TASS 2015, septiembre 2015, pp 59-64 recibido 20-07-15 revisado 24-07-15 aceptado 29-07-15

Participación de SINAI DW2Vec en TASS 2015*

SINAI DW2Vec participation in TASS 2015

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

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

Resumen: Este artículo describe el sistema de clasificación de la polaridad utilizado por el equipo SINAI-DW2Vec en la tarea 1 del workshop TASS 2015. Nuestro sistema no sigue el modelo espacio vectorial clásico ni aplica análisis sintáctico o léxico alguno. Nuestra solución se basa en un método supervisado con SVM sobre vectores de pesos concatenados. Dichos vectores se calculan utilizando dos técnicas: Word2Vec y Doc2Vec. La primera obtiene la sumatoria de vectores de palabras con un modelo generado a partir de la Wikipedia en español. Con la técnica Doc2Vec se generan vectores de características a partir de la colección de tweets de entrenamiento, en este caso a nivel de párrafo (o tweet) en lugar de a nivel de palabra como lo hace Word2Vec. La experimentación realizada demuestra que ambas técnicas conjuntas consiguen mejorar al uso de cada técnica por separado.
Palabras clave: Análisis de sentimientos, clasificación de la polaridad, deep-learning, Word2Vec, Doc2Vec

Abstract: This paper introduces the polarity classification system used by the SINAI-DW2Vec team for the task 1 at the TASS 2015 workshop. Our approach does not follow the Vector Space Model nor applies syntactic or lexical analyses. This solution is based on a supervised learning algorithm over vectors resulting from concatenating two different weighted vectors. Those vectors are computed using two different, yet related, algorithms: Word2Vec and Doc2Vec. The first algorithm is applied so as to generate a word vector from a deep neural net trained over Spanish Wikipedia. For Doc2Vec, the vector is generated with paragraph vectors (instead of word vectors) from a neural net trained over the tweets of the training collection. The experiments show that the combination of both vector distributions leads to better results rather than using them isolated.
Keywords: Sentiment analysis, polarity classification, deep learning, Word2Vec, Doc2Vec

1 Introducción

En este artículo describimos el sistema construido para participar en la tarea 1 del workshop TASS (Sentiment Analysis at global level), en su edición de 2015 (Villena-Román et al., 2015). Nuestra solución continúa con las técnicas aplicadas en el TASS 2014, utilizando aprendizaje profundo para representar el texto, y dando un paso más generando una representación no sólo a nivel de palabras sino también de frases o documentos. Para ello utilizamos el método *Word2Vec* utilizado con buenos resultados el año anterior, junto con la técnica *Doc2Vec* que nos permite representar un trozo variable de texto, por ejemplo una frase, en un espacio n-dimensional. Por lo tanto, utilizando Word2Vec generamos un vector para cada palabra del tweet, y realizamos la media de dichos vectores para obtener una única representación con Word2Vec. A dicho vector le concatenamos el vector obtenido con el modelo Doc2Vec, para generar una única representación del tweet. Una vez obtenidos los vectores de todos los tweets utilizamos un proceso de aprendizaje supervisado, a partir del conjunto de entrenamiento facilitado por la organización y el algoritmo SVM. Nuestros resultados demuestran que el uso conjunto de

* Esta investigación ha sido subvencionada parcialmente por el proyecto del gobierno español ATTOS (TIN2012-38536-C03-0), por la Comisión Europea bajo el Séptimo programa Marco (FP7 - 2007-2013) a través del proyecto FIRST (FP7-287607).

ambas técnicas mejora los resultados obtenidos utilizando sólo una de las técnicas presentadas.

Estos experimentos se presentan al amparo del TASS (Taller de Análisis de Sentimientos en la SEPLN), que es un evento satélite del congreso SEPLN, que nace en 2012 con la finalidad de potenciar dentro de la comunidad investigadora en tecnologías del lenguaje (TLH) la investigación del tratamiento de la información subjetiva en español. En 2015 se vuelven a proponer los mismos dos objetivos que en la convocatoria anterior. Por un lado observar la evolución de los sistemas de análisis de sentimientos, y por otro lado evaluar sistemas de detección de polaridad basados en aspectos.

La tarea del TASS en 2015 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 capítulo 2 describe el estado del arte de los sistemas de clasificación de polaridad en español. En el capítulo 3 se describe el sistema desarrollado y en el capítulo 4 los experimentos realizados, los resultados obtenidos y el análisis de los mismos. Finalmente, en el último capítulo 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 relevante 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). Otros sistemas, compararon diferentes técnicas de clasificación (Fernández Anta et al., 2012) implementadas en WEKA (Hall et al., 2009), o trataron la clasificación de forma binaria (Batista y Ribeiro, 2012), lanzando en paralelo distintos clasificadores binarios y combinando posteriormente los resultados. También se utilizó *naive-bayes multinomial* para construir un modelo del lenguaje (Trilla y Alías, 2012), un lexicón afectivo para representar el texto como un conjunto de emociones (Martín-Wanton y Carrillo de Albornoz, 2012), recuperación de información (RI) basado en divergencia del lenguaje para generar modelos de polaridad (Castellanos, Cigarrán, y García-Serrano, 2012), y un enfoque basado en el recurso léxico Sentitext, asignando una etiqueta de polaridad a cada término encontrado (Moreno-Ortiz y Pérez-Hernández, 2012).

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 alcanzaron 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. Otro equipo (Martínez Cámara et al., 2013) optó por una estrategia complemente no supervisada, frente a la supervisada desarrollada en 2012. Usaron como recursos lingüísticos SentiWordNet, Q-WordNet y iSOL, combinando los resultados y normalizando los valores.

3 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 cáculo del modelo Word2Vec hemos recurrido al software indicado, creado por los propios autores del método.

Basándose en Word2Vec, Le y Mikolov crearon el modelo Doc2Vec (Le y Mikolov, 2014). Este nuevo modelo calcula directamente un vector para cada párrafo o trozo de texto de longitud variable. El sistema para calcular dichos vectores es similar a Word2Vec, con la salvedad de que el contexto de cada palabra es inicializado en cada frase. Al igual que Word2Vec también existen dos topologías para dichos contextos de la palabras, bolsa de palabras distribuida (DCBOW) o memoria distribuida (DM - *Distributed Memory*).

Para calcular y utilizar el modelo Doc2Vec se ha utilizado una biblioteca para Python, denominada *gensim*[2]. Esta biblioteca también nos permite trabajar con el modelo Word2Vec generado anteriormente.

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[3] 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 200, logrando un modelo con más de 1,2 millones de palabras en su vocabulario.

Para crear el modelo de Doc2Vec hemos utilizado los propios tweets de entrenamiento y test. El motivo de esta decisión se debe principalmente a que la biblioteca Python para la creación de vectores Doc2Vec no nos ha permitido procesar toda la wikipedia (la misma que la utilizada para Word2Vec). Para utilizar los propios tweets hemos etiquetado cada uno con un identificador único que nos permita recuperar su vector del modelo. Además hemos generado un modelo con los siguientes parámetros: una ventana de 10 términos, el modelo DM y un número de dimensiones de 300. Estos parámetros se han eligido a partir de distintas pruebas empíricas realizadas con los tweets de entrenamiento.

Como puede verse en la Figura 1, nuestro sistema tiene tres 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. Otra en la que se entrena el modelo Doc2Vec con todos los tweets disponibles, tanto los tweets de entrenamiento como los de test. Y por último, otra en la que representamos cada tweet como la concatenación del vector obtenido con Doc2Vec y el vector como la media de los vectores Word2Vec de cada palabra en el tweet. Una simple normalización previa sobre el tweet es llevada a cabo, eliminando repetición de letras y poniendo todo a minúsculas. Así, el algoritmo SVM se entrena con un vector de 500 características como dimensión, resultado de dicha concatenación. La implementación de SVM utilizada es la basada en kernel lineal con entrenamiento SGD (Stochastic Gradient Descent) proporcionada por la biblioteca Sci-kit Learn[4] (Pedregosa et al., 2011).

Obtenemos así tres modelos: uno para los vectores de palabras según Wikipedia con

[1] `https://code.google.com/p/word2vec/`
[2] `http://radimrehurek.com/gensim/`
[3] `http://dumps.wikimedia.org/eswiki`
[4] `http://scikit-learn.org/`

Word2Vec, otro con los vectores de tweets según Doc2Vec, y otro para la clasificación de la polaridad con SVM. Esta solución es la utilizada en las dos variantes de la tarea 1 del TASS con predicción de 4 clases: la que utiliza el corpus de tweets completo (full test corpus) y el que utiliza el corpus balanceado (1k test corpus).

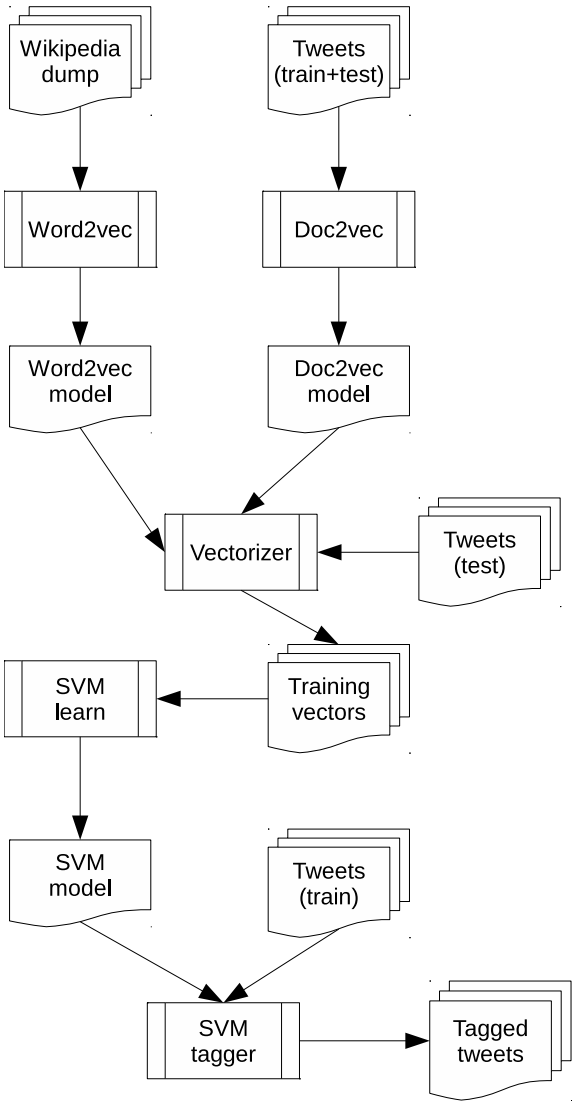

Figura 1: Flujo de datos del sistema completo

4 Resultados obtenidos

Para evaluar nuestro sistema hemos realizado diversos tipos de experimentos. Estos se diferencian según dos aspectos:

- Según el modelo utilizado para crear los vectores. Se han realizado experimentos utilizando sólo Word2Vec (*w2v*), sólo Doc2Vec (*d2v*) y concatenando los vectores de ambos modelos (*dw2v*).
- Según la colección de evaluación utilizada: Los organizadores pusieron a disposición de los participantes la colección completa (*full*) y una colección con un número de etiquetas más homogéneo que sólo contiene 1.000 tweets. Los experimentos con esta última colección han sido nombrados como *1k*.

Como se puede observar en la Tabla 1, los experimentos con mejores resultados son aquellos que utilizan los vectores generados por ambos modelos y la colección más homogenea llegando a alcanzar una precisión del **63 %** y un 46 % de Macro-F1. Con la colección completa, también se alcanzan los mejores resultados utilizando ambos modelos a la vez, obteniendo una precisión del **62 %** aproximadamente y un **47 %** de Macro-F1.

Modelo	Test coll	Accuracy	Macro-F1
wd2v	full	**0,619**	**0,477**
d2v	full	0,429	0,360
w2v	full	0,604	0,466
wd2v	1k	**0,633**	0,460
d2v	1k	0,510	0,306
w2v	1k	0,627	0,466

Tabla 1: Resultados obtenidos en los experimentos

Estos datos nos indican que, aún siendo un sistema bastante sencillo, se obtienen unos resultados prometedores. En ambas colecciones se han mejorado los resultados obtenidos con un único modelo (w2v y d2v) utilizando la conscatenación de ambos (wd2v). Sin embargo nuestra clasificiación no ha obtenido los resultados esperados, debido a que la mejora obtenida uniendo ambos modelos es muy pequeña en comparación con la utilización del modelo Word2Vec. Esto significa, que la utilización del modelos Doc2Vec en nuestros experimentos no es la correcta.

5 Conclusiones y trabajo futuro

Este trabajo describe una novedosa aplicación de los vectores de palabras generados por el método Word2Vec y Doc2Vec a la clasificación de la polaridad, consiguiendo una pequeña mejora en los resultados de precision y Macro-F1 en la competición TASS 2015, tarea 1. Estos resultados son destacables dada la simplicidad de nuestro sistema, que realiza un aprendizaje no supervisado para generar un modelo para representar cada tweet. No

obstante, existen diseños experimentales que no han podido ser acometidos y que esperamos poder realizar para evaluar mejor nuestro sistema, como por ejemplo utilizar una colección de tweets mucho mayor para entrenar el sistema Doc2Vec, o incluso la propia Wikipedia segmentada en frases o párrafos. Aunque para el uso de la Wikipedia con Doc2Vec es necesario un gran sistema computacional, nuestro primer objetivo sería reducir el número de párrafos seleccionando estos de forma aleatoria o utilizando alguna métrica de selección de características. De esta forma, podríamos observar si esta gran fuente de conocimiento es un recurso útil para Doc2Vec y posteriormente estudiar la manera de usar el recurso completo.

Los algoritmos de aprendizaje profundo prometen novedosas soluciones en el campo del procesamiento del lenguaje natural. Los resultados obtenidos con un modelo de palabras general no orientado a dominio específico alguno, ni a la tarea propia de clasificación de la polaridad, así como la no necesidad de aplicar técnicas avanzadas de análisis de texto (análisis léxico, sintáctico, resolución de anáfora, tratamiento de la negación, etc.) nos llevan a continuar nuestra investigación en una adecuación más específica de estos modelos neuronales en tareas concretas.

Es nuestra intención, por tanto, construir un modelo propio de aprendizaje profundo orientado a la clasificación de la polaridad. Gracias a los grandes volúmenes de datos, estas técnicas de aprendizaje profundo pueden aportar buenos resultados en este campo científico. En cualquier caso, es necesario un diseño cuidadoso de estas redes para lograr resultados mas ventajosos y cercanos a otros grupos que han participado en esta edición del TASS 2015, siendo este nuestro objetivo futuro.

Bibliografía

Batista, Fernando y Ricardo Ribeiro. 2012. The l2f strategy for sentiment analysis and topic classification. En *TASS 2012 Working Notes*.

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.

Castellanos, Angel, Juan Cigarrán, y Ana García-Serrano. 2012. Unedtass: Using information retrieval techniques for topic-based sentiment analysis through divergence models. En *TASS 2012 Working Notes*.

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.

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*.

Fernández Anta, Antonio, Philippe Morere, Luis Núñez Chiroque, y Agustín Santos. 2012. Techniques for sentiment analysis and topic detection of spanish tweets: Preliminary report. En *TASS 2012 Working Notes*.

Hall, Mark, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, y Ian H. Witten. 2009. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, Noviembre.

Le, Quoc V y Tomas Mikolov. 2014. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*.

Martín-Wanton, Tamara y Jorge Carrillo de Albornoz. 2012. Uned en tass 2012: Sistema para la clasificación de la polaridad y seguimiento de temas. En *TASS 2012 Working Notes*.

Martínez Cámara, Eugenio, Miguel Ángel García Cumbreras, M. Teresa Martín Valdivia, y L. Alfonso Ureña López. 2013. Sinai-emml: Sinai-emml: Combinación de recursos lingüíticos para el análisis de la opinión en twitter. En *In Proc. of the TASS workshop at SEPLN 2013*.

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*.

Moreno-Ortiz, Antonio y Chantal Pérez-Hernández. 2012. Lexicon-based sentiment analysis of twitter messages in spanish. En *TASS 2012 Working Notes*.

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.

Trilla, Alexandre y Francesc Alías. 2012. Sentiment analysis of twitter messages based on multinomial naive bayes. En *TASS 2012 Working Notes*.

Villena-Román, Julio, Janine García-Morera, Miguel A. García-Cumbreras, Eugenio Martínez-Cámara, M. Teresa Martín-Valdivia, y L. Alfonso Ureña-López. 2015. Overview of tass 2015. En *In Proc. of TASS 2015: Workshop on Sentiment Analysis at SEPLN. CEUR-WS.org*, volumen 1397.