TASS 2016: Workshop on Sentiment Analysis at SEPLN, septiembre 2016, pág. 41-45

Participación de SINAI en TASS 2016*

SINAI participation in TASS 2016

A. Montejo-Ráez
University of Jaén
23071 Jaén (Spain)
amontejo@ujaen.es

M.C. Díaz-Galiano
University of Jaén
23071 Jaén (Spain)
mcdiaz@ujaen.es

Resumen: Este artículo describe el sistema de clasificación de la polaridad utilizado por el equipo SINAI en la tarea 1 del taller TASS 2016. Como en participaciones anteriores, nuestro sistema se basa en un método supervisado con SVM a partir de vectores de palabras. Dichos vectores se calculan utilizando la técnicas de *deep-learning* Word2Vec, usando modelos generados a partir de una colección de tweets expresamente generada para esta tarea y el volcado de la Wikipedia en español. Nuestros experimentos muestran que el uso de colecciones de datos masivos de Twitter pueden ayudar a mejorar sensiblemente el rendimiento del clasificador.
Palabras clave: Análisis de sentimientos, clasificación de la polaridad, deep-learning, Word2Vec

Abstract: This paper introduces the polarity classification system used by the SINAI team for the task 1 at the TASS 2016 workshop. Our approach is based on a supervised learning algorithm over vectors resulting from a weighted vector. This vector is computed using a deep-learning algorithm called Word2Vec. The algorithm is applied so as to generate a word vector from a deep neural net trained over a specific tweets collection and the Spanish Wikipedia. Our experiments show massive data from Twitter can lead to a slight improvement in classificaciones accuracy.
Keywords: Sentiment analysis, polarity classification, deep learning, Word2Vec, Doc2Vec

1 Introducción

En este trabajo describimos las aportaciones realizadas para participar en la tarea 1 del taller TASS (Sentiment Analysis at global level), en su edición de 2016 (García-Cumbreras et al., 2016). Nuestra solución continúa con las técnicas aplicadas en el TASS 2014 (Montejo-Ráez, García-Cumbreras, y Díaz-Galiano, 2014) y 2015 (Díaz-Galiano y Montejo-Ráez, 2015), utilizando aprendizaje profundo para representar el texto y una colección de entrenamiento creada con tweets que contienen emoticonos que expresan emociones de felicidad o tristeza. Para ello utilizamos el método *Word2Vec*, ya que ha obtenido los mejores resultados en años anteriores. Por lo tanto, generamos un vector de pesos para cada palabra del tweet utilizando Word2Vec, y realizamos la media de dichos vectores para obtener una única representación vectorial. Nuestros resultados demuestran que el rendimiento del sistema de clasificación puede verse sensiblemente mejorado gracias a la introducción de estos datos en la generación del modelo de palabras, no así en el entrenamiento del clasificador de polaridad final.

La tarea del TASS en 2016 denominada *Sentiment Analysis at global level* consiste en el desarrollo y evaluación de sistemas que determinan la polaridad global de cada tweet del corpus general. Los sistemas presentados deben predecir la polaridad de cada tweet utilizando 6 o 4 etiquetas de clase (granularidad fina y gruesa respectivamente).

El resto del artículo está organizado de la siguiente forma. El apartado 2 describe el estado del arte de los sistemas de clasificación de polaridad en español. A continuación, se describe la colección de tweets con emoticonos utilizada para entrenar el clasificador. En el apartado 4 se describe el sistema desarro-

* Este estudio está parcialmente financiado por el proyecto TIN2015-65136-C2-1-R otorgado por el Ministerio de Economía y Competitividad del Gobierno de España.

llado y en el apartado 5 los experimentos realizados, los resultados obtenidos y el análisis de los mismos. Finalmente, en el último apartado exponemos las conclusiones y el trabajo futuro.

2 Clasificación de la polaridad en español

La mayor parte de los sistemas de clasificación de polaridad están centrados en textos en inglés, y para textos en español el sistema más completo, en cuanto a técnicas lingüísticas aplicadas, posiblemente sea *The Spanish SO Calculator* (Brooke, Tofiloski, y Taboada, 2009), que además de resolver la polaridad de los componentes clásicos (adjetivos, sustantivos, verbos y adverbios) trabaja con modificadores como la detección de negación o los intensificadores.

Los algoritmos de aprendizaje profundo (*deep-learning* en inglés) están dando buenos resultados en tareas donde el estado del arte parecía haberse estancado (Bengio, 2009). Estas técnicas también son de aplicación en el procesamiento del lenguaje natural (Collobert y Weston, 2008), e incluso ya existen sistemas orientados al análisis de sentimientos, como el de Socher et al. (Socher et al., 2011). Los algoritmos de aprendizaje automático no son nuevos, pero sí están resurgiendo gracias a una mejora de las técnicas y la disposición de grandes volúmenes de datos necesarios para su entrenamiento efectivo.

En la edición de TASS en 2012 el equipo que obtuvo mejores resultados (Saralegi Urizar y San Vicente Roncal, 2012) presentaron un sistema completo de pre-procesamiento de los tweets y aplicaron un lexicón derivado del inglés para polarizar los tweets. Sus resultados eran robustos en granularidad fina (65 % de accuracy) y gruesa (71 % de accuracy).

En la edición de TASS en 2013 el mejor equipo (Fernández et al., 2013) tuvo todos sus experimentos en el top 10 de los resultados, y la combinación de ellos alcanzó la primera posición. Presentaron un sistema con dos variantes: una versión modificada del algoritmo de ranking (RA-SR) utilizando bigramas, y una nueva propuesta basada en skipgrams. Con estas dos variantes crearon lexicones sobre sentimientos, y los utilizaron junto con aprendizaje automático (SVM) para detectar la polaridad de los tweets.

En 2014 el equipo con mejores resultados en TASS se denominaba ELiRF-UPV (Hurtado y Pla, 2014). Abordaron la tarea como un problema de clasificación, utilizando SVM. Utilizaron una estrategia uno-contra-todos donde entrenan un sistema binario para cada polaridad. Los tweets fueron tokenizados para utilizar las palabras o los lemas como características y el valor de cada característica era su coeficiente tf-idf. Posteriormente realizaron una validación cruzada para determinar el mejor conjunto de características y parámetros a utilizar.

El equipo ELiRF-UPV (Hurtado, Pla, y Buscaldi, 2015) volvió a obtener los mejores resultados en la edición de TASS 2015 con una técnica muy similar a la edición anterior (SVM, tokenización, clasificadores binarios y coeficientes tf-idf). En este caso utilizaron un sistema de votación simple entre un mayor número de clasificadores con parámetros distintos. Los mejores resultados los obtuvieron con un sistema que combinaba 192 sistemas SVM con configuraciones diferentes, utilizando un nuevo sistema SVM para realizar dicha combinación.

3 Colección de tweets con emoticonos

Los algoritmos de deep-learning necesitan grandes volúmenes de datos para su entrenamiento. Por ese motivo se ha creado una colección de tweets específica para la detección de polaridad. Para crear dicha colección se han recuperado tweets con las siguientes características:

- Que contengan emoticonos que expresen la polaridad del tweet. En este caso se han utilizado los siguientes emoticonos:
  - Positivos: `:)` `:-)` `:D` `:-D`
  - Negativos: `:(` `:-(`
- Que los tweets no contengan URLs, para evitar tweets cuyo contenido principal se encuentra en el enlace.
- Que no sean retweets, para reducir el número de tweets repetidos.

La captura de dichos tweets se realizó durante 22 días, del 18/07/2016 hasta el 9/08/2016, recuperando unos 100.000 tweets diarios aproximadamente. Tal y como se ve en la Figura 1 la recuperación fue muy homogénea y se obtuvieron más de 2.000.000 de tweets.

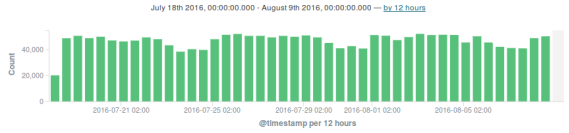

Figura 1: Número de tweets recuperados cada 12 horas

Posteriormente, se realizó un filtrado de dichos tweets eliminando aquellos que contubieran menos de 5 palabras, teniendo en cuenta que consideramos palabra todo término que sólo contenga letras (sin números, ni caracteres especiales).

Al final quedaron 1.777.279 clasificados según el emoticono que contienen de la siguiente manera:

- Positivos: 869.339 tweets
- Negativos: 907.940 tweets

Por último, se realiza la siguiente limpieza de tweets:

- Convertir el texto a minúsculas.
- Eliminar menciones (nombres de usuario que empiezan el caracter @).
- Sustituir letras acentuadas por sus versiones sin acentuar.
- Quitar las palabras vacías de contenido (*stopwords*).
- Normalizar las palabras para que no contengan letras repetidas, sustituyendo las repeticiones de letras contiguas para dejar sólo 3 repeticiones.

4 *Descripción del sistema*

Word2Vec[1] es una implementación de la arquitectura de representación de las palabras mediante vectores en el espacio continuo, basada en bolsas de palabras o n-gramas concebida por Tomas Mikolov et al. (Mikolov et al., 2013). Su capacidad para capturar la semántica de las palabras queda comprobada en su aplicabilidad a problemas como la analogía entre términos o el agrupamiento de palabras. El método consiste en proyectar las palabras a un espacio n-dimensional, cuyos pesos se determinan a partir de una estructura de red neuronal mediante un algoritmo recurrente. El modelo se puede configurar para que utilice una topología de bolsa de palabras (CBOW) o *skip-gram*, muy similar al anterior, pero en la que se intenta predecir los términos acompañantes a partir de un término dado. Con estas topologías, si disponemos de un volumen de textos suficiente, esta representación puede llegar a capturar la semántica de cada palabra. El número de dimensiones (longitud de los vectores de cada palabra) puede elegirse libremente. Para el cálculo del modelo Word2Vec hemos recurrido al software indicado, creado por los propios autores del método.

Tal y como se ha indicado, para obtener los vectores Word2Vec representativos para cada palabra tenemos que generar un modelo a partir de un volumen de texto grande. Para ello hemos utilizado los parámetros que mejores resultados obtuvieron en nuestra participación del 2014 (Montejo-Ráez, García-Cumbreras, y Díaz-Galiano, 2014). Por lo tanto, a partir de un volcado de Wikipedia[2] en Español de los artículos en XML, hemos extraído el texto de los mismos. Obtenemos así unos 2,2 GB de texto plano que alimenta al programa *word2vec* con los parámetros siguientes: una ventana de 5 términos, el modelo *skip-gram* y un número de dimensiones esperado de 300, logrando un modelo con más de 1,2 millones de palabras en su vocabulario.

Como puede verse en la Figura 2, nuestro sistema realiza la clasificación de los tweets utilizando dos fases de aprendizaje, una en la que entrenamos el modelo Word2Vec haciendo uso de un volcado de la enciclopedia on-line Wikipedia, en su versión en español, como hemos indicado anteriormente. De esta forma representamos cada tweet con el vector resultado de calcular la media de los vectores Word2Vec de cada palabra en el tweet y su desviación típica (por lo que cada vector de palabras por modelo es de 600 dimensiones). Se lleva a cabo una simple normalización previa sobre el tweet, eliminando repetición de letras y poniendo todo a minúsculas. La segunda fase de entrenamiento utiliza el algoritmo SVM y se entrena con la colección de tweets con emoticonos explicada en el apartado 3. La implementación de SVM utilizada es la basada en kernel lineal con entrenamiento SGD (Stochastic Gradient Descent) proporcionada por la biblioteca Sci-kit Learn[3] (Pedregosa et al., 2011).

Esta solución es la utilizada en las dos variantes de la tarea 1 del TASS con predicción

[1] `https://code.google.com/p/word2vec/`

[2] `http://dumps.wikimedia.org/eswiki`

[3] `http://scikit-learn.org/`

de 4 clases: la que utiliza el corpus de tweets completo (full test corpus) y el que utiliza el corpus balanceado (1k test corpus).

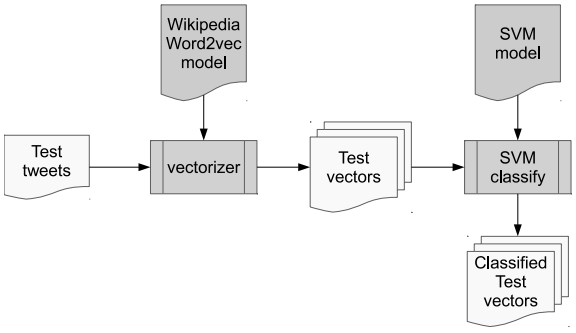

Figura 2: Flujo de datos del sistema completo

5 Resultados obtenidos

Hemos experimentado con el efecto que tienen en el rendimiento del sistema el uso de una colección de datos generada a partir de la captura de tweets y que han sido etiquetados según los emoticonos que contienen en la forma comentada anteriormente. La colección de más de 1,7 millones de tweets ha sido utilizada al completo para generar un modelo de vectores de palabras, cuya combinación con el de Wikipedia se ha analizado. También hemos comprobado cómo el uso de dicha colección de tweets afecta cuando se usa para el entrenamiento del modelo de clasificación de la polaridad. Para ello se han seleccionado 500,000 tweets aleatoriamente de esta colección, con sus correspondientes etiquetas P (positivo) o N (negativo) y se han combiando con la colecciónd de entrenamiento de TASS.

Los resultados según las medidaas de *Accuracy* y *Macro F1* obtenidas se muestran en la tabla 1. La primera columna nos indica a partir de cuáles datos se han generado los modelos de vectores de palabras, bien sólo con Wikipedia (W) o como combinación de ésta con los tweets del corpus construido (W+T). La segunda columna indica cómo se ha entrenado el clasificador de polaridad a partir de los textos etiquetados vectorizados con los modelos generados en el paso previo, bien sólo usando los datos de entrenamiento proporcionados por la organizacion (TASS) o incorporando los etiquetados a partir de emoticonos (TASS+T).

Como podemos observar, el uso de una colección de tweets para ampliar la capacidad de representar un modelo basado en vectores de palabras mejora sensiblemente al generado solamente con Wikipedia, pasando de 61,31 % de ajuste a un 62,39 %. En cambio, utilizar los tweets capturados para la fase de entrenamiento supervisado no lleva sino a una caída del rendimiento del sistema.

Tabla 1: Resultados obtenidos sobre el conjunto *full*

w2v	**SVM**	**Accuracy**	**Macro**-F1
W	TASS	61,31 %	48,55 %
W+T	TASS	62,39 %	50,44 %
W	TASS+T	49,28 %	40,20 %
W+T	TASS+T	53,72 %	44,10 %

Esto nos lleva a plantearnos la pregunta de qué ocurriría si utilizáramos sólo los tweets recopilados para generar un modelo de vectores de palabras. Los resultados que se obtienen son un 59,05 % de ajuste y un 44,43 % de F1. No cabe duda de que conviene explorar el uso de modelos de generación de características a partir de vectores de palabras.

Estos resultados mejoran nuestros datos del año pasado, en los que obtuvimos un ajuste del 61,19 % combinando vectores de palabras (Word2Vec) y vectores de documentos (Doc2Vec).

6 Conclusiones y trabajo futuro

A partir de los resultados obtenidos, encontramos que resulta interesante la incorporación de texto no formal (tweets) para la generación de los modelos de palabras, lo cual tiene su sentido en una tarea de clasificación que, precisamente, trabaja sobre textos no formales que tienen la misma red social como fuente. En cambio, el considerar que los emoticonos en un tweet pueden ayudar a un clasificador como SVM a mejorar en la determinación de la polaridad ha resultado una hipótesis fallida. Esto puede entenderse echando un vistazo a algunos de los tweets capturados por el sistema, donde se evidencia la dificultad, incluso para una persona, de poner en contexto el sentido del tweet y su consideración como positivo o negativo si no disponemos de un emoticono asociado.

Como trabajo futuro nos proponemos diseñar una red neuronal profunda más elaborada, pero que parta también de textos de entrenamiento tanto formales como no formales, si bien teniendo en cuanta información lingüística más avanzada como la sintáctica, en lugar de trabajar con simples bolsas de palabras. También queremos explorar el uso

de redes de este tipo en el proceso de clasficación en sí, y no sólo en la generación de características. Una posibilidad es utilizar una red de tipo DBN (Deep Belief Network) (Hinton y Salakhutdinov, 2006) en la que se añade una última fase donde se realiza el etiquetado de los ejemplos.

Bibliografía

Bengio, Yoshua. 2009. Learning deep architectures for ai. *Foundations and trends in Machine Learning*, 2(1):1–127.

Brooke, Julian, Milan Tofiloski, y Maite Taboada. 2009. Cross-linguistic sentiment analysis: From english to spanish. En Galia Angelova Kalina Bontcheva Ruslan Mitkov Nicolas Nicolov, y Nikolai Nikolov, editores, *RANLP*, páginas 50–54. RANLP 2009 Organising Committee / ACL.

Collobert, Ronan y Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. En *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, páginas 160–167, New York, NY, USA. ACM.

Díaz-Galiano, M.C. y A. Montejo-Ráez. 2015. Participación de SINAI DW2Vec en TASS 2015. En *In Proc. of TASS 2015: Workshop on Sentiment Analysis at SEPLN. CEUR-WS.org*, volumen 1397.

Fernández, Javi, Yoan Gutiérrez, José M. Gómez, Patricio Martínez-Barco, Andrés Montoyo, y Rafael Muñoz. 2013. Sentiment analysis of spanish tweets using a ranking algorithm and skipgrams. En *In Proc. of the TASS workshop at SEPLN 2013*.

García-Cumbreras, Miguel Ángel, Julio Villena-Román, Eugenio Martínez-Cámara, Manuel Carlos Díaz-Galiano, Mª. Teresa Martín-Valdivia, y L. Alfonso Ureña-López. 2016. Overview of tass 2016. En *Proceedings of TASS 2016: Workshop on Sentiment Analysis at SEPLN co-located with the 32nd SEPLN Conference (SEPLN 2016)*, Salamanca, Spain, September.

Hinton, Geoffrey E y Ruslan R Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.

Hurtado, Lluís F y Ferran Pla. 2014. Elirf-upv en tass 2014: Análisis de sentimientos, detección de tópicos y análisis de sentimientos de aspectos en twitter. En *In Proc. of the TASS workshop at SEPLN 2014*.

Hurtado, Lluís-F, Ferran Pla, y Davide Buscaldi. 2015. Elirf-upv en tass 2015: Análisis de sentimientos en twitter. En *In Proc. of TASS 2015: Workshop on Sentiment Analysis at SEPLN. CEUR-WS.org*, volumen 1397, páginas 35–40.

Mikolov, Tomas, Kai Chen, Greg Corrado, y Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Montejo-Ráez, A., M.A. García-Cumbreras, y M.C. Díaz-Galiano. 2014. Participación de SINAI Word2Vec en TASS 2014. En *In Proc. of the TASS workshop at SEPLN 2014*.

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, y others. 2011. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830.

Saralegi Urizar, Xabier y Iñaki San Vicente Roncal. 2012. Tass: Detecting sentiments in spanish tweets. En *TASS 2012 Working Notes*.

Socher, Richard, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, y Christopher D. Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. En *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, páginas 151–161, Stroudsburg, PA, USA. Association for Computational Linguistics.