Clasificación de correos electrónicos usando características superficiales y profundas.

Valentin Morales Moreno, Juan Carlos Gómez Carranza
Departamento de Ingeniería Electrónica, División de Ingenierías Campus Irapuato-Salamanca, Universidad de Guanajuato, Salamanca, México
{v.moralesmoreno, jc.gomez}@ugto.mx

Resumen

El servicio de correo electrónico es uno de los medios de comunicación más populares mundialmente. Sin embargo, enfrenta varios serios problemas, el más importante, se debe a que es usado para llenar a los usuarios con publicidad no solicitada, lo cual resulta en una reducción de la productividad de los usuarios. Es por esto por lo que, a fin de reducir este problema, se han realizado varios estudios y experimentos a lo largo de los últimos años, los cuales tiene como objetivo separar automáticamente aquellos correos importantes (legítimos), de aquellos que no tienen ninguna relevancia (spam). En este trabajo, se presenta el estudio de un conjunto de modelos de aprendizaje supervisado y características basadas en el contenido para el problema de clasificación de correos electrónicos. Se realizaron diferentes experimentos usando diferentes conjuntos de datos (datasets), además de características superficiales y profundas. El desempeño de los modelos se evaluó usando el área bajo la curva ROC, o AUC, la cual es una métrica muy popular en la clasificación de correos electrónicos. Los resultados muestran datos interesantes sobre el problema.

Introducción

El correo electrónico, mejor conocido como *email*, es uno de los medios de comunicación preferidos por personas y empresas hoy en día. Esto, debido a su fácil acceso, bajo costo y, por supuesto, velocidad. Tal es su popularidad, que se estima que aproximadamente 100 millones de cuentas son creadas cada año, y por lo cual se espera que para el 2023 el número de usuarios sea 4.4 mil millones ("Number of e-mail users worldwide 2023",2019). No obstante, estas características y su gran popularidad dieron lugar a un gran problema el cual consiste en que no todos los correos que reciben los usuarios son de su interés o confiables. Para tratar de enfrentar este problema, la clasificación de correos electrónicos se ha estudiado arduamente durante varios años, a fin de ayudar al filtrado automático y separación de correos que no tienen interés y/o no son confiables.

Existen varios enfoques que han sido explorados en la clasificación de correos electrónicos (Cormack, 2008; Guzella & Caminhas, 2009), la mayor a basándose en el texto en los correos (Dada et al.,2009). Manteniendo esta misma línea de investigacion, investigadores a través de los años han construido varios modelos usando diferentes características, tales como palabras(Zhang et al.,2004; Kanaris et al., 2007), n-gramas (Meyer & Whateley,2004), n-gramas de caracteres (Kanaris et al., 2007), estilística (Kumar et al., 2007), y características de aprendizaje profundo (Jain et al., 2019). Además de esto, tambien se han implementado una gran variedad de metodos para la selección (Gomez et al., 2012; Wang et al., 2015) y extracción (Gomez et al., 2012) de características. Asi mismo, varios modelos de aprendizaje automático han sido aplicados al problema de clasificación de correos, tales como máquinas de soporte de vectores (SVM) (Drucker et al., 1999), naive bayes (NB) (Sahami et al., 1998; Metsis et al., 2006), K-vecinos más cercanos (KNN) (Firte at al., 2010), regresiones logísticas (LR) (Chang et al., 2010), reconstrucciones PCA (análisis de componentes principales, por sus siglas en inglés) (Gomez & Mohens, 2012), arboles de decisión (DT) (Sharma et al., 2014), redes neuronales (Clark et al., 2003) y conjuntos de clasificación (Sakkis et al., 2001).

En el presente trabajo se presenta un enfoque innovador usando características superficiales y profundas de los correos electrónicos, este es un escenario basado en el contenido de los correos, específicamente en la información contenida en el cuerpo y tema/asunto de estos, las características superficiales hacen alusión a elementos especificos encontrados en el correo, mientas que las profundas se refieren al contexto asociado con cada palabra.

Adicionalmente, se experimentó con 5 populares modelos de aprendizaje automático, los cuales se pueden agrupar en 4 enfoques: discriminativo (SVM y LR), probabilístico (NB), basado en instancias (KNN) y arboles de decisión (RF).

Objetivos

Cuando se habla de clasificación de correos electrónicos, la meta más popular es separar los correos en dos grandes categorías, spam y ham. La primera categoría se refiere a aquellos correos los cuales no tienen ninguna importancia para el usuario, y, por lo tanto, no quiere recibir, este tipo de mensajes son principalmente publicidad, aunque en algunos casos pueden ser peligrosos y tratar de robar algun tipo de información. La segunda categoría, tambien conocida como correos legítimos, son todos aquellos correos que contienen información relevante para el usuario y, por tanto, son aquellos que quiere ver en su bandeja de entrada.

Asi pues, el objetivo principal del presente trabajo es clasificar correctamente correos electrónicos en dos categorías de acuerdo con la importancia que estos tienen para el usuario (spam y ham, o legítimos). Además, evaluar el desempeño de diferentes modelos de aprendizaje supervisado usando características superficiales y profundas del contenido de los correos.

Justificación

Los correos spam representan un grave problema hoy en dia debido a la cantidad de personas que eligen este medio para comunicarse con sus familias, amigos, compañeros de trabajo, etc. De acuerdo con una importante compañía internacional dedicada a la seguridad informática, en el segundo cuarto del año 2019 más del 55% del total de correos electrónicos eran spam (Vergelis et al.,

"Spam and phishing in Q1 2019", 2019). Toda esa cantidad de correos spam se puede ver traducida en pérdida de tiempo, y, por consiguiente, en pérdida de dinero, además esto conlleva otros problemas tales como desperdicio de ancho de banda para el tráfico de datos, desperdicio de almacenamiento, y mal uso de recursos computacionales disponibles.

A pesar de los grandes avances en el area y la reducción en el porcentaje de spam en el tráfico mundial en los últimos años, estos aun representan más del 50%, lo cual teniendo en cuenta el número de cuentas activas y el número de cuentas que se crean cada año resulta aún en un serio problema, razón por la cual este aun es tema abierto y se ve la necesidad de explorar nuevos y novedosos enfoques como lo es el presentado en este proyecto.

Metodología

Recolección de datos

Debido al gran número de trabajos en el campo, existen varios datasets disponibles, los usados en este proyecto son: SpamAssassin (SA) ("Apache SpamAssassin: Welcome", SF), Enron (EN) (“Natural Language Processing Group”, SF), TREC 2007 (TR) ("2007 TREC Public Spam Corpus", SF), GenSpam (GS) (“GenSpam”, SF), y Ling Spam (LS) ("Spam filtering datasets", SF). Todos ellos son gratuitos y estan públicamente disponibles.

El dataset SA originalmente está dividido en 5 archivos, easy_ham, easy_ham_2, hard_ham, spam y spam_2. Los primeros dos archivos contienen correos fáciles de diferenciar de correos spam. El tercer archivo contiene correos legítimos muy parecidos a spam. Finalmente, el ultimo y penúltimos archivos contienen correos recibidos de fuentes que no son trampas de spam. Para este dataset los correos se combinaron en dos archivos, uno para correos spam y otro para legítimos.

Para el caso del dataset GS, este esta originalmente dividido en 6 directorios, train_GEN, train_SPAM, adapt_GEN, adapt_SPAM, test_GEN, test_SPAM, los cuales contienen correos spam (SPAM) y legítimos (GEN) para cada fase de entrenamiento (train), validación (adapt) y prueba (test), respectivamente.

El dataset EN está dividido en 6 carpetas, cada una de las cuales contienen ambos tipos de correos, para este dataset el contenido de las 6 carpetas se combinó en dos capetas, una para correos spam y otra para legítimos.

El dataset LS contiene diferentes versiones de los datos, para este trabajo solo se usó la carpeta con el nombre *bare*, debido a que los correos que contiene no presentan ningún tipo de procesamiento ni estan codificados. Esta versión de los datos esta divididos en 10 partes, las cuales contienen correos legítimos y spam. Por último, del dataset TR, se usó la carpeta con el nombre full, ya que estos correos esta pre-etiquetados como spam o legítimos.

En la Tabla 1 se muestra la distribución de los correos para cada clase en cada dataset. En la mayoría de los casos existe una clase predominante, la única excepción es para EN, donde los datos prácticamente están balanceados.

Dataset	Spam	Ham	Total	% Spam	% Ham
TR	50071	25217	75288	66.51	33.49
GS	30761	9186	39947	77.00	23.00
SA	1892	4144	6036	31.35	68.65
EN	17110	16544	33654	50.84	49.16
LS	481	2412	2893	16.63	83.37

Tabla 1: Contenido de los datasets

Procesamiento

Antes de utilizar los datasets en la construcción y prueba de los modelos de aprendizaje supervisado, se les aplicaron varias técnicas de procesamiento de datos para transfórmalos y extraer las características que se buscaban. Primero, se tomó solamente el texto dentro del cuerpo y la línea del asunto de los correos, excluyendo etiquetas como *re*, *fw* y *fwd*, y, para aquellos datasets que se encontraban en un formato XML, todas las etiquetas indicando el inicio y fin de un elemento. Posteriormente, se usaron expresiones regulares para extraer únicamente las palabras, las cuales representan la mayoría del contenido de los correos ya que estas expresan el propósito del mensaje. Una vez que se extrajeron solo las palabras, se eliminaron palabras muy cortas (longitud < 3) o

muy largas (longitud > 35), además se eliminaron stopwords, las cuales se refieren a palabras las cuales aparecen demasiadas veces en un texto, pero no aportan ninguna información, tal es el caso de los artículos, para esta tarea se usó la librería NLTK de Python, usando como idioma el inglés debido a que los correos se encuentran escritos en este idioma. Finalmente, cada uno de los correos se transformó en un vector de documento usando el método de frecuencia de termino-frecuencia inversa del documento ($tf - idf$ por sus siglas en inglés). Este método está definido como $tf - idf(t,d) = tf(d,t) \times idf(t)$, donde $tf(d,t)$ es la frecuencia de la característica t en un documento d, y $idf(t)$ está dado por $idf(t) = \log\frac{1+n_d}{1+df(d,t)} + 1$. A su vez, $df(d,t)$ es el número de correos que contienen la característica t; mientras n_d es el numero total de correos electrónicos en el conjunto de entrenamiento.

Adicionalmente, se utilizó una representación de aprendizaje profundo al emplear la técnica conocida como Word2Vect (Mikolov et al., 2013) para expresar cada palabra como un vector. En este caso para cada correo se tomaron las palabras previamente filtradas mediante expresiones regulares y se pasaron a través del modelo pre-entrenado de Google ("Google Code Archive", 2013), de lo cual se obtuvo como salida un vector de palabras incrustadas (word embedding) de 300 dimensiones. Posteriormente, se calculó el promedio todos estos vectores de un correo como su vector final de documento. Para ambos casos (palabras y W2V) se normalizaron los vectores correspondientes usando la norma euclidiana.

Construcción de modelos y experimentos

Usando las características superficiales y profundas extraídas, se construyeron modelos basados en cinco populares metodos de aprendizaje supervisado buscando explorar diferentes enfoques. Los metodos usados incluyen dos clasificadores discriminativos (SVM y LR), un probabilístico (NB), uno basado en instancias (KNN), y uno basado en arboles de decisión (RF).

Los experimentos se desarrollaron mediante una validación cruzada de 10 carpetas, para la mayoría de los datasets la validación cruzada fue estratificada aleatoriamente, mientras que para LS se usó la separación original, al igual que

con GS, se usó los datos correspondientes a entrenamiento, validación y prueba como ya se encontraban previamente divididos. Durante la validación cruzada de 10 carpetas, para cada iteración, se extrajo un vocabulario independiente del conjunto de entrenamiento, compuesto por 9 carpetas, se calculó el idf, y posteriormente se usó el vocabulario y las idfs para vectorizar el conjunto de entrenamiento asi como el de prueba. La vectorización se realiza ya sea usando tf-idf en el caso de las palabras, o word embeddings para W2V, usando el vocabulario extraído de la parte de entrenamiento.

Adicionalmente, los clasificadores SVM, LR, KNN y RF tienen hiperparámetros (HP) que afectan su comportamiento. A fin de optimizar el desempeño individual de los modelos, se llevó a cabo una validación cruzada de 3 carpetas para cada conjunto de entrenamiento y cada posible valor de los hiperparámetros.

La Tabla 2 muestra los hiperparámetros que se optimizaron en cada metodos y los valores considerados para esto a fin de maximizar el puntaje AUC.

CLASIFICADOR	HP	DESCRIPCIÓN	VALORES
SVM	C	Parámetro de regularización	[0.1, 1, 10, 100]
LR	C	Parámetro de regularización	[0.1, 1, 10, 100]
KNN	K	Número de vecinos	[1, 2, 3, 5, 10]
RF	n	Numero de árboles en el bosque	[5, 10, 15, 20]

Tabla 2: Hiperparámetros a optimizar

Naturalmente le problema de clasificación de correos electrónicos trata con distribuciones desequilibradas de datos, donde se tiene más datos de una clase que de otra (véase Tabla 1). Debido a esto, métricas como la exactitud no son recomendadas para evaluar el desempeño de un modelo, ya que el resultado tendrá una inclinación hacia la clase predominante. Por este motivo, en el presente trabajo, se utilizó el area debajo de la curva de característica operativa del receptor (ROC por sus siglas en ingles), tambien conocida como AUC (area under the curve), para evaluar el desempeño de los diferentes modelos. ROC es una curva de probabilidad que grafica la tasa verdadera-positiva contra la tasa

falsa-positiva en varios límites. Por su parte, AUC evalúa el grado de separabilidad, al calcular la probabilidad que un modelo clasifique aleatoriamente un elemento positivo más que a uno negativo (Fawcett, 2006). AUC puede tomar valores entre 0 y 1, sin embargo, un clasificador aleatorio producirá una diagonal como curva ROC, por lo cual obtendrá un puntaje de 0.5 en AUC, lo cual representa el desempeño base.

El procesamiento y modelos de clasificación fueron implementados en Python, usando las librerías NLTK, numpy y Scikit-learn. Los experimentos se realizaron en una estación de trabajo Linux con un procesador Xeon Silver de 2.1 HGz y 128 GB de memoria RAM.

Resultados

En la Tablas 3 y 4 se presentan los resultados AUC usando palabras y W2V. Cada fila en las tablas corresponde al dataset con el que se realizaron las pruebas, mientras que las columnas indican el modelo utilizado. Los valores en negritas representan los mejores resultados en la fila correspondiente, mientras que la última columna muestra el promedio y desviación estándar de todos los valores en una fila, de manera similar, la última fila muestra el promedio y desviación de estándar de los valores obtenidos por cada modelo. Finalmente, los valores en la esquina inferior derecha indican el desempeño general de una característica.

En la Tabla 3 se puede observar que, en general, todos los datasets muestran un buen desempeño sin importar el modelo de clasificación, a excepción de cuando se intentan clasificar los datos contenidos en LS usando NB, al igual que cuando se usa KNN para clasificar los correos en GS. Por otra parte, TR y EN son los datasets que muestran los valores más altos, lo que indican que estos datasets son más fáciles de clasificar cuando se usan palabras. En cuanto a los modelos de clasificación, los clasificadores discriminativos (SVM y LR) son los que muestra en promedio un mejor desempeño sobre todos los datasets.

De manera similar, en la Tabla 4 se muestran los resultados de los experimentos realizados usando W2V. En esta tabla se puede observar que los resultados se asemejan a aquellos obtenidos usando palabras, pero con algunas discrepancias,

la más notoria es, que en general, el clasificador NB tienen un desempeño bajo para la gran mayoría de datasets. Para W2V, los datasets que obtuvieron los mejores puntajes, y por tanto son más fáciles de clasificar, fueron LS y EN. Finalmente, LR es el clasificador con los mejores puntajes en promedio, seguido muy de cerca por SVM y KNN.

	Modelo					
	SVM	LR	NB	KNN	RF	AVG
TR	**0.98**	**0.98**	0.97	0.97	0.95	0.97 (0.01)
GS	**0.93**	**0.93**	0.91	0.75	0.90	0.88 (0.07)
SA	**0.95**	**0.95**	0.83	0.92	0.92	0.91 (0.04)
EN	**0.98**	**0.98**	**0.98**	0.96	**0.98**	0.98 (0.01)
LS	**0.97**	0.96	0.70	0.96	0.93	0.90 (0.10)
AVG	**0.96 (0.02)**	**0.96 (0.02)**	0.88 (0.10)	0.91 (0.08)	0.94 (0.03)	0.94 (0.05)

Tabla 3: Resultados usando palabras

	Modelo					
	SVM	LR	NB	KNN	RF	AVG
TR	0.95	0.95	0.50	**0.96**	0.95	0.86 (0.18)
GS	0.94	**0.95**	0.50	0.93	0.88	0.84 (0.17)
SA	**0.93**	**0.93**	0.50	0.92	0.91	0.84 (0.17)
EN	0.96	0.96	0.85	**0.97**	0.95	0.94 (0.04)
LS	**0.99**	**0.99**	0.94	0.98	0.96	0.97 (0.02)
AVG	0.95 (0.02)	**0.96 (0.02)**	0.66 (0.20)	0.95 (0.02)	0.93 (0.03)	0.89 (0.06)

Tabla 4: Resultados usando W2V

Conclusiones

En el presente proyecto se realizó el análisis de diferentes modelos de aprendizaje supervisado y de características basadas en el contenido de los correos electrónicos para la clasificación de estos en dos clases llamadas spam y

ham (legítimos). Se realizaron pruebas con los modelos y características usando 5 datasets diferentes. A partir de los resultados obtenidos, se puede concluir los siguiente:

- Los clasificadores discriminativos tienden a tener un mejor desempeño en promedio más que otros clasificadores, esto sin importar la característica usada para la construcción de los modelos.
- A pesar de que W2V utiliza una representación más pequeña para los correos, esta característica logra unos resultados muy cercanos a los obtenidos al usar palabras.
- A pesar de que LS es el dataset más pequeño obtiene muy buenos resultados usando ambas características, significando que el contenido de sus pocos correos contiene información valiosa para la clasificación de estos.

Futuros trabajos podrían incluir el uso de otras características tales como proyecciones sobre espacios discriminativos (análisis de discriminante lineal o análisis de discriminante sesgado), y otros tipos de características profundas (Doc2Vec o vectores de pensamiento); incluso el uso de modelos de selección de características, para seleccionar aquellas características con el mayor poder discriminativo entre clases.

Referencias

Apache SpamAssassin: Welcome. (n.d.). Retrieved from https://spamassassin.apache.org/.

Chang, M. W., Yih, W. T., & Meek, C. (2008, August). Partitioned logistic regression for spam filtering. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 97-105). ACM.

Clark, J., Koprinska, I., & Poon, J. (2003, October). A neural network based approach to automated e-mail classification. In *Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003)* (pp. 702-705). IEEE.

Cormack, G. V. (2008). Email spam filtering: A systematic review. *Foundations and Trends® in Information Retrieval*, *1*(4), 335-455.

Dada, E. G., Bassi, J. S., Chiroma, H., Adetunmbi, A. O., & Ajibuwa, O. E. (2019). Machine learning for email spam filtering: review, approaches and open research problems. *Heliyon*, *5*(6), e01802.

Drucker, H., Wu, D., & Vapnik, V. N. (1999). Support vector machines for spam categorization. *IEEE Transactions on Neural networks*, *10*(5), 1048-1054.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern recognition letters*, *27*(8), 861-874.

Firte, L., Lemnaru, C., & Potolea, R. (2010, August). Spam detection filter using KNN algorithm and resampling. In *Proceedings of the 2010 IEEE 6th International Conference on Intelligent Computer Communication and Processing* (pp. 27-33). IEEE.

GenSpam. (n.d.). Retrieved from http://www.benmedlock.co.uk/genspam.html?LMCL=LA9WH5.

Gomez, J. C., Boiy, E., & Moens, M. F. (2012). Highly discriminative statistical features for email classification. *Knowledge and information systems*, *31*(1), 23-53.

Gomez, J. C., & Moens, M. F. (2012). PCA document reconstruction for email classification. *Computational Statistics & Data Analysis*, *56*(3), 741-751.

Google Code Archive . (2013, July 29). Retrieved from https://code.google.com/archive/p/word2vec/.

Guzella, T. S., & Caminhas, W. M. (2009). A review of machine learning approaches to spam filtering. *Expert Systems with Applications*, *36*(7), 10206-10222.

Jain, G., Sharma, M., & Agarwal, B. (2019). Optimizing semantic LSTM for spam detection. *International Journal of Information Technology*, *11*(2), 239-250.

Kanaris, I., Kanaris, K., Houvardas, I., & Stamatatos, E. (2007). Words versus character n-grams for anti-spam filtering. *International Journal on Artificial Intelligence Tools*, *16*(06), 1047-1067.

Kumar, R. K., Poonkuzhali, G., & Sudhakar, P. (2012, March). Comparative study on email spam classifier using data mining techniques. In *Proceedings of the International MultiConference of Engineers and Computer Scientists* (Vol. 1, pp. 14-16).

Metsis, V., Androutsopoulos, I., & Paliouras, G. (2006, July). Spam filtering with naive bayes-which naive bayes?. In *CEAS* (Vol. 17, pp. 28-69).

Meyer, T. A., & Whateley, B. (2004, July). SpamBayes: Effective open-source, Bayesian based, email classification system. In *CEAS*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111-3119).

Natural Language Processing Group (n.d.). Retrieved from http://nlp.cs.aueb.gr/software.html.

Number of e-mail users worldwide 2023. (2019, Agosto 9). Retrieved from https://bit.ly/2sE0bJ7.

Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998, July). A Bayesian approach to filtering junk e-mail. In *Learning for Text Categorization: Papers from the 1998 workshop* (Vol. 62, pp. 98-105).

Sakkis, G., Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Spyropoulos, C. D., & Stamatopoulos, P. (2001). Stacking classifiers for anti-spam filtering of e-mail. *arXiv preprint cs/0106040*.

Sharma, A. K., Prajapat, S. K., & Aslam, M. (2014). A comparative study between naïve Bayes and neural network (MLP) classifier for spam email detection. *Int. J. Comput. Appl.*

Spam filtering datasets. (n.d.). Retrieved from https://aclweb.org/aclwiki/Spam_filtering_datasets.

Vergelis, M., Shcherbakova, T., & Sidorina, T. (2019, Mayo 15). Spam and phishing in Q1 2019. Retrieved from https://bit.ly/2KafOz6.

Wang, Y., Liu, Y., Feng, L., & Zhu, X. (2015). Novel feature selection method based on harmony search for email classification. *Knowledge-Based Systems*, *73*, 311-323.

Zhang, L., Zhu, J., & Yao, T. (2004). An evaluation of statistical spam filtering techniques. *ACM Transactions on Asian Language Information Processing (TALIP)*, *3*(4), 243-269.

2007 TREC Public Spam Corpus. (n.d.). Retrieved from https://plg.uwaterloo.ca/~gvcormac/treccorpus07/.