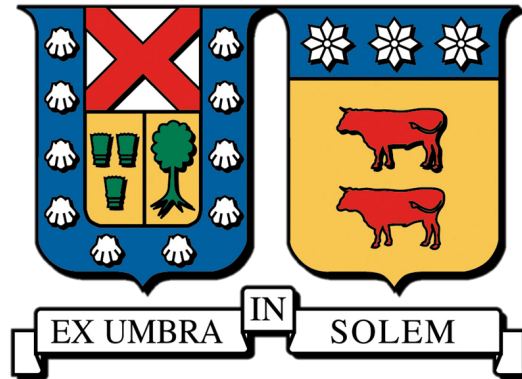


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

DEPARTAMENTO DE INGENIERÍA INFORMÁTICA



**CARACTERIZACIÓN DE LA INFORMACIÓN PRESERVADA
EN UNA RED DE PALABRAS EN FUNCIÓN DE SU CONSTRUCCIÓN**

Ignacio Andrés Quintana Jara

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO/A CIVIL INFORMÁTICO/A

Guía: Andrés Moreira Wenzel
Co-referentes: Marcelo Mendoza Rocha

DICIEMBRE-2024

Resumen

En este trabajo estudiamos cómo se comparan las características de distintas redes de texto al utilizar dos métodos distintos de construcción. Construimos redes mediante el método simple de coocurrencia, y en modo probabilista utilizando las probabilidades de conexión aprendidas por el modelo *word2vec* para generar una versión aleatoria. Posteriormente además añadimos pesos a ambas redes. Encontramos que ambos tipos de redes presentan características de redes complejas como el efecto mundo pequeño con distancias promedio que escalan con $\frac{\log n}{\log k}$ y en el rango 2.45–2.98. También ambos métodos de construcción presentan altos coeficientes de *clustering* en el caso de redes sin pesos. Sin embargo, observamos que ninguno de los métodos de construcción producen redes con distribución de grados libre de escala. Encontramos que en ambos tipos de redes los nodos se asocian de forma disortativa cuando no se consideran pesos, pero tal patrón desaparece en redes de coocurrencia cuando se añaden pesos. Tal fenómeno ocurre en menor medida en el caso de las redes construidas mediante *word2vec* pues el mecanismo en que se asignan los pesos unas pocas conexiones entre nodos de alto grado no sesga el gráfico de grado promedio de vecinos. En ese sentido, asignar pesos mediante este método en vez de frecuencias podría ayudar a resaltar el tipo de conexiones que se forma entre nodos.

Por otro lado el resultado de ejecutar el algoritmo de *Louvain* dio como resultado que en el caso de las redes construidas por medio del modelo *word2vec* se observa mayor modularidad. Lo anterior se debe a que el modelo aprende nuevas relaciones entre palabras que no se dan por simple coocurrencia. Estas relaciones de similaridad semántica producen nuevas conexiones entre sinónimos y antónimos. A pesar de ello, debido al enfoque estadístico del modelo, algunas relaciones de alta similaridad se pierden por no aparecer un número significativo de veces. Creemos que las nuevas relaciones que se crean en el caso de *word2vec* tienen como consecuencia que las comunidades no reflejen tan bien el flujo del texto como en el caso de coocurrencia.

También evaluamos qué tan similares son las comunidades entre ambas construcciones utilizando el índice de Jaccard y encontramos que la mayor parte de estas eran poco similares con unas pocas con valores medianos de similaridad. Fue posible observar que en aquellas donde existía un grado de similaridad la intersección de las redes mostraba palabras semánticamente similares. Finalmente obtuvimos representaciones vectoriales de las redes utilizando *node2vec* y utilizando *k-means*

comparamos la similaridad de los *clusters* con las comunidades reales y contra las comunidades de una versión aleatorizada de la red. Ninguna de las dos redes es lo suficientemente informativa en términos de poder recuperar la estructura de comunidad de Ambos enfoques producen comunidades significativamente distintas de sus versiones aleatorizadas.

Finalmente, creemos que ambos enfoques de construcción son complementarios entre si. Mientras las comunidades generadas por coocurrencia parecen reflejar más fielmente la estructura narrativa del texto, las comunidades del modelo *word2vec* parecen reflejar, además, ciertas relaciones semánticas entre palabras distantes en el texto.

Abstract

In this work, we study how the characteristics of different text networks compare when using two different construction methods. We built networks using the co-occurrence method and in a probabilistic manner using the connection probabilities learned by the *word2vec* model to generate a random version. Subsequently, we add weights to both networks. We find that both types of networks exhibit complex network characteristics, such as the small-world effect, with average distances scaling as $\frac{\log n}{\log k}$ and in the range of 2.45 – 2.98. Both construction methods also show high clustering coefficients for the unweighted networks. However, we observe that neither construction method produces networks with a scale-free degree distribution. In both types of networks, nodes are associatively dissortative when weights are not considered, but this pattern disappears in co-occurrence networks when weights are added. This phenomenon occurs to a lesser extent in *word2vec* networks, as the weighting mechanism assigns the few weights between high-degree nodes without skewing the average degree of neighbors plot. In this sense, assigning weights using this method instead of frequencies could help highlight the type of connections formed between nodes.

Additionally, the result of running the *Louvain* algorithm shows that networks constructed via the *word2vec* model exhibit higher modularity. This is because the model learns new relationships between words that do not occur through simple co-occurrence. These semantic similarity relationships produce new connections between synonyms and antonyms. However, due to the statistical nature of the model, some highly similar relationships are lost due to their infrequent occurrences. We believe that the new relationships created by *word2vec* cause the communities to reflect the text flow less accurately than in the co-occurrence case.

We also assessed the similarity of the communities between both constructions using the Jaccard index and found that most of them were dissimilar, with only a few showing medium similarity values. It was possible to observe that in the communities with similarity, the intersection of networks showed semantically similar words. Finally, we obtained vector representations of the networks using *node2vec*, and by applying *k-means*, we compared the similarity of the clusters with the actual communities and those from a randomized version of the network. Neither of the two networks was sufficiently informative in terms of recovering the community structure. Both approaches produced communities significantly different from their randomized versions.

In conclusion, we believe that both construction approaches are complementary. While the communities generated by co-occurrence more faithfully reflect the narrative structure of the text, the communities from the *word2vec* model also capture certain semantic relationships between words that are distant within the text.

Agradecimientos

Este trabajo esta dedicado a mis queridos padres Juan y Elizabeth, gracias por haber estado ahí siempre, gracias por la paciencia y los sacrificios que han tenido que hacer para llegar a este momento. Sin su incondicional apoyo y consejos no podría estar aquí. También quiero agradecer a mis amigos puesto que sin nuestro sacrificio en conjunto durante todos estos años no podría haber terminado la carrera.

On that day, I bid you remember this...
That no matter how far your journey may
take you, you stand where you stand by
virtue of the road you walked to get there.
For in times of hardship, when you fear
you cannot go on... The joy you have
known, the pain you have felt, the prayers
you have whispered and answered—they
shall ever be your strength and your
comfort. This I hope—I believe, here at
memory's end.

Crystal Exarch

Tabla de Contenidos

1	Introducción	1
2	Objetivos	3
2.1	Objetivos específicos	3
3	Estado del Arte	4
3.1	Redes complejas para representación de texto	4
3.2	Word embeddings para representación de texto	5
3.3	Reconciliando embeddings y redes complejas	6
4	Redes complejas	7
4.1	Fenómenos que se observan en redes complejas	8
4.1.1	El efecto mundo pequeño	8
4.1.2	Clustering	8
4.1.3	Redes libre de escala	9
4.1.4	Asortatividad	10
4.2	Métodos de redes	11
4.2.1	El método de Louvain	11
4.2.2	<i>k-core</i>	12
5	Aprendizaje automático	14
5.1	Word2vec	14
5.1.1	Entrada del modelo	15
5.1.2	Muestreo negativo	15
5.1.3	Hiperparámetros del modelo	16
5.1.4	Arquitectura de la red neuronal	16
5.1.5	Función objetivo estándar	18
5.1.6	Función objetivo optimizada con muestreo negativo	19
5.2	Node2vec	21

5.2.1	Entrada del modelo	22
5.2.2	Estrategias para la obtención de vecindades	22
5.2.3	Paseos aleatorios	22
5.2.4	Hiperparámetros del modelo	23
5.2.5	El algoritmo node2vec	24
5.3	Conectando word2vec y redes complejas	25
6	Diseño experimental	26
6.1	Textos y preprocesamiento	26
6.1.1	Textos a utilizar	26
6.1.2	Preprocesamiento del texto	26
6.2	Construcción de las redes	27
6.2.1	Construcción de la red de coocurrencia	27
6.2.2	Construcción de la red basada en el modelo word2vec	27
6.3	Estudio de las redes	28
6.3.1	Variabilidad de la red word2vec	28
6.3.2	Comparando propiedades generales entre ambas redes	29
6.3.3	Comparando estructura de comunidad de las redes	30
6.3.4	Similaridad de arcos	31
6.3.5	Comparando informatividad de ambas redes	31
7	Resultados	32
7.1	Resultados del preprocesamiento	32
7.2	Resultados de variabilidad de redes generadas por medio de word2vec	32
7.2.1	Matriz de probabilidad fija	32
7.2.2	Matriz de probabilidad provenientes de modelos distintos	33
7.3	Resultados de cantidad de nodos y arcos, efecto mundo pequeño y clustering promedio	34
7.3.1	Coeficiente de clustering en función del grado	35
7.4	Distribuciones de grado	39
7.5	Asortatividad	41
7.5.1	Redes sin pesos	41
7.5.2	Redes con pesos	43
7.6	Resultados de estructura de comunidades	45
7.6.1	Redes sin pesos	45
7.6.2	Redes con pesos	51
7.7	Distribuciones de similaridad	54
7.8	Node2vec	59

8 Conclusiones y Recomendaciones	63
Referencias	67
A Tablas de asociaciones comunidades <i>k-core</i> sin pesos	I
A.1 Alicia en el país de las maravillas	I
A.2 La Ilíada	XI
A.3 Pinocho	XVII
A.4 Guerra y paz	XXVII
B Tablas de asociaciones comunidades <i>k-core</i> con pesos	XXXII
B.0.1 Alicia en el país de las maravillas	XXXII
B.0.2 La Ilíada	XLII
B.0.3 Pinocho	L
B.0.4 Guerra y paz	LVII
C Tablas de asociaciones Node2vec	LXVI
C.1 Alicia en el país de las maravillas	LXVI
C.1.1 Red de co-ocurrencia	LXVI
C.1.2 Word2vec	LXXXI

Lista de Tablas

7.1	Cantidad de palabras luego de aplicar preprocesamiento	32
7.2	Tabla de test de Kolmogorov-Smirnov generando matrices de adyacencia a partir de un mismo modelo entrenado	33
7.3	Tabla de test de Kolmogorov-Smirnov generando matrices de adyacencia a partir de un mismo modelo entrenado	33
7.4	Tabla de cantidad de nodos no aislados n , arcos m , efecto mundo pequeño l , <i>clustering</i> promedio sin pesos C y <i>clustering</i> promedio con pesos $\tilde{C}$	34
7.5	Tabla con gráficos de <i>clustering</i> promedio sin peso en función del grado para los distintos textos	36
7.6	Tabla con gráficos de <i>clustering</i> promedio con peso en función del grado para los distintos textos	38
7.7	Tabla con gráficos de las distribuciones de grados para los distintos textos y tipos de red	40
7.8	Tabla con gráficos de las distribuciones de asortatividad sin pesos para los distintos textos y tipos de red	42
7.9	Tabla con gráficos de las distribuciones de asortatividad con pesos para los distintos textos y tipos de red	44
7.10	Tabla con el número de comunidades, modularidad, modularidad de redes aleatorizadas y z_{score} para los textos considerando redes sin pesos	45
7.11	Tabla con el número de comunidades y modularidad para los textos considerando redes con pesos	51
7.12	Comparación de similaridad de arcos de Alicia en el país de las Maravillas	55
7.13	Comparación de similaridad de arcos de La Ilíada	56
7.14	Comparación de similaridad de arcos de Pinocho	57
7.15	Comparación de similaridad de arcos de Guerra y Paz	58
7.16	Comparación de distribuciones de valores de Jaccard contra versiones aleatorizadas de las redes de Alicia en el país de las maravillas	59

7.17	Comparación de distribuciones de valores de Jaccard contra versiones aleatorizadas de las redes de La Ilíada	60
7.18	Comparación de distribuciones de valores de Jaccard contra versiones aleatorizadas de las redes de Pinocho	61
7.19	Comparación de distribuciones de valores de Jaccard contra versiones aleatorizadas de las redes de Guerra y paz	62
A.1	Tabla n° 1 de asociaciones de comunidades k-core para Alicia en el país de las maravillas	I
A.2	Tabla n° 2 de asociaciones de comunidades k-core para Alicia en el país de las maravillas	II
A.3	Tabla n° 3 de asociaciones de comunidades k-core para Alicia en el país de las maravillas	II
A.4	Tabla n° 4 de asociaciones de comunidades k-core para Alicia en el país de las maravillas	III
A.5	Tabla n° 5 de asociaciones de comunidades k-core para Alicia en el país de las maravillas	III
A.6	Tabla n° 6 de asociaciones de comunidades k-core para Alicia en el país de las maravillas	IV
A.7	Tabla n° 7 de asociaciones de comunidades k-core para Alicia en el país de las maravillas	IV
A.8	Tabla n° 8 de asociaciones de comunidades k-core para Alicia en el país de las maravillas	V
A.9	Tabla n° 9 de asociaciones de comunidades k-core para Alicia en el país de las maravillas	VI
A.10	Tabla n° 10 de asociaciones de comunidades k-core para Alicia en el país de las maravillas	VII
A.11	Tabla n° 11 de asociaciones de comunidades k-core para Alicia en el país de las maravillas	VIII
A.12	Tabla n° 12 de asociaciones de comunidades k-core para Alicia en el país de las maravillas	VIII
A.13	Tabla n° 13 de asociaciones de comunidades k-core para Alicia en el país de las maravillas	IX
A.14	Tabla n° 14 de asociaciones de comunidades k-core para Alicia en el país de las maravillas	X
A.15	Tabla n° 1 de asociaciones de comunidades k-core para La Ilíada	XI
A.16	Tabla n° 2 de asociaciones de comunidades k-core para La Ilíada	XI
A.17	Tabla n° 3 de asociaciones de comunidades k-core para La Ilíada	XII
A.18	Tabla n° 4 de asociaciones de comunidades k-core para La Ilíada	XII
A.19	Tabla n° 5 de asociaciones de comunidades k-core para La Ilíada	XIII
A.20	Tabla n° 6 de asociaciones de comunidades k-core para La Ilíada	XIII
A.21	Tabla n° 7 de asociaciones de comunidades k-core para La Ilíada	XIV
A.22	Tabla n° 8 de asociaciones de comunidades k-core para La Ilíada	XV
A.23	Tabla n° 9 de asociaciones de comunidades k-core para La Ilíada	XVI
A.24	Tabla n° 1 de asociaciones de comunidades k-core para Pinocho	XVII
A.25	Tabla n° 2 de asociaciones de comunidades k-core para Pinocho	XVIII

A.26	Tabla n° 3 de asociaciones de comunidades k-core para Pinocho	XIX
A.27	Tabla n° 4 de asociaciones de comunidades k-core para Pinocho	XIX
A.28	Tabla n° 5 de asociaciones de comunidades k-core para Pinocho	XX
A.29	Tabla n° 6 de asociaciones de comunidades k-core para Pinocho	XXI
A.30	Tabla n° 7 de asociaciones de comunidades k-core para Pinocho	XXI
A.31	Tabla n° 8 de asociaciones de comunidades k-core para Pinocho	XXII
A.32	Tabla n° 9 de asociaciones de comunidades k-core para Pinocho	XXIII
A.33	Tabla n° 10 de asociaciones de comunidades k-core para Pinocho	XXIV
A.34	Tabla n° 11 de asociaciones de comunidades k-core para Pinocho	XXV
A.35	Tabla n° 12 de asociaciones de comunidades k-core para Pinocho	XXVI
A.36	Tabla n° 1 de asociaciones de comunidades k-core para Guerra y paz	XXVII
A.37	Tabla n° 2 de asociaciones de comunidades k-core para Guerra y paz	XXVIII
A.38	Tabla n° 3 de asociaciones de comunidades k-core para Guerra y paz	XXVIII
A.39	Tabla n° 4 de asociaciones de comunidades k-core para Guerra y paz	XXIX
A.40	Tabla n° 5 de asociaciones de comunidades k-core para Guerra y paz	XXIX
A.41	Tabla n° 6 de asociaciones de comunidades k-core para Guerra y paz	XXX
A.42	Tabla n° 7 de asociaciones de comunidades k-core para Guerra y paz	XXXI
B.1	Tabla n° 1 de asociaciones de comunidades k-core con pesos para Alicia	XXXII
B.2	Tabla n° 2 de asociaciones de comunidades k-core con pesos para Alicia	XXXIII
B.3	Tabla n° 3 de asociaciones de comunidades k-core con pesos para Alicia	XXXIV
B.4	Tabla n° 4 de asociaciones de comunidades k-core con pesos para Alicia	XXXV
B.5	Tabla n° 5 de asociaciones de comunidades k-core con pesos para Alicia	XXXVI
B.6	Tabla n° 6 de asociaciones de comunidades k-core con pesos para Alicia	XXXVII
B.7	Tabla n° 7 de asociaciones de comunidades k-core con pesos para Alicia	XXXVIII
B.8	Tabla n° 8 de asociaciones de comunidades k-core con pesos para Alicia	XXXIX
B.9	Tabla n° 9 de asociaciones de comunidades k-core con pesos para Alicia	XL
B.10	Tabla n° 10 de asociaciones de comunidades k-core con pesos para Alicia	XL
B.11	Tabla n° 11 de asociaciones de comunidades k-core con pesos para Alicia	XLI
B.12	Tabla n° 12 de asociaciones de comunidades k-core con pesos para Alicia	XLI
B.13	Tabla n° 1 de asociaciones de comunidades k-core con pesos para La Ilíada	XLII
B.14	Tabla n° 2 de asociaciones de comunidades k-core con pesos para La Ilíada	XLIII
B.15	Tabla n° 3 de asociaciones de comunidades k-core con pesos para La Ilíada	XLIII
B.16	Tabla n° 4 de asociaciones de comunidades k-core con pesos para La Ilíada	XLIV
B.17	Tabla n° 5 de asociaciones de comunidades k-core con pesos para La Ilíada	XLV
B.18	Tabla n° 6 de asociaciones de comunidades k-core con pesos para La Ilíada	XLVI
B.19	Tabla n° 7 de asociaciones de comunidades k-core con pesos para La Ilíada	XLVII

B.20	Tabla n° 8 de asociaciones de comunidades k-core con pesos para La Ilíada	XLVIII
B.21	Tabla n° 9 de asociaciones de comunidades k-core con pesos para La Ilíada	XLVIII
B.22	Tabla n° 10 de asociaciones de comunidades k-core con pesos para La Ilíada	XLIX
B.23	Tabla n° 1 de asociaciones de comunidades k-core con pesos para Pinocho	L
B.24	Tabla n° 2 de asociaciones de comunidades k-core con pesos para Pinocho	L
B.25	Tabla n° 3 de asociaciones de comunidades k-core con pesos para Pinocho	LI
B.26	Tabla n° 4 de asociaciones de comunidades k-core con pesos para Pinocho	LII
B.27	Tabla n° 5 de asociaciones de comunidades k-core con pesos para Pinocho	LIII
B.28	Tabla n° 6 de asociaciones de comunidades k-core con pesos para Pinocho	LIII
B.29	Tabla n° 7 de asociaciones de comunidades k-core con pesos para Pinocho	LIV
B.30	Tabla n° 8 de asociaciones de comunidades k-core con pesos para Pinocho	LIV
B.31	Tabla n° 9 de asociaciones de comunidades k-core con pesos para Pinocho	LV
B.32	Tabla n° 10 de asociaciones de comunidades k-core con pesos para Pinocho	LVI
B.33	Tabla n° 1 de asociaciones de comunidades k-core con pesos para Guerra y paz . . .	LVII
B.34	Tabla n° 2 de asociaciones de comunidades k-core con pesos para Guerra y paz . . .	LVIII
B.35	Tabla n° 3 de asociaciones de comunidades k-core con pesos para Guerra y paz . . .	LIX
B.36	Tabla n° 4 de asociaciones de comunidades k-core con pesos para Guerra y paz . . .	LX
B.37	Tabla n° 5 de asociaciones de comunidades k-core con pesos para Guerra y paz . . .	LXI
B.38	Tabla n° 6 de asociaciones de comunidades k-core con pesos para Guerra y paz . . .	LXI
B.39	Tabla n° 7 de asociaciones de comunidades k-core con pesos para Guerra y paz . . .	LXII
B.40	Tabla n° 8 de asociaciones de comunidades k-core con pesos para Guerra y paz . . .	LXII
B.41	Tabla n° 9 de asociaciones de comunidades k-core con pesos para Guerra y paz . . .	LXIII
B.42	Tabla n° 10 de asociaciones de comunidades k-core con pesos para Guerra y paz . .	LXIII
B.43	Tabla n° 11 de asociaciones de comunidades k-core con pesos para Guerra y paz . .	LXIV
B.44	Tabla n° 12 de asociaciones de comunidades k-core con pesos para Guerra y paz . .	LXV
C.1	Tabla n° 1 de asociaciones entre comunidad de la red de co-ocurrencia y cluster construido en base a la red co-ocurrencia para Alicia en el país de las maravillas . .	LXVII
C.2	Tabla n° 2 de asociaciones entre comunidad de la red de co-ocurrencia y cluster construido en base a la red co-ocurrencia para Alicia en el país de las maravillas . .	LXVIII
C.3	Tabla n° 3 de asociaciones entre comunidad de la red de co-ocurrencia y cluster construido en base a la red co-ocurrencia para Alicia en el país de las maravillas . .	LXIX
C.4	Tabla n° 4 de asociaciones entre comunidad de la red de co-ocurrencia y cluster construido en base a la red co-ocurrencia para Alicia en el país de las maravillas . .	LXX
C.5	Tabla n° 5 de asociaciones entre comunidad de la red de co-ocurrencia y cluster construido en base a la red co-ocurrencia para Alicia en el país de las maravillas . .	LXXI

C.6	Tabla n° 6 de asociaciones entre comunidad de la red de co-ocurrencia y cluster construido en base a la red co-ocurrencia para Alicia en el país de las maravillas . .	LXXII
C.7	Tabla n° 7 de asociaciones entre comunidad de la red de co-ocurrencia y cluster construido en base a la red co-ocurrencia para Alicia en el país de las maravillas . .	LXXIII
C.8	Tabla n° 8 de asociaciones entre comunidad de la red de co-ocurrencia y cluster construido en base a la red co-ocurrencia para Alicia en el país de las maravillas . .	LXXIV
C.9	Tabla n° 9 de asociaciones entre comunidad de la red de co-ocurrencia y cluster construido en base a la red co-ocurrencia para Alicia en el país de las maravillas . .	LXXV
C.10	Tabla n° 10 de asociaciones entre comunidad de la red de co-ocurrencia y cluster construido en base a la red co-ocurrencia para Alicia en el país de las maravillas . .	LXXVI
C.11	Tabla n° 11 de asociaciones entre comunidad de la red de co-ocurrencia y cluster construido en base a la red co-ocurrencia para Alicia en el país de las maravillas . .	LXXVII
C.12	Tabla n° 12 de asociaciones entre comunidad de la red de co-ocurrencia y cluster construido en base a la red co-ocurrencia para Alicia en el país de las maravillas . .	LXXVIII
C.13	Tabla n° 13 de asociaciones entre comunidad de la red de co-ocurrencia y cluster construido en base a la red co-ocurrencia para Alicia en el país de las maravillas . .	LXXIX
C.14	Tabla n° 14 de asociaciones entre comunidad de la red de co-ocurrencia y cluster construido en base a la red co-ocurrencia para Alicia en el país de las maravillas . .	LXXX
C.15	Tabla n° 1 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas	LXXXI
C.16	Tabla n° 2 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas	LXXXII
C.17	Tabla n° 3 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas	LXXXIII
C.18	Tabla n° 4 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas	LXXXIII
C.19	Tabla n° 5 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas	LXXXIV
C.20	Tabla n° 6 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas	LXXXV
C.21	Tabla n° 7 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas	LXXXVI
C.22	Tabla n° 8 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas	LXXXVII
C.23	Tabla n° 9 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas	LXXXVIII

C.24	Tabla n° 10 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas	LXXXIX
C.25	Tabla n° 11 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas	XC
C.26	Tabla n° 12 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas	XCI
C.27	Tabla n° 13 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas	XCII
C.28	Tabla n° 14 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas	XCIII
C.29	Tabla n° 15 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas	XCIV
C.30	Tabla n° 16 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas	XCV
C.31	Tabla n° 17 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas	XCVI
C.32	Tabla n° 18 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas	XCVII
C.33	Tabla n° 19 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas	XCVIII
C.34	Tabla n° 20 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas	XCIX

Lista de Figuras

1.1	Árbol del problema. Fuente: Elaboración propia	2
3.1	Relacion entre el <i>word embedding</i> generado por el modelo <i>Word2vec</i> , redes complejas y espacio hiperbólico.	6
5.1	Diagrama del primer paso de la arquitectura <i>skip-gram word2vec</i>	17
5.2	Diagrama del segundo paso de la arquitectura <i>skip-gram word2vec</i>	17
5.3	Grafo de ejemplo para el algoritmo <i>node2vec</i>	21
7.1	Distribución de valores de Jaccard para emparejamientos de <i>k-core</i> de comunidades de redes sin pesos	47
7.2	Distribución de valores de Jaccard para emparejamientos de <i>k-core</i> de comunidades de redes con pesos	52

Capítulo 1

Introducción

El lenguaje es denominado “una de las maravillas del mundo natural” [1]. Es por ello, y debido además a la creciente cantidad de texto disponible en áreas relacionadas a las ciencias y las humanidades, que ha habido un interés resurgido por nuevas formas de comprender y analizar el lenguaje. Desde la perspectiva de las redes complejas el lenguaje puede ser concebido como un sistema complejo de unidades lingüísticas y sus relaciones. Sin embargo, los supuestos realizados al momento de construir un modelo basado en redes afectan el análisis que puede ser realizado posteriormente sobre este [2]. Ya que ningún modelo es capaz de representar adecuadamente las distintas facetas del lenguaje humano y siguiendo el enfoque reduccionista de la ciencia moderna es que distintas representaciones son utilizadas para segmentar un lenguaje en distintos subsistemas, cada uno tratando de capturar algún aspecto particular del lenguaje [3]. Un sistema en redes complejas puede ser modelado con el modelo de red que definimos a continuación.

Definición 1. *Modelo básico de red.* El modelo básico de red es un grafo no dirigido $G = (V, E)$ con V el conjunto de vértices y E el conjunto de aristas.

Definido el modelo la red puede construirse, por ejemplo, mediante una matriz de coocurrencia. Una palabra w_i coocurre con una palabra w_j en un texto t si existe una aparición de w_i con una distancia menor a un tamaño de ventana d a w_j en el texto t . Si las palabras coocurren, entonces $A_{i,j} = 1$ y un arco es creado entre los nodos representando a w_i y w_j . Existen métodos más sofisticados provenientes del área de redes que tratan de capturar la estructura mesoscópica del texto [4].

Otro enfoque para construir una representación de texto proviene del área de procesamiento de lenguaje natural. Un *word embedding* es una proyección de las palabras de un texto en un determinado espacio tal que a cada palabra es asignado un único vector. La idea es que palabras similares semánticamente y/o sintácticamente presentan vectores cercanos en el espacio proyectado. Diferentes enfoques basados en redes neuronales y reducción de dimensionalidad han sido utilizados

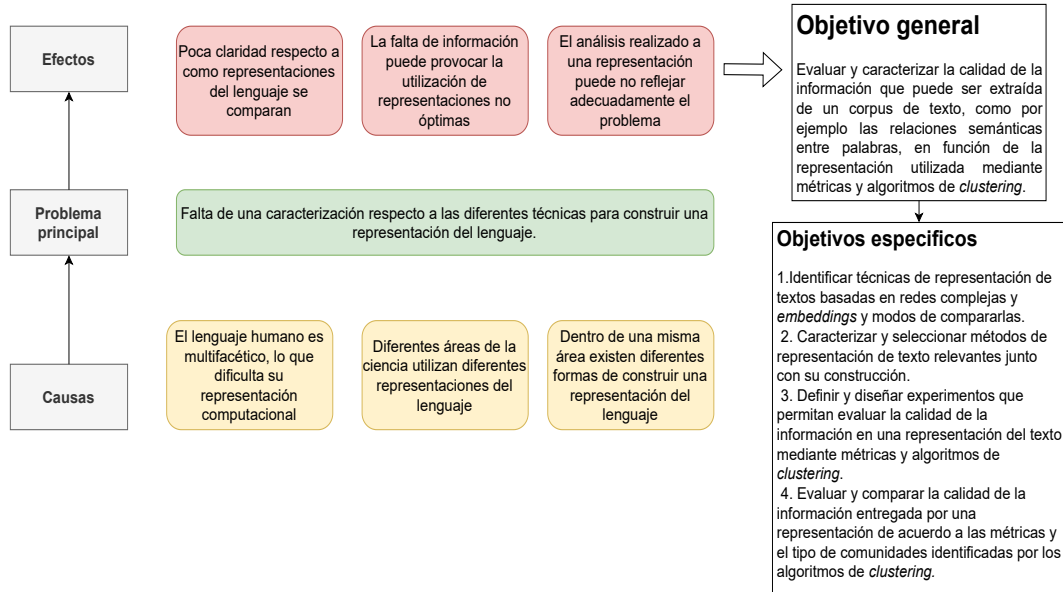


Figura 1.1: Árbol del problema. Fuente: Elaboración propia

para construir *embeddings* [5] [6].

Idealmente nos gustaría conocer cómo las distintas características del texto son preservadas y qué tipo información se puede extraer al usar una representación u otra, sin embargo, las diferencias irreconciliables entre una representación basada en redes y una basada en *embeddings* resultan en una difícil comparación directa entre ambos enfoques. De lo mencionado resulta evidente que existen diversos métodos para construir una representación del lenguaje, sin embargo, al mejor de nuestro conocimiento no existe un estudio acerca de cómo la variación de la representación puede afectar la información que puede ser extraída de esta y cómo diferentes enfoques se comparan entre si. Se busca evaluar las distintas representaciones bajo un corpus de textos usando métricas y algoritmos de detección de comunidades y *clustering* para identificar las características de las representaciones construidas. Esta información puede resultar útil para investigadores en el área de análisis de textos para adecuar la representación utilizada al problema siendo abordado.

El árbol del problema se presenta en la Figura 1.1.

Capítulo 2

Objetivos

El objetivo general de este trabajo consiste en evaluar y caracterizar cómo el método de construcción de una red de palabras afecta las características que poseen y la información que puede ser extraída de estas. Para este propósito utilizamos diferentes métricas y técnicas provenientes de las áreas de ciencia de redes y aprendizaje automático. Particularmente, construimos redes utilizando el método de coocurrencia y mediante probabilidad de conexión según las representaciones vectoriales aprendidas por el modelo *word2vec* sobre una serie de textos. Comparamos las redes generadas por ambos métodos de acuerdo a su distribución de grados, grado promedio de los nodos vecinos, comunidades generadas, etc. Posteriormente utilizamos el algoritmo *node2vec* para transformar las redes a representaciones vectoriales y evaluamos la similaridad de los *clusters* con las comunidades de su red respectiva.

2.1 Objetivos específicos

1. Identificar técnicas de representación de textos basadas en redes complejas y representaciones vectoriales o *embeddings*, y modos de compararlas.
2. Caracterizar y seleccionar métodos de representación de texto relevantes junto con su construcción.
3. Definir y diseñar experimentos junto con sus respectivas métricas y algoritmos que permitan extraer y caracterizar la información de cada una de las representaciones.
4. Evaluar las características y estructura de las representaciones a partir de métricas y algoritmos de detección de comunidades y *clustering*.

Capítulo 3

Estado del Arte

En esta sección discutimos métodos existentes y bibliografía relacionada.

3.1 Redes complejas para representación de texto

Uno de los primeros trabajos que estudió la forma en que las palabras concurren corresponde al de Church y Hanks [7]. En este los autores argumentan que una forma natural de establecer qué palabras están asociadas es cuando coocurren en una frase e introducen técnicas para medir estadísticamente la asociación entre palabras, por ejemplo, mediante información mutua. Posteriormente Ferrer et. al [8] proponen que una forma natural de analizar cómo las palabras coocurren es mediante un grafo en donde dos palabras estarán conectadas si coocurren dentro de un tamaño de ventana w . Además, establecieron una de las primeras conexiones entre la lingüística y la ciencia de redes. Los autores encontraron que una red de palabras construida utilizando una matriz de coocurrencia y tamaño de ventana 2 sobre un corpus de texto exhibe características comúnmente encontradas en sistemas complejos como:

1. El efecto mundo pequeño. Esto es, la distancia promedio entre dos nodos $d \propto \log N$.
2. Distribución de grados libres de escala. Es decir, la probabilidad de encontrar un nodo de grado k es aproximadamente $P(k) = k^{-\gamma}$.

Arruda et al. [9] propusieron los modelos de representación basados en párrafos *Paragraph-based* (PB) y *Adapted paragraph-based* (APB). En el modelo PB los nodos corresponden a palabras y se conectan todas las palabras en un mismo párrafo cuya distancia es menor a d . Por otro lado, el modelo APB extiende el modelo PB considerando únicamente conexiones estadísticamente significativas entre palabras de un mismo párrafo. Estos modelos logran capturar relaciones semánticas y se comportan mejor en la detección de tópicos que métodos basados en una estrategia

bag-of-words. Posteriormente Arruda et al. extienden la idea de representaciones basadas en párrafos [4] representando secuencias de párrafos como nodos y utilizando métricas extraídas sobre la red para entrenar un clasificador que distinga entre textos reales y textos aleatorizados.

Una vez construida la red el área de redes complejas provee una serie de métricas que hacen posible extraer información acerca de la estructura. Una review de Oliva et al. [10] destaca diferentes métricas para identificar características de las redes como lo son el grado de los nodos, modularidad, coeficiente de *clustering* y centralidad. Por otro lado, algoritmos de detección de comunidades como *Louvain* [11] y *Greedy Modularity* pueden ser utilizados para identificar la estructura de comunidad dentro de la red, esto es, conjuntos de nodos que están más conectados entre si que con el resto.

3.2 Word embeddings para representación de texto

Los *word embeddings* pueden ser clasificados dentro de dos grandes familias: (1) estáticos y (2) contextualizados [12]. En una representación estática cada palabra es asociada a un vector único a pesar de que su significado pueda variar de acuerdo al contexto, por ejemplo la palabra banco puede referirse a una institución financiera o a un asiento de madera. Por otro lado, en una representación contextualizada el valor del vector depende de las palabras o *tokens* que la rodean.

Mikolov et al. [5] introdujeron *word2vec*, uno de los primeros métodos eficientes para construir *embeddings* estáticos entrenando una red neuronal bajo un paradigma semi-supervisado. En particular los autores proponen dos arquitecturas denominadas *continuous bag-of-words* y *skip-gram* para entrenar redes neuronales *feedforward* cuyo resultado es un *word embedding*. En la arquitectura *skip-gram* cada palabra es asignada inicialmente a un vector $v \in R^d$ de forma aleatoria y la red es entrenada para predecir el contexto de la palabra, es decir, los vectores correspondientes a las palabras de alrededor. Por otro lado, en la arquitectura *continuous bag-of-words* la red es entrenada para predecir el vector correspondiente a la palabra utilizando el contexto de esta.

Posteriormente comenzó la aparición de enfoques contextualizados como *ELMo* y *BERT* que consideran además el contexto o palabras alrededor de la palabra objetivo y en función de ello asignan un vector numérico [6].

Recientemente el auge de los grandes modelos de lenguajes o *LLM* ha significado la aparición de nuevos modelos basados en la arquitectura *transformer* [13]. Ratford et. [14] al introdujeron *GPT-1*, un modelo *transformer* entrenado en dos etapas. En la primera parte el modelo es entrenado bajo un paradigma de aprendizaje no supervisado sobre un gran corpus de texto. Posteriormente el modelo es especializado realizando una segunda etapa pero de entrenamiento supervisado en alguna tarea como la clasificación de texto. Los autores encontraron que el *embedding* producido en el entrenamiento no supervisado logra capturar distintos aspectos del lenguaje y el modelo logra mejores resultados que *ELMo* y *BERT* en tareas como las de establecer similaridad semántica y



Figura 3.1: Relación entre el *word embedding* generado por el modelo *Word2vec*, redes complejas y espacio hiperbólico.

responder preguntas.

Construido el *embedding* el área de *machine learning* provee diversos algoritmos para identificar grupos de objetos similares. MacQueen introdujo *k-means* [15], un método para asignar n datos a k grupos o *clusters*. Ester et al. [16] introdujeron *DBSCAN*, un método de *clustering* con hiperparámetros que no requiere como entrada el número de *clusters* y que logra mejores resultados que *CLARANS*, una variante de *k-means* en el benchmark SEQUOIA 2000.

3.3 Reconciliando embeddings y redes complejas

Mientras que un enfoque para representación de texto produce como resultado un grafo, un enfoque basado en *embeddings* produce vectores resultando en una dificultosa comparación entre ambas representaciones. Diferentes técnicas han sido desarrollados para transformar redes complejas en *embeddings*, y viceversa. Grover y Leskovec propusieron *node2vec* [17], un método para transformar los nodos de una red a una representación continua tratando de preservar la estructura de vecindades. Cabe destacar que no solo enfoques para transformar redes a *embeddings* han sido desarrollados. Clauset et al. [18] sugieren que una estructura jerárquica en forma de árbol es un principio organizador en redes complejas y que distintas propiedades de estas pueden ser explicadas en base a esta. En función de estas observaciones Krioukov et al. [19] proponen un *framework* para estudiar redes complejas como estructuras jerárquicas en espacio hiperbólico. Intuitivamente las características de un nodo pueden ser vistas como un disco en R^2 , si las características de un nodo n_1 son similares a las de un nodo n_2 , entonces el disco de características n_1 esta contenido en el disco de n_2 o viceversa. Más recientemente Assylbekov y Jangeldin [20] establecieron una serie de conexiones entre *word embeddings*, información mutua, redes complejas y espacio hiperbolico. Entrenar una red neuronal para obtener un *word embedding* utilizando un modelo *skip-gram* con muestreo negativo es aproximadamente equivalente a obtener una matriz de información mutua σ SPMI (*squashed shifted pointwise mutual information*). Las entradas de la matriz σ SPMI satisfacen la propiedad que $\sigma\text{SPMI}_{i,j} \in [0, 1[$ y esta puede ser interpretada como una matriz de adyacencia con las probabilidades de conexión de un grafo aleatorio. Una vez obtenido el grafo es posible comparar la red obtenida con una red construida directamente sobre el texto o en un espacio hiperbólico. Un resumen del argumento se muestra en la Figura 3.1.

Capítulo 4

Redes complejas

La ciencia de redes es un área relativamente nueva del quehacer humano y dar una definición formal resulta difícil. Aun así, siguiendo el enfoque de Latora et al. [21], podemos ilustrar la situación como sigue:

Imaginemos que nos invitan a una fiesta y nos concentramos en las interacciones que se producen entre las personas invitadas. Con seguridad ocurrirá que se formarán distintos grupos de conversación entre personas con intereses similares. Más aún existirán personas que se moverán entre grupos y otros que se agregarán a alguno por alguna amistad mutua con lo que los tamaños de los grupos y su composición puede ser distinta. Por otro lado existirán aquellos que actuarán como el centro de atención y liderarán la conversación mientras que otros simplemente se dedicarán a escuchar.

El sistema social anteriormente descrito es un ejemplo de un sistema complejo. Los sistemas complejos son aquellos sistemas compuestos de unidades interactuando de forma tal que el comportamiento del sistema no es el resultado de la simple combinación de los comportamientos de individuales de cada una de las unidades. Típicamente representamos tales sistemas como una red en donde los nodos representan las unidades y los arcos las interacciones entre estos. Redes de diversa índole han sido estudiadas en la literatura tales como redes de colaboración académica, de relación entre compañías, de texto, etc. Debido a que el tamaño de las redes se ha incrementado a lo largo del tiempo resulta imposible utilizar simplemente la vista para analizar las características y el enfoque ha pasado al análisis estadístico de las propiedades de la red. A pesar del diverso origen que tienen estas redes sorprendentemente poseen múltiples características en común que discutiremos a continuación.

4.1 Fenómenos que se observan en redes complejas

4.1.1 El efecto mundo pequeño

El efecto mundo pequeño está asociado históricamente con Stanley Milgram [22] y se refiere a que un individuo en redes sociales (y nodos en general, en otras redes) pueden llegar a cualquier otro individuo de la red un número pequeño de pasos. Este efecto ha sido observado en redes de diversa índole como redes sociales, de información y biológicas[23]. Watts y Strogatz[24] encontraron que tomando una red regular y reconectando los arcos con probabilidad p es posible obtener una red que no es ni regular ni aleatoria y que posee las características de mundo pequeño. Más formalmente, decimos que una red posee el efecto mundo pequeño si la distancia promedio entre nodos $l \approx l_{random} \propto \frac{\log n}{\log k}$ donde n es la cantidad de nodos y k es el grado promedio.

La distancia promedio entre nodos la podemos caracterizar como

$$l = \frac{1}{n(n-1)} \sum_{(i,j) \in V} d(i,j) \quad (4.1)$$

donde n corresponde a la cantidad de nodos y $d(i,j)$ es la distancia geodésica o distancia más corta entre los nodos i y j .

4.1.2 Clustering

El *clustering* o transitividad se refiere a un fenómeno de redes en que si un nodo A está conectado a un nodo B y el nodo B está conectado a un nodo C , entonces hay más probabilidades que el nodo A esté conectado al nodo C [23]. Un gran número de redes en la literatura presentan esta tendencia de generar conexiones entre nodos vecinos la cual se desvía de redes aleatorias en donde estos triángulos son escasos [25]. Este fenómeno fue observado inicialmente en redes sociales en donde muy probablemente los amigos de una persona son también amigos.

Se define el coeficiente de *clustering* para el nodo i como:

$$C_i = \frac{2 \cdot \text{número de triángulos que pasan por el nodo } i}{\text{número de triples centrados en el nodo } i} \quad (4.2)$$

La ecuación 4.2 puede ser reescrita para grafos no dirigidos como:

$$C_i = \frac{2 \cdot \text{número de triángulos conectados al nodo } i}{\deg(i)(\deg(i) - 1)} \quad (4.3)$$

donde $\deg(i)$ corresponde al grado del nodo i . Una vez computados los coeficientes de *clustering* para cada nodo de la red obtenemos el *clustering promedio* como:

$$C = \frac{1}{n} \cdot \sum_{v \in V} C_v \quad (4.4)$$

Diferentes generalizaciones han sido propuestas en la literatura para el caso de redes con pesos [25]. Particularmente Onnela et al. [26] introducen una generalización para evaluar la intensidad de las conexiones entre nodos que forman triángulos en la red en comparación al máximo peso de la red. Los autores definen el coeficiente de *clustering* para un nodo particular i como

$$\tilde{C}_i = \frac{1}{k_i(k_i - 1)} \sum_{j,k} (\hat{w}_{i,j} \hat{w}_{j,k} \hat{w}_{k,i})^{1/3} \quad (4.5)$$

donde $\hat{w}_{i,j} = \frac{w_{i,j}}{\max(w)}$. Los autores además prueban que $\tilde{C}_i \rightarrow C_i$ en medida que los pesos se vuelven binarios y proveen una ecuación para relacionar el coeficiente de *clustering* sin pesos C_i con el coeficiente de *clustering* con pesos $\tilde{C}_i$ que introducimos a continuación. Sea $G(V, E)$ un grafo y sea $H(V', E') \subseteq G$ un subgrafo de $G(V, E)$, definimos la intensidad $I(H)$ del subgrafo como

$$I(H) = \left(\prod_{(i,j) \in E'} \hat{w}_{ij} \right)^{1/|E'|} \quad (4.6)$$

Ya que cada triángulo de la red corresponde a un subgrafo es posible calcular para cada nodo i la intensidad de cada uno de los triángulos que participa. Consecuentemente podemos además calcular la intensidad promedio $\bar{I}_i$ como

$$\bar{I}_i = \frac{1}{t_i} \sum_{g \in N(i)} I(g) \quad (4.7)$$

donde t_i son los triángulos en los que participa el nodo i , $N(i)$ es el conjunto vecindad y g es un triángulo. Podemos entonces notar que

$$\tilde{C}_i = \bar{I}_i C_i \quad (4.8)$$

La ecuación 4.8 relaciona el *clustering* con y sin pesos utilizando la intensidad promedio.

4.1.3 Redes libre de escala

Otro fenómeno que resulta común a múltiples redes es una distribución de grados libre de escala. Los nodos de una red siguen una distribución libre de escala si el grado de los nodos satisface la expresión:

$$P(k) = k^{-\gamma} \quad (4.9)$$

Lo anterior implica intuitivamente que la mayor parte de los nodos red tendrán pocas conexiones mientras que unos pocos nodos acaparan la mayor parte de las conexiones de la red. Podemos poner a prueba la condición anterior notando que podemos reescribir la ecuación 4.9 como:

$$\begin{aligned} \log P(k) &= \log(k^{-\gamma}) \\ &= -\gamma \log(k) \end{aligned} \tag{4.10}$$

Es decir, si la distribución de grados de la red satisface 4.9 debiésemos observar al graficar la distribución con ambos ejes en escala logarítmica una relación aproximadamente lineal.

4.1.4 Asortatividad

Uno de los fenómenos a menudo observado en redes sociales corresponde a la propensión de los individuos a relacionarse con otros individuos similares. Por ejemplo, un estudio de los habitantes de la ciudad de San Francisco reveló que los participantes de distintas razas tendían a elegir una pareja de su misma raza [23].

Una métrica más agnóstica y que resulta útil para establecer este tipo de fenómenos es mediante el grado promedio de los nodos vecinos que definimos a continuación. Sea i un nodo; definimos el grado promedio de los nodos vecinos, $k_{nn,i}$, como

$$k_{nn,i} = \frac{1}{|N(i)|} \sum_{j \in N(i)} k_j \tag{4.11}$$

donde $N(i)$ es el conjunto de nodos que son vecinos directos del nodo i y k_j corresponde al grado del nodo j . Si la red presenta una estructura asortativa en términos de correlación de grados, entonces esperaríamos observar que los nodos de menor grado tienden a conectarse con nodos de menor grado y que su grado promedio de los nodos vecinos es bajo. Del mismo modo esperaríamos en el caso de nodos de alto grado que el grado promedio de los nodos vecinos sea alto.

Esta métrica fue generalizada por Barrat et al. [27] para el caso de redes con pesos. Definimos el grado promedio de los nodos vecinos con pesos, $k_{nn,i}^w$, como

$$k_{nn,i}^w = \frac{1}{s_i} \sum_{j \in N(i)} w_{ij} k_j \tag{4.12}$$

donde s_i es la suma de los pesos de los arcos del nodo i .

Una review de M. E. J. Newman [23] identifico las propiedades y coeficientes previamente descritos para diversos tipos de redes complejas.

4.2 Métodos de redes

Habiendo establecido el hecho que las representaciones vectores palabra aprendidos por el modelo *word2vec* pueden ser transformados a redes queda la pregunta de cómo comparar esta representación con la obtenida mediante coocurrencia. Un enfoque directo sería comparar las características comunes a varias redes complejas descritas en la sección previa. Otro enfoque sería comparar las estructuras posiblemente de miles de nodos que describen ambas redes. A continuación describimos métodos para extraer información respecto a la estructura de redes.

4.2.1 El método de Louvain

Uno de los objetivos del estudio de redes es comprender el funcionamiento del sistema que representan. Uno de los enfoques para lograr esto es identificar grupos o *clusters* en la red que permitan revelar la organización y estructura del sistema. Más aún, el encontrar grupos nos permite separar la red en un conjunto de redes más pequeñas y trabajables que pueden ser analizadas por separado. El método de *Louvain* pertenece a la categoría de algoritmos de detección de comunidades que apuntan a descubrir los distintos grupos dentro de las redes. Informalmente la idea es que los grupos encontrados deben ser conjuntos de nodos que están más densamente conectados entre si que con el resto de nodos. Para medir el grado de cohesión de un grupo es necesario introducir algún tipo de métrica y asignar los nodos a los distintos grupos a modo de maximizarla. Tal métrica se denomina la modularidad, que introducimos a continuación.

Supongamos que la red ha sido separada en N grupos y que g_i denota el grupo al cual pertenece el nodo i con $g_i \in 1, \dots, N$. Sea $G(V, E)$ un grafo con matriz de adyacencia A , podemos calcular la cantidad de conexiones o arcos dentro de nodos asignados a un mismo grupo como

$$\sum_{(i,j)} \frac{1}{2} A_{ij} \delta(g_i, g_j) \quad (4.13)$$

donde el término $\frac{1}{2}$ previene contar cada arco dos veces y $\delta(g_i, g_j)$ es la función Kronecker delta definida como

$$\delta(g_i, g_j) = \begin{cases} 0, & g_i \neq g_j \\ 1, & g_i = g_j \end{cases} \quad (4.14)$$

Esta última permite únicamente sumar los arcos entre nodos asignados a un mismo grupo. Por otro lado si k_i representa el grado del nodo i y m la cantidad de arcos en la red, el valor esperado de la cantidad de arcos entre todos los nodos de un mismo tipo si los arcos fuesen asignados al azar es

$$\frac{1}{2} \sum_{(i,j)} \frac{k_i k_j}{2m} \delta(g_i, g_j) \quad (4.15)$$

Finalmente comparamos la diferencia entre la cantidad de arcos entre nodos de un mismo tipo y el valor esperado, y definimos la modularidad Q como

$$Q = \frac{1}{2m} \sum_{(i,j)} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(g_i, g_j) \quad (4.16)$$

El método de Louvain apunta a asignar a los nodos a comunidades de modo de maximizar Q . Inicialmente todos los nodos son asignados a su propia comunidad. Posteriormente para cada nodo i se computa la ganancia de modularidad ΔQ obtenida al remover al nodo i de su propia comunidad e insertarlo en la comunidad de un nodo vecino g_j . Este proceso se repite para cada posible comunidad y una vez computadas todas las variaciones el nodo i es entonces asignado a la comunidad que provee el mayor incremento en modularidad. Luego el proceso se repite para el siguiente nodo. Posteriormente cada comunidad es reducida a un único nodo asignando cada uno de estos nuevos nodos a una comunidad única y el proceso se repite.

4.2.2 *k-core*

El concepto de *k-core* es típicamente asociado a Stephen Seidman. Puesto que una red puede contener una gran cantidad de nodos un enfoque podría ser el de simplificar la red a únicamente aquellos nodos significativos en alguna manera. El algoritmo de *k-core* apunta a simplificar la red de tal modo que únicamente aquellos nodos más densamente conectados entre si permanezcan en esta. A continuación damos una definición formal del concepto *k-core*.

Definición 2. *k-core.* Sea $G(V, E)$ un grafo y $H(W, F)$ un subgrafo de $G(V, E)$ tal que $W \subseteq V$ y $F = \{(u, v) \in E \mid u, v \in W\}$. Si $\deg_H(v)$ denota el grado del nodo v en $H(W, F)$, decimos que H es un *k-core* si H es el subgrafo maximal que satisface que $\deg_H(v) \geq k$ para todo $v \in W$.

La definición anterior implica que si G es un grafo y $k \in \mathbb{N}$, el *k-core* es un subgrafo de G en donde todos los nodos tienen al menos k arcos o conexiones entre si. Por otro lado, puesto que la cantidad de conexiones entre nodos es finita eventualmente existe un máximo valor de k tal que el $(k+1)$ -core es vacío. Para cada nodo v de G se define el número de *core*, $\text{core}(v)$, como el máximo valor de k tal que el nodo v pertenece al *k-core* pero no al $(k+1)$ -core. El primer algoritmo para obtener una descomposición de *k-core* fue propuesto por Batagelj y Zaverinik [28]. Puesto que el *k-core* es único, el algoritmo utilizado resulta en el mismo subgrafo y la elección de este depende de la eficiencia que se requiera. Aquí presentamos el enfoque seguido en [28] en dos pasos, obtener el

número de *core* de cada nodo y luego utilizarlo para recortar la red. El algoritmo se basa en la siguiente proposición auxiliar.

Proposición 1. (Batagelj y Zavernik [28]). Sea $G(V, E)$ un grafo, si recursivamente eliminamos los nodos cuyo grado $\deg(v) < k$ y los arcos que inciden sobre ellos, el subgrafo resultante es un k -core.

El método para obtener el número de *core* de cada nodo se presente en el algoritmo 1. Una vez obtenido el número de *core* el subgrafo k -core se puede obtener simplemente eliminando todos los nodos cuyo valor de *core* sea menor a k y los arcos incidentes a este.

Function *Core number* $G(V, E)$:

Input: Grafo $G(V, E)$

Output: Lista con el *core number*, $\text{core}(v)$, de cada nodo.

// Ordenar nodos por la cantidad de arcos

$V_ordered := \text{sort}(V, \text{by}='degree', \text{order}='ascending')$

for $node$ *in* $V_ordered$ **do**

$\text{core}[node] := \deg(node)$

for u *in* $\text{Neighborhood}(node, G)$ **do**

 // Iterar por nodos vecinos reduciendo el grado. Equivalente a
 recursion

if $\deg(u) > \deg(node)$ **then**

$\deg(u) := \deg(u) - 1$

$V_ordered := \text{sort}(V, \text{by}='degree', \text{order}='ascending')$

return core

Algoritmo 1: Core number

Capítulo 5

Aprendizaje automático

En esta sección describimos las distintas técnicas del área de aprendizaje automático que utilizamos.

5.1 Word2vec

El modelo *word2vec* se presenta típicamente en la forma de una red neuronal *feedforward* la cual puede tener 2 tipos de arquitecturas, *skip-gram* o *continuous bag-of-words*. Para nuestro propósito únicamente nos interesa la arquitectura *skip-gram* que describiremos a continuación. El objetivo computacional de *word2vec* con la arquitectura *skip-gram*, es dado un corpus de texto T , aprender una representación vectorial o *embedding* de las palabras en T que permita recuperar para cada palabra $w \in T$ las palabras que se encuentran en el contexto de w . Supongamos que el corpus T contiene la frase: “El nombre de la canción es llamado ‘Ojos de bacalao’.” En el caso más simple podemos decir que el contexto de la palabra ‘Ojos’ son las palabras ‘llamado’ y ‘de’. Dada la palabra ‘Ojos’ esperamos que el modelo prediga ‘llamado’ y ‘de’ como las palabras que se encuentran en el contexto. Dada una palabra central la cantidad de palabras a la izquierda y la derecha a considerar en el modelo corresponden a un hiperparámetro. Las representaciones vectoriales que aprende la red las denominamos vectores palabra. La salida del modelo no serán las palabras como tal sino que dada una palabra se utilizarán los vectores palabra aprendidos para calcular para cada palabra la probabilidad de que ésta esté en el contexto de la entrada. La red neuronal almacenará en su capa oculta dos matrices $\mathbf{W}$ y $\mathbf{C}$ cuyas filas representarán los vectores palabras aprendidos cuando la palabra ocurre como palabra central en una sentencia y cuando ocurre en el contexto de otra palabra respectivamente; los valores de estos vectores son ajustados durante el entrenamiento.

5.1.1 Entrada del modelo

El texto T no es entregado propiamente tal de forma pura al modelo sino que debe ser previamente procesado transformándolo a una lista de palabras. En lo que sigue consideraremos que el texto ha sido preprocesado a una lista de palabras que denotamos D . Una descripción más detallada de como se preprocesa el texto se presenta en la sección 6.1.2. Una simplificación hecha al momento de entregar las entradas al modelo *skip-gram* es la de no entregar directamente las palabras sino una representación de ellas. Para ello supongamos que V es el vocabulario o conjunto de palabras únicas en el texto, este vocabulario es obtenido identificando las palabras únicas que ocurren en la lista de palabras D . Representamos las palabras utilizando la notación *one-hot encoding*, esto es, a cada palabra $w \in V$ asignamos un índice único $i \in 1, \dots, |V|$ y entregamos al modelo una palabra como un vector que tiene ceros en todas las posiciones con excepción de aquella que corresponde al índice de la palabra. La dimensionalidad de los vectores *one-hot encoding* corresponderá a la cardinalidad del conjunto vocabulario $|V|$. Lo anterior permite obtener el vector correspondiente a la palabra en el *embedding* como una simple multiplicación matricial. Si $\mathbf{e}_w$ es la representación *one-hot encoding* de $w \in V$ y $\mathbf{W}$ es un *embedding*, podemos obtener la representación de w en $\mathbf{W}$ como el producto $\mathbf{W}\mathbf{e}_w$. La dimensionalidad de los vectores palabra en el *embedding* es un hiperparámetro que denotaremos por L . Consecuentemente la dimensionalidad de las matrices *embedding* es $L \times |V|$.

5.1.2 Muestreo negativo

Entrenar el modelo *skip-gram* sin realizar ningún tipo de optimización es extremadamente costoso. El problema viene dado por el hecho de que la salida del modelo serán probabilidades de que cada una de las palabras se encuentren en el contexto de la entrada. Estas probabilidades serán estimadas utilizando la función *softmax* definida como $\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{|V|} e^{z_j}}$. Lo anterior implica que la capa de salida de la red neuronal tendrá tamaño igual a la cantidad de palabras únicas $|V|$. Durante el entrenamiento la red iterará sobre cada una de las palabras en la lista de palabras D . Sea S el tamaño de la ventana, entonces se tomarán $2 \cdot S$ palabras en el contexto del centro (S palabras a la izquierda y S a la derecha). Puesto que el proceso de estimar las probabilidades usando las representaciones aprendidas requiere estimar sobre todo el vocabulario $|V|$, el costo computacional de entrenamiento por palabra en la lista $2 \cdot S \cdot |V|$. Finalmente, puesto que el proceso se repite para cada palabra en el vocabulario, tenemos que el costo computacional es $\mathcal{O}(S \cdot |V| \cdot |D|)$.

Lo anterior resulta prohibitivo cuando se tiene un número muy grande de palabras y se requieren alternativas para hacer el entrenamiento más eficiente. Una de esas alternativas corresponde al muestreo negativo que se basa en la idea de no predecir las probabilidades para todas las $|V|$ palabras del vocabulario sino que para una palabra w seleccionar un número $K < |V|$ de palabras

que no están en el contexto de esta y computar la pérdida con estas. Estas K palabras que no pertenecen al contexto de w se denominan una muestra negativa.

5.1.3 Hiperparámetros del modelo

Teniendo en cuenta lo anterior enumeramos los hiperparámetros del modelo

1. $|V|$: Cardinalidad del conjunto vocabulario. Determina la cantidad de columnas o vectores en los embeddings.
2. S : Tamaño de la ventana. Dada una palabra central w controla la cantidad de palabras a la izquierda y a la derecha de esta a considerar como contexto.
3. L : Dimensionalidad de los vectores palabra en los *embeddings*.
4. K : Número de ejemplos negativos a mostrar a la red. Típicamente se toman valores en el rango 5-20.

Uno de los hiperparámetros más importantes del modelo es la dimensionalidad de los vectores *embeddings* L puesto que esto determina la cantidad de parámetros entrenables de la red. Dado un vector palabra, cada una de sus componentes corresponde a un parámetro entrenable de la red puesto que ajustamos estas componentes para que palabras dentro del contexto tengan vectores similares. Puesto que el tamaño del vocabulario es $|V|$ y el tamaño de cada vector palabra será L , cada *embedding* poseerá $L \cdot |V|$ parámetros entrenables. Puesto que la red tendrá dos *embeddings* $\mathbf{W}$ y $\mathbf{C}$ tenemos en total $2 \cdot L \cdot |V|$ parámetros entrenables.

5.1.4 Arquitectura de la red neuronal

A continuación describimos un paso por la red neuronal. En principio trabajaremos el modelo Word2vec skip-gram estándar y luego incorporaremos el muestreo negativo. Supondremos que los vectores palabras en los embeddings $\mathbf{W}$ y $\mathbf{C}$ son inicializados con valores al azar. El primer paso corresponde a entregar la entrada al modelo. Para ello una palabra es transformada a su representación *one-hot encoding* según se describió en la sección 5.1.1. Sea $w \in V$ una palabra y $\mathbf{e}_w$ su vector *one-hot*, la red obtendrá la representación de w como un vector palabra realizando una multiplicación matriz vector de la forma $\mathbf{W}\mathbf{e}_w$. Puesto que $\mathbf{e}_w$ es un vector fila con todas sus componentes 0 con excepción de aquella correspondiente al índice i asignado a la palabra esta multiplicación matricial únicamente esta seleccionando el vector palabra central asignado a la palabra w en el *embedding* $\mathbf{W}$. Este primer paso por la red neuronal esta representada en la Figura 5.1.

$$\begin{array}{ccccc}
\mathbf{e}_w = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} & \longrightarrow & \begin{pmatrix} W_{1,1} & \dots & W_{1,i} & \dots & W_{1,|V|} \\ W_{2,1} & \dots & W_{2,i} & \dots & W_{2,|V|} \\ \vdots & & \vdots & & \vdots \\ W_{L,1} & \dots & W_{L,i} & \dots & W_{L,|V|} \end{pmatrix} & \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} & \longrightarrow & \begin{pmatrix} W_{1,i} \\ W_{2,i} \\ \vdots \\ W_{L,i} \end{pmatrix} \\
\mathbf{e}_w & & \mathbf{W} & & \mathbf{e}_w & & \mathbf{We}_w
\end{array}$$

Figura 5.1: Diagrama del primer paso de la arquitectura *skip-gram word2vec*.

$$\begin{array}{ccccc}
\mathbf{We}_w = \begin{pmatrix} W_{1,i} \\ W_{2,i} \\ \vdots \\ W_{L,i} \end{pmatrix} & \longrightarrow & \begin{pmatrix} C_{1,1} & C_{2,1} & \dots & C_{L,1} \\ C_{1,2} & C_{2,2} & \dots & C_{L,2} \\ \vdots & & \vdots & \vdots \\ C_{1,|V|} & C_{2,|V|} & \dots & C_{L,|V|} \end{pmatrix} & \begin{pmatrix} W_{1,i} \\ W_{2,i} \\ \vdots \\ W_{L,i} \end{pmatrix} & \longrightarrow & \begin{pmatrix} \mathbf{C}_1(\mathbf{We}_w) \\ \mathbf{C}_2(\mathbf{We}_w) \\ \vdots \\ \mathbf{C}_{|V|}(\mathbf{We}_w) \end{pmatrix} \\
\mathbf{We}_w & & \mathbf{C}^\top & & \mathbf{We}_w & & \mathbf{C}^\top(\mathbf{We}_w)
\end{array}$$

Figura 5.2: Diagrama del segundo paso de la arquitectura *skip-gram word2vec*.

La forma estándar de determinar si dos vectores son similares es computar el producto punto entre estos. Una vez obtenida la representación vector palabra de w el siguiente paso corresponde a computar el producto punto con el resto de vectores palabra contexto en el *embedding* $\mathbf{C}$. Puesto que las filas de la matriz $\mathbf{C}$ son los vectores palabra del contexto, la matriz $\mathbf{C}$ también tiene dimensiones $L \times |V|$. Podemos computar el producto punto con cada uno de los vectores en $\mathbf{C}$ como $\mathbf{C}^\top(\mathbf{We}_w)$. Lo anterior se representa en la figura 5.2.

Una vez computados los productos puntos la última capa de la red aplicará la función *softmax* sobre cada uno de estos para llevarlos al rango $[0, 1]$. Para cada palabra $w_i \in V$ computamos la probabilidad que w_i esté en el contexto de la entrada como

$$\sigma[\mathbf{C}^\top(\mathbf{We}_w)]_i = \frac{e^{\mathbf{C}_i^\top(\mathbf{we}_w)}}{\sum_{j=1} e^{\mathbf{C}_j^\top(\mathbf{we}_w)}} \quad (5.1)$$

Esto finaliza un paso por la red neuronal. Posteriormente los parámetros del modelo serán actualizados de acuerdo a la función objetivo.

5.1.5 Función objetivo estándar

Para optimizar el modelo consideremos una palabra w_i . Sea S el tamaño de la ventana y supongamos que w_{i-k} y w_{i+k} son palabras que se encuentran k posiciones a la izquierda y k posiciones a la derecha de w_i respectivamente con $k \in 1, \dots, S$. Puesto que w_{i-k} y w_{i+k} se encuentran dentro del contexto de w_i nos interesa maximizar la probabilidad de observarlas dado el hecho que entregamos w_i como entrada al modelo. Si denotamos tales probabilidades como $p(w_{i-k}|w_i; \theta)$ entonces queremos maximizar

$$\prod_{i-S \leq j \leq i+S, j \neq i} p(w_j|w_i; \theta) \quad (5.2)$$

Considerando cada una de las palabras en la lista de palabras D esto se transforma en

$$\arg \max_{\theta} \prod_{i=1}^{|D|} \prod_{i-S \leq j \leq i+S, j \neq i} p(w_j|w_i; \theta) \quad (5.3)$$

En Aprendizaje automático resulta más común minimizar funciones que maximizarlas. Por otro lado, puesto que la función logaritmo es monótona, maximizar la ecuación 5.3 es equivalente a

$$\arg \min_{\theta} - \left[\sum_{i=1}^{|D|} \sum_{i-S \leq j \leq i+S, j \neq i} \log p(w_j|w_i; \theta) \right] \quad (5.4)$$

Luego, puesto que estimamos $p(w_j|w_i; \theta)$ usando la función *softmax*, la ecuación 5.4 se transforma en

$$\begin{aligned} & \arg \min_{\theta} - \left[\sum_{i=1}^{|D|} \sum_{i-S \leq j \leq i+S, j \neq i} \log \frac{e^{\mathbf{C}_j^{\top}(\mathbf{W}\mathbf{e}_i)}}{\sum_{k=1}^{|D|} e^{\mathbf{C}_k^{\top}(\mathbf{W}\mathbf{e}_i)}} \right] \\ &= \arg \min_{\theta} - \left[\sum_{i=1}^{|D|} \sum_{i-S \leq j \leq i+S, j \neq i} \log e^{\mathbf{C}_j^{\top}(\mathbf{W}\mathbf{e}_i)} - \log \sum_{k=1}^{|D|} e^{\mathbf{C}_k^{\top}(\mathbf{W}\mathbf{e}_i)} \right] \end{aligned} \quad (5.5)$$

La ecuación 5.5 corresponde al objetivo estándar o *naive* del modelo *word2vec skip-gram*. Notemos que el término $\log \sum_{k=1}^{|D|} e^{\mathbf{C}_k^{\top}(\mathbf{W}\mathbf{e}_i)}$ previene que el modelo asigne a todas las palabras el mismo vector palabra. Lo anterior es un modo de garantizar que las representaciones obtenidas sean significativas respecto al objetivo. Habiendo obtenido la función objetivo lo único que queda es entrenar el modelo. Para ello se utilizan métodos estándar de redes neuronales como *backpropagation*. Como se mencionó en la sección 5.1.2 entrenar el modelo usando este objetivo es extremadamente costoso. Para solventar este problema se modifica la función objetivo para incorporar el así llamado

muestreo negativo.

5.1.6 Función objetivo optimizada con muestreo negativo

Para obtener una nueva función objetivo con muestreo negativo planteamos la situación siguiendo el enfoque de Goldberg y Levy [29]. Sea (w, c) un par palabra y contexto. Denotemos por $p(D = 1|w, c)$ la probabilidad que el par (w, c) pertenezca al corpus y consecuentemente $p(D = 0|w, c)$ la probabilidad que no pertenezca al corpus. Podemos formular el objetivo como encontrar parámetros que maximizan la probabilidad

$$\begin{aligned} & \arg \max_{\theta} \prod_{w \in W} \prod_{c \in C(W)} p(D = 1|w, c; \theta) \\ &= \arg \max_{\theta} \prod_{(w, c) \in D} p(D = 1|w, c; \theta) \end{aligned} \quad (5.6)$$

Sin embargo, tal problema tiene una solución trivial poniendo $p(D = 1|w, c) = 1$ para cada par (w, c) incluso cuando la palabra c no se encuentre en el contexto de w . Esto supone un problema pues queremos que nuestro modelo aprenda relaciones significativas entre palabras. Para solventar este problema consideremos un par (w, c') tal que c' no se encuentra en el contexto de la palabra w . Ya que c' no se encuentra en el contexto de la palabra deberíamos tener que la probabilidad $p(D = 1|w, c')$ fuese baja o equivalentemente que la probabilidad $p(D = 0|w, c')$ fuese alta. Es decir, queremos que $p(D = 1|w, c') = 1 - p(D = 0|w, c')$ sea baja cuando c' no pertenezca al contexto w . Con esto en mente podemos reformular el objetivo. Sea D' un conjunto de pares palabra-contexto aleatorios (w, c') , queremos maximizar la productoria

$$\arg \max_{\theta} \prod_{(w, c) \in D} p(D = 1|w, c; \theta) \prod_{(w, c') \in D'} p(D = 0|w, c'; \theta) \quad (5.7)$$

Para parametrizar el modelo estimamos $p(D = 1|w, c; \theta)$ utilizando la función sigmoideal,

$$p(D = 1|w, c; \theta) = \frac{1}{1 + e^{-w \cdot c}}$$

Así pues, podemos reescribir la ecuación (5.7) como

$$\begin{aligned} & \arg \max_{\theta} \prod_{(w, c) \in D} P(D = 1|w, c; \theta) \prod_{(w, c') \in D'} P(D = 0|w, c'; \theta) \\ &= \arg \max_{\theta} \prod_{(w, c) \in D} \frac{1}{1 + e^{-w \cdot c}} \prod_{(w, c') \in D'} \left[1 - \frac{1}{1 + e^{-w \cdot c'}} \right] \end{aligned} \quad (5.8)$$

Denotando $\sigma(x) = \frac{1}{1+e^{-x}}$ reescribimos la ecuación 5.8 como

$$\begin{aligned}
& \arg \max_{\theta} \prod_{(w,c) \in D} \frac{1}{1+e^{-w \cdot c}} \prod_{(w,c') \in D'} \left[1 - \frac{1}{1+e^{-w \cdot c'}} \right] \\
&= \arg \max_{\theta} \prod_{(w,c) \in D} \frac{1}{1+e^{-w \cdot c}} \prod_{(w,c') \in D'} \frac{e^{-w \cdot c'}}{1+e^{-w \cdot c'}} \\
&= \arg \max_{\theta} \prod_{(w,c) \in D} \frac{1}{1+e^{-w \cdot c}} \prod_{(w,c') \in D'} \frac{1}{e^{w \cdot c'}(1+e^{-w \cdot c'})} \\
&= \arg \max_{\theta} \prod_{(w,c) \in D} \frac{1}{1+e^{-w \cdot c}} \prod_{(w,c') \in D'} \frac{1}{1+e^{w \cdot c'}} \\
&= \arg \max_{\theta} \prod_{(w,c) \in D} \sigma(w \cdot c) \prod_{(w,c') \in D'} \sigma(-w \cdot c')
\end{aligned} \tag{5.9}$$

Finalmente, podemos utilizar la función logaritmo para transformar la productoria 5.9 en una suma

$$\begin{aligned}
& \arg \max_{\theta} \prod_{(w,c) \in D} \sigma(w \cdot c) \prod_{(w,c') \in D'} \sigma(-w \cdot c') \\
&= \arg \max_{\theta} \sum_{(w,c) \in D} \log \sigma(w \cdot c) + \sum_{(w,c') \in D'} \log \sigma(-w \cdot c')
\end{aligned} \tag{5.10}$$

Los pares negativos (w, c') son obtenidos de una distribución unigram elevada a la potencia $\frac{3}{4}$, esto es, $P(w_i) = \frac{Freq(w_i)^{3/4}}{Z}$ donde $Freq(w_i)$ corresponde a la frecuencia con la que aparece la palabra w_i en el texto y Z es una constante de normalización. Para implementar la selección de las muestras negativas utilizamos una tabla de *lookup* para que el costo computacional asociado a estas sea $\mathcal{O}(1)$. La ecuación 5.10 corresponde al objetivo del modelo *word2vec* con muestreo negativo. Por último haremos dos consideraciones. La primera es que el objetivo se calculará para cada palabra únicamente con K muestras de la distribución $P(w_i)$ y la segunda es que consideramos una arquitectura con dos *embeddings* $\mathbf{W}$ y $\mathbf{C}$. Para entrenar utilizando este nuevo objetivo debemos iterar nuevamente sobre cada una de las palabras en D seleccionando $2 \cdot S$ palabras en su contexto y además seleccionando K muestras negativas. Puesto que K es constante el nuevo costo computacional de entrenar el modelo es $\mathcal{O}(S \cdot |D|)$.

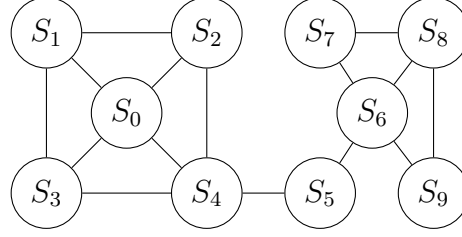


Figura 5.3: Grafo de ejemplo para el algoritmo node2vec

5.2 Node2vec

Habiendo establecido el hecho que los *embeddings* aprendidos por el modelo *word2vec* pueden ser transformados a una red y posteriormente analizados mediante técnicas de redes una pregunta razonable es si podemos transformar redes a *embeddings* en forma eficiente para así analizarlos mediante técnicas del área de reconocimiento de patrones y aprendizaje automático. Esta pregunta fue respondida positivamente por Grover y Leskovec [17] quienes propusieron *node2vec*, una técnica para aprender representaciones continuas de los nodos de una red. *Node2vec* está íntimamente relacionado con el modelo *word2vec* previamente introducido, en particular *node2vec* es un algoritmo que transforma la red a un equivalente del texto en el modelo *word2vec* y luego utiliza el modelo *skip-gram* para obtener las representaciones vectoriales. El objetivo computacional de *node2vec* es aprender un *embedding* o representación vectorial de los nodos en la red G que maximice la probabilidad de preservar las vecindades de los nodos en la red. La principal contribución de *node2vec* es flexibilizar la noción de vecindad para un nodo. A continuación presentamos un ejemplo.

Si observamos la red de la Figura 5.3 podemos notar que las vecindades de los nodos S_0 y S_6 corresponden a los conjuntos de nodos $\{S_1, S_2, S_3, S_4\}$ y $\{S_5, S_7, S_8, S_9\}$ respectivamente. Dando como entrada al modelo el nodo S_0 esperaríamos que las representaciones vectoriales aprendidas nos permitan recuperar como salida la vecindad del nodo, esto es el conjunto $\{S_1, S_2, S_3, S_4\}$. Sin embargo, idealmente estas representaciones de los nodos debiesen no solo capturar la estructura local sino también la estructura global de la red. En la Figura 5.3 los nodos S_0 y S_6 poseen un rol similar en sus respectivas vecindades actuando como un nodo central y por ende sus representaciones también debiesen guardar un grado de similaridad. Para lograr esto *node2vec* extiende la idea de vecindad no solo a aquellos nodos inmediatos sino que a nodos que posiblemente están más lejos en la red. La vecindad de un nodo u es obtenida mediante un paseo aleatorio por la red comenzando en el nodo u que tiene probabilidades asociadas a visitar nodos más cercanos y nodos más lejanos en la red. Las probabilidades asociadas a visitar nodos más cercanos o lejanos así como el largo de la caminata son hiperparámetros del modelo. Comenzando en el nodo S_0 un paseo aleatorio de largo 6 podría por ejemplo ser la secuencia $(S_0, S_2, S_4, S_5, S_6, S_8)$, tal secuencia puede ser interpretada como una frase en el modelo *skip-gram* de *word2vec* y por tanto si pensamos en S_4

como una palabra, el contexto de S_4 son los nodos S_2 y S_5 . En caso de considerar un contexto de tamaño 2 la vecindad de S_4 incluiría además los nodos S_0 y S_6 . Al igual que en modelo *word2vec* el tamaño del contexto a considerar es un hiperparámetro.

5.2.1 Entrada del modelo

La entrada del modelo *node2vec* corresponderá simplemente a un grafo $G(V, E)$. Como se mencionó anteriormente el modelo *node2vec* utilizará la arquitectura *skip-gram* del modelo *word2vec* con lo cual se compartirán las mismas entradas. Sin embargo, antes de entrenar el modelo *skip-gram* debemos transformar el grafo a un equivalente del texto. El equivalente al texto en este modelo corresponderá a una secuencia de paseos aleatorios. Sea $G(V, E)$ un grafo, para cada nodo $v \in V$ de la red se realizará un paseo aleatorio comenzando en el nodo v de largo k . Para disminuir la aleatoriedad para cada nodo se repite el paseo r veces. A partir de cada uno de los paseos aleatorios obtenemos ejemplos de entrenamientos para el modelo *skip-gram*.

5.2.2 Estrategias para la obtención de vecindades

Para determinar las vecindades de un nodo el modelo *node2vec* realizará un paseo aleatorio en donde alternará entre dos estrategias de búsqueda clásicas, BFS y DFS.

- Breadth-first Sampling (BFS): Dado un nodo inicial v , la técnica BFS se limita a visitar únicamente los nodos que son vecinos inmediatos del nodo v . En la Figura 5.3 dado S_1 como nodo inicial una estrategia BFS daría como resultado los nodos $\{S_0, S_1, S_2\}$.
- Depth-first Sampling (DFS): Dado un nodo inicial v , la técnica DFS visita nodos a distancias incrementales de v . En la Figura 5.3 dado S_0 como nodo inicial la estrategia DFS visita los nodos $\{S_4, S_5, S_6\}$

Para controlar la estrategia de búsqueda a utilizar el modelo provee dos hiperparámetros p y q .

5.2.3 Paseos aleatorios

Un paseo aleatorio por la red comienza en nodo v y posteriormente, con una probabilidad determinada, uno de sus vecinos es visitado. Estando en el nodo vecino el proceso se repite. Tradicionalmente la probabilidad de visitar un nodo w en el paso i -ésimo está dada por

$$P(c_i = w | c_{i-1} = v) = \begin{cases} \frac{\pi_{vw}}{Z}, & (v, w) \in V \\ 0, & (v, w) \notin V \end{cases} \quad (5.11)$$

en donde π_{vw} es la probabilidad de transición no normalizada y Z es una constante de normalización. En el caso de *node2vec* utilizamos la probabilidad de transición como el peso de los arcos entre nodos. El procedimiento actualmente definido aún no incorpora las estrategias de búsqueda alternadas definidas en la sección 5.2.2. Para incorporarlas se define un paseo aleatorio de segundo orden que incorpora un sesgo de búsqueda α que utilizara información del nodo previamente visitado. La nueva probabilidad de transición no normalizada estará dada por $\pi_{vw} = \alpha_{tw} \cdot w_{vw}$ donde

$$\alpha_{tw} = \begin{cases} \frac{1}{p}, d_{tw} = 0 \\ 1, d_{tw} = 1 \\ \frac{1}{q}, d_{tw} = 2 \end{cases} \quad (5.12)$$

Aquí d_{tw} corresponde a la distancia entre el nodo t y el nodo w . Los parámetros p y q permiten guiar la búsqueda. Discutimos estos parámetros en la siguiente sección.

5.2.4 Hiperparámetros del modelo

A continuación listamos los hiperparámetros del modelo,

1. S : Controla el tamaño de la ventana.
2. L : Dimensionalidad de los *embeddings*.
3. K : Número de ejemplos negativos a mostrar a la red por cada ejemplo de entrenamiento.
4. r : Número de paseos aleatorios a generar por nodo.
5. l : Largo de los paseos aleatorios.
6. p : Parámetro que controla la probabilidad de retorno en un paseo.
7. q : Parámetro que controla el sesgo en un paseo hacia nodos dentro de la vecindad y hacia afuera de la vecindad.

Puesto que la idea de los paseos aleatorios es generar un equivalente a las frases en un texto para así entrenar el modelo *skip-gram*, se comparten todos los parámetros arquitecturales asociados. Para evitar posibles sesgos en los paseos generados para cada nodo se realizan r paseos aleatorios de largo l . El parámetro p es utilizado para controlar el retorno hacia un nodo. Si el paseo se encuentra en el nodo v habiendo llegado a este mediante el arco (t, v) , entonces con probabilidad $\frac{1}{p} \frac{w_{vt}}{Z}$ se retorna al nodo t . Asignar un valor pequeño a p incrementa la probabilidad de volver a un nodo ya visitado en el paseo y por tanto hacer la búsqueda más “local” en torno al nodo de inicio.

Por otro lado, el parámetro q permite sesgar la búsqueda entre nodos interiores y exteriores. Si el paseo llevó al nodo v pasando por el arco (t, v) , entonces con probabilidad $\frac{1}{q} \frac{w_{tv}}{Z}$ seleccionamos como siguiente nodo del paseo un vecino de v que no es vecino del nodo t y por tanto llevamos la búsqueda hacia afuera de la vecindad de t . Si q posee un valor pequeño entonces la probabilidad de seleccionar un nodo exterior a la vecindad de t aumenta y el paseo aleatorio se aproxima más a una estrategia del tipo DFS. En contrario si q posee un valor alto la búsqueda estará más sesgada hacia nodos cercanos a t y por ende el paseo se aproxima más a la estrategia BFS.

5.2.5 El algoritmo `node2vec`

Habiendo introducido los conceptos necesarios a continuación presentamos el pseudocódigo para el algoritmo `node2vec`.

Function `Node2vec`:

```

Input: Grafo  $G(V, E)$ , Dimensionalidad  $L$ , Paseos por nodo  $r$ , Largo del paseo  $l$ ,
Tamaño del contexto  $S$ , Parámetro de retorno  $p$ , Parámetro de sesgo interior-exterior  $q$ 
Output: Matriz embedding de tamaño  $|V| \times L$ 
// Se precomputan las probabilidades para los paseos aleatorios
 $\pi := \text{PreComputarProbs}(G, p, q)$  walks := []
for  $i$  in  $\text{range}(0, r)$  do
    walks := []
    for  $node \in V$  do
        // Iterar por nodos vecinos reduciendo el grado. Equivalente a
        recursion
        walk := PaseoAleatorio2doOrden( $G, node, \pi, l$ )
        walks.append(walk)
 $f := \text{SkipGramWord2vec}(\text{walks}, L, S)$ 
return  $f$ 

```

Algoritmo 2: Node2vec

Es importante ver que el algoritmo `node2vec` simplemente implementa una forma de transformar la red a una secuencia de paseos aleatorios que representan de algún modo frases y así construir una entrada compatible con la arquitectura *skip-gram* de `word2vec`. Notemos que una vez generadas las caminatas aleatorias simplemente se optimiza el modelo *skip-gram* utilizando como entradas los paseos.

5.3 Conectando word2vec y redes complejas

Puesto que el modelo poseerá dos embeddings es natural hacerse la pregunta si es que acaso está aprendiendo algún tipo de factorización matricial entre $\mathbf{W}$ y $\mathbf{C}$. Esta pregunta fue respondida satisfactoriamente por Goldberg y Levy en [30] donde probaron que entrenar la red utilizando el objetivo con muestreo negativo es equivalente a factorizar la matriz $\mathbf{M}$ definida como

$$\mathbf{M}_{ij} = \mathbf{W}_i \mathbf{C}_j = PMI(w_i, c_j) - \log k \quad (5.13)$$

donde $PMI(w, c) = \log \frac{\#(w, c) \cdot D}{\#w \cdot \#c}$. Intuitivamente $PMI(w, c)$ mide si las palabras w y c están correlacionadas computando la fracción de veces en que w y c aparecen juntas versus las apariciones totales de w y c en el texto. Assylbekov y Jangeldin [20] demostraron que modificando levemente la función objetivo removiendo la transformación sigmoideal se obtienen *embeddings* de calidad levemente inferior pero en donde empíricamente las redes obtenidas a partir de los embeddings poseen características de red compleja. En particular remover la transformación sigmoideal resulta en que el objetivo del modelo alcanza su óptimo cuando el producto entre los vectores palabra central y vectores palabra contexto satisface

$$\mathbf{C}_j^\top (\mathbf{W} \mathbf{e}_i) = \sigma [PMI(w_i, c_j) - \log k] \quad (5.14)$$

La función sigmoideal posee la propiedad de estar acotada en el rango $[0, 1]$ y los autores argumentan que es posible interpretar los valores como probabilidades de conexión entre las palabras en un grafo aleatorio. Al generar una red utilizando estas probabilidades el resultado es una grafo que exhibe características comunes a las redes complejas con lo que entrenando el modelo *word2vec skip-gram* podemos construir otro tipo de red compleja. Particularmente el nuevo objetivo no requiere modificar la arquitectura sino solo la función objetivo. La nueva función objetivo se muestra en la ecuación 5.15.

$$\arg \max_{\theta} \sum_{(w, c) \in D} \log (w \cdot c) - \sum_{(w, c') \in D'} \log (-w \cdot c') \quad (5.15)$$

Capítulo 6

Diseño experimental

Podemos resumir a grandes rasgos nuestros experimentos como el de construir redes sin y con pesos basadas en el principio de coocurrencia y el modelo *word2vec*, y comparar las características que ambas presentan de acuerdo a lo descrito en el Capítulo 4. A continuación describimos los distintos preprocesamientos y procedimientos realizados.

6.1 Textos y preprocesamiento

6.1.1 Textos a utilizar

Utilizamos 4 textos de acceso abierto provenientes de *Project Gutenberg* [31]. Los textos y sus respectivos autores se listan a continuación:

1. Alicia en el País de las Maravillas de Lewis Carroll (1865).
2. La Ilíada de Homero (750 A.C).
3. Pinocho de Carlo Collodi (1881).
4. Guerra y paz de León Tolstói (1867).

6.1.2 Preprocesamiento del texto

Inicialmente se removi  manualmente a todos los textos apartados relacionados a licencias para evitar su interferencia en el an lisis. Posteriormente se removieron caracteres especiales, *stopwords* y puntuaci n adem s de transformar el texto a min scula utilizando *NLTK* [32]. Para separar cada una de las palabras y as  obtener el texto como una lista de palabras se utilizo un *tokenizer* de *PyTorch* [33], m s espec ficamente, un *tokenizer* del tipo *basic_english*. Cabe destacar que algunos

textos poseen palabras en otros idiomas o en inglés arcaico con lo que cual nuestro preprocesamiento no es perfecto. El resultado de esta operación es una lista de palabras que denotamos por D para cada uno de los textos T .

6.2 Construcción de las redes

6.2.1 Construcción de la red de coocurrencia

Para obtener una representación del texto basada en coocurrencia seguimos el procedimiento estándar. Utilizando la lista de palabras preprocesadas D cada palabra única es representada en la red como un nodo y dos palabras estarán conectadas por un arco si coocurren en la lista con un tamaño de ventana $S = 5$. El resultado es una red sin pesos. Por otro lado en ciencia de redes resulta natural la idea de añadir pesos a las aristas o conexiones entre nodos a modo de representar la fortaleza de la relación. Existen distintas formas que podríamos pensar en añadir pesos, por ejemplo, en base a algún criterio relacionado a información mutua de las palabras como $\log \frac{P(w_1 \cap w_2)}{P(w_1)P(w_2)}$, sin embargo, tal criterio llevaría a redes con pesos demasiado similares puesto que aquella generada a partir de *word2vec* tiene pesos dados por $\sigma(\log \frac{P(w_1 \cap w_2)}{P(w_1)P(w_2)} - \log K)$ y ambas funciones σ y $\log$ son monótonas. Consecuentemente en este trabajo recurrimos a la forma estándar de asignar pesos en la literatura que corresponde a la frecuencia en que las palabras coocurren. Para incorporar esta idea creamos una segunda red con exactamente los mismos arcos pero en donde estos incorporan pesos asociados a la cantidad de veces que las palabras coocurren en la lista D .

Utilizamos la librería *NetworkX* [34] para el manejo de las redes.

6.2.2 Construcción de la red basada en el modelo *word2vec*

Como mencionamos en el Capítulo 5.3 es posible construir una red modificando levemente el objetivo del modelo *word2vec* e interpretando el producto entre vectores palabra como probabilidades de conexión.

Entrenamiento del modelo *word2vec*

Para obtener una representación basada en *embeddings* del texto utilizamos la arquitectura del modelo *word2vec* con muestreo negativo descrito en la sección 5.1 con las palabras preprocesadas. La función objetivo corresponderá a la ecuación 5.15. Para ser consistentes con la red de coocurrencia seleccionamos como tamaño de ventana $S = 5$. Puesto que para nuestro problema no es preocupación que el modelo presente *overfitting* seleccionamos *epochs* = 100. El resto de hiperparámetros los

seleccionamos en concordancia con la literatura, la dimensionalidad de los *embeddings* sera $L = 200$ y la cantidad de muestras negativas $K = 20$. El modelo es implementado usando *PyTorch* [33]. Los parámetros la red neural son actualizados utilizando *Adam* [35] con tasa de aprendizaje $lr = 0.001$.

Construcción de la red word2vec

Una vez entrenado el modelo la probabilidad de conexión entre las palabras i y j , σPMI_{ij} , es computada como:

$$\sigma PMI_{ij} = \mathbf{C}_j^\top \mathbf{W}_i$$

En donde $\mathbf{W}_i$ corresponde a la representación vectorial de la i -ésima palabra en $\mathbf{W}$ y $\mathbf{C}_j$ la representación vectorial de la j -ésima palabra en $\mathbf{C}$. Para obtener una red a partir de la matriz de probabilidades computamos la matriz de adyacencia usando muestreo aleatorio de los arcos como se muestra en el Algoritmo 3.

Function *MuestreoArcos*(σPMI):
Input: Matriz de probs. de conexion σPMI
Output: Matriz de adyacencia A
// Obtener cantidad de palabras
 $n_words := \text{len}(\sigma PMI)$;
for $i = 0; i < n_words; i++$ **do**
 for $j = i; j < n_words; j++$ **do**
 // Muestreo usando $\text{np.random.rand}()$
 if $\text{np.random.rand}() < \sigma PMI_{ij}$ **then**
 $A_{ij} := 1$;
 else
 $A_{ij} := 0$;
return A

Algoritmo 3: Muestreo aleatorio de arcos(σPMI)

A partir de la matriz de adyacencia es posible obtener el grafo/red. Similar al caso de coocurrencia construimos una segunda red con pesos. Si dos palabras i y j tienen un arco establecemos su peso como σPMI_{ij} .

6.3 Estudio de las redes

6.3.1 Variabilidad de la red word2vec

Puesto que el modelo presenta dos fuentes de variabilidad presentes en el entrenamiento de la red y en la generación de la matriz de adyacencia primero estudiamos el efecto de esta aleatoriedad en

las redes. Utilizando como base el texto “Alicia en el país de las maravillas” seguimos el siguiente procedimiento:

1. Dado un modelo *word2vec* entrenado M_0 generamos 5 matrices de adyacencia manteniendo fija la matriz de probabilidades de conexión y evaluamos estadísticamente para cada par si su distribución de grados es la misma utilizando una prueba de Kolmogorov-Smirnov.
2. Entrenamos 5 modelos *word2vec* produciendo 5 matrices de probabilidad de conexión distintos y en base a ellas 5 redes distintas (una por cada matriz), y evaluamos estadísticamente si la distribución de grados es la misma usando Kolmogorov-Smirnov.

Utilizamos un nivel de significancia $\alpha = 0.05$ y las siguientes hipótesis,

- $\mathcal{H}_0$: Ambos pares de muestras de grados provienen de la misma distribución.
- $\mathcal{H}_1$: Los pares de muestras de grados provienen de distribuciones distintas.

6.3.2 Comparando propiedades generales entre ambas redes

Estando en una representación común basada en grafos es posible comparar los diversos aspectos de cada red.

Primero determinaremos para ambas redes la cantidad de nodos no aislados n (es decir, nodos cuya distancia al resto de nodos es no infinita) y la cantidad de arcos m .

Posteriormente evaluamos si las redes poseen el efecto de mundo pequeño descrito en la sección 4.1.1. Para ello computamos el coeficiente l definido en la ecuación 4.1 sobre los nodos no aislados y comparamos su magnitud con $\frac{\log n}{\log k}$. Para este propósito utilizamos la función *average_shortest_path_length(graph, weight=None)* de *NetworkX*.

Para evaluar la transitividad de las redes calculamos el coeficiente de *clustering* promedio C en el caso de redes sin pesos y el *clustering* promedio pesado $\tilde{C}$ en el caso de redes con pesos. También graficamos los coeficientes de *clustering* C y $\tilde{C}$ pesos en función de los grados. El calculo de estos coeficientes los realizamos con la función *average_clustering()*.

Para determinar si las redes poseen una distribución de grados libres de escala graficamos la distribución utilizando un *lineplot* de *seaborn* y colocamos los ejes en escala logarítmica.

Luego, para evaluar la asortatividad en las redes calculamos el grado promedio de los vecinos k_{nn} en el caso de redes sin pesos y k_{nn}^w en el caso de redes con pesos. Computamos los valores utilizando la función *average_neighbor_degree()* de *NetworkX* y graficamos en función del grado de los nodos utilizando un *lineplot*.

6.3.3 Comparando estructura de comunidad de las redes

Para extraer información respecto a la estructura de las redes primero aplicaremos una combinación de los métodos definidos en la sección 4.2 en un proceso de 3 pasos.

1. Primero particionamos las redes con y sin pesos generadas por coocurrencia y *word2vec* a comunidades utilizando el método de *Louvain*. Reportamos la modularidad y en el caso de redes sin pesos la modularidad promedio de una versión aleatorizada de la red y el z_{score} . La modularidad promedio aleatorizada se obtiene sobre 10 redes generadas usando la función *configuration_model()* de *NetworkX*.
2. Luego reducimos la cantidad de nodos en cada comunidad a aquel subconjunto más densamente conectando utilizando *k-core*. Para ello utilizamos la función *k_core* de *NetworkX*.
3. Finalmente tratamos de emparejar cada comunidad de la red de coocurrencia con una y única comunidad en la red *word2vec* maximizando la cantidad de nodos en la intersección. Para este propósito utilizaremos el índice de Jaccard.

Notemos que puesto que la cantidad de comunidades difiere entre las redes se asignaran únicamente un número de pares igual al mínimo entre la cantidad de comunidades. La situación descrita anteriormente es equivalente a resolver el problema de asignación en optimización combinatorial. Utilizaremos el método *linear_sum_assignment* de la librería *Scipy* para este propósito. Por otro lado, una de las características observadas en el modelo Word2vec es su capacidad de generar analogías lineales simples entre los vectores palabras. Por analogía nos referimos al uso que se le da en [36] respecto a capturar relaciones de semántica y morfología mediante operaciones vectoriales. Particularmente, si tenemos vectores representando a las palabras *king*, *queen*, *man* y *woman*, el modelo *word2vec* es capaz de capturar relaciones de la forma $king - man + woman \approx queen$ [37]. Del mismo modo otro fenómeno en la literatura es que $car - cars \approx apple - apples$ [36]. Claramente en un texto no esperamos observar que palabras morfológicamente similares ocurran mutuamente dentro de un pequeño contexto, por ejemplo, no esperamos observar ambas *god* y su plural *gods* con un tamaño de ventana 5. Sin embargo, creemos posible que este fenómeno tiene como consecuencia la creación de forma indirecta de relaciones que no ocurren explícitamente por concurrencia.

En función de las observaciones anteriores, procederemos del siguiente modo

1. Analizamos relaciones semánticas entre las palabras en la intersección de ambas comunidades y aquellas exclusivas a cada una.
2. Identificamos, en caso de existir, relaciones del tipo analogías y morfológicas entre palabras dentro de una misma comunidad.

3. Calculamos y analizamos la similaridad entre las comunidades de coocurrencia y Word2vec mediante la distribución de valores del índice de Jaccard.

6.3.4 Similaridad de arcos

En vista de las nuevas asociaciones que el modelo *word2vec* debería crear en función de analogías, utilizaremos el modelo *word2vec* preentrenado *Google news* de la librería *Gensim* para evaluar similaridad semántica de las palabras que forman arcos. Para ello extraemos aquellos arcos entre palabras que son exclusivos a cada red y para cada par computamos la similaridad coseno. Posteriormente graficamos la distribución de valores con un *histplot* de *Seaborn*.

6.3.5 Comparando informatividad de ambas redes

Para comparar la informatividad de ambas redes ejecutaremos el algoritmo *node2vec* sobre las redes sin pesos. Nos interesa determinar si es que acaso la red es lo suficientemente informativa para el algoritmo en términos de poder recuperar la estructura de comunidades en el espacio proyectado. Tomamos como parámetros aquellos recomendados por los autores, es decir, $S = 16$, $L = 128$, $K = 20$, $r = 20$, $l = 70$, $p = 1$ y $q = 1/2$. Nos apoyamos de la librería *node2vec* que contiene una implementación del algoritmo. Posteriormente ejecutaremos *k-means* sobre los *embeddings* con $k = \text{número de comunidades}$ en la red respectiva. Finalmente para evaluar qué tan similares son las comunidades con los *clusters* obtenidos por *node2vec* los emparejamos. Además, con los *clusters* reales obtenidos, comparamos con el resultado de emparejar con las comunidades de una versión aleatorizada de la red respectiva. Para el emparejamiento resolvemos el problema de asignación utilizando *linear_sum_assignment()* maximizando el índice de Jaccard. Las versiones aleatorizadas de las redes siguen la misma distribución de grados y son obtenidas utilizando la función *configuration_model()* de *NetworkX*.

Capítulo 7

Resultados

7.1 Resultados del preprocesamiento

La cantidad de palabras no únicas y únicas obtenidas luego de preprocesar cada uno de los textos se muestra en la tabla 7.1.

Texto	N° palabras únicas	N° palabras no únicas
Alicia en el país de las maravillas	2773	12733
La Ilíada	10176	90676
Pinocho	3463	19058
Guerra y Paz	20709	273889

Tabla 7.1: Cantidad de palabras luego de aplicar preprocesamiento

7.2 Resultados de variabilidad de redes generadas por medio de word2vec

7.2.1 Matriz de probabilidad fija

Los resultados correspondientes al test de Kolmogorov-Smirnov manteniendo la matriz de probabilidad de conexión entre palabras fija (es decir, a partir de un mismo modelo) se muestran en la Tabla 7.2.

Observamos que con un nivel de significancia $\alpha = 0.05$ no es posible rechazar la hipótesis de que las redes provienen de la misma distribución de grados.

Red	Red n° 2	Red n° 3	Red n° 4	Red n° 5
Red n° 1	$KS = 0.008/p = 1$	$KS = 0.015/p = 0.908$	$KS = 0.017/p = 0.801$	$KS = 0.012/p = 0.989$
Red n° 2	-	$KS = 0.010/p = 0.998$	$KS = 0.010/p = 0.998$	$KS = 0.011/p = 0.995$
Red n° 3	-	-	$KS = 0.011/p = 0.992$	$KS = 0.008/p = 0.999$
Red n° 4	-	-	-	$KS = 0.008/p = 0.999$

Tabla 7.2: Tabla de test de Kolmogorov-Smirnov generando matrices de adyacencia a partir de un mismo modelo entrenado

7.2.2 Matriz de probabilidad provenientes de modelos distintos

Los resultados correspondientes al test de Kolmogorov-Smirnov entrenando 5 modelos distintos y generando una red con la matriz de probabilidad de cada modelo se muestra en la Tabla 7.3.

Red	Red n° 2	Red n° 3	Red n° 4	Red n° 5
Red n° 1	$KS = 0.012/p = 0.973$	$KS = 0.013/p = 0.955$	$KS = 0.011/p = 0.992$	$KS = 0.018/p = 0.737$
Red n° 2	-	$KS = 0.015/p = 0.904$	$KS = 0.016/p = 0.840$	$KS = 0.016/p = 0.842$
Red n° 3	-	-	$KS = 0.011/p = 0.995$	$KS = 0.010/p = 0.998$
Red n° 4	-	-	-	$KS = 0.012/p = 0.986$

Tabla 7.3: Tabla de test de Kolmogorov-Smirnov generando matrices de adyacencia a partir de un mismo modelo entrenado

Nuevamente observamos que con un nivel de significancia $\alpha = 0.05$ no es posible rechazar la hipótesis de que las redes provienen de la misma distribución de grados incluso cuando provienen de distintos modelos entrenados.

Texto	Tipo de red	n	m	l	$\frac{\log n}{\log k}$	C	$\tilde{C}$
Alicia en el País de las maravillas	Co-ocurrencia	2773	47 712	2.45	2.24	0.57	0.003
	Basada en embedding	2768	26 057	2.98	2.70	0.25	0.160
La Iliada	Co-ocurrencia	10176	346 866	2.55	2.19	0.51	0.005
	Basada en embedding	10174	213 672	2.75	2.47	0.17	0.106
Pinocho	Co-ocurrencia	3463	72 039	2.46	2.19	0.54	0.009
	Basada en embedding	3459	38 269	2.92	2.63	0.23	0.148
Guerra y paz	Co-ocurrencia	20 709	894 811	2.55	2.23	0.57	0.001
	Basada en embedding	20 709	624 250	2.70	2.42	0.11	0.065

Tabla 7.4: Tabla de cantidad de nodos no aislados n , arcos m , efecto mundo pequeño l , *clustering* promedio sin pesos C y *clustering* promedio con pesos $\tilde{C}$.

7.3 Resultados de cantidad de nodos y arcos, efecto mundo pequeño y clustering promedio

Los resultados correspondientes a la cantidad de nodos y los coeficientes descritos en las secciones se muestran en la tabla 7.4. Es posible notar que con excepción de "Guerra y paz" en todo el resto de casos la cantidad de nodos no aislados es levemente mayor en el caso de redes generadas por coocurrencia que aquellas generadas por *embeddings*. Cabe notar que "Guerra y paz" corresponde al texto con mayor cantidad de palabras únicas y no únicas. Por otro lado, todas las redes evidencian el así llamado efecto de mundo pequeño, las distancias más cortas promedio l se encuentran en el mismo orden de magnitud que $\frac{\log n}{\log k}$, además, los valores de l se encuentran en el rango entre 2.45 y 3. Lo anterior se encuentra en línea con las distancias más cortas promedios reportadas para distintos tipos de redes en la literatura [23]. Además, el método utilizado para generar la red no parece influir en el valor de l pues los valores son similares independiente del tipo de red.

Luego en todos los casos la cantidad de arcos producida por la red obtenida a partir del modelo *word2vec* es menor que en el caso de redes producidas por coocurrencia observándose diferencias notables. Lo anterior se puede explicar producto del hecho de que para que dos palabras estén conectadas en el método *word2vec* la cantidad de veces que ambas coocurren debe ser significativa respecto a la cantidad total de veces que ambas aparecen en el texto. En el enfoque de construcción de redes basado en coocurrencia basta que una palabra w_1 esté en el contexto de otra w_2 para que ambas estén conectados en la red. Por otro lado en el enfoque basado en el modelo *word2vec* esto no garantiza que ambas palabras se conecten. Esto puesto a que como se vio en el Capítulo 5.3 la probabilidad de conexión entre dos palabras estará dada por $\sigma[PMI(w_i, c_j) - \log k] = \sigma[\log \frac{\#(w_i, c_j) \cdot D}{\#w_i \cdot \#c_j} - \log k]$ con lo cual para garantizar una alta probabilidad de conexión entre ambas palabras deben coocurrir una cantidad significativa de veces en el texto. Respecto al coeficiente de *clustering* promedio C observamos que en todos los casos el enfoque

basado en coocurrencia produce redes con una mayor transitividad en comparación al enfoque basado en *word2vec* observándose una gran diferencia en el caso de Guerra y paz. Podemos argumentar del mismo modo anterior y notar que si en el texto observamos una secuencia de palabras $w_{i-1} w_i w_{i+1}$ el enfoque basado en co-ocurrencia garantiza con un tamaño de ventana 2 que dado el hecho que w_i esta conectado a w_{i+1} y w_{i+1} esta conectado a w_{i-1} , w_i estará conectado a w_{i-1} . Por otro lado en el enfoque basado en *word2vec* si existe una conexión entre w_i y w_{i+1} y w_{i+1} esta conectado a w_{i-1} , para que w_i tenga una alta probabilidad de conectarse a w_{i-1} ambas palabras deben coocurrir una cantidad significativa de veces lo cual puede no ser cierto. Respecto al coeficiente de *clustering* promedio con pesos $\tilde{C}$ tenemos que en el caso de coocurrencia este disminuye notablemente lo cual nos indica que los triángulos que se forman en la red poseen pesos relativamente pequeños en comparación al máximo peso en la red lo cual indica que los triángulos son más frecuentes entre palabras que aparecen juntas pocas veces en los textos. La red generada mediante *word2vec* también ve una disminución en su *clustering* con pesos pero esta disminución no es tan notable como en el caso de coocurrencia. Para esta red es importante destacar un fenómeno particular, si es que acaso un triangulo es formado por palabras que aparecen una única vez en el texto entonces la probabilidad de conexión estará dada por $\sigma(\log D - \log K) \rightarrow 1$ cuando $D \rightarrow \infty$ con lo cual la red es más robusta con respecto a este índice.

7.3.1 Coeficiente de clustering en función del grado

Redes sin pesos

Los gráficos correspondientes al coeficiente de *clustering* en función del grado se muestran en la Tabla 7.5. Es posible observar que sin importar el tipo de construcción utilizada el patrón es el mismo pero a una escala distinta, a medida que aumenta el grado de los nodos el coeficiente de *clustering* promedio disminuye. Más particularmente en el caso de nodos de menor grado es posible decir que estos tienden a formar grupos más conectados con aquellos nodos vecinos. Por otro lado los nodos de mayor grado tienden a tener menor *clustering* y por ende sus vecinos no tienden tanto a conectarse entre sí. Lo anterior sugiere que en la red se forma una denominada estructura jerárquica en donde los nodos de mayor grado actúan como *hubs* o enlaces entre grupos de nodos fuertemente conectados entre si. Esta es una característica típica de redes complejas [18]. Finalmente se observa que en el caso de redes generadas por simple coocurrencia los nodos de menor grado tienen más transitividad que los nodos de menor grado en la red generada mediante el modelo *word2vec* y que por ende los triángulos entre palabras poco frecuentes son menos frecuentes en el caso del modelo *word2vec*.

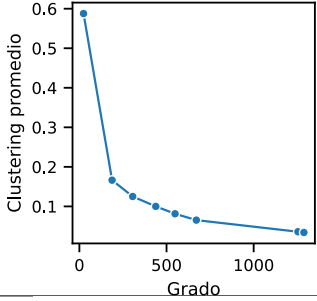
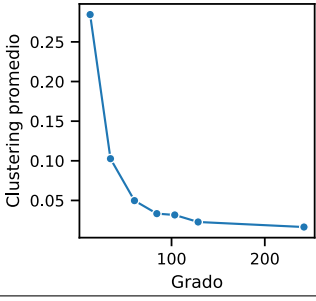
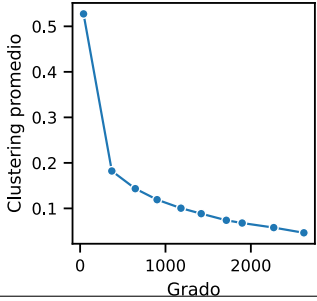
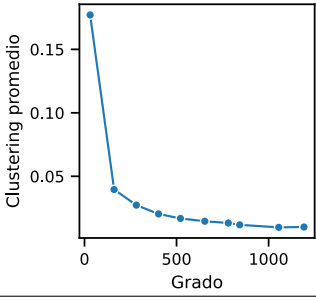
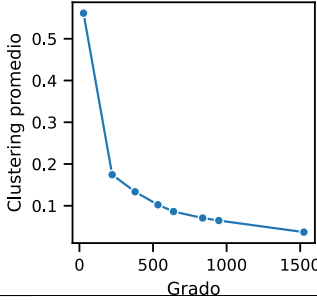
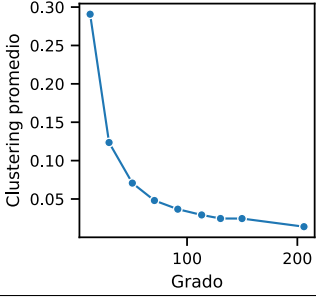
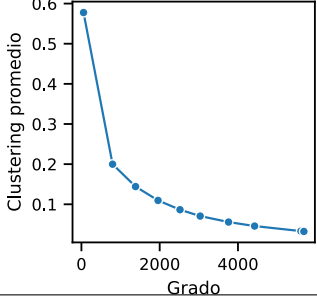
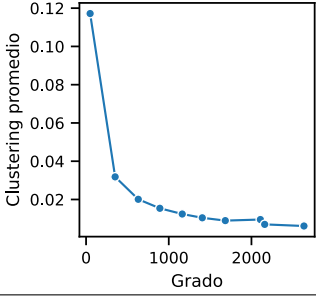
Clustering promedio coocurrencia	Clustering promedio <i>word2vec</i>	Descripción
		Distribuciones de <i>clustering</i> en función del grado de Alicia en el país de las maravillas (red sin pesos).
		Distribuciones de <i>clustering</i> en función del grado de La Ilíada (red sin pesos).
		Distribuciones de <i>clustering</i> en función del grado de Pinocho (red sin pesos).
		Distribuciones de <i>clustering</i> en función del grado de Guerra y Paz (red sin pesos).

Tabla 7.5: Tabla con gráficos de *clustering* promedio sin peso en función del grado para los distintos textos

Redes con pesos

Los gráficos de coeficiente de *clustering* con pesos en función del grado se muestran en las figuras de la Tabla 7.6. En todos los casos se repite el mismo patrón observado en las redes sin pesos pero a una menor escala, a medida que aumenta el grado el *clustering* disminuye. En el caso de redes generadas por coocurrencia se produce un menor *clustering* promedio con pesos en los nodos de menor grado que en el caso de la red *word2vec*. En el caso de la red de coocurrencia este patrón nos indica dos cosas, la primera de ellas es que las palabras poco frecuentes o con menor grado tienden a coocurrir frecuentemente con un conjunto pequeño de palabras. La segunda de ellas es que las palabras más frecuentes en el texto tienden a aparecer en contextos variados, es decir, coocurren con un conjunto significativamente más grande de palabras pero de forma menos frecuente. En el caso de *word2vec* el *clustering* de los nodos de menor grado disminuye pero no tanto como en el caso coocurrencia, esto puede deberse a que, como mencionamos previamente, existen palabras en los textos que aparecen una única vez en el texto y lo hacen de forma conjunta tal que se conectan con peso 1 lo cual corresponde al máximo posible en esta red. También es posible observar que el *clustering* $\tilde{C}$ llega a ser más cercano a 0 para nodos de grado alto que en el caso de coocurrencia. Debido al mecanismo por el cual asignamos los pesos esto también nos indica que las palabras más frecuentes están apareciendo en contextos variados con un gran conjunto de palabras pero muy pocas veces conjuntamente.

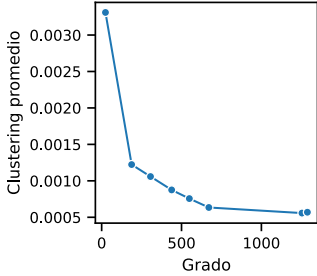
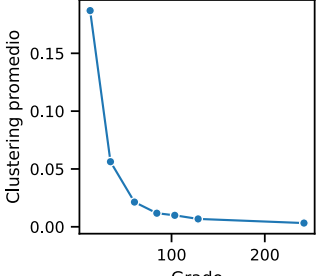
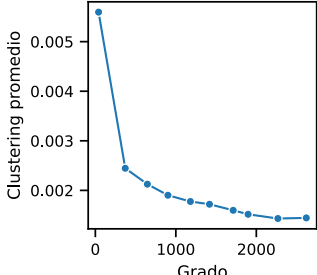
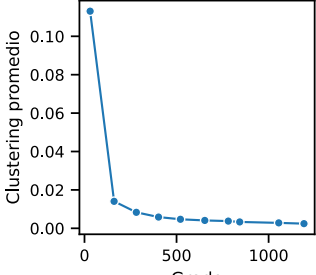
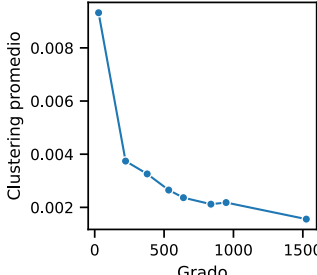
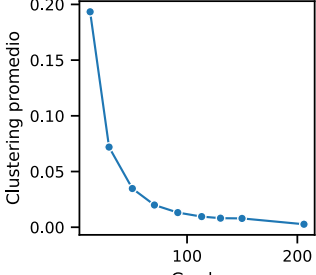
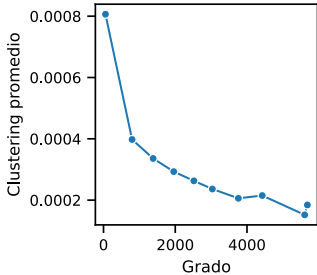
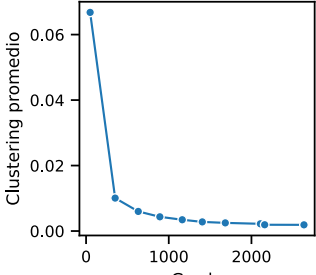
Clustering promedio con peso coocurrencia	Clustering promedio con peso <i>word2vec</i>	Descripción
		Distribuciones de <i>clustering</i> en función del grado de Alicia en el país de las maravillas (red sin pesos).
		Distribuciones de <i>clustering</i> en función del grado de La Ilíada (red con pesos).
		Distribuciones de <i>clustering</i> en función del grado de Pinocho (red con pesos).
		Distribuciones de <i>clustering</i> en función del grado de Guerra y Paz (red con pesos).

Tabla 7.6: Tabla con gráficos de *clustering* promedio con peso en función del grado para los distintos textos

7.4 Distribuciones de grado

Los gráficos de las distribuciones de grados se presentan en las figuras de la Tabla 7.7. Independiente del método utilizado es posible notar que las redes de texto no poseen una distribución de grados libre de escala. En caso de que la distribución fuese libre de escala esperaríamos observar en los gráficos un patrón aproximadamente lineal en donde la probabilidad o frecuencia de observar un nodo de grado k decae linealmente a medida que aumenta k . En los casos de redes generadas por coocurrencia se observa que los nodos de menor grado son más frecuentes en la red y en medida que el grado aumenta la frecuencia disminuye. Por otro lado, en el caso de redes generadas por medio de *word2vec* se observa una distribución con dos colas donde tanto nodos con alto grado y con bajo grado son infrecuentes en la red. También es posible observar que la distribución alcanza su máximo en nodos que tienen un grado medio a diferencia de las redes de coocurrencia que lo hacían en los nodos de menor grado. Este efecto puede explicarse debido a que el criterio bajo el cual dos palabras se conectan en la red *word2vec* es más estricto. En el caso de palabras poco frecuentes es probable que estas no coocurran un número de veces estadísticamente significativamente de veces lo cual provoca que ciertas conexiones que existían en la red de coocurrencia se corten al pasar a la red *word2vec*. Esto explica el hecho que las palabras con grados en el rango $[0 - 10]$ se encuentran más distribuidas en el caso de *word2vec*. Finalmente se puede observar que independiente del método de construcción o texto utilizado los nodos con un alto grado son infrecuentes (recordar que la función logaritmo es monótona).

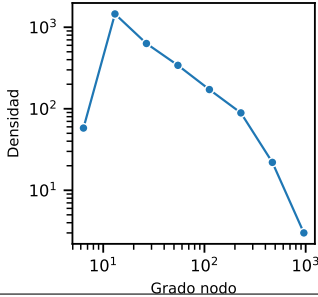
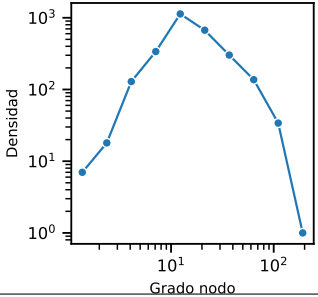
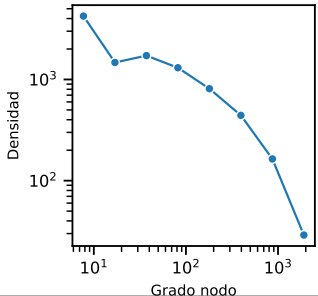
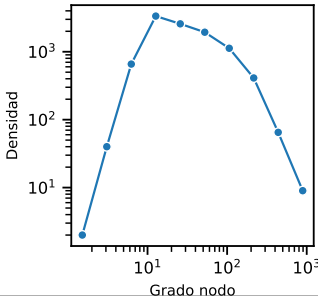
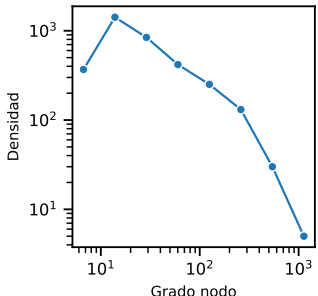
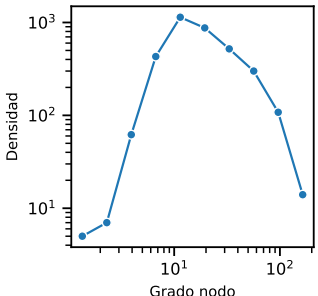
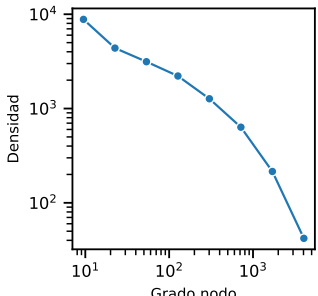
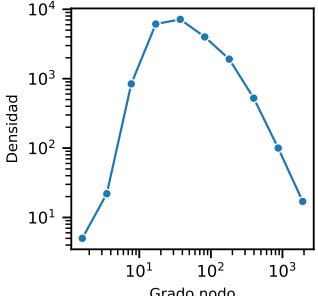
Grado coocurrencia	Grado <i>word2vec</i>	Descripción
		Distribuciones de grado de Alicia en el país de las maravillas.
		Distribuciones de grado de La Ilíada.
		Distribuciones de grado de Pinocho.
		Distribuciones de grado de Guerra y Paz.

Tabla 7.7: Tabla con gráficos de las distribuciones de grados para los distintos textos y tipos de red

7.5 Asortatividad

7.5.1 Redes sin pesos

Los gráficos correspondientes a los grados promedios de los nodos vecinos dado el grado del nodo se muestran en las figuras de la Tabla 7.8. En el caso de Alicia en el país de las maravillas es posible notar en el caso de coocurrencia un patrón fuerte de disortatividad en donde aquellos nodos con menor grado tienden a conectarse con nodos con un alto grado mientras que en el caso de los nodos alto grado tienen en promedio vecinos bajo grado. Este patrón no es evidente en el caso de la red *word2vec* y no es posible afirmar que sea asortativa o disortativa. Por otro lado, en el caso de La Ilíada hay un patrón disortativo tanto en caso de la red de co-ocurrencia como en aquella generada por el modelo Word2vec. En ambas redes los nodos de menor grado tienen en promedio nodos vecinos de alto grado, y viceversa. En el caso de Pinocho se evidencia un patrón claramente disortativo en el caso de la red de coocurrencia mientras que en el caso de la red *word2vec* este patrón no resulta tan evidente. Cabe destacar que tanto Pinocho como Alicia en el país de las maravillas son los textos menos extensos y con menor número de palabras por lo que está la posibilidad que estos patrones no emerjan en el modelo *word2vec* debido a la extensión. Finalmente, en el caso de Guerra y paz , que corresponde al texto más extenso, se evidencia en ambas redes que las palabras con mayor grado se conectan con palabras de menor grado, y viceversa. Independientemente del texto todas las redes generadas por medio de coocurrencia evidencian un claro patrón disortativo. Puesto a que como vimos las palabras menos conectadas en las redes tienden a tener mayor *clustering* y que las palabras más conectadas menor *clustering*, las palabras de mayor grado en las redes tienden a conectarse menos entre si prefiriendo conexiones con términos de menor grado. Lo anterior pareciera indicar que las palabras de alto grado actúan como puentes o *hubs* entre pequeños grupos de palabras con poco grado y alta conectividad entre si lo cual causaría el sesgo hacia un patrón disortativo en el gráfico de k_{nn} .

Asortatividad sin pesos coocurrencia	Asortatividad sin pesos <i>word2vec</i>	Descripción
		Distribuciones de assortatividad de Alicia en el país de las maravillas (red sin pesos).
		Distribuciones de assortatividad de La Ilíada (red sin pesos).
		Distribuciones de assortatividad de Pinocho (red sin pesos).
		Distribuciones de assortatividad de Guerra y Paz (red sin pesos).

Tabla 7.8: Tabla con gráficos de las distribuciones de assortatividad sin pesos para los distintos textos y tipos de red

7.5.2 Redes con pesos

Las distribuciones de asortatividad con peso en función del grado se muestran en las figuras de la Tabla 7.9. Es posible observar que al añadir pesos a la red el patrón de disortatividad observado en las redes de coocurrencia desaparece y las gráficas se vuelven relativamente estables. Puesto que los pesos corresponden a la frecuencia con que coocurren esto puede ser un efecto indeseado o debido simplemente a que las interacciones entre palabras son variadas y uniformes. En el caso de las redes generadas por el modelo *word2vec* nuevamente se observa que las redes de Alicia en el País de las Maravillas y Pinocho no tienen patrones disortativos muy marcados a diferencia de La Ilíada y Guerra y paz. Podemos observar que para los nodos de menor grado tanto en La Ilíada como en Guerra y paz el grado promedio de los nodos vecinos con peso k_{nn}^w es menor a su versión sin pesos k_{nn} . Pese a ello, debido a la forma en que se asignan los pesos de la red *word2vec*, el patrón disortativo nos advierte de un fenómeno curioso. Sabemos de la discusión previa que en ambas redes las palabras de alto grado tienden a conectarse más a palabras de bajo grado, sin embargo, al usar simplemente la frecuencia, ciertos patrones son perdidos. Sean w_1 , w_2 y w_3 palabras tal que w_1 poco frecuente, y w_2 y w_3 son altamente frecuentes. Consecuentemente, pese a que $freq(w_1 \wedge w_2)$ es bajo el grado de w_2 es alto por lo que el término $freq(w_1 \wedge w_2) \cdot k(w_2)$ debiese ser relativamente alto. Del mismo modo, la mayor parte de conexiones que tendrá w_2 será con nodos de grado bajo tal como podría ser w_1 . Por tanto, $freq(w_1 \wedge w_2) \cdot k(w_1)$ será también bajo, pero eso no es lo que observamos en promedio directamente en el gráfico. Esto se debe a que w_2 debe necesariamente tener unas pocas conexiones con palabras alta de frecuencia como w_3 tal que el término $freq(w_2 \wedge w_3) \cdot k(w_3)$ realiza una contribución significativa dominando el promedio del grado de vecinos y elevando el gráfico en el extremo derecho (nodos de alto grado) lo cual hace desaparecer la disortatividad. En ese sentido el enfoque basado en coocurrencia con asignación de pesos en base a frecuencias parece oscurecer la forma en que se asocian los nodos de la red. En este sentido la red *word2vec* podría parecer más robusta pues al usar pesos normalizados esta no sesga el gráfico debido a la aparición de unas pocas conexiones entre palabras de alto grado. En este sentido, un enfoque basado en coocurrencia pero con pesos asignados de acuerdo a *word2vec* podría ser una mejor opción al momento de analizar como se asocian las palabras.

Asortatividad con pesos coocurrencia	Asortatividad con pesos <i>word2vec</i>	Descripción
		Distribuciones de assortatividad de Alicia en el país de las maravillas (red con pesos).
		Distribuciones de assortatividad de La Ilíada (red con pesos).
		Distribuciones de assortatividad de Pinocho (red con pesos).
		Distribuciones de assortatividad de Guerra y Paz (red con pesos).

Tabla 7.9: Tabla con gráficos de las distribuciones de assortatividad con pesos para los distintos textos y tipos de red

Texto	Tipo de red	$N_{comunidades}$	Q	$\overline{Q}_{rand}$	z_{score}
Alicia en el País de las Maravillas	Co-ocurrencia	14	0.22	0.13	60.3
	Basada en embedding	20	0.35	0.20	87.3
La Ilíada	Co-ocurrencia	9	0.16	0.09	136.7
	Basada en embedding	11	0.24	0.12	61.9
Pinocho	Co-ocurrencia	12	0.19	0.11	35.58
	Basada en embedding	18	0.31	0.18	84.07
Guerra y paz	Co-ocurrencia	12	0.17	0.07	108.6
	Basada en embedding	7	0.24	0.10	172.2

Tabla 7.10: Tabla con el número de comunidades, modularidad, modularidad de redes aleatorizadas y z_{score} para los textos considerando redes sin pesos

7.6 Resultados de estructura de comunidades

7.6.1 Redes sin pesos

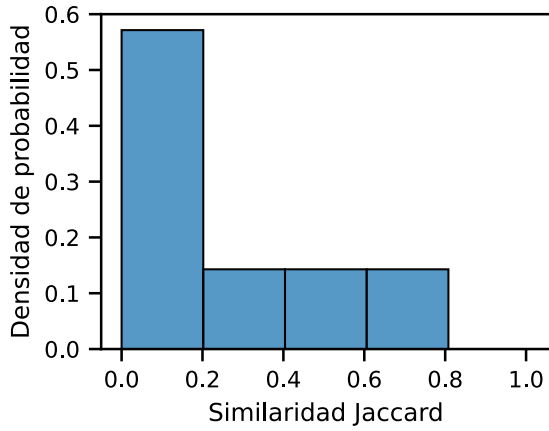
La cantidad de comunidades luego de ejecutar el método de Louvain se muestra en la tabla 7.10. Consistentemente ocurre que en el caso de los texto más cortos el método basado en co-ocurrencia genera un número menor de comunidades que el método basado en Word2vec mientras que en el caso de los textos más largos Word2vec tiende a generar menos comunidades. Por otro lado, en todos los textos ocurre que las redes generadas por el modelo Word2vec poseen mayor modularidad que las redes generadas por co-ocurrencia. Además, en todos los casos excepto en La Ilíada ocurre que al comparar la modularidad de la red contra su versión aleatorizada usando el z_{score} obtenemos que las redes Word2vec se desvían más del valor esperado de modularidad que las redes de co-ocurrencia. Lo anterior resulta sorprendente considerando que utilizamos un grafo aleatorio para generar el modelo basado en Word2vec. Además, la alta modularidad obtenida en las redes generadas por el modelo Word2vec indican que las estructuras de comunidad se encuentra mejor definidas que en el caso de co-ocurrencia. Por otro lado, la diferencia con La Iliada puede deberse simplemente a la aleatoriedad del método de Louvain que nos entregó un mal particionamiento. En el caso de Alicia en el país de las maravillas y Pinocho la cantidad de comunidades generadas por co-ocurrencia es menor que en el caso de Word2vec, puesto que estos textos son más cortos podemos explicar este fenómeno por el hecho que la cantidad de veces que co-ocurren palabras puede ser muy baja como para que la probabilidad de conexión entre palabras el método basado en Word2vec sea alta. Como consecuencia de lo anterior esperaríamos ver que los nodos de la red se encuentran más aislados generando así más comunidades. Por otro lado en el caso de La Ilíada y de Guerra y paz que corresponden a los textos más largos existe una menor cantidad de comunidades que en el caso de co-ocurrencia, como estos textos son más largos podríamos intuir que la cantidad de veces que las palabras co-ocurren es más alta generando una mayor probabilidad de conexión y por ende un grafo

más denso con comunidades menos definidas lo cual reduce su número. Lo anterior es confirmado por el hecho de que estas redes tienen menos modularidad que en el caso de los textos más cortos.

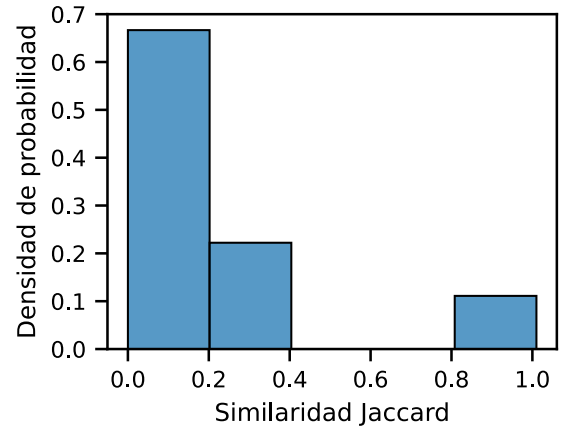
En los apéndices A.1, A.2, A.3 y A.4 se muestran las tablas generadas por las asociaciones entre comunidades reducidas a su *k-core*.

Distribución de valores de Jaccard para emparejamientos

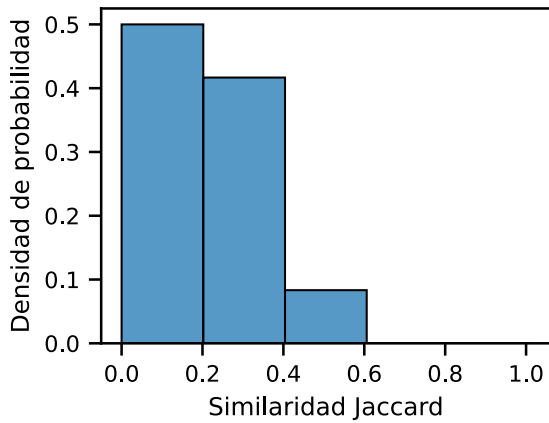
Las distribuciones de valores de Jaccard para los emparejamientos de los *k-core* de las comunidades se muestran en las figuras 7.1a, 7.1b, 7.1c y 7.1d. Notemos que para Alicia en el país de las maravillas un poco menos del 60% de los emparejamientos se encuentran en el rango $[0 - 0.2[$ lo cual indica un bajo nivel de similaridad, luego unos pocos conjuntos (alrededor del 40%) tienen valores levemente más altos de similaridad en el rango $[0.2 - 0.8]$. Particularmente un 10% (es decir, 1) de los emparejamientos se encuentran en el rango $[0.6 - 0.8]$. En el caso de La Ilíada cerca del 70% de las comunidades tienen bajos niveles de similaridad en el rango $[0.0 - 0.2[$ mientras que solo un 1 emparejamiento logra una similaridad alta en el rango $[0.8 - 1.0]$. En el caso Pinocho el 50% de los emparejamientos tiene valores bajos mientras que el 40% se encuentra en el intervalo $[0.2 - 0.4]$. El caso de Guerra y paz es similar al de Pinocho. En general se observa que la mayor parte de las comunidades emparejadas tienen bajos valores de similaridad Jaccard. Esto indica que la proporción de nodos o palabras comunes entre los *k-core* de ambas comunidades es relativamente baja respecto a la unión entre las palabras en ambos grafos. Lo anterior pareciera indicar que las comunidades de palabras que se forman entre ambos enfoques son estructuralmente distintas. Cabe destacar que el índice de Jaccard es una métrica relativamente estricta y que existen otras métricas como el *Overlap coefficient* que son más permisivas.



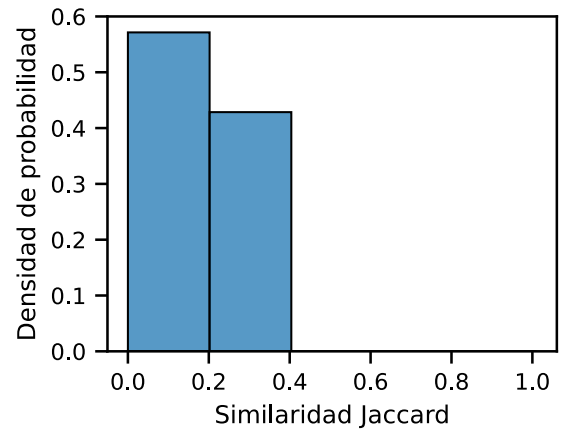
(a) Alicia - Distribución de valores de Jaccard emparejamiento redes sin pesos



(b) La Ilíada- Distribución de valores de Jaccard emparejamiento redes sin pesos



(c) Pinocho - Distribución de valores de Jaccard emparejamiento redes sin pesos



(d) Guerra y paz - Distribución de valores de Jaccard emparejamiento redes sin pesos

Figura 7.1: Distribución de valores de Jaccard para emparejamientos de k -core de comunidades de redes sin pesos

Análisis de similaridad semántica

En el caso de Alicia en el país de las maravillas se observa un alto grado de similaridad Jaccard para el caso de las asociaciones en las Tablas A.3, A.6, A.11 y A.12 con valores de 0.467, 0.806, 0.55 y 0.8 respectivamente. En la tabla A.3 la intersección contiene varias palabras que derivan a partir de *mouse* como *mouse-to* y *mouse-a*, además, ambas redes asociaron juntas las palabras *latin* y *grammar*. La Tabla A.6 presenta la mayor similaridad Jaccard del texto, sin embargo, es posible apreciar que los distintos términos en la intersección parecieran no guardar ninguna relación semántica. En la tabla A.11 se pueden apreciar palabras relacionadas a animales y comida como *owl*, *panther*, *meat* y *gravy*. Finalmente, ocurre curiosamente que en la Tabla A.12 pareciera que queda descrita una secuencia donde un personaje saca una jarra de mermelada de naranja de una despensa para encontrarla vacía. Cabe destacar también que la asociación descrita en la Tabla A.9 presenta un valor más bajo de 0.246, sin embargo, el subconjunto de palabras en la intersección describe un cierto panorama político y geográfico incluyendo palabras como *earls* (miembros de la nobleza), *morcar*, *mercias*, *northumbria* y *edwin*. Edwin es un personaje histórico cuyo hermano Morcar poseía el título de Earl de Northumbria, un reino inglés que tenía de vecino al reino de Mercia. Por otro lado, es posible observar que las palabras en la intersección de las redes k-core resultan ser palabras densamente conectadas presentando un grado alto en sus respectivos k-core. En el caso de La Ilíada en general se obtienen valores de similaridad menores que en Alicia en el país de las maravillas llegando incluso a tener asociaciones con similaridad 0 como en Tabla A.18. Las asociaciones con mayor similaridad corresponden a aquellas en las Tablas A.17, A.19, A.21 y A.23 con valores de 0.193, 0.275, 0.275 y 0.810 respectivamente. La Tabla A.17 presenta a los distintos héroes de la guerra de Troya y las fuerzas en conflicto incluyendo palabras como *hector*, *achilles*, *ajax*, *patroclus*, *greeks* y *trojans*. En la Tabla A.19 se incluyen términos relacionados a las extremidades de los personajes e interacciones con estas incluyendo palabras como *head*, *hand*, *arm*, *breast*, *lance* y *stood*. En la asociación mostrada en la Tabla A.21 se aprecian distintas palabras típicas del inglés arcaico como *thy*, *shall*, *thou* y *thee*. Finalmente en la Tabla A.23 se aprecia una cuestión teórica que vale la pena destacar. Al final de La Ilíada se incluye una cita de M. Aurel Anton la cual es “Ton theon de eupoiia—to mae epi pleon me procophai en poiaetiki kai allois epitaeoeimasi en ois isos a kateschethaen, ei aesthomaen emautan euodos proionta.”, como se mencionó previamente la probabilidad de que dos palabras se conecten en la red generada por el modelo Word2vec depende de la proporción de veces que las palabras aparecen juntas en comparación con el número total de veces que aparecen individualmente. Puesto que las palabras en la cita ocurren una única vez en el texto y además lo hacen juntas existe una probabilidad que garantiza que casi seguramente estas palabras adyacentes estarán conectadas lo cual tiene como consecuencia que se preserve esta comunidad en ambas redes. Este pequeño análisis puede extenderse un poco más, si un texto está compuesto de palabras que ocurren una única vez entonces el modelo Word2vec “aprenderá”

exactamente la red de co-ocurrencia. En el caso de Pinocho las asociaciones más similares son aquellas en las Tablas A.24, A.28, A.29, A.33 y A.34. La asociación en la Tabla A.24 presenta palabras relacionadas a juegos y juguetes como *marbles*, *played*, *bicycles*, *hopscotch* y *ball*. La Tabla A.29 presenta palabras asociadas a prendas y la composición de estas incluyendo palabras como *coat*, *silk*, *velvet*, *tail*, *waist* y *slippers*. En la Tabla A.33 se aprecian términos relacionados a comida y animales como *hare*, *dozen*, *chickens*, *partridges* y *tender*. Finalmente, la Tabla A.34 contiene distintos términos asociados a la nobleza como *courts*, *europe*, *queens*, *emperors* y *kings*. Por otro lado la asociación en la Tabla A.26 presenta un valor más bajo de similaridad Jaccard pero en la intersección se encuentran palabras que son semánticamente similares y relacionadas a las extremidades de los personajes incluyendo palabras como *hands*, *legs*, *mouth*, *feet* y *neck*. Respecto al caso de Guerra y Paz únicamente dos asociaciones destacan entre las siete, estas corresponden a las Tablas A.36 y A.40 con valores de similaridad de 0.347 y 0.111 respectivamente. La Tabla A.40 muestra un fenómeno curioso, las palabras en la intersección se encuentran bastante más conectadas en el caso de la red de co-ocurrencia que en el caso de la red generada con el modelo Word2vec. En el caso de la palabra *face* esta se encuentra conectada en el k-core de co-ocurrencia con otras 279 palabras mientras que en el caso de la red k-core de Word2vec con apenas 40. También es posible notar que esta asociación describe partes de los personajes incluyendo términos como *head*, *face* y *eyes*. Respecto a la asociación de la Tabla A.40 esta posee un valor más bajo de similaridad pero la intersección entre ambas comunidades es cohesiva semánticamente incluyendo términos militares como *army*, *french*, *moscow*, *napoleon* y *general*. Nuevamente es posible notar que las palabras están más conectadas en el caso de la red de co-ocurrencia que en el caso del modelo k-core. Cabe destacar que Guerra y paz es el texto más largo y con el mayor número de palabras utilizado y por ende puede haber ocurrido que el tamaño del embedding haya sido demasiado pequeño para capturar todas las relaciones.

El hecho de que a pesar de la baja similaridad Jaccard exista relación semántica entre las palabras de la intersección podría deberse a lo estricta de la métrica.

Análisis de analogías simples

Es posible observar en el caso de Alicia en el País de las maravillas que aparentemente la red de coocurrencia no preserva bien relaciones sintácticas. Las palabras *go* y *went* representan ambas la forma presente y pasada de un mismo verbo, sin embargo, *go* queda en la comunidad de la Tabla A.2 y *went* en la Tabla A.1. Lo mismo ocurre con palabras como *think* y *though* que quedan en las comunidades A.4 y A.2 respectivamente. La peor ofensa, sin embargo, es con *say*, *saying* y *said* que quedan en las comunidades A.2, A.5 y A.7. Por otro lado en el caso de Word2vec es posible apreciar que las palabra *beautiful* y sus variantes guionizadas *beauti—ful* y *beau—ootiful* quedan en la comunidad A.14. Lo mismo se aprecia en la Tabla A.10 en donde la red Word2vec captura las

analogías entre *sea* y *seaography*, y entre *drawlingmaster* y *drawling-the*. También cabe destacar que ambos métodos logran capturar analogías derivadas de la palabra *mouse* como se aprecia en la Tabla A.3.

En el caso de La Iliada nuevamente se aprecian analogías superiores en el caso de la red Word2vec. La palabra *trojan* y su plural *trojans* quedan en las comunidades A.18 y A.21. Lo mismo ocurre con *god* y *gods* que quedan en las comunidades A.21 y A.22 en el caso de coocurrencia mientras que en Word2vec quedan ambas en la comunidad A.21. Similarmente las palabras *plain* y *plains* quedan ambas en la comunidad A.22 para la red Word2vec.

Para Pinocho podemos notar que en el caso de Word2vec en la comunidad A.24 se encuentran las palabras *shouting*, *bowling* y *defeaning* y que, por ejemplo, $shouting - shout + bowl \approx bowling$ pese a que *bowl* no se encuentra en el $k - core$. Además, es posible observar una analogía en la comunidad para el caso de A.28 pues $fire + wood \approx wood$ y $firewood - wood \approx fire$. A pesar de ello ni la red de coocurrencia ni la red Word2vec logran dejar *pinoccio* junto con su posesivo *pinoccio's* en la comunidad A.27.

Para Guerra y Paz en el caso de Word2vec destaca la comunidad A.41 que posee las palabras *know* y *knew* que corresponden al mismo verbo en distintos tiempos, y la comunidad A.42 que contiene a *feeling* y su plural *feelings*.

Claramente no esperamos que todas las analogías presentes en el modelo Word2vec se trasladen directamente a la red, únicamente aquellas donde $v_{w_1} \approx v_{w_2} - v_{w_3}$ y los vectores producto del entrenamiento resultan cercanos entre. Puesto que estamos analizando únicamente los $k - core$ de cada comunidad y sus palabras únicas, cabe destacar que analogías presentes en ambas redes y poco densas no quedan dentro del análisis. Por otro lado, la más alta modularidad de las redes Word2vec la asociamos de dos factores. Primero, a que debido a la forma en que se construye la red se eliminan asociaciones que simplemente corresponden a ruido, concentrándose entre asociaciones que son estadísticamente significativas, y segundo a que las analogías entre palabras que está aprendiendo el modelo Word2vec se traducen, en algunos casos, a asociaciones entre nodos en la red.

Reflexiones respecto a la estructura de comunidades

Cuando uno analiza más en detalle la forma que tienen las comunidades considerando todos los nodos que la componen y no únicamente el $k - core$, se aprecian ciertas diferencias entre ambos enfoques de construcción. La asociación de comunidades que se muestra en la Tabla A.11 se centra principalmente en el Capítulo 10 de Alicia en el País de las Maravillas. En este capítulo el Grifo lleva a Alicia a conocer a la Falsa Tortuga. Durante su conversación Alicia relata el poema “Tis the Voice of the Lobster”. El poema relata un banquete en el que participan un búho y una pantera. Es por ello que observamos distintas palabras en ambas comunidades relacionadas a comida como

Texto	Tipo de red	$N_{comunidades}$	Q
Alicia en el país de las maravillas	Co-ocurrencia	12	0.21
	Basada en embedding	29	0.47
La Ilíada	Co-ocurrencia	10	0.18
	Basada en embedding	17	0.31
Pinocho	Co-ocurrencia	10	0.19
	Basada en embedding	27	0.40
Guerra y paz	Co-ocurrencia	14	0.17
	Basada en embedding	12	0.31

Tabla 7.11: Tabla con el número de comunidades y modularidad para los textos considerando redes con pesos

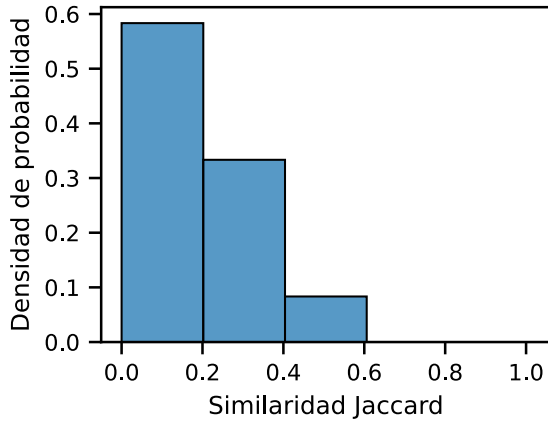
knife, *fork*, *pie* y *piecrust* además de los animales *owl* y *panther*. Sin embargo, también es posible notar que la red *word2vec* también agrega términos como *walrus* e *hippopotamus* que aparecen una única vez en el texto y en el capítulo 2. Esto creemos es debido a que la red esta creando nuevas relaciones basadas en semántica que no aparecen directamente en el flujo del texto. Por otro lado en la asociación de comunidades que se muestra en la Tabla A.14 aparecen distintos términos relacionados al capítulo 6 “Cerdo y pimienta” de Alicia en el País de las Maravillas. En el se relata como Alicia llega a la casa de la Duquesa en donde su criada está preparando una sopa con tanta pimienta que el bebé de la Duquesa estornudaba. En este capítulo además se introduce el personaje del gato de Cheshire. Particularmente, se agregan también a la comunidad palabras en el capítulo 9 debido al uso de palabras como *soup* y *chorus* también presentes en este. Sin embargo, en este caso particular resulta interesante notar que las palabras que son esenciales a la narrativa de este capítulo no aparecen en el caso de la red *word2vec* como lo son *baby*, *duchess* y *sneeze* y *cheshire*, el gato de la Duquesa.

7.6.2 Redes con pesos

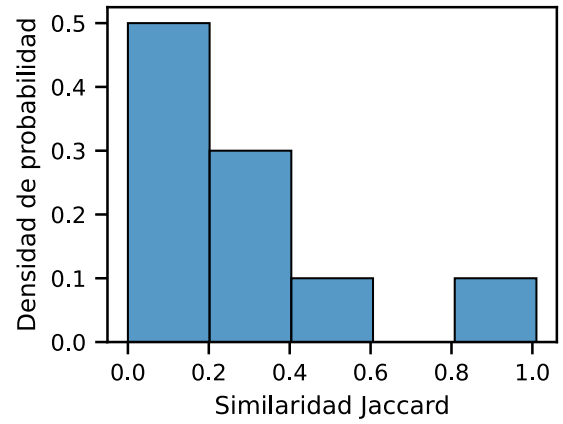
La cantidad de comunidades y modularidad luego de ejecutar el método de *Louvain* sobre las redes con peso se muestran en la tabla 7.11. En este caso consistentemente ocurre que la redes generadas por el método *word2vec* presentan mayores valores de modularidad que en el caso de coocurrencia, además, con la excepción de Guerra y paz, el método de coocurrencia genera menos comunidades. En los apéndices A.1, A.2, A.3 y A.4 se muestran las tablas generadas por las asociaciones entre comunidades reducidas a su *k-core*.

Distribución de valores de Jaccard para emparejamientos

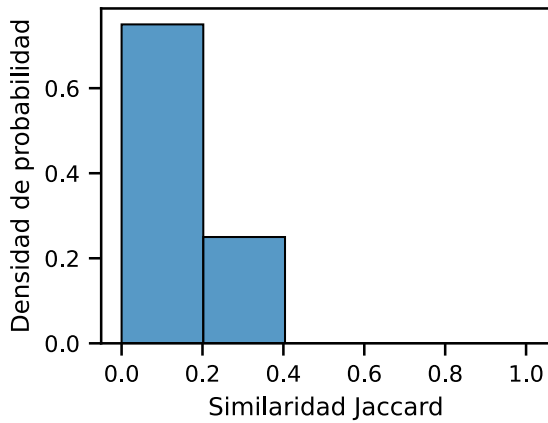
Las distribuciones de valores de Jaccard para los emparejamientos de los *k-core* de las comunidades se muestran en las figuras 7.2a, 7.2b, 7.2c y 7.2d. En el caso de Alicia el emparejamiento en el



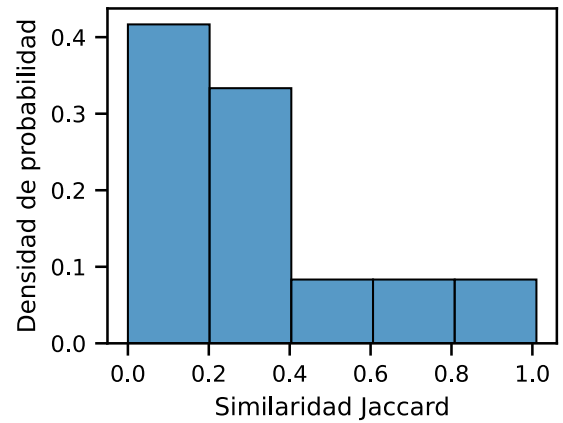
(a) Alicia - Distribución de valores de Jaccard emparejamiento redes sin pesos



(b) La Iliada- Distribución de valores de Jaccard emparejamiento redes con pesos



(c) Pinocho - Distribución de valores de Jaccard emparejamiento redes con pesos



(d) Guerra y paz - Distribución de valores de Jaccard emparejamiento redes sin pesos

Figura 7.2: Distribución de valores de Jaccard para emparejamientos de k -core de comunidades de redes con pesos

rango $[0.6 - 0.8]$ que existía en redes sin pesos desaparece. Para La Ilíada el emparejamiento que existía en el rango $[0.8 - 1.0]$ en el caso sin pesos se preserva al añadir pesos a la red. Para Pinocho la cantidad de emparejamientos de baja similaridad aumenta desde alrededor de un 50% a más de un 60%. En el caso de Guerra y paz se observa un incremento en las comunidades con un alto grado de similaridad concentrándose alrededor de un 30% en el rango $[0.4 - 1.0]$. También se redujo la cantidad de emparejamientos con bajo nivel de similaridad en el rango $[0.0 - 0.2]$ desde un 60% a un 40%.

Análisis de similaridad semántica

Para el caso de Alicia en el país de las maravillas los emparejamientos con mayor similaridad corresponden a aquellos en las Tablas B.4, B.5 y B.7 con valores de similaridad de 0.329, 0.467 y 0.256 respectivamente. En la Tabla B.4 es posible observar en la intersección diversos animales como *turtle*, *gryphon* y *lobsters*, además de términos relacionados al agua como *sea* y *wash*. Al igual que en el caso de las redes sin pesos en la Tabla B.5 aparecen términos derivados de la palabra *mouse* como *mouse-of* y *mouse-to*. Por otro lado en la asociación mostrada en la Tabla B.7 pese a haber un valor de similaridad medio no se observa relación aparente entre los conceptos. Cabe destacar que la asociación relacionada al panorama político sigue existiendo y se muestra en la Tabla B.8 con un valor de 0.182 y sigue manteniendo términos como *edwin*, *earls* y *mercia*.

Respecto a la Ilíada los emparejamientos con mayor similaridad corresponden a aquellos en las Tablas B.17, B.20 y B.22 con valores de 0.4, 0.5 y 0.81 respectivamente. En la asociación de la Tabla B.17 se observan múltiples términos del inglés arcaico como *thee*, *thy*, *thou* y *jove*. En la Tabla B.20 la intersección muestra diversos personajes menores de la mitología griega como *Evippus* e *Iphis*. Sorprendentemente estos nos quedan juntos con los héroes. Cabe destacar que al igual que en el caso de redes sin pesos existe un emparejamiento que muestra a los distintos héroes de la guerra de Troya y este se muestra en la Tabla B.13. Pese a tener una valor de similaridad relativamente bajo de alrededor de 0.101 incluye nombres como los de *hector*, *ajax* y *achilles*. Finalmente la Tabla B.22 presenta una comunidad correspondiente a la cita de M. Aurel Anton que también se observa con un alta similaridad en el caso de redes sin pesos.

Para Pinocho únicamente destaca el emparejamiento de la Tabla B.26 con un valor de similaridad de 0.349, en el resto de emparejamientos no se observa relación semántica aparente en las palabras de la intersección. La intersección de la Tabla B.26 muestra términos relacionados a fechas como *sunday*, *saturdays*, *vacation* y *january*.

El caso de Guerra y paz resulta más interesante; los emparejamientos con mayor similaridad corresponden a aquellos en las Tablas B.33, B.34, B.37, B.39, B.43 y B.44 con valores de similaridad 0.321, 0.255, 1, 0.647 y 0.295. El emparejamiento de la Tabla B.33 muestra diferentes palabras con faltas ortograficas como *wuined*, *bettah* y *conscwription*. Las palabras de la intersección mostradas

en la Tabla B.34 no parecen tener una relación semántica aparente pese al alto valor del índice de Jaccard. La asociación en la Tabla B.37 también aparece en el caso de redes sin pesos y muestra diferentes términos militares como *french*, *army*, *russian*, *battle* y *troops*. La Tabla B.39 muestra las palabras de la frase en Latín *Sic transit gloria mundi*. Sorprendentemente esta comunidad es exactamente la misma en ambas redes y no se incorporan palabras externas con lo cual se alcanza un valor de similaridad 1. Respecto a la Tabla B.43 esta presenta diferentes palabras del Alemán como *den*, *gewiss* y *nicht*. Finalmente, el emparejamiento de la Tabla B.44 muestra palabras relacionadas a descripciones de los personajes como *eyes*, *face*, *head* y *round*. Puesto que Guerra y paz corresponde al texto más largo teorizamos que estos patrones de alta similaridad semántica que emergen dependen del largo del corpus.

7.7 Distribuciones de similaridad

Las distribuciones de similaridad para Alicia en el País de las Maravillas, La Ilíada, Pinocho y Guerra y Paz se muestran en las Tablas 7.12, 7.13, 7.14 y 7.15. Observamos en todos los casos que las redes construidas mediante *word2vec* poseen menos arcos de baja similaridad. Lo anterior es consistente con que el modelo aprende relaciones estadísticamente significativas entre palabras. También es posible notar en el caso de Alicia en el País de las maravillas una disminución en los arcos de alta similaridad mientras que en La Ilíada y Pinocho la proporción permanece relativamente constante entre ambos métodos de construcción. Por otro lado, en el caso de Guerra y Paz es posible observar que la red *word2vec* posee casi el doble de arcos de alta similaridad que en el caso de coocurrencia. Puesto que estamos analizando los arcos que son únicos a cada una de las construcciones es posible concluir que el enfoque *word2vec* esta construyendo nuevas asociaciones de alta similaridad que no aparecen explícitamente por simple coocurrencia. Sin embargo, también ciertas relaciones de alta similaridad son eliminadas. Esto puede deberse a que las palabras no aparecen juntas una cantidad estadísticamente significativa de veces. Este efecto parece ser menor en el caso de Guerra y Paz que corresponde al texto más extenso y que por tanto provee una mayor cantidad de ejemplos al modelo. Algunos ejemplos de arcos en Guerra y Paz que aparecen en la red *word2vec* y no en coocurrencia son (*commensurable*, *reasonable*), (*occupation*, *position*), (*grumbled*, *quiet*), (*doctrine*, *religious*), (*decisive*, *experienced*), (*innocence*, *experienced*). Lo anterior da claros indicios acerca del fenómeno que está ocurriendo: el modelo *word2vec* aprende nuevas relaciones entre palabras de acuerdo a un uso similar en diferentes contextos, esto crea nuevas conexiones directas entre palabras que no aparecen directamente por coocurrencia. Es por ello que creemos que el modelo *word2vec* puede ser utilizado para analizar relaciones semánticas que se producen de acuerdo al uso que distintos autores dan a las palabras.

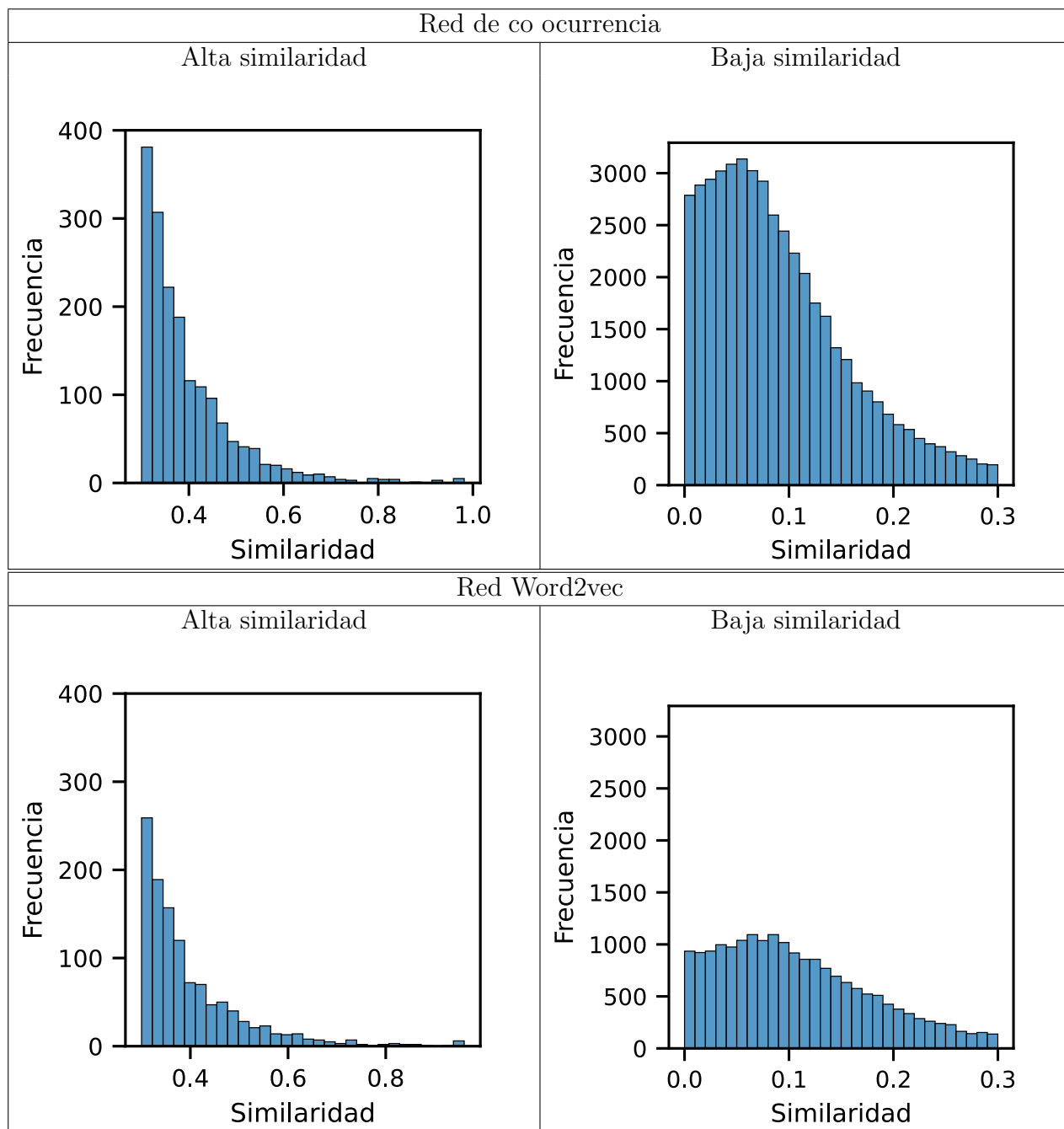


Tabla 7.12: Comparación de similitud de arcos de Alicia en el país de las Maravillas

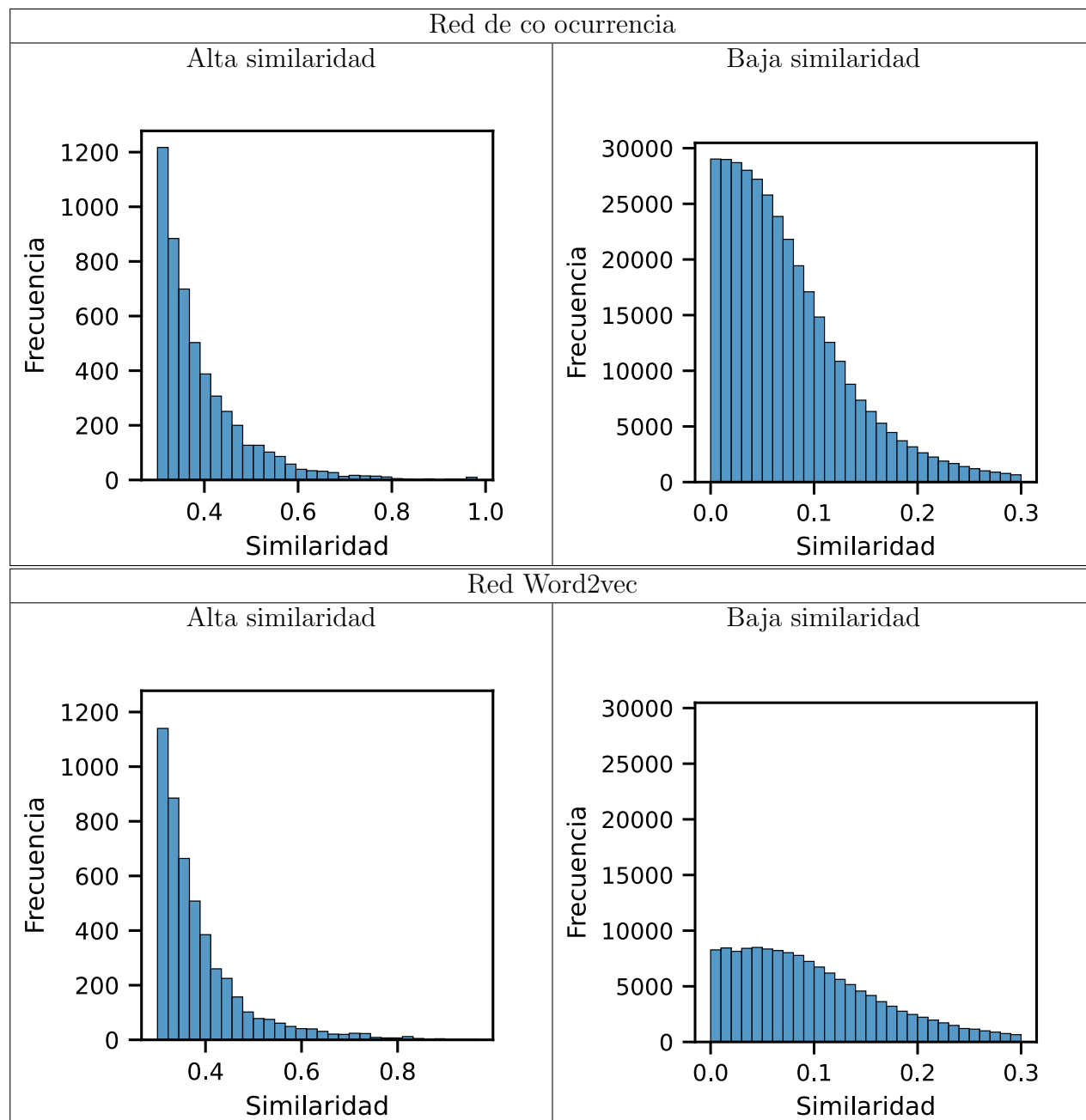


Tabla 7.13: Comparación de similitud de arcos de La Iliada

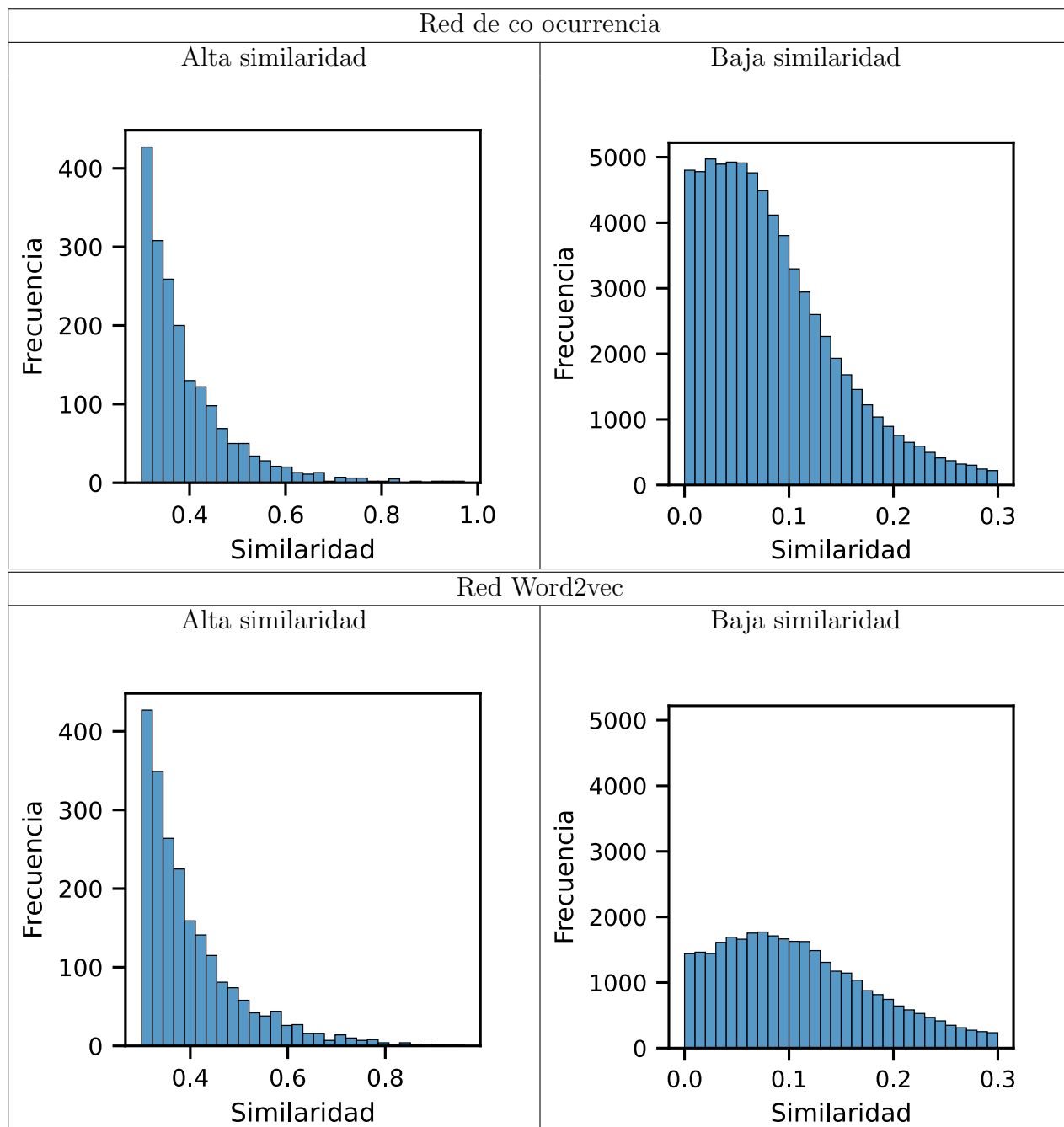


Tabla 7.14: Comparación de similitud de arcos de Pinocho

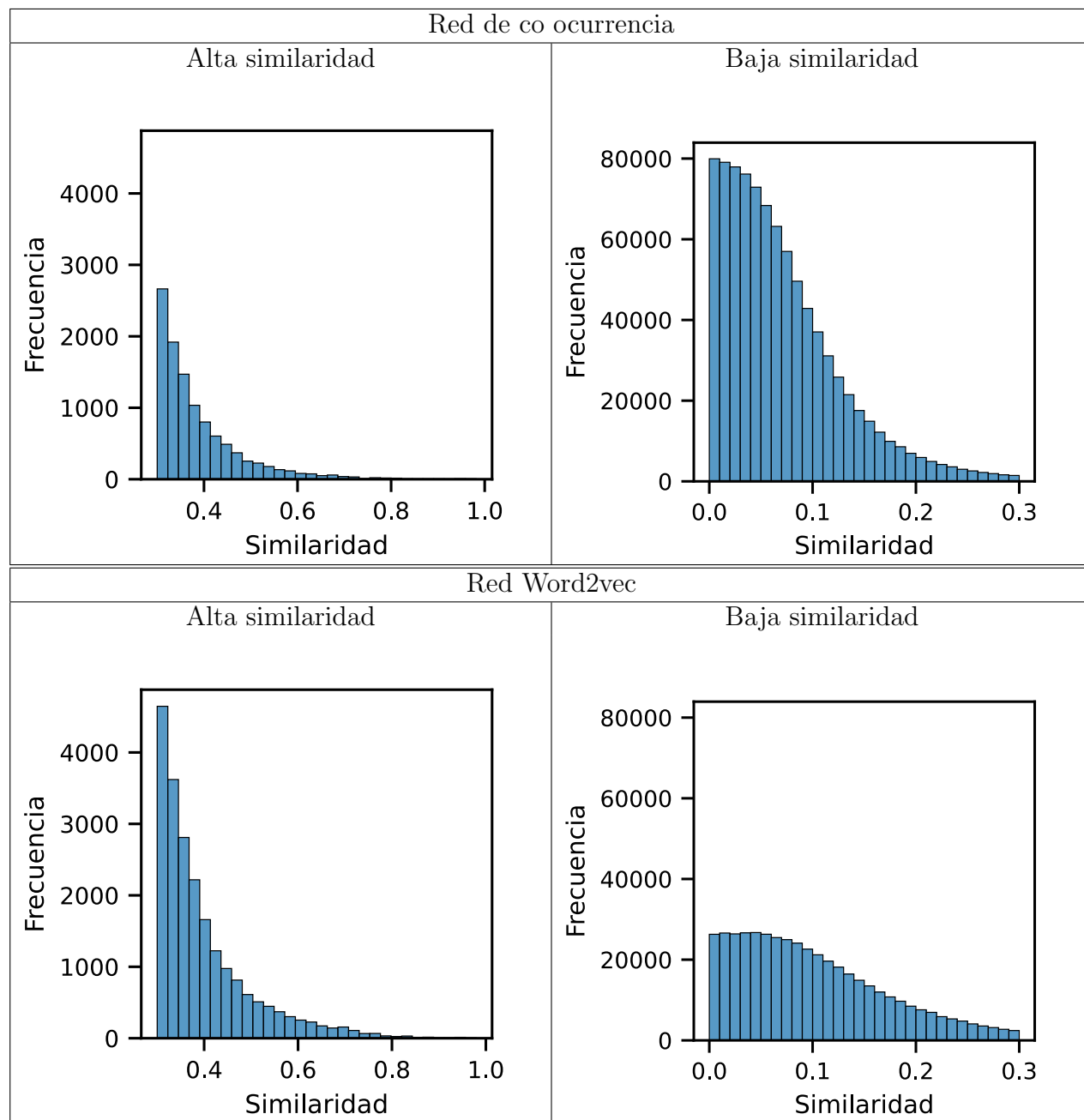


Tabla 7.15: Comparación de similitud de arcos de Guerra y Paz

7.8 Node2vec

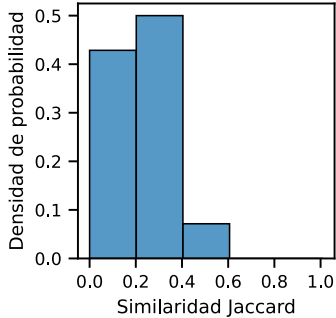
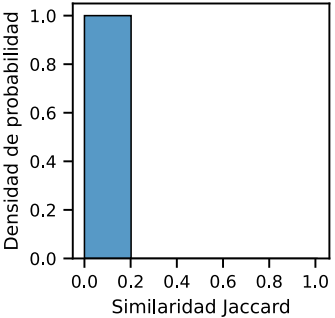
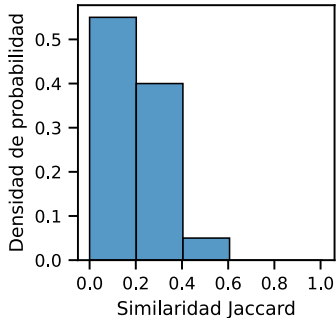
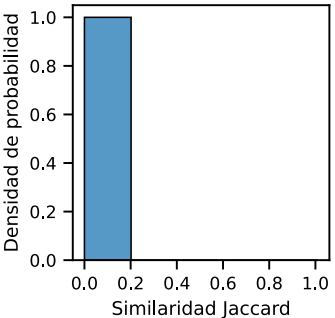
Red de coocurrencia	Red de coocurrencia aleatorizada	Descripción
 (a) Distribución similitud Jaccard emparejamiento comunidad-<i>cluster</i>	 (b) Distribución similitud Jaccard emparejamiento comunidad-<i>cluster</i>	Distribucion de similitud Jaccard para emparejamientos de comunidades en redes de coocurrencia con <i>cluster node2vec</i>
Red <i>word2vec</i>	Red <i>word2vec</i> aleatorizada	Descripción
 (c) Distribución similitud Jaccard emparejamiento comunidad-<i>cluster</i>	 (d) Distribución similitud Jaccard emparejamiento comunidad-<i>cluster</i>	Distribución de similitud Jaccard para emparejamientos de comunidades en redes <i>word2vec</i> con <i>cluster node2vec</i>

Tabla 7.16: Comparación de distribuciones de valores de Jaccard contra versiones aleatorizadas de las redes de Alicia en el país de las maravillas

Los gráficos correspondientes al emparejamiento comunidad-*cluster* con redes sin y con aleatorizar se muestran en las Tablas 7.16, 7.17, 7.18 y 7.19. En todos los casos es posible observar que las redes aleatorizadas producen comunidades que son poco similares con los *clusters* obtenidos por el modelo *node2vec* concentrándose el 100% de la distribución de valores en el rango $[0.0 - 0.2[$. En el caso de Alicia se obtiene tanto para la red por coocurrencia como la red *word2vec* distribuciones similares. En el caso de La Ilíada podemos observar que tanto en la red de coocurrencia como en la red *word2vec* un 50% de las comunidades quedan asociadas con *clusters* de baja similitud en el rango $[0 - 0.2[$. Por otro lado es posible notar que en la red de coocurrencia existe una comunidad

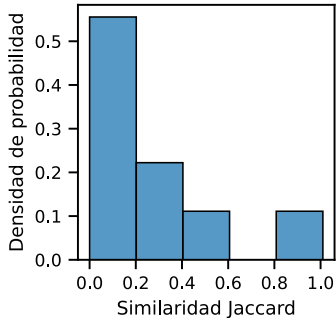
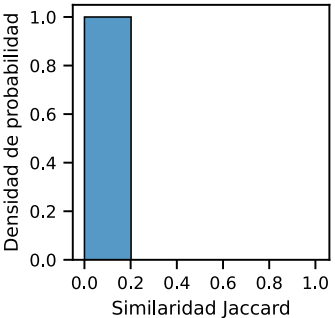
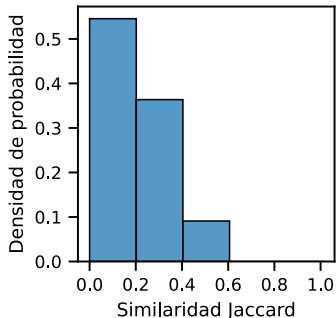
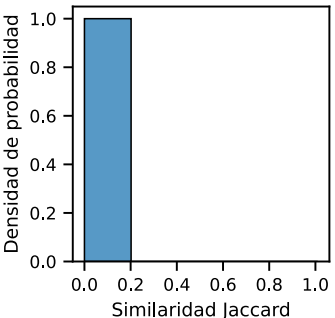
Red de coocurrencia	Red de coocurrencia aleatorizada	Descripción
 (a) Distribución similaridad Jaccard emparejamiento comunidad-<i>cluster</i>	 (b) Distribución similaridad Jaccard emparejamiento comunidad-<i>cluster</i>	Distribucion de similaridad Jaccard para emparejamientos de comunidades en redes de coocurrencia con <i>cluster node2vec</i>
Red <i>word2vec</i>	Red <i>word2vec</i> aleatorizada	Descripción
 (c) Distribución similaridad Jaccard emparejamiento comunidad-<i>cluster</i>	 (d) Distribución similaridad Jaccard emparejamiento comunidad-<i>cluster</i>	Distribución de similaridad Jaccard para emparejamientos de comunidades en redes <i>word2vec</i> con <i>cluster node2vec</i>

Tabla 7.17: Comparación de distribuciones de valores de Jaccard contra versiones aleatorizadas de las redes de La Ilíada

que presenta una similaridad extremadamente alta con un *cluster* en el rango $[0.8 - 1.0]$ cosa que no ocurre en la red *word2vec*. En el caso de Pinocho la red de coocurrencia produce un 60% de los emparejamientos con baja similaridad con los *clusters* en el rango $[0.0 - 0.2[$ mientras que la red *word2vec* un 50%. Guerra y paz nuevamente es la excepción y en el caso de la red *word2vec* tan solo un 30% de los emparejamientos producen un baja similaridad mientras que en coocurrencia lo hace un 60%. También es posible notar que en la red de coocurrencia existe una comunidad con una similaridad en el rango $[0.6 - 0.8[$ lo cual no ocurre en la red *word2vec*. La red *word2vec* también en este caso parece producir similaridades más estables concentrándose un 30% de los emparejamientos en el rango $[0.4 - 0.6[$ y el 40% faltante en el rango $[0.2 - 0.4[$

Un ejemplo de las tablas de asociación para el caso particular de Alicia entre los *embeddings*

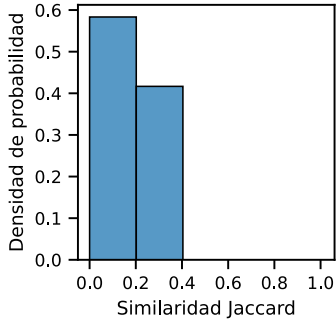
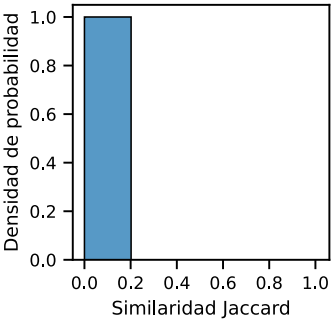
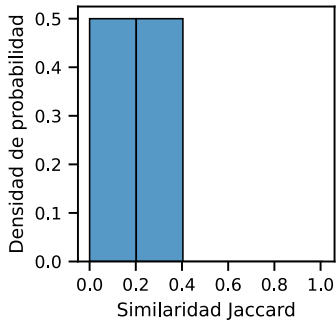
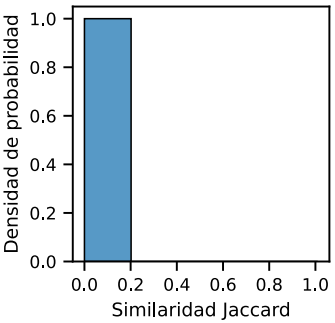
Red de coocurrencia	Red de coocurrencia aleatorizada	Descripción
 (a) Distribución similaridad Jaccard emparejamiento comunidad-<i>cluster</i>	 (b) Distribución similaridad Jaccard emparejamiento comunidad-<i>cluster</i>	Distribucion de similaridad Jaccard para emparejamientos de comunidades en redes de coocurrencia con <i>cluster node2vec</i>
Red <i>word2vec</i>	Red <i>word2vec</i> aleatorizada	
 (c) Distribución similaridad Jaccard emparejamiento comunidad-<i>cluster</i>	 (d) Distribución similaridad Jaccard emparejamiento comunidad-<i>cluster</i>	Distribución de similaridad Jaccard para emparejamientos de comunidades en redes <i>word2vec</i> con <i>cluster node2vec</i>

Tabla 7.18: Comparación de distribuciones de valores de Jaccard contra versiones aleatorizadas de las redes de Pinocho

generados por *node2vec* y las redes se muestra en el Apéndice C.

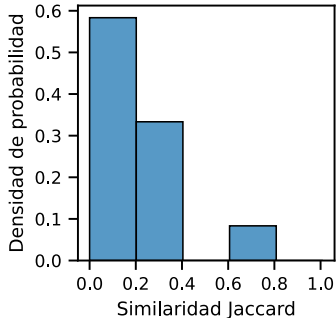
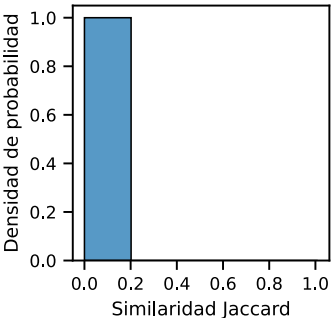
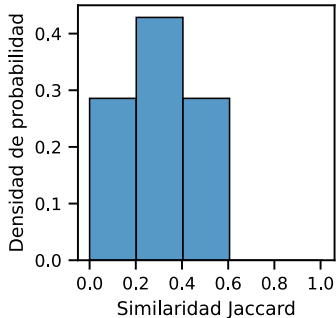
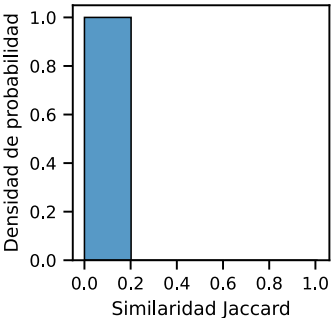
Red de coocurrencia	Red de coocurrencia aleatorizada	Descripción
 (a) Distribución similitud Jaccard emparejamiento comunidad-<i>cluster</i>	 (b) Distribución similitud Jaccard emparejamiento comunidad-<i>cluster</i>	Distribucion de similitaridad Jaccard para emparejamientos de comunidades en redes de coocurrencia con <i>cluster node2vec</i>
Red <i>word2vec</i>	Red <i>word2vec</i> aleatorizada	
 (c) Distribución similitud Jaccard emparejamiento comunidad-<i>cluster</i>	 (d) Distribución similitud Jaccard emparejamiento comunidad-<i>cluster</i>	Distribución de similitaridad Jaccard para emparejamientos de comunidades en redes <i>word2vec</i> con <i>cluster node2vec</i>

Tabla 7.19: Comparación de distribuciones de valores de Jaccard contra versiones aleatorizadas de las redes de Guerra y paz

Capítulo 8

Conclusiones y Recomendaciones

Construimos redes con y sin pesos usando textos y basadas en el principio de coocurrencia y a partir del modelo *word2vec*. Para evaluar la aleatoriedad de nuestros métodos evaluamos el efecto del entrenamiento con la red generada por *word2vec* el entrenamiento con *Adam* y el algoritmo probabilista para determinar si dos palabras se conectan y encontramos con un nivel de significancia $\alpha = 0.05$ que las redes siguen una misma distribución de grados.

Empíricamente encontramos que ambos tipos de redes presentan el llamado efecto mundo pequeño con distancias máximas entre nodos que escalan con $\frac{\log n}{\log k}$ y en el rango 2.45-2.92 lo que se encuentra en línea con los valores reportados en la literatura. Observamos también que la cantidad de arcos m en las redes *word2vec* es consistentemente menor que las redes de coocurrencia. Lo anterior pues el modelo *word2vec* únicamente se centra en relaciones que son estadísticamente significativas entre palabras. Creemos que esto ayuda a reducir el ruido en la red. A pesar de la menor cantidad en la red *word2vec* también es posible notar que esto no afecta significativamente la conectividad de la red y que la distancia promedio entre nodos de la red permanece en el mismo rango que la red de coocurrencia.

En el caso del coeficiente de *clustering* C promedio observamos mayores valores en el caso de la red generada por coocurrencia que la red generada mediante el modelo *word2vec*, sin embargo, al añadir pesos, esta situación se invierte y la red generada mediante *word2vec* pasa a tener un mayor valor $\tilde{C}$ en todos los casos. Lo anterior lo asociamos a que en esta última red el mecanismo generador puede producir que palabras pocas frecuentes que formen triángulos pueden poseer pesos con valor 1 que corresponde al máximo.

Al graficar el coeficiente de *clustering* C en función del grado sin considerar pesos observamos en ambos métodos el mismo patrón, a medida que aumenta el grado de los nodos el coeficiente disminuye. Al añadir pesos el patrón del coeficiente $\tilde{C}$ permanece constante pero la escala disminuye notablemente en el caso de coocurrencia. Esto nos revela que los triángulos se dan entre palabras que aparecen pocas veces juntas en el texto.

Encontramos que en ninguno de los casos las redes poseen una distribución libre de escala, independiente del método usado para la construcción. Particularmente en el caso de redes de coocurrencia se observa que palabras de bajo grado poseen una mayor frecuencia y que a medida que aumenta el grado la frecuencia disminuye. Por otro lado en las redes generadas por *word2vec* se observa que tanto las palabras con un bajo grado y alto grado (frecuencia en el texto) son poco frecuentes mientras que aquellas que poseen un grado intermedio son las más frecuentes.

En el caso de la asortatividad sin pesos en función del grado tenemos que en todos los textos las redes generadas por coocurrencia presentan un patrón disortativo en donde los nodos de menor grado tienden a conectarse con nodos de mayor grado. En el caso de *word2vec* para las redes de los textos más cortos (Alicia y Pinocho) los gráficos no presentan una forma clara disortativa pero en el caso de los textos más largos sí. Al añadir pesos a la red de coocurrencia basándose en frecuencia observamos que el patrón disortativo desaparece pero que esto es simplemente un efecto indeseable pues las palabras de menor grado se conectan a aquellas de más grado. Esto se observa en menor medida en el caso de redes *word2vec* y creemos que este método de asignar pesos puede ser combinado para evitar sesgos al momento de analizar la manera en que se organizan los nodos de la red.

Respecto a la estructura de comunidad encontramos que independiente del método de construcción las redes presentan una alta modularidad, lo cual indica la presencia de estructuras no aleatorias. En el caso de redes sin pesos tanto las redes por coocurrencia como las redes Word2vec presentan valores de modularidad que se alejan de la modularidad de una red aleatoria según medimos utilizando el z_{score} . Por otro lado las redes Word2vec poseen valores de modularidad más altos que las redes de coocurrencia y que generalmente se alejan más del valor esperado de redes aleatorias.

Al evaluar la similaridad entre las comunidades obtenidas por coocurrencia contra las comunidades de las redes *word2vec* encontramos que en general estas comunidades eran poco similares encontrándose la mayoría en el rango $[0.0 - 0.2[$, pese a ello existían unas pocas comunidades con valores más altos de similaridad en el rango $[0.2 - 0.6]$ en donde una investigación más detallada de las palabras que quedaban en la intersección reveló que existía similaridad semántica entre los términos. En el caso del texto Guerra y paz, que corresponde al texto más largo, al añadir pesos se observa una comunidad en la que el 30% los emparejamientos quedaban en el rango $[0.4 - 1.0]$ obteniendo incluso una similaridad 1.0. Esto nos lleva a pensar que pueden existir patrones de similaridad entre redes de coocurrencia y redes generadas por el modelo *word2vec* que sólo emergen cuando los textos sobre los cuales se trabaja son lo suficientemente grandes.

Por otro lado, cabe destacar que en el caso de redes sin peso podemos observar que ciertas analogías y relaciones de morfología que naturalmente aprende el modelo Word2vec se traducen en forma de nodos dentro de una misma comunidad. Por ejemplo, las palabras *fire*, *firewood* y *wood* quedan dentro de una misma comunidad y que *firewood* - *fire* $\approx$ *wood*. En el caso de

coocurrencia es posible notar que generalmente palabras morfológicamente similares como *god*, *gods*, *say* y *saying* quedan en comunidades diferentes. Creemos que los valores más altos de modularidad en el caso de las redes Word2vec se deben a posibles nuevas asociaciones entre palabras que no ocurren explícitamente por coocurrencia y a la eliminación del ruido al concentrarse únicamente en asociaciones estadísticamente significativas.

También es necesario destacar que en el caso de Alicia en el País de las maravillas las comunidades parecieran reflejar bien más una estructura local siguiendo el flujo del texto y agrupándose juntas palabras que describen capítulos dentro del libro. Sin embargo, al utilizar el modelo *word2vec* ciertas palabras se pierden en las comunidades agregándose otras semánticamente similares. Un estudio posterior podría ahondar en el tipo de relaciones que se pierdan o agregan.

Encontramos también que la red *word2vec* elimina una gran cantidad de nodos de baja similaridad de acuerdo a lo medido con el modelo de *Gensim*, esto es consistente con el hecho que la red *word2vec* aprende relaciones estadísticamente significativas. Sin embargo, como se aprecia en el caso de Alicia en el País de las Maravillas, algunas relaciones de alta similaridad también se pierden por no ser lo suficientemente robustas.

Luego al ejecutar *node2vec* y emparejar las comunidades de las redes con los *clusters* es posible observar que las versiones aleatorizadas de las redes producen comunidades extremadamente distintas que los *clusters* con el 100% de los valores de similaridad en el rango $[0.0 - 0.2[$. También fue posible observar que la red de coocurrencia produce ciertas comunidades extremadamente similares a los *clusters* con valores en el rango $[0.6 - 1.0]$ en el caso de La Ilíada y Guerra y paz lo cual no fue posible observar en la red *word2vec*. Ninguna de las dos redes pareciera ser lo suficientemente informativa para el modelo *node2vec* en términos de poder recuperar la estructura de comunidad en espacio vectorial.

En función de nuestros resultados creemos que ambos tipos de redes son complementarias entre si y describen distintas características del texto. Observamos que las comunidades de la red de coocurrencia parecen preservar más el flujo natural del texto que el enfoque *word2vec*. Por otro lado, la más alta modularidad de las redes *word2vec* se explica por la creación de nuevas asociaciones que capturan diversas relaciones semánticas que no aparecen simplemente por coocurrencia. A pesar de ello, otras asociaciones entre palabras que según *Gensim* son similares también son eliminadas, esto ocurre debido a que no ocurren una cantidad estadísticamente significativa en los textos. Este fenómeno se observa en menor medida en el caso de Guerra y Paz, que corresponde al texto más extenso. Creemos que la red de coocurrencia resulta más interesante y fiel si nos interesa observar el flujo de las palabras en el texto. Por otro lado, la red *word2vec* resulta más interesante en términos de las asociaciones semánticas que se crean indirectamente de acuerdo al uso que el autor le da a las palabras.

Respecto a posibles futuras líneas de investigación recomendamos investigar distintas formas de

añadir pesos a la red generada por coocurrencia tomando un criterio más robusto que simplemente la frecuencia. Un criterio que considere pesos en el rango 0-1 podría aprovechar métricas, como el Jaccard con pesos definido en [38], para realizar el emparejamiento entre comunidades por coocurrencia y *word2vec*. Creemos que tal mejora podría influenciar positivamente el análisis de similaridad semántica que realizamos en este trabajo. Por otro lado, es posible considerar formas de construir la red en base a otros modelos con *embeddings* estáticos como GloVe [39].

Respecto a mejoras en la metodología creemos razonable que una futura iteración de este experimento considere entrenar el modelo Word2vec sobre los corpus de texto juntos y luego crear la red considerando únicamente los nodos que ocurren en cada uno de los textos. Puesto que el rendimiento de las redes neuronales depende de la cantidad de ejemplos que estas poseen creemos que esto mejorará considerablemente la calidad de los *embeddings* y redes construidas.

Por otro lado queda también abierta la cuestión de cómo la red generada por coocurrencia se compara con la red *word2vec* pero en espacio hiperbólico y que métodos en cuestión pueden ser utilizados dada la complejidad de este nuevo espacio. Además, queda investigar si es que acaso el modelo Word2vec aprende relaciones sintácticas más complejas y si es que estas se reflejan en la red construida.

Referencias

- [1] Steven Pinker. *The Language Instinct*. William Morrow & Co, 1994.
- [2] Carter T Butts. Revisiting the foundations of network analysis. *Science*, 325(5939):414–416, 2009.
- [3] Jinyun Cong and Haitao Liu. Approaching human language with complex networks. *Physics of Life Reviews*, 11(3):398–418, 2014.
- [4] Henrique Ferraz de Arruda, Filipi Nascimento Silva, Vanessa Queiroz Marinho, Diego Raphael Amancio, and Luciano da Fontoura Costa. Representation of texts as complex networks: a mesoscopic approach. *Journal of Complex Networks*, 6(1):125–144, 2018.
- [5] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.
- [7] Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. In *Proceedings of the 28th Annual Meeting on Association for Computational Linguistics*, pages 22–29. Association for Computational Linguistics, 1990.
- [8] Ramon Ferrer i Cancho and Ricard V Solé. The small world of human language. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 268(1482):2261–2265, 2001.
- [9] Henrique F. de Arruda, Filipi N. Silva, Vanessa Q. Marinho, Diego R. Amancio, and Luciano da F. Costa. Topic segmentation via community detection in complex networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 26(6):063120, 2016.

- [10] Everton Oliva, Livia Murta, Diego R Amancio, Osvaldo N Oliveira Jr, Gonzalo Travieso, and Luciano da F. Costa. Text structuring methods based on complex network: a systematic review. *Scientometrics*, 126(2):1757–1789, 2021.
- [11] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: theory and experiment*, 2008(10):P10008, 2008.
- [12] Zhenisbek Assylbekov, Sultan Nurmukhamedov, Arsen Sheverdin, and Thomas Mach. From hyperbolic geometry back to word embeddings. In *Proceedings of the 7th Workshop on Representation Learning for NLP*, pages 39–45. Association for Computational Linguistics, 2022.
- [13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.
- [14] Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. Technical report, OpenAI, 2018.
- [15] James MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297, 1967.
- [16] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.
- [17] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.
- [18] Aaron Clauset, Cristopher Moore, and Mark EJ Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453(7191):98–101, 2008.
- [19] Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and María Boguñá. Hyperbolic geometry of complex networks. *Physical Review E*, 82(3):036106, 2010.
- [20] Zhenisbek Assylbekov, Rustem Jangeldin, and Mukhtar Karatayev. Squashed shifted PMI matrix: Bridging word embeddings and hyperbolic spaces. *arXiv preprint arXiv:2002.12005*, 2020.

- [21] Vito Latora, Vincenzo Nicosia, and Giovanni Russo. *Complex networks: Principles, methods and applications*. Cambridge University Press, 2017.
- [22] Stanley Milgram. The small world problem. *Psychology today*, 2(1):60–67, 1967.
- [23] Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.
- [24] Duncan J Watts and Steven H Strogatz. Collective dynamics of ‘small-world’ networks. *Nature*, 393(6684):440–442, 1998.
- [25] Jari Saramaki, Mikko Kivelä, Jukka-Pekka Onnela, Kimmo Kaski, and János Kertész. Generalizations of the clustering coefficient to weighted complex networks. *Physical Review E*, 75(2):027105, 2007.
- [26] Jukka-Pekka Onnela, Jari Saramäki, János Kertész, and Kimmo Kaski. Intensity and coherence of motifs in weighted complex networks. *Physical Review E*, 71(6):065103, 2005.
- [27] Alain Barrat, Marc Barthélemy, Romualdo Pastor-Satorras, and Alessandro Vespignani. The architecture of complex weighted networks. *Proceedings of the National Academy of Sciences*, 101(11):3747–3752, 2004.
- [28] Vladimir Batagelj and Matjaž Zaveršnik. Fast algorithms for determining (generalized) core groups in social networks. *Advances in Data Analysis and Classification*, 5(2):129–145, 2011.
- [29] Yoav Goldberg and Omer Levy. word2vec explained: Deriving Mikolov et al.’s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.
- [30] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*, pages 2177–2185, 2014.
- [31] Project gutenber. <https://www.gutenberg.org/>. Accedido el 2 de Abril de 2024.
- [32] Edward Loper and Steven Bird. Nltk: The natural language toolkit, 2002.
- [33] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

- [34] Aric A Hagberg, Daniel A Schult, and Pieter J Swart. Exploring network structure, dynamics, and function using networkx. *Proceedings of the 7th Python in Science Conference (SciPy2008)*, pages 11–15, 2008.
- [35] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.
- [36] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference on Neural Information Processing Systems (NIPS 2013)*, pages 746–754, 2013.
- [37] Carl Allen and Timothy Hospedales. Analogies explained: Towards understanding word embeddings. *arXiv preprint arXiv:1901.09813*, 2019.
- [38] Sergey Ioffe. Improved Consistent Sampling, Weighted Minhash and L1 Sketching. In *2010 IEEE International Conference on Data Mining*, pages 246–255. IEEE, 2010.
- [39] Jeffrey Pennington, Richard Socher, and Christopher D Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 1532–1543, 2014.

Apéndice A

Tablas de asociaciones comunidades k -core sin pesos

A.1 Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.028			
Top palabras intersección		Grado: Co-ocurrencia / Word2vec	
way		35 / 6	
find		32 / 7	
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
one	50	send	13
little	49	crawling	13
went	47	furrow	11
door	43	wherever	11
got	40	act	11
found	38	hedgehog	10
quite	36	besides	9
get	36	ridge	9
much	35	generally	9
back	35	unrolled	8

Tabla A.1: Tabla n° 1 de asociaciones de comunidades k-core para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.012			
Top palabras intersección		Grado: Co-ocurrencia / Word2vec	
go		22 / 8	
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
thought	27	white	17
see	24	dark	16
right	23	among	15
wonder	20	bill	15
never	19	coming	13
must	19	crash—now	11
oh	19	wander	11
it'll	18	loose	11
say	18	cool	11
shall	18	returning	11

Tabla A.2: Tabla n° 2 de asociaciones de comunidades k-core para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.467			
Top palabras intersección		Grado: Co-ocurrencia / Word2vec	
mouse—a		6 / 6	
grammar		6 / 6	
mouse—o		5 / 6	
mouse—to		6 / 6	
latin		6 / 6	
mouse—of		6 / 6	
brother's		5 / 6	
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
		spoke—fancy	7
		curtsey	7
		australia	7
		tried	7
		curtseying	7
		zealand	7
		ma'am	6
		falling	6

Tabla A.3: Tabla n° 3 de asociaciones de comunidades k-core para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.060			
Top palabras intersección		Grado: Co-ocurrencia / Word2vec	
don't		18 / 8	
can't		17 / 9	
i'll		15 / 7	
remember		15 / 12	
dinah		11 / 10	
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
think	24	moral	15
could	21	alice's	14
dear	21	together'	12
know	20	'em	12
things	17	right	11
mouse	17	fetch	10
nothing	17	odd	10
it's	16	hope	10
take	16	tie	10
talk	15	they'll	10

Tabla A.4: Tabla n° 4 de asociaciones de comunidades k-core para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.018			
Top palabras intersección		Grado: Co-ocurrencia / Word2vec	
knew		12 / 10	
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
thing	15	change	15
began	14	tinkling	14
like	13	teacups	12
something	12	clamour	12
lobster	12	reeds—the	12
sure	12	would	11
eat	12	rustling	11
heard	12	queen's	10
tell	11	sheepbells	10
saying	11	rippling	10

Tabla A.5: Tabla n° 5 de asociaciones de comunidades k-core para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.806			
Top palabras intersección		Grado: Co-ocurrencia / Word2vec	
youth		22 / 19	
yet		16 / 11	
jaws		7 / 15	
grey		7 / 12	
bones		7 / 12	
pray		10 / 12	
beak—		6 / 11	
sage		7 / 11	
backsomersault		8 / 11	
tougher		7 / 10	
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
use	10	finished	9
turned	9		
kept	8		
grown	7		
mentioned	6		

Tabla A.6: Tabla n° 6 de asociaciones de comunidades k-core para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.054			
Top palabras intersección		Grado: Co-ocurrencia / Word2vec	
added		30 / 6	
answer		19 / 6	
slates		15 / 6	
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
said	51	dates	6
alice	50	pence	6
hatter	41	shillings	6
king	40	reduced	6
march	33		
hare	33		
tone	28		
replied	28		
i'm	27		
well	27		

Tabla A.7: Tabla n° 7 de asociaciones de comunidades k-core para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.100			
Top palabras intersección		Grado: Co-ocurrencia / Word2vec	
soldiers		25 / 8	
gardeners		19 / 7	
ten		16 / 7	
ornamented		12 / 7	
courtiers		11 / 8	
flat		10 / 7	
diamonds		9 / 8	
corners		7 / 8	
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
queen	46	clubs	10
came	36	shaped	9
three	32	oblong	7
two	29		
next	27		
white	21		
rabbit	19		
hearts	18		
procession	15		
stood	15		

Tabla A.8: Tabla n° 8 de asociaciones de comunidades k-core para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.246			
Top palabras intersección		Grado: Co-ocurrencia / Word2vec	
earls		13 / 16	
morcar		13 / 15	
mercia		12 / 14	
usurpation		8 / 13	
northumbria—'		6 / 13	
edwin		8 / 12	
conquest		8 / 12	
'edwin		7 / 12	
accustomed		8 / 9	
ugh		5 / 8	
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
english	11	crowd	18
pope	9	assembled	13
favoured	8	them—all	11
immediate	7	shiver	10
wanted	7	lay	9
adoption	7	sorts	9
energetic	7	birds	9
cause	7	sprawling	9
submitted	7	feathers	9
spades	6	late	8

Tabla A.9: Tabla n° 9 de asociaciones de comunidades k-core para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.088			
Top palabras intersección		Grado: Co-ocurrencia / Word2vec	
mock		24 / 13	
turtle		24 / 12	
gryphon		24 / 9	
old		14 / 8	
sea		11 / 7	
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
eyes	23	subjects	12
time	22	modern	11
last	22	counting	11
half	21	ancient	11
looked	20	seaography	11
come	19	—mystery	10
caterpillar	18	flappers	10
voice	18	drawlingmaster	10
speak	17	lobsters	9
good	16	drawling—the	9

Tabla A.10: Tabla n° 10 de asociaciones de comunidades k-core para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.550			
Top palabras intersección		Grado: Co-ocurrencia / Word2vec	
owl		16 / 8	
panther		13 / 11	
later		7 / 11	
follows		9 / 11	
pie—		7 / 10	
sharing		7 / 10	
editions		8 / 10	
piecrust		9 / 10	
gravy		9 / 9	
meat		9 / 9	
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
pie	9	sands	9
share	9		
treat	9		
dish	8		
finished	8		
boon	8		
kindly	7		
permitted	6		

Tabla A.11: Tabla n° 11 de asociaciones de comunidades k-core para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.800			
Top palabras intersección		Grado: Co-ocurrencia / Word2vec	
disappointment		7 / 8	
shelves		6 / 8	
marmalade		7 / 8	
labelled		8 / 8	
jar		7 / 8	
orange		8 / 8	
empty		6 / 7	
passed		7 / 7	
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
great	8	pegs	8

Tabla A.12: Tabla n° 12 de asociaciones de comunidades k-core para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.118			
Top palabras intersección		Grado: Co-ocurrencia / Word2vec	
court		10 / 11	
officers		6 / 9	
applause		9 / 6	
immediately		8 / 9	
attempts		9 / 6	
suppressed		7 / 7	
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
read	12	dormouse	13
trials	8	clock	12
end	8	march	12
silence	7	hare	11
persons	7	pace	9
fortytwo	7	wish	9
book	6	crossed	9
newspapers	6	suppose	9
mile	6	sulkily	8
rule	6	time	8

Tabla A.13: Tabla n° 13 de asociaciones de comunidades k-core para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.314			
Top palabras intersección		Grado: Co-ocurrencia / Word2vec	
chorus		17 / 10	
soup		10 / 14	
beautiful		8 / 14	
sneezes		13 / 11	
beat		12 / 11	
boy		10 / 11	
words—		11 / 10	
annoy		8 / 10	
teases		10 / 9	
knows		9 / 10	
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
baby	21	evening	11
cook	20	tureen	10
duchess	18	green	9
pepper	11	hot	9
sneezing	10	beauti—ful	9
wow	10	dainties	8
even	9	stoop	8
howling	9	rich	8
kitchen	8	beau—ootiful	8
joined	8	waiting	8

Tabla A.14: Tabla n° 14 de asociaciones de comunidades k-core para Alicia en el país de las maravillas

A.2 La Ilíada

Matching: Jaccard Similarity = 0.015			
Top palabras intersección		Grado: Co-ocurrencia / Word2vec	
fair		5 / 67	
pasture		8 / 9	
orchomenian		6 / 8	
lowly		8 / 8	
downs		5 / 7	
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
neighbouring	12	next	69
towns	8	bold	59
lands	8	sprung	49
hears	7	rich	47
boebe's	7	famed	43
phenean	7	led	41
bordering	7	divine	39
incircle	6	great	39
abydos'	6	line	31
tegea's	6	captives	29

Tabla A.15: Tabla n° 1 de asociaciones de comunidades k-core para La Ilíada

Matching: Jaccard Similarity = 0.000			
No Intersection Nodes			
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
death	18	bowls	18
dead	17	wine	15
main	16	goblets	14
breath	15	feast	14
toil	13	pure	12
lie	13	nectar	12
bowers	12	libations	11
glow	12	enhanced	11
pant	11	crown'd	11
immortal	11	repress'd	11

Tabla A.16: Tabla n° 2 de asociaciones de comunidades k-core para La Ilíada

Matching: Jaccard Similarity = 0.193			
Top palabras intersección		Grado: Co-ocurrencia / Word2vec	
book		35 / 74	
hector		69 / 37	
achilles		66 / 31	
battle		62 / 52	
greeks		60 / 24	
argument		27 / 56	
trojans		55 / 30	
illustration		40 / 53	
ajax		51 / 19	
combat		50 / 15	
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
war	64	army	28
great	62	catalogue	26
first	58	apotheosis	26
ships	47	described	26
bold	47	acts	26
stand	46	taken	24
rest	46	put	23
troops	45	advice	22
train	44	pursuance	21
walls	43	iris	21

Tabla A.17: Tabla n° 3 de asociaciones de comunidades k-core para La Illíada

Matching: Jaccard Similarity = 0.000			
No Intersection Nodes			
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
voices	2	aleian	4
looking	2	way170	2
backwards	2	forsaking	2
		forlorn	2
		humankind	2

Tabla A.18: Tabla n° 4 de asociaciones de comunidades k-core para La Illíada

Matching: Jaccard Similarity = 0.275			
Top palabras intersección		Grado: Co-ocurrencia / Word2vec	
ground		169 / 64	
head		163 / 57	
hand		157 / 73	
stood		145 / 40	
lance		145 / 51	
breast		140 / 57	
arm		139 / 39	
beneath		138 / 37	
foe		137 / 36	
car		137 / 20	

Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
o'er	202	atrides	44
plain	174	bone	43
field	169	forceful	40
round	164	discharged	39
around	160	corslet	38
along	158	dropp'd	38
high	157	burst	37
fierce	141	stone	36
steeds	141	passage	36
swift	141	arms	35

Tabla A.19: Tabla n° 5 de asociaciones de comunidades k-core para La Illíada

Matching: Jaccard Similarity = 0.000			
No Intersection Nodes			
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
twelfth	5	doris	12
sixteenth	5	thoa	12
thirteenth	5	pherusa	10
fifteenth	5	panope	10
seventeenth	5	proto	10
fourteenth	5	rear	9
		melita	9
		appear	9
		doto	9
		limnoria	9

Tabla A.20: Tabla n° 6 de asociaciones de comunidades k-core para La Illíada

Matching: Jaccard Similarity = 0.275			
Top palabras intersección		Grado: Co-ocurrencia / Word2vec	
thy		167 / 82	
shall		162 / 164	
thou		159 / 79	
jove		156 / 57	
let		155 / 111	
thee		151 / 71	
troy		151 / 76	
may		129 / 150	
yet		149 / 86	
gods		146 / 139	
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
son	145	god	117
day	142	wisdom	81
trojan	137	war	73
soul	125	birth	65
slain	117	said	65
chiefs	115	quit	65
proud	109	hast	63
mighty	106	thus	61
achilles'	85	right	61
eternal	80	country	60

Tabla A.21: Tabla n° 7 de asociaciones de comunidades k-core para La Illíada

Matching: Jaccard Similarity = 0.057			
Top palabras intersección		Grado: Co-ocurrencia / Word2vec	
fight		54 / 44	
like		51 / 43	
shade		32 / 47	
sight		45 / 35	
bands		44 / 23	
height		24 / 42	
beheld		30 / 42	
voice		41 / 39	
lofty		31 / 41	
wait		25 / 40	
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
thus	82	o'er	128
arms	68	along	115
god	61	fields	106
eyes	60	plain	81
sacred	59	troops	79
said	57	thick	76
whose	56	loud	73
every	56	skies	70
hands	55	shores	70
hero	52	plains	69

Tabla A.22: Tabla n° 8 de asociaciones de comunidades k-core para La Ilíada

Matching: Jaccard Similarity = 0.810			
Top palabras intersección		Grado: Co-ocurrencia / Word2vec	
ois		10 / 16	
kateschethaen		10 / 16	
en		14 / 16	
procophai		10 / 16	
allois		9 / 16	
epitaeoeimasi		9 / 16	
de		10 / 15	
aesthomaen		10 / 14	
isos		10 / 14	
pleon		9 / 13	
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
proionta	8		
aurel	7		
eupoia—to	6		
anton	6		

Tabla A.23: Tabla n° 9 de asociaciones de comunidades k-core para La Illíada

A.3 Pinocho

Matching: Jaccard Similarity = 0.259			
Top palabras intersección		Grado: Co-ocurrencia / Word2vec	
marbles		6 / 14	
bicycles		6 / 13	
played		6 / 13	
others		6 / 12	
hopscotch		6 / 11	
ball		6 / 10	
rode		6 / 9	
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
		gathered	13
		street	12
		shouting	12
		blowing	12
		blindman's	12
		horses	11
		deafening	10
		racket	10
		everywhere	10
		buff	10

Tabla A.24: Tabla n° 1 de asociaciones de comunidades k-core para Pinocho

Matching: Jaccard Similarity = 0.178			
Top palabras intersección		Grado: Co-ocurrencia / Word2vec	
pinocchi		10 / 7	
living		8 / 7	
childrenand		8 / 7	
lucky		8 / 7	
richest		8 / 7	
begged		8 / 7	
choosing		7 / 7	
pinocchia		6 / 7	
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
geppetto	27		
mastro	19		
name	17		
lovely	14		
carpenter	14		
color	14		
antonio	11		
red	11		
azure	10		
knew	10		

Tabla A.25: Tabla n° 2 de asociaciones de comunidades k-core para Pinocho

Matching: Jaccard Similarity = 0.139			
Top palabras intersección		Grado: Co-ocurrencia / Word2vec	
hands		37 / 9	
mouth		34 / 14	
legs		34 / 11	
feet		33 / 12	
open		31 / 9	
pinocchio's		30 / 14	
ran		27 / 17	
pulled		27 / 13	
neck		23 / 25	
around		25 / 24	

Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
two	41	opened	23
head	37	collar	21
put	33	rope	19
long	33	see	18
eyes	30	giant	18
took	27	chin	17
nose	26	caught	17
made	26	oak	16
gave	26	window	16
tried	25	till	15

Tabla A.26: Tabla n° 3 de asociaciones de comunidades k-core para Pinocho

Matching: Jaccard Similarity = 0.000			
No Intersection Nodes			
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
still	9	heads	8
door	9	flounders	8
left	9	splendid	8
another	9	tasty	8
animal	8	whitefish	8
hours	8	crabs	7
half	8	delicious	7
snail	8	bass	7
opened	8	anchovies	7
hour	8		

Tabla A.27: Tabla n° 4 de asociaciones de comunidades k-core para Pinocho

Matching: Jaccard Similarity = 0.217			
Top palabras intersección		Grado: Co-ocurrencia / Word2vec	
fire		28 / 22	
make		27 / 13	
wood		22 / 18	
harlequin		14 / 19	
eater		19 / 15	
stage		9 / 16	
yelled		7 / 15	
given		15 / 8	
pulcinella		7 / 14	
set		13 / 8	
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
one	42	block	15
day	27	dry	13
first	22	hoarse	12
fox	21	firewood	12
cat	19	thick	11
sure	16	piece	11
loud	16	solid	11
sunday	14	he'll	10
vacation	14	needed	10
begins	13	cooking	10

Tabla A.28: Tabla n° 5 de asociaciones de comunidades k-core para Pinocho

Matching: Jaccard Similarity = 0.464			
Top palabras intersección		Grado: Co-ocurrencia / Word2vec	
coat		30 / 8	
silk		23 / 22	
dropped		20 / 8	
tail		19 / 13	
waist		14 / 18	
velvet		18 / 14	
many		15 / 8	
slippers		10 / 14	
curls		13 / 14	
silverbuckled		10 / 14	

Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
buy	23	medoro	9
new	14	tassels	8
book	14	white	8
abc	13		
pair	12		
breeches	12		
sold	11		
sell	11		
cold	11		
suit	10		

Tabla A.29: Tabla n° 6 de asociaciones de comunidades k-core para Pinocho

Matching: Jaccard Similarity = 0.000			
No Intersection Nodes			
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
said	39	greetings	9
marionette	31	dramatic	9
donkey	30	company	8
well	27	strange	8
let	22	friendly	7
say	21	actors	7
take	20	embraces	6
farmer	19	knocks	6
place	19	received	6
wall	15	actresses	6

Tabla A.30: Tabla n° 7 de asociaciones de comunidades k-core para Pinocho

Matching: Jaccard Similarity = 0.096			
Top palabras intersección		Grado: Co-ocurrencia / Word2vec	
little		67 / 19	
poor		58 / 27	
time		49 / 15	
saw		43 / 16	
turned		42 / 17	
voice		40 / 18	
asked		39 / 30	
room		37 / 16	
felt		37 / 15	
toward		36 / 11	
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
came	57	tried	31
went	51	nose	29
away	49	pinocchio	28
looked	48	cry	24
sea	47	answered	23
soon	47	tears	22
could	47	long	20
heard	43	dead	20
last	39	fell	20
found	37	talking	19

Tabla A.31: Tabla n° 8 de asociaciones de comunidades k-core para Pinocho

Matching: Jaccard Similarity = 0.249			
Top palabras intersección		Grado: Co-ocurrencia / Word2vec	
good		75 / 20	
boy		70 / 16	
fairy		70 / 27	
want		68 / 26	
i'll		67 / 26	
come		66 / 19	
don't		65 / 25	
father		63 / 15	
home		63 / 24	
go		62 / 34	
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
answered	67	hurry	20
know	66	money	20
see	65	four	19
dear	64	take	19
like	61	refuse	18
think	60	word	18
cried	59	pay	18
oh	57	return	17
old	56	lesson	16
tell	56	wanted	16

Tabla A.32: Tabla n° 9 de asociaciones de comunidades k-core para Pinocho

Matching: Jaccard Similarity = 0.289			
Top palabras intersección		Grado: Co-ocurrencia / Word2vec	
ordered		10 / 18	
hare		11 / 10	
dozen		11 / 10	
partridges		9 / 10	
chickens		10 / 9	
young		7 / 10	
tender		8 / 9	
spring		9 / 9	
pheasants		8 / 9	
couple		7 / 8	
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
dressed	6	enemy	15
		celebration	13
		simons	11
		books	10
		simple	10
		victory	10
		geographies	10
		kinds	10
		histories	10
		emperor	10

Tabla A.33: Tabla n° 10 de asociaciones de comunidades k-core para Pinocho

Matching: Jaccard Similarity = 0.389			
Top palabras intersección		Grado: Co-ocurrencia / Word2vec	
courts		10 / 6	
europe		9 / 6	
emperors		8 / 6	
queens		8 / 6	
kings		7 / 6	
attention		7 / 6	
performing		6 / 6	
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
great	14		
honor	10		
thank	9		
allow	8		
offend	7		
deeply	7		
speech	6		
paying	5		
bill	5		
greeted	5		

Tabla A.34: Tabla n° 11 de asociaciones de comunidades k-core para Pinocho

Matching: Jaccard Similarity = 0.065			
Top palabras intersección		Grado: Co-ocurrencia / Word2vec	
gold		24 / 17	
pieces		22 / 11	
bury		9 / 5	
hole		7 / 6	
robbed		7 / 6	
punishment		7 / 5	
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
pinocchio	26	fox	26
five	16	cat	19
morning	12	paw	15
piece	12	lame	13
hundred	11	eyes	12
thousand	11	held	12
chapter	11	blind	12
four	11	unconsciously	12
next	11	supposed	11
beautiful	11	laughingly	11

Tabla A.35: Tabla n° 12 de asociaciones de comunidades k-core para Pinocho

A.4 Guerra y paz

Matching: Jaccard Similarity = 0.347			
Top palabras intersección		Grado: Co-ocurrencia / Word2vec	
came		284 / 76	
eyes		279 / 147	
face		279 / 40	
looked		278 / 152	
went		277 / 44	
back		277 / 223	
away		272 / 67	
head		271 / 170	
round		270 / 105	
another		270 / 78	
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
little	267	seat	107
like	262	road	101
saw	260	shoes	98
way	248	sometimes	96
look	245	armchair	96
heard	242	reached	92
suddenly	240	waved	90
denísov	219	occasionally	89
house	210	thrown	87
seen	205	cossacks	86

Tabla A.36: Tabla n° 1 de asociaciones de comunidades k-core para Guerra y paz

Matching: Jaccard Similarity = 0.000			
No Intersection Nodes			
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
crown—but	2	honey	12
poles—all	2	buttermilk	12
beggarly	2	chicken	11
		roasted	10
		jam	9
		preserves	9
		nutandhoney	9
		cakes	8
		comb	7
		ham	7

Tabla A.37: Tabla n° 2 de asociaciones de comunidades k-core para Guerra y paz

Matching: Jaccard Similarity = 0.207			
Top palabras intersección		Grado: Co-ocurrencia / Word2vec	
de		18 / 58	
la		15 / 49	
que		16 / 45	
faire		10 / 43	
le		17 / 40	
vous		17 / 38	
des		11 / 37	
pas		15 / 34	
et		15 / 34	
à		13 / 32	
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
fait	11	l'angleterre	36
les	11	extrémités	34
		reconnaissante	31
		j'explique	31
		moscou	30
		grâce	30
		moyens	28
		paraissent	27
		avec	26
		être	26

Tabla A.38: Tabla n° 3 de asociaciones de comunidades k-core para Guerra y paz

Matching: Jaccard Similarity = 0.000			
No Intersection Nodes			
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
kari	5	manna	7
safi	5	astraea	7
mala	5	scotch	7
tafa	5	fuss	7
kaská	5	seekers	7
muter	5	authentic	7
		charter	6
		lodge	6

Tabla A.39: Tabla n° 4 de asociaciones de comunidades k-core para Guerra y paz

Matching: Jaccard Similarity = 0.111			
Top palabras intersección		Grado: Co-ocurrencia / Word2vec	
army		146 / 173	
french		149 / 37	
russian		142 / 79	
moscow		142 / 31	
first		139 / 34	
napoleon		136 / 101	
general		136 / 53	
emperor		136 / 35	
battle		135 / 79	
troops		135 / 85	
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
day	139	power	181
kutúzov	133	necessary	169
made	130	historians	158
already	121	force	138
sent	120	action	133
three	119	history	130
take	116	conception	121
road	115	certain	120
last	114	events	117
regiment	110	relation	115

Tabla A.40: Tabla n° 5 de asociaciones de comunidades k-core para Guerra y paz

Matching: Jaccard Similarity = 0.344			
Top palabras intersección		Grado: Co-ocurrencia / Word2vec	
tell		127 / 142	
said		134 / 59	
prince		133 / 37	
know		133 / 74	
asked		133 / 107	
go		133 / 68	
pierre		133 / 76	
see		132 / 96	
well		131 / 103	
come		131 / 113	
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
honor	90	never	100
woman	86	could	98
says	85	everything	94
kissed	81	take	92
daughter	81	reply	80
hélène	79	knew	78
listened	78	house	76
		thought	75
		anything	74
		happened	73

Tabla A.41: Tabla n° 6 de asociaciones de comunidades k-core para Guerra y paz

Matching: Jaccard Similarity = 0.101			
Top palabras intersección		Grado: Co-ocurrencia / Word2vec	
life		211 / 28	
felt		206 / 44	
seemed		196 / 21	
without		196 / 18	
yet		188 / 18	
feeling		182 / 26	
moment		180 / 17	
new		164 / 69	
expression		164 / 92	
mind		158 / 40	
Top palabras exclusivas co-ocurrencia	Grado	Top palabras exclusivas Word2vec	Grado
would	218	became	50
could	214	solemn	48
man	213	sympathy	48
time	212	sorrow	46
thought	203	tender	44
people	202	sadness	38
always	202	quiet	37
men	202	moments	37
even	199	feelings	37
whole	196	pity	36

Tabla A.42: Tabla n° 7 de asociaciones de comunidades k-core para Guerra y paz

Apéndice B

Tablas de asociaciones comunidades *k-core* con pesos

B.0.1 Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.024			
Top palabras intersección N = 2		Grado: Co-ocurrencia / Word2vec	
looked		37 / 8	
garden		22 / 10	
Top palabras exclusivas co-ocurrencia N = 53	Grado	Top palabras exclusivas Word2vec N = 29	Grado
little	50	dark	14
time	40	among	11
quite	39	row	11
get	38	wander	10
back	36	loveliest	10
door	36	cool	10
way	35	flowers	10
found	34	lodging	10
round	34	bright	9
another	33	spades	9

Tabla B.1: Tabla n° 1 de asociaciones de comunidades k-core con pesos para Alicia

Matching: Jaccard Similarity = 0.226			
Top palabras intersección N = 12		Grado: Co-ocurrencia / Word2vec	
would		35 / 9	
change		11 / 14	
knew		12 / 9	
reeds—the		7 / 12	
teacups		7 / 12	
dull		12 / 7	
rustling		8 / 10	
rippling		7 / 9	
wind		9 / 9	
rattling		7 / 9	
Top palabras exclusivas co-ocurrencia N = 24	Grado	Top palabras exclusivas Word2vec N = 17	Grado
could	20	tinkling	14
see	18	queen's	11
things	18	clamour	11
going	10	sheepbells	10
heart	10	cries	10
go	10	noises	9
hear	10	shrill	9
look	10	shepherd	8
whether	10	boy—and	8
keep	10	farmyard—while	8

Tabla B.2: Tabla n° 2 de asociaciones de comunidades k-core con pesos para Alicia

Matching: Jaccard Similarity = 0.067			
Top palabras intersección N = 6		Grado: Co-ocurrencia / Word2vec	
rabbit		30 / 7	
white		27 / 13	
gloves		16 / 10	
kid		12 / 8	
pair		11 / 11	
fan		11 / 10	

Top palabras exclusivas co-ocurrencia N = 35	Grado	Top palabras exclusivas Word2vec N = 48	Grado
voice	20	upon	16
hand	19	things—	14
began	19	unjust	13
heard	13	pictures	12
put	13	fall	12
next	12	sticks	10
walking	11	skurried	10
anxiously	11	shakespeare	10
hurried	11	well	10
fetch	10	hung	9

Tabla B.3: Tabla n° 3 de asociaciones de comunidades k-core con pesos para Alicia

Matching: Jaccard Similarity = 0.329			
Top palabras intersección N = 26		Grado: Co-ocurrencia / Word2vec	
turtle		52 / 13	
mock		50 / 14	
gryphon		44 / 12	
sea		30 / 12	
lobsters		22 / 14	
advance		15 / 8	
far		13 / 6	
reason		13 / 8	
wash		8 / 11	
somersault		10 / 6	
Top palabras exclusivas co-ocurrencia N = 48	Grado	Top palabras exclusivas Word2vec N = 5	Grado
went	25	yelled	7
old	24	set	7
cried	17	replied	7
though	14	thoughtfully	6
us	14	mouths—and	6
interrupted	13		
lobster	13		
used	12		
turn	12		
school	10		

Tabla B.4: Tabla n° 4 de asociaciones de comunidades k-core con pesos para Alicia

Matching: Jaccard Similarity = 0.467			
Top palabras intersección N = 7		Grado: Co-ocurrencia / Word2vec	
latin		6 / 6	
mouse—of		6 / 6	
mouse—to		6 / 6	
brother’s		5 / 6	
mouse—o		5 / 6	
mouse—a		6 / 6	
grammar		6 / 6	
Top palabras exclusivas co-ocurrencia N = 0	Grado	Top palabras exclusivas Word2vec N = 8	Grado
		tried	7
		curtsey	7
		curtseying	7
		australia	7
		spoke—fancy	7
		zealand	7
		falling	6
		ma’am	6

Tabla B.5: Tabla n° 5 de asociaciones de comunidades k-core con pesos para Alicia

Matching: Jaccard Similarity = 0.190			
Top palabras intersección N = 8		Grado: Co-ocurrencia / Word2vec	
wish		16 / 6	
dinah		13 / 9	
mice		11 / 6	
milk		8 / 9	
they'll		7 / 8	
hope		8 / 7	
saucer		7 / 8	
teatime		8 / 8	
Top palabras exclusivas co-ocurrencia N = 31	Grado	Top palabras exclusivas Word2vec N = 3	Grado
bottle	16	tonight	7
dear	16	remember	6
almost	16	think	6
drink	12		
words	12		
marked	11		
saying	10		
oh	9		
eat	9		
says	8		

Tabla B.6: Tabla n° 6 de asociaciones de comunidades k-core con pesos para Alicia

Matching: Jaccard Similarity = 0.256			
Top palabras intersección N = 10		Grado: Co-ocurrencia / Word2vec	
chorus		11 / 10	
boy		8 / 11	
beat		10 / 11	
sneezes		11 / 11	
enjoy		8 / 10	
annoy		7 / 10	
knows		7 / 10	
teases		8 / 9	
thoroughly		8 / 9	
pleases		7 / 8	
Top palabras exclusivas co-ocurrencia N = 3	Grado	Top palabras exclusivas Word2vec N = 26	Grado
pepper	9	hot	15
cook	7	beautiful	13
wow	7	soup	13
		evening	11
		roast	10
		toffee	10
		mixed	10
		words—	10
		pineapple	10
		turkey	10

Tabla B.7: Tabla n° 7 de asociaciones de comunidades k-core con pesos para Alicia

Matching: Jaccard Similarity = 0.182			
Top palabras intersección N = 8		Grado: Co-ocurrencia / Word2vec	
earls		13 / 9	
mercia		13 / 9	
morcar		12 / 9	
northumbria—'		5 / 9	
usurpation		8 / 9	
edwin		7 / 9	
'edwin		7 / 8	
conquest		8 / 7	
Top palabras exclusivas co-ocurrencia N = 34	Grado	Top palabras exclusivas Word2vec N = 2	Grado
wanted	13	shiver	8
pope	9	ugh	7
northumbria	9		
generally	8		
ridge	8		
furrow	8		
favoured	8		
declared	8		
accustomed	8		
wherever	8		

Tabla B.8: Tabla n° 8 de asociaciones de comunidades k-core con pesos para Alicia

Matching: Jaccard Similarity = 0.077			
Top palabras intersección N = 5		Grado: Co-ocurrencia / Word2vec	
queen		28 / 7	
live		15 / 6	
repeated		11 / 7	
play		8 / 6	
croquet		8 / 7	
Top palabras exclusivas co-ocurrencia N = 40	Grado	Top palabras exclusivas Word2vec N = 20	Grado
dormouse	25	flamingoes	11
great	21	invitation	9
soldiers	18	solemn	9
two	18	mustard	8
hare	17	flock	8
looking	16	is—‘birds	8
march	16	bite	8
three	15	balls	7
bottom	15	poky	7
minute	15	together’	7

Tabla B.9: Tabla n° 9 de asociaciones de comunidades k-core con pesos para Alicia

Matching: Jaccard Similarity = 0.042			
Top palabras intersección N = 1		Grado: Co-ocurrencia / Word2vec	
gave		7 / 7	
Top palabras exclusivas co-ocurrencia N = 16	Grado	Top palabras exclusivas Word2vec N = 7	Grado
one	16	muscular	7
close	10	strength	7
first	9	lasted	7
make	9	wife	7
become	8	argued	7
bill	8	jaw	6
exactly	7	case	6
obliged	7		
hunting	7		
got	7		

Tabla B.10: Tabla n° 10 de asociaciones de comunidades k-core con pesos para Alicia

Matching: Jaccard Similarity = 0.012			
Top palabras intersección N = 1		Grado: Co-ocurrencia / Word2vec	
nothing		47 / 5	
Top palabras exclusivas co-ocurrencia N = 76	Grado	Top palabras exclusivas Word2vec N = 5	Grado
said	76	again—‘before	5
alice	76	fit—’	5
know	66	table	5
like	60	tarts	5
don’t	59	clearer	5
say	57		
think	56		
thought	55		
must	55		
king	55		

Tabla B.11: Tabla n° 11 de asociaciones de comunidades k-core con pesos para Alicia

Matching: Jaccard Similarity = 0.254			
Top palabras intersección N = 16		Grado: Co-ocurrencia / Word2vec	
youth		12 / 16	
grey		7 / 12	
pray		8 / 11	
sage		6 / 11	
beak—		7 / 10	
ointment—one		10 / 10	
box—		8 / 10	
bones		6 / 10	
limbs		9 / 7	
supple		9 / 9	
Top palabras exclusivas co-ocurrencia N = 40	Grado	Top palabras exclusivas Word2vec N = 7	Grado
panther	19	jaws	13
use	17	backsomersault	11
owl	17	door—	9
finished	15	yet	9
later	12	uncommonly	8
editions	12	fat	7
follows	11	tougher	7
continued	11		
buttons	10		
belt	10		

Tabla B.12: Tabla n° 12 de asociaciones de comunidades k-core con pesos para Alicia

B.0.2 La Ilíada

Matching: Jaccard Similarity = 0.101			
Top palabras intersección N = 20		Grado: Co-ocurrencia / Word2vec	
battle		36 / 60	
illustration		25 / 53	
hector		42 / 44	
achilles		43 / 36	
greeks		38 / 24	
trojans		37 / 33	
ajax		36 / 18	
patroclus		35 / 21	
agamemnon		24 / 35	
upon		25 / 32	
Top palabras exclusivas co-ocurrencia N = 24	Grado	Top palabras exclusivas Word2vec N = 155	Grado
great	42	book	80
son	40	argument	61
first	39	jupiter	50
fight	36	return	38
bold	35	described	32
stand	34	army	29
death	34	night	27
combat	33	acts	27
dead	30	catalogue	26
troops	29	apotheosis	26

Tabla B.13: Tabla n° 1 de asociaciones de comunidades k-core con pesos para La Ilíada

Matching: Jaccard Similarity = 0.014			
Top palabras intersección N = 1		Grado: Co-ocurrencia / Word2vec	
rose		30 / 6	
Top palabras exclusivas co-ocurrencia N = 51	Grado	Top palabras exclusivas Word2vec N = 18	Grado
thus	51	morn	11
sacred	43	streak'd	10
spoke	43	rosy	10
chief	40	early	9
hero	35	saffron	9
long	35	lawn	8
grecian	35	aurora	8
rest	34	spread219	8
tent	33	dewy	8
monarch	33	blushes	8

Tabla B.14: Tabla n° 2 de asociaciones de comunidades k-core con pesos para La Ilíada

Matching: Jaccard Similarity = 0.029			
Top palabras intersección N = 2		Grado: Co-ocurrencia / Word2vec	
melancholy		15 / 7	
coast		11 / 7	
Top palabras exclusivas co-ocurrencia N = 58	Grado	Top palabras exclusivas Word2vec N = 8	Grado
eyes	47	naked	9
arms	44	plaintive	8
sad	38	wandering	8
fair	35	dreary	7
face	31	flits	6
grief	30	lone	6
tears	30	ghost	6
soft	29	clay	6
whose	28		
mother	27		

Tabla B.15: Tabla n° 3 de asociaciones de comunidades k-core con pesos para La Ilíada

Matching: Jaccard Similarity = 0.055			
Top palabras intersección N = 7		Grado: Co-ocurrencia / Word2vec	
polished		5 / 12	
toil		6 / 12	
bath		6 / 11	
refresh'd		6 / 10	
dissolving		5 / 10	
joints		6 / 9	
supple		6 / 8	
Top palabras exclusivas co-ocurrencia N = 0	Grado	Top palabras exclusivas Word2vec N = 120	Grado
		bowls	30
		wine	23
		sweet	21
		thirst	20
		pleasing	20
		draughts	18
		copious	17
		soft	17
		repast	16
		feast	16

Tabla B.16: Tabla n° 4 de asociaciones de comunidades k-core con pesos para La Ilíada

Matching: Jaccard Similarity = 0.400			
Top palabras intersección N = 188		Grado: Co-ocurrencia / Word2vec	
thy		212 / 71	
shall		201 / 136	
thou		195 / 66	
jove		191 / 49	
yet		190 / 78	
let		189 / 97	
troy		185 / 62	
thee		180 / 61	
war		180 / 62	
heaven		179 / 59	
Top palabras exclusivas co-ocurrencia N = 26	Grado	Top palabras exclusivas Word2vec N = 256	Grado
day	168	grecian	67
one	153	quit	59
soul	150	goddess	58
proud	121	hast	57
mighty	121	great	56
achilles'	102	deeds	55
wall	99	thus	54
warriors	95	hence	51
eternal	91	ask	51
late	86	worth	49

Tabla B.17: Tabla n° 5 de asociaciones de comunidades k-core con pesos para La Ilíada

Matching: Jaccard Similarity = 0.380			
Top palabras intersección N = 120		Grado: Co-ocurrencia / Word2vec	
ground		112 / 73	
head		111 / 66	
spear		110 / 49	
hand		109 / 70	
breast		109 / 60	
lance		104 / 54	
arm		103 / 39	
javelin		101 / 67	
wound		101 / 54	
fell		101 / 57	
Top palabras exclusivas co-ocurrencia N = 23	Grado	Top palabras exclusivas Word2vec N = 173	Grado
around	98	headlong	44
force	91	bone	42
deep	84	discharged	41
next	80	round	40
fury	77	corslet	40
trembling	77	steeds	40
dreadful	76	entrails	37
every	65	chariot	37
furious	62	solid	36
golden	62	waved	32

Tabla B.18: Tabla n° 6 de asociaciones de comunidades k-core con pesos para La Ilíada

Matching: Jaccard Similarity = 0.314			
Top palabras intersección N = 110		Grado: Co-ocurrencia / Word2vec	
o'er		131 / 105	
plain		108 / 60	
high		106 / 31	
skies		106 / 57	
along		101 / 97	
fields		91 / 93	
shore		91 / 36	
till		88 / 48	
ships		87 / 36	
way		84 / 29	
Top palabras exclusivas co-ocurrencia N = 29	Grado	Top palabras exclusivas Word2vec N = 211	Grado
round	108	troops	62
field	101	flight	49
swift	97	mountains	48
steeds	90	tumultuous	47
fierce	88	trembling	47
coursers	75	stand	39
walls	74	wood	39
chariot	70	hundred	37
night	65	dusky	37
light	63	rushing	37

Tabla B.19: Tabla n° 7 de asociaciones de comunidades k-core con pesos para La Ilíada

Matching: Jaccard Similarity = 0.500			
Top palabras intersección N = 7		Grado: Co-ocurrencia / Word2vec	
evippus		6 / 6	
polymelus		6 / 6	
ipheas		6 / 6	
echius		4 / 5	
amphoterus		4 / 5	
erymas		4 / 5	
epaltes		4 / 5	
Top palabras exclusivas co-ocurrencia N = 0	Grado	Top palabras exclusivas Word2vec N = 7	Grado
		nice	6
		matron	6
		poor	6
		exactness	6
		balance	6
		laborious	5
		woolly	5

Tabla B.20: Tabla n° 8 de asociaciones de comunidades k-core con pesos para La Ilíada

Matching: Jaccard Similarity = 0.000			
Sin nodos en la interseccion			
Top palabras exclusivas co-ocurrencia N = 4	Grado	Top palabras exclusivas Word2vec N = 15	Grado
deform	3	astyplus	7
lineaments	3	ænius	7
lifeblood	3	mydon	7
scarcely	3	thrasius	7
		mnesus	7
		slew	7
		things	6
		harpalion	6
		balmy	6
		beyond	6

Tabla B.21: Tabla n° 9 de asociaciones de comunidades k-core con pesos para La Ilíada

Matching: Jaccard Similarity = 0.810			
Top palabras intersección N = 17		Grado: Co-ocurrencia / Word2vec	
ois		10 / 16	
procophai		10 / 16	
en		14 / 16	
allos		9 / 16	
epitaeoeimasi		9 / 16	
kateschethaen		10 / 16	
de		10 / 15	
aesthomaen		10 / 14	
isos		10 / 14	
pleon		9 / 13	
Top palabras exclusivas co-ocurrencia N = 4	Grado	Top palabras exclusivas Word2vec N = 0	Grado
proionta	8		
aurel	7		
eupoia—to	6		
anton	6		

Tabla B.22: Tabla n° 10 de asociaciones de comunidades k-core con pesos para La Ilíada

B.0.3 Pinocho

Matching: Jaccard Similarity = 0.159			
Top palabras intersección N = 7		Grado: Co-ocurrencia / Word2vec	
lovely		13 / 6	
hair		11 / 7	
cry		8 / 10	
azure		10 / 8	
brilliant		9 / 6	
maiden		8 / 7	
wanted		7 / 7	
Top palabras exclusivas co-ocurrencia N = 15	Grado	Top palabras exclusivas Word2vec N = 22	Grado
geppetto	15	window	9
wood	12	grief	9
mastro	12	command	9
like	11	alabaster	9
upon	11	charming	9
made	10	ledge	9
sound	10	bemoan	8
color	10	reverence	8
piece	10	falcon	8
began	9	fairy	8

Tabla B.23: Tabla n° 1 de asociaciones de comunidades k-core con pesos para Pinocho

Matching: Jaccard Similarity = 0.018			
Top palabras intersección N = 1		Grado: Co-ocurrencia / Word2vec	
disappeared		13 / 6	
Top palabras exclusivas co-ocurrencia N = 32	Grado	Top palabras exclusivas Word2vec N = 23	Grado
little	30	chewed	11
man	28	straw	10
came	28	macaroni	8
poor	25	tossed	8
heard	23	tasted	8
voice	23	disappointment	8
suddenly	21	like	7
tiny	20	waves	7
turned	20	serve	6
walked	20	manger	6

Tabla B.24: Tabla n° 2 de asociaciones de comunidades k-core con pesos para Pinocho

Matching: Jaccard Similarity = 0.104			
Top palabras intersección N = 7		Grado: Co-ocurrencia / Word2vec	
legs		33 / 9	
mouth		33 / 8	
eyes		32 / 5	
open		25 / 7	
wide		22 / 7	
hard		18 / 5	
tongue		17 / 9	
Top palabras exclusivas co-ocurrencia N = 38	Grado	Top palabras exclusivas Word2vec N = 22	Grado
head	31	woe	13
long	30	grapes	10
door	29	pick	9
hands	28	tempted	9
put	27	thancrack	8
took	27	grapevine	8
pinocchio's	26	rattled	7
opened	26	tinkled	7
nose	25	joints	7
feet	24	sooner	7

Tabla B.25: Tabla n° 3 de asociaciones de comunidades k-core con pesos para Pinocho

Matching: Jaccard Similarity = 0.349			
Top palabras intersección N = 15		Grado: Co-ocurrencia / Word2vec	
sunday		13 / 16	
sure		15 / 13	
vacation		11 / 12	
saturdays		11 / 12	
begins		11 / 8	
saturday		8 / 10	
ends		10 / 8	
january		10 / 7	
composed		7 / 10	
six		7 / 9	
Top palabras exclusivas co-ocurrencia N = 14	Grado	Top palabras exclusivas Word2vec N = 14	Grado
first	13	fourteen	13
country	10	age	10
whitefish	9	eight	9
flounders	9	youngest	9
bass	9	blowing	9
place	8	street	8
december	8	racket	8
splendid	7	day	8
tasty	6	shouting	8
crabs	6	oldest	8

Tabla B.26: Tabla n° 4 de asociaciones de comunidades k-core con pesos para Pinocho

Matching: Jaccard Similarity = 0.087			
Top palabras intersección N = 9		Grado: Co-ocurrencia / Word2vec	
harlequin		24 / 16	
fire		21 / 14	
stage		11 / 19	
pulcinella		10 / 18	
eater		14 / 12	
turning		14 / 7	
forgives		9 / 6	
threatening		9 / 7	
usual		9 / 7	
Top palabras exclusivas co-ocurrencia N = 66	Grado	Top palabras exclusivas Word2vec N = 28	Grado
pinocchio	74	yelled	14
said	51	cooking	10
last	29	brother	9
ran	28	cheers	9
started	24	spit	9
friend	23	lamb	9
farmer	22	needed	9
look	21	signora	8
gave	20	reciting	8
done	18	side	8

Tabla B.27: Tabla n° 5 de asociaciones de comunidades k-core con pesos para Pinocho

Matching: Jaccard Similarity = 0.000			
Sin nodos en la interseccion			
Top palabras exclusivas co-ocurrencia N = 17	Grado	Top palabras exclusivas Word2vec N = 7	Grado
one	16	stuttering	6
could	15	remind	6
far	14	ofyes	6
away	14	look	6
went	13	becauseyou	6
water	12	likeyou	6
nothing	12	stammering	6
looked	12		
saw	12		
house	11		

Tabla B.28: Tabla n° 6 de asociaciones de comunidades k-core con pesos para Pinocho

Matching: Jaccard Similarity = 0.039			
Top palabras intersección N = 4		Grado: Co-ocurrencia / Word2vec	
never		61 / 7	
remember		55 / 6	
study		39 / 8	
friends		35 / 8	
Top palabras exclusivas co-ocurrencia N = 84	Grado	Top palabras exclusivas Word2vec N = 15	Grado
good	78	addio	8
fairy	76	says	8
come	72	proverb	8
see	71	'stolen	7
want	71	cared	7
i'll	70	bears	7
don't	70	false	7
boy	70	jot	7
go	68	success	6
answered	68	teacher	6

Tabla B.29: Tabla n° 7 de asociaciones de comunidades k-core con pesos para Pinocho

Matching: Jaccard Similarity = 0.009			
Top palabras intersección N = 1		Grado: Co-ocurrencia / Word2vec	
serpent		10 / 10	
Top palabras exclusivas co-ocurrencia N = 15	Grado	Top palabras exclusivas Word2vec N = 92	Grado
well	14	stomach	16
would	11	green	15
marionette	10	backs	15
laughing	10	thick	15
might	9	sea	14
laugh	8	monster	14
wall	8	dry	14
aside	8	thrown	14
asked	8	escape	13
dead	7	took	13

Tabla B.30: Tabla n° 8 de asociaciones de comunidades k-core con pesos para Pinocho

Matching: Jaccard Similarity = 0.078			
Top palabras intersección N = 5		Grado: Co-ocurrencia / Word2vec	
sold		18 / 6	
bought		10 / 11	
book		11 / 11	
sell		10 / 7	
abc		7 / 6	
Top palabras exclusivas co-ocurrencia N = 21	Grado	Top palabras exclusivas Word2vec N = 38	Grado
coat	17	arithmetic	17
donkey	17	volume	14
buy	14	i'll	12
circus	14	hatchet	11
owner	14	lot	10
market	13	clever	10
lame	12	money	10
soon	12	gain	9
new	10	text	9
wants	9	pages	9

Tabla B.31: Tabla n° 9 de asociaciones de comunidades k-core con pesos para Pinocho

Matching: Jaccard Similarity = 0.117			
Top palabras intersección N = 11		Grado: Co-ocurrencia / Word2vec	
fox		24 / 14	
cat		24 / 20	
left		21 / 9	
inn		20 / 14	
three		17 / 15	
next		16 / 6	
red		14 / 10	
eaten		13 / 14	
met		14 / 9	
alone		14 / 7	
Top palabras exclusivas co-ocurrencia N = 22	Grado	Top palabras exclusivas Word2vec N = 61	Grado
two	26	among	14
gold	22	paw	13
pieces	19	lame	12
night	17	shiver	11
morning	15	thought	10
many	15	one	10
four	14	laughingly	10
assassins	14	mere	10
let	14	table	10
minutes	14	time	9

Tabla B.32: Tabla n° 10 de asociaciones de comunidades k-core con pesos para Pinocho

B.0.4 Guerra y paz

Matching: Jaccard Similarity = 0.321			
Top palabras intersección N = 9		Grado: Co-ocurrencia / Word2vec	
farming		10 / 6	
wuined		5 / 8	
ou'		6 / 8	
bettah		6 / 8	
conscwiption		8 / 8	
o'		7 / 8	
neithah		4 / 6	
empia		4 / 6	
wetun		4 / 6	
Top palabras exclusivas co-ocurrencia N = 4	Grado	Top palabras exclusivas Word2vec N = 15	Grado
ermishin	4	twists	7
thrifty	4	ardently	7
matthew	4	smithereens	7
welltodo	4	desire'	7
		installs	7
		palace	7
		oxygen	6
		manures	6
		revolutions—in	6
		plows	6

Tabla B.33: Tabla n° 1 de asociaciones de comunidades k-core con pesos para Guerra y paz

Matching: Jaccard Similarity = 0.415			
Top palabras intersección N = 127		Grado: Co-ocurrencia / Word2vec	
prince		182 / 25	
said		182 / 48	
see		181 / 71	
well		181 / 88	
come		181 / 85	
thought		180 / 49	
go		180 / 45	
say		180 / 31	
know		180 / 56	
like		179 / 45	
Top palabras exclusivas co-ocurrencia N = 56	Grado	Top palabras exclusivas Word2vec N = 123	Grado
pierre	182	sorry	61
andrew	176	didn't	54
something	174	charming	53
never	173	doesn't	53
nicholas	173	you're	52
count	172	indeed	52
nothing	168	fool	49
look	168	visitor	47
us	167	devil	47
right	165	nice	47

Tabla B.34: Tabla n° 2 de asociaciones de comunidades k-core con pesos para Guerra y paz

Matching: Jaccard Similarity = 0.187			
Top palabras intersección N = 118		Grado: Co-ocurrencia / Word2vec	
power		113 / 191	
necessary		91 / 186	
could		139 / 166	
force		95 / 160	
new		103 / 156	
historians		66 / 149	
life		140 / 78	
present		81 / 136	
always		135 / 67	
history		99 / 135	
Top palabras exclusivas co-ocurrencia N = 27	Grado	Top palabras exclusivas Word2vec N = 486	Grado
would	143	never	166
one	142	everything	149
man	141	relation	117
time	139	need	109
without	134	military	107
even	127	importance	103
another	125	part	94
though	122	effect	94
words	117	real	94
feeling	116	important	93

Tabla B.35: Tabla n° 3 de asociaciones de comunidades k-core con pesos para Guerra y paz

Matching: Jaccard Similarity = 0.207			
Top palabras intersección N = 18		Grado: Co-ocurrencia / Word2vec	
de		18 / 58	
la		15 / 49	
que		16 / 45	
faire		10 / 43	
le		17 / 40	
vous		17 / 38	
des		11 / 37	
pas		15 / 34	
et		15 / 34	
à		13 / 32	
Top palabras exclusivas co-ocurrencia N = 2	Grado	Top palabras exclusivas Word2vec N = 67	Grado
les	11	l'angleterre	36
fait	11	extrémités	34
		j'explique	31
		reconnaissante	31
		grâce	30
		moscou	30
		moyens	28
		paraissent	27
		rentes	26
		faites	26

Tabla B.36: Tabla n° 4 de asociaciones de comunidades k-core con pesos para Guerra y paz

Matching: Jaccard Similarity = 0.255			
Top palabras intersección N = 75		Grado: Co-ocurrencia / Word2vec	
french		145 / 40	
army		142 / 86	
left		139 / 37	
russian		135 / 49	
battle		133 / 56	
troops		131 / 90	
kutúzov		131 / 26	
emperor		130 / 21	
position		130 / 79	
commander		128 / 68	
Top palabras exclusivas co-ocurrencia N = 73	Grado	Top palabras exclusivas Word2vec N = 146	Grado
moscow	139	captured	56
men	138	columns	55
two	138	guns	53
napoleon	138	officers	49
first	137	reported	47
day	137	center	44
whole	133	column	44
place	127	artillery	43
already	122	bagration's	43
last	119	suite	41

Tabla B.37: Tabla n° 5 de asociaciones de comunidades k-core con pesos para Guerra y paz

Matching: Jaccard Similarity = 0.000			
Sin nodos en la interseccion			
Top palabras exclusivas co-ocurrencia N = 7	Grado	Top palabras exclusivas Word2vec N = 15	Grado
hoch	6	buttermilk	13
kaiser	3	chicken	12
und	3	honey	12
russen	3	roasted	11
ganze	3	sweets	10
oestreicher	3	preserves	10
welt	3	nutandhoney	10
		jam	10
		cakes	9
		honeyandnut	8

Tabla B.38: Tabla n° 6 de asociaciones de comunidades k-core con pesos para Guerra y paz

Matching: Jaccard Similarity = 1.000			
Top palabras intersección N = 4		Grado: Co-ocurrencia / Word2vec	
mundi		3 / 3	
transit		3 / 3	
gloria		3 / 3	
sic		3 / 3	
Top palabras exclusivas co-ocurrencia N = 0	Grado	Top palabras exclusivas Word2vec N = 0	Grado

Tabla B.39: Tabla n° 7 de asociaciones de comunidades k-core con pesos para Guerra y paz

Matching: Jaccard Similarity = 0.000			
Sin nodos en la interseccion			
Top palabras exclusivas co-ocurrencia N = 6	Grado	Top palabras exclusivas Word2vec N = 12	Grado
pockets'	5	died	10
'will	5	robespierre	6
silent'	5	canonized	6
'vewy	5	beheaded	6
good'	4	martyr's	6
fill	4	alive—live	6
		sanctuary	5
		decay	5
		reek	5
		guarded	5

Tabla B.40: Tabla n° 8 de asociaciones de comunidades k-core con pesos para Guerra y paz

Matching: Jaccard Similarity = 0.094			
Top palabras intersección N = 6		Grado: Co-ocurrencia / Word2vec	
david		5 / 18	
goliath		5 / 14	
gideon		5 / 14	
midian		5 / 12	
moses		5 / 9	
amalek		5 / 9	
Top palabras exclusivas co-ocurrencia N = 0	Grado	Top palabras exclusivas Word2vec N = 58	Grado
		god	40
		thou	39
		thy	29
		art	28
		hast	25
		thee	23
		lord	21
		pebble	19
		destiny	18
		couldst	17

Tabla B.41: Tabla n° 9 de asociaciones de comunidades k-core con pesos para Guerra y paz

Matching: Jaccard Similarity = 0.000			
Sin nodos en la interseccion			
Top palabras exclusivas co-ocurrencia N = 6	Grado	Top palabras exclusivas Word2vec N = 269	Grado
tafa	5	house	95
safi	5	chapter	78
mala	5	nicholas	74
muter	5	last	67
kari	5	knew	67
kaská	5	pierre	66
		rostóvs	65
		day	61
		returned	60
		next	53

Tabla B.42: Tabla n° 10 de asociaciones de comunidades k-core con pesos para Guerra y paz

Matching: Jaccard Similarity = 0.647			
Top palabras intersección N = 11		Grado: Co-ocurrencia / Word2vec	
den		10 / 16	
gewiss		7 / 16	
nicht		6 / 16	
feind		9 / 16	
nur		8 / 16	
kann		8 / 16	
zweck		6 / 16	
ist		7 / 16	
schwächen		8 / 15	
der		8 / 15	
Top palabras exclusivas co-ocurrencia N = 0	Grado	Top palabras exclusivas Word2vec N = 6	Grado
		achtung	16
		privatpersonen	16
		nehmen	14
		verlust	14
		ja	14
		genug	14

Tabla B.43: Tabla n° 11 de asociaciones de comunidades k-core con pesos para Guerra y paz

Matching: Jaccard Similarity = 0.295			
Top palabras intersección N = 206		Grado: Co-ocurrencia / Word2vec	
eyes		233 / 160	
looked		232 / 152	
came		232 / 68	
face		229 / 47	
went		229 / 44	
back		226 / 204	
head		226 / 186	
still		224 / 69	
round		223 / 104	
began		223 / 77	
Top palabras exclusivas co-ocurrencia N = 39	Grado	Top palabras exclusivas Word2vec N = 453	Grado
away	220	shirt	124
saw	219	shoes	108
suddenly	202	chair	106
heard	198	armchair	106
seen	171	chest	102
passed	171	shed	93
house	171	seat	93
officers	170	waved	92
entered	167	occasionally	92
rode	161	rosy	92

Tabla B.44: Tabla n° 12 de asociaciones de comunidades k-core con pesos para Guerra y paz

Apéndice C

Tablas de asociaciones Node2vec

C.1 Alicia en el país de las maravillas

C.1.1 Red de co-ocurrencia

Matching: Jaccard Similarity = 0.368			
Top palabras intersección		Frecuencia	
little		127	
one		101	
went		83	
quite		55	
much		51	
way		50	
first		49	
head		46	
got		45	
get		45	
Top palabras exclusivas comunidad red co-ocurrencia	Frecuencia	Top palabras exclusivas cluster Node2vec	Frecuencia
put	31	alice	385
make	26	like	84
ever	22	could	77
find	21	time	68
talking	17	see	65
high	16	began	57
spoke	16	say	50
indeed	15	thing	49
garden	14	looked	45
saw	14	came	40

Tabla C.1: Tabla n° 1 de asociaciones entre comunidad de la red de co-ocurrencia y cluster construido en base a la red co-ocurrencia para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.061			
Top palabras intersección		Frecuencia	
live		8	
moral		7	
true		3	
bite		2	
flamingoes		2	
bird		2	
mustard		2	
experiment		2	
is—‘birds		1	
feather		1	
Top palabras exclusivas comunidad red co-ocurrencia	Frecuencia	Top palabras exclusivas cluster Node2vec	Frecuencia
thought	74	isn’t	7
see	65	anxious	3
go	50	hedgehogs	3
say	50		
never	45		
must	44		
oh	38		
i’ve	34		
that’s	33		
long	32		

Tabla C.2: Tabla n° 2 de asociaciones entre comunidad de la red de co-ocurrencia y cluster construido en base a la red co-ocurrencia para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.000			
Sin nodos en la intersección			
Top palabras exclusivas comunidad co-ocurrencia	Frecuencia red	Top palabras exclusivas cluster Node2vec	Frecuencia
brother's	1	every	12
mouse—o	1	tail	9
mouse—of	1	golden	7
grammar	1	grin	6
mouse—a	1	seems	5
mouse—to	1	spread	3
latin	1	gently	3
		doth	3
		neatly	2
		smiling	2

Tabla C.3: Tabla n° 3 de asociaciones entre comunidad de la red de co-ocurrencia y cluster construido en base a la red co-ocurrencia para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.239			
Top palabras intersección		Frecuencia	
know		81	
don't		60	
it's		54	
think		50	
cat		35	
i'll		29	
can't		28	
dear		28	
things		27	
won't		24	
Top palabras exclusivas comunidad red co-ocurrencia	Frecuencia	Top palabras exclusivas cluster Node2vec	Frecuencia
could	77	said	458
mouse	38	well	58
nothing	34	i'm	57
hear	14	go	50
talk	14	must	44
begin	13	come	43
always	12	tone	40
makes	11	oh	38
name	10	that's	33
keep	10	tell	31

Tabla C.4: Tabla n° 4 de asociaciones entre comunidad de la red de co-ocurrencia y cluster construido en base a la red co-ocurrencia para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.105			
Top palabras intersección		Frecuencia	
sort		20	
fact		8	
nice		6	
hot		5	
paws		4	
ventured		4	
fire		3	
purring		2	
nicely		2	
mixed		2	
Top palabras exclusivas comunidad red co-ocurrencia	Frecuencia	Top palabras exclusivas cluster Node2vec	Frecuencia
like	84	duchess	39
began	57	soup	17
thing	49	speak	14
tell	31	beautiful	13
heard	30	cook	13
might	28	waiting	9
sure	24	followed	8
something	18	soo—oop	7
eat	17	pepper	7
bit	16	middle	7

Tabla C.5: Tabla n° 5 de asociaciones entre comunidad de la red de co-ocurrencia y cluster construido en base a la red co-ocurrencia para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.338			
Top palabras intersección		Frecuencia	
yet		22	
grown		7	
youth		6	
mentioned		3	
pray		3	
allow		3	
sell		2	
locks		2	
jaws		2	
weak		2	
Top palabras exclusivas comunidad red co-ocurrencia	Frecuencia	Top palabras exclusivas cluster Node2vec	Frecuencia
use	18	old	19
turned	16	reason	9
kept	13	shook	9
grin	6	manage	7
seems	5	father	6
verses	4	case	5
letter	3	william	5
spread	3	none	4
muttering	3	perfectly	4
neatly	2	age	4

Tabla C.6: Tabla n° 6 de asociaciones entre comunidad de la red de co-ocurrencia y cluster construido en base a la red co-ocurrencia para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.234			
Top palabras intersección		Frecuencia	
hatter		55	
dormouse		39	
march		33	
hare		31	
may		13	
hurry		11	
tea		11	
repeated		10	
haven't		8	
asleep		8	
Top palabras exclusivas comunidad red co-ocurrencia	Frecuencia	Top palabras exclusivas cluster Node2vec	Frecuencia
said	458	getting	22
alice	385	'i	7
king	61	dare	5
well	58	breadandbutter	5
i'm	57	bottom	4
tone	40	twinkling	4
looking	30	questions	3
replied	29	fly	3
without	25	crown	3
rather	25	whiskers	3

Tabla C.7: Tabla n° 7 de asociaciones entre comunidad de la red de co-ocurrencia y cluster construido en base a la red co-ocurrencia para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.231			
Top palabras intersección		Frecuencia	
queen		68	
rabbit		43	
two		38	
next		30	
white		30	
three		26	
low		14	
left		14	
game		12	
many		12	
Top palabras exclusivas comunidad red co-ocurrencia	Frecuencia	Top palabras exclusivas cluster Node2vec	Frecuencia
came	40	thought	74
minute	21	king	61
face	14	voice	46
suddenly	13	never	45
whole	13	heard	30
gloves	10	make	26
footman	10	rather	25
waiting	9	day	24
running	8	ever	22
hearts	8	find	21

Tabla C.8: Tabla n° 8 de asociaciones entre comunidad de la red de co-ocurrencia y cluster construido en base a la red co-ocurrencia para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.181			
Top palabras intersección		Frecuencia	
english		6	
wanted		4	
cause		3	
earls		2	
conqueror		2	
mercia		2	
proceed		2	
ugh		2	
morcar		2	
whose		2	
Top palabras exclusivas comunidad red co-ocurrencia	Frecuencia	Top palabras exclusivas cluster Node2vec	Frecuencia
hedgehog	7	mouse	38
number	5	even	18
digging	4	dodo	13
besides	4	birds	9
solemnly	4	beg	8
move	3	party	8
wherever	2	lory	7
row	2	pardon	6
railway	2	late	6
provoking	1	lizard	5

Tabla C.9: Tabla n° 9 de asociaciones entre comunidad de la red de co-ocurrencia y cluster construido en base a la red co-ocurrencia para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.316			
Top palabras intersección		Frecuencia	
mock		56	
turtle		55	
gryphon		55	
cried		20	
far		13	
dance		13	
us		13	
sea		12	
used		12	
turn		10	
Top palabras exclusivas comunidad red co-ocurrencia	Frecuencia	Top palabras exclusivas cluster Node2vec	Frecuencia
time	68	replied	29
voice	46	course	24
looked	45	yes	12
come	43	creatures	10
last	33	interrupted	9
eyes	27	whiting	8
seemed	27	happened	7
caterpillar	27	history	7
good	24	kind	7
took	24	broken	6

Tabla C.10: Tabla n° 10 de asociaciones entre comunidad de la red de co-ocurrencia y cluster construido en base a la red co-ocurrencia para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.587			
Top palabras intersección		Frecuencia	
finished		12	
continued		9	
dry		8	
dish		4	
later		3	
panther		3	
turns		3	
growl		3	
owl		3	
toes		3	
Top palabras exclusivas comunidad red co-ocurrencia	Frecuencia	Top palabras exclusivas cluster Node2vec	Frecuencia
nose	7	talk	14
saucepan	1	eye	7
flew	1	marked	6
agony	1	pocket	5
terror	1	shriek	5
unusually	1	certain	3
precious	1	poison	2
		crash	2
		plates	2
		sooner	2

Tabla C.11: Tabla n° 11 de asociaciones entre comunidad de la red de co-ocurrencia y cluster construido en base a la red co-ocurrencia para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.165			
Top palabras intersección		Frecuencia	
passed		5	
fear		4	
jar		2	
underneath		2	
shelves		1	
labelled		1	
disappointment		1	
drop		1	
upstairs		1	
killing		1	
Top palabras exclusivas comunidad red co-ocurrencia	Frecuencia	Top palabras exclusivas cluster Node2vec	Frecuencia
great	39	somebody	7
getting	22	managed	4
gone	13	pictures	4
hadn't	8	cupboards	2
begun	7	pegs	1
almost	6	maps	1
breadandbutter	5	hung	1
outside	4	past	1
twinkling	4	bookshelves	1
mary	4		

Tabla C.12: Tabla n° 12 de asociaciones entre comunidad de la red de co-ocurrencia y cluster construido en base a la red co-ocurrencia para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.000			
Sin nodos en la intersección			
Top palabras exclusivas comunidad red co-ocurrencia	Frecuencia	Top palabras exclusivas cluster Node2vec	Frecuencia
end	18	garden	14
court	18	saw	14
silence	14	among	12
bill	12	chapter	11
read	10	hurried	11
hold	9	hall	9
book	7	hearts	8
roof	6	stood	7
shan't	6	queen's	7
rule	5	bright	7

Tabla C.13: Tabla n° 13 de asociaciones entre comunidad de la red de co-ocurrencia y cluster construido en base a la red co-ocurrencia para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.203			
Top palabras intersección		Frecuencia	
would		83	
change		13	
waving		5	
simple		5	
confused		4	
dull		3	
heart		2	
wind		2	
teacups		2	
rattling		2	
Top palabras exclusivas comunidad red co-ocurrencia	Frecuencia	Top palabras exclusivas cluster Node2vec	Frecuencia
duchess	39	hear	14
going	27	pool	10
even	18	queer	10
soup	17	keep	10
beautiful	13	feel	8
cook	13	shrill	5
baby	12	nearer	5
every	12	lives	4
word	10	days	4
child	10	grass	4

Tabla C.14: Tabla n° 14 de asociaciones entre comunidad de la red de co-ocurrencia y cluster construido en base a la red co-ocurrencia para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.221			
Top palabras intersección		Frecuencia	
rabbit		43	
white		30	
hurried		11	
fan		10	
gloves		10	
coming		8	
frightened		7	
kid		5	
pair		5	
tiny		4	
Top palabras exclusivas comunidad red Word2vec	Frecuencia	Top palabras exclusivas cluster Node2vec	Frecuencia
go	50	hand	20
get	45	witness	10
looked	45	read	10
garden	14	window	8
anxiously	14	hard	8
among	12	dropped	5
bill	12	walking	5
tea	11	nervous	5
small	10	violently	4
key	9	trumpet	3

Tabla C.15: Tabla n° 1 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas

C.1.2 Word2vec

Matching: Jaccard Similarity = 0.063			
Top palabras intersección		Frecuencia	
railway		2	
sand		1	
shiny		1	
pretending		1	
spades		1	
wooden		1	
machines		1	
lodging		1	
antipathies		1	
houses		1	
Top palabras exclusivas comunidad red Word2vec	Frecuencia	Top palabras exclusivas cluster Node2vec	Frecuencia
got	45	quite	55
things	27	mouse	38
make	26	ever	22
rather	25	find	21
people	13	curious	19
behind	13	talking	17
makes	11	indeed	15
children	10	gave	15
remark	10	seen	14
puzzled	9	gone	13

Tabla C.16: Tabla n° 2 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.015			
Top palabras intersección		Frecuencia	
pool		10	
and—oh		2	
rippling		1	
Top palabras exclusivas comunidad red Word2vec	Frecuencia	Top palabras exclusivas cluster Node2vec	Frecuencia
would	83	half	21
voice	46	give	12
perhaps	17	tears	11
knew	15	generally	7
hear	14	she's	7
low	14	hedgehog	7
change	13	neck	7
cats	13	nice	6
creatures	10	swam	5
keep	10	paws	4

Tabla C.17: Tabla n° 3 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.017			
Top palabras intersección		Frecuencia	
meet		2	
becoming		1	
Top palabras exclusivas comunidad red Word2vec	Frecuencia	Top palabras exclusivas cluster Node2vec	Frecuencia
first	49	great	39
want	9	put	31
william	5	took	24
judge	4	saw	14
crown	3	arm	12
hurriedly	2	trial	7
wig	2	somebody	7
'i'll	2	whispered	5
offer	2	seems	5
wasting	2	passed	5

Tabla C.18: Tabla n° 4 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.404			
Top palabras intersección		Frecuencia	
much		51	
dodo		13	
birds		9	
beg		8	
party		8	
lory		7	
late		6	
crowded		5	
animals		4	
fallen		4	
Top palabras exclusivas comunidad red Word2vec	Frecuencia	Top palabras exclusivas cluster Node2vec	Frecuencia
little	127	soon	25
round	40	even	18
found	31	puzzled	9
felt	23	english	6
asked	17	tale	4
idea	15	several	4
saying	15	duck	4
speak	14	wanted	4
thinking	11	cause	3
heads	10	caucusrace	3

Tabla C.19: Tabla n° 5 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.253			
Top palabras intersección		Frecuencia	
last		33	
till		21	
mouth		10	
mushroom		8	
arms		6	
quietly		5	
taking		5	
frowning		4	
managed		4	
folded		3	
Top palabras exclusivas comunidad red Word2vec	Frecuencia	Top palabras exclusivas cluster Node2vec	Frecuencia
size	13	eyes	27
somebody	7	caterpillar	27
surprised	7	bit	16
wood	7	waited	11
written	6	foot	10
passed	5	top	8
dare	5	sit	8
fear	4	open	7
hearing	4	ear	6
shoulders	4	drew	5

Tabla C.20: Tabla n° 6 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.165			
Top palabras intersección		Frecuencia	
day		24	
pigeon		12	
turn		10	
ready		8	
haven't		8	
you've		6	
you'll		6	
eggs		5	
nearer		5	
sleep		5	
Top palabras exclusivas comunidad red Word2vec	Frecuencia	Top palabras exclusivas cluster Node2vec	Frecuencia
like	84	said	458
think	50	mock	56
mouse	38	turtle	55
i've	34	gryphon	55
you're	23	first	49
tried	19	make	26
wonder	18	course	24
side	17	cried	20
talking	17	asked	17
seen	14	enough	17

Tabla C.21: Tabla n° 7 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.176			
Top palabras intersección		Frecuencia	
replied		29	
old		19	
us		13	
sea		12	
used		12	
though		11	
call		9	
learn		7	
repeat		7	
ah		5	
Top palabras exclusivas comunidad red Word2vec	Frecuencia	Top palabras exclusivas cluster Node2vec	Frecuencia
said	458	wasn't	11
mock	56	lessons	10
turtle	55	angrily	9
gryphon	55	william	5
thing	49	perfectly	4
come	43	fellow	4
back	38	repeating	3
caterpillar	27	courage	3
shall	25	part	2
won't	24	inside	2

Tabla C.22: Tabla n° 8 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.136			
Top palabras intersección		Frecuencia	
made		30	
another		22	
ran		16	
air		14	
hurry		11	
puppy		6	
jumped		6	
held		4	
stick		4	
delight		3	
Top palabras exclusivas comunidad red Word2vec	Frecuencia	Top palabras exclusivas cluster Node2vec	Frecuencia
great	39	next	30
large	33	three	26
moment	29	feet	19
take	21	high	16
please	19	behind	13
question	17	walked	10
bit	16	soldiers	10
silence	14	heads	10
still	13	gardeners	8
always	12	roof	6

Tabla C.23: Tabla n° 9 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.230			
Top palabras intersección		Frecuencia	
time		68	
hatter		55	
dormouse		39	
march		33	
hare		31	
looking		30	
court		18	
wouldn't		13	
room		13	
yes		12	
Top palabras exclusivas comunidad red Word2vec	Frecuencia	Top palabras exclusivas cluster Node2vec	Frecuencia
one	101	little	127
soon	25	went	83
wish	21	looked	45
high	16	long	32
suppose	14	without	25
grow	13	rather	25
begin	13	however	20
gone	13	please	19
wasn't	11	turned	16
growing	11	knew	15

Tabla C.24: Tabla n° 10 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.173			
Top palabras intersección		Frecuencia	
alice		385	
see		65	
say		50	
tell		31	
something		18	
sat		17	
eat		17	
ought		14	
better		14	
trying		13	
Top palabras exclusivas comunidad red Word2vec	Frecuencia	Top palabras exclusivas cluster Node2vec	Frecuencia
never	45	like	84
tone	40	could	77
heard	30	thought	74
dear	28	well	58
words	19	it's	54
lessons	10	thing	49
bottle	10	cat	35
help	9	might	28
different	9	poor	27
five	8	going	27

Tabla C.25: Tabla n° 11 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.126			
Top palabras intersección		Frecuencia	
away		25	
kept		13	
opportunity		8	
across		5	
flamingo		5	
arches		4	
plan		4	
difficulty		4	
doubt		4	
tucked		3	
Top palabras exclusivas comunidad red Word2vec	Frecuencia	Top palabras exclusivas cluster Node2vec	Frecuencia
began	57	one	101
way	50	round	40
poor	27	back	38
find	21	found	31
enough	17	seemed	27
generally	7	getting	22
hedgehog	7	anything	19
english	6	spoke	16
man	5	idea	15
walking	5	anxiously	14

Tabla C.26: Tabla n° 12 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.258			
Top palabras intersección		Frecuencia	
duchess		39	
look		28	
near		15	
alice's		11	
remarked		10	
chin		7	
feeling		7	
isn't		7	
moral		7	
sharp		6	
Top palabras exclusivas comunidad red Word2vec	Frecuencia	Top palabras exclusivas cluster Node2vec	Frecuencia
know	81	digging	4
don't	60	sounds	4
put	31	fond	3
right	31	flamingoes	2
i'll	29	tut	2
can't	28	attended	1
half	21	is—the	1
left	14	be'—or	1
remember	14	shape	1
dinah	11	burning	1

Tabla C.27: Tabla n° 13 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.163			
Top palabras intersección		Frecuencia	
queen		68	
head		46	
suddenly		13	
shouted		9	
knave		9	
play		8	
hearts		8	
wondering		7	
croquet		6	
procession		5	
Top palabras exclusivas comunidad red Word2vec	Frecuencia	Top palabras exclusivas cluster Node2vec	Frecuencia
well	58	would	83
came	40	began	57
next	30	moment	29
upon	26	question	17
three	26	side	17
good	24	called	15
ever	22	low	14
curious	19	face	14
feet	19	game	12
house	18	child	10

Tabla C.28: Tabla n° 14 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.246			
Top palabras intersección		Frecuencia	
yet		22	
use		18	
manage		7	
youth		6	
father		6	
trouble		6	
age		4	
longer		3	
mentioned		3	
pray		3	
Top palabras exclusivas comunidad red Word2vec	Frecuencia	Top palabras exclusivas cluster Node2vec	Frecuencia
oh	38	reason	9
long	32	shook	9
anything	19	grown	7
turned	16	none	4
hastily	16	smiling	2
gave	15	doors	2
close	13	conversations	2
finished	12	eyes—tell	1
life	11	fair	1
nearly	11	positively	1

Tabla C.29: Tabla n° 15 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.355			
Top palabras intersección		Frecuencia	
talk		14	
continued		9	
dry		8	
pocket		5	
dish		4	
later		3	
panther		3	
turns		3	
growl		3	
owl		3	
Top palabras exclusivas comunidad red Word2vec	Frecuencia	Top palabras exclusivas cluster Node2vec	Frecuencia
must	44	finished	12
eyes	27	bottle	10
name	10	eye	7
nose	7	marked	6
hair	7	broken	6
finger	5	certain	3
duck	4	box	3
repeating	3	poison	2
plate	3	drunk	2
outoftheway	3	crash	2

Tabla C.30: Tabla n° 16 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.309			
Top palabras intersección		Frecuencia	
two		38	
minute		21	
end		18	
soup		17	
beautiful		13	
cook		13	
every		12	
chapter		11	
word		10	
waiting		9	
Top palabras exclusivas comunidad red Word2vec	Frecuencia	Top palabras exclusivas cluster Node2vec	Frecuencia
seemed	27	get	45
sort	20	words	19
even	18	table	18
spoke	16	garden	14
game	12	speak	14
child	10	still	13
timidly	9	among	12
neck	7	many	12
sing	6	growing	11
lizard	5	small	10

Tabla C.31: Tabla n° 17 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.068			
Top palabras intersección		Frecuencia	
trees		7	
angry		5	
dead		4	
gently		3	
sad		3	
bark		2	
bank		2	
lonely		2	
wandering		2	
pat		2	
Top palabras exclusivas comunidad red Word2vec	Frecuencia	Top palabras exclusivas cluster Node2vec	Frecuencia
thought	74	came	40
quite	55	large	33
it's	54	heard	30
cat	35	door	28
nothing	34	upon	26
that's	33	house	18
sure	24	silence	14
hand	20	close	13
table	18	nearly	11
mad	14	opened	10

Tabla C.32: Tabla n° 18 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.043			
Top palabras intersección		Frecuencia	
king		61	
there's		24	
added		23	
doesn't		16	
jury		15	
explain		10	
rate		9	
meaning		8	
important		5	
paper		4	
Top palabras exclusivas comunidad red Word2vec	Frecuencia	Top palabras exclusivas cluster Node2vec	Frecuencia
didn't	14	know	81
beginning	13	don't	60
majesty	12	i'm	57
answer	9	go	50
interrupted	9	think	50
matter	9	way	50
eagerly	8	never	45
slates	7	got	45
write	6	must	44
shut	5	come	43

Tabla C.33: Tabla n° 19 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas

Matching: Jaccard Similarity = 0.251			
Top palabras intersección		Frecuencia	
baby		12	
sneezing		6	
knee		5	
notice		5	
shriek		5	
begins		4	
howling		3	
filled		3	
duchess's		3	
pause		2	
Top palabras exclusivas comunidad red Word2vec	Frecuencia	Top palabras exclusivas cluster Node2vec	Frecuencia
went	83	voice	46
could	77	change	13
i'm	57	queen's	7
might	28	shrill	5
door	28	surprise	5
going	27	suppressed	4
without	25	guineapigs	4
took	24	guests	3
however	20	unfortunate	3
called	15	around	3

Tabla C.34: Tabla n° 20 de asociaciones entre comunidad de la red de Word2vec y cluster construido en base a la red Word2vec para Alicia en el país de las maravillas