ISSN 1870-4069

Captura de atributos discriminativos

Alberto Tapia-Palacios, Darnes Vilariño, Beatriz Beltrán,
María Josefa Somodevilla, José De Jesús Lavalle

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación, Puebla, México

alberto.tapia.p@gmail.com, {darnes, bbeltran, mariasg,
jlavalle}@cs.buap.mx

Resumen. En el siguiente artículo se describe como se resolvió la tarea 10 de SemEval 2018 llamada captura de atributos discriminativos, la cual consiste en encontrar la diferencia semántica entre la relación de dos conceptos como una característica. El modelo desarrollado se basa sobre el significado de la semántica y léxica de cada palabra para ampliar su conjunto de datos.

Palabras clave: Similitud semántica, semántica léxica, atributos discriminativos.

Capture of Discriminative Attributes

Abstract. This paper describes how the task 10 of SemEval 2018 called capture of discriminative attributes was solved, this consists to find the semantic difference between the relation of two concepts as a feature. The developed model is based on the meaning of the semantics and lexical of each word to expand the data set.

Keywords. Semantic similarity, lexical semantic, discriminative attributes.

1. Introducción

El avance de las tecnologías de hoy en día ha impulsado el desarrollo de mejorar la comunicación humano-computadora; además de que gran parte de la información se encuentra de forma digital en diferentes tipos de colecciones de datos, desde foros, *blogs*, *wikis* hasta redes sociales. Estas colecciones son inmensas, además de que se encuentran creciendo exponencialmente día a día gracias al impulso de internet, donde la información en su mayoría se encuentra sin ningún tipo de clasificación, generando que las investigaciones de la compresión y el uso de forma automática de los lenguajes naturales se vea incrementado en los últimos años.

Sin duda una tarea que es de las más vitales y con un alto grado de complejidad dentro del procesamiento del lenguaje natural (PLN) es la detección

de similitud semántica textual entre pares de sentencias. La semántica es la disciplina que estudia el significado de las expresiones lingüísticas, por lo que es común separar el estudio de significado de las palabras y de expresiones más complejas, lo que se distingue como semántica léxica y semántica composicional respectivamente [10]. La semántica léxica es el estudio de todo lo relativo al significado de las palabras lo que quiere decir que existe una colección de datos como un diccionario léxico que incluye el significado de las palabras de la lengua [1].

Faruqui, Tsvetkov, Rastogi y Dyer [3] describen algunos de los problemas más comunes con las tareas de tipo de evaluación de similitud para los modelos de vectores de palabras, sugiriendo que el uso de estos modelos sin supervisión puede conducir a resultados incorrectos, de manera que se requiere especial atención a la hora de evaluación y desarrollo de la tarea [2].

Una posible solución a este problema propuesta por Krebs y Papermo [5] consiste en extraer diferencias semánticas entre las palabras, al referirse a sus atributos que componen a cada palabra, de manera que una diferencia se puede expresar como la presencia o ausencia de un atributo en específico, así que ocupar atributos discriminativos puede brindar mejores resultados.

Tratando los atributos de un objeto o entidad como variable proporciona la ventaja de utilizar etiquetas de clasificación para diferenciar la información y poder predecir resultados más fiables [11,4,6].

El articulo está conformado por una sección de la descripción de la tarea, la siguiente sección trata acerca de la metodología, continuando con el análisis de resultados y finalmente tener la conclusión de la investigacón desarrollada.

2. Descripción de la tarea y conjunto de datos

La tarea de captura de atributos discriminativos consiste en una tarea de clasificación, que dado un par de palabras y un atributo discriminativo permite clasificar si tiene relación o no con la primera palabra. Entonces para cada par de palabras se verifica que la primera (pivote) tenga relación con el atributo proporcionado, pero la segunda (comparación) no tenga relación alguna con el atributo, para agregarlo a lista de ejemplos candidatos positivos.

- *palabra1* (pivote),
- *palabra2* (comparación),
- *atributo*,
- etiqueta (1 si el atributo es característica de *palabra1* pero no de *palabra2*, 0 en otro caso).

Un ejemplo de lo anterior seria *palabra1*: airplane (avión), *palabra2*: helicopter (helicóptero) y *atributo*: wings (alas), donde la *palabra1* tiene relación con el *atributo* alas, pero no la *palabra2*, ya que un helicóptero no tiene alas; para el caso de *palabra1*: helicóptero, *palabra2*: avión, *atributo*: alas se tomará en otra lista, ya que la *palabra1* no tiene relación con el *atributo*, se pueden ver algunos ejemplos en la Tabla 1. Para la lista de ejemplos negativos se tomarán en cuenta los casos donde ambas palabras tengan relación con el atributo y donde

las dos palabras no tienen relación alguna con el atributo, para este último caso la cantidad de ejemplos es muy grande.

Tabla 1. Algunos ejemplos del conjunto de datos.

Palabra1	Palabra2	Atributo
airplane	helicopter	wings
bagpipe	accordion	pipes
canoe	sailboat	fibreglass
dolphin	seal	fins
gorilla	crocodile	bananas
oak	pine	leaves
octopus	lobster	tentacles
pajamas	necklace	silk
skirt	jacket	pleats
subway	train	dirty

Dentro del conjunto de datos proporcionado se van a utilizar una parte para el entrenamiento (training), otra para la validación (validation) y finalmente para las pruebas (tests) para su desarrollo, esta información se encuentra detallada en la Tabla 2.

Tabla 2. Información acerca del conjunto de datos.

Conjunto de datos	Ejemplos	Negativos	Positivos	Atributos	Negativos	Positivos
Entrenamiento	17,782	11191	6591	1,292	1290	333
Validación	2,722	1358	1364	576	383	410
Pruebas	2,340			577		

Los conjuntos de datos proporcionados fueron de entrenamiento y validación, las pruebas para entrenar el modelo fueron con el conjunto de datos de entrenamiento y se ajustaron algunos parámetros con el conjunto de datos de validación, el conjunto de datos de pruebas no contiene etiquetas, este se tiene que enviar a la página de *CodaLab* para su evaluación. A contiuación, se explica la metodología utilizada.

3. Metodologia

Los resultados presentados para la tarea 10 de evaluación se obtuvieron a partir de un modelo propuesto de clasificación, el cual se basa sobre los

principios de semántica léxica descritos por Oana [1] y Escandell [10], asi como los problemas que mencionan Faruqui et al. [3].

Algo fundamental que utiliza el modelo propuesto son diccionarios de datos, que a partir del conjunto de entrenamiento se encarga de clasificar los datos para su correcto almacenamiento en los diccionarios asignados, como se mencionó en la sección de descripción de la tarea y conjunto de datos, estos tienen una etiqueta que los diferencia (0,1), donde 1 corresponde a que *palabra1* tiene relación con el *atributo* pero no la *palabra2*, sin embargo para cualquier otro caso la etiqueta es 0, lo que tiene como consecuencia inmediata las opciones donde *palabra1* y *palabra2* no tienen ninguna relación con el *atributo*, también donde la *palabra1* y el atributo no tienen relación alguna, asi como la situación donde la *palabra1* y *palabra2* están relacionadas con el *atributo*, dentro de este conjunto también se encuentran combinaciones clave, que si son ignoradas puede disminuir el resultado en la precisión del modelo.

Cabe hacer mención que durante la extracción de datos para el entrenamiento se encontró que había mucho ruido en los datos proporcionados por lo que no eran consistentes, es decir que para el conjunto de entrenamiento (training) se encuentran entradas que no pueden ser clasificadas correctamente, una de las razones es que su orden de magnitud es mayor, dado que la mayoría de los ejemplos son negativos, además de ser generado automáticamente por lo que algunas entradas pueden ser incorrectas, esto con el fin de tener un entrenamiento del sistema rico en parámetros [5].

Una vez que los diccionarios de datos contienen la información del conjunto de entrenamiento estos se utilizan de referencia para futuras predicciones del conjunto de pruebas, los diccionarios principalmente utilizados son los de las listas de palabras-atributos positivos (*dicPos*) y negativos (*dicNeg*), los cuales tienen una llave primaria (*Key*) conformada por una palabra y un atributo, esto con el fin de agilizar las búsquedas; también hay un control sobre las palabras a buscar con el diccionario *dicWeb*, este consiste en que al hacer una búsqueda de una palabra-atributo esta quede registrada y no se repita el proceso de clasificación, ver Tabla 3.

Tabla 3. Elementos utilizados.

Tipo	Lista	Key
Diccionario	positivo	Palabra, Atributo
Diccionario	negativo	Palabra, Atributo
Diccionario	Web	Palabra

Cada entrada del conjunto de pruebas se busca en los diccionarios creados a partir del conjunto de entrenamiento, para poder determinar si existe y si es un atributo discriminativo palabra-atributo en alguno de los diccionarios, en caso

contrario de no encontrar ninguna coincidencia en los diccionarios se procederá a realizar una búsqueda con los siguientes criterios:

- Significado de la palabra, la definición de la palabra con palabras de la misma lengua.
- Primer Sinónimo de la palabra.
- Hiperónimo de la palabra, es decir es la relación existente de una palabra cuyo significado engloba otras palabras, ejemplo Árbol.
- Hipónimos son aquellas palabras que señalan, de una manera. específica y precisa, a todos los seres que pertenecen al mismo conjunto, género o clase, ejemplo Álamo, roble, pino (son tipos de árboles).
- Miembros holónimos Son aquellas palabras que señalan el todo de una estructura, ejemplo bicicleta.
- Miembros merónimos son aquellas partes que representan algunas partes, pero no todas tienen el mismo tipo de cohesión con respecto al conjunto, ejemplo ruedas, manubrio, pedal (son partes de una bicicleta).

La búsqueda de los criterios anteriores devuelve un diccionario con el contenido encontrado, para el caso del significado de la palabra se eliminan las palabras vacías, este diccionario contiene la palabra de la búsqueda y sus atributos encontrados por el modelo, para posteriormente añadir las combinaciones a la lista positiva y ampliar el conjunto de datos lo más posible para futuras predicciones. En caso de no encontrar la combinación con los criterios descritos anteriormente, seguirá realizar una búsqueda en los sitios *dictionary* y *Wikipedia* con la ayuda de las herramientas *request* y *Wikipedia API*, los cuales amplían considerablemente la lista de elementos positivos, de igual manera los resultados solo envían palabra-atributo, eliminando las palabras vacías encontradas en la búsqueda; se agrega la palabra al diccionario *dicWeb* para no repetir nuevamente el proceso.

compareWorAtr Extrae todas las características (Atributos) de una palabra de acuerdo con el significado del diccionario del corpus de *WordNet*, elimina las palabras vacías y agrega sinónimos (actualiza el diccionario principal y muestra si tienen relación las palabras).

comparewordfeature Busca el significado de la palabra en sitios web de dictionary.com y wikipedia.org, elimina palabras vacías y devuelve un diccionario con la palabra y atributos encontrados.

A continuación, se presenta el algoritmo del modelo propuesto:

Algoritmo 1. Modelo propuesto.

```
Inicio:
 dicPos={} //tipo diccionario
 dicNeg={}
 dicWeb={}
 archivo=abrirArchivo(validation.txt)
 para linea en archivo leerlinea:
```

```
   //linea es de la forma palabra1,palabra2,atributo
   linea=separar(',') //ahora es palabra1, palabra2, atributo
   w1aPos=(existe (palabra1,atributo) en dicPos) //verdad ó falso
   w2aPos=(existe (palabra2,atributo) en dicPos)
   w1aNeg=(existe (palabra1,atributo) en dicNeg)
   w2aNeg=(existe (palabra2,atributo) en dicNeg)
   Si not(w1aPos): //w1a = falso
     w1a=compareWorAtr(palabra1,atributo)
     //buscar la palabra con requests y wikipedia api
     Si not(w1a) & not(existe (palabra1) en dicWeb):
       w1a,features=compare_word_feature(palabra1,atributo)
       dicPos = actualizaDiccionario(features)
       dicWeb[palabra1]=1 //fue buscado en web
       Si not(w1a) & not(existe (atributo) en dicWeb):
         w1a,features=compare_word_feature(atributo,palabra1)
         dicPos = actualizaDiccionario (features)
         dicWeb[atributo]=1 //fue buscado en web
   sino: //w1a = verdadero
     w1a=Verdadero
 Si not(w2aPos): //w2a = falso
   w2a=compareWorAtr(palabra2,atributo)
   Si not(w2a) & not(existe (palabra2) en dicWeb):
     w2a,features=compare_word_feature(palabra2,atributo)
     dicPos = actualizaDiccionario(features)
     dicWeb[palabra2]=1 //fue buscado en web
     Si not(w2a) & not(existe atributo en dicWeb):
       w2a,features=compare_word_feature(palabra2,atributo)
       dicPos = actualizaDiccionario (features)
       dicWeb[palabra2]=1 //fue buscado en web
 sino:
   w2a=Verdadero
 Si ((w1aPos & w2aNeg) || (w1aPos & not(w2a)) ||
     (w1a & w2aNeg) || (w1a & not(w2a)) ):
   imprimir("palabra1,palabra2,atributo,1") //Es atributo discriminativo
 sino: //w1aNeg || (w1aPos and w2aPos)
   imprimir("palabra1,palabra2,atributo,0") //No es atributo discriminativo
 archivo cerrarArchivo()
Fin
```

Para la implementación de lo mencionado anteriormente se utilizaron las herramientas y bibliotecas basadas en Python.

WordNet es una gran base de datos léxica de inglés. Los sustantivos, verbos, adjetivos y adverbios se agrupan en conjuntos de sinónimos cognitivos (synsets), cada uno expresando un concepto distinto. Los sintonizadores están interrelacionados por medio de relaciones semántico-conceptuales y léxicas. Utilizándose como un recurso principal para ampliar el conjunto de datos con los elementos mencionados anteriormente [8].

NLTK[1] es una plataforma que trabaja con datos de lenguaje humano que proporciona un corpus, junto con un conjunto de bibliotecas de procesamiento de texto para clasificación, tokenización, derivación, etiquetado, análisis y razonamiento semántico, envoltorios para bibliotecas de PLN, entre otros. Utilizándose como herramienta indispensable para implementar las funciones y corpus de *WordNet*, para la obtención de la lista de palabras vacías, sinónimos, hiperónimos, hiponimos por mencionar algunos[7].

Spacy[2] tiene modelos de etiquetado, análisis sintáctico y reconocimiento de entidades, se utilizó junto con el modelo "en vectors web lg" de lenguaje inglés que contiene 300 vectores dimensionales de palabras entrenadas en rastreo común con Glove[3] [9].

Textblob[4] proporciona métodos simples de implementar la integración de *WordNet*, así como etiquetado parcial, extracción de frase nominal, análisis de sentimiento, clasificación, traducción, por mencionar algunos.

Requests[5] es una biblioteca HTTP que se utilizó para obtener la información de una página web de manera simple.

BeautifulSoup[6] es una biblioteca para extraer datos de archivos html y xml. Se utilizó en conjunto con la biblioteca *Requests* para obtener solo la información necesaria de una página web, es decir solo su contenido relevante.

Wikipedia Api[7] obtiene el contenido de una consulta desde la página de Wikipedia, lo que ayuda a obtener las definiciones de las palabras o atributos que no encuentre el modelo, ampliando de manera considerable el conjunto de datos.

4. Análisis de resultados

Como mencionan Faruqui y Dyer [2] la evaluación de similitud semántica entre palabras supone que existe una sola noción de similitud, por lo que utilizar algunos de los diferentes modelos existentes mostrara resultados diferentes independientemente del problema a resolver, de igual manera no existe un estándar para la representación de palabras en vectores, lo que tiene como consecuencia que en algunos conjuntos de datos resulte difícil encontrar diferencias significativas.

El entrenamiento del modelo utilizo el conjunto de datos *training* por otra parte el conjunto de pruebas utiliza el conjunto de datos *validation*, en la Tabla 4

[1] http://www.nltk.org
[2] https://spacy.io/
[3] Algoritmo de aprendizaje no supervisado para obtener representaciones de vectores para palabras.
[4] https://textblob.readthedocs.io/en/dev/
[5] http://docs.python-requests.org/en/master/user/quickstart/
[6] https://www.crummy.com/software/BeautifulSoup/
[7] http://wikipedia-api.readthedocs.io/

se muestran los resultados obtenidos. Como se puede observar se hicieron pruebas con los modelos de similitud de:

WordNet La relación principal entre las palabras en WordNet es la sinonimia, los sinónimos (palabras que denotan el mismo concepto y son intercambiables en muchos contextos) se agrupan en conjuntos no ordenados (synsets). Cada uno de los 117 000 synsets de WordNet está vinculado a otros synsets por medio de un pequeño número de relaciones conceptuales.
Spacy 300 vectores de palabras tridimensionales entrenados en el rastreo común con Glove, con 1.1m de llaves y 1.1m de vectores únicos (300 dimensiones).

Los modelos anteriormente mencionados utilizan los mismos conjuntos de datos, con lo que muestran los resultados de 50.44 y 53.41 de precisión para NLTK y Spacy respectivamente, mientras que el modelo propuesto tiene un 62.41 de precisión, con esto se puede observar la mejora en los resultados obtenidos al utilizar atributos discriminativos en la tarea de similitud.

Tabla 4. Resultados *train - validation*.

Total del archivo						
	WordNet		**Spacy**		**Modelo**	
Validación	**Líneas**	**Porcentaje**	**Líneas**	**Porcentaje**	**Líneas**	**Porcentaje**
Total de líneas	2722		2722		2722	
Líneas positivas	1364	50.11	1364	50.11	1364	50.11
Líneas negativas	1358	49.89	1358	49.89	1358	49.89
Líneas erróneas						
	WordNet		**Spacy**		**Modelo**	
Validación	**Líneas**	**Porcentaje**	**Líneas**	**Porcentaje**	**Líneas**	**Porcentaje**
Líneas positivas (1,0)	1308	95.90	1189	87.18	901	66.06
Líneas negativas (0,1)	41	3.02	79	5.82	122	8.99
Total de líneas erróneas	1349	49.56	1268	46.59	1023	37.59
Líneas acertadas						
	WordNet		**Spacy**		**Modelo**	
Validación	**Líneas**	**Porcentaje**	**Líneas**	**Porcentaje**	**Líneas**	**Porcentaje**
Líneas positivas	56	4.10	175	12.82	463	33.94
Líneas negativas	1317	96.98	1279	94.18	1236	91.01
Total de líneas acertadas	1373	50.44	1454	53.41	**1699**	**62.41**

Para los resultados obtenidos con el modelo propuesto se utiliza el conjunto de entrenamiento *training* y *validation*, para el conjunto de pruebas se utiliza test, sin embargo, este no contiene las etiquetas de respuesta, por lo que se tiene que enviar a *CodaLab* para su evaluación en línea. Los resultados de la evaluación del modelo en la página de CodaLab[8] obtuvieron una calificación correcta de 0.63 de precisión.

5. Conclusión

En este artículo presentamos nuestro modelo como solución a la tarea de captura de atributos discriminativos entre pares de palabras, basado principal-

[8] `https://competitions.codalab.org/competitions/17326#results`

mente en la extracción de características según el significado o definiciones de una palabra, basado en semántica léxica. Los resultados obtenidos muestran que el tamaño del diccionario o conjunto de datos que utiliza cada modelo depende mucho de sus resultados, por ese motivo la precisión que resulta de cada uno varía según su tamaño y forma en que clasifica la similitud entre pares de palabras. De igual manera el conjunto de entrenamiento *training* es inconsistente, al ser generado automáticamente, además de tener una gran cantidad de ejemplos negativos, por lo cual el diccionario negativo es de extensas dimensiones. Se planea extender el modelo propuesto al añadir entrenamiento con vectores de palabras, con lo cual se espera aumentar el grado de precisión.

Referencias

1. Duţă, O.: Semántica léxica y oposiciones de sentido: un enfoque teórico. Annals of the University of Craiova. Series Philology. Linguistics. Recuperado de www. ceeol. com (2009)
2. Faruqui, M., Dyer, C.: Community evaluation and exchange of word vectors at wordvectors.org. In: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations. pp. 19–24. Association for Computational Linguistics (2014), http://www.aclweb.org/anthology/P14-5004
3. Faruqui, M., Tsvetkov, Y., Rastogi, P., Dyer, C.: Problems with evaluation of word embeddings using word similarity tasks. CoRR abs/1605.02276 (2016), http://arxiv.org/abs/1605.02276
4. Guo, Y., Ding, G., Jin, X., Wang, J.: Learning predictable and discriminative attributes for visual recognition. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence. pp. 3783–3789. AAAI'15, AAAI Press (2015), http://dl.acm.org/citation.cfm?id=2888116.2888241
5. Krebs, A., Paperno, D.: Capturing discriminative attributes in a distributional space: Task proposal. In: RepEval@ACL (2016)
6. Lazaridou, A., Pham, N.T., Baroni, M.: The red one!: On learning to refer to things based on their discriminative properties. CoRR abs/1603.02618 (2016), http://arxiv.org/abs/1603.02618
7. Loper, E., Bird, S.: Nltk: The natural language toolkit. In: Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1. pp. 63–70. ETMTNLP '02, Association for Computational Linguistics, Stroudsburg, PA, USA (2002), https://doi.org/10.3115/1118108.1118117
8. Miller, G.A.: Wordnet: A lexical database for english. COMMUNICATIONS OF THE ACM 38, 39–41 (1995)
9. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543 (2014), http://www.aclweb.org/anthology/D14-1162
10. Vidal, M.: Apuntes de semántica léxica. Cuadernos de la UNED, Universidad Nacional de Educación a Distancia (2007), https://books.google.com.mx/books?id=RWOGPgAACAAJ
11. Wang, Y., Mori, G.: A discriminative latent model of object classes and attributes. In: Proceedings of the 11th European Conference on Computer Vision: Part V. pp. 155–168. ECCV'10, Springer-Verlag, Berlin, Heidelberg (2010), http://dl.acm.org/citation.cfm?id=1888150.1888163