Desarrollo de un modelo para encontrar la similitud semántica multilingüe

Emanuel Aguilar, Darnes Vilariño, Claudia Zepeda,
Mireya Tovar, Beatriz Beltrán

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

{emanuel.aguilar.benitez}@hotmail.com, {dvilarinoayala, czepedac}@gmail.com,
{mtovar, bbeltran}@cs.buap.mx

Resumen. En el presente trabajo se desarrollan dos modelos para detectar el grado de similitud semántica entre pares de sentencias. El primer modelo está basado en aprendizaje supervisado, este utiliza un vector compuesto por dieciséis características para la representación de cada par de sentencias, con el que se entrena un clasificador. El segundo es un modelo no supervisado, el cual, basa su funcionamiento en la reconstrucción de una de las sentencias por medio de la otra, apoyándose para esto en los sinónimos de las palabras que las componen. Los dos modelos fueron probados para los idiomas inglés y español, y presentan un desempeño aceptable para ambos idiomas.

Palabras clave: Similitud semántica multilingüe, grafo de co-ocurrencia, intertextualidad, reconstrucción de sentencias.

1. Introducción

La similitud semántica tiene como objetivo determinar que tan semejantes son los sentidos de dos textos, y es por esto que se ha convertido en objeto de estudio durante muchos años dentro del área del Procesamiento de Lenguaje Natural (PLN). La similitud semántica cuenta con un amplio rango de aplicaciones, por ejemplo: máquinas de traducción, construcción automática de resúmenes, atribución de autoría, pruebas de lectura comprensivas, recuperación de información y muchas otras que necesitan medir el grado de similitud entre dos textos dados.

Existen diversos sistemas para la detección de similitud semántica, pero la mayoría sólo se enfocan en el idioma inglés. Por esta razón, se pretende el desarrollo de un modelo capaz de encontrar el grado de similitud semántica entre pares de sentencias que logre un buen desempeño en los idiomas inglés y español. El objetivo del modelo es dado un par de sentencias S_1 y S_2, ofrecer un valor entre 0 y 5 que represente el grado de similitud semántica entre ambas sentencias considerando los siguientes valores:

– 5 Las dos sentencias son completamente equivalentes, ya que significan lo mismo.

- 4 Las dos sentencias son en su mayoría equivalentes, pero difieren algunos detalles sin importancia.
- 3 Las dos sentencias son más o menos equivalentes, pero alguna información importante difiere o está ausente en una de las sentencias.
- 2 Las dos sentencias no son equivalentes, pero comparten algunos detalles.
- 1 Las dos sentencias no son equivalentes, pero tratan sobre el mismo tema.
- 0 Las dos sentencias tratan sobre temas distintos.

En el presente trabajo se presentan dos modelos que permiten detectar la similitud semántica entre pares de sentencias. Este artículo se encuentra estructurado de la siguiente manera: en la sección 2 se exponen algunos trabajos relacionados a este tema, en la sección 3 se presentan los modelos desarrollados para resolver la problemática que se plantea, posteriormente en la sección 4 se muestran los resultados obtenidos y por último en la sección 5 se abordan las conclusiones de la investigación.

2. Trabajo relacionado

Se han desarrollado diversas investigaciones para detectar el grado de similitud semántica entre pares de sentencias, a continuación se mencionan algunas asociadas a los trabajos presentados en el marco de la Conferencia SemEval 2013[1].
En el trabajo desarrollado en [1], se describe un sistema en el cual para estimar la similitud semántica entre dos enunciados se usan modelos de regresión que incluyen las siguientes características: n-gramas repetidos entre enunciados, similitud léxico-semántica entre palabras distintas, métricas de similitud de cadenas, similitud de contenido afectivo y longitud del enunciado. Todas estas características se combinan usando un modelo, ya sea de regresión múltiple lineal, o bien un modelo que detecta la relación que en cuanto a longitud poseen cada una de las sentencias. Dicha propuesta obtiene un 47% de aciertos reportados con los datos del SemEval 2013.

En el trabajo presentado en [2], se presenta un modelo de estimación lineal usando regresión *ridge*. Se analiza el comportamiento de la propuesta aplicando validación cruzada de 5 pliegues con los datos de entrenamiento. Esto permite afinar la sanción α que se emplea en la regresión *ridge*, con $\alpha \in 2^{\{-5,-4,\ldots,4\}}$. Dadas dos sentencias de entrada el sistema extrae: características de superposición de n-gramas, características de longitud y características de sentimientos. Posteriormente el sistema estandariza los valores de las características, substrayendo la media de las características y dividiendo por su desviación estándar. Se usan adaptaciones al dominio para facilitar la generalización a nuevos dominios. Este sistema presenta en promedio una tasa de aciertos del 45.03%.

El sistema que se describe en [3] desambigua el sentido de cada palabra en el enunciado empleando el contexto de cada una de ellas. La similitud de enunciados es calculada con el número de sentidos que comparten, es razonable asumir que enunciados similares deben tener más sentidos superpuestos. Para determinar la superposición del sentido se comparan las características de las palabras, primero se conceptualizan las palabras y después se calcula su similitud basándose en una estructura jerárquica, que a su vez se basa en WordNet; cada palabra en un enunciado

[1] http://www.cs.york.ac.uk/semeval-2013/index.php?id=tasks

es asignada a un sentido en el diccionario WordNet. Para reducir las limitaciones del diccionario, se utilizan los términos de los sentidos relacionados con la palabra objetivo, se asume que las palabras que co-ocurren en un enunciado comparten relación del sentido, y mientras más similares sean los enunciados compartirán más términos en las definiciones de sus palabras. Por ello no sólo se extraen los términos del enunciado principal, sino que también, para cada palabra del enunciado principal, se extraen los términos de su *hypernym, hyponym, meronym, holonym y troponym*; formando con esto el conjunto contexto. Finalmente se compara el contexto del enunciado con diferentes conjuntos de contextos para determinar cuál sentido debe ser asignado a las palabras. La implementación de este sistema reporta un 41.51% de aciertos.

Otro trabajo que es importante destacar es el propuesto en [4], que procesa características que se obtienen de diferentes bases de conocimiento como son WordNet, Wikipedia y Wiktionary. Las puntuaciones de similitud derivadas de estas características son introducidas dentro de varios perceptrones multicapa. Dependiendo del tamaño de los textos a comparar se usan diferentes parámetros para las redes neuronales; esto es que para cada grupo de longitud de enunciados, los pesos del perceptrón multicapa se calcularon separadamente. Cada perceptrón fue definido con 48 capas de entrada, correspondientes a las puntuaciones extraídas de las características, 4 capas ocultas y una capa de salida que representa la puntuación de similitud entre los enunciados. Además de las características que se obtienen de las bases de conocimiento aplicaron características de números y expresiones financieras y características de n-gramas. El sistema se desempeña bastante bien cuando los textos de entrada son cortos y de tamaños semejantes, pero cuando existe una variación considerable de tamaño entre estos, o son textos muy largos el desempeño del sistema cae considerablemente. La media de aciertos en este sistema es de 0.16.

Por último, en el sistema *Textual Similarity based on Lexical-Semantic Features* que se presenta en [5], se usan diferentes tipos de características léxicas y semánticas para entrenar un clasificador de bolsas de palabras que se utiliza para decidir la similitud entre los enunciados. Se implementaron tres variaciones del sistema cada una de estas variaciones utiliza un grupo particular de características. Cada par de enunciados es tokenizado, lematizado y post-etiquetado. Posteriormente varios métodos y algoritmos son aplicados para extraer todas las características necesarias para el sistema de máquina de aprendizaje. La primera variación llamada MultiSemLex, toma en cuenta todas las características extraídas y entrena un modelo con un clasificador de bolsas, la segunda variación llamada MultiLex y la tercera llamada MultiSem usan el mismo clasificador, pero incluyen diferentes características. MultiLex utiliza características extraídas de métricas léxico-semánticas y alineamiento léxico-semántico. Por otra parte, MultiSem, utiliza características extraídas solamente de alineación semántica. El sistema obtuvo un coeficiente de correlación general de 0.61.

Para esta investigación se proponen dos modelos, el primero basado en aprendizaje supervisado y un segundo modelo no supervisado. A continuación se presentan ambos modelos.

3. Metodología propuesta

Para resolver la problemática que se plantea en este artículo se desarrollaron dos modelos. El primer modelo extrae un vector de 16 características con el que se

representa cada par de sentencias, el cual es utilizado para entrenar un clasificador, ya sea máquina de soporte vectorial, Naïve Bayes o perceptrón simple; que posteriormente será empleado para determinar el grado de similitud semántica entre las sentencias. En el caso del segundo modelo, la similitud semántica se obtiene por medio de la reconstrucción de la sentencia de menor longitud, utilizando para esto la sentencia de mayor longitud. Para llevar a cabo los modelos se utilizaron las herramientas Clips Pattern[2], Network X[3], WordNet[4] y OpenThesaurus-es[5]. Ambos modelos se explican a continuación.

3.1. Modelo basado en aprendizaje supervisado

La primera parte de este modelo consiste en un pre-procesamiento de los datos de entrada. Este pre-procesamiento se encarga de sustituir los caracteres que no entran en la codificación ANSI por sus correspondientes, luego las sentencias se convierten a su representación en minúsculas, además se eliminan símbolos, signos de puntuación y las *stopwords* que se muestran en la Tabla 1, por ultimo se eliminan las palabras repetidas de cada sentencia.

Tabla 1. Stopwords a eliminar en el pre-procesamiento.

Idioma	Stopwords
Inglés	the, a, of, in, and, to, or, is, for, on, with, that, by, as, at, from, an, was, are, said, be, has, it, this, its, not, after, us, which, will, have, his, were, but, into, over, who, new, up, two, more, he, some, had, i, also, about, their, something, one, we, no, out, can, man, against, they, you, would, being, all, s, may
Español	de, y, la, en, el, que, a, del, los, un, se, por, es, para, una, con, las, al, o, su, como, más, no, entre, ha, fue, sus, desde, son, este, sobre, está, también, según

En la segunda parte de este modelo se obtienen las características de cada par de sentencias de entrada, con lo que se consigue el *Vector de Características*, que junto con su grado de similitud son utilizados por el clasificador para entrenar y así obtener el modelo utilizado para asignar un grado de similitud a las sentencias de prueba.

Por ser este un modelo de aprendizaje supervisado, la parte mas importante es la obtención de las características. En este modelo se proponen dos variaciones, la primera denominada mínimos y la segunda promedios, dado un par de sentencias S_1 y S_2, se extraen las características que se muestran en la Tabla 2, y con estas se forma el vector que representa a cada par de sentencias. Las primeras tres características (TTR[6], número de palabras en la sentencia y número de palabras clave[7] en la sentencia) son propias de cada sentencia, y sobre estas es calculado el mínimo o el promedio según sea el caso.

[2] http://www.clips.ua.ac.be/pattern
[3] https://networkx.github.io/
[4] https://wordnet.princeton.edu/
[5] http://openoffice-es.sourceforge.net/thesaurus/
[6] Type-Token Ratio, medida de variación léxica utilizada en varios tipos de análisis lingüísticos.
[7] Se consideran palabras clave todas aquellas que no son consideradas *stopwords* por la herramienta Clips Pattern.

Tabla 2. Características usadas para la representación de cada par de sentencias.

Variación Mínimos	**Variación Promedios**
TTR mínimo entre S_1 y S_2	TTR promedio de S_1 y S_2
Número mínimo de palabras en S_1 y S_2	Promedio de palabras en S_1 y S_2
Número mínimo de palabras clave en S_1 y S_2	Promedio de palabras clave en S_1 y S_2
Similitud coseno entre S_1 y S_2	
Similitud euclidiana entre S_1 y S_2	
Porcentaje de palabras clave que comparten S_1 y S_2	
Porcentaje de palabras clave que comparten S_1 y S_2 utilizando como *stemmer* el algoritmo de Porter	
Promedio de la intertextualidad entre S_1 y S_2	
Isomorfismo de las representaciones como grafos de S_1 y S_2(ancho de ventana igual a 1)	
Isomorfismo de las representaciones como grafos de S_1 y S_2(ancho de ventana igual a 2)	
Isomorfismo de las representaciones como grafos de S_1 y S_2(ancho de ventana igual a 1, sólo palabras clave)	
Isomorfismo de las representaciones como grafos de S_1 y S_2(ancho de ventana igual a 2, sólo palabras clave)	
Similitud Levenshtein entre S_1 y S_2	
Similitud Dice entre S_1 y S_2	
Distancia coseno entre las características[8] de S_1 y S_2	
Distancia euclidiana entre las características de S_1 y S_2	

En el caso de las características de isomorfismo el valor que pueden tomar es: 1 si las representaciones como grafo de co-ocurrencia de ambas sentencias son isomorfas, y 0 en caso contrario. La representación de cada sentencia por medio de un grafo de co-ocurrencia se llevó a cabo utilizando el siguiente algoritmo:

1. Se crea un grafo vacío no dirigido.
2. Se obtienen las palabras de la sentencia, o sólo las palabras clave, dependiendo el caso, estas pasarán a ser los nodos dentro del grafo.
3. Para cada palabra que se obtuvo se agrega una arista que una a esta palabra y a las siguientes *n* palabras, donde *n* es el valor del ancho de ventana.
4. Se remueven los auto-ciclos[9].

Como se puede notar, cada palabra distinta en la sentencia se convertirá en un nodo dentro del grafo, así bien los nodos se conectarán con otros nodos si las palabras que representan dichos nodos co-ocurren dentro del valor del ancho de la ventana.

3.2. Modelo basado en reconstrucción

El segundo modelo propuesto se basa en un algoritmo simple que no requiere entrenamiento, en este se determina la similitud semántica de dos sentencias por medio de la intertextualidad, que existe entre la sentencia que posee el menor número de palabras distintas y la reconstrucción de está utilizando la sentencia con mayor número de palabras distintas.

[8] TTR, número de palabras de la sentencia, número de palabras clave de la sentencia.
[9] Se considera auto-ciclo cuando una arista conecta a un nodo consigo mismo.

Este modelo está formado por cuatro etapas, tras las que una vez terminado el proceso se obtendrá un grado de similitud representado por un valor decimal entre cero y uno, donde, un valor igual a cero significa que las sentencias son totalmente distintas y un valor igual a uno que ambas sentencias representan el mismo concepto.

La primera etapa se encarga del pre-procesamiento de los datos de entrada, aquí las sentencias se convierten a su representación en minúsculas, posteriormente se eliminan los símbolos y signos de puntación, además de las palabras repetidas y las *stopwords* correspondientes al idioma de la sentencia (ver Tabla 1). En la segunda etapa se determina cual de las dos sentencias se intentará reconstruir, se propone que dicha sentencia sea la de menor longitud en términos de palabras distintas. La sentencia que se intentará reconstruir se conoce como *sentencia objetivo*.

En la tercera etapa, usando la sentencia de mayor longitud se intentará reconstruir la sentencia objetivo aplicando el siguiente algoritmo:

```
for w in Sm:
      if(w in So):
            Sr=Sr+w
      else:
            if(existeSinonimo(w,So)):
                  Sr=Sr+sinonimo(w,So)
```

donde:

w es una palabra que pertenece a *Sm*.
Sm representa la sentencia de mayor longitud.
So representa la sentencia objetivo.
Sr representa la sentencia resultado de la reconstrucción.
La función *existeSinonimo()* determina si está presente un sinónimo de *w* en *So*.
La función *sinonimo()* devuelve el sinónimo de *w* presente en *So*.

Para determinar la sinonimia entre las palabras se usó, WordNet para el idioma inglés, y para el idioma español se realizó una adaptación del OpenThesaurus-es.
Por último, en la cuarta etapa, se utiliza la intertextualidad para medir cuanto de la sentencia objetivo se pudo reconstruir en la sentencia resultado, el cálculo de la intertextualidad se realizó mediante la función *intertextuality()* que proporciona la herramienta Clips Pattern, tomando el resultado de esta operación como el grado de similitud semántica entre el par de sentencias de entrada.

4. Resultados

Durante el desarrollo y prueba de ambos modelos planteados, se han utilizado los corpus proporcionados en la tarea 10 (*Multilingual Semantic Textual Similarity*) de la conferencia SemEval 2014[10]. A continuación se describen los datos empleados durante la presente investigación.

4.1. Conjunto de datos

Se dispone de un corpus conformado por 6,627 pares de sentencias en idioma inglés, el cual se utiliza para el entrenamiento del primer modelo, tanto para el idioma inglés

[10] http://alt.qcri.org/semeval2014/task10/

como español. También se cuenta con dos corpus de prueba, el primero conformado por 3,000 pares de sentencias en idioma inglés y el segundo de 65 pares de sentencias en idioma español.

4.2. Resultados obtenidos

Los resultados obtenidos en las pruebas del modelo basado en aprendizaje supervisado se muestran en la Tabla 3. Cabe aclarar que los clasificadores usados fueron los que ofrece la herramienta Clips Pattern. Como se puede observar, en este modelo el mejor resultado para el idioma inglés se obtuvo por medio de la variación denominada promedios, utilizando como clasificador una máquina de soporte vectorial con un kernel lineal, logrando un porcentaje de 35.16% de aciertos. Por otra parte el mejor desempeño en el idioma español se dio utilizando la variación denominada mínimos junto con una máquina de soporte vectorial usando un kernel lineal obteniendo un 64.61% de aciertos.

Tabla 3. Resultados del modelo basado en aprendizaje supervisado.

Clasificador	Variación	Idioma	Aciertos
SVM kernel lineal	Mínimos	Inglés	34.30%
		Español	64.61%
	Promedios	Inglés	35.16%
		Español	52.30%
SVM kernel polinomial grado 2	Mínimos	Inglés	16.76%
		Español	0%
	Promedios	Inglés	16.76%
		Español	0%
SVM kernel polinomial grado 3	Mínimos	Inglés	16.76%
		Español	0%
	Promedios	Inglés	16.76%
		Español	0%
SVM kernel radial	Mínimos	Inglés	16.76%
		Español	0%
	Promedios	Inglés	16.76%
		Español	0%
Naïve Bayes	Mínimos	Inglés	29.73%
		Español	13.84%
	Promedios	Inglés	29.73%
		Español	13.84%
Perceptrón simple	Mínimos	Inglés	24.70%
		Español	01.53%
	Promedios	Inglés	27.66%
		Español	10.76%

En el caso del modelo basado en reconstrucción, tomando en cuenta que dicho modelo representa la similitud semántica con un valor en el rango [0,1], se ha

multiplicado el valor que retorna por cinco, el cual se comparó con los resultados proporcionados en los corpus de prueba.

Como se observa en la Tabla 4, este modelo presenta una mayor tasa de aciertos con respecto al primer modelo. En este segundo modelo, el porcentaje que se obtiene para el idioma inglés fue de 37.86%, mientras que para el idioma español se obtiene un porcentaje de 70.76%.

Tabla 4. Resultados de precisión para el modelo basado en reconstrucción.

Idioma	Relación de Aciertos	Porcentaje
Inglés	1136/3000	37.86%
Español	46/65	70.76%

5. Conclusión

En este artículo se presentan dos modelos para encontrar la similitud semántica multilingüe. Se observa que el modelo basado en la reconstrucción de las sentencias obtiene un mejor desempeño para ambos idiomas, en contraste con el modelo basado en aprendizaje supervisado. Se cree que los resultados para el primer modelo pudieran mejorar si se construyen representaciones por medio de grafos que incluyan los sinónimos de las palabras que componen a cada sentencia.

Con los resultados obtenidos en el modelo 1 queda claro que las características extraídas no logran descubrir realmente el grado de similitud entre las dos sentencias, pues se basan en la similitud entre las palabras que las componen, sin lograr descubrir el significado de cada una de ellas.

El segundo modelo tiene la ventaja de ser no supervisado, sin embargo, se considera que la diferencia tan grande entre el porcentaje de aciertos se debe a las características del conjunto de datos, pero con los experimentos desarrollados no se puede afirmar que ofrezca mayor precisión.

Se planea extender la reconstrucción de sentencias de manera que no solo busque sinónimos de palabras, si no también sinónimos sobre series de palabras, con lo cual se espera aumentar el grado de precisión.

Referencias

1. Malandrakis, N., Iosif, E., Prokopi, V., Potamianos, A., Narayanan, S.: Lexical, String and Affective Feature Fusion for Sentence-Level Semantic Similarity Estimation. In: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity, Vol. 1, pp. 103–108 (2013)
2. Heilman, M., Madnani, N.: Domain Adaptation and Stacking for Text Similarity. In: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity. Vol. 1, pp. 96–102 (2013)
3. Xu, J., Lu, Q.: Computing Semantic Textual Similarity using Overlapped Senses. In: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity, Vol. 1, pp. 90–95 (2013)
4. Ziak, H., Kern, R.: Semantic Text Similarity by use of Knowledge Bases. In: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity, Vol. 1, pp. 138–142 (2013)

5. Chávez, A., Fernández, A., Dávila, H., Gutiérrez, Y., Collazo, A., Abreu, J., Montoyo, A., Muñoz, R.: Textual Similarity Based on Lexical-Semantic Features. In: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity, Vol. 1, pp. 109–118 (2013)