número de ediciones existen comportamientos diferentes.

[6] generate_predictions.py (11)
[7] corpus_filter.py (11)

Las carácterísticas del corpus y sus ficheros derivados dan lugar a que puedan ser considerados log de eventos pues reunen todas las carácterísticas tal y como se detalla en el capítulo 3. Sin embargo, aún se encuentran en formato CSV por lo que para ser utilizados en la minería de procesos aún deben ser convertidos al formato estándar: XES.

## 6.2. Transformación a XES

De cara a realizar minería de procesos con ProM es necesario el uso del formáto estándar del IEEE para los logs de eventos [23]. La conversión CSV a XES se realiza mediante la herramienta de conversión proporcionada por el propio ProM.

El formato XES se encontraba compuesto por eventos que refieren una actividad asociados a un caso particular (una traza). Además cada evento puede contar con atributos como timestamp indicando el tiempo en el que sucedio dicho evento o org:resource que representa el actor que ejecutó tal evento. En este caso, se ha realizado la conversión al formato XES de todos los ficheros obtenidos en la sección anterior siguiendo el esquema de la figura 6.3.

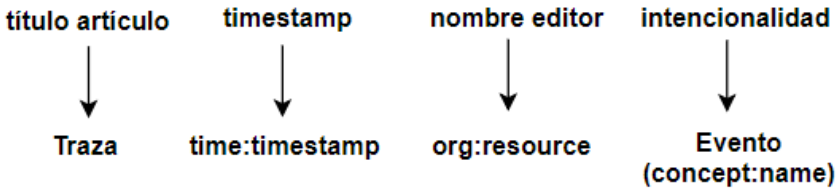



Figura 6.3: Representación de los atributos del formato CSV tras la conversión XES

De esta manera nos encontramos con que hemos determinado cada traza como cada artículo y cada evento como cada intención, de modo que cada revisión está determinada por esta. Además, hemos seleccionado el nombre del editor como actor ejecutor de cada evento, es decir, org:resource. El timestamp se ha utilizado para determinar la hora a la que se realizó la edición.

Así, contamos ya con el log de eventos **corpus** así como los log de eventos generados a partir del mismo en función de las revisiones de los editores en formato XES.

## 6.3. Análisis exploratorio de los datos

Antes de comenzar a aplicar la minería de procesos, se va a realizar un análisis exploratorio del log de eventos **corpus**, en formato XES, para obtener una mayor información del mismo con la esperanza de que esto pueda traer beneficios a la hora de realizar un análisis de los resultados obtenidos mediante la minería.

En la imagen 6.4 observamos la representación del archivo XES del corpus en ProM. El log de eventos corpus había sido divido en casos (trazas) en función de cada artículo y se

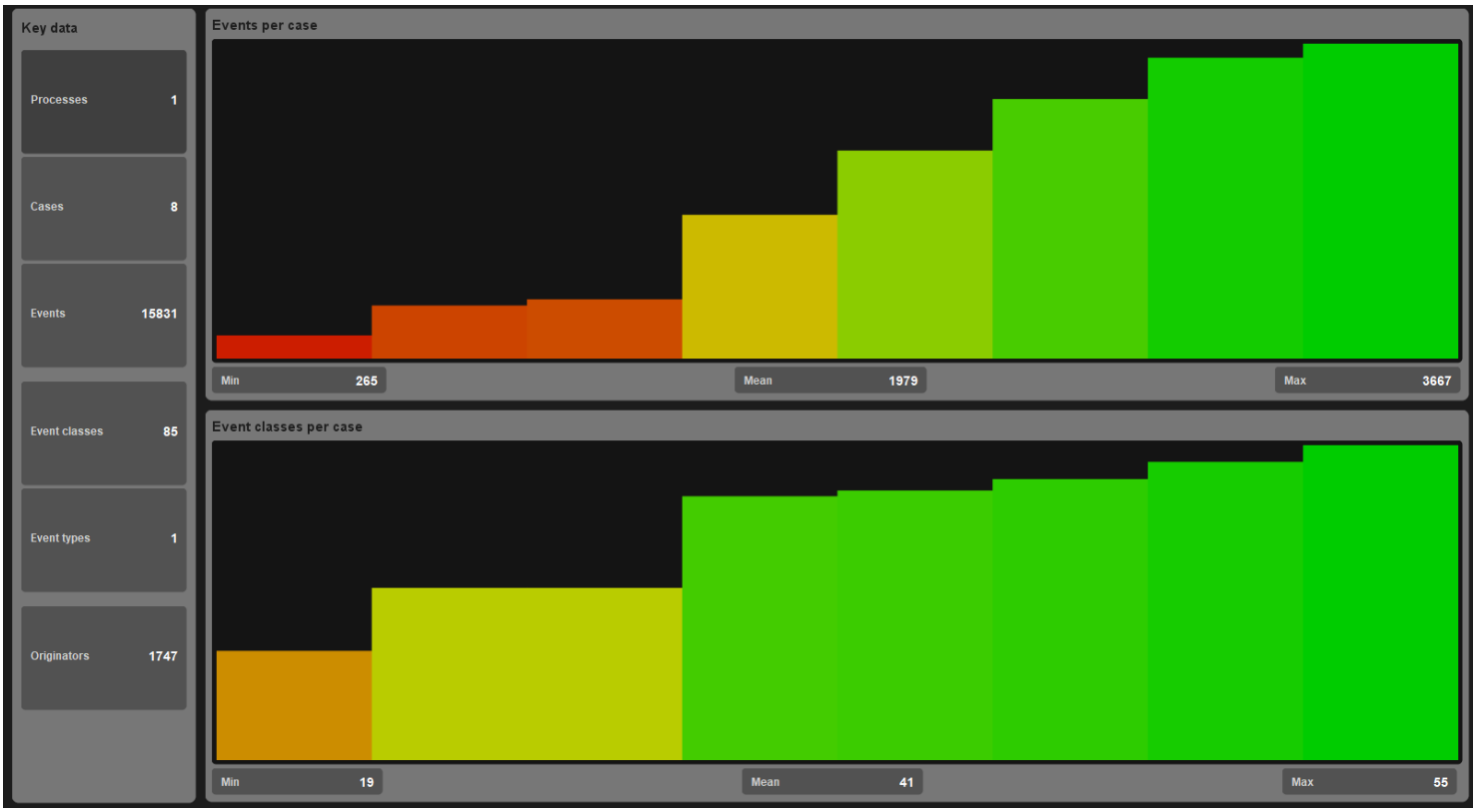



Figura 6.4: Corpus en formato XES

había constituido cada revisión como un evento determinado por cada intención. La figura puede descomponerse en tres secciones:

- Margen izquierdo: contiene datos clave del log de eventos:
  1. Processes: Indica el número de procesos existentes, en este caso solo 1 pues solo hacemos uso de un corpus.
  2. Cases: Número de casos, dado que hemos separado cada caso por cada artículo, tenemos 8 casos.
  3. Events: Número de eventos, en este caso, revisiones, determinadas por su intención tal y como hemos determinado al transformar a formato XES. Contamos con 15831 revisiones.
  4. Event classes: Clases de eventos diferentes. Al venir determinados por su intención, esto indica el tipo de intencionalidades diferentes observadas. En este caso se observan 85 tipos de intencionalidad diferentes. Teniendo en cuenta que la taxonomía de intenciones cuenta con 13 intenciones que pueden combinarse entre si mismas, el valor máximo de esta casilla sería:

$$\sum_{n=1}^{13} \binom{13}{n}$$

  5. Event types: Hace referencia a los tipos de evento. En este caso sólo hemos determinado cada evento (revisión) por su intencionalidad por lo que el valor de esta casilla

es 1. Si se determinasen, por ejemplo, mediante intencionalidad y org:resource su valor sería de 2.

6. Originators: Representa en este caso la cantidad de actores diferentes, es decir, la cantidad de editores. Contamos con 1747 editores diferentes repartidos a lo largo de 8 artículos.

- Gráfica superior: Aquí observamos un gráfico de barras que representa el número de eventos por caso. Esto es, el número de revisiones por artículo. Claramente, hay mucha disparidad entre los 8 artículos, con el más pequeño de todo compuesto solo de 265 revisiones y el mayor con 3667. No obstante, todos los artículos pertenecen a la categoría de *artículo destacado* lo que podría denotar que el número de revisiones no es un indicador fiable de la posible calidad de un artículo.
- Gráfica inferior: De nuevo se trata de un gráfico de barras que representa la cantidad de clases de eventos en cada artículo. Así, cada barra de las 8 existentes muestra el número de intenciones diferentes tras el conjunto de revisiones en su respectivo artículo. Aquí, también se encuentran disparidades. El artículo con menor variedad de intencionalidades registrando sólo 19 diferentes en contraposición con el artículo con mayor variedad: 55.

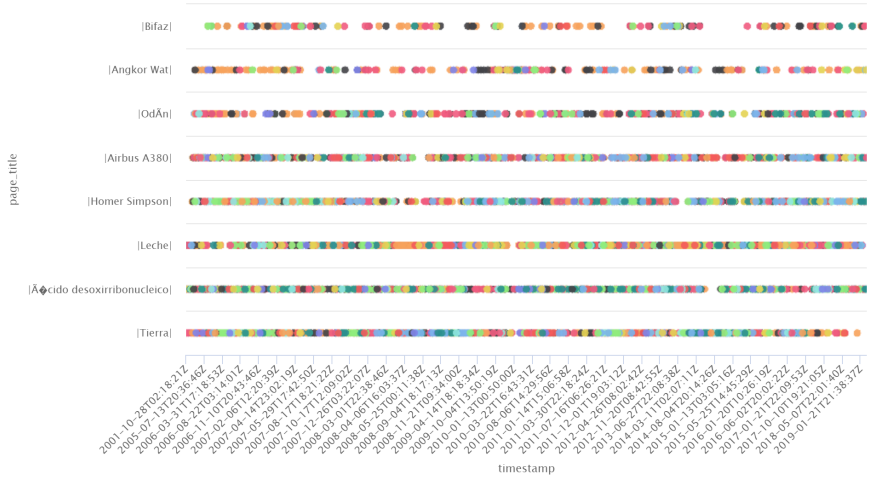



Figura 6.5: Revisiones a lo largo del tiempo por artículo

En la gráfica 6.5 podemos ver las revisiones de los artículos a lo largo de su existencia. Todos los artículos fueron creados antes de 2006, sin embargo, los más antiguos datan de 2001. Se puede observar como todos los artículos han sido editados sin descanso hasta el día de hoy pues se observan muy pocas interrupciones en general. El artículo Bifaz además de ser el último en crearse, es el que más 'parones' ha tenido en su historial de revisiones. Estos dos factores, son los que hacen que sea el artículo con menos revisiones del corpus: 265. A pesar de ello y al igual que los demás, se trata de un *artículo destacado*, por lo que podríamos extraer que la calidad del artículo no es directamente proporcional a la cantidad de revisiones que tiene, aunque para confirmarlo habría que hacer uso de un corpus de mayor extensión. Además, se puede ver en el eje temporal de la gráfica que hay un salto temporal de 2001 a 2005. Esto, es debido a que las fechas son mostradas proporcionalmente en función del número de revisiones realizado entre cada período de tiempo representado en el eje. Teniendo en cuenta que

la Wikipedia se fundó en 2001 no resulta sorprendente que entre 2001 y 2005 haya la misma cantidad de revisiones que en periodos de tiempo mucho más reducidos a partir de 2005 pues se observa en los datos que solo hay 247 revisiones antes de ese año a lo largo de todo el corpus.

Por otro lado, en la figura 6.6 se puede ver la evolución de las intenciones del conjunto del corpus en el tiempo. Intenciones como wikification, elaboration, copy-editing y vandalism evolucionan rápidamente de modo lineal frente a todas las demás intenciones con menos frecuencia de aparición.Además, en los últimos años se observa como la frencuencia de aparición de fact-update cambia y empieza a crecer a mayor velocidad. De este modo lás intenciones con mayor frecuencia como elaboration o copy-editing o wikification son asociadas con alta calidad de los artículos [24] lo cual concuerda con el estatus de *artículo destacado* de los artículos que conforman el corpus.

Otras intenciones como re-factoring y point of view se observa que empiezan a aparecer pasado un tiempo, indicando que se trata de una intención propia artículos con cierto recorrido.

El vandalismo en Wikipedia, suele caracterizarse por su persistencia en el tiempo [16]. De este modo, también en nuestro caso el vandalismo es persistente en el tiempo, indicando quizá que la madurez de un artículo no tiene relación con el nivel de vandalismo que recibe.

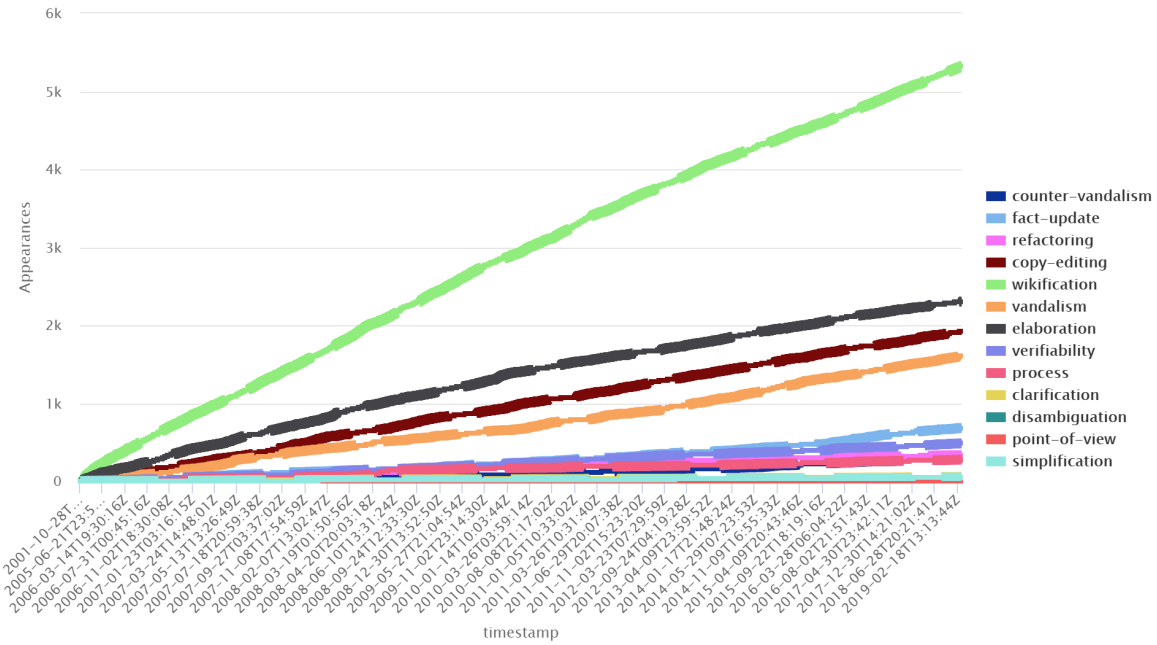



Figura 6.6: Intenciones a lo largo del tiempo en el corpus

En cuanto a los eventos (revisiones), hemos visto que hay 85 clases diferentes, para ver su representación en el corpus, se ha creado una gráfica tipo WordCloud que puede observarse en la figura 6.7.

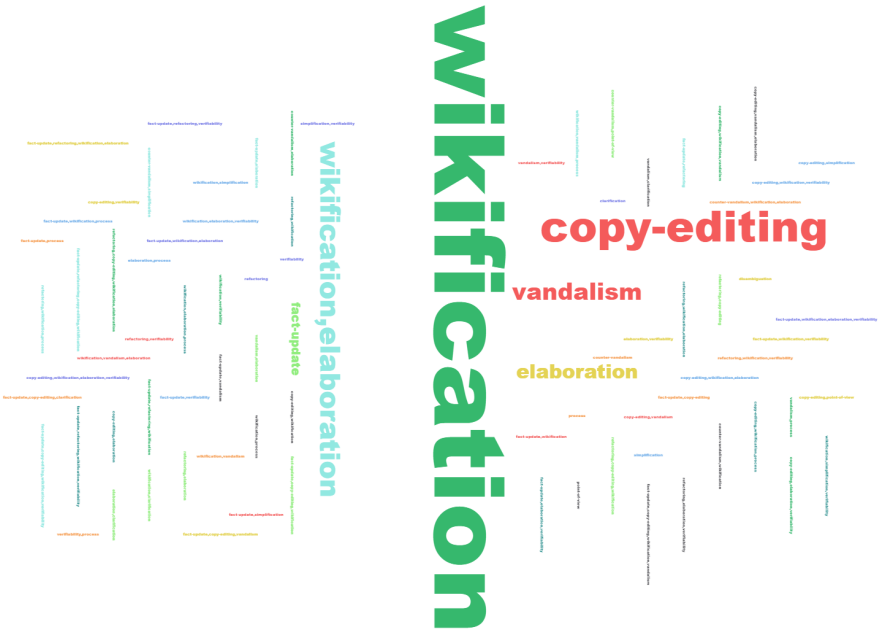



Figura 6.7: Representación de las distintas intenciones

Entre las diferentes intenciones posibles en una revisión, la que tiene mayor frecuencia de aparación es sin duda wikification, seguido de copy-editing, wikification+elaboration, elaboration y vandalism. La frecuencia de aparición de wikification es considerablemente más alta que la de cualquier otra intención: 3569 apariciones en 15831 revisiones sin tener en cuenta aquellas revisiones en los que aparece combinada con otras, además, aparece constantemente desde el inicio de un artículo y durante toda su evolución. Esto, podría indicar un problema en sí mismo con las herramientas de Wikipedia a la hora de escribir y dar formato al artículo. No parece especialmente óptimo que constantemente se tenga que estar editando el formato de los artículos a pesar de que según avanza la calidad de un artículo, aumenta la importancia de determinadas intenciones, como wikification y con ello su frecuencia [24].

Por último, están los editores. A lo largo de todo el corpus, se encuentran 1747 editores diferentes. Sin embargo, solo 869 han realizado más de 1 edición y 380 más de 5, así, pocos autores realizan grandes números de ediciones, es decir, la mayoría de editores son 'casuales' o de baja actividad. En gran parte las revisiones han sido realizadas por usuarios anónimos, representando aproximadamente un tercio del total de revisiones. Los usuarios individuales representan una fracción pequeña del total de revisiones a nivel individual, sin embargo, existen grandes diferencias entre ellos. Es decir, la mayoría de editores son usuarios que han realizado un número reducido de revisiones.

En resumen, nos encontramos con un log de eventos compuesto de 8 artículos y 15831 revisiones formado por 85 combinaciones diferentes de intenciones, siendo los artículos diferentes entre ellos con la calidad y la antiguedad como nexo de unión. Por otro lado, se cuenta con 1747 editores diferentes dando lugar a una gran variedad de datos para la aplicación de minería social a nivel de editor. Ahora que se conoce el **corpus** y los datos que lo componen,

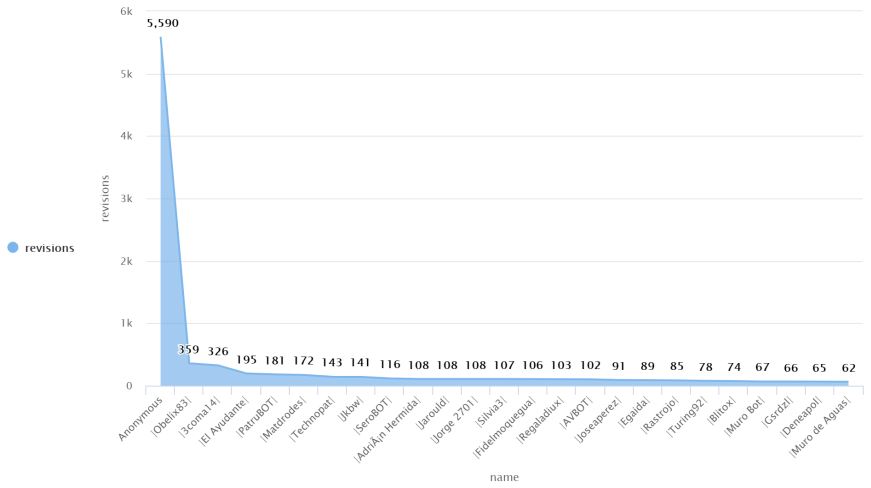



Figura 6.8: Número de revisiones por editor

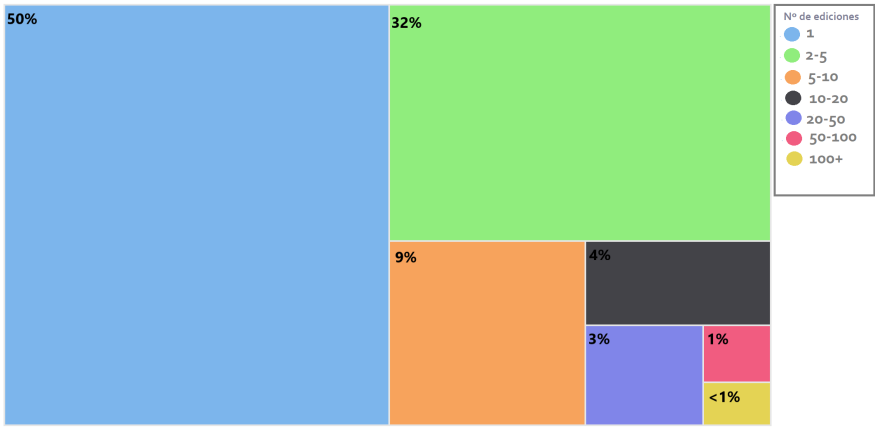



Figura 6.9: Número de editores según cantidad de ediciones realizadas

se procede a aplicar la minería de procesos. Para esto, se utilizarán los enfoques previamente introducidos: a nivel artículo y a nivel editor.

## 6.4. Análisis a nivel artículo

El objetivo de esta sección es realizar un análisis mediante la utilización de técnicas de minería de procesos para intentar estudiar los procesos inherentes a la propia evolución de los artículos en la Wikipedia española. En este caso el objetivo estará puesto sobre las intenciones tras cada revisión y su fecha de realización, sin importar el usuario que la realizó.

Así, partimos del mismo archivo XES (**corpus**) que ha sido analizado en la sección previa.

El primer paso de todos es escoger el algoritmo minero adecuado para nuestro tipo de log

de eventos, que en este caso será el minero heurístico. De entre las opciones disponibles en ProM, se ha hecho uso de el minero heurístico ya que tal como se describe en la sección 5.2 de este documento es un algoritmo práctico que puede utilizarse para explicar el comportamiento principal registrado en un log de eventos además de lidiar muy bien con el posible ruido de los datos [22]. Además, ha sido utilizado con éxito en ámbitos como la escritura colaborativa [19] y en la educación [6], similares en esencia al nuestro, especialmente la escritura colaborativa ya que es precisamente en lo que se basan las comunidades de conocimiento colaborativas como las wikis.

Aplicando el minero heurístico con los parámetros por defecto al **corpus**, obtenemos una red heurística que es transformada en red de petri automáticamente por ProM, con un fitness de 0.7996. Este valor, representa en una escala de 0 a 1 la porción de los eventos observados en el log de eventos que puede ser reproducida en la red. De este modo, se trata de un valor razonablemente alto para el contexto en el que nos encontramos. Sin embargo, es una red de petri muy compleja como para analizar a ojo (6.10).

Para poder simplificar esta red de petri de modo que se pueda extraer información de ella de modo visual, se aplica una técnica de aprendizaje no supervisado: clustering. El clustering consiste en agrupar un conjunto de objetos de modo que cada grupo esté compuesto por objetos similares. En este caso, el objetivo de realizar clustering en el **corpus** es el de agrupar las revisiones por similitud en base a los procesos que se observen. Esto es realizado de modo trivial mediante la herramienta incluida en ProM 'Discover clusters'.

Una vez que se ha aplicado 'Discover clusters' al **corpus**, se utiliza tanto el conjunto de clusters obtenidos como el propio **corpus** y se aplica 'Discover using Decomposition' que hace uso del minero heurístico teniendo en cuenta la agrupación realizada generando una versión descompuesta en diferentes partes de la red de petri previa (6.10).

De esta manera, hemos logrado simplificar la red de petri anterior en varias redes diferentes de menor tamaño y mayor legibilidad. Así, contamos con (i) la red de petri general (6.10), (ii) la red de petri descompuesta 1 (6.11), (iii) la red de petri descompuesta 2 (6.12) y (iv) el conjunto de mini redes de petri descompuestas 3 (6.13)

Claramente puede verse que aún siendo más simples, siguen siendo complejas de analizar para el ojo humano, tanto la red 1 (ii) como la red 2 (iii). Esto, no es sorprendente puesto que el proceso que siguen los artículos durante su evolución es esencialmente anárquico en cuanto a la frecuencia o la cantidad de usuarios que colaboran, dando lugar a procesos potencialmente diferentes entre artículos. No obstante, a pesar de ser un corpus de ocho artículos diferentes compuesto de muchos editores diferentes y con un recorrido muy largo en el tiempo, la red es razonablemente comprensible. Se observa que los procesos están llenos de bucles, por lo tanto son naturalmente y de modo inevitable iterativos, no hay una sola cadena de ediciones a seguir si no muchas posibilidades diferentes. Esencialmente podría traducirse como que no existe un proceso unificado y real que se siga en Wikipedia, lo cual es esperado pues surge del conjunto de trabajo de muchos usuarios potencialmente sin colaboración explícita entre ellos. Sin embargo, observamos diferencias entre las redes simplificadas.

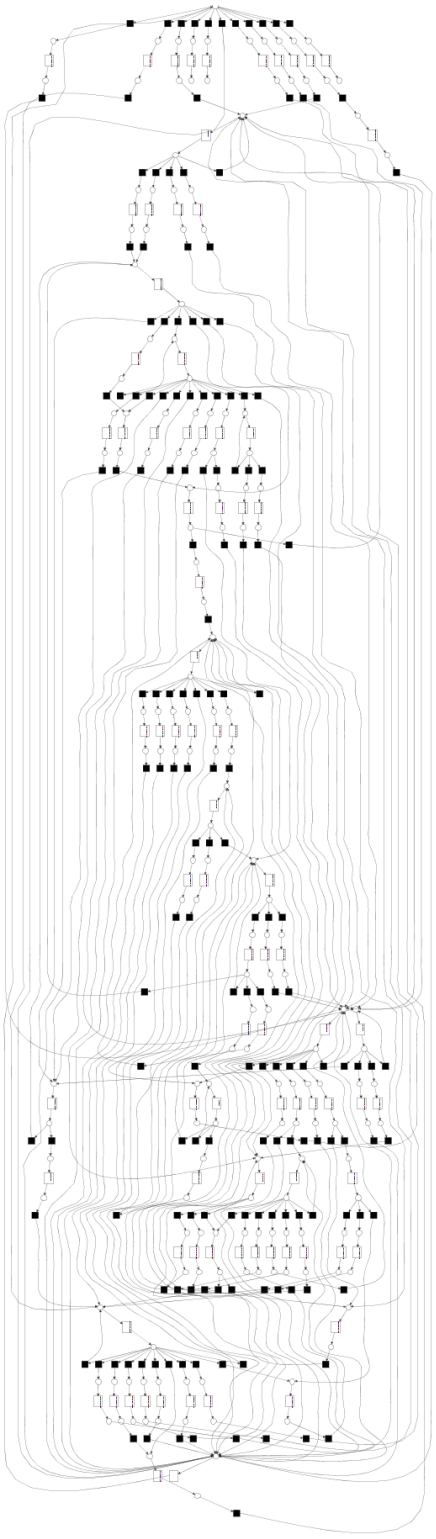


Figura 6.10: Petri net obtenida con Minero Heurístico

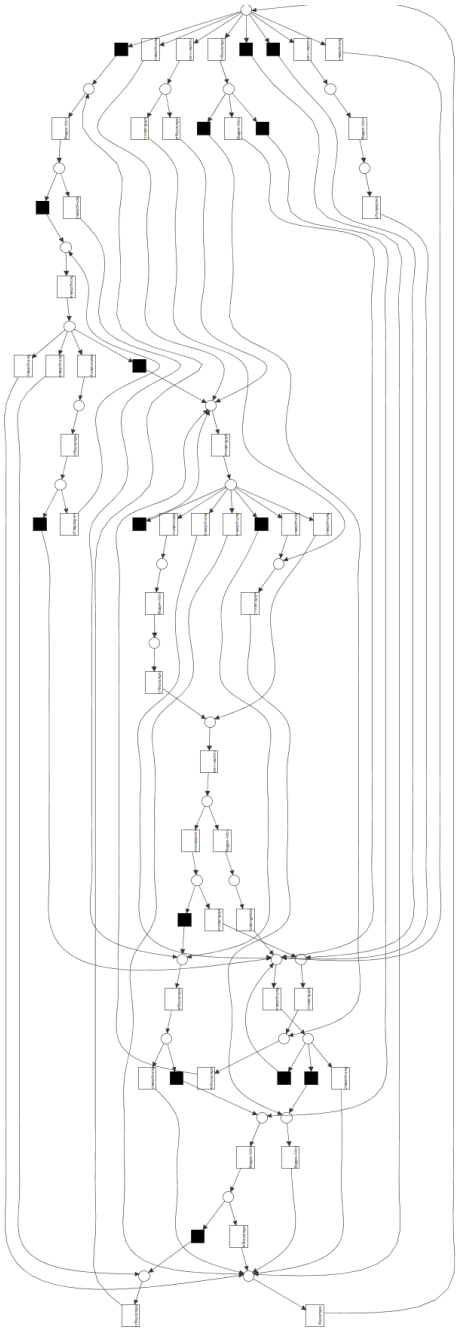


Figura 6.11: Petri net del proceso de edición tras su descomposición 1

### 6.4.1. Análisis de la red de Petri descompuesta 1

Como vemos en la red de petri de la imagen 6.11 a grandes rasgos se observa un proceso con mucha interconexión y de gran complejidad. Para facilitar su análisis se ha dividido la red

en tres fragmentos para poder observarlos con más en detalle. Esto, está motivado en parte por que el inicio tiene interes especial pues ver si hay un inicio específico dentro del proceso podría ayudar a identificarlo al mismo tiempo que el final aporta utilidad para descubrir si existen puntos finales en el proceso o es abierto y por tanto potencialmente 'infinitamente' iterativo.

El inicio de la red de petri 1 se puede ver en la figura 6.14. A simple vista resalta la cantidad de intenciones complejas que se ven en el proceso, siendo en todos casos intenciones combinadas. A priori se observa como estos posibles inicios del proceso van marcados por intenciones como refactoring+copy-editing, fact-update+refactoring+verifiability, fact-update+refactoring+wikification+elaboration... Estas intenciones observadas tendrían sentido en etapas tempranas de un artículo donde aún está todo por hacer, lo cual podría explicar que se encadenen muchas revisiones con intenciones combinadas complejas. Durante la investigación 'Estabilidad turbulenta de los roles emergentes' determinan que durante las etapas tempranas de desarrollo de un artículo, el 60 % de sus editores toman el rol de 'All round contributor' [5] lo que significa que son usuarios todo terreno que hacen un poco de todo. Esto se ve reforzado por los resultados observados en esta sección de la red de petri donde las intenciones a menudo contienen elementos de todo tipo como refactoring+copy-editing+fact-update+wikification.

En la figura 6.15 vemos la siguiente parte de la red de petri descompuesta 1. Más concretamente su sección intermedia. Las intenciones se van simplificando poco a poco, empezando a ser las combinaciones de 3 o más intenciones menos frecuentes. Esto es esperado pues según aumenta en antiguedad el artículo se reduce el porcentaje de revisores que representan 'All round-contributor' en pos de roles más específicos como copy-editors o layour-shapers o 'quick and dirty editors' [5]. Esto puede observarse en intenciones como refactoring+elaboration o simplification+verifiability o incluso vandalism+verifiability que puede estar relacionado con aquellos bajo el rol 'quick and dirty editors' ya que a veces sus rápidas ediciones son confundidas por vandalismo [5]. Sin embargo aún se siguen dando secuencias complejas pues se observan conexiones que vuelven al inicio de la red indicando que el archivo puede estar pasando por fases intensas de re-escritura.

La figura 6.16 muestra el último tramo de la red de petri descompuesta 1. No aporta mucha información adicional respecto a la sección intermedia pues se observa un comportamiento similar en cuanto a las intenciones que se ven. Sin embargo vemos que no existe un punto final en el proceso. Tras llegar al final la naturaleza iterativa de proceso de re-edición continuo de Wikipedia hace que exista la posibilidad de volver a otros puntos del proceso. Es decir, es un proceso abierto.

En definitiva, aunque no podemos extraer demasiada información mas allá de observar ciertas similitudes con otros estudios ya realizados previamente al respecto de los roles de usuario y cómo se edita a lo largo de la vida útil de un artículo en Wikipedia, se observa que en diferentes etapas del artículo las intenciones realizadas varían ligeramente, siendo ediciones muy complejas en el inicio (en el sentido de múltiples intenciones tras cada revisión) mientras que se observa que la secuencia de ediciones se simplifica después de la sección inicial. En definitiva la natureleza anárquica de la escritura colaborativa dificulta la existencia de un proceso claro y unificado, pero a pesar de ello, se observa como los conjuntos complejos de intencionalidades se van simplificando según avanza el proceso, es decir, el tipo de revisiónes

realizadas parece cambiar con el tiempo.

### 6.4.2. Análisis de la red de Petri descompuesta 2

Como vemos en la red de petri de la imagen 6.12 y al igual que la red de Petri descompuesta 1, a grandes rasgos se observa un proceso con mucha interconexión y de gran complejidad. Para facilitar su análisis se han seleccionado el inicio y el final de la red para poder observarlos más en detalle. Esto, está motivado en parte porque el inicio tiene interes especial: ver si hay un inicio específico dentro del proceso podría ayudar a identificarlo. Al mismo tiempo, el final aporta utilidad para descubrir si existen puntos finales en el proceso o es abierto y por tanto potencialmente 'infinitamente' iterativo.

Como se ve en el inicio de la red en la figura 6.17 el inicio puede ir determinado por copy-editing+elaboration+verifiability o wikification+vandalism. Otra posibilidad es comenzar con disambiguation sin embargo esto debe ser puesto en contexto. Dado que como se observa que hay caminos que vuelven al nodo inicial desde puntos más avanzados de la red, resulta normal asumir que esta intención no se da de modo inicial, sino en iteraciones futuras. En caso de comenzar realizando wikification+vandalism el flujo prosigue con copy-editing seguido de o bien de un flujo iterativo de fact-update o refactoring+wikification+elaboration o wikification+process. Es curioso ver las grandes diferencias existentes con el inicio de la otra red obtenida fruto de la descomposición (6.11). Aquí, se observan intenciones más centradas en un área específica ya sea añadir contenido o editar formato por ejemplo, fruto de editores más especializados y menos generalistas.

Por otro lado, observando el final de la red vemos que el patrón es el mismo, intenciones de mayor simplicidad al no estar combinadas, tareas de contra-vandalismo seguidas de fact-update o wikification. Sin embargo el número de caminos que se observan es muy grande, sumando cierta incertidumbre al proceso real que se pueda seguir.

### 6.4.3. Análisis del conjunto de mini redes de Petri descompuestas 3

En la figura 6.13 vemos un conjunto de redes formadas por un lugar de inicio y final y una sola transición compuesta por intenciones combinadas. Esto, no determina ningún comportamiento específico sino que es el resultado de la técnica de minería utilizada intentando de nuevo reproducir todo el comportamiento observado. Estas intenciones no han conseguido ser agrupadas dentro de ninguno de los anteriores procesos descubiertos por lo que se han convertido en redes en sí mismas para poder reproducir esa sección específica del log de eventos.

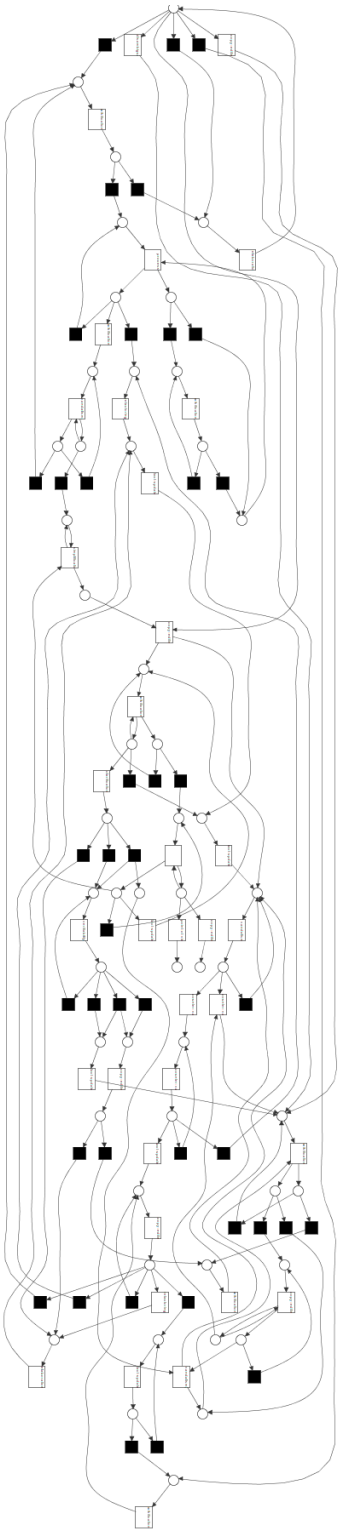


Figura 6.12: Petri net del proceso de edición tras su descomposición 2

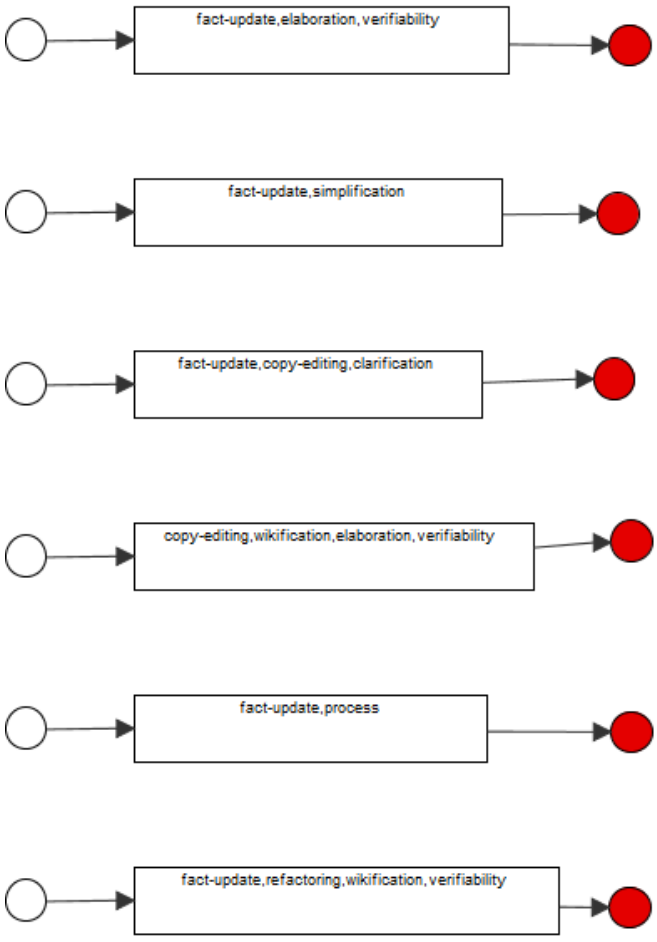



Figura 6.13: Redes de petri del proceso de edición tras su descomposición 3

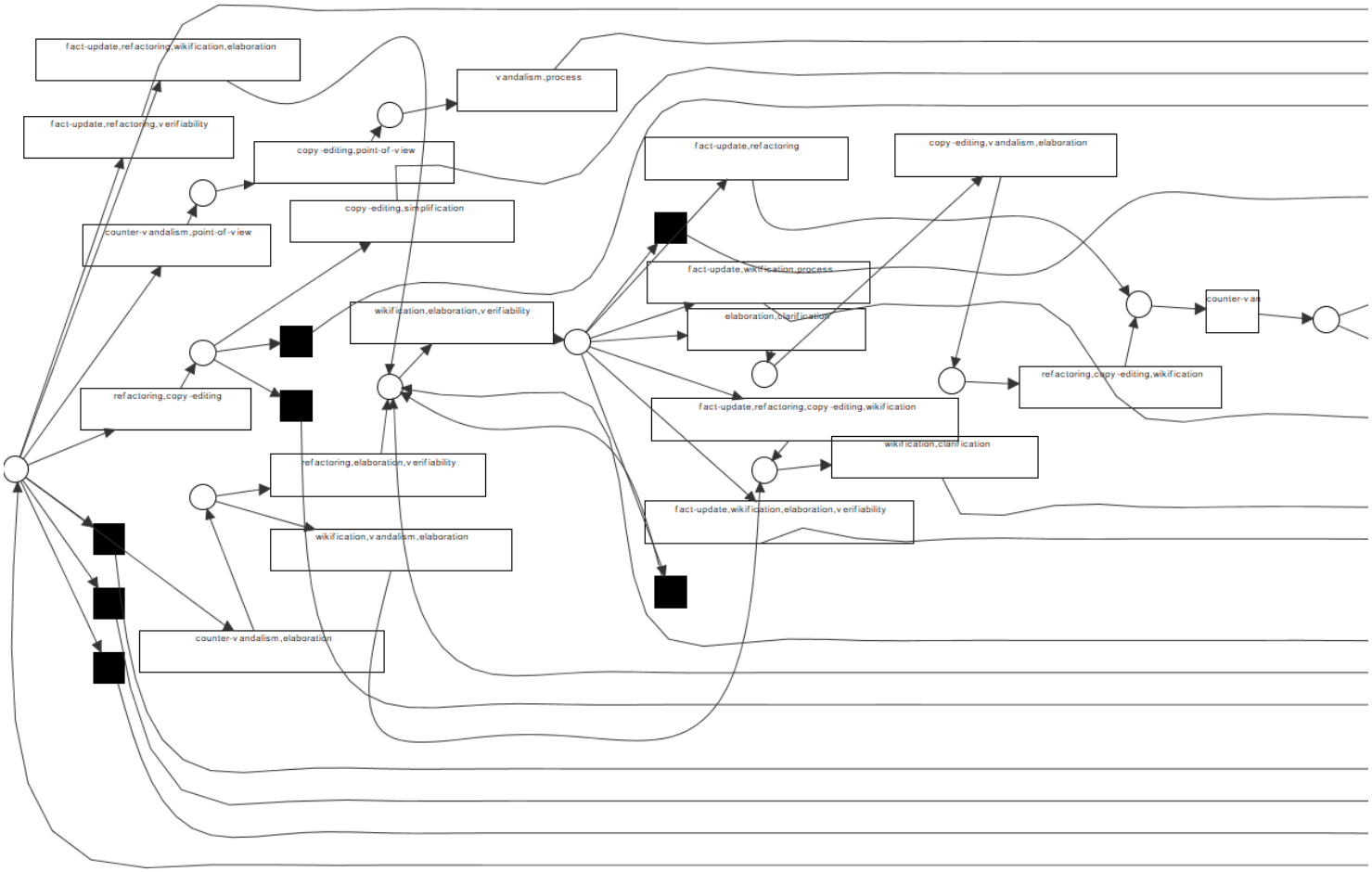



Figura 6.14: Inicio de la red de petri del proceso de edición tras su descomposición 1

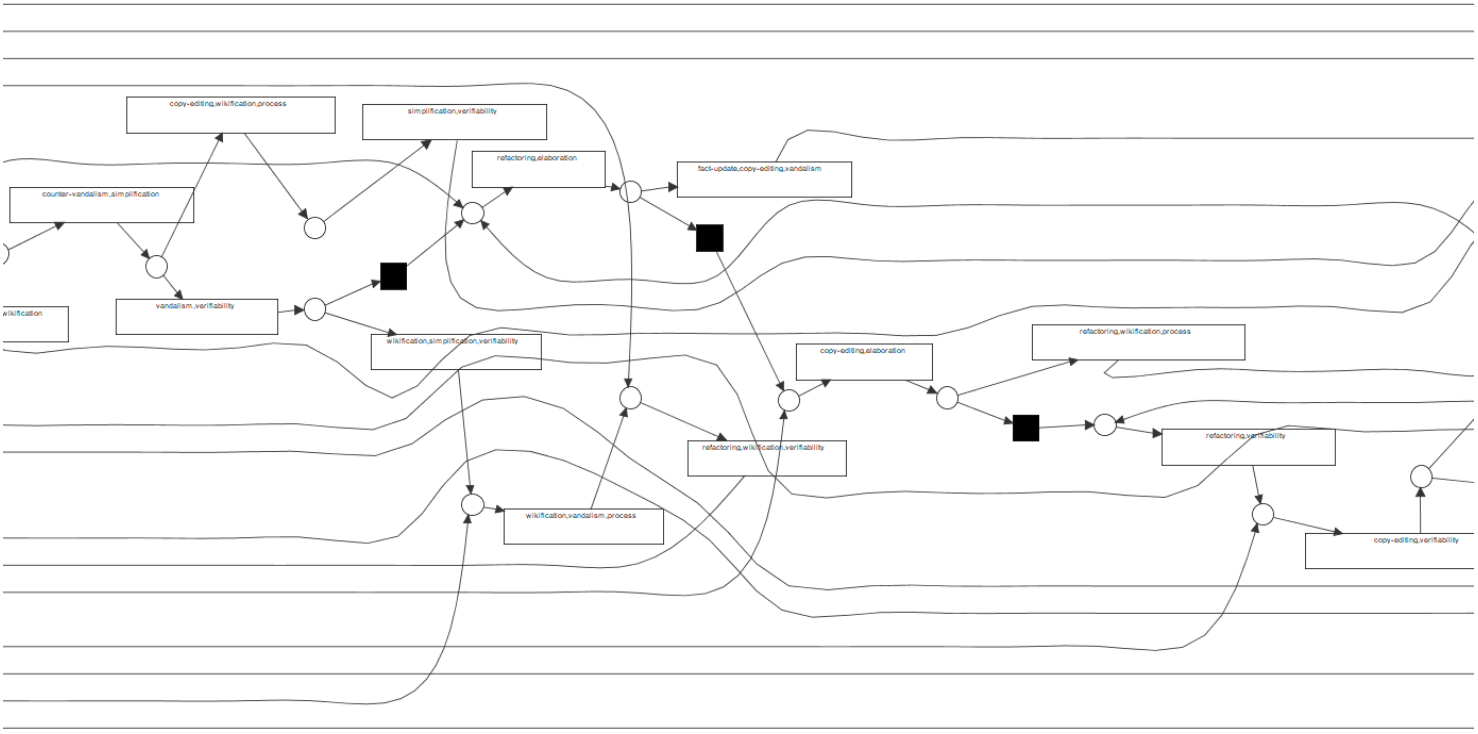



Figura 6.15: Sección intermedia de la red de petri del proceso de edición tras su descomposición 1

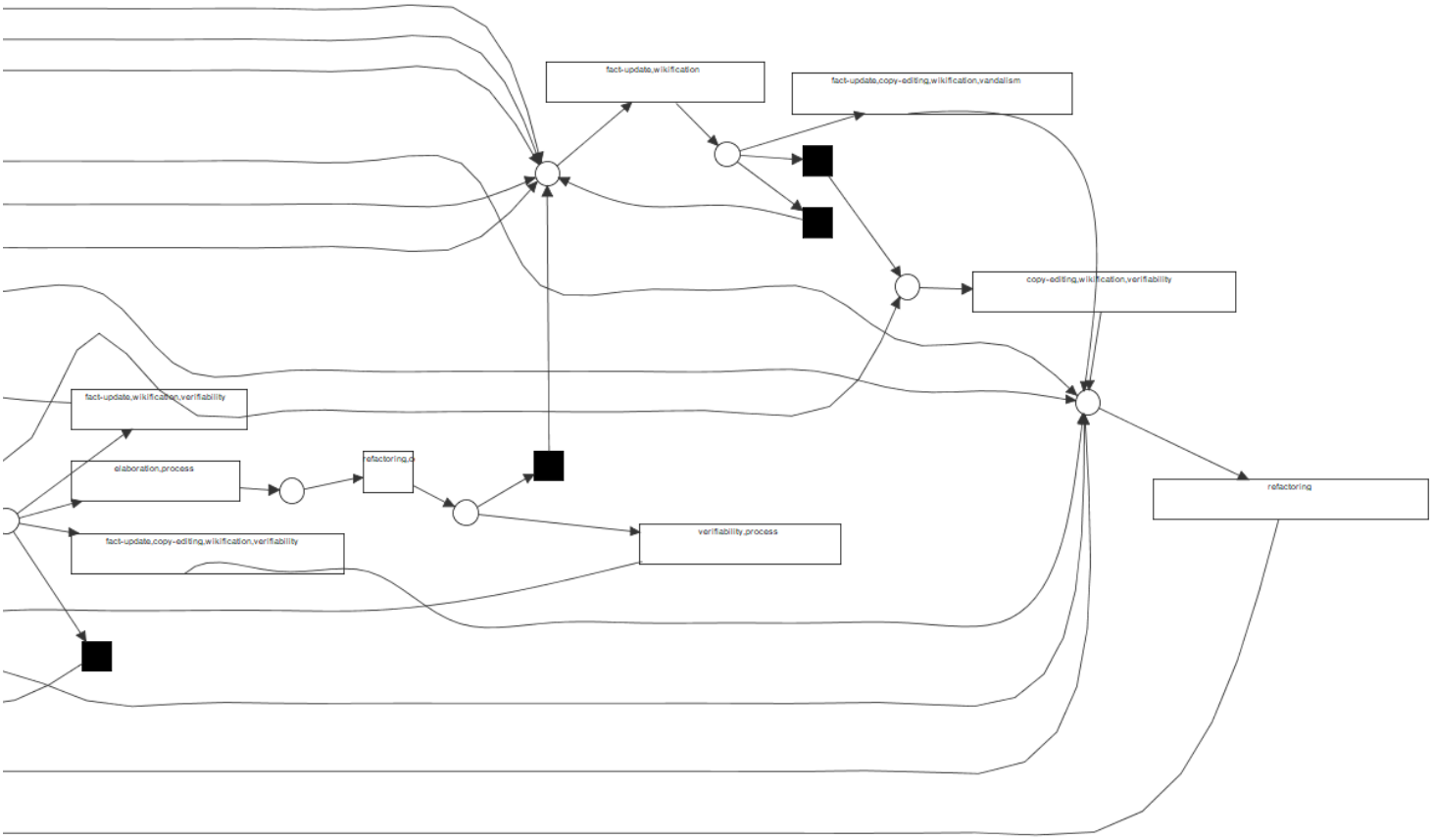



Figura 6.16: Final de la red de petri del proceso de edición tras su descomposición 1

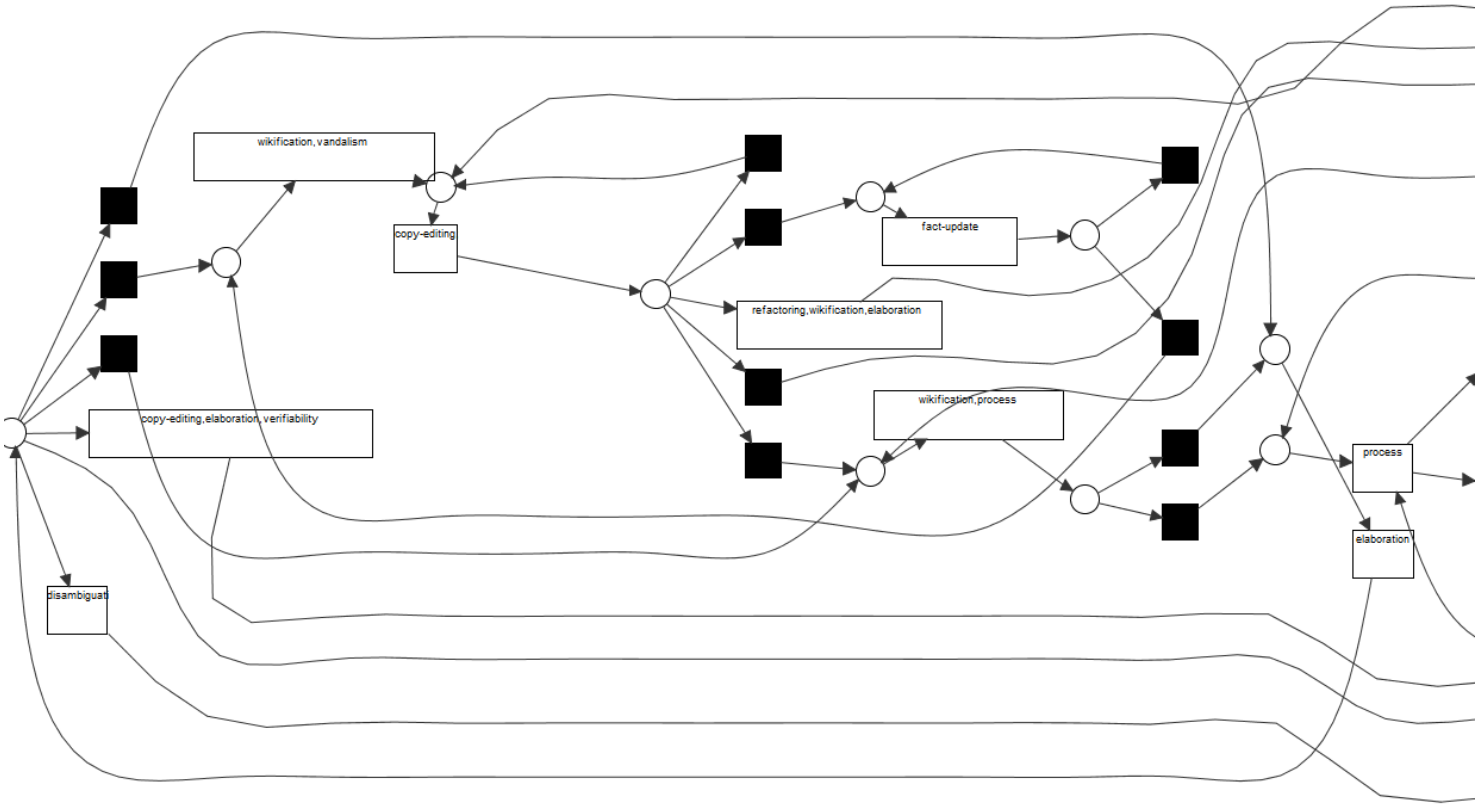



Figura 6.17: Inicio de la red de petri del proceso de edición tras su descomposición 2

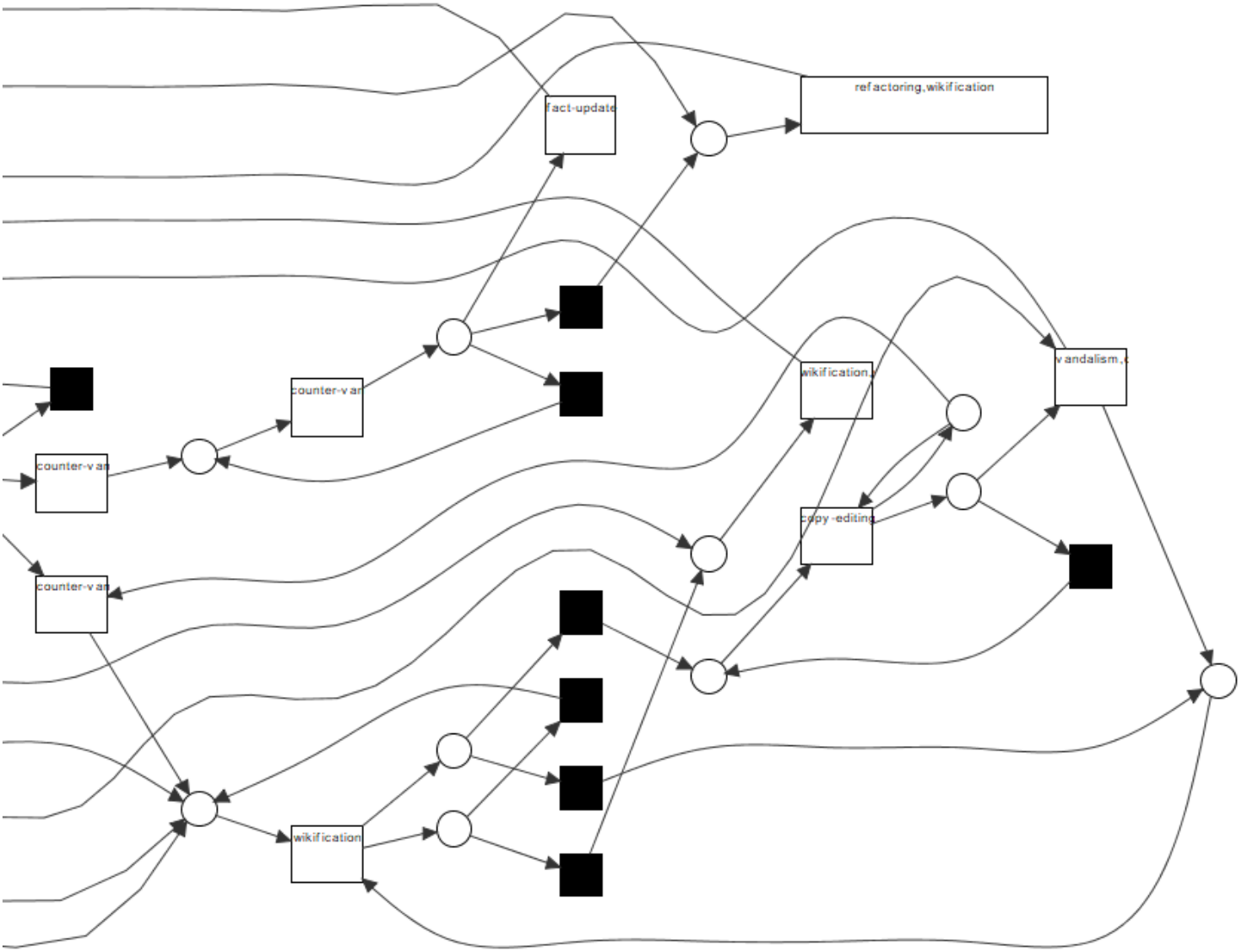



Figura 6.18: Final de la red de petri del proceso de edición tras su descomposición 2

## 6.5. Análisis a nivel editor

En esta sección se hará uso de los ficheros de revisiones de editores con actividad baja/intermedia/alta obtenidos a partir del **corpus** durante la sección Obtención de datos en formato XES. De esta manera, el objetivo es responder a la pregunta previamente formulada *¿qué procesos, en caso de existir, siguen los usuarios durante su historial de ediciones?*. La minería de procesos cobra especial interes de cara a este análisis pues podría ayudar a descubrir si los propios usuarios en función de su rol o actividad en la red siguen un patrón concreto de comportamiento mas allá de editar eminentemente con un tipo u otro de intención relacionándolo con la taxonomía de roles de usuario.

Así los log de eventos a utilizar separan a los editores en tres categorías[8] para poder descubrir si existe un proceso común a cada categoría pero diferente a las demás. Por lo que al dividir de la siguiente manera evitamos el ruido de las demás categorías.

A lo largo de las sucesivas seciones estos log de eventos serán referidos de la misma manera: fichero de revisiones de editores con actividad baja/intermedia/alta.

Estos tres ficheros, sin embargo, aún no están listos para generar un proceso que determine los diferentes caminos que siguen los editores. Necesitamos cambiar su estructura de modo que cada caso dentro del log de eventos venga determinado por cada editor en lugar de cada artículo. Para ello, hacemos uso de una herramienta de ProM cuyo objetivo es precisamente modificar cada log de eventos en base a org:resource, los editores, llamado 'Generate log from org:perspective'.

En esta sección además se aplicará el minero inductivo para el descubrimiento de cada proceso ya que permite intentar generar redes con fitness perfecto.

Sin embargo, un fitness perfecto puede dar lugar a redes que engloban muchos comportamientos diferentes y que por lo tanto imposibilitan detectar comportamientos específicos, como las redes con forma de flor. Estas redes en forma de flor son aquellas que permiten cualquier tipo de comportamiento en base a un conjunto de actividades dado. En este caso las actividades serían los eventos, es decir, la intencionalidad que determina cada revisión.

### 6.5.1. Editores de actividad baja

Una vez que contamos con el **log de eventos de los editores con baja acitividad**: solo los revisores que han realizado menos de cinco revisiones. Se aplica 'Generate log from org:perspective' y obtenemos un log de eventos compuesto por 1367 casos y 2100 eventos. Es decir, tenemos 1367 editores diferentes que han realizado 2100 revisiones. Descomponemos en grupos del mismo modo que en la sección anterior haciendo uso de la herramienta 'Discover Clusters' y aplicamos 'Discover using Decomposition' en este caso seleccionando el minero inductivo en su variante 'Perfect Fitness'. Así, obtenemos diferentes redes como vemos en las figuras 6.19, 6.20, 6.21, 6.22, 6.25

[8] Haciendo uso de corpus_filter.py (11)

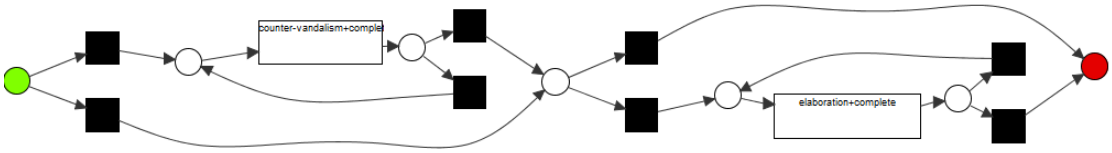



Figura 6.19: 1º Petri net del proceso seguido por los editores de baja actividad

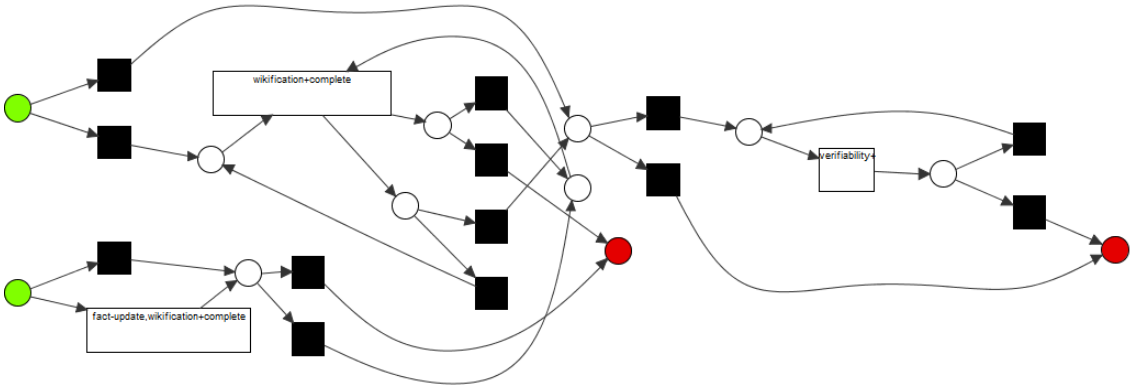



Figura 6.20: 2º Petri net del proceso seguido por los editores de baja actividad

Observamos que hay un total de cinco redes diferentes, tres de ellas teniendo dos puntos de inicio diferentes. Con esto, podemos decir que se han registrado cinco tipos de comportamientos observados entre los 1367 editores de baja actividad.

- La red 6.19 registra un flujo de ediciones donde se aplica counter vandalism y elaboration, sin embargo, dentro del flujo es posible que sólo se realize una de las dos o ninguna. Por este motivo, se puede inferir que los usuarios que siguen este proceso pueden ser englobados dentro de la categoría Watchdogs, usuarios que se dedican a hacer reversiones tras el vandalismo [5]. Por otro lado, hay determinadas intenciones asociadas a usuarios con gran experiencia y que no son adecuadas para usuarios nuevos: elaboration es una categoría relacionada con la baja supervivencia de los nuevos usuarios al ser propia de usuarios con experiencia [24]. Esto podría ser una explicación a la existencia de este flujo de trabajo dentro de los usuarios con menos de cinco revisiones. Otra posibilidad es que los usuarios tras realizar tareas de contra vandalismo, realizen pequeñas elaboraciones puntuales o que simplemente este editor aun teniendo experiencia solo haya realizado pocas ediciones en este conjunto de artículos.

- La red 6.20 representa una actividad donde se edita con la intención de wikification o wikification+fact update donde ocasionalmente, los usuarios realizaban después ediciones con intención verifiability. En este caso, esto puede relacionarse con el rol de 'Content Shaper' o 'Layout Shaper' de la taxonomía de roles previamente introducida. Las intenciones relacionadas con el formato requieren cierta experiencia en Wikipedia por lo que podrían ser editores experimentados que han tenido una implicación casual en este conjunto de artículos.

- En cuanto a la red 6.21, nos encontramos con dos puntos de inicio diferentes. Los editores

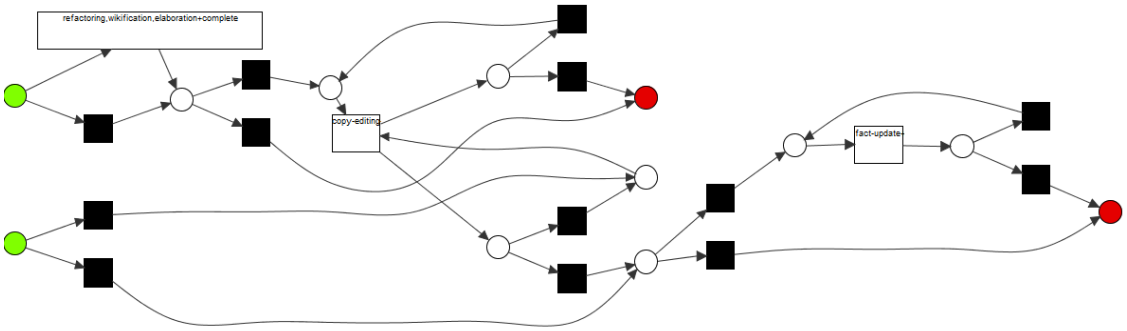



Figura 6.21: 3º Petri net del proceso seguido por los editores de actividad baja

que se engloban bajo esta forma de trabajar o bien comienzan con intenciones sofisticadas como refactoring+wikification+elaboration o bien copy-editing o fact-update, más simples y por tanto con un requisito menor de experiencia por parte del usuario. En general las diferentes posibilidades que ofrece esta red y el hecho de que por su estructura puede haber usuarios que solo hayan realizado una de estas tres intenciones hace dificil poder determinar un comportamiento específico.

- La red 6.22 no muestra mucha información respecto a ningún comportamiento real pues representa el resultado de la asociación de todos los usuarios que no han seguido un proceso específico, como aquellos que solo han realizado una revisión, entre las opciones posibles dentro de esta red, se encuentra el vandalismo. Esto es una de las desventajas comentadas anteriormente del minero inductivo. Sin embargo, se ve que entre toda esta red, hay 2 caminos que merece la pena analizar. Estos son los caminos que se observan en la parte superior e inferior.

  - En la figura 6.23 vemos la sección inferior de la red donde se observa que existe cierto posible comportamiento. Se ve un conjunto muy diverso de intenciones dentro de un bucle iterativo. Destaca como mientras que los caminos superiores cuentan con counter-vandalism+wikification en los inferiores se ve vandalismo en unión con otras intenciones como fact-update o verifiability. Este vandalismo, puede ir de la mano de usuarios novatos que vandalizan al realizar ediciones sin ser ese su objetivo. Por otro lado, la naturaleza iterativa de esta sección hace complicado extraer conclusiones respecto al flujo seguido por los propior usuarios.
  - En la sección superior ilustrada en 6.24 vemos un bucle de copy-editing+vandalism. Dado que copy-editing se centra en mejorar gramática, puntuación etc... del mismo modo que anteriormente, resulta dificil determinar si esto va determinado por un usuario novato que desconoce las reglas, o algo realizado a conciencia con el objetivo de vandalizar el artículo. Una posibilidad también puede ser la clasificación incorrecta de esta intención por parte del bosque aleatorio desarrollado en el capítulo anterior.

- Por último la red 6.25 muestra usuarios que realizan tareas de wikification y refactoring, en algunos casos, finalizando con process. La rama superior de la red muestra usuarios que han realizado solo revisiones de refactoring+wikification. El hecho de que solo hayan

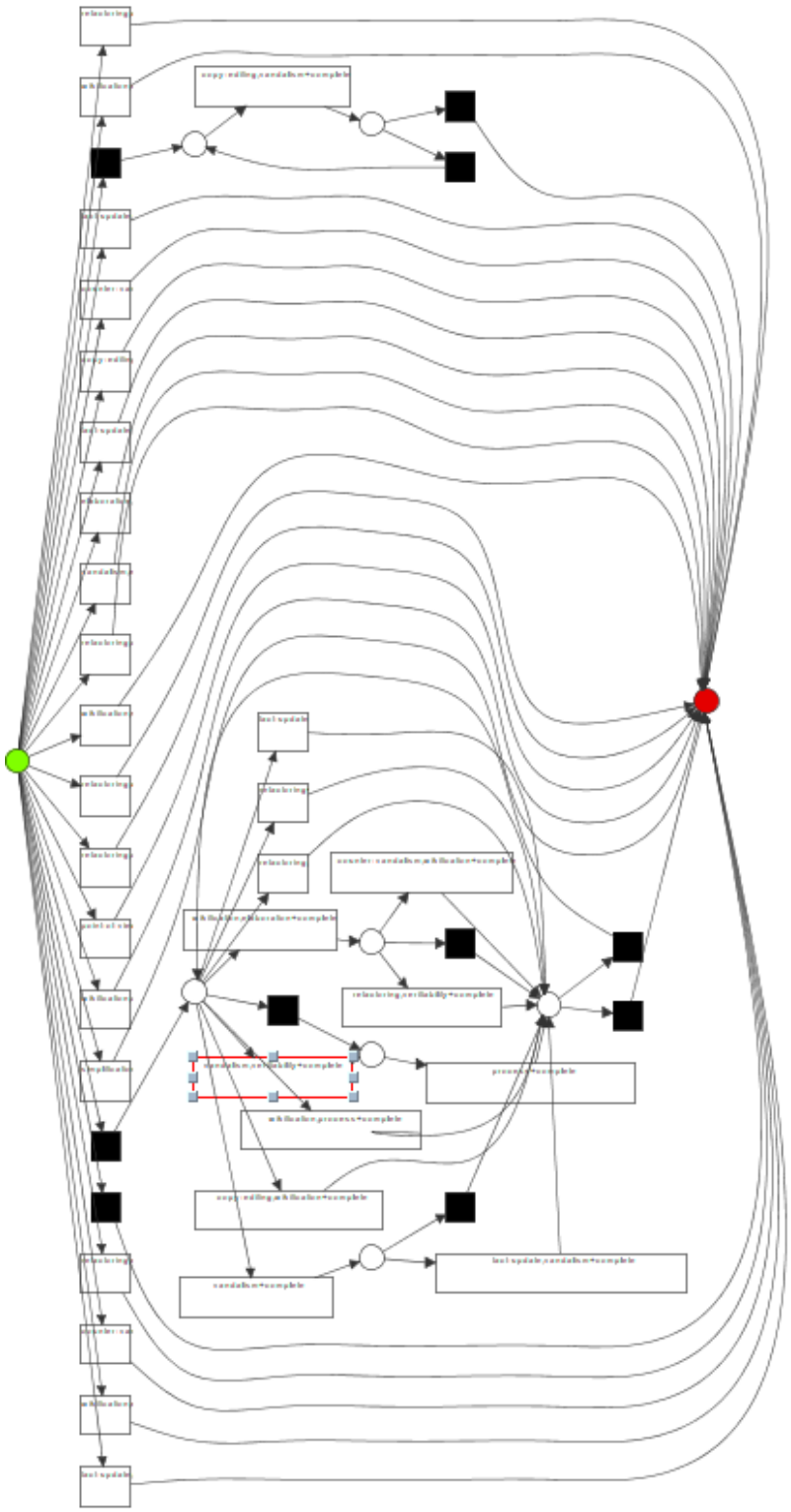


Figura 6.22: 4º Petri net del proceso seguido por los editores de actividad baja

realizado esa intención tiene sentido pues refactoring es una intención no muy apropiada para principiantes [24], aunque no hay garantía desde el alcance de este análisis para saber si los usuarios que han realizado esto son principiantes.

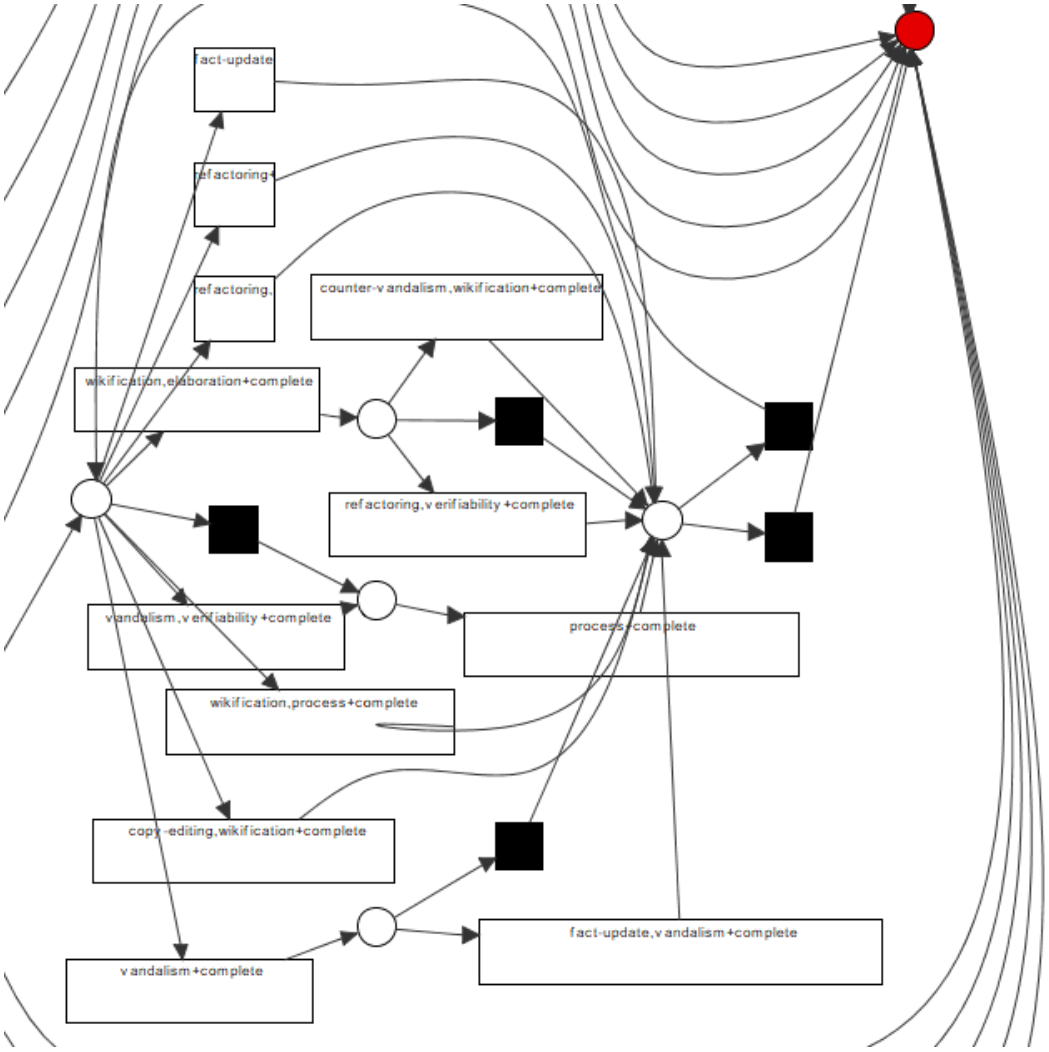



Figura 6.23: Sección inferior de la 4º Petri net del proceso seguido por los editores de actividad baja

### 6.5.2. Editores de actividad intermedia

Una vez que contamos con el **log de eventos de los editores con actividad intermedia**: que han realizado entre cinco y cincuenta revisiones y lo importamos en ProM aplicamos 'Generate log from org:perspective' de nuevo y obtenemos un log de eventos compuesto por 343 casos y 4429 eventos. Es decir, tenemos 343 editores diferentes que han realizado 4429 revisiones. Descomponemos en cluster haciendo uso de la herramienta 'Discover Clusters' y aplicamos 'Discover using Decomposition' en este caso seleccionando el minero inductivo en su variante 'Perfect Fitness'. Así, obtenemos diferentes redes como vemos en las figuras 6.26,

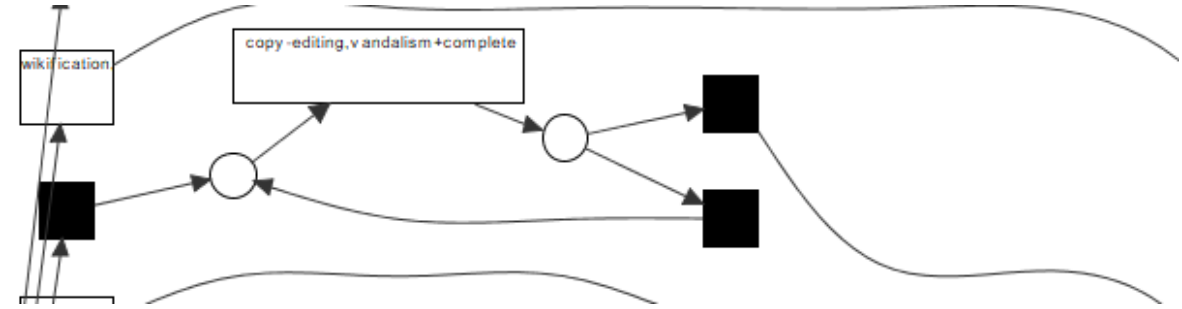



Figura 6.24: Sección superior de la 4º Petri net del proceso seguido por los editores de actividad baja

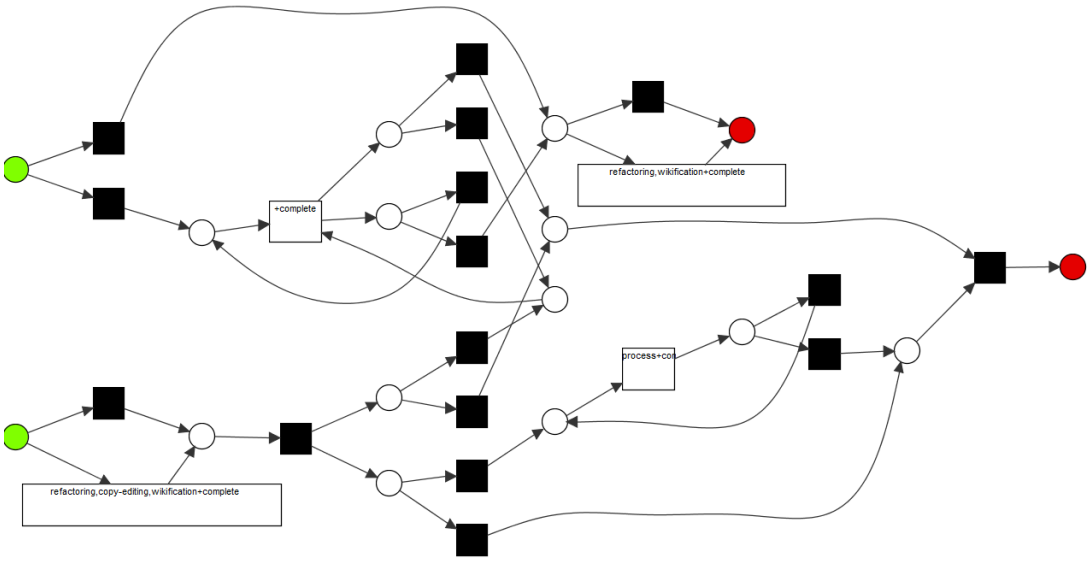



Figura 6.25: 5º Petri net del proceso seguido por los editores de actividad baja

6.27, 6.28, 6.29

En total, nos encontramos con un conjunto de redes bastante diverso, con flujos de trabajo complejos y otros razonablemente simples.

- La figura 6.26 muestra 2 flujos de trabajo diferentes. En el superior, vemos dos posibles opciones, o los usuarios editan realizando fact-update+wikification o bien realizan revisiones con la intención de elaborar. Dado que nos encontramos en un flujo seguido por revisores que realizan entre 5 y 50 revisiones, esto da lugar a que esta red superior esté compuesta por usuarios que de manera repetitiva se han centrado en un tipo de intención específica tras sus revisiones. Más concreatemente, entrarían dentro del rol de 'Quick and dirty editors' aquellos que realizan elaboration iterativamente y de 'Content Shapers' aquellos que realizan fact-update+wikification, siguiendo la taxonomía de roles introducida. En cuanto a la red inferior, los editores o bien hacen labores de wiki-

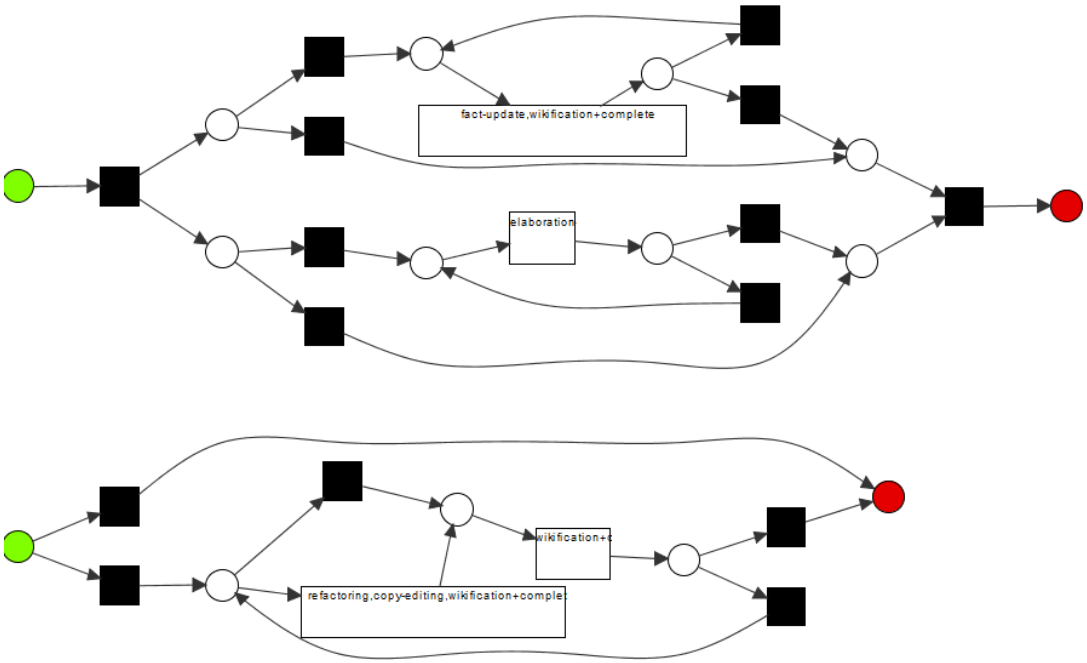



Figura 6.26: 1º Petri net del proceso seguido por los editores de actividad intermedia

fication o refactoring+copy-editing+wikification seguido de wikification, lo cual indica claramente un comportamiento propio de los 'Content Shapers'.

- La red 6.27 tiene dos puntos de inicio y es compleja en tanto que el flujo seguido es caótico.

  - En el punto de inicio superior observamos que de nuevo vemos una rama cuya única intención es el vandalismo, que puede suceder de modo iterativo, por lo que se infiere que no es una intención propia solo momentos casuales si no que hay editores que se dedican, continuamente, a realizar vandalismo en un artículo o varios. Contrastando con esto, la otra opción dentro de esta rama de la red es copy-editing+wikification iterativamente, indicando de nuevo la existencia de los usuarios denominados 'Content Shapers'.
  - Por otro lado, en la rama que comienza en el punto de inicio inferior de la red se ve un flujo iterativo de revisiones bajo la intención process, es decir, hay revisores específicamente centrados en realizar tareas muy específicas como marcar un artículo con noticias referentes a su limpieza, borrado... Además se ve como el resto de intenciones están relacionadas con la elaboración y la verificación con la ocasional wikification, de nuevo, 'Quick and dirty editors'. Sin embargo en este caso y en contra de lo comentado en la investigación 'Estabilidad turbulenta de los roles emergentes' no se observa vandalismo asociado a esto, quizá relacionado con la veteranía de los editores [5].

- La figura 6.28 nos muestra una red como la obtenida con los editores de baja actividad (6.22) donde se observa que la cantidad de posibilidades existentes implica que represen-

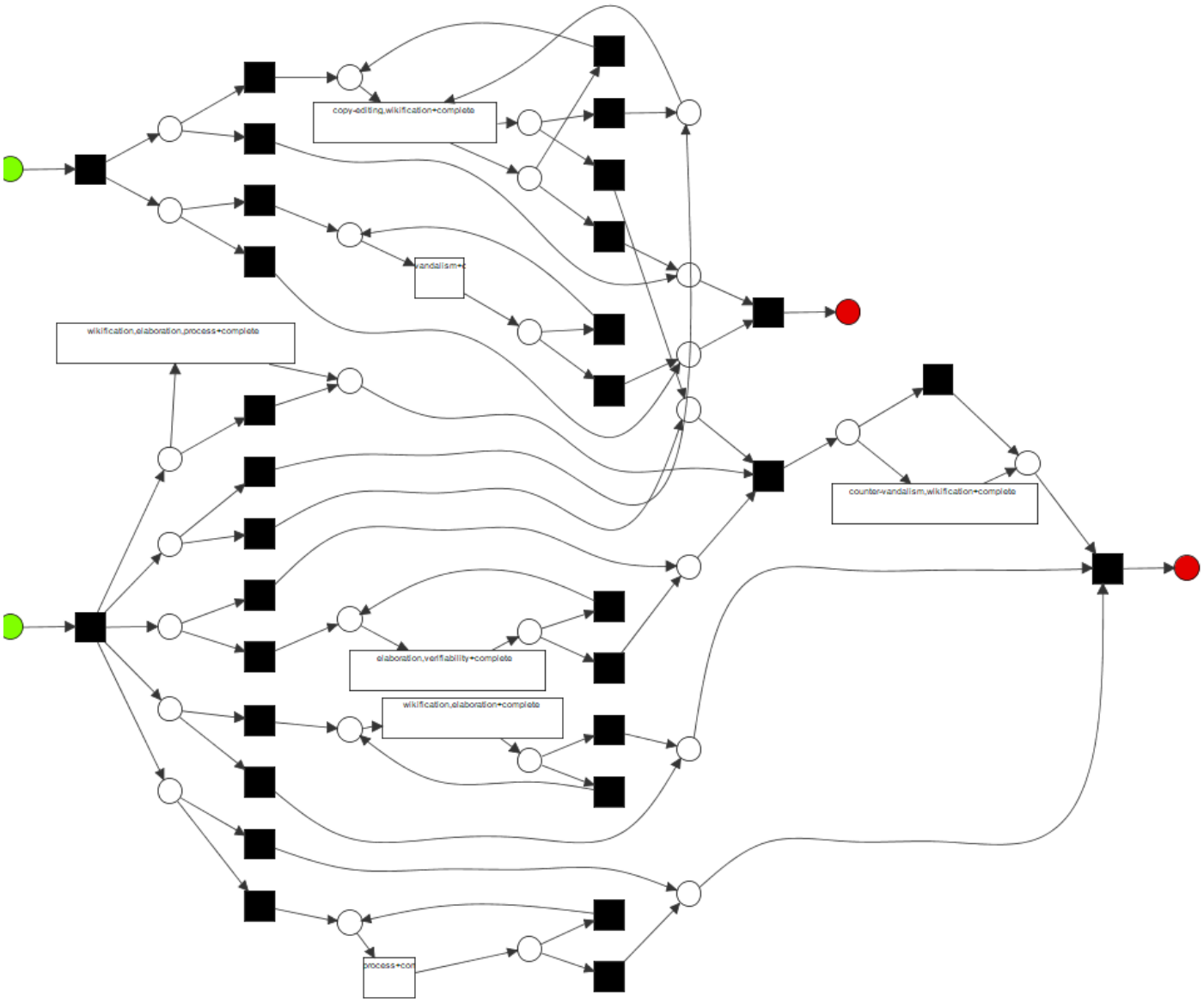



Figura 6.27: 2º Petri net del proceso seguido por los editores de actividad intermedia

ta una acumulación de todos aquellos workflows individuales que no ha logrado agrupar debido al funcionamiento del algoritmo minero inductivo.

- La red 6.29 representa un conjunto amplio de diferentes posibilidades, sin embargo, mayoritariamente las intenciones que nos muestra son copy-editing, fact-update y wikification, tanto independientemente como combinadas entre sí. Además, este proceso incluye la intención de counter-Vandalism de modo que tras realizar contra-vandalismo se vuelve de nuevo a realizar las intenciones anteriores. Esto, encaja con la descripción del rol 'All round contributors' donde este tipo de usuarios realizan tareas de adición de contenido y cambios en el texto actual además del formato y actualización de referencias.

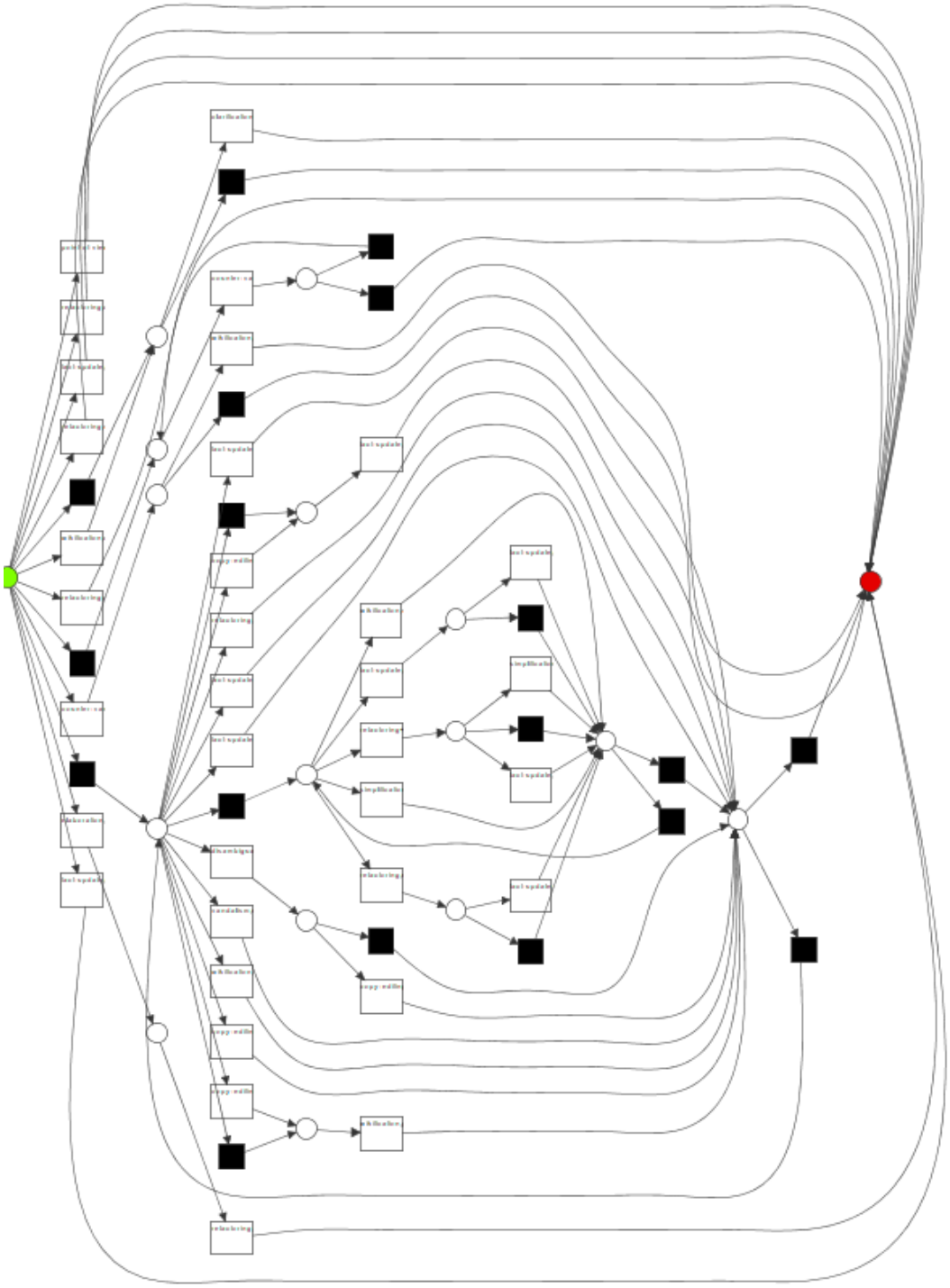


Figura 6.28: 3º Petri net del proceso seguido por los editores de actividad intermedia [5].

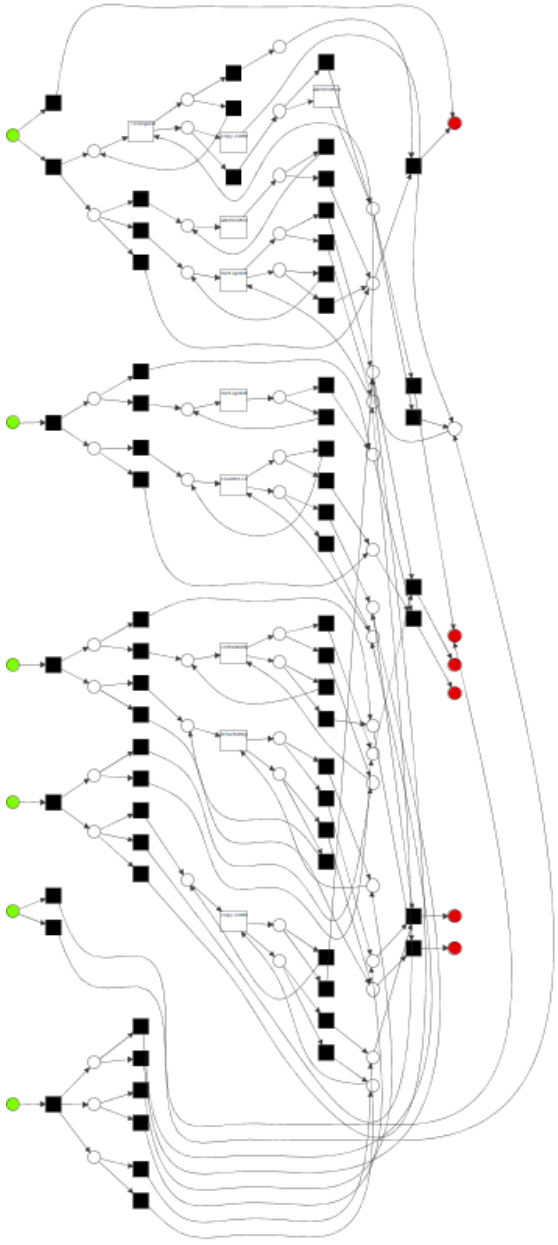


Figura 6.29: 4º Petri net del proceso seguido por los editores de actividad intermedia

### 6.5.3. Editores de actividad alta

Por último, nos encontramos ante el **log de eventos de los editores con actividad alta** compuesto por aquellos revisores con más de 50 revisiones a sus espaldas, es decir, aquellos con una alta actividad. Excluyendo, además, los agrupados como Anónimo. Esto es debido a que no se espera que los anónimos puedan aportar un flujo coherente de trabajo, además de que por el tipo de algoritmo de minería aplicado, sólo contaminaría las redes. Aplicamos 'Generate log from org:perspective' de nuevo y obtenemos un log de eventos compuesto por 37 trazas y 3712 eventos. Es decir, tenemos 37 editores diferentes que han realizado 3712

revisiones. Descomponemos en clusters haciendo uso de la herramienta 'Discover Clusters' y aplicamos 'Discover using Decomposition' en este caso seleccionando el minero inductivo en su variante 'Perfect Fitness'. Así, obtenemos diferentes redes como vemos en las figuras 6.30, 6.31, 6.32

Contamos con cinco redes diferentes, de las cuales dos son de considerable complejidad mientras que las otras tres son sorprendentemente simples.

- Las redes de la figura 6.30 sorprenden por su simplicidad teniendo en cuenta el número tan alto de ediciones que tienen los editores en este conjunto.

    1. La red superior se compone de simplemente de refactoring+copy-editing. Estos editores, entrarían en el rol de 'Copy-editors' y 'Content-shapers' al mismo tiempo de la taxonomía de roles. Esto, claramente denota usuarios completamente dedicados a unas intenciones específicas. Sin embargo, al representar dos roles diferentes, se propone añadir a la taxonomía el rol de 'Article fixers' ya que arreglan errores gramáticales y erratas además de organizar el texto existente mediante tareas de formato.
    2. La red intermedia representa tres comportamientos diferentes entre sí y excluyentes:
        *a*) El primero se basa usuarios que sólo editan con la intención de copy-editing+wikification los cuales podrían entrar dentro del rol propuesto previamente 'Article Fixers' ya que además de arreglar errores gramáticales y erratas, arreglan el formato del artículo con wikification.
        *b*) El segundo comportamiento está determinado por vandalismo+elaboration. Esto, a diferencia de la sección anterior con los usuarios de actividad intermedia, corrobora el comportamiento esperado por aquellos usuarios bajo el rol 'Quick and dirty editors' de la taxonomía. Dando lugar a que efectivamente este fenónemo de realizar vandalismo+elaboration no es dependiente de la veteranía del usuario, si no que es algo propio de este estilo de edición.
        *c*) El tercer comportamiento está formado únicamente por el contra-vandalismo, por lo que hay usuarios que actúan bajo el rol de 'Watchdog' salvaguardando los artículos a lo largo del tiempo. Por último, la red inferior se compone de diferentes posibilidades sin embargo todas las intenciones encontradas (wikification, fact-update+verifiability, simplification y fact-update+wikification+refactoring) son diversas y afectan tanto a formato como contenido por lo que podrían ser propias de 'All round contributors'.

- La red 6.31 cuenta con cinco puntos de origen diferentes y es bastante compleja pues los caminos se entrelazan entre sí en numerosos y diferentes bucles. En primer lugar y debido a su estructura, hay muchas intenciones diferentes dentro de esta red. Esto, sumado a la gran cantidad de caminos y bucles existentes da lugar a que los usuarios bajo el rol 'All-round contributors' entren en este flujo. Sin embargo, se observan más roles diferentes. En el punto de inicio superior se observan bucles en las intenciones fact-update y elaboration. La ruta de elaboration puede ir directamente hacia el final del flujo por lo que este proceso lo seguirían aquellos bajo el rol 'Quick and dirty editors', sin embargo y de nuevo, no se observa que haya vandalismo asociado a sus revisiones

a diferencia de lo encontrado por Daxenberger [5]. Por otro lado las demás intenciones están relacionadas con formato y arreglo del artículo, por lo que de nuevo dentro de este flujo se pueden encontrar también 'Article Fixers'.

- La 6.32 no sirve para extrar un comportamiento determinado pues representa una acumulación de todos aquellos flujos de trabajo individuales que no ha logrado agrupar debido al funcionamiento del minero inductivo al igual que en las redes 6.28 y 6.22 de el log de eventos de los editores con actividad intermedia y baja respectivamente.

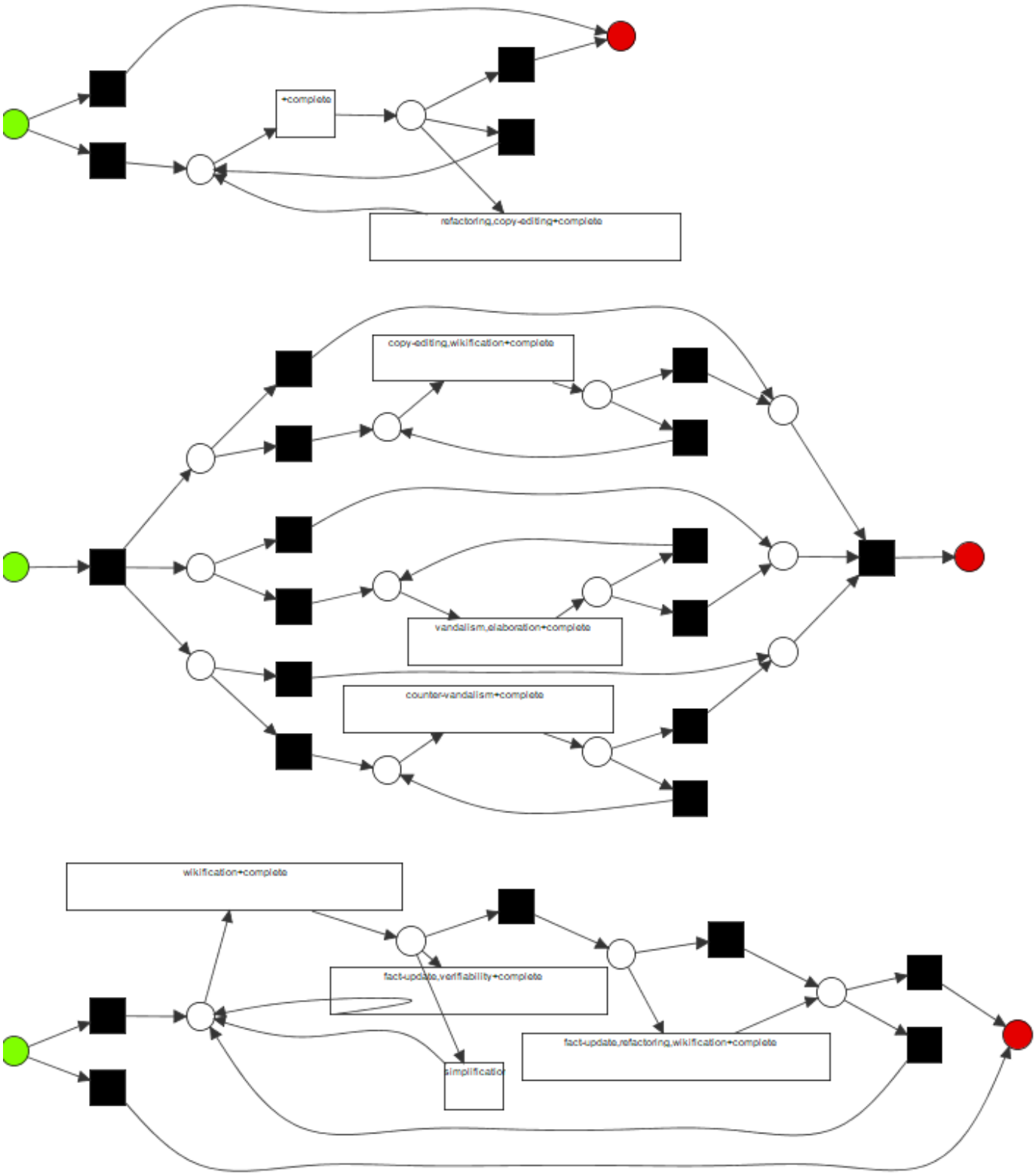



Figura 6.30: 1º Petri net del proceso seguido por los editores de actividad alta

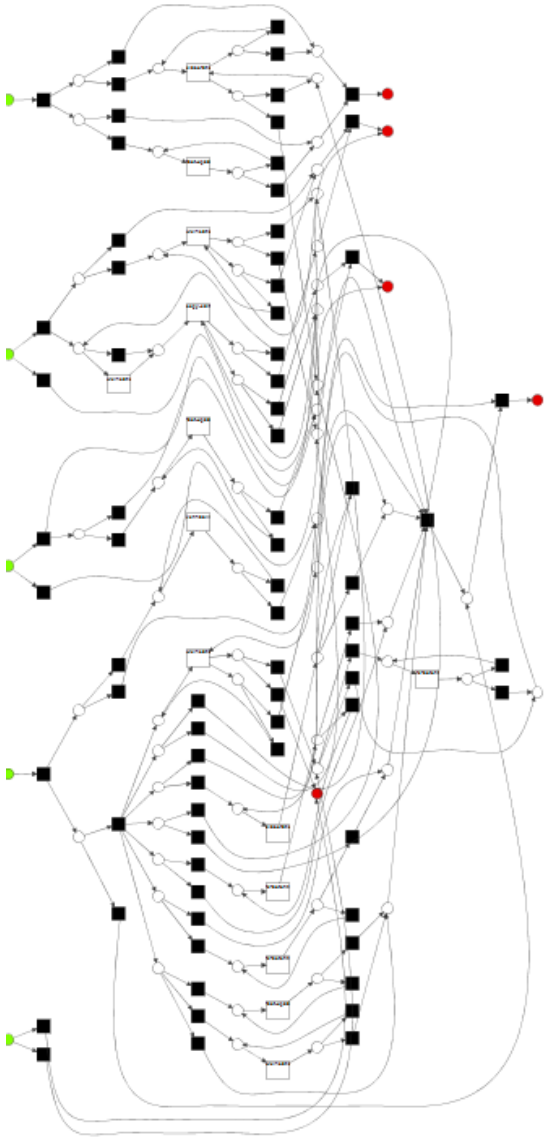


Figura 6.31: 2º Petri net del proceso seguido por los editores de actividad alta

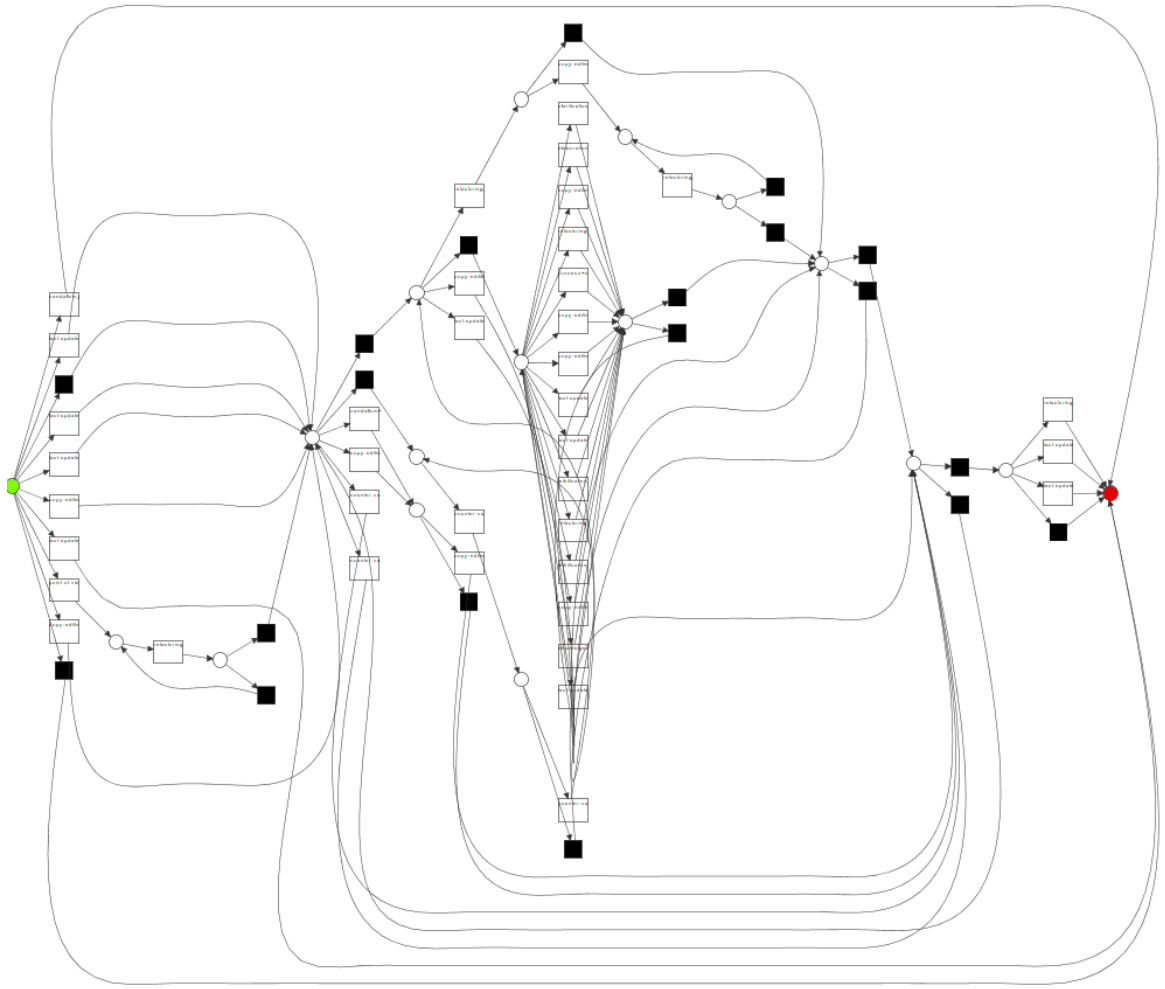


Figura 6.32: 3º Petri net del proceso seguido por los editores de actividad alta

# Capítulo 7

# Análisis con minería social

Retomando lo detallado en el capítulo 3, la minería social hace uso de técnicas de sociometría y análisis de redes sociales [3] para observar y conocer las posibles relaciones existentes entre los diferentes elementos existentes en un log de eventos. De esta manera, se tratará de realizar un estudio de las relaciones existentes entre los editores de un artículo de Wikipedia.

La herramienta ProM 6.8 brinda una librería basada en métricas establecidas por Wil M.P. van der Aalst en la investigcación 'Descubriendo redes sociales en logs de eventos' ([2]). Estas métricas son handover of work, subcontracting, working together y similar task.

Estas métricas representan las relaciones existentes en forma de grafo donde cada actor es un nodo y cada arista una relación, dependiendo el peso de la intensidad de esta relación y hacen uso de los mismos ficheros de datos que la minería de procesos: log de eventos.

Durante esta sección se hará uso de dos *artículos destacados* de Wikipedia como log de eventos. Estos artículos serán analizados y comparados inidividualmente para determinar diferencias y similaritudes. En concreto el proceso a seguir será el siguiente:

1. Obtención de datos: Se determinará qué artículos de Wikipedia serán analizados y el modo en el cual se han obtenido.

2. Handover of Work: Aplica handover of work a cada artículo para observar el relevo de trabajo entre los editores.

3. Subcontracting: Del mismo modo que aplicamos handover or work se aplicará subcontracting a ambos artículos por separado para determinar la 'subcontratación' entre editores.

Se ha determinado que la aplicación de Working Together no aporta información al estudiar artículos individuales por lo que esta no se aplicará. Esto es debido a que al medir la frecuencia con la cual los editores trabajan juntos en el mismo artículo, para que pueda aportar resultados interesantes se necesita un conjunto de artículos de mayor tamaño, como por ejemplo, una wiki entera.

Adicionalmente, tampoco se hará uso de Similar-Task. En los criterios de similitud de similar task la frecuencia tiene un peso importante, haciendo que aunque dos editores editen de la misma manera, una frecuencia de edición ligeramente menor podría hacer que se computen

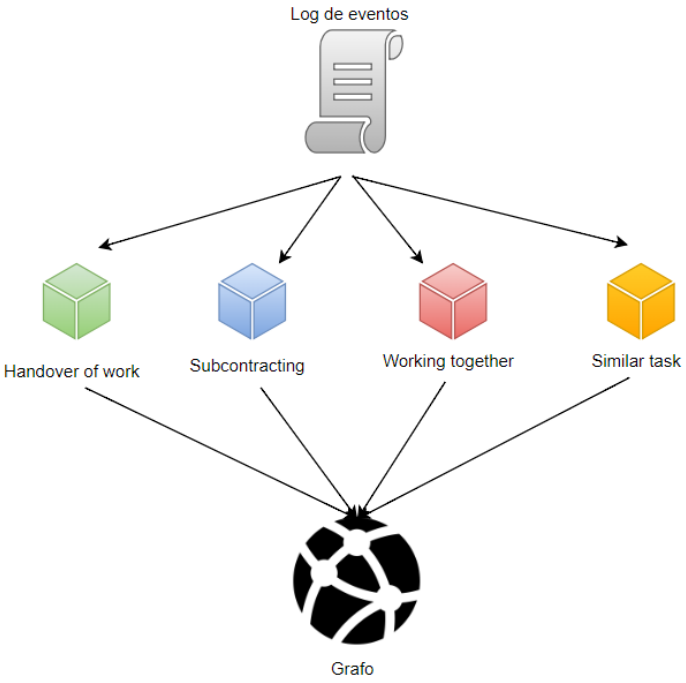



Figura 7.1: Estructura de la minería social en ProM

como usuarios de diferente comportamiento y dada la gran variedad de cantidad de revisiones en nuestros usuarios esto puede afectar a la interpretación de los resultados.

## 7.1. Obtención de datos

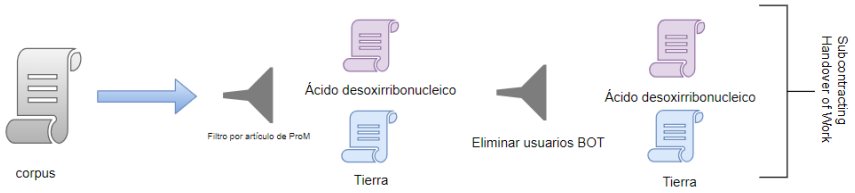



Figura 7.2: Proceso a seguir para la obtención de los datos usados en la minería social

En este capítulo la obtención de los datos es considerablemente más simple que en la sección anterior. Esto es debido a que el punto de partida es el propio **corpus** utilizado como log de eventos en la sección anterior. El proceso a seguir será el observado en el esquema 7.2

Los datos a utilizar serán:

- Handover of Work y Subcontracting harán uso de dos artículos de entre los existentes en el **corpus** por separado. Estos serán separados del **corpus** mediante un filtro de ProM que permite separar por traza los log de eventos y los usuarios BOT serán eliminados mediate un script de filtrado.[1]

Los archivos seleccionados han sido, de entre los ocho existentes en el **corpus**: el artículo Tierra y Ácido desoxirribonucleico pues tienen un período de vida similar, existiendo desde los inicios de Wikipedia hasta el día de hoy y por tanto teniendo una rica historia de ediciones. Sin embargo, existe la necesidad de filtrar a sus editores BOT.

Los usuarios BOT son programas informáticos automatizados que realizan tareas repetitivas de revisión por lo que de cara a este análisis esto puede emborronar las existentes colaboraciones entre los editores reales. Por este motivo, han sido eliminados de cara a estudiar el handover of Work y el Subcontracting de los artículos seleccionados.

## 7.2. Handover of Work

En esta sección se aplicará el algoritmo de minería social Handover of Work a los dos artículos seleccionados del **corpus**: Ácido desoxirribonucleico y Tierra con los usuarios BOT filtrados.
Handover of work consistía en estudiar relevo en el trabajo de un individuo $i$ a un individuo $j$ si hay dos actividades subsecuentes entre ellos dentro de un log de eventos [2]. Así, sus representaciones gráficas son representadas siguiendo los mismos parámetros en ambos casos. Los colores representan los diferentes grupos existentes y están organizado de modo que todos los grupos se encuentren juntos. Debido a que la paleta de colores es reducida se pueden observar grupos del mismo color separados, por lo que se debe tener en cuenta color y región en la que se encuentre a la hora de determinar los distintos grupos existentes. Esta agrupación se basa

[1] corpus_filter.py (11)

en el peso de las aristas, cuantas más veces el editor $j$ haya editado tras $i$ y viceversa, mayor será el peso de la arista que los una.

Por otro lado el tamaño del nodo dependerá de su grado. En este caso, aquellos editores que hayan realizado más revisiones antes o después que otros serán representados por un nodo de mayor tamaño.

### 7.2.1. Artículo Tierra

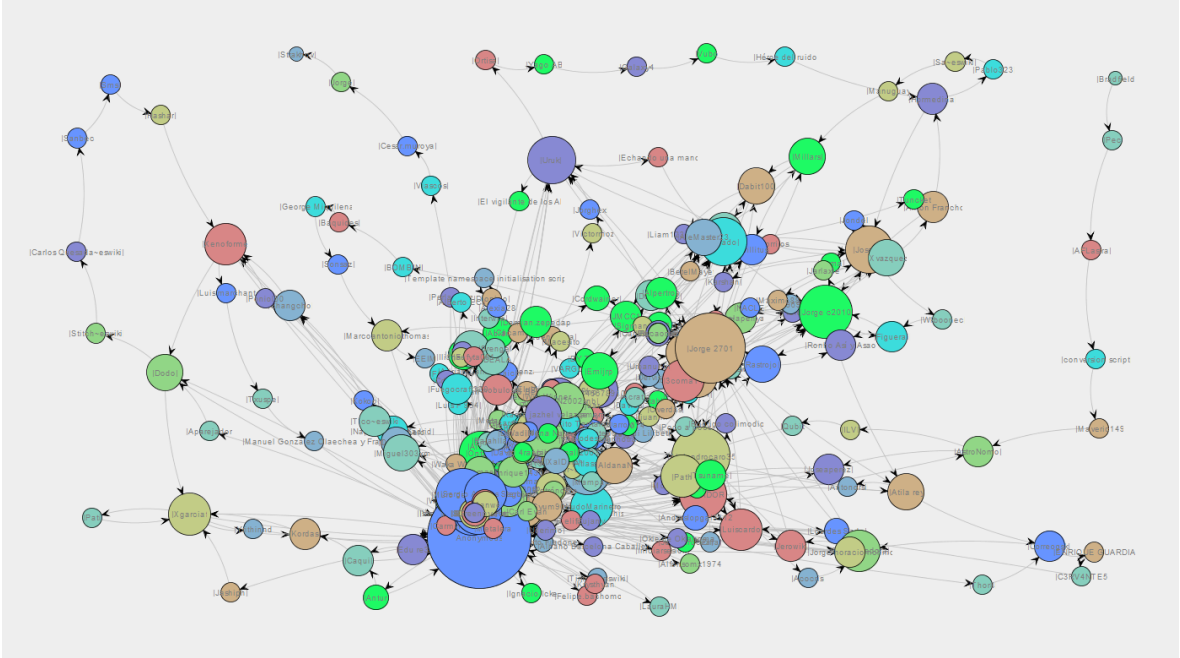


Figura 7.3: Grafo handover of work del artículo Tierra

El artículo tierra consta de 2654 revisiones y 460 editores. Por lo tanto contamos con una gran cantidad de editores propia de la antiguedad del artículo. Esto va a dar lugar a un grafo de handover complejo, con un gran número de nodos y aristas.

En la figura 7.3 se encuentra el grafo de handover of work generado para el artículo Tierra tras filtrar a los usuarios BOT. Una gran cantidad del conjunto de editores se encuentra en la zona central del grafo mientras que hay una porción de los usuarios que se encuentran en la periferia estando conectados únicamente con 1 o 2 revisores y con un tamaño reducido. Esto pueden ser ediciones realizadas por usuarios que han trabajado en momentos puntuales, sin embargo los motivos de esto son desconocidos dentro del alcance de este análisis.

Se ve como sin embargo no todos los nodos de la periferia han realizado pocas revisiones, algunos tienen un mayor tamaño implicando un mayor grado y su conexión con los nodos del conjunto central es casi directa, sin embargo sus aristas de mayor peso están conectadas con los nodos periféricos y por ello se encuentran en esa localización. Esto podría implicar alta actividad por parte de estos usuarios pero en una franja de tiempo reducida dando lugar a

colaboraciones con una piscina de usuarios menor que un editor que haya podido estar realizando pocas ediciones pero a lo largo de toda la vida del artículo, que en este caso, son más de 15 años.

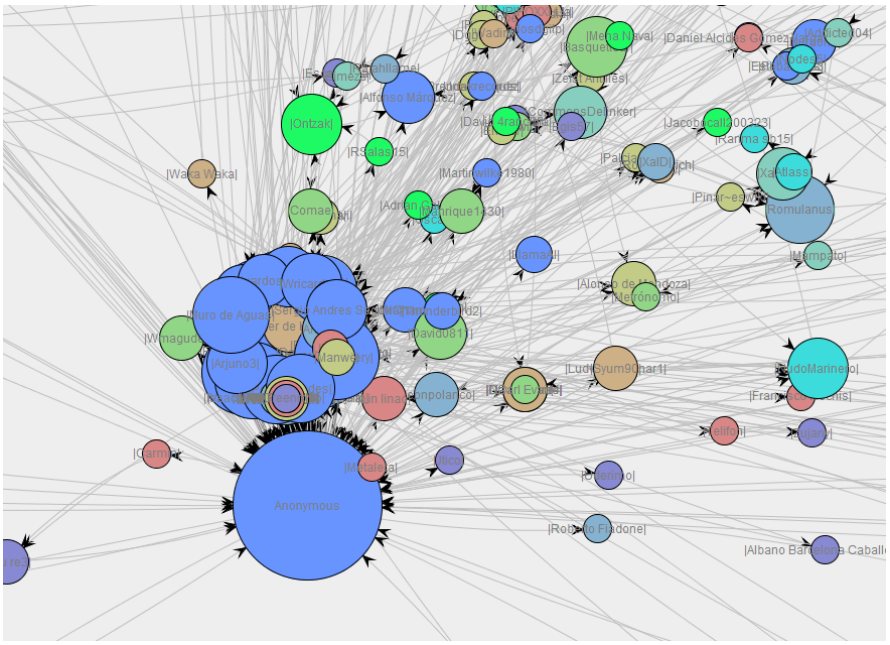



Figura 7.4: Zoom grupo central del grafo handover of work del artículo Tierra

Centrando el foco de atención en el grupo central, tenemos la ampliación del grafo anterior en la figura 7.4. Claramente se observa un grupo muy cercano de usuarios con un grado alto a juzgar por su tamaño en conjunto con otros de menor tamaño. Debido a su centralidad dentro del grafo estos nodos están relacionados con los usuarios de más peso dentro del artículo tal y como se puede ve por el nodo que representan los usuarios anónimos y su enorme cantidad de aristas. Del mismo modo y en menor medida se observa una gran cantidad de aristas conectando con el conjunto de usuarios agrupados (color morado) que se observa.

Tal y como se comenta antes del comienzo de la sección, el color de los nodos y su localización determina su pertenencia a un grupo específico. Aquí vemos como todo este conjunto central de usuarios influyentes se encuentra coloreado con el mismo color, indicando que entre sí forman un único conjunto. De esto se extrae que aquellos usuarios que hacen numerosas ediciones, en la mayoría de los casos, son activos durante un largo periodo de tiempo, cosechando un alto número de conexiones bajo la métrica handover of work.

### 7.2.2. Artículo Ácido desoxirribonucleico

En este caso contamos con un artículo compuesto por 3308 ediciones y 396 editores. Del mismo modo que anteriormente, esto da lugar dará lugar a un grafo de handover complejo, con un gran número de nodos y aristas.

El grafo resultante, representado en la figura 7.5, muestra una situación similar al observado en el artículo Tierra: la mayoría de editores se encuentran agrupados en el centro del

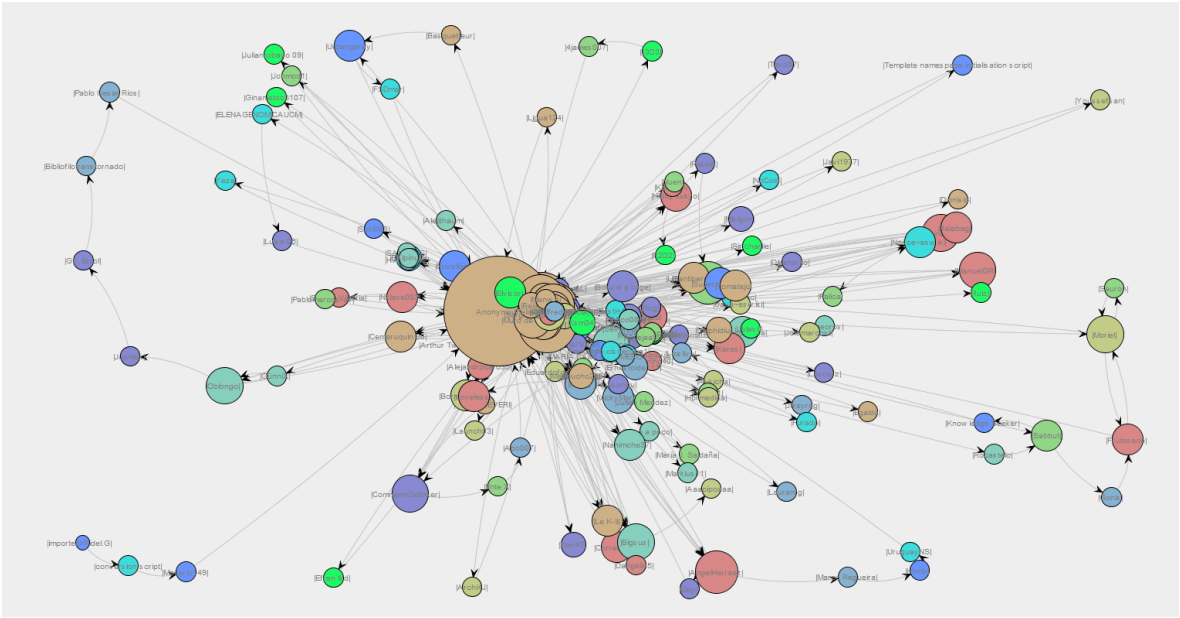


Figura 7.5: Grafo handover of work del artículo Ácido desoxirribonucleico

grafo mientras que una pequeña porción de los mismos se encuentra en la periferia.

Estos nodos que componen la periferia siguen la misma estructura que en el caso del artículo Tierra. La mayoría son nodos pequeños con conexiones a uno o dos nodos con una porción de nodos de mayor tamaño y por ende grado probablemente debido a editar durante un espacio pequeño en el tiempo aunque con un grado de actividad mayor que los demás.

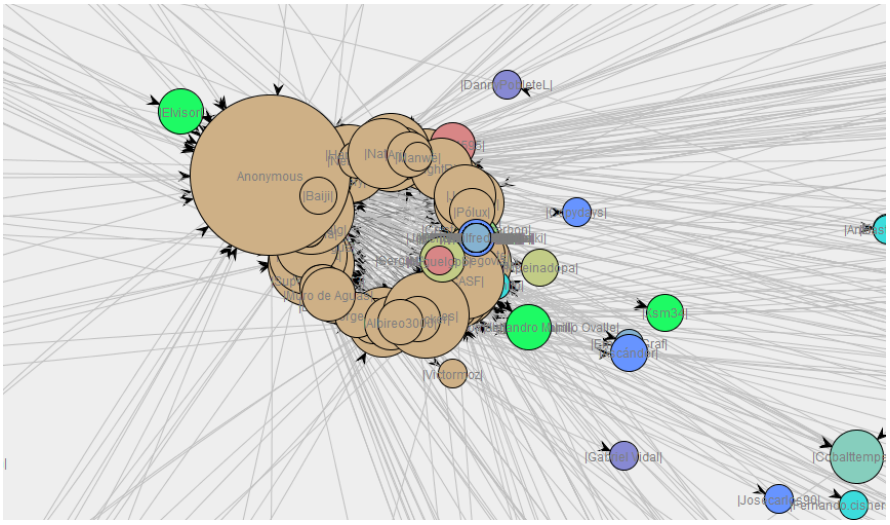



Figura 7.6: Zoom grupo central del grafo handover of work del artículo Ácido desoxirribonucleico

Poniendo el foco en el grupo central, visible en 7.6, vemos de nuevo un fenómeno similar

al anterior artículo, un conjunto de editores pertenecientes al mismo grupo como denota su color y localización de un tamaño grande indicando una gran cantidad de conexiones. Es decir, los usuarios más influyentes a lo largo de la evolución del artículo se encuentran aquí. Además, se encuentran relacionados entre ellos, indicando colaboración entre los mismos. Básicamente, la conclusión que puede extraerse es la misma que antes: estos editores que hacen numerosas ediciones, en la mayoría de los casos, son activos durante un largo periodo de tiempo, cosechando un alto número de conexiones bajo la métrica handover of work.

## 7.3. Subcontracting

En esta sección se aplicará el algoritmo de minería social Subcontracting a los dos artículos seleccionados del **corpus**: Ácido desoxirribonucleico y Tierra con los usuarios BOT filtrados. Subcontracting cuenta el número de veces que un individuo $j$ ejecuta una actividad entre dos actividades ejecutadas por el individuo $i$ [2]. Así, sus representaciones gráficas son representadas siguiendo los mismos parámetros que en el caso del Handover of Work: los colores representan los diferentes grupos existentes y están organizado de modo que todos los grupos se encuentren juntos. Esta agrupación de nuevo se basa en el peso de las aristas, cuantas más veces el editor j haya editado entre dos ediciones del editor i y viceversa, mayor será el peso de la arista que los una.

Por otro lado el tamaño del nodo también dependerá de su grado y su situación geográfica dentro de la red afecta a su pertenencia a determinado grupo a pesar de su color.

### 7.3.1. Artículo Tierra

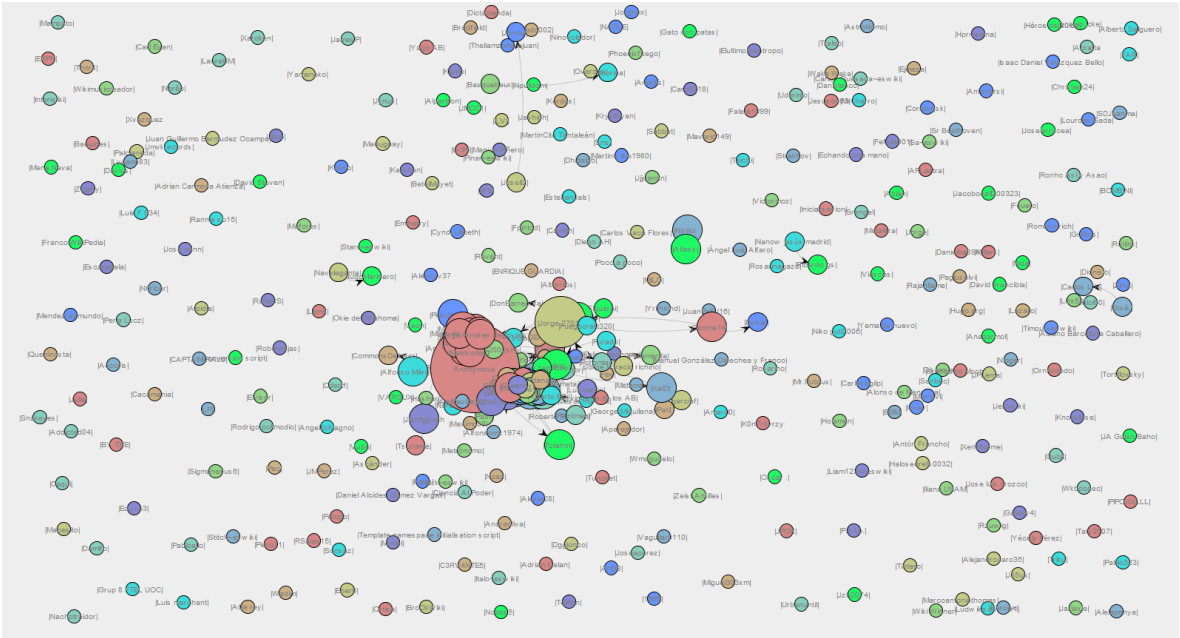


Figura 7.7: Grafo subcontracting del artículo Tierra

Observando el grafo resultante en la figura 7.7 se ve que la mayor parte de los nodos se

encuentran aislados. Esto, en el subcontracting, se traduce a que son usuarios que han editado durante una sola sesión. En subcontracting los vértices se forman entre los editores $i$ y $j$ si $i$ edita entre dos ediciones de $j$. Dado que en este caso no hay conexiones, solo realizaron revisiones una vez, sin interrupciones de otros usuarios. El número de revisiones realizado no está determinado, puede ser una o más, pero carecemos de esa información bajo esta perspectiva. En resumen, se puede extraer lo siguiente: la mayoría de editores editan durante una sola sesión en este artículo. Sin embargo, también encontramos un grupo de usuarios en el centro del grafo.

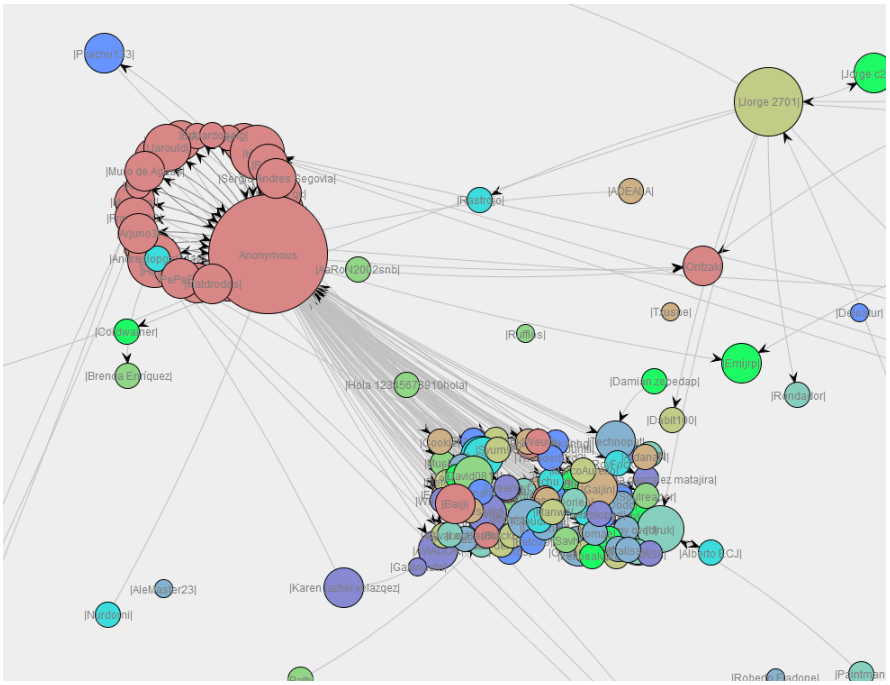



Figura 7.8: Zoom grupo central del grafo subcontracting del artículo Tierra

Este grupo central, visualizable en 7.8, vemos que en realidad está compuesto de dos grupos diferentes.

- El primer grupo, compuesto por nodos grandes en color rojo. Vemos como se trata de un grupo de editores que tienen un grado alto por su tamaño y que además suelen hacer subcontracting entre ellos. Sin embargo no solo se reducen a su mismo grupo si no que hay numerosas aristas hacia otros nodos diferentes localizados fuera del grupo central. Esto de nuevo podría implicar un largo historial de ediciones en el tiempo por parte de estos usuarios.
- En segundo lugar tenemos al conjunto de nodos localizado a la derecha. Estos nodos se encuentran agrupados entre sí por la cercanía y el peso de sus aristas sin embargo se observa que no necesariamente pertenece al mismo grupo si no que hay varios grupos entremezclados. Sus conexiones se observa van en la mayoría de los casos exclusivamente al nodo que representa a los usuarios anónimos por lo que este conjunto más que denotar un comportamiento específico muestra los editores que hicieron revisiones entre las revisiones realizadas por usuarios anónimos.

### 7.3.2. Artículo Ácido desoxirribonucleico

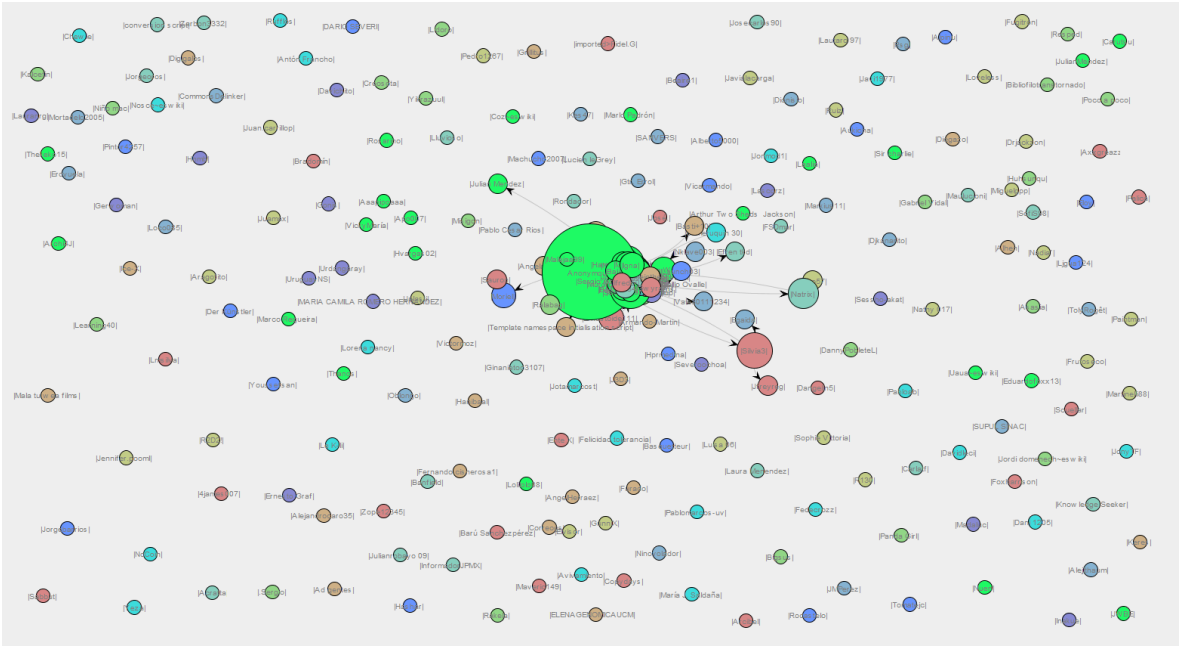


Figura 7.9: Grafo subcontracting del artículo Ácido desoxirribonucleico

El grafo de subcontracting del artículo Ácido Desoxirribonucleico, 7.9, muestra de nuevo, un comportamiento muy similar al subcontracting obtenido en el artículo Tierra. Se observa una gran cantidad de nodos aislados y un grupo central. Estos nodos se encuentran aislados debido al mismo motivo que aquellos en la subsección previa: Solo han realizado ediciones durante una sesión, dando lugar a una secuencia de 1 o más ediciones ininterrumpidas por otro editor.

En el conjunto central, 7.10, otra vez encontramos la misma estructura que en el artículo Tierra. Un grupo central de editores pertenecientes al mismo grupo, en verde (en el artículo Tierra en rojo) y otro compuesto por usuarios de menor grado de diferentes grupos agrupados en el centro del grafo debido a sus interacciones con los usuarios anónimos.

La diferencia en este caso, es el tamaño de los nodos del grupo central verde. Estos tienen menor tamaño en comparación con los del anterior artículo. Sin embargo, este artículo cuenta con 3308 ediciones y 396 mientras que el artículo Tierra con 2654 revisiones y 460 editores. Al medir subcontracting, un mayor número de editores existentes da lugar a un mayor número posible de conexiones y con esto un mayor tamaño en sus nodos más influyentes. Debido a esto, el artículo Ácido Desoxirribonucleico cuenta con un menor número de editores y un mayor número de ediciones que el artículo Tierra, dando lugar así a menos posibles conexiones entre usuarios reduciendo el tamaño de los nodos.

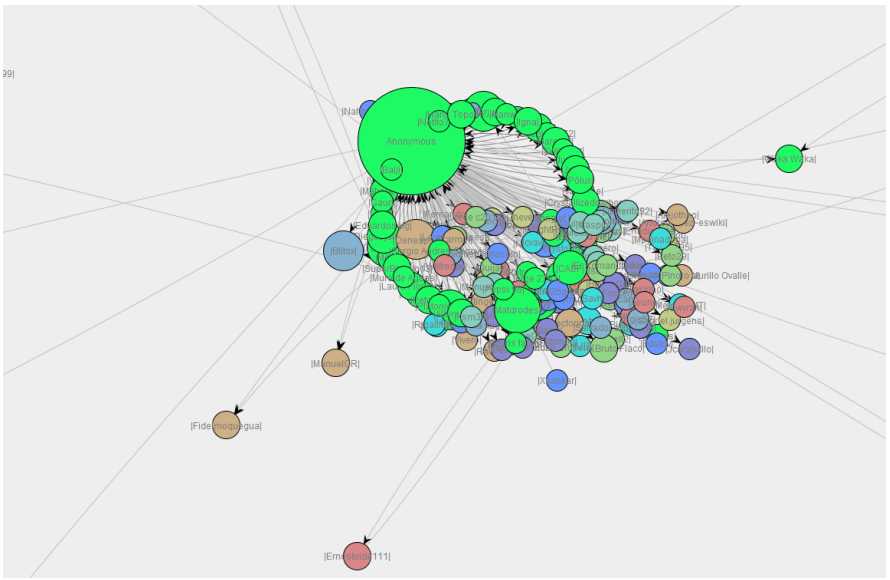



Figura 7.10: Zoom grupo central del grafo Subcontracting del artículo Ácido desoxirribonucleico

# Capítulo 8

# Conclusiones

## 8.1. Conclusiones minería de procesos a nivel artículo

Se ha aplicado minería de procesos a un **corpus** compuesto de 8 *artículos destacados* de la Wikipedia Española desde el punto de vista del artículo. Es decir, conformando un log de eventos donde cada caso es representado por un artículo y cada 'actividad' realizada llamada evento es representada por la intencionalidad semántica tras cada revisión. El resultado ha sido una red de gran complejidad e imposible de analizar para el ojo humano.

Esta red obtenida representa la complejidad del proceso de edición que siguen los artículos. Al tratarse Wikipedia de una comunidad de conocimiento colaborativo abierta, cualquier persona puede convertirse en un editor. Esto intuitivamente se traduce en una gran cantidad de usuarios con variados rangos de conocimiento y habilidad y por ende muchos estilos de edición diferente. Sin embargo, para obtener más información acerca de esta red se descompuso en sub-redes de mayor simplicidad.

Estas sub-redes muestran un panorama similar al anterior, su complejidad es alta aunque con diferencias entre ellas. Mientras que no representan un proceso específico que se siga a al pie de la letra y de ahí su complejidad, muestran comportamientos diferentes.

Una de las redes obtenida muestra intencionalidades propias de usuarios todo-terreno con intenciones combinadas de cierta complejidad desde el comienzo mientras que la otra red muestra intenciones más específicas propias de usuarios con mayor grado de especialización.

En resumen, se puede concluir que (i) no hay un proceso unificado que se siga en la creación y evolución de los artículos en Wikipedia aunque si se ve una variación en las intenciones según evoluciona el artículo y (ii) la especialización o el generalismo de los editores en etapas tempranas de un artículo tiene influencia en la evolución de los mismos como vemos por las diferentes redes obtenidas tras la descomposición, dando lugar a diferentes flujos de trabajo.

## 8.2. Conclusiones minería de procesos a nivel editor

En este caso, la minería de procesos fue aplicada dándole un giro al log de eventos de la sección anterior orientándolo al editor. De esta manera, se agrupan los editores en tres

categorías diferentes en función de su actividad dentro del conjunto ocho artículos utilizado (Baja/Intermedia/Alta actividad) medido por el número de ediciones realizado.

Analizando los procesos seguidos por los usuarios en sus sesiones de edicion, se observan grandes similaritudes con aquellos comportamientos que identifican en la investigación 'Estabilidad turbulenta de roles emergentes' ([5]).

De esta manera, a lo largo de las tres agrupaciones realizadas, se observan cambios importantes en los roles observados en función del número de ediciones.

Entre aquellos usuarios con baja actividad, se observa un poco de todo, aunque por supuesto no hay gente que se pueda identificar como 'All round contributors' pues no han realizado una cantidad suficiente de ediciones para poder determinar esto. En general, esta categoría se compone de intenciones no demasiado complejas.

En aquellos editores con actividad intermedia, se comienzan a observar intenciones más refinadas y aparecen los 'All round contributors'. Sigue habiendo vándalos dentro de esta categoría. Por otro lado, se comienza a ver que los usuarios que encajarían con los roles 'Copy editors', 'Content shapers' y 'Layout shapers' con frecuencia realizan tareas de cualquiera de los 3 roles.

En el caso de los editores con alta actividad, es decir, más de 50 revisiones se observan no sólo todo los roles, si no que se refuerza ese solapamiento de los roles 'Copy editors', 'Content shapers' y 'Layout shapers'.

De esta manera, se ha comprobado la existencia de diferentes roles entre los usuarios de Wikipedia desde el punto de vista de la minería de procesos, reforzando aquellos resultados obtenidos por Daxenberger en 'Estabilidad turbulenta de roles emergentes'. Sin embargo, debido al solapamiento de los 3 roles 'Copy editors', 'Content shapers' y 'Layout shapers' se propone una alteración en su taxonomía de roles añadiendo un nuevo rol denominado 'Article fixers'.

Estos 'Article fixers' se encargan tanto de realizar tareas de arreglo de formato de Wikipedia y texto como de faltas de ortografía o mejoras en sintáxis y han sido observados tanto en aquellos autores con una actividad intermedia (en menor medida) como en aquellos con una alta actividad.

## 8.3. Conclusiones minería social

Dentro de la minería social, se han aplicado los algoritmos de Subcontracting y Handover of work a dos artículos diferentes de la Wikipedia obteniendo en ambos casos resultados muy similares verificando mutuamente los resultados obtenidos.

El algoritmo working together mide el traspaso de trabajo entre editores. Así, en ambos artículos hemos obtenido estructuras de organización muy similares.
Un porcentaje de los editores transpasan trabajo a un número muy reducido de editores,

entre 1 y 2, implicando que sus contribuciones al artículo son realizadas durante un momento específico del tiempo y no han editado en el artículo a lo largo de una temporada. Por otro lado, aquellos editores más influyentes y con mayor número de revisiones muestran muchas conexiones implicando que han trabajo con muchos otros editores dando lugar a ediciones durante un periodo de tiempo extendido.

Por otro lado el Subcontracting nos muestra la 'subcontratación' entre editores, es decir, si un editor edita entre dos ediciones de otro, lo cual para suceder de modo consistente requiere de cierta colaboración o un número muy reducido de usuarios. Los resultados obtenidos son similares de nuevo entre ambos artículos.
La mayoría de usuarios realiza una sola sesión de edición donde realizan una cantidad variada de ediciones sin interrupciones de otros editores. Por otro lado, existe un núcleo de editores de mayor influencia que realizan subcontrataciones entre ellos, mostrando que posiblemente exista una colaboración explícita entre estos usuarios.

Sin embargo, para afianzar estas conclusiones debemos observar los resultados de ambas métricas conjuntamente. Así, vemos que aquellos usuarios aislados en subcontracting, son aquellos que en handover of work cuentan con dos conexiones pues han realizado una sesión de edición y nada más, de manera que su trabajo es continuado por otro editor dando lugar a la conexión en el handover of work.

Siguiendo esta misma línea de razonamiento, se ve por lo tanto que aquellos editores de mayor influencia representan diferente cara de la misma moneda en ambas métricas. En handover of work, estos usuarios forman un conjunto con númerosas conexiones entre ellos y a numerosos nodos mientras que en subcontracting estas conexiones se observan en mayor medida entre ellos. Es decir, mientras que realizan ediciones en general, hay momentos donde se realizan colaboraciones con otros editores de gran influencia dentro del artículo. Es decir existe una colaboración explícita entre los usuarios más influyentes dentro de un artículo.

En resumen, nos encontramos con que hay un gran porcentaje de editores dentro de Wikipedia que realizan una sola sesión de edicion en un artículo para no volver, mientras que hay una minoría de editores asiduos que realizan tareas de edición durante periodos de tiempo grandes tanto casual como de manera organizada puntualmente.

## 8.4. Conclusiones globales

De modo general, se pueden resumir los hallazgos encontrados en:

1. No existe un proceso unificado de edición durante la evolución de un artículo. Sin embargo, se observa como el tipo o la sofisticación de las intenciones puede varíar a lo largo de diferentes etapas del artículo

2. Además, la generalidad o especificidad de los editores en etapas tempranas del artículo muestra diferentes maneras de proceder. Es decir, los editores iniciales tienen influencia en la evolución posterior del artículo.

3. Se observan los diferentes roles de editor de la taxonomía desarrollada en 'Estabilidad turbulenta de roles emergentes ([5]). Además, se propone la adición del rol 'Article

Fixer': editor que realiza tareas de formato del artículo y texto así como mejoras en gramática y sintáxis. Es decir, hacen presentable el artículo sin entrar en la corrección de la información y sin añadir más contenido.

4. La mayoría de editores lo hacen de modo casual, con una sola sesión de ediciones en un artículo al que no vuelven.

5. Por otro lado, existen editores dedicados que editan durante la evolución de los artículos realizando colaboraciones organizadas puntuales con otros editores. Estos editores, además, suelen ser aquellos con la mayor influencia en los artículos (es decir, aquellos con mayor número de revisiones).

# Capítulo 9

# Conclusions

## 9.1. Process mining applied to the article: conclusions

Process mining was applied to a **corpus** composed of 8 *featured articles* from the Spanish Wikipedia. The event log represented each case as each article and each event as the revision. The resulting petri net was too complex to analyze by the human eye.

This complex net shows the complexity of the process followed by the articles. Wikipedia is an open community of shared knowledge. This means that anybody can perform an edit on a specific article anytime with no previous organization with other users. Hence, the number of existing editors in Wikipedia, is incredibly high.

The high complexity and multiple paths in the net show that there is not an specific process followed in Wikipedia's article's edition history. However, in order to obtain any insight from the net, it was decomposed into simpler sub-nets. This sub-net are sill complex yet understandable and show different behaviours.

One of the sub-nets show intentionalities behind each revision akin to 'All round contributors' since the beginning of the net while the other net displays more specific intentions.

With this information, the following can be concluded: (i) There is not a unified process in the edition and evolution of Wikipedia articles. However, the intentions in the revisions seem to evolve along with the article. (ii) The specialization of the editors in the early stages of an article have an influence in the evolution of the article.

## 9.2. Process mining applied to the editor: conclusions

In this approach, the process mining techniques were applied to discover the processes followed by the editors, instead of the article. The editors were grouped based on the quantity of the revisions they made: low, medium and high activity.

The processes followed by the editors in their edit sessions show great similarities with the taxonomy of roles used throughout this document.

In addition to this, in each set of grouped editors different behaviours can be appreciated.

Among the low activity users, those that performed less than five editions, all roles are found with the exception of 'All round-contributors'. However not very complex intentions are observed in this group.

Medium activity editors, with 5 to 50 revisions, show a more refined combination of intentions and the 'all round contributors' start to appear. Also there is still some vandalism in this category. On the other hand, editors with the role of 'Copy editors', 'Content shapers' and 'Layout shapers' are seen to be performing tasks associated with any of those three roles instead of just sticking to one of them.

When it comes to the high activity users, those with more than 50 revisions, all the roles are observed. Again editors performing tasks associated with the role of 'Copy editors', 'Content shapers' and 'Layout shapers' are not sticking to only one role but the three of them.

In conclusion, the editor roles developed in 'Turbulent stability of emergent roles' ([24]) have been clearly seen with the methodology of the process mining, reinforcing its veracity. However, given the observed behaviour of editors performing tasks of layout and content shape modification and copy editing we propose the addition of a new role called 'Article fixers'.

'Article fixers' are the editors in charge of performing tasks related with the formatting of content and article and fixing typos, grammar and syntax. This role is similar to an 'All round contributor' but without performing any task related to the value and depth of the information.

## 9.3. Social mining conclusions

Using social mining techniques, the subcontracting and handover of work algorithms were applied to two different articles of Wikipedia out of our used **corpus**. In both cases the obtained results were very similar.

Handover of work measures the flow of work from one editor to the other. Both articles show very similar organizative structures. A percentage of the editors work only with a very reduced number of editors, one or two maximum. This means that their contributions to the article were probably performed in an specific moment of time and not during a large time span. On the other hand the most influential editors (those with the biggest amount of editions) have a lot of different connections meaning they edited prior to a lot of different users meaning that their relationship with the article extends over time.

Subcontracting establishes a connection between two editors if one of the editors performed an edit between two edits of the other user. In communities with a lot of users such as Wikipedia, consistent connections of subcontracting between 2 editors might mean an explicit collaboration between said users as the probability of editing between edits of another user routinely is very low. Interestingly enough, both articles yielded similar results.

The majority of the editors make only one edit session composed of one or more edits, without interruption from other users. On the other hand, there is a cluster of editors with higher influence, that is performing subcontracting with each other. It shows that there is probably an explicit collaboration between those users.

However, in order to obtain the full picture, both metrics should be interpreted together.

In the handover of work a big percentage of the users were seen to have handed over work to only 2 users and in subcontracting a big percentage of the users were isolated. In both cases the editors portrayed were the same, as it shows two faces of the same coin.

Following this very same line of thought applied to the cluster of influential editors in both metrics is exactly the same. Those that are the most influential in the community handed over work to a lot of users. However, their connections in subcontracting are more limited to the editors that composed that very same cluster. This can be translated into the following: the editors that put time and dedication into this process makes editions as they see it necessary with specific moments of explicit collaboration with other users. However this collaborations seem primarily restricted to prolific editors.

To sump up, there is a huge percentage of editors in Wikipedia that make editions in an specific moment of time only and do not edit in the article again. In opposition to this, there is a minority of habitual users that edit during large time spans sometimes even collaborating explicitly with other editors.

## 9.4. Global conclusions

The findings of this document can be summed up in the following list:

1. There is not a unified editing process during the evolution of an article. However, there is a variation in the intentions observed as the articles evolve.

2. In addition, the skill set of the editors in early stages of the article show different paths. This, means that the skill set of the initial editors have an influence in the evolution of the article.

3. The different roles presented in the taxonomy of roles in 'Turbulent stability of emergent roles' can be observed in the processes followed by the editors. However, the addition of a new role is suggested: 'Article fixer'. This role is for editors focused on the formatting of the content and the article that also fix grammar, syntax and typos without paying attention to the actual content of the article.

4. Most editors are casuals. They make only one edit session and don't edit in the article again.

5. On the other hand, there are some dedicated editors that perform edits consistently throughout large time spans with specific explicit collaborations with other editors. This editors are also the most influential ones regarding the amount of editions.

# Capítulo 10

# Trabajo futuro

Este proyecto ha sido realizado generando un corpus basado en ocho artículos destacados diferentes de la Wikipedia española. Sin embargo, el propio tamaño reducido del corpus limita en gran parte la posibilidad de generalizar los resultados obtenidos. Es debido a esto que como trabajo futuro sería muy interesante poder hacer uso de un corpus de mucho mayor tamaño o hacer uso de una wiki de tamaño reducido como las de Wikia (por ejemplo Wiki Cocktails o Hitchikers Wiki) ya que poder analizar una comunidad entera en su conjunto aporta resultados muchos mas significativos que un fragmento de la misma.

El motivo por el cual esto no se ha realizado es porque cada wiki existente hace uso de una API propia derivada de la encontrada en `https://www.mediawiki.org/wiki/API:Main_page` como por ejemplo `https://cocktails.fandom.com/api.php`. Sin embargo, como bien indican se encuentra en desarrollo y no funcionaba correctamente bajo el script desarrollado para 'Indentificando intenciones semánticas en las revisiones de Wikipedia' ([24], más en: 11). Concretamente, la salida del diff entre revisiones se encontraba vacío.
Cuando la API se encuentre totalmente desarrollada y funcione de modo correcto sería muy interesante poder trabajar con ella.

Por otro lado, en el futuro se podría extender la investigación agrupando a todos los editores bajo los roles de la taxonomía de roles utilizada en el proyecto y presentada en el capítulo de fundamentos teóricos (3). Con todos los editores agrupados se podrían estudiar los procesos seguidos por cada rol para analizar el cambio y evolución de los mismos así como las interacciones entre los diferentes roles.

# Capítulo 11

# Código

Este proyecto cuenta con código de autoría propia así como código realizado por terceras partes.

De este modo, el código desarrollado se encuentra alojado en https://github.com/FRYoussef/TFG_Wiki. Este código de autoría propia cuenta con licencia MIT y se encuentra compuesto por los siguientes scripts:

- corpus_filter.py: Este script contiene 4 filtros posibles de entre casual, low, intermediate y high para filtrar los editores de un historial de revisiones en función de su número de revisiones o para agruparlos como es el caso del filtro casual. También un filtro para eliminar las revisiones realizadas por BOTs. En este caso no requiere más entrada que un parámetro que especifique el tipo de filtro a aplicar pues hace uso automáticamente del corpus y del fichero generado por el siguiente script de la lista editor_count_aggregator.py y su salida es el corpus filtrado.
- editor_count_aggregator.py: Agrega el número de revisiones realizado por cada autor, generando un fichero de texto donde se encuentran los autores y su conteo total de revisiones a través del corpus.
- model_generation.ipynb: Notebook donde se realiza un análisis detallado en busca del mejor modelo predictivo posible para detectar las intenciones inherentes a cada revisión, los mejores modelos determinados son exportados y usados por el siguiente script de la lista: generate_predictions.py
- generate_predictions.py: Hace uso de los 13 modelos generados (uno por cada intención) para predecir las intenciones tras cada revisión del archivo que se pase como entrada y da formato a los resultados modificando el csv inicial del conjunto de historial de revisiones como salida.
- revision_id_extractor.py: Su utilidad es crear un fichero csv que almacene el id de revisión y la intencionalidad. Debido a que en este punto la intención aún no se conoce, añade un 0 por defecto en su lugar. El motivo de esto es que un script de una tercera parte hace uso de un archivo con este formato como entrada.
- wikipedia_dump_downloader.py: Se trata de una modificación de https://phabricator.wikimedia.org/diffusion/PWBC/browse/master/scripts/maintenance/download_dump.

py desarrollada por el autor de este proyecto en conjunto con Youssef El Faqir El Rhazoui. La modificación, consiste en la eliminación de dependencias con la libreria PyWiki además de la implementación de utilidades como selección de idioma de la wiki de descarga, descargar una lista de artículos y la unión de los diferentes fragmentos descargados pues los artículos son divididos en fragmentos para su adecuada descarga.

Así, el código de terceras partes utilizado ha sido:

- wiki_dump_parser.py: Localizable en https://github.com/Grasia/wiki-scripts y desarrollado por Abel Serrano Juste como parte de un conjunto de scripts para obtener y procesar datos de una wiki. El programa en cuestión se trata de un script para obtener información útil y dar formato a los datos descargados por wiki_dump_downloader.py en forma de csv.
- arffToCsv.py es un script simple y sencillo para transformar un archivo arff a formato csv desarrollado por Haloboy777 y localizable en https://github.com/haloboy777/arfftocsv bajo licencia MIT
- Código desarrollado para la investigación 'Identificando intenciones semánticas en las revisiones de Wikipedia' con el objetivo de obtener las diferentes carácterísticas de cada revisión en función de la anterior en una wiki. Ha habido que realizar ligeros cambios como librerias obsoletas o añadir diferentes funcionalidades como la posibilidad de añadir el id de revisión al conjunto de datos o decidir el idioma o tipo de wiki. Se encuentra alojado en https://github.com/diyiy/Wiki_Semantic_Intention y se ha realizado un fork con los cambios en https://github.com/ignacioGarsami/Wiki_Semantic_Intention.

# Bibliografía


[1] Wil M. P. Aalst. *Decomposing Petri Nets for Process Mining: A Generic Approach*. Vol. 31. Ene. de 2012. DOI: 10.1007/s10619-013-7127-5.

[2] Wil M. P. Van Der Aalst, Hajo A. Reijers y Minseok Song. "Discovering Social Networks from Event Logs". En: *Computer Supported Cooperative Work (CSCW)* 14.6 (2005), 549–593. DOI: 10.1007/s10606-005-9005-9.

[3] Wil M. P. Van Der Aalst y Minseok Song. "Mining Social Networks: Uncovering Interaction Patterns in Business Processes". En: *Lecture Notes in Computer Science Business Process Management* (2004), 244–260. DOI: 10.1007/978-3-540-25970-1_16.

[4] Wil van der Aalst y col. *Process Mining Manifesto*. 2011. URL: https://doi.org/10.1007/978-3-642-28108-2_19.

[5] Ofer Arazy y col. "Turbulent Stability of Emergent Roles: The Dualistic Nature of Self-Organizing Knowledge Co-Production". En: *Information Systems Research* 27 (ene. de 2017), págs. 792-812. DOI: 10.1287/isre.2016.0647.

[6] Hicheur Awatef y col. "Process Mining in the Education Domain". En: feb. de 2015.

[7] Tim Bray y col. *Extensible markup language (XML) 1.0*. 2000.

[8] Andriy Burkov. *The Hundred-Page Machine Learning Book*. 1.ª ed. Kindle Direct Publishing, 2019. ISBN: 9781790485000.

[9] Soumen Chakrabarti y col. *Data Mining Curriculum: A Proposal*. 2006. URL: https://www.kdd.org/exploration_files/CURMay06.pdf.

[10] B. F. van Dongen y col. *The ProM Framework: A New Era in Process Mining Tool Support*. 2005. URL: https://link.springer.com/chapter/10.1007/11494744_25.

[11] Haibo He y col. "ADASYN: Adaptive synthetic sampling approach for imbalanced learning". En: *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)* (2008). DOI: 10.1109/ijcnn.2008.4633969.

[12] Andrew Bruce Peter C. B. *Practical statistics for data scientists : 50 essential concepts*. 1st. O'Reilly Media, Inc., 2017.

[13] James L. Peterson. "Petri Nets". En: *ACM Computing Surveys* 9.3 (1977), 223–252. DOI: 10.1145/356698.356702.

[14] *ProM 6.8*. 2018. URL: http://www.promtools.org/doku.php?id=prom68.

[15] Michal Rosik. *3 Industries and Companies Doing Process Mining Right*. URL: https://www.minit.io/blog/3-industries-and-companies-doing-process-mining-right.

[16] Pnina Shachaf y Noriko Hara. "Beyond vandalism: Wikipedia trolls". En: *Journal of Information Science* 36.3 (2010), 357–370. DOI: 10.1177/0165551510365390.

[17] *sklearn.dummy.DummyClassifier*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.dummy.DummyClassifier.html.

[18] Marina Sokolova y Guy Lapalme. "A systematic analysis of performance measures for classification tasks". En: *Information Processing Management* 45.4 (2009), 427–437. DOI: 10.1016/j.ipm.2009.03.002.

[19] Vilaythong Southavilay, Kalina Yacef y Rafael A. Calvo. "Analysis of Collaborative Writing Processes Using Hidden Markov Models and Semantic Heuristics". En: *2010 IEEE International Conference on Data Mining Workshops* (2010). DOI: 10.1109/icdmw.2010.118.

[20] Jake VanderPlas. *Python Data Science Handbook: Essential Tools for Working with Data*. 1st. O'Reilly Media, Inc., 2016. ISBN: 1491912057, 9781491912058.

[21] Christian Wagner y Pattarawan Prasarnphanich. "Innovating Collaborative Content Creation: The Role of Altruism and Wiki Technology". En: *2007 40th Annual Hawaii International Conference on System Sciences (HICSS07)* (2007). DOI: 10.1109/hicss.2007.277.

[22] A Weijters, Wil M. P. Aalst y Alves A K Medeiros. *Process Mining with the Heuristics Miner-algorithm*. Vol. 166. Ene. de 2006.

[23] *XES*. 2018. URL: http://xes-standard.org/.

[24] Diyi Yang y col. "Identifying Semantic Edit Intentions from Revisions in Wikipedia". En: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (2017). DOI: 10.18653/v1/d17-1213.