TASS 2017: Workshop on Semantic Analysis at SEPLN, septiembre 2017, págs. 91-96

SINAI en TASS 2017: clasificación de la polaridad de tweets integrando información de usuario

SINAI in TASS 2017: tweet polarity classification integrating user information

M. García-Vega, A. Montejo-Ráez, M.C. Díaz-Galiano, S.M. Jiménez-Zafra
Universidad de Jaén
23071 Jaén (Spain)
{mgarcia,amontejo,mcdiaz,sjzafra}@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 2017. En esta edición hemos explorado dos nuevas soluciones al problema de la clasificación de polaridad en Español, con lo que se han llevado a cabo tres experimentos: clasificación con SVM sobre vectores de palabras, mismo que en el anterior pero integrando vectores de palabras generados a partir de la historia del usuario, y una nueva propuesta aplicando una estructura de red neuronal profunda. Los resultados indican que la primera propuesta es la mejor.
Palabras clave: Análisis de sentimientos, clasificación de la polaridad, deep-learning, vectores de palabras, perfil de usuario

Abstract: This paper introduces the polarity classification system used by the SINAI team for the task 1 at the TASS 2017 workshop. This year, we have tried two new approaches to the polarity classification problem in Spanish, so three experiments have been accomplished: SVM classification over word embeddings, same as before, but adding word embeddings from user's timeline, and a new approach applying a neural network with a deep architecture. Results show that the first approach is the best.
Keywords: Sentiment analysis, polarity classification, deep learning, word embeddings, user profile

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 2017 (Martínez-Cámara et al., 2017), si bien este año se facilita una nueva colección para la tarea 1 (clasificación de polaridad a nivel de tweet), con 1008 tweets de entrenamiento, 506 tweets de desarrollo y 1899 tweets de test. Nuestro trabajo mantiene las técnicas aplicadas en el TASS 2014 (Montejo-Ráez, García-Cumbreras, y Díaz-Galiano, 2014), 2015 (Díaz-Galiano y Montejo-Ráez, 2015) y TASS 2016 (Montejo-Ráez y Díaz-Galiano, 2016) utilizando aprendizaje profundo para representar el texto mediante vectores de palabras. Para ello utilizamos el método *Word2Vec*, ya que ha obtenido los mejores resultados en años anteriores, si bien este año hemos utilizado, en lugar de Wikipedia para el modelo de vectores de palabras, el *Spanish Billion Words Corpus and Embeddings* (SBWCE[1]) preparado por Cristian Cardellino (Cardellino, 2016). 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 2017 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 uti-

[1]Puede descargarse desde http://crscardellino.me/SBWCE/

ISSN 1613-0073
Copyright © 2017 by the paper's authors. Copying permitted for private and academic purposes.

lizando 4 etiquetas de clase (P, N, NEU, NONE).

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, con una revisión breve a las distintas ediciones de TASS. A continuación, se describe el sistema desarrollado y en el apartado 4 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. 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 que parecían 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 preprocesamiento de los tweets y aplicaron un lexicón derivado del inglés para establecer la polaridad de 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.

Tal y como se refleja en el resumen de las actas (García-Cumbreras et al., 2016), los sistemas en la edición de 2016 mantuvieron una técnica similar a ediciones anteriores, optando por metaclasificadores (combinación de clasificadores) basados en clasificadores con distintas configuraciones, como el equipo ELiRF-UPV (Hurtado y Pla, 2016), el cual vuelve a proponer el mejor sistema un año más. Otro aspecto a destacar es que comienza a ser más habitual el uso de vectores de palabras como características de entrada. Si bien, el uso de arquitecturas profundas de redes neuronales no ha llegado a plantear, hasta ahora, sistemas como los que pueden verse con normalidad en competiciones como SemEval (Rosenthal, Farra, y Nakov, 2017), donde el uso de redes recurrentes como LSTM o redes convolucionales son práctica común y han logrado los mejores resultados (Cliche, 2017).

3 Descripción del sistema

3.1 Experimento 1: vectores de palabras con SVM (*w2v-nouser*)

Word2Vec[2] 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 y sus colaboradores (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 bolsas 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 hay que generar un modelo a partir de un volumen de texto grande. Para ello, hemos utilizado el corpus SBWCE (Cardellino, 2016), que consiste en un recopilación de textos desde diversos corpus y fuentes (Wikipedia, Ancora, OPUS Project...) con un total de casi 1.500 millones de palabras. Hemos utilizado el modelo que facilita este autor, que ha sido generado con los siguientes parámetros de Word2Vec:

- Modelo *skip-gram* con *negative-sampling*
- Mínima frecuencia de palabra de 5
- El valor para las palabras "ruido" en *negative-sampling* es de 20
- Las 273 palabras más comunes fueron reducidas
- El número de dimensiones finales de los vectores de palabras es de 300

De esta forma, a diferencia de nuestras propuestas anteriores, optamos por un modelo de vectores de palabras ya preparado. En cualquier caso, seguimos representando 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 entrenamiento facilitada por la organización, a diferencia de la colección de tweets con emoticonos usada en 2016 (Montejo-Ráez y Díaz-Galiano, 2016). De nuevo, 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).

3.2 Experimento 2: integrando información de usuario (*w2v-user*)

Como segundo experimento hemos llevado a cabo una recopilación de los *timeline* de los usuarios. De esta forma, obtenemos mediante el uso de la API de Twitter los 200 últimos tweets de ese usuario. Con estos microtextos generamos un texto completo sobre el que calculamos también el vector de palabras medio. Esta información se concatena a los vectores de palabras de cada tweet en entrenamiento, desarrollo y test. Indicar que en nuestros experimentos hemos usado tanto la partición de entrenamiento como de desarrollo como un único conjunto de entrenamiento.

De esta manera, tenemos vectores de 1.200 dimensiones (300 de la media de los vectores de cada palabra, 300 de las desviaciones típicas de esas dimensiones y, de igual forma, otro par de 300 más 300 asociado al histórico reciente del usuario). Nuestra pretensión con este sistema es incorporar a los usuarios en el entrenamiento, creando un modelo para cada uno de ellos, que vendrá expresado en w2v con su vocabulario y expresiones, concatenando estas componentes con el vector del tweet que se intenta etiquetar, como for-

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

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

ma para reconocerlos en los tweets y afinar el modelo a los propios usuarios.

3.3 Experimento 3: probando una arquitectura profunda completa (*embed-rnn* y *tfidf-rnn*)

Como tercer experimento hemos realizado una primera implementación de una red neuronal profunda sencilla. Para ello hemos utilizado la biblioteca de *deep learning* para Python denominada Keras[4]. Nuestra red neuronal se alimenta directamente de los textos de los tweets, con una primera capa que calcula los vectores de palabras, una segunda que utiliza una red recurrente LSTM clásica tal y como está definida en (Hochreiter y Schmidhuber, 1997). Hemos realizado 50 iteraciones (*epochs*) con un tamaño de *batch* igual a 32. La red se ha entrenado en pocos segundos.

La capa final que genera las polaridades es una capa totalmente conectada (*dense layer*) con cuatro nodos, uno por clase (P, N, NEU, NONE), con una función de activación "softmax"de tal forma que el peso en cada neurona de salida es equivalente a la "probabilidad"de pertenencia a la clase que la red asigna el texto de entrada. En la Figura 1 puede verse la configuración completa de la red propuesta.

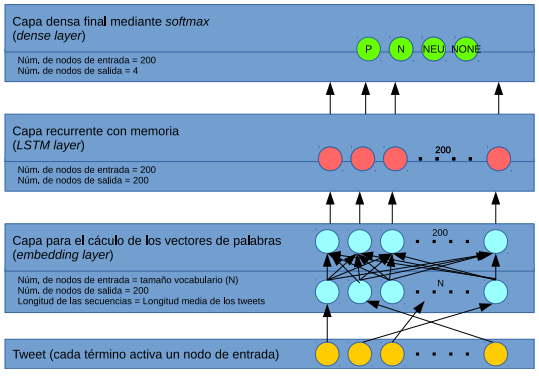

Figura 1: Topología de la red neuronal profunda

La red así configurada se ha entrenado mediante el método de optimización estocástico por grandiente ADAM (Kingma y Ba, 2014), que es muy eficiente desde el punto de vista computacional.

Dado que no es viable entrenar una capa de vectores de palabras (*word embeddings*) con tan pocos datos, hemos realizado una segunda implementación introduciendo directamente a la red LSTM atributos obtenidos siguiendo un modelo clásico TF.IDF. Este experimento es el que hemos denominado *tfidf-rnn*.

[4] `https://keras.io/`

4 Resultados obtenidos

Los resultados según las medidas de *Accuracy* y *Macro F1* obtenidas se muestran en la Tabla 1. Nuestro mejor resultado (*w2v-nouser*) ocupa la posición 16 del total de 48 runs enviados por todos los participantes.

Experimento	Accuracy	Macro-F1
w2v-nouser	57,5 %	44,2 %
w2v-user	56,9 %	42,8 %
embed-rnn	33,3 %	39,1 %
tfidf-rnn	40,4 %	14,4 %

Tabla 1: Resultados obtenidos sobre la colección *InterTASS*

La introducción de características de usuario (*w2v-user*) no ha mejorado los resultados con respecto a *w2v-nouser*, aunque tampoco ha llevado a un empeoramiento significativo de los mismos. Consideramos que, tal y como se han generado los vectores de usuario (ver descripción en la Sección 3.3), estas características añaden algo de ruido, en lugar de permitir al clasificador afinar sobre la decisión de polaridad final. Como comentamos en las conclusiones, queremos seguir estudiando la mejor forma de incorporar la información de usuario en el vector de tweet, pues en otros trabajos esto sí ha llevado a una mejoría del sistema (García-Cumbreras, Montejo-Ráez, y Díaz-Galiano, 2013). Tal vez sea conveniente una fórmula que sintetice mejor el carácter del autor.

Con respecto al uso de arquitecturas profundas en redes neuronales, no encontramos tampoco una mejoría con respecto al primer experimento. En este caso, dadas las características de estas redes, consideramos que el motivo es la escasa información utilizada para su entrenamiento, con poco más de 1500 tweets. Esto se evidencia al descartar la primera capa, en la que se entrenan los vectores de palabras, y sustituirla directamente por los pesos TF.IDF. La capa LSTM, no obstante, puede que no haya sido capaz de entrenarse adecuadamente. Una posible solución a esto es entrenar esa capa con más textos (un corpus de tweets mayor) y luego reajustarla con los datos de entrenamiento.

Al revisar las matrices de confusión para cada experimento encontraremos que los cuatro sistemas propuestos tiene principalmente dificultades con la etiqueta NEU (neutral). Esto puede entenderse desde el punto de vista semántico, ya que esta etiqueta indica que hay una opinión en el tweet (es decir, una carga emocional determinada) pero que dicha opinión no es claramente positiva o negativa. La relativa ambigüedad de esta etiqueta es una dificultad y un reto para los algoritmos expuestos.

5 Conclusiones y trabajo futuro

La experiencia que ganamos en la edición anterior nos desaconsejaba usar la colección de tweets etiquetada con emoticonos, además, ante la disponibilidad de una nueva colección en esta edición del TASS, hemos optado por descartar esa posibilidad.

Sí que también nos resultó interesante la incorporación de texto no formal (tweets) para la generación de los modelos de palabras en los experimentos de 2016. Hemos optado por usar sólo el modelo de SBWCE dado su tamaño, si bien los textos no son tan informales como los que pueden obtenerse con una recopilación de Twitter. Nos planteamos esta posibilidad para el año que viene usando, por ejemplo, alguno de los modelos ya generados como el de Godin y sus colaboradores (Godin et al., 2015).

En el proceso de análisis de los resultados, reflexionamos sobre la viabilidad de usar otros tipos de medidas de agregación diferentes a la media. Consideramos que hacer una media aritmética sobre los vectores de todas las palabras de un tweet puede no ser lo deseable según se trate de obtener un vector representativo del tweet o representativo del usuario. Puede que, de nuevo, una red neuronal sea capaz de representar al usuario, usándola como transductor.

Sí que hemos realizado una primera exploración en arquitecturas profundas, tal y como nos propusimos en la participación anterior como tarea pendiente, pero los resultados no son prometedores. Esto era de esperar, sobre todo en la primera capa de la red neuronal en la que se calculan los vectores de palabras, ya que una colección de poco más de 1.500 tweets es claramente insuficiente para entrenar este tipo de redes, donde el uso de gigabytes de texto son la práctica habitual. Para ello, tenemos como experimento a realizar en breve el alimentar la red neuronal implementada con Keras directamente con los vectores de palabras que obtenemos de SBWCE.

Agradecimientos

Este estudio está parcialmente financiado por el proyecto TIN2015-65136-C2-1-R otorgado por el Ministerio de Economía y Competitividad y por el Ministerio de Educación, Cultura y Deporte (MECD - ayuda FPU014/00983) del Gobierno de España.

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.

Cardellino, Cristian. 2016. Spanish Billion Words Corpus and Embeddings, March.

Cliche, Mathieu. 2017. Bb_twtr at semeval-2017 task 4: Twitter sentiment analysis with cnns and lstms. En *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, páginas 573–580, Vancouver, Canada, August. Association for Computational Linguistics.

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, Arturo Montejo-Ráez, y Manuel Carlos Díaz-Galiano. 2013. Pessimists and optimists: Improving collaborative filtering through sentiment analysis. *Expert Systems with Applications*, 40(17):6758–6765.

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.

Godin, Fréderic, Baptist Vandersmissen, Wesley De Neve, y Rik Van de Walle. 2015. Multimedia lab@ acl w-nut ner shared task: named entity recognition for twitter microposts using distributed word representations. *ACL-IJCNLP*, 2015:146–153.

Hochreiter, Sepp y Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

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 y Ferran Pla. 2016. Elirf-upv en tass 2016: Análisis de sentimientos en twitter. En *TASS@ SEPLN*, páginas 47–51.

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.

Kingma, Diederik P. y Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Martínez-Cámara, Eugenio, Manuel C. Díaz-Galiano, Miguel Á. García-Cumbreras, Manuel García-Vega, y Julio Villena-Román. 2017. Overview of tass 2017. En Julio Villena Román M. Ángel' García Cumbreras Eugenio Martínez-Cámara M. Carlos Díaz Galiano, y Manuel García Vega, editores, *Proceedings of TASS 2017: Workshop on Semantic Analysis at SEPLN (TASS 2017)*, volumen 1896 de *CEUR Workshop Proceedings*, Murcia, Spain, September. CEUR-WS.

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

Montejo-Ráez, Arturo y Manuel Carlos Díaz-Galiano. 2016. Participación de sinai en tass 2016. En *TASS@ SEPLN*, páginas 41–45.

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.

Rosenthal, Sara, Noura Farra, y Preslav Nakov. 2017. Semeval-2017 task 4: Sentiment analysis in twitter. En *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, páginas 502–518, Vancouver, Canada, August. Association for Computational Linguistics.

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.