

# Tabla de Contenido

<b>1. Introducción</b>	<b>1</b>
1.1. Historia . . . . .	1
1.2. Estado del arte y ejemplos . . . . .	2
1.2.1. Estado del arte . . . . .	2
1.2.2. Ejemplos de aplicación . . . . .	3
1.3. Contribución . . . . .	4
1.4. Recursos disponibles en NLP . . . . .	4
1.4.1. <i>MOOC</i> . . . . .	4
1.4.2. Personas de importancia en el dominio . . . . .	4
1.4.3. Artículos . . . . .	5
1.4.4. Presentaciones . . . . .	5
1.4.5. Otros . . . . .	5
<b>2. Preliminares</b>	<b>6</b>
2.1. Herramientas básicas de NLP . . . . .	6
2.1.1. <i>Regular expression</i> . . . . .	6
2.1.2. Tokenización de palabras . . . . .	7
2.1.3. Identificación de las palabras de menor sentido: <i>stopwords</i> . . . . .	7
2.1.4. Normalización de palabras . . . . .	7
2.1.5. Segmentación de frases . . . . .	8
2.1.6. Distancia de edición mínima . . . . .	8
2.1.7. Estudio de grupos de palabras sucesivas: <i>n-grams</i> . . . . .	10
2.1.8. El análisis de sentimiento con <i>Bag-of-Words</i> . . . . .	10
2.2. El algoritmo <i>word2vec</i> . . . . .	11
2.2.1. ¿Cómo funciona <i>word2vec</i> ? . . . . .	12
2.2.2. Parámetros del algoritmo . . . . .	14
2.3. De la palabra al documento . . . . .	17
2.3.1. Otras posibilidades: más allá de <i>word2vec</i> . . . . .	17
2.3.2. <i>Latent Semantic Analysis</i> . . . . .	17
2.3.3. <i>Latent Dirichlet Allocation</i> . . . . .	18
2.4. Otros métodos usados en esta memoria . . . . .	19
2.4.1. Random Forests . . . . .	19
2.4.2. Cross-Validation . . . . .	19
2.4.3. <i>t-distributed Stochastic Neighbor Embedding</i> . . . . .	20
2.4.4. <i>Clustering</i> . . . . .	20
2.5. Implementaciones en Python . . . . .	20

<b>3. Ejemplo de uso de <i>word2vec</i>: detección de preguntas duplicadas en <i>Quora</i></b>	<b>21</b>
3.1. Presentación de la competencia . . . . .	22
3.2. Exploración de los datos . . . . .	22
3.2.1. Repartición de las clases . . . . .	22
3.2.2. Análisis preliminar de las preguntas . . . . .	23
3.3. El rol de <i>word2vec</i> . . . . .	28
3.4. Primer planteamiento del problema: un vector promedio por pregunta . . . .	29
3.4.1. De la necesidad ponderar las palabras por importancia: <i>TF-IDF</i> . . . .	30
3.4.2. Preprocesamiento de las preguntas . . . . .	31
3.4.3. Resultados . . . . .	32
3.5. Propuestas para mejorar el modelo . . . . .	33
3.5.1. Varios tipos de palabras: ¿cómo tomarlos en cuenta? . . . . .	34
3.5.2. Análisis de bi-grams . . . . .	34
3.6. Implementación en <i>Python</i> del modelo . . . . .	35
<b>4. Análisis literario</b>	<b>36</b>
4.1. Elección de los libros . . . . .	36
4.2. <i>Clusters</i> de palabras . . . . .	36
4.2.1. Ejemplo ilustrativo . . . . .	37
4.2.2. Comentarios . . . . .	42
4.2.3. Clustering . . . . .	42
4.2.4. Resultados . . . . .	42
4.3. Propuestas de mejoramiento de resultados . . . . .	44
4.3.1. Un método estadístico . . . . .	44
4.3.2. Lematización y círculos de vecinos palabras . . . . .	44
4.3.3. Ampliar el tamaño de los modelos . . . . .	45
<b>5. Conclusión</b>	<b>47</b>
5.1. Resumen del trabajo hecho . . . . .	47
5.2. Trabajo futuro . . . . .	48
<b>Glosario</b>	<b>49</b>
<b>Bibliografía</b>	<b>49</b>
<b>Anexos</b>	<b>51</b>

# Índice de Tablas

2.1.	Reglas generales en <i>Regex</i> . . . . .	6
2.2.	Operaciones necesarias para pasar de la palabra “ocurrencia” a “referencia”. Aquí notamos <i>s</i> por “substitución”. . . . .	8
2.3.	Varios alineamientos entre dos palabras. Aquí notamos <i>s</i> por “substitución” y <i>d</i> por “supresión”. . . . .	9
2.4.	Ejemplos de modelos e hipótesis correspondientes . . . . .	10
2.5.	Pares de palabras obtenidas en <i>word2vec</i> (extraído de Mikolov, Chen, et al. (2013), tabla 8) . . . . .	12
2.6.	Extracto del documento de test. Pares de vocabulario: nacionalidad y gentilicio	16
2.7.	Extracto del documento de test. Pares de verbos: gerundio y pretérito . . . . .	16
3.1.	Organización de los datos proveídos por <i>Quora</i> . Dos pares son distintos, un par es duplicado . . . . .	22
3.2.	Ejemplo de par de preguntas extraído del conjunto de entrenamiento, antes y después de transformarlo para <i>word2vec</i> . . . . .	26
3.3.	Palabras más parecidas a “ <i>Quora</i> ”, “ <i>Facebook</i> ”, y “ <i>government</i> ” . . . . .	27
3.4.	Palabras más parecidas a “ <i>man</i> ”, “ <i>woman</i> ”, y “ <i>house</i> ” en el modelo <i>Wikipedia</i>	28
3.5.	Extracto de pares ingenuos . . . . .	29
3.6.	Pares más sutiles: se diferencian por pocas palabras . . . . .	30
3.7.	Similitud y similitud ponderada usando <i>TF-IDF</i> . . . . .	31
3.8.	Resultados obtenidos con 300 árboles. Extraído de la salida del algoritmo <i>gridsearch</i> de <i>scikit-learn</i> . . . . .	32
3.9.	Falsos negativos más penalizadores . . . . .	32
3.10.	Falsos positivos más penalizadores . . . . .	33
4.1.	Criterio de dimensionalidad de los espacios . . . . .	39
4.2.	intersección de <code>mod[i].most_sim(“<i>homme</i>”)</code> para <code>i = [“Germinal”, “Le Rouge et le Noir”]</code> . . . . .	41
4.3.	intersección de <code>mod[i].most_sim(“<i>alcool</i>”)</code> para <code>i = [“Germinal”, “Le Rouge et le Noir”]</code> . . . . .	41
4.4.	Los ocho <i>clusters</i> que obtuvimos con el método de <i>clusters de palabras</i> . . . . .	43
5.1.	Lista de libros (1/2) . . . . .	55
5.2.	Lista de libros (2/2) . . . . .	56

# Índice de Ilustraciones

2.1.	Árbol para decidir si un punto constituye un fin de frase (FDF). Extraído y traducido de la primera lectura del ramo <i>CS 224N</i> de Stanford . . . . .	8
2.2.	Dos secuencias de ADN, cuya distancia está considerada desde el punto de vista de alineamiento. Extraído de la primera lectura del ramo <i>CS 224N</i> de Stanford . . . . .	9
2.3.	Modelos <i>CBOW</i> y <i>Skip-Gram</i> (extraído de Mikolov, Chen, et al. (2013), figura 1) . . . . .	15
2.4.	Clasificación de subpartes de imágenes. Extracto del <i>paper</i> “ <i>Describing visual scenes</i> ” . . . . .	19
3.1.	Número de caracteres en preguntas <i>vs</i> frecuencia . . . . .	23
3.2.	Número de palabras en preguntas <i>vs</i> frecuencia . . . . .	24
3.3.	Representación de los términos más frecuentes en las palabras, vía el paquete <i>WordCloud</i> . . . . .	25
3.4.	Representación mediante <i>t-SNE</i> de los vectores de palabras . . . . .	27
4.1.	Informaciones recuperadas al entrenar el algoritmo sobre <i>Germinal</i> . . . . .	40