Trabajo de Final del Máster Universitario en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital

Identificación de la Variedad del Lenguaje para la Mejora del Geoposicionamiento en Social Media

Autor: Raül Fabra Boluda

Directores: Dr. Paolo Rosso y Dr. Francisco Rangel

Septiembre de 2016

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Resum

Amb l'auge de la web 2.0 i els medis socials, els usuaris dels mateixos prenen protagonisme per ser els principals generadors d'informació de tot tipus. A més, el nombre d'usuaris d'aquestos medis augmenta cada any. Nous escenaris plantegen noves necessitats i reptes, com ara geoposicionar als usuaris dels medis socials. Una forma d'abordar el geoposicionament dels usuaris a nivell de país pot consistir en la identificació de la varietat de l'idioma que utilitzen: per exemple, si un usuari escriu en espanyol, tractar de determinar de quina varietat del espanyol es tracta. Aquesta tasca es coneix com a identificació de la varietat del llenguatge o *Language Variety Identification* (LVI). Pot enfocar-se com una tasca d'*Author Profiling*, on el tret a identificar es la varietat de l'idioma de l'usuari, tractant d'inferir-la a partir dels seus textos.

Aquestes tasques són molt noves, per la qual cosa hi ha escassesa en els recursos disponibles. En aquest treball proposem una metodologia per a la construcció d'un corpus i la posem en pràctica. El resultat ha sigut HispaTweets, un corpus construït amb usuaris de Twitter i fins a 1.000 dels seus missatges (tweets). Aquestos usuaris procedeixen de set països de parla hispana: Argentina, Chile, Colòmbia, Espanya, Mèxic, Perú i Veneçuela. Per a la seua construcció hem desenvolupat una ferramenta que recupera els tweets emesos des de les ciutats més importants dels països nomenats. Amb els autors d'aquestos tweets hem elaborat un llistat d'usuaris per geografia, dels quals havem descarregat els seus últims 1.000 tweets més recents. Sobre el corpus obtingut havem realitzat tres filtrats per a que el corpus siga representatiu de les varietats de l'idioma i estiga equilibrat: i) geogràfic, ii) temporal i iii) per freqüència. HispaTweets compta amb 4.550 usuaris (650 per país) i quasi quatre milions de tweets.

Per a l'avaluació del corpus hem emprat característiques basades en n-grames de paraules i caracters, com freqüències d'n-grames, TF-IDF, i un mètode de representació en baixa dimensionalitat o *Low-Dimensionality Representation* (LDR), que representa cada usuari mitjançant un nombre reduït de característiques. Per a predir la varietat de l'idioma havem emprat diferents mètodes de classificació: Naïve Bayes, Arbres de Decisió i Màquines de Vectors de Suport. L'avaluació l'hem dut a terme baix un esquema de validació creuada en 5 blocs, mantenint separats els usuaris d'entrenament i de validació durant tot el procés. A més d'aquestes tècniques, també hem desenvolupat un algorisme que tracta de predir la ubicació de l'usuari a partir del camp "*location*" del seu perfil a Twitter. L'objectiu d'aquest algorisme era estudiar quins resultats es podien obtindre a partir de la informació del perfil de l'usuari, sense tècniques per a LVI. També hem comparat els resultats obtinguts amb els millors resultats del corpus HispaBlogs. Finalment expliquem la nostra participació a la tasca DSL 2015 al taller LT4VarDial, que ha consistit en la identificació d'idiomes similars i varietats de l'idioma.

Resumen

Con el auge de la Web 2.0 i los medios sociales, los usuarios de los mismos toman protagonismo al ser los principales generadores de información de todo tipo. Además, el número de usuarios de estos medios aumenta cada año. Nuevos escenarios plantean nuevas necesidades y retos, por ejemplo geoposicionar a los usuarios de los medios sociales. Una forma de abordar el geoposicionamiento de los usuarios a nivel de país puede consistir en la identificación de la variedad del idioma que utilizan: si un usuario escribe en español, tratar de determinar de qué variedad del español se trata. Esta tarea se conoce como identificación de la variedad del idioma o *Language Variety Identification* (LVI). Puede enfocarse como una tarea de *Author Profiling*, donde el rasgo a identificar es la variedad del idioma del usuario, tratando de inferirla a partir de sus textos.

Estas tareas son muy novedosas, por lo cual existe escasez en los recursos disponibles. En este trabajo proponemos una metodología para la construcción de un corpus anotado y la ponemos en práctica. El resultado ha sido HispaTweets, un corpus construido con usuarios de Twitter i hasta 1.000 de sus mensajes (tweets). Estos usuarios proceden de siete países de habla hispana: Argentina, Chile, Colombia, España, México, Perú y Venezuela. Para su construcción hemos desarrollado una herramienta que recupera los tweets emitidos desde las ciudades más importantes de los países mencionados. Con los autores de estos tweets hemos elaborado un listado de usuarios por geografía, de los cuales hemos descargado sus últimos 1.000 tweets más recientes. Sobre el corpus obtenido realizamos tres filtrados para que el corpus sea representativo de las variedades del idioma y esté equilibrado: i) geográfico; ii) temporal y iii) por frecuencia. HispaTweets cuenta con 4.550 usuarios (650 por país) y casi cuatro millones de tweets.

Para la evaluación del corpus hemos empleado características basadas en n-gramas de palabras y caracteres, como frecuencias de n-gramas, TF-IDF, y un método de representación en baja dimensionalidad o *Low-Dimensionality Representation* (LDR), que representa cada usuario mediante un número reducido de características. Para predecir la variedad del idioma empleamos distintos métodos de clasificación: Naïve Bayes, Árboles de Decisión y Máquinas de Vectores de Soporte. La evaluación la hemos llevado a cabo bajo un esquema de validación cruzada en 5 bloques, manteniendo separado los usuarios de entrenamiento y validación durante todo el proceso. Además de estas técnicas, también hemos desarrollado un algoritmo que trata de predecir la ubicación del usuario a partir del campo "*location*" de su perfil en Twitter. El objetivo de este algoritmo era estudiar qué resultados se podían obtener a partir de información del perfil del usuario, sin técnicas para LVI. También hemos comparado los resultados obtenidos con los mejores resultados del corpus HispaBlogs. Finalmente explicamos nuestra participación en la tarea DSL 2015 en el taller LT4VarDial, que ha consistido en la identificación de idiomas similares y variedades del idioma.

Índice general

Índice de tablas

Índice de figuras

Capítulo 1

Introducción

Desde el ascenso de la Web 2.0 a medianos de los años 2000, la forma de comunicarnos ha cambiado significativamente. Desde entonces han surgido nuevas plataformas orientadas a facilitar que los usuarios de Internet sean capaces de compartir información de todo tipo. Por ejemplo, intercambian experiencias personales, comparten información que consideran interesante, expresan sus opiniones políticas, sobre la actualidad, sobre los productos que adquieren, etc. Este tipo de plataformas donde el contenido es generado por los propios usuarios se denominan medios sociales. Algunos de ellos se han introducido y estandarizado en nuestra vida cotidiana, como los blogs, Facebook[1] o Twitter[2]. El número de usuarios de los medios sociales ha ido en aumento año tras año, generando cada vez más información por su parte.No es de extrañar que el interés en el estudio de estos medios sea cada vez mayor, puesto que generan nuevas necesidades y retos. La propia naturaleza de Internet provoca que las fronteras geográficas queden desdibujadas, no siendo siempre posible determinar desde donde se genera el contenido. Los gobiernos y las organizaciones pueden tener interés en conocer la ubicación desde la cual se emite la información. El hecho de poder segmentar geográficamente la información y los autores que la emiten puede tener distintas aplicaciones. Podrían usarse por ejemplo en *marketing*, para realizar una segmentación geográfica o demográfica de las opiniones de los usuarios al lanzar un nuevo producto y emplear estos datos en análisis posteriores. También podrían utilizarse en materia de seguridad y lingüística forense, por ejemplo para tratar de geoposicionar al autor de una amenaza. Como apenas un 2 % de los usuarios georreferencia sus contenidos, surge la necesidad de realizar esa segmentación geográfica mediante otros métodos. Una forma de abordarla puede ser identificar la variedad del idioma de los usuarios, con el fin de geoposicionar el contenido a nivel de país. Podemos tratar de inferir la variedad del idioma del usuario a partir de sus textos. Por ejemplo, si tenemos un texto en español podemos tratar de identificar a que país hispanohablante pertenece su autor. El siguiente ejemplo muestra el razonamiento de cómo hacerlo de forma intuitiva: si nos

[1] https://www.facebook.com/
[2] https://twitter.com/

llega un texto para el cual dudamos si ha sido escrito por alguien en España o alguien en Argentina y vemos que aparece muchas veces la palabra "boludo", es posible deducir que este texto ha sido escrito por alguien de Argentina, ya que esa palabra se usa mucho más frecuentemente allí que en España.

Podemos abordar la identificación de la variedad del idioma de un usuario como una tarea de *Author Profiling* (AP). Es un campo de investigación muy reciente que pretende identificar rasgos personales de los autores de los textos a partir de su forma de escribir. Rasgos como la edad, el sexo, la personalidad, el idioma nativo o la variedad regional del idioma. Resulta obvio que una persona de 50 años escribe de forma muy distinta a una de 15 (por ejemplo, en el uso de neologismos), o que una persona de Argentina escribe diferente a una de España, tanto en los términos que utilice como en los temas que trate. Estos rasgos dejan huella en la escritura del autor, que depende de su idiosincrasia: cada individuo posee sus propias creencias e influencias socioculturales y educacionales que definen su forma de escribir. Estos pueden marcar el estilo discursivo del autor, el vocabulario empleado, los temas que trate, etc.

AP es un tema de investigación candente. En los últimos años se han organizado diferentes talleres en estas líneas, como:

- los talleres PAN[3] 2013, 2014, 2015 y 2016 [1, 2, 3, 4] en la conferencia CLEF, donde se han abordado los problemas de la identificación de la edad y el sexo en los medios sociales (2013 y 2014), también a nivel *cross-genre* (2016), además del reconocimiento de personalidad en Twitter (2015).
- el proyecto myPersonality[4] en ICWSM-13.
- la tarea de identificación del idioma nativo[5] [5] en el taller BEA-8 en NAACL-HLT 2013;
- la tarea PR-SOCO[6] en la conferencia CLEF, dedicada al reconocimiento de la personalidad de los programadores a partir de sus códigos fuente.

Desde la óptica del AP, la tarea del geoposicionamiento puede aproximarse mediante la identificación de la variedad del idioma o *Language Variety Identification* (LVI). Este problema consiste en la identificación de la variedad en que está escrito un texto. Por ejemplo, dado un texto en español, identificar si se trata de español argentino, mexicano, peninsular, etc. Este problema tiene su raíz en otro, la identificación del idioma o *Language Identification* (LI), que consiste en determinar en qué idioma está escrito un texto. Fue planteado los años 60 [6] y entre los años 90 y 2012 han surgido distintas soluciones satisfactorias. Algunas de las técnicas empleadas han sido los n-gramas de caracteres [7], n-gramas de caracteres implementados mediante Modelos Ocultos de Markov [8], *string kernels* [9] o n-gramas de caracteres con métodos de selección de características

[3] http://pan.webis.de
[4] http://mypersonality.org/wiki/doku.php?id=wcpr13
[5] https://sites.google.com/site/nlisharedtask2013/
[6] http://www.autoritas.es/prsoco/

sofisticados [10]. En general estos trabajos muestran *accuracies*[7] superiores al 90-95 %, por lo que hay quien lo considera un problema prácticamente resuelto. Existen algunas carencias en la forma de abordar este tipo de tareas [11] bajo un escenario más realista, como el hecho de que no se tratara con documentos en idiomas desconocidos para el sistema, asumir que los documentos están un un sólo idioma o que la identificación se realice a nivel de documento y no de fragmento de texto. También resulta complicada si en lugar de tratar con texto formal, como el del ámbito periodístico, se trata con texto corto y espontáneo como el de las redes sociales [12]. Si además los idiomas a diferenciar son muy parecidos entre si, las técnicas existentes en LI pueden resultar insuficientes por la similitud de los textos, como es habitual en LVI.

Estas tareas se pueden abordar como un caso concreto de categorización de documentos, donde las categorías son los idiomas o sus variedades. Nosotros queremos ir un paso más allá y aproximar la LVI desde la perspectiva del AP. Esto nos permite centrar el estudio a nivel de autor, siendo por tanto más aplicable a los medios sociales donde los autores (usuarios) son los protagonistas. Para ello no basta con disponer de textos en las distintas variedades de un idioma, sino que además han de estar agrupados por autores para que sus rasgos personales puedan reflejarse en su escritura. La LVI vista desde la perspectiva del AP puede ayudar con el problema del geoposicionamiento, tratando de identificar la variedad del idioma de los usuarios de los medios sociales. Conviene prestar atención otras tareas similares, como la la identificación del idioma nativo o *Native Language Identification* (NLI). Esta tarea también pertenece al campo del AP y guarda algunas semejanzas con LVI. NLI consiste en la identificación del idioma nativo de un autor a partir de sus textos escritos en un segundo idioma. Por ejemplo, averiguar el idioma nativo de un periodista que escribe en inglés. Así como en en NLI todos los textos están escritos en un mismo idioma y se pretende identificar el idioma nativo del autor, en LVI los textos también están en un mismo idioma y lo que se pretende identificar es la variedad del idioma empleada por el autor. Otra tarea a la que también conviene prestar atención es la discriminación de idiomas similares o *Discrimination between Similar Languages* (DSL). Este problema es un caso límite de la LI, donde los idiomas a diferenciar son muy parecidos entre si, siendo por tanto difícil de abordar con las técnicas de LI convencionales. Aunque esta tarea no suele aproximarse desde la perspectiva del AP, comparte parte de la problemática con LVI porque la idiosincrasia cultural puede influir en las diferencias entre las lenguas, así como en las diferencias entre las variedades de una lengua.

Como estos campos de investigación son relativamente recientes, los recursos para abordar este tipo de tareas son escasos. Este trabajo se ha centrado en el desarrollo de una metodología para la construcción de nuevos recursos para LVI, que se ha aprovechado para abordar la tarea de geoposicionamiento planteada. Siguiendo esta metodología se ha construido HispaTweets, un corpus compuesto por un listado de usuarios de Twitter con hasta 1.000 tweets de cada

[7] El *accuracy* se define como la proporción de muestras correctamente identificadas frente al total. En la sección 4.4.3 se describe el *accuracy*.

uno, todos ellos pertenecientes a países de habla hispana. Se ha evaluado este corpus mediante distintas técnicas basadas en n-gramas, actualmente utilizadas en el estado del arte para este tipo de tareas. También se ha empleado un método novedoso de representación en baja dimensionalidad para la generación de características. En cuanto a algoritmos de clasificación se han utilizado tres tipos de métodos: Naïve Bayes, Árboles de Decisión y Máquinas de Vectores de Soporte o *Support Vector Machines* (SVM). Además, se ha desarrollado un algoritmo para identificar la ubicación del autor a partir de su perfil en Twitter.

El resto de la tesina está estructurada como sigue. En el Capítulo 2 se presentan algunos de los trabajos más relevantes en las tareas de NLI, LVI y DSL. Dedicamos el Capítulo 3 a explicar el procedimiento seguido para la construcción del corpus HispaTweets, y realizamos un análisis detallado de sus características. El Capítulo 4 detalla las técnicas utilizadas para la evaluación de HispaTweets y el marco experimental. En el Capítulo 5 se muestran los resultados obtenidos con el proceso de evaluación. Explicamos en el Capítulo 6 nuestra participación en la tarea DSL 2015 en el taller LT4VarDial. Finalmente, en el Capítulo 7 se exponen las conclusiones obtenidas a lo largo del trabajo y planteamos varias líneas para trabajos futuros.

Capítulo 2

Revisión de la Identificación de la Variedad del Idioma

En este capítulo se describen los trabajos más recientes en los tres problemas previamente mencionados: identificación del idioma nativo (NLI), identificación de la variedad del idioma (LVI) y discriminación de idiomas similares (DSL). Los tres pertenecen al campo del AP y son distintos entre si, pero están estrechamente relacionados.

El problema de la NLI consiste en identificar el idioma nativo (L1) del autor que ha escrito un texto en otro idioma (L2). Por ejemplo, la identificación del idioma nativo de investigadores o periodistas que publiquen artículos en inglés. Cuando una persona escribe en un idioma extranjero tiende a cometer errores en la escritura y a usar ciertas expresiones que vienen influenciadas por el idioma nativo [13]. Por ejemplo, cuando los españoles escriben en inglés tienden a confundir el uso de las dobles consonantes, como '*inteligent*' en lugar de '*intelligent*'. Esta y otras conclusiones se desprenden del trabajo de trabajo de Koppel et al. [14]. Los autores desarrollaron un sistema capaz de identificar el idioma nativo de entre cinco idiomas posibles, a partir de los textos de los autores en inglés. Como características emplearon 400 palabras función[1], 200 n-gramas de caracteres, 185 tipos error y 250 bi-gramas de etiquetas POS[2] infrecuentes. Mediante un clasificador multiclase basado en SVM obtuvieron un *accuracy* del 80,2 %, evaluando el corpus ICLE [15] con validación cruzada en 10 bloques. Tofighti et al. [16] también abordaron la tarea de NLI. Para ello recopilaron un total de 600 textos de agencias de noticias en inglés, el idioma nativo de cuyos autores es inglés, persa, turco o alemán. Emplearon cuatro tipos de características: léxicas, sintácticas, estructurales y específicas del contenido. Como características léxicas emplearon n-gramas de caracteres, frecuencias de

[1] Palabras con poco significado léxico o ambiguo, pero que expresan relaciones gramaticales con otras palabras dentro de una oración.

[2] *Part of Speech*. Etiquetas a nivel de palabra con información asociada. Típicamente, categorías gramaticales.

longitud de palabras, características basadas en la riqueza del vocabulario. Para las características sintácticas consideraron los signos de puntuación y las palabras función. Emplearon distintas características estructurales como número de líneas, frases, párrafos, su longitud, etc. En cuanto a las características específicas del contenido, utilizaron aquellos n-gramas de palabras cuya frecuencia fuera superior a 10. Evaluaron su corpus bajo un esquema de validación cruzada en 10 bloques y con SVM obtuvieron sus mejores *accuracies*, alrededor del 70-80 %. También probaron con otros clasificadores, Naïve Bayes y Árboles de Decisión (C4.5). En base al interés creciente en este tipo de tareas, Tetreault et al. [5] organizaron la primera tarea compartida en NLI en el taller BEA-8 en el NAACL-HT. Emplearon el corpus TOEF11 [17], compuesto de 12.100 ensayos escritos para exámenes de acceso a la universidad, para alumnos de 11 nacionalidades. La tarea presentaba tres modalidades: i) *closed-training*, donde sólo podían usarse el corpus proporcionado para entrenar los modelos; ii) *open-training 1*, donde los modelos podían entrenarse con cualquier corpus excepto el proporcionado; iii) *open-training 2*, donde los modelos podían entrenarse con cualquier corpus, incluyendo el proporcionado. Entre los sistemas reportados cabe destacar el presentado por Brook y Hirst [18], en el cual entrenaron sus modelos a partir de distintos corpus y utilizaron varios conjuntos de características: n-gramas de palabras y una mezcla de n-gramas de palabras función con etiquetas POS. Con SVM como clasificador, los autores obtuvieron *accuracies* del 80,2 %, 56,5 % y 81,6 % para las modalidades *closed-training*, *open-training 1* y *open-training 2* respectivamente. También es interesante destacar el trabajo de Bykh y Meurers [19]. Los autores alinearon distintos corpus existentes para crear uno nuevo, el cual utilizaron para entrenar los modelos con los que evaluaron el corpus proporcionado. Emplearon características basadas en gramáticas libres de contexto que utilizaron para entrenar un modelo de regresión logística. Reportaron un *accuracy* del 84,82 %.

Como hemos mencionado, la tarea de LVI consiste en identificar la variedad del idioma empleada por el autor de un texto. Por ejemplo, en el caso de un texto en español, determinar si está escrito en español de España, México, Argentina, etc. Esta tarea tiene aspectos en común con la de NLI. En LVI también se dispone de textos escritos por distintos autores, todos en un mismo idioma. La diferencia es que tratamos de identificar la variedad del idioma en lugar del idioma nativo. Los textos de un autor reflejan sus propios rasgos socioculturales entre los cuales podemos destacar la variedad del idioma. Por ejemplo, hay países hispanohablantes en los que el voseo es utilizado de forma generalizada, como Argentina y Uruguay, mientras que otros se utiliza poco o es inexistente, como Perú o México [20]. La LVI es un tema de investigación candente. Zampieri y Gebre desarrollaron un sistema para diferenciar el portugués de Portugal del de Brasil [21]. Construyeron un corpus recopilando 1.000 textos: 500 textos de un periódico de Portugal (*Diário de Notícias*) publicados en 2007 y 500 textos de un periódico de Brasil (*Folha de São Paulo*) publicados en 2004. Utilizaron distintos tipos de características: léxicas (uni-gramas de palabras), léxico-sintácticas (bi-gramas de palabras) y ortográficas (n-gramas de de caracteres, para n de 2 a 6). Para la clasificación, emplearon modelos del

lenguaje calculados mediante distribuciones de probabilidad de Laplace. Para la identificación de la variedad, calcularon la probabilidad (log-verosimilitud) del texto de pertenecer a cada una de las variedades, asignándolo a aquella variedad cuya probabilidad era más alta. Evaluaron el corpus mediante una partición 50-50 % para entrenamiento y test. Los mejores resultados los obtuvieron con las características ortográficas: 99,8 % de *accuracy* para 4-gramas de caracteres. Con las características léxico sintácticas (bi-gramas de palabras) obtuvieron un *accuracy* del 91,2 %. Finalmente, con las características léxicas (uni-gramas de palabras) consiguieron un 99,6 %. Concluyeron que las características ortográficas y léxicas sirven mejor para la discriminación que las léxico-sintácticas. Otro trabajo a destacar fue el de Sadat et al. [22]. Los autores recopilaron textos de blogs y foros de 18 países que representan 18 variedades del árabe, divididos en seis grupos: i) la egipcia; ii) la iraquí; iii) la del golfo, que incluye las variedades del Bahrein, Emiratos, Kuwait, Qatar, Omán y Arabia Saudí; iv) la magrebí, que incluye las variedades de Algeria, Túnez, Marruecos, Libia y la Mauritania; v) la levantina, que incluye Jordania, Líbano, Palestina y Siria; vi) la de Sudán. Los autores experimentaron con uni-gramas, bi-gramas y tri-gramas de caracteres para averiguar el efecto de los afijos en la discriminación. Como clasificadores emplearon, por un lado, modelos de lenguaje basados en Modelos Ocultos de Markov y por el otro Naïve Bayes. Concluyeron que distintos afijos (valores de n en los n-gramas) ayudan a diferenciar distintas variedades, mostrando *accuracies* superiores al 96 % y *F-scores*[3] del 70-80 %. En global, obtuvieron el mejor resultado con bi-gramas de caracteres y Naïve Bayes, reportando un *accuracy* global del 98 % evaluando el corpus con una partición 50-50 % para entrenamiento y test. En cuanto al español, Maier y Rodríguez [23] recopilaron 100.000 tweets de cinco países: Argentina, Colombia, España, México y Chile. Dedicaron el 80 % de los tweets de cada país para entrenamiento, el 10 % para validación y el 10 % restante para test. Emplearon cuatro métodos: i) perfiles de frecuencias de n-gramas; ii) modelos de lenguaje de n-gramas de caracteres; iii) modelos de lenguaje para n-gramas de sílabas y iv) algoritmo de compresión LZW. Con los modelos de lenguaje para n-gramas de caracteres obtuvieron una *F-score* del 66,96 %, para valores de n iguales o mayores a 6. El modelo de n-gramas de sílabas funcionó algo peor, obteniendo una *F-score* global del 57,88 % para 4-gramas. Finalmente, utilizaron un metaclasificador por voto combinando las salidas de los modelos de lenguaje de caracteres (5-gramas y 6-gramas) y de sílabas (3-gramas y 4-gramas). Con esta aproximación obtuvieron una *F-score* global del 67,72 %.

También resulta interesante hablar de los trabajos en DSL, ya que esta tarea comparte algunas similitudes con LVI. La idiosincrasia cultural puede influir en las diferencias existentes entre las lenguas, así como en las diferencias entre las variedades de una lengua. Tiedemann y Ljubešić [24] se centraron en la identificación del bosnio, el serbio y el croata. Extrajeron un corpus paralelo de un portal de noticias (SETimes) de la región de sureste de Europa para entrenar

[3]La *F-score* indica en qué medida el sistema acierta al clasificar las muestras positivas, así como su capacidad para detectarlas. En la sección 4.4.3 se describe la *F-score*.

sus modelos. Recopilaron otros 600 textos (200 por idioma) de tres recursos *online* para el conjunto de test. En sus primeros experimentos demostraron que los enfoques más recientes para LI ofrecen resultados mejorables, alcanzando un 87,7 % de *accuracy* con `langid.py` [25]. Con palabras como características, los autores abordaron del problema de dos formas: por un lado, un método basado en clasificación de documentos (con clasificador Naïve Bayes) y por el otro un clasificador basado en listas negras de palabras. Estas son listas con palabras que ocurren con frecuencia en un idioma pero no en el otro. Con esa última aproximación obtuvieron un *accuracy* global del 97 %, superando las otras aproximaciones. Más recientemente, Ljubešić y Kranjčić [26] realizaron un trabajo sobre Twitter para la identificación de cuatro idiomas eslavos del sur similares: Bosnio, Croata, Montenegrino y Serbio. Este trabajo se centró en la identificación a nivel de usuario, y para ello recopilaron un corpus con 490 usuarios para entrenamiento y validación, y otro con 101 usuarios para test. Experimentaron con distintas representaciones: bolsa de palabras, tri-gramas de caracteres, 6-gramas de caracteres y palabras y 6-gramas de caracteres. Probaron con distintos clasificadores: Naïve Bayes, Árboles de Decisión, vecino más cercano y SVM lineales. El mejor resultado lo obtuvieron mediante bolsa de palabras con Naïve Bayes, realizando previamente una selección de las 320 características más relevantes con el test estadístico F1 ANOVA. Con esta configuración obtuvieron un 97,97 % bajo un esquema de validación cruzada en 10 bloques, y del 99 % para el conjunto de test.

Este tipo de tareas son un tema de investigación novedoso. Se han organizado recientemente distintos talleres al respecto, como:

- el taller LT4CloseLang[4] en EMNLP 2014, centrada en la identificación de idiomas similares y con el objetivo de poder aplicar los recursos existentes para explotar las peculiaridades de esta tarea.
- el taller VarDial[5] [27] en COLING 2014, con el objetivo de tratar los temas relacionados con variaciones lingüísticas.
- el taller LT4VarDial[6] [28] en RANLP 2015, centrado en la discriminación de idiomas similares y variedades del lenguaje bajo un escenario realista, con textos cortos e idiomas desconocidos para el sistema.

Al tratarse de áreas de investigación novedosas, una de sus principales carencias es la escasez de corpus para la evaluación de LVI y DSL. Normalmente, los investigadores deben recopilar y evaluar su propio corpus para poder abordar esta tarea. Un corpus estandarizado para su evaluación es el DSLCC, utilizado en los talleres VarDial y LT4VarDial y disponible en varias versiones que incluyen distintos conjuntos de idiomas/variedades:

- DSLCC[7], con 20.000 recortes de noticias por cada uno de los siguientes

[4] http://alt.qcri.org/LT4CloseLang/index.html
[5] http://corporavm.uni-koeln.de/vardial/sharedtask.html
[6] http://ttg.uni-saarland.de/lt4vardial2015/dsl.html
[7] https://bitbucket.org/alvations/dslsharedtask2014

idiomas: bosnio, croata, serbio, indonesio, malayo, checo, eslovaco, portugués brasileño, portugués europeo, español peninsular, español argentino, inglés americano e inglés británico.

- DSLCC v2.0[8], con 20.000 recortes de noticias por cada uno de los siguientes idiomas o variedades: búlgaro, macedonio, serbio, croata, bosnio, checo, eslovaco, español argentino, español peninsular, portugués brasileño, portugués europeo, malayo, indonesio y un grupo conteniendo textos escritos en otros idiomas, como por ejemplo el catalán, ruso o tagalo.

- DSLCC v2.1, el mismo que el DSLCC v2.0 pero añadiendo español mexicano y portugués de Macao.

Muchos de los trabajos presentados se han llevado a cabo sobre corpus relacionados con el dominio periodístico, como noticias o recortes de artículos. Debido a la importancia que están cobrando los medios sociales (blogs, foros, redes sociales, etc.) resultaría conveniente disponer de corpus centrados en ellos para la evaluación de estas tareas. Con la finalidad de evaluar la tarea de LVI en los medios sociales se desarrolló el corpus HispaBlogs [29, 30], una colección de blogs en español de hasta cinco países distintos: Argentina, Chile, México, Perú y España. Para cada variedad del idioma, el corpus dispone de 450 blogs para entrenamiento y 200 para test. Por tanto, HispaBlogs dispone en total de 2.250 blogs para entrenamiento y 1.000 para test. Cada uno de estos blogs contiene como mínimo 10 posts, y cada post una longitud mínima de 10 palabras. Todos los blogs tienen una longitud mínima de 100 palabras.

En este trabajo se ha evaluado la tarea de LVI en un medio social de gran proliferación como es Twitter. Para ello se ha construido HispaTweets, un corpus compuesto por usuarios de Twitter para siete variedades del español: de Argentina, Chile, Colombia, España, México, Perú y Venezuela. Se ha liberado este corpus a la comunidad[9].

[8] https://github.com/Simdiva/DSL-Task
[9] https://github.com/autoritas/RD-Lab/tree/master/data/HispaTweets

Capítulo 3

Metodología para la Construcción de un Corpus Anotado con Variedades del Lenguaje: HispaTweets

Dedicamos este capítulo a detallar la metodología seguida para la construcción de un corpus anotado con variedades del lenguaje. Como caso práctico hemos construido HispaTweets. Este corpus está basado en Twitter, una red social que ofrece un servicio de microblogging donde los usuarios publican mensajes cortos (tweets) de hasta 140 caracteres. Los usuarios pueden reemitir los tweets de otros usuarios (retweet) o seguir los mensajes que publican, convirtiéndose en sus *followers*. Los tweets no son únicamente texto plano, sino que existen una serie de metacaracteres que permiten enriquecer el contenido del mensaje. Por ejemplo, el uso de *hashtags* ('#') para asociarle un tema al tweet o el uso del caracter '@' para mencionar a otros usuarios. Los tweets que un usuario emite (o retweetea) se publican en su *timeline*.

Para el estudio del problema de la LVI en Twitter construimos un corpus con usuarios de habla hispana. El proceso de construcción está resumido en la Figura 3.1 y fue aplicado sobre siete[1] países: Argentina, Chile, Colombia, España, México, Perú y Venezuela.

Para llevar a cabo este proceso hemos desarrollado un sistema que actúa en cuatro etapas. En primer lugar el sistema obtiene la latitud y la longitud de las ciudades más pobladas de los países mencionados. Esta información la extraemos de los artículos de la Wikipedia[2] de esas ciudades. Describimos este

[1] Cinco de dichos países coinciden con los de HispaBlogs. Inicialmente realizamos la búsqueda incluyendo otros países, pero los descartamos por no lograr suficiente volumen de usuarios. Estos países fueron Cuba, Bolivia, Ecuador y República Dominicana.

[2] https://es.wikipedia.org/

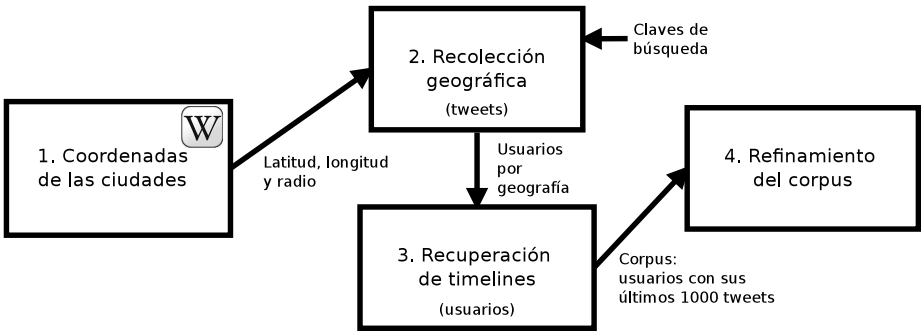


Figura 3.1: Proceso de construcción de HispaTweets.

proceso con más detalle en la sección 3.1, junto con la arquitectura del sistema. La segunda etapa consiste en recopilar aquellos tweets que hayan sido emitidos desde las coordenadas obtenidas. Además de estas coordenadas, para la recolección geográfica necesitamos un radio y unas claves de búsqueda, los cuales escogemos manualmente. Con esta información, el sistema realiza una búsqueda geolocalizada para recolectar aquellos tweets que respondan a las claves de búsqueda proporcionadas y que se hayan emitido desde las coordenadas especificadas, dentro del radio indicado. A partir de los autores de los tweets recuperados elaboramos un listado de usuarios etiquetados geográficamente. Todo este proceso está detallado en la sección 3.2. En la tercera etapa recuperamos las *timelines* de los usuarios obtenidos en la etapa anterior. Este proceso consiste en descargar los 1.000 tweets más recientes de los usuarios recopilados. El estado del corpus tras esta descarga lo detallamos en la sección 3.3. Hasta este punto hemos recopilado un listado de usuarios por geografía y recuperado sus últimos 1.000 tweets. Además de recolectar suficiente información para abordar la tarea, pretendemos que el corpus final esté tan equilibrado como nos sea posible, conteniendo una cantidad de información similar por país. También queremos que esa información sea lo más representativa posible de cada país. Con estas dos motivaciones llevamos a cabo la cuarta y última etapa: el refinamiento del corpus. Este refinamiento consta de tres filtrados: i) filtrado geográfico; ii) filtrado temporal y iii) filtrado por frecuencia. Motivamos y explicamos en detalle todos ellos en la sección 3.4.

3.1 Arquitectura del Sistema

La Figura 3.2 muestra la arquitectura del sistema desarrollado para la recolección de tweets y usuarios. Para la búsqueda geolocalizada es necesario definir los puntos geográficos donde buscar, tarea que realiza el módulo `WikiCities`[3]. Este toma como entrada una serie de ficheros de texto, uno por país, conteniendo cada uno de ellos una URL por línea referente al artículo de una ciudad en

[3] https://github.com/autoritas/RD-Lab/tree/master/src/WikiCities

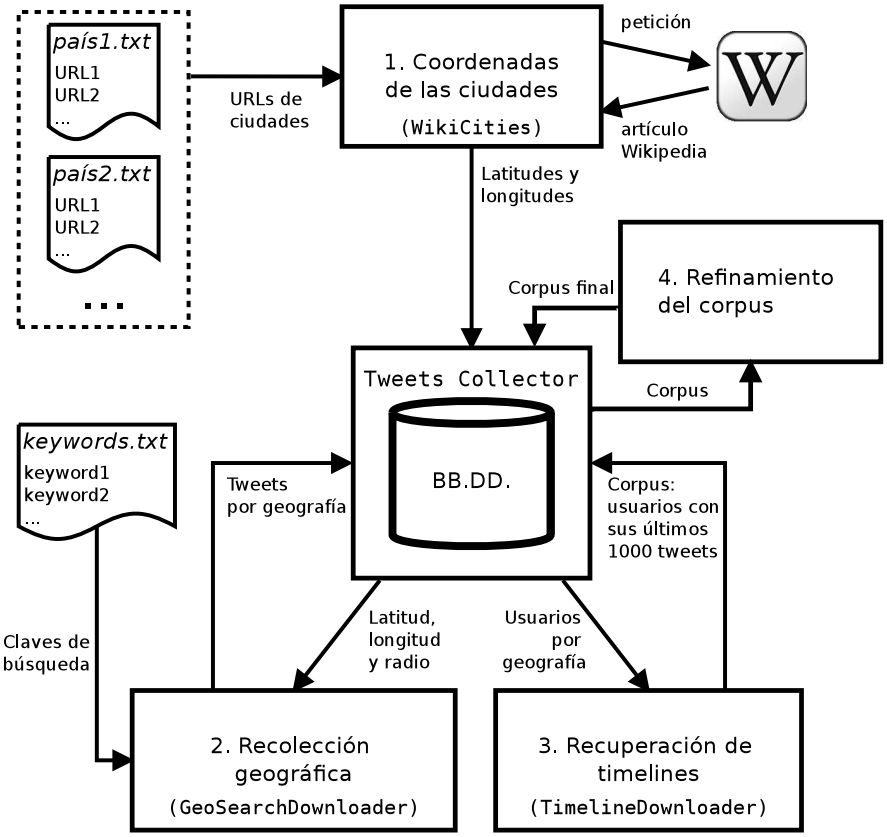


Figura 3.2: Módulos del sistema y flujo de información.

la Wikipedia. Estos artículos pueden obtenerse a partir de unos anexos de la Wikipedia, que contienen para cada país un listado de sus ciudades ordenadas por población. `WikiCities` descarga esos artículos y extrae la información relevante para la búsqueda. De la URL obtenemos el nombre de la ciudad y de la la *InfoBox*[4] las coordenadas geográficas mediante las siguientes expresiones regulares:

- Latitud: $class=\text{“}latitude\text{”}>-*\backslash d+\backslash.\backslash d*$
- Longitud: $class=\text{“}longitude\text{”}>-*\backslash d+\backslash.\backslash d*$

Además, al insertar en la base de datos se tiene en cuenta el orden de inserción de las ciudades para cada país. Esto permite insertar las ciudades de forma ordenada según cualquier criterio arbitrario. En nuestro caso hemos decidido

[4] Una *InfoBox* es una tabla de formato fijo que suele aparecer en la esquina superior izquierda de los artículos de la Wikipedia. Estas tablas suelen mostrar información relevante del artículo, en el caso de las ciudades las coordenadas geográficas.

insertarlas según su población, en orden descendente. Esto permite realizar la búsqueda primero en las ciudades más pobladas.

El siguiente módulo que interviene en el proceso es `GeoSearchDownloader`[5]. Este toma como entrada un fichero con un listado de palabras clave y las ubicaciones previamente obtenidas. Para cada palabra clave, `GeoSearchDownloader` realiza una consulta para cada una de las ubicaciones especificadas por su latitud, longitud y radio. Los tweets recuperados son guardados en la base de datos, a excepción de los retweets. Prescindimos de ellos porque aunque sabemos desde que ubicación han sido emitidos, no podemos asegurar que el autor del tweet original sea de la nacionalidad esperada. Por ejemplo, un usuario en Argentina puede retweetear mensajes de usuarios de España o México.

Tras este proceso conseguimos un listado de tweets por geografía, y con los autores de esos tweets elaboramos un listado de usuarios por geografía. `TimelineDownloader`[6] recibe este listado de usuarios y descarga hasta los últimos 1.000 tweets de sus respectivas *timelines*. Con el corpus obtenido tras todo el proceso, realizamos una serie de refinamientos para la elaboración del corpus final, consistente en filtrar los usuarios y tweets atendiendo a los tres criterios descritos en la sección 3.4.

3.2 Recolección Geográfica

Como hemos comentado, para realizar la búsqueda geolocalizada necesitamos i) listado de ubicaciones con un radio y ii) listado de palabras clave. En cuanto a las ubicaciones, extraemos de la Wikipedia la latitud y la de longitud las ciudades más pobladas de los siete países mencionados y los insertamos en nuestra base de datos, tal y como hemos descrito en la sección 3.1. No podemos insertar todas las ciudades disponibles en los anexos de la Wikipedia porque hay artículos en los que no aparece la ubicación. En esos casos, el sistema ignora esa ciudad y trata de insertar la siguiente. La Tabla A.1 en el Apéndice A muestra aquellas ciudades que no hemos podido añadir.

Insertamos aproximadamente las 50 ciudades más pobladas de cada uno de los siete países mencionados, a excepción de Chile que sólo dispone de 25. A continuación, realizamos una primera búsqueda geolocalizada con las 15 ciudades más pobladas de cada país. Tras esta búsqueda obtuvimos suficiente volumen de tweets para todos los países salvo Chile, Colombia y Perú. En estos tres casos repetimos las búsquedas empleando todas aquellas ciudades de las cuales disponíamos. La Tabla 3.1 muestra el número de ciudades insertadas en la base de datos para cada país, junto con el número de ciudades para los que llevamos a cabo la descarga. Las Tablas A.3–A.9 del Apéndice A detallan las ubicaciones empleadas para las búsquedas. En todas las búsquedas hemos empleado un radio de 8,5 km, suficientemente grande como para captar una buena cantidad de tweets de las grandes ciudades y sin invadir ciudades vecinas, en la medida de lo posible.

[5] https://github.com/autoritas/RD-Lab/tree/master/src/GeoSearchDownloader
[6] https://github.com/autoritas/RD-Lab/tree/master/src/TimelineDownloader

País	Núm. ciudades (base de datos)	Núm. ciudades (búsqueda)
Argentina	50	15
Chile	25	25
Colombia	49	49
España	50	15
México	48	15
Perú	51	51
Venezuela	48	15
Total	321	185

Tabla 3.1: Número de ciudades añadidas a la base de datos para cada país, junto con el número de ciudades para las cuales se ha realizado la búsqueda.

En cuanto a las claves de búsqueda, hemos utilizado 35 palabras clave relacionadas con distintos temas como tecnología, cultura, política/actualidad y salud/alimentación. Como claves hemos tratado de escoger palabras que son usadas comúnmente en todos los países de habla hispana. Así se pretende minimizar el sesgo producido por las claves de búsqueda. En la Tabla A.18 del Apéndice A mostramos los resultados de la búsqueda para cada país y palabra clave.

La Tabla 3.2 muestra el número de tweets y usuarios obtenidos para cada país tras ejecutar la búsqueda, junto con el número medio de tweets por usuario. También mostramos la población y la proporción entre el número de usuarios obtenidos y la población.[7]

Argentina y México son los países para los que hemos recuperado mayor cantidad de tweets, 4.051 y 3.773 respectivamente, seguidos de Venezuela con 3.247 y España con 2.922. La cantidad de tweets ha sido algo menor en Colombia, Chile y sobre todo Perú, que con 2.049 tweets ha sido el país para el que menos hemos obtenido. Resulta llamativo que aunque hayamos descargado más tweets de Argentina que de México, el número de usuarios haya sido mayor en México (2.684) que en Argentina (2.393). En Venezuela también hemos conseguido una alta cantidad de tweets, bastante superior a Chile y Colombia. Sin embargo hemos recuperado una cantidad similar o menor de usuarios que en estos países. El país que menos usuarios y tweets ha obtenido es Perú, con 1.113 usuarios. Vemos que para cada país recuperamos 1,71 tweets por usuario de media, con una desviación de 0,28. Llama la atención el caso de Venezuela, donde conseguimos 2,32 tweets por usuario. Argentina es el país que parece tener una mayor penetración la red social con una proporción entre usuarios y población del 0,023 %, seguido de España con un 0,018 %. Esta proporción es un poco más reducida en Chile y México, con un 0,012 % y 0,011 % respectivamente. Colombia, Perú y Venezuela son las que presentan una proporción menor, alrededor del 0,006 %. Las Tablas A.10–A.16 del Apéndice A presentan estos

[7] Con estos últimos datos pretendemos mostrar el posible sesgo producido por la penetración de la red social en el país.

País	Tweets (búsqueda)	Usuarios	Tweets/ usuario	Población	Usuarios/ población (%)
Argentina	4.051	2.393	1,69	10.484.487	0,02282
Chile	2.399	1.378	1,74	11.381.781	0,01211
Colombia	2.590	1.719	1,51	27.653.833	0,00622
España	2.922	1.909	1,53	10.554.373	0,01809
México	3.773	2.684	1,41	24.502.948	0,01095
Perú	2.049	1.113	1,84	19.160.865	0,00581
Venezuela	3.247	1.400	2,32	23.986.536	0,00584
Total	21.031	12.596	1,67	127.724.823	0,00986
Media	3.004,43	1.799,43	1,71	18.246.403,29	0,01169
SDev	676,42	529,37	0,28	6.844.885,09	0,00614

Tabla 3.2: Número de usuarios y tweets obtenidos para cada país mediante la búsqueda geolocalizada. La población viene dada por la suma de las poblaciones de aquellas ciudades empleadas para la búsqueda. También mostramos las cifras totales, así como la media y la desviación típica para los totales de los países.

mismos resultados desglosados a nivel de ciudad.

3.3 Recuperación de Timelines

El siguiente paso consiste en recuperar los últimos 1.000 tweets de las *timelines* de los usuarios obtenidos en el paso anterior. La Tabla 3.3 muestra el estado del corpus tras este proceso. Más concretamente mostramos el número de usuarios, el número de tweets tras la descarga, el número de tweets por usuario y la longitud media de los tweets, en palabras y caracteres. Vemos que aquel país para el cual recuperamos más tweets es Argentina con 1.684.662, seguido de México con 1.571.859 tweets. En los casos de Chile, Colombia y España hemos obtenido entre 1.050.000 y 1.115.000 tweets. Perú y Venezuela son los países que menos tweets han recuperado, con 676.623 y 715.987 tweets respectivamente. En total disponemos de casi ocho millones de tweets, algo más de 1.100.000 tweets por país de media. La cantidad de tweets por usuario es superior en Chile (769,03) y Argentina (704). Esta proporción para el resto de los países oscila entre los 585 y 630 aproximadamente, salvo Venezuela que presenta 511,42 tweets por usuario. En cuanto a la longitud de los tweets, vemos que oscila entre los 65-90 caracteres y entre 10 y 14 palabras. Los usuarios de Venezuela y España son los que escriben tweets más largos y los de Argentina los más cortos. Vemos que de media los tweets difieren en menos de una palabra (poco más de siete caracteres) entre si. Las Tablas A.19–A.25 en el Apéndice A contienen estas mismas estadísticas para cada ciudad.

País	Usuarios	Tweets (*timelines*)	Tweets/ usuario	Longitud media	
				Palabras	Caracteres
Argentina	2.393	1.684.662	704,00	10,72	65,18
Chile	1.378	1.059.724	769,03	11,71	77,45
Colombia	1.719	1.084.965	631,16	12,13	79,96
España	1.909	1.148.523	601,64	12,81	84,12
México	2.684	1.571.859	585,64	11,82	76,87
Perú	1.113	676.623	607,93	11,34	72,87
Venezuela	1.400	715.987	511,42	13,55	90,33
Total	12.596	7.942.343	630,88	11,87	76,81
Media	1.799,44	1.134.620,43	630,12	12,01	78,11
SDev	529,37	355.970,97	77,58	0,87	7,42

Tabla 3.3: Número de usuarios, tweets, proporción de tweets por usuario y longitud media (en palabras y caracteres) de los tweets tras la descarga de las *timelines*. También mostramos las cifras totales, así como la media y la desviación típica para los totales de los países.

La Tabla 3.4 presenta un análisis relativo a la fecha de publicación de los tweets. Por un lado, presentamos las fechas del primer y último tweet de cada país, independientemente del usuario, y la diferencia en meses entre ambas. Vemos que para la mayoría de los países el primer tweet disponible es de Febrero-Marzo de 2009, mientras que el último es de finales de 2015 o principios de 2016. España es el país que mayor periodo abarca, desde Julio de 2007 hasta finales de 2015, con 102,93 meses de diferencia, seguido de Argentina con 93,93 meses de diferencia. En el resto de países esta diferencia es cercana al 82-83, algo menor en Venezuela (77,03). Las Tablas A.26–A.32 del Apéndice A muestran este mismo análisis de las fechas a nivel de ciudad.

País	Primer tweet	Último tweet	Diferencia (meses)
Argentina	08-04-2008	27-12-2015	93,93
Chile	23-04-2009	04-01-2016	81,57
Colombia	03-03-2009	05-01-2016	83,30
España	16-07-2007	29-12-2015	102,93
México	27-03-2009	30-12-2015	82,30
Perú	01-04-2009	21-01-2016	82,83
Venezuela	03-09-2009	01-01-2016	77,03
Total	16-07-2007	21-01-2016	103,67

Tabla 3.4: Fechas del primer y último tweet para cada país y la diferencia en meses, independientemente del autor.

Finalmente, en la Tabla 3.5 mostramos las medidas de posición para las diferencias entre las fechas del primer y el último tweet de cada usuario, individualmente. Vemos el primer cuartil suele ubicarse entre los 2 y 3 meses, algo superior en Chile y Colombia y algo inferior España, cercano a un un mes y en México que es 0. Por otro lado, el tercer cuartil se posiciona entre los 17 y 31 meses. La mediana ronda los 10 meses, con una desviación típica de 2,24 meses. Vemos que para el total el rango intercuartílico es de 22,64 meses. En cuanto a los máximos, en general están entre los 76 y 84 meses a excepción de Argentina que llega a los 93,83 y España con 102,40 meses. Observamos que para la mayoría de los usuarios la media está entre 12 y 19 meses, con una desviación típica algo mayor. De nuevo, mostramos estos datos desglosados por ciudad en el las Tablas A.33–A.39 del Apéndice A.

País	Min	1Q	Media	SDev	Mediana	3Q	Max
Argentina	0	2,13	12,73	15,38	6,53	17,60	93,83
Chile	0	3,30	16,58	17,70	10,07	24,42	81,23
Colombia	0	3,12	18,11	17,80	12,67	29,15	83,17
España	0	1,10	15,44	16,51	10,17	24,50	102,40
México	0	0,00	14,97	17,61	7,95	23,71	81,83
Perú	0	2,80	17,47	18,86	10,87	26,33	79,87
Venezuela	0	2,33	19,94	20,43	13,37	31,28	76,27
Total	0	2,00	15,99	17,67	9,63	24,64	102,40
Media	0	2,11	16,46	17,76	10,23	25,28	85,51
SDev	0	1,18	1,78	1,10	1,99	3,49	8,26

Tabla 3.5: Medidas de posición para la diferencia en meses entre el primer y último tweet de cada usuario individualmente. Mostramos el mínimo, el primer cuartil (1Q), la media, la desviación típica, la mediana, el tercer cuartil (3Q) y el máximo. También mostramos estas estadísticas sobre el total de los usuarios, así como la media y la desviación típica para los totales de los países.

3.4 Refinamiento del Corpus

Una vez recopilado el corpus previamente descrito, procedemos a la construcción del corpus final. El objetivo es, por un lado, que el corpus resultante sea representativo de las variedades implicadas, y por el otro que esté tan equilibrado como sea posible, con una cantidad de información similar para todos los países. Con estas finalidades realizamos tres filtrados sobre el corpus:

- **Filtrado geográfico**, donde eliminamos aquellos usuarios pertenecientes a ciudades fronterizas o cercanas a la frontera.
- **Filtrado temporal**, donde prescindimos de aquellos tweets que no se encuentren entre el periodo que establecemos.

- **Filtrado por frecuencia**, donde conservamos aquellos usuarios con un número mínimo de tweets.

3.4.1 Filtrado Geográfico

En primer lugar realizamos un filtrado geográfico, que consiste en prescindir de aquellos usuarios obtenidos para ciudades fronterizas o cercanas a la frontera. Este filtrado se ve motivado porque la variedad del idioma en estas ciudades suele verse afectado por las lenguas o variedades de los países vecinos. Por tanto, consideramos que dichas ciudades no son representativas la variedad del idioma a identificar. Este filtrado lo realizamos manualmente.

La Tabla 3.6 muestra aquellas ciudades de las que hemos decidido prescindir, junto con la cantidad de usuarios y tweets que perdemos. En total descartamos 15 ciudades, perdiendo 428 usuarios y 275.445 tweets. La cantidad de información perdida no resulta relevante comparada con la disponible. El corpus contiene en total 12.168 usuarios y 7.666.898 tweets después de este filtrado. El país con menor volumen de usuarios es Perú, siendo por tanto un elemento crítico. Sin embargo, para este país únicamente perdemos 12 usuarios y 8.687 tweets, frente a los 1.101 usuarios y 667.936 tweets disponibles.

País	Ciudades descartadas	Num. usuarios perdidos	Num. tweets perdidos
Argentina	Posadas, Resistencia y Corrientes	183 (7,65 %)	128.578 (7,63 %)
Chile	Arica y Coyhaque	25 (1,81 %)	16.115 (1,52 %)
Colombia	Cúcuta, Maicao y Ipiales	52 (3,02 %)	31.899 (2,94 %)
España	Vigo	29 (1,52 %)	19.091 (1,66 %)
México	Juárez y Tijuana	35 (1,30 %)	21.813 (1,39 %)
Perú	Puerto Maldonado, Tumbes y Tacna	12 (1,08 %)	8.687 (1,28 %)
Venezuela	San Cristóbal	92 (6,57 %)	49.262 (7,38 %)
Total	15 ciudades	428 (3,40 %)	275.445 (3,47 %)

Tabla 3.6: Ciudades descartadas para cada país tras el filtrado geográfico, junto con el número de usuarios y tweets perdidos.

3.4.2 Filtrado Temporal

El siguiente filtrado que realizamos es el temporal, que consiste en conservar aquellos tweets publicados entre las dos fechas que escojamos, siendo las mismas para todos los países. La motivación para realizar este filtrado es que las lenguas evolucionan constantemente en el tiempo y cada vez más rápidamente. Resulta obvio que el lenguaje y los términos empleados actualmente no son los mismos

que los utilizados hace 5 ó 10 años, pues a lo largo del tiempo se habla de distintos temas y surgen nuevas necesidades o tendencias que dan lugar a nuevos términos. Por ejemplo, antes del auge de las redes sociales y de los dispositivos inteligentes no existían términos como "tuitear", "guasapear" o "selfie".

Teniendo en cuenta estos hechos, se ha considerado que era adecuado descargar los últimos 1.000 tweets de cada usuario, como hemos explicado anteriormente en la sección 3.3. Este número permite obtener suficiente información por usuario sin obtener tweets demasiado alejados en el tiempo. En cualquier caso, resulta adecuado realizar un acotado temporal para evitar que los tweets contengan términos que no sean representativos del idioma actual. En la Tabla 3.4 de la sección 3.3 hemos visto que la fecha de los últimos tweets es a finales de diciembre de 2015 y principios de enero de 2016, mientras que las fechas iniciales están más dispersas (entre 2007 y 2009). Esto ocurre porque descargamos los tweets más recientes de cada usuario, por tanto tendemos a obtener los más cercanos a 2016.

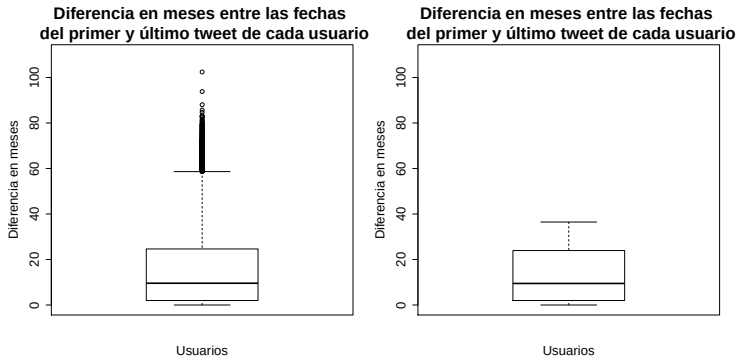
Si además analizamos la Tabla 3.5 con las medidas de posición, observamos que la mediana está ubicada alrededor de los 9-10 meses y el tercer cuartil supera ligeramente los 30 meses. Con estos datos decidimos conservar aquellos tweets comprendidos en un periodo de 36 meses, tres años naturales, publicados del 1 de enero de 2013 al 1 de enero de 2016.

Como la mayoría de los tweets se concentran en fechas cercanas al inicio de 2016, este filtrado no afecta demasiado a la cantidad de tweets que conservamos, como se deduce de la Tabla 3.7. En total, perdemos 296.269 tweets y conservamos 7.370.629.

País	Tweets (antes)	Tweets (después)	Diferencia
Argentina	1.556.084	1.529.616	26.468 (1,70 %)
Chile	1.043.609	1.000.448	43.161 (4,14 %)
Colombia	1.053.066	1.008.202	44.864 (4,26 %)
España	1.129.432	1.088.459	40.973 (3,63 %)
México	1.550.046	1.476.413	73.633 (4,75 %)
Perú	667.936	631.447	36.489 (5,46 %)
Venezuela	666.725	636.044	30.681 (4,60 %)
Total	7.666.898	7.370.629	296.269 (3,86 %)

Tabla 3.7: Número de tweets antes y después del filtrado temporal para cada país.

La Figura 3.3 muestra el efecto del filtrado temporal. Observamos que antes del filtrado (izquierda) existen valores atípicos a partir de los 60 meses aproximadamente. Con el acotado temporal eliminamos esta dispersión.



Antes del filtrado temporal. Después del filtrado temporal.

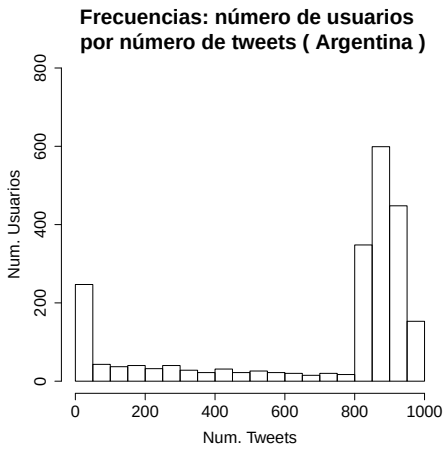
Figura 3.3: Estadísticos para la diferencia en meses entre el primer y último tweet de cada usuario, antes y después del filtrado temporal.

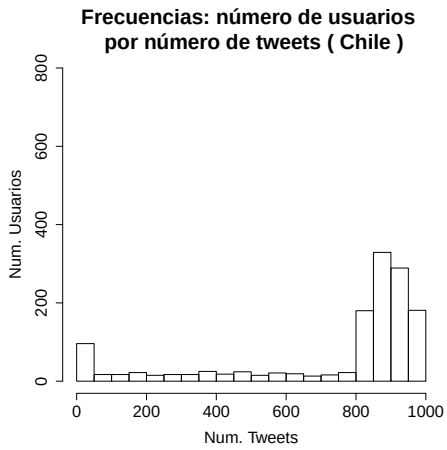
3.4.3 Filtrado por Frecuencia

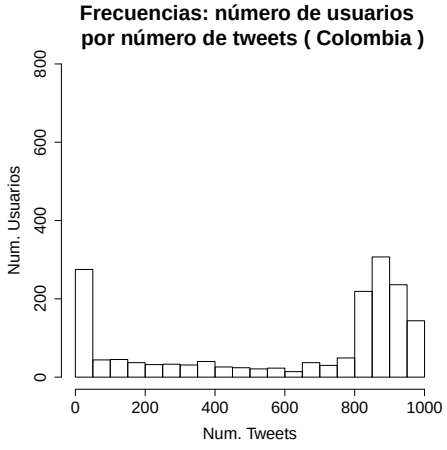
Hay que recordar que uno de nuestros objetivos es obtener un corpus equilibrado, que disponga de la misma cantidad de información para cada país en la medida de lo posible. Esto implica que el número de muestras (usuarios) para cada clase debe ser similar, pero también la cantidad de tweets y como se distribuyen entre los usuarios. Con esta finalidad aplicamos el tercer filtrado, relativo al número de usuarios.

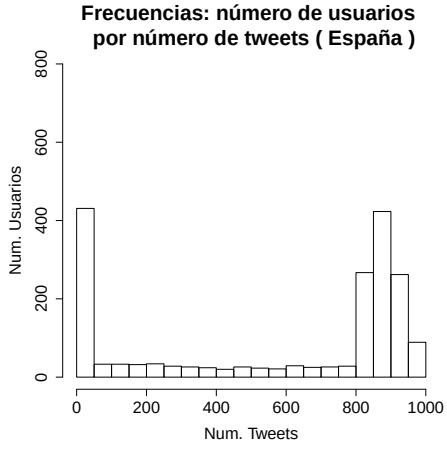
La Figura 3.4 muestra un histograma con las frecuencias del número de tweets para cada país. En el eje horizontal mostramos en número de tweets que puede haber por usuario, entre 1 y 1.000. En el eje vertical, el número de usuarios para los que recuperamos esa cantidad de tweets. Observamos que los histogramas presentan formas similares para todos los países. Hemos obtenido un número elevado de usuarios con entre 0 y 50 tweets, pocos con 100-800 y alcanzamos los máximos con entre 800 y 1.000 tweets. Que la cantidad de tweets se distribuya de forma similar entre los usuarios de cada país es un buen indicativo de cara a construir un corpus equilibrado.

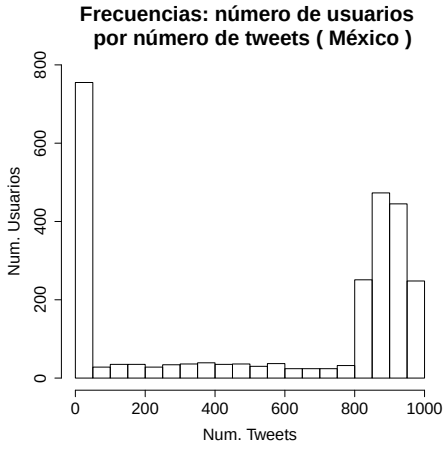
Teniendo en cuenta que queremos conservar la misma cantidad de usuarios para cada país y distribuidos de igual forma, escogemos conservar aquellos usuarios con 500 tweets como mínimo. De entre los restantes elegimos aleatoriamente 650 usuarios de cada país.

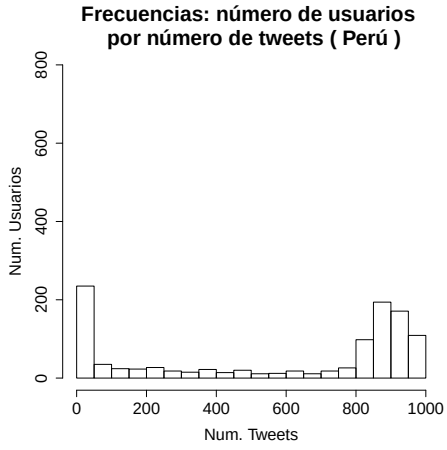

Frecuencias: número de usuarios
por número de tweets (Argentina)
Num. Usuarios
0
200
400
600
800
0
200
400
600
800
1000
Num. Tweets


Frecuencias: número de usuarios
por número de tweets (Chile)
Num. Usuarios
0
200
400
600
800
0
200
400
600
800
1000
Num. Tweets


Frecuencias: número de usuarios
por número de tweets (Colombia)
Num. Usuarios
0
200
400
600
800
0
200
400
600
800
1000
Num. Tweets


Frecuencias: número de usuarios
por número de tweets (España)
Num. Usuarios
0
200
400
600
800
0
200
400
600
800
1000
Num. Tweets


Frecuencias: número de usuarios
por número de tweets (México)
Num. Usuarios
0
200
400
600
800
0
200
400
600
800
1000
Num. Tweets


Frecuencias: número de usuarios
por número de tweets (Perú)
Num. Usuarios
0
200
400
600
800
0
200
400
600
800
1000
Num. Tweets

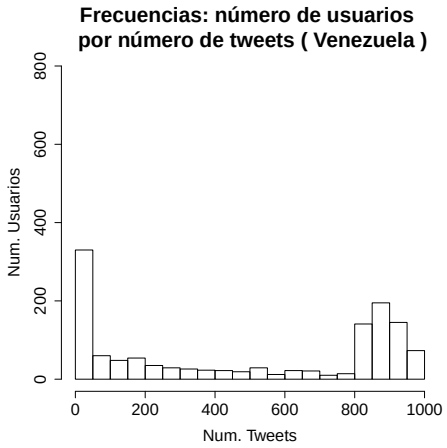


Figura 3.4: Histogramas de frecuencias: número de tweets frente a número de usuarios para Argentina, Chile, Colombia, España, México, Perú y Venezuela.

3.4.4 Corpus Final

La Tabla 3.8 muestra el estado del corpus final, tras aplicar los tres filtrados y la selección de usuarios. En las Tablas A.40–A.46 del Apéndice A mostramos la misma información desglosada a nivel de ciudad.

País	Usuarios	Tweets (*timelines*)	Tweets/ usuario	Longitud media Palabras	Caracteres
Argentina	650	566.113	870,94	10,63	64,32
Chile	650	569.190	875,68	11,71	77,51
Colombia	650	559.619	860,95	12,17	80,34
España	650	558.253	858,85	12,72	83,91
México	650	567.734	873,44	11,75	76,42
Perú	650	564.203	868,00	11,26	72,35
Venezuela	650	552.978	850,74	13,54	90,94
Total	4.550	3.938.090	865,51	11,96	77,91
Media	650	562.584,29	865,51	11,97	77,97
Sdev	0,00	5.412,43	8,33	0,89	7,83

Tabla 3.8: Número de usuarios, tweets, proporción de tweets por usuario y longitud media (en palabras y caracteres) para el corpus final, tras los tres filtrados y la selección de usuarios. También mostramos las cifras totales, así como la media y la desviación típica para los totales de los países.

Observamos que el corpus resultante está bien equilibrado. Contamos con 650 usuarios por país y con una cantidad similar de tweets. Hay poca diferencia entre Chile, el país con más tweets (581.165 tweets) y Venezuela, el país con menos (560.192 tweets) De media contamos con 561.884 tweets por país, con una desviación cercana a 6.000 tweets. En total recopilamos 4.550 usuarios y 3.933.185 tweets. Vemos que la cantidad de tweets por usuario también es similar para todos los países, con 861,83 como mínimo (en Venezuela) y 894,10 como máximo (en Chile). Llama la atención que la desviación es de menos de 10 tweets por usuario. En cuanto a la longitud media de los tweets, no varía significativamente respecto a la que hemos obtenido en los anteriores pasos, ni en palabras ni caracteres. Lo que llama la atención es que el total y la media están muy cercanos, junto con una desviación bastante reducida. Según la desviación, los tweets difieren en menos de una palabra (u ocho caracteres) los unos de los otros.

La Tabla 3.9 muestra las fechas del primer y último tweet de cada país en el corpus final junto con la diferencia en meses, independientemente del usuario. Observamos que la fecha inicial es la marcada y que la fecha final varia por días de un país a otro, siendo insignificante la diferencia en meses. Las Tablas A.47–A.53 del Apéndice A muestran estos datos desglosados por ciudad.

País	Primer tweet	Último tweet	Diferencia (meses)
Argentina	01-01-2013	27-12-2015	36,30
Chile	01-01-2013	31-12-2015	36,47
Colombia	01-01-2013	31-12-2015	36,47
España	01-01-2013	29-12-2015	36,40
México	01-01-2013	30-12-2015	36,43
Perú	01-01-2013	31-12-2015	36,47
Venezuela	01-01-2013	31-12-2015	36,47
Total	01-01-2013	31-12-2015	36,47

Tabla 3.9: Fechas del primer y último tweet para cada país y la diferencia en meses, independientemente del autor.

La Tabla 3.10 explica la distribución de la diferencia entre el primer y último tweet de cada usuario individualmente, mostrando la media, la desviación típica y las medidas de posición. Vemos que este periodo es de 14,70 meses de media, con una desviación de 11,22. Antes del filtrado temporal la desviación era de 17,67 (ver Tabla 3.4), con lo cual hemos reducido la dispersión. El mínimo es superior a 0 debido a que no hay ningún usuario con un sólo tweet publicado. El máximo se ha estabilizado en los 36 meses y medio aproximadamente. En las Tablas A.54–A.60 del Apéndice A también detallamos estos resultados a nivel de ciudad.

País	Min	1Q	Media	SDev	Mediana	3Q	Max
Argentina	0,27	3,07	10,73	9,64	7,45	15,38	36,20
Chile	0,23	4,54	13,59	10,89	10,58	20,58	36,47
Colombia	0,10	6,26	16,73	11,62	14,52	26,92	36,47
España	0,20	6,21	15,83	11,10	13,90	24,16	36,37
México	0,47	5,83	15,56	11,19	13,07	24,67	36,43
Perú	0,20	4,84	14,03	10,56	11,45	21,47	36,47
Venezuela	0,30	6,42	16,63	11,47	14,98	26,17	36,47
Total	0,10	5,13	14,73	11,11	12,13	22,83	36,47
Media	0,25	5,31	12,28	14,73	22,76	36,41	10,92
SDev	0,11	1,14	2,46	1,97	3,69	0,09	0,62

Tabla 3.10: Medidas de posición para la diferencia en meses entre el primer y último tweet de cada usuario individualmente y por país, para el corpus final. Mostramos el mínimo, el primer cuartil (1Q), la media, la desviación típica, la mediana, el tercer cuartil (3Q) y el máximo. También mostramos estas estadísticas sobre el total de los usuarios, así como la media y la desviación típica para los totales de los países.

Capítulo 4

Marco de Evaluación

Con el objetivo de evaluar la calidad del corpus HispaTweets construido con la metodología previamente descrita, en este capítulo describimos algunas de las técnicas más comunes en el estado del arte para abordar este tipo de problemas. Experimentamos con distintas representaciones de documentos basadas en n-gramas de palabras y de caracteres. También empleamos TF-IDF, que permite ponderar las frecuencias de términos para dar mayor puntuación a aquellos que tengan más relevancia. Aparte de estas técnicas, también aplicamos el método de representación en baja dimensionalidad o *Low-Dimensionality Representation* (LDR) [30], que representa cada documento mediante un número reducido de características. Para la identificación de la variedad hemos escogido distintos tipos de clasificadores: Naïve Bayes, Árboles de Decisión y Máquinas de Vectores de Soporte.

Además de estas aproximaciones, también abordamos el problema con un algoritmo (localización por perfil) que trata de predecir la ubicación de un usuario a partir su perfil de Twitter mediante el campo *location*, contrastando la información de este campo con las ciudades y países de nuestra base de datos. El objetivo de esta aproximación es evaluar los resultados ofrecidos por un algoritmo que emplee únicamente esta información, además de servir como referencia para comparar con las técnicas de LVI.

En la sección 4.1 describimos el algoritmo de localización por perfil. Seguidamente, en la sección 4.2 explicamos las distintas representaciones empleadas. En la sección 4.3 describimos brevemente las técnicas utilizadas para la clasificación. Finalmente, en la sección 4.4 explicamos como hemos diseñado los experimentos.

4.1 Algoritmo de Localización por Perfil

En esta sección presentamos un algoritmo que trata de geoposicionar al usuario a partir de la información del campo *location* de su perfil. Este campo fue creado con la idea de que el usuario introdujera su ubicación. Twitter ofrece

valores predeterminados para este campo, aunque variables. Por ejemplo, para un usuario de Valencia Twitter puede sugerir "Valencia", "España", "Valencia, España" o "Valencia, Comunidad Valenciana", entre otros. Además al usuario se le permite introducir texto personalizado en este campo, con lo cual éste puede no tener relevancia de cara a la identificación de su ubicación. Hay ciudades con el mismo nombre en distintos países, como Valencia que existe en España y en Venezuela. Si el usuario escribe solamente la ciudad podemos caer en problemas de ambigüedad, puesto que en principio no es posible predecir de que ciudad se trata solamente con este enfoque. Además, este campo es opcional y es frecuente que el usuario lo deje en blanco, o incluso que se invente ubicaciones como "Narnia", "Marte" o "La Tierra Media".

La idea básica del método consiste en comparar las palabras del campo *location* con los países de de la base de datos. Si hay sólo una coincidencia, asignamos este país al usuario y terminamos. En caso contrario, asumimos que el contenido del campo es una ciudad y tratamos de averiguar a qué país pertenece. Si no coincide ninguna ciudad o coincide más de una para distintos países, consideramos que no somos capaces de identificar el país por problemas de ambigüedad. En caso contrario, al usuario se le asigna el país correspondiente a la ciudad. Cada comparación la realizamos contando el número de coincidencias entre las palabras del campo *location* y las palabras de los países o ciudades. Escogemos aquella ciudad o país con un mayor número de coincidencias. Este método permite, por ejemplo, tratar con la siguiente situación. En México hay una ciudad llamada 'Puebla de Zaragoza', al igual que en España existe 'Zaragoza'. Si el usuario únicamente escribe 'Zaragoza' en su perfil, no somos capaces de desambiguar puesto que en ambos casos coincide una única palabra que puede asociarse a los dos países. En cambio, si el usuario escribe 'Puebla de Zaragoza' habrá una coincidencia con España (Zaragoza) y dos con México (Puebla y Zaragoza), con lo que podemos concluir que es de México. Como hemos comentado, al usuario se le permite introducir texto personalizado en este campo. Si este texto contiene el nombre de un país o ciudad, el algoritmo implícitamente ignora aquellas palabras que no tengan que ver con países o ciudades. Por ejemplo, hay un usuario cuya ubicación dice "México DF Ciudad Surrealista" y que puede ser de México.

Describimos el método en el Algoritmo 1. En primer lugar, normalizamos el campo *location* (línea 1). Es decir, eliminamos los caracteres no alfabéticos y las tildes. Seguidamente, dividimos el texto en palabras y las insertamos en un conjunto (`loc_set`). A continuación comprobamos si alguna de las palabras de este conjunto es uno de los países a identificar (líneas 4-5). Si hay sólo una coincidencia, devolvemos este resultado como predicción (líneas 15-17). Si no hay ninguna, procedemos a comparar las palabras del campo *location* con los nombres de las ciudades (líneas 6-13). Para cada ciudad, creamos un conjunto con las palabras que forman su nombre (`city_set`) y calculamos el número de coincidencias entre ambos conjuntos (`score`), manteniendo siempre la puntuación más alta para cada país (líneas 11-12). Finalmente, recuperamos aquellos países con mayor número de coincidencias (línea 13). En el caso que haya sólo una predicción, la devolvemos. Si no hay ninguna o hay más de una, consideramos

que no disponemos de bastante información para asignarle ningún país.

```
def location_by_profile(location, cities, countries):
    loc_set = set(tokenize(normalize(location)))
    predicted = None
    pred_countries = set([word for word in loc_set
                          if word in countries])
    if len(pred_countries) == 0:
        countries_score = {c : 0 for c in countries}
        for city in cities:
            city_set = set([tok for tok in tokenize(city.name)])
            score = len(loc_set & city_set)
            if score > countries_score[city.country]:
                countries_score[city.country] = score
        pred_countries = get_max(countries_score)

    if len(pred_countries) == 1:
        predicted = pred_countries[0]
    return predicted
```

Algoritmo 1: Algoritmo de localización por perfil.

4.2 Representación de los Documentos

En esta sección describimos las distintas técnicas para la representación de documentos que hemos empleado. Estas técnicas consisten en representar cada documento mediante vectores de características basadas por un lado en n-gramas de caracteres, y por el otro n-gramas de palabras. Las características las ponderamos mediante distintos métodos: frecuencia de términos o *Term Frecuency* (TF), *Term Frequency-Inverse Document Frecuency* (TF-IDF) y el método de representación en baja dimensionalidad o *Low-Dimensionality Representation* (LDR).

4.2.1 Bolsas de n-gramas de Caracteres

La técnica de los n-gramas ha sido aplicada con resultados satisfactorios en una gran variedad de problemas, como la estimación de modelos de lenguaje para reconocimiento automático del habla [31], traducción automática [32] o incluso en campos relacionados con la bioinformática [33]. Es una técnica relativamente sencilla y aplicable de distintas maneras cuando los objetos de estudio son secuencias. Dada una secuencia, definimos un n-grama como una subsecuencia de n elementos consecutivos. Dado que un texto puede considerarse una secuencia de caracteres, podemos construir un conjunto de características basado en n-gramas de caracteres.

El Algoritmo 2 muestra el proceso de extracción de n-gramas de un texto. Dado un valor para n y un documento, el proceso consiste en recorrer el texto del documento con una ventana de tamaño n, guardando en cada iteración las secuencias de n caracteres que caen dentro de la ventana. Esta forma de descomponer el texto (la extracción de sus subsecuencias) es la base de cualquier técnica basada en n-gramas. Esta técnica puede emplearse de distintas maneras. En este trabajo utilizamos una aproximación de bolsas de n-gramas para representar cada documento mediante vectores de características.

```
def n_grams(N, doc):
    ngrams = []
    for i in range(len(doc) - N + 1):
        ngrams.append(doc[i:i+N])
    return ngrams
```

Algoritmo 2: Extracción de n-gramas.

La Tabla 4.1 muestra como quedaría el texto "alcalde" tras su descomposición en bi-gramas y tri-gramas de caracteres.

n-gramas de caracteres obtenidos para el texto: "alcalde"

n	n-gramas de caracteres obtenidos
2	(a, l), (l, c), (c, a), (a, l), (l, d), (d, e)
3	(a, l, c), (l, c, a), (c, a, l), (a, l, d), (l, d, e)

Tabla 4.1: Ejemplo de la extracción de n-gramas de caracteres, para n=2 y n=3.

El proceso de generación de características requiere varios pasos. El primero de ellos es la extracción de términos, descrita en el Algoritmo 3. Se analizan todos los documentos del corpus, segmentando su texto en n-gramas y almacenándolos en un conjunto (bolsa) o vocabulario (V), junto con la frecuencia con la que aparecen (líneas 3-9). A los n-gramas que mantenemos en V también los llamamos "términos". El número de términos resultante suele ser muy elevado, así que para aliviar el coste computacional es habitual considerar únicamente un subconjunto de V que contenga los términos más relevantes. Existen distintas técnicas para la selección de los términos. Una aproximación muy sencilla consiste en conservar aquellos que aparecen con mayor frecuencia. En nuestro caso, los ordenamos descendentemente según su frecuencia y conservamos los T términos más frecuentes (líneas 12-14). El resultado de todo este proceso es un vocabulario V con $|V| = T$ que contiene los términos o n-gramas más frecuentes en el corpus.

```
1  def extract_terms(docs, N):
2      V = dict()
3      for doc in docs:
4          terms = n_grams(N, doc)
5          for term in terms:
6              if not term in V:
7                  V[term] = 1
8              else:
9                  V[term] += 1
10     return V
11
12 def filter_by_freq(V, T):
13     remaining = sorted(V)
14     return remaining[:T]
```

Algoritmo 3: Extracción de términos.

El siguiente paso consiste en generar, para cada documento, un vector de características que contenga la frecuencia con la que cada término $t \in V$ aparece en ese documento. Esta técnica para la generación de características se llama TF. El procedimiento está descrito en el Algoritmo 4. Para cada documento, extraemos sus términos (línea 5) y guardamos el número de veces que cada término aparece en ese documento (líneas 8-12), siempre y cuando ese término esté en V.

El resultado final de la generación de características es la matriz $TF_{M \times T}$ mostrada en la Ecuación 4.1, siendo $D = [d_1, d_2, ..., d_M]$ el conjunto de documentos y $V = [t_1, t_2, ..., t_T]$ el vocabulario de términos. Cada elemento tf_{ij} representa el número de veces que el término $t_j \in V$ aparece en el documento $d_i \in D$. Es decir, cada fila en D representa un documento, y cada columna la frecuencia con la que aparece la palabra asociada.

$$TF = \begin{pmatrix} tf_{11} & tf_{12} & \cdots & tf_{1T} \\ tf_{12} & tf_{22} & \cdots & tf_{2T} \\ \cdots & \cdots & \cdots & \cdots \\ tf_{M1} & tf_{M2} & \cdots & tf_{MT} \end{pmatrix} \tag{4.1}$$

```
def TF(docs, V, N):
    tf = dict()
    for doc in docs:
        tf[doc] = dict()
        doc_terms = n_grams(N, doc)

        for t in doc_terms:
            if t in V:
                if t not in tf[uID]:
                    tf[doc][t] = 1
                else:
                    tf[doc][t] += 1
    return tf

def generate_TF(docs, N, T):
    V = extract_terms(docs, N)
    V = filter_by_freq(V, T)
    return TF(docs, V, N)
```

Algoritmo 4: Generación de características basadas en frecuencias de términos de n-gramas de caracteres.

4.2.2 Bolsas de n-gramas de Palabras

Al principio de la sección 4.2.1 hemos comentado que la técnica de los n-gramas es aplicable cuando los objetos de estudio son secuencias, siendo un texto una secuencia de caracteres. Sin embargo, también podemos considerar el texto como una secuencia de sílabas, palabras, frases o párrafos.

En este trabajo también exploramos el estudio del problema a nivel de palabra. El proceso para la aplicación de esta técnica es el mismo que hemos explicado en la sección 4.2.1, con la diferencia de que es necesario dividir el texto en palabras antes de proceder con la generación de términos y características.

El proceso de dividir el texto en unidades más básicas (bien sean caracteres, palabras, etc.) se llama *tokenización*. Cada una de las unidades en que se divide el texto es un *token*. Nosotros hemos utilizado el tokenizador en palabras de NLTK [34]. Aunque en el resultado también hay *tokens* que no son palabras (por ejemplo, signos de puntuación y emoticonos) utilizamos el término "palabras" indistintamente. La Tabla 4.2 muestra el resultado de descomponer una frase en bi-gramas y tri-gramas.

n-gramas de palabras obtenidos para el texto:
"el alcalde encendió su ordenador"

n	n-gramas de palabras obtenidos
2	(el, alcalde), (alcalde, encendió), (encendió, su), (su, ordenador)
3	(el, alcalde, encendió), (alcalde, encendió, su), (encendió, su, ordenador)

Tabla 4.2: Proceso de extracción de n-gramas de palabras, para n=2 y n=3.

4.2.3 TF-IDF: Ponderación de las Frecuencias de los Términos

Las representaciones que hemos explicado hasta ahora están basadas en la frecuencia en que ciertos términos (n-gramas) aparecen en el texto a analizar. Esta aproximación no modela cómo de relevantes son los términos en un documento. Por ejemplo, las palabras más comunes (como artículos, preposiciones, etc.) tienden a aparecer mucho más frecuentemente que otras y en muchos más documentos. Esto hace que no sean tan útiles para la discriminación. La técnica del TF-IDF trata de atenuar este efecto.

Dado un término t y un documento d, definimos el $tf(d,t)$ como el número de veces que el término t aparece en el documento d. El $idf(t)$ lo definimos como el número de documentos en que aparece t. El TF-IDF devuelve una puntuación $w_{d,t}$ que indica la relevancia que el término t tiene en el documento d. La Ecuación 4.2 muestra como calcular $w_{d,t}$:

$$w_{d,t} = tf(d,t) \cdot \ln\left(\frac{N}{1 + idf(t)}\right) \tag{4.2}$$

Donde N es el número de documentos. La parte izquierda del producto es la frecuencia de términos, como la explicada en las anteriores aproximaciones. En cuanto a la parte derecha, vemos que cuanto mayor es el número de documentos en que aparece t, mayor será el denominador y el cociente será más cercano a uno. Tras aplicar el logaritmo se obtiene un valor próximo a cero, dando lugar a una puntuación más baja del término t en el documento d. Es decir, aquellas palabras que sean comunes a muchos documentos obtendrán una puntuación más baja que aquellas que aparezcan en pocos. Estas últimas son más relevantes para la discriminación. Tras aplicar el TF-IDF en lugar de la frecuencia de términos, la matriz de documentos W queda como sigue:

$$W = \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1T} \\ w_{12} & w_{22} & \cdots & w_{2T} \\ \cdots & \cdots & \cdots & \cdots \\ w_{M1} & w_{M2} & \cdots & w_{MT} \end{pmatrix} \tag{4.3}$$

Donde T es el número de términos y w_{ij} es la puntuación obtenida para el término j-ésimo en el documento i-ésimo según la Ecuación 4.2.

4.2.4 LDR: Método de Representación en Baja Dimensionalidad

Hasta ahora hemos explicado algunas de las técnicas de representación de documentos más comunes: n-gramas de caracteres, palabras y TF-IDF. Además de evaluar el corpus presentado con ellas, también hemos utilizado un método de representación en baja dimensionalidad o *Low-Dimensionality Representation* (LDR) [30] para la generación de características. Este método las genera en función de una puntuación calculada para cada término y variedad, y que indica la confianza con que consideramos que ese término pertenece a esa variedad.

A continuación explicamos el proceso al completo. En primer lugar se obtiene la representación de los documentos basada en el esquema TF-IDF presentado previamente. Una vez obtenida la matriz de documentos W como en la ecuación 4.3, le añadimos una última columna con la etiqueta de clase de cada documento, $\delta(d_i)$. La matriz Δ obtenida se muestra en 4.4.

$$\Delta = \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1T} & \delta(d_1) \\ w_{12} & w_{22} & \cdots & w_{2T} & \delta(d_2) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ w_{M1} & w_{M2} & \cdots & w_{MT} & \delta(d_M) \end{pmatrix} \tag{4.4}$$

La idea básica consiste en, para cada término, calcular una puntuación por clase que indique con qué confianza el término pertenece a cada una de las clases. Esta puntuación está acotada entre 0 y 1, más cercana a 1 cuanto más seguros estemos de que el término pertenece a esa variedad. En base a esas puntuaciones, calculamos un número reducido de características.

Para cada clase c y cada término $t \in V$, calculamos la puntuación $S(t/c)$. Esta puntuación se obtiene sumando los pesos obtenidos con el TF-IDF para el término t en los documentos $d \in D$ pertenecientes a la clase c, y se divide entre la suma de los pesos para ese término en todos los documentos (Ecuación 4.5).

$$S(t,c) = \frac{\sum_{d \in D/c=\delta(d)} w_{dt}}{\sum_{d \in D} w_{dt}}, \forall d \in D, c \in C \tag{4.5}$$

A partir de estas puntuaciones, representamos un documento d como sigue:

$$d = \{F(c_1), F(c_2), \ \ldots \ , F(c_n)\} \sim \forall c \in C \tag{4.6}$$

Cada $F(c_i)$ representa las seis características listadas a continuación:

1. **Media**. Dada por la suma de las puntuaciones de los términos del documento, dividida por el número total de términos que contiene.

2. **Desviación típica**. Calculada como la raíz cuadrada de la suma de todas las puntuaciones menos la media.

3. **Puntuación mínima**. La mínima de las puntuaciones que aparece en el documento.

4. **Puntuación máxima**. La máxima de las puntuaciones que aparece en el documento.

5. **Puntuación global**. La suma de las puntuaciones dividida entre el número total de términos del documento.

6. **Proporción**. Proporción entre el número de términos del vocabulario que aparecen en el documento y el número total de términos del documento.

Es decir, para cada clase (variedad del idioma) se calculan estas seis medidas basándose en las puntuaciones descritas previamente. Por tanto, mediante este método el tamaño del documento como vector de características será de seis multiplicado por el número de clases, lo que supone una reducción considerable de la dimensionalidad.

4.3 Algoritmos de Clasificación

En esta sección describimos los distintos clasificadores utilizados para predecir la variedad del idioma de los documentos. Empleamos métodos bayesianos (clasificador Naïve Bayes), métodos que implican una selección de características (Árboles de Decisión) y Máquinas de Vectores de Soporte o *Support Vector Machines* (SVM). Todas estas herramientas vienen implementados en el *toolkit* `scikit-learn`[1] para Python.

4.3.1 Naïve Bayes

La regla de clasificación de Naïve Bayes es la mostrada en la Ecuación 4.7. Este clasificador está basado en el teorema de Bayes, asumiendo que todas las características son independientes entre si.

$$\hat{c} = \arg\max_{c} P(c) \prod_{i=1}^{n} P(x_i|c) \tag{4.7}$$

Esta asunción resulta bastante fuerte y parece inadecuada para tratar con texto, pues las frases son construidas en base a las restricciones impuestas por el propio idioma. Por ejemplo, un sustantivo y un adjetivo que lo referencie deben concordar en género y número, por tanto no hay independencia entre las palabras que aparecen (ni en su orden). Sin esta asunción, la Ecuación 4.7 debería considerar también las dependencias condicionales entre las características, lo cual supondría un sobrecoste elevado en tiempo de computación. A pesar de todo, estos clasificadores han servido para resolver exitosamente distintos problemas en Procesamiento del Lenguaje Natural (PLN), como la creación de filtros anti-spam [35] u otros problemas de clasificación de documentos [36].

[1] http://scikit-learn.org/

4.3.2 Árboles de Decisión

Los Árboles de Decisión son un método de aprendizaje supervisado utilizado para tareas de clasificación o regresión. La idea básica consiste en inferir una serie de reglas de decisión simples a partir de los datos de entrenamiento.

De forma intuitiva, el proceso de clasificación es el siguiente. Dado un nodo, este se encarga de clasificar la muestra de entrada utilizando una única característica, previamente seleccionada. En función del valor que toma esta característica se escogerá uno de sus hijos, que evaluará otra característica. Este proceso se repite de forma recursiva, empezando por el nodo raíz y hasta que el algoritmo alcance alguna hoja. Cada hoja contiene una etiqueta, que se le asigna a la muestra. A cada iteración, es deseable que el algoritmo escoja la característica que más información aporte, que sea más útil para la discriminación. Un criterio para la seleccionar las características más discriminativas es *Information Gain* [37].

Existen distintas implementaciones y algoritmos para la construcción del árbol de decisión. El *toolkit* `scikit-learn` dispone de la versión CART [38] (*Classification and Regression Trees*).

4.3.3 Máquinas de Vectores de Soporte

Otro de los clasificadores que utilizamos son las SVM [39]. Más concretamente utilizamos las SVM lineales implementadas en scikit-learn, que a su vez están implementadas sobre LIBLINEAR [40].

Las SVM tratan de determinar el hiperplano que separa las muestras de ambas clases, de forma que que el margen entre la distancia entre las muestras más próximas de cualquier clase y el hiperplano sea la máxima. Las SVM hacen uso de métodos basados en *kernels* para operar en espacios de alta dimensionalidad sin tener que calcular todos los puntos explícitamente. La implementación de LIBLINEAR utiliza *kernels* lineales.

4.4 Configuración Experimental

Una vez explicadas las distintas etapas y técnicas que vamos a usar para la evaluación, procedemos a explicar brevemente los corpus empleados, el diseño de los experimentos que hemos realizado y las medidas de evaluación empleadas.

4.4.1 Corpus de Evaluación

Realizamos el proceso de evaluación sobre dos corpus. El primero de ellos es HispaTweets, constituido por usuarios de siete países: Argentina, Chile, Colombia, España, México, Perú y Venezuela. Este corpus cuenta con 650 usuarios por país (4.550 en total) y de cada usuario conservamos entre 500 y 1.000 tweets, 7.942.343 de tweets en total. Todos los tweets del corpus fueron publicados entre el 1 de enero de 2013 y 1 de enero de 2016. Las Tablas 3.8, 3.9 y 3.10 mostradas en el Capítulo 3 detallan sus características.

También establecemos una comparativa con el corpus HispaBlogs [29, 30], compuesto por posts de blogueros de hasta cinco países de habla hispana: Argentina, Chile, España, México y Perú. Para cada variedad del idioma, el corpus dispone de 450 blogs para entrenamiento y 200 para test. En total son de 2.250 blogs para entrenamiento y 1.000 para test, de forma que los autores sólo aparecen en uno de los dos conjuntos. Cada uno de estos blogs contiene como mínimo 10 posts, y cada post una longitud mínima de 10 palabras. Todos los blogs tienen una longitud mínima de 100 palabras.

Los cinco países de HispaBlogs coinciden con cinco de HispaTweets, lo que permite analizar los resultados para ambos corpus comparativamente.

4.4.2 Método Experimental

El marco de experimentación general que aplicamos está resumido en la Figura 4.1.

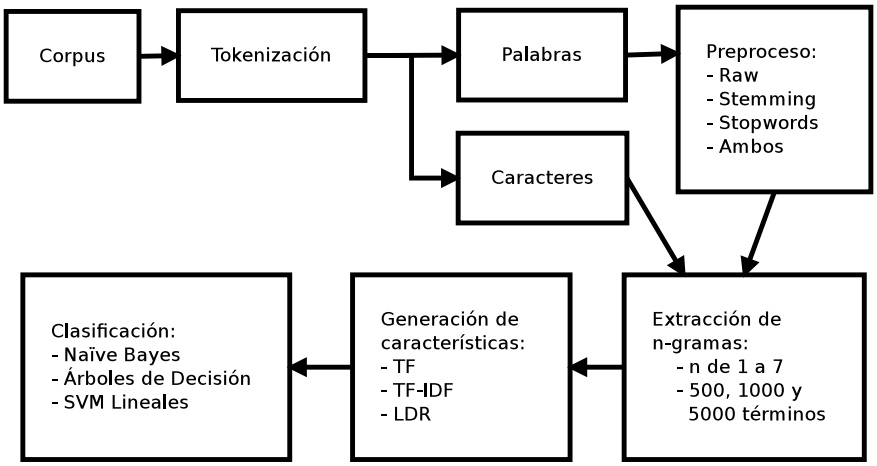


Figura 4.1: Marco experimental.

En primer lugar aplicamos el proceso de tokenización sobre HispaTweets para dividir el texto en secuencias de palabras/caracteres. En el caso de las palabras, probamos a aplicar las opciones de preproceso que explicamos a continuación:

Raw: mantenemos el texto como lo obtenemos de Twitter, con la excepción de que convertimos a minúscula todas las palabras. Esta opción la aplicamos siempre, independientemente de las opciones de preproceso o la representación.

Stemming: consiste en reducir cada palabra a su raíz o base, eliminando los afijos venidos de las inflexiones o derivaciones de la palabra. Por ejemplo, las palabras *librería* o *libros* quedarían como *libr* tras el proceso de *stemming*. Utilizamos el Snowball *stemmer* de NLTK.

Eliminación de las stopwords: las *stopwords* son aquellas palabras que aparecen muy frecuentemente en los documentos y que no aportan información relevante para la clasificación, como artículos o preposiciones. La eliminación de estas palabras suele a mejores resultados en tareas de clasificación de textos [41]. Además, pueden ser útiles en tareas de *Author Profiling* [42]. Utilizamos el listado de *stopwords* para español disponible en NLTK.

Aplicación de ambas: eliminar las *stopwords* y aplicar *stemming* sobre el texto resultante.

Con las secuencias de palabras y caracteres obtenidas construimos distintos vocabularios de n-gramas o términos para distintos valores de n (de 1 a 7) y variando el tamaño de los vocabularios (500, 1000 y 5000 términos). Para cada configuración generamos las características según las técnicas presentadas previamente: TF, TF-IDF, LDR. El último paso es el de clasificación, donde experimentamos con Naïve Bayes, Árboles de Decisión y SVM lineales. Los experimentos están organizados en cuatro conjuntos que mostramos en la Tabla 4.3

Los experimentos descritos los hemos llevado a cabo bajo un esquema de validación cruzada en 5 bloques [2]. En cada experimento mantenemos separados los usuarios de entrenamiento de los de test, durante todo el proceso descrito.

1. Algoritmo de Localización por Perfil	El algoritmo descrito en la sección 4.1 que geoposiciona al usuario a partir de la información de su perfil.
2. n-gramas y LDR sin Preproceso	Estos experimentos incluyen la parte de tokenización y extracción de n-gramas de palabras y n-gramas de caracteres, con un barrido para n de 1 a 7. También analizamos como varía el resultado en función del número de términos. Experimentamos con 500, 1000 y 5000 términos. Utilizamos las tres representaciones: TF, TF-IDF y LDR.
3. Efectos del Preproceso	Sobre aquella configuración que mejor funcione aplicamos las opciones de preproceso previamente descritas.
4. Comparación con HispaBlogs	Establecemos una comparativa entre los resultados obtenidos para HispaTweets e HispaBlogs [30].

Tabla 4.3: Organización de los experimentos.

[2] La evaluación estándar suele ser con 10 bloques, pero como el volumen de información es elevado lo hace prohibitivo. Por ello hemos usado 5 bloques, bajo la suposición de no perder la independencia en la evaluación.

4.4.3 Medidas de Evaluación

Para evaluar el corpus empleamos las cuatro métricas que describimos a continuación.

***Accuracy*:** definida como la relación entre el número de muestras bien clasificadas (N_{hits}) frente al total (N) de muestras del conjunto de test.

$$accuracy = \frac{N_{hits}}{N}$$

***Precision*:** definida como la relación entre los verdaderos positivos (TP) frente la suma de verdaderos positivos (TP) y falsos positivos (FP). Indica la habilidad del clasificador para no etiquetar como positivas las muestras que son negativas.

$$precision = \frac{TP}{TP + FP}$$

***Recall*:** definida como la relación entre el número de verdaderos positivos (TP) frente a la suma de verdaderos positivos y falsos negativos. Indica la habilidad del clasificador para encontrar todas las muestras positivas.

$$recall = \frac{TP}{TP + FN}$$

***F-score*:** definida como la media harmónica entre la *precision* y el *recall*.

$$F\text{-}score = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

Capítulo 5

Resultados Experimentales

En este capítulo describimos los resultados obtenidos durante el proceso de evaluación. Aplicamos las técnicas y representaciones explicados a lo largo del Capítulo 4: algoritmo de localización por perfil; representación de documentos basadas en n-gramas de palabras y caracteres con TF, TF-IDF y LDR para la generación de características y Naïve Bayes, Árboles de Decisión y SVM lineales para la clasificación.

5.1 Algoritmo de Localización por Perfil

En primer lugar mostramos los resultados que hemos obtenido con el algoritmo de localización por perfil. Recordemos que este algoritmo tiene como finalidad cuantificar la proporción de usuarios que podemos identificar mediante la información contenida en el perfil del usuario, en el campo *location*. La Tabla 5.1 muestra los resultados obtenidos aplicando este algoritmo sobre todos los usuarios de HispaTweets.

Predicción	Usuarios	% Usuarios
Correcta	2.731	60,02
Incorrecta	129	2,84
Indefinible	1.690	37,14
Total	4.550	100,00

Tabla 5.1: Resultados del algoritmo de localización por perfil sobre HispaTweets.

Etiquetamos correctamente 2.731 usuarios de 4.550 disponibles, un 60,02 % del total. No somos capaces de etiquetar el 37,14 % de los usuarios, lo cual ocurre por tres razones: i) campo *location* vacío; ii) frases personales no relacionadas con la ubicación o iii) nombres de ciudades existentes en más de un país. El número de usuarios que etiquetamos erróneamente no llega al 3 % del total. Hay principalmente dos causas: i) la ubicación escrita no coincide con el país

de descarga. Por ejemplo, usuarios cuya ubicación indica "España" pero sus tweets los hemos encontrado en Perú; ii) la ubicación escrita no es una ubicación realmente, pero sus palabras coinciden con los nombres de las ciudades. Por ejemplo, en el corpus hay un usuario cuya ubicación dice: "en una piña, debajo del mar". Hay una coincidencia con Mar del Plata, ciudad de Argentina, por lo que el usuario es etiquetado como de Argentina.

Aunque hay una cantidad considerable de usuarios que no podemos etiquetar, resulta interesante analizar por separado el subconjunto de usuarios que si ha sido etiquetado. La Tabla 5.2 muestra el número de usuarios etiquetados de cada país, el *accuracy* global obtenido, la *precision*, el *recall* y la *F-score* para cada uno de los países y en total.

Accuracy: 0.95

País	Número usuarios	*Precision*	*Recall*	*F-score*
Argentina	358	0,91	0,96	0,93
Chile	403	0,97	0,97	0,97
Colombia	474	0,98	0,96	0,97
España	335	0,95	0,94	0,94
México	387	0,93	0,97	0,95
Perú	470	0,97	0,91	0,94
Venezuela	433	0,96	0,99	0,97
Total	2.860	0,96	0,95	0,95

Tabla 5.2: Resultados del algoritmo de localización por perfil sobre aquellos usuarios a los que se ha podido asignar una etiqueta de clase.

Hemos obtenido un *accuracy* del 95 %, y el resto de las métricas toma valores similares para el total. Estas métricas varían un poco de un país a otro, pero siempre quedan por encima del 90 %.

5.2 n-gramas y LDR sin Preproceso

Las siguientes figuras muestran los resultados obtenidos tras evaluar el corpus mediante las representaciones basadas en n-gramas y el método LDR. Experimentamos con n-gramas de palabras y caracteres mediante las representaciones basadas en TF y TF-IDF. Hacemos un barrido para n de 1 a 7, empleando los tres clasificadores descritos: Naïve Bayes, Árboles de Decisión y SVM lineales. Experimentamos con distintos tamaños de vocabulario: con 500 términos (Figura 5.1), 1000 términos (Figura 5.2) y 5000 términos (Figura 5.3).

Observando los resultados globalmente, vemos que obtenemos resultados ligeramente mejores cuanto mayor es el tamaño del vocabulario. Por ejemplo, para 500 términos obtenemos el mejor resultado con frecuencias de uni-gramas (TF), un *accuracy* del 87 %. Bajo la misma configuración pero con 5000 términos, este resultado es del 91 %. También vemos que en todos los casos las SVM

lineales son el clasificador que mejor funciona.

En cuanto a n-gramas de caracteres, obtenemos los mejores resultados para $n \geq 4$, aunque a partir de $n = 6$ empieza a empeorar. Esto puede deberse a que esos tamaños de ventana son los suficientemente grandes para captar palabras enteras o sus partes más relevantes, como lexemas y afijos. También llama la atención que con 1000 términos y TF de uni-gramas de caracteres (ver Figura 5.2), somos capaces de diferenciar correctamente poco más del 50 % de los usuarios.

Los mejores resultados con los n-gramas de caracteres siempre salen peores que los mejores con uni-gramas de palabras. En el mejor de los casos (5000 términos, Figura 5.3), los 4-gramas de caracteres casi igualan el resultado de los uni-gramas de palabras. El mejor de los resultados es un 92 % de *accuracy* y viene dado bajo la configuración de uni-gramas de palabras con TF-IDF y SVM. En el caso de las palabras también vemos que para valores de $n > 1$, el resultado empieza a empeorar notablemente.

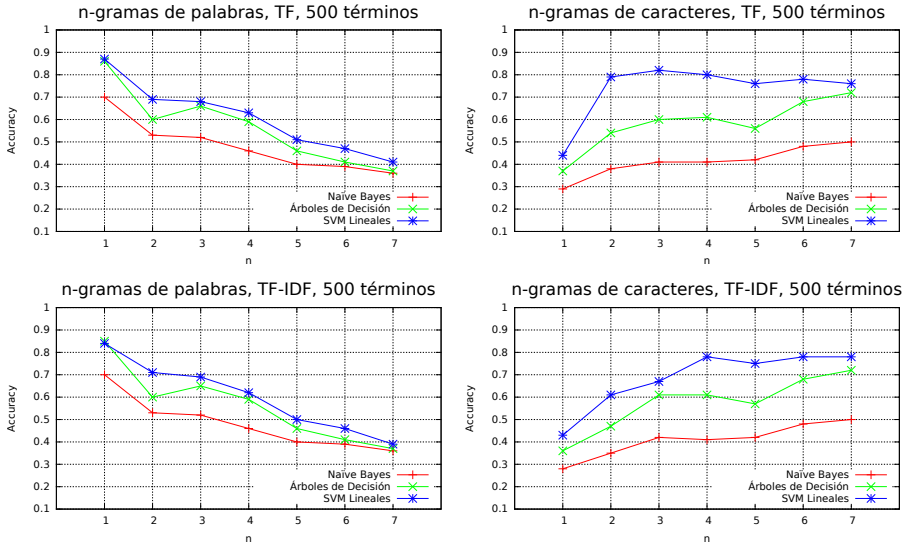


Figura 5.1: *Accuracies* que hemos obtenido para las representaciones TF (arriba) y TF-IDF (abajo) basadas en n-gramas de palabras (izquierda) y de caracteres (derecha). Clasificación mediante Naïve Bayes, Árboles de Decisión y SVM lineales. El tamaño del vocabulario es de 500 términos.

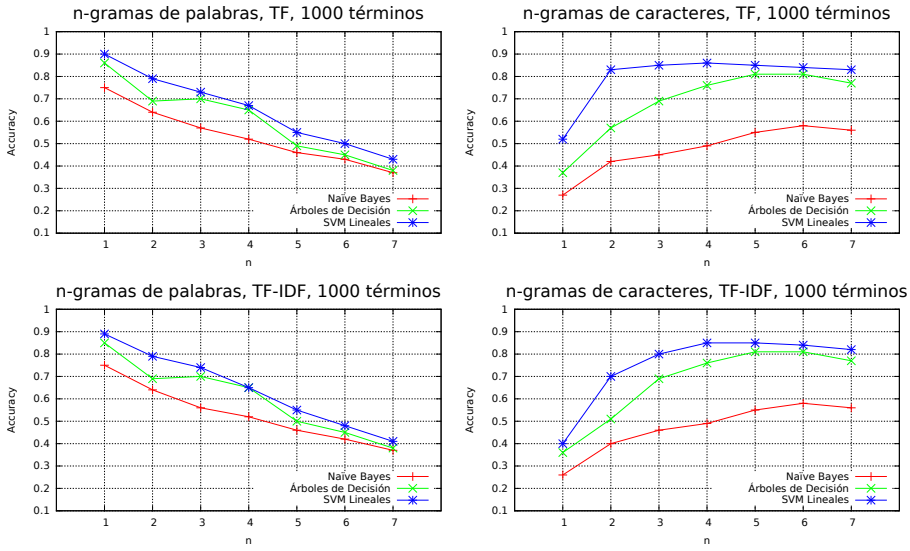


Figura 5.2: *Accuracies* que hemos obtenido para las representaciones TF (arriba) y TF-IDF (abajo) basadas en n-gramas de palabras (izquierda) y de caracteres (derecha). Clasificación mediante Naïve Bayes, Árboles de Decisión y SVM lineales. El tamaño del vocabulario es de 1000 términos.

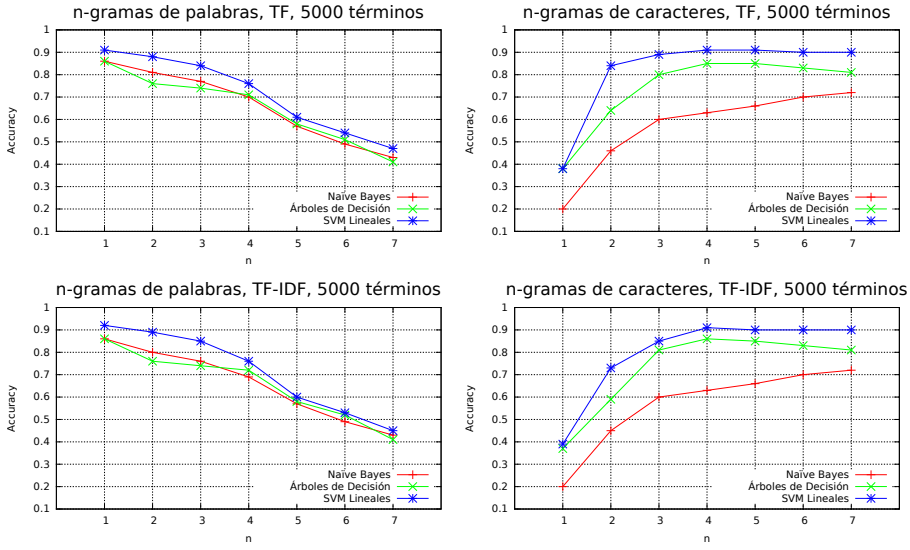


Figura 5.3: *Accuracies* que hemos obtenido para las representaciones TF (arriba) y TF-IDF (abajo) basadas en n-gramas de palabras (izquierda) y de caracteres (derecha). Clasificación mediante Naïve Bayes, Árboles de Decisión y SVM lineales. El tamaño del vocabulario es de 5000 términos.

La configuración que mejor funciona viene dada por los uni-gramas de palabras bajo el esquema TF-IDF, con SVM como clasificador y con un tamaño de vocabulario de 5000 términos. La Tabla 5.3 muestra, además del *accuracy* global obtenido, la *precision*, *recall* y *F-score* en total y por país. Vemos que los valores de estas métricas no difieren mucho entre si. Esto sugiere que el corpus construido está bien equilibrado en cuanto a número de muestras. Para los totales estas métricas presentan el mismo valor (92 %) y para cada país difieren algo más las unas de las otras. Los países que presentan mejores resultados son Chile y México, donde estas medidas oscilan entre el 93-95 %. El país con peores resultados es Perú, donde el *recall* baja hasta el 88 % y la *F-score* hasta el 90 %. Las medidas para el resto de países (Argentina, Colombia, España y Venezuela) toman valores similares al total.

Accuracy: 0.92

País	*Precision*	*Recall*	*F-score*
Argentina	0,92	0,94	0,93
Chile	0,94	0,93	0,94
Colombia	0,93	0,91	0,92
España	0,92	0,92	0,92
México	0,93	0,95	0,94
Perú	0,92	0,88	0,90
Venezuela	0,91	0,93	0,92
Total	0,92	0,92	0,92

Tabla 5.3: Resultados con representación basada en TF-IDF sobre uni-gramas de palabras y SVM como clasificador. Mostramos el *accuracy* global, la *precision*, el *recall* y la *F-score*.

Dado que la mejor configuración es la descrita en la Tabla 5.3, la mantenemos para experimentar con LDR. Es decir, aplicamos este método sobre uni-gramas de palabras con SVM como clasificador y un vocabulario de 5000 términos. Los resultados de este experimento los mostramos en la Tabla 5.4. El método LDR demuestra funcionar mejor para esta configuración, de forma que para el total las cuatro medidas pasan del 92 % al 96 % En cuanto a las medidas a nivel de país, observamos una mejora global en todas ellas. Los países que mejor resultado ofrecen son Chile y México, aunque Argentina, Venezuela, Colombia y España obtienen resultados similares, entorno al 95-97 %. El único país que decae respecto al resto es Perú, con un *recall* del 90 %. Sin embargo, la precisión para Perú es la más elevada de todas con un 98 %.

Accuracy: 0.96

País	*Precision*	*Recall*	*F-score*
Argentina	0,95	0,96	0,96
Chile	0,97	0,96	0,97
Colombia	0,95	0,95	0,95
España	0,95	0,96	0,95
México	0,95	0,98	0,96
Perú	0,98	0,90	0,94
Venezuela	0,94	0,98	0,96
Total	0,96	0,96	0,96

Tabla 5.4: Resultados con representación basada en LDR sobre uni-gramas de palabras y SVM como clasificador. Mostramos el *accuracy* global, la *precision*, el *recall* y la *F-score*.

5.3 Efectos del Preproceso

En esta sección mostramos los efectos resultantes de aplicar las operaciones de preproceso descritas previamente (eliminación de *stopwords*, *stemming* y ambas) aplicada sobre uni-gramas de palabras, para las tres representaciones (TF, TF-IDF y LDR). Los resultados de estos experimentos los mostramos en la Tabla 5.5.

Representación	Preproceso	*Accuracy*	*Precision*	*Recall*	*F-score*
1-gramas de palabras, TF, (SVM)	Raw	0,91	0,92	0,92	0,92
	Stopwords	0,92*	0,92	0,92	0,92
	Stem	0,92*	0,92	0,92	0,92
	Ambas	0,92*	0,92	0,92	0,92
1-gramas de palabras, TF-IDF. (SVM)	Raw	0,92	0,92	0,92	0,92
	Stopwords	0,92	0,92	0,92	0,92
	Stem	0,92	0,92	0,92	0,92
	Ambas	0,92	0,92	0,92	0,92
1-gramas de palabras, LDR, (SVM)	Raw	0,96	0,96	0,96	0,96
	Stopwords	0,96	0,96	0,96	0,96
	Stem	0,96	0,96	0,95	0,95
	Ambas	0,96	0,96	0,96	0,96

Tabla 5.5: Resultados obtenidos con uni-gramas de palabras, aplicando distintas opciones de preproceso sobre cada una de las representaciones. Como clasificador hemos usado SVM Lineales. Marcamos con * aquellas configuraciones que son estadísticamente significativas al 95 % según el test de *t*-student.

Cualquiera de las opciones de preproceso mejora el *accuracy* de 91 % a 92 % para TF. Esta mejora es estadísticamente significativa según el test de t-student, comparándolas con el resultado Raw obtenido. Para el resto de las representaciones el preproceso no muestra ninguna mejora. De hecho, combinando el LDR con *stemming* el *recall* y la *F-score* empeoran un 1 %. Por tanto, decidimos no aplicar ninguna opción de preproceso en el resto de experimentos.

5.4 Comparación con HispaBlogs

En esta sección presentamos una comparativa entre los corpus HispaTweets e HispaBlogs. Hemos empleado la configuración para HispaBlogs que mejores resultados ofrece, descrita en [30]. Esta consiste en uni-gramas de palabras con LDR y con un clasificador multiclase basado en SVM. Los autores emplearon todos los términos disponibles en el conjunto de entrenamiento. En el caso de HispaTweets, la configuración que ofrece mejores resultados viene dada por unigramas de palabras con LDR y SVM para la clasificación, como hemos descrito en las secciones anteriores. En este caso, hemos empleado los 5000 términos más frecuentes.

La Tabla 5.6 muestra los resultados alineados para ambos corpus. Vemos que las métricas toman valores más bajos para el corpus HispaBlogs que para HispaTweets. El *accuracy* para HispaBlogs es de 71 %, mientras que en HispaTweets es del 96 %. La *precision* por país en HispaBlogs se halla entre el 66 % y el 78 %, 71 % de media. En HispaTweets ésta se encuentra entre el 94 % y el 98 %, con una media del 96 %. Los países con precisiones más altas en HispaTweets son Perú y Chile, mientras que en HispaBlogs son México y Perú. Esto sugiere que estas variedades pueden ser más fáciles de diferenciar. En cuanto al *recall*, para HispaTweets toma valores entre 95 % y 98 %, salvo Perú que baja a 90 %. En el caso de HispaBlogs, estos valores oscilan entre el 66 % como mínimo y el 77 % como máximo. El hecho de que México y Venezuela presenten un *recall* más alto en HispaTweets, indica que tienden a confundirse menos con las otras variedades. En el caso de HispaBlogs, los países que presentan un *recall* más alto son España y Chile. Si analizamos la *F-score* para aquellas variedades compartidas por ambos corpus, observamos que los tres países que mejor resultado obtienen son Chile (97 %) y Argentina, México y Venezuela (96 %) en el caso de HispaTweets, y España (74 %) en el caso de HispaBlogs, seguido de cerca por Chile, México y Perú, con una *F-score* del 71 %. La *F-score* global en HispaBlogs es del 71 %.

Accuracy HT: 0.96
Accuracy HB: 0.71

País	*Precision*		*Recall*		*F-score*	
	HT	HB	HT	HB	HT	HB
Argentina	0,95	0,66	0,96	0,72	0,96	0,69
Chile	0,97	0,67	0,96	0,76	0,97	0,71
Colombia	0,95	-	0,95	-	0,95	-
España	0,95	0,71	0,96	0,77	0,95	0,74
México	0,95	0,78	0,98	0,66	0,96	0,71
Perú	0,98	0,77	0,90	0,66	0,94	0,71
Venezuela	0,94	-	0,98	-	0,96	-
Total	0,96	0,71	0,96	0,71	0,96	0,71

Tabla 5.6: Resultados alineados para HispaTweets (HT) e HispaBlogs (HB). En ambos casos, con representación basada en LDR sobre uni-gramas de palabras y SVM como clasificador. Mostramos el *accuracy* global, la *precision*, el *recall* y la *F-score*.

Capítulo 6

Aplicación del Método LDR en la Tarea de Discriminación de Idiomas Similares

En este capítulo describimos nuestra participación en la tarea DSL 2015[1] en el taller LT4VarDial 2015 [28] bajo el nombre de NLEL_UPV_Autoritas [43]. En la sección 6.1 introducimos la tarea y la motivación. Describimos las aproximaciones empleadas en la sección 6.2, para los dos modalidades disponibles de la tarea. A continuación detallamos en la sección 6.3 el corpus proporcionado y los resultados que obtuvimos. Finalmente, en la sección 6.3.5 exponemos nuestras conclusiones.

6.1 Introducción

La tarea de discriminación de idiomas similares o *Discrimination between Similar Languages* (DSL) es muy similar a la de identificación de variedades del idioma (LVI) que hemos abordado en esta tesina. Ambas pueden verse como un caso límite de la identificación del idioma, puesto que comparten una buena parte de la problemática. Por ejemplo, el hecho de que los idiomas similares o variedades a diferenciar compartan una gran parte del léxico. La tarea DSL 2015 [28] en el taller LT4VarDial aborda conjuntamente los problemas de DSL y LVI, puesto que pretende identificar tanto variedades (por ejemplo, del español y del portugués) así como idiomas similares (por ejemplo, bosnio, serbio y croata).

Resulta interesante analizar los sistemas presentados en la edición anterior. Goutte et al. [44] abordaron la tarea con un sistema en dos etapas. En primer

[1] http://ttg.uni-saarland.de/lt4vardial2015/dsl.html

lugar, predijeron el grupo de idiomas al que pertenecía la variedad con un clasificador probabilístico en 6 clases. Finalmente, predijeron la variedad con una combinación por voto de clasificadores discriminativos. Utilizaron n-gramas de caracteres y palabras y reportaron un *accuracy* del 95,71 %. El sistema presentado por Porta y Sancho [45] utilizó un clasificador jerárquico basado en clasificadores de máxima entropía. El primer nivel predijo el grupo de idiomas y el segundo la variedad dentro del grupo. Experimentaron con n-gramas de caracteres y palabras, junto con una lista de palabras que pertenecen exclusivamente a cada variedad del idioma. El *accuracy* reportado fue de 92,6 %. Puver [46] utilizó SVM con n-gramas de caracteres y palabras. Analizó en profundidad como el parámetro coste influenció en los resultados de la clasificación, reportando un *accuracy* global del 95 %. El sistema reportado por King et al. [47] combinó n-gramas de caracteres y palabras con técnicas de selección de características tales como *Information Gain*. Sin embargo, obtuvieron sus mejores resultados (87,32 % de *accuracy*) sin selección de características y con clasificador Naïve Bayes. Lui et al. [48] reportaron sus mejores resultados con una versión jerárquica de su herramienta `langID` [25]. En este trabajo describimos nuestra participación la tarea DSL 2015 [49]. Participamos bajo el nombre NLEL_UPV_Autoritas en las dos modalidades disponibles: abierta y cerrada. Para la modalidad abierta desarrollamos un sistema que funciona en dos etapas. En primer lugar, aplicamos un detector de idioma para identificar los distintos grupos que se corresponden con familias de idiomas, para después distinguir dentro de cada grupo las posibles variedades mediante el método LDR. Para la modalidad cerrada, aplicamos el método LDR para entrenar un clasificador multiclase con todas las variedades del idioma juntas.

6.2 Identificación de Variedades del Idioma

Nuestro objetivo era evaluar el rendimiento de nuestra aproximación cuando dividimos el proceso en dos etapas y cuando realizamos la identificación sobre todas las variedades juntas, con el método LDR.

La Figura 6.1 ilustra el proceso seguido en cada una de las modalidades. Para la modalidad abierta hemos desarrollado un método en dos pasos. El primero consiste en identificar los distintos grupos de idiomas/variedades mediante un detector de idioma. Por ejemplo, las instancias correspondientes al español forman parte del mismo grupo, sin importar la variedad. Los idiomas similares como el bosnio, el serbio y el croata también se agrupan en un único grupo, puesto que son detectados como croata. Para este paso utilizamos el detector `ldig` desarrollado en [50]. El autor estimó sus modelos mediante n-gramas de caracteres aplicado sobre los *abstracts* de la Wikipedia, y utilizó Naïve Bayes como algoritmo de clasificación. Reportó *accuracies* del 99,1 % para hasta 53 idiomas. El segundo paso consiste en generar las características mediante el método LDR de forma local a cada uno de los grupos obtenidos anteriormente. Finalmente, entrenamos un clasificador basado en Redes Bayesianas para cada uno de los grupos.

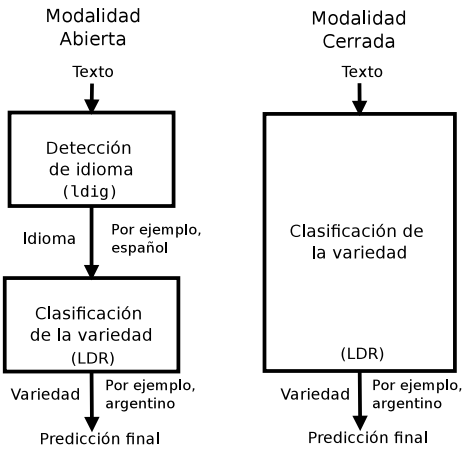


Figura 6.1: Esquema de los sistemas implementados para las modalidades abierta y cerrada.

El proceso al completo funciona como sigue. Para predecir la variedad del idioma de un nuevo texto, primero detectamos el idioma o grupo al que pertenece con el detector `ldig`. Una vez asignado el grupo, generamos las características mediante LDR localmente a ese grupo, y predecimos la variedad con las Redes Bayesianas.

En cuanto a la modalidad cerrada, representamos todas las variedades juntas. Esto implica generar las características con el LDR para todos los idiomas/variedades juntos, y predecir la correcta mediante un único clasificador multiclase. Como método de clasificación, hemos empleado Naïve Bayes debido a cuestiones de rendimiento en la fase de entrenamiento.

6.3 Resultados Experimentales

En esta sección mostramos la evaluación de la metodología propuesta en nuestra participación en la tarea DSL 2015 del taller LT4VarDial. En primer lugar, describimos el corpus y la metodología para la evaluación. Seguidamente describimos nuestros resultados. Finalmente, explicamos nuestra participación en ambas modalidades y las discutimos con la intención establecer una comparativa.

6.3.1 Corpus y Metodología

Hemos usado el corpus DSLCC v.2.0 [51]. Éste contiene frases extraídas de noticias en diferentes idiomas y variedades, tal y como se muestra en la Tabla 6.1. El grupo codificado como *xx* está construido con frases en distintos idiomas "desconocidos", diferentes a los que había que identificar. Entre ellos se encontraban por ejemplo el catalán y el tagalo.

Grupo	Idioma	Código
Eslavos del sureste	Búlgaro	bg
	Macedonio	mk
Español	Argentino	es-AR
	Peninsular	es-ES
Portugués	Brasileño	pt-BR
	Europeo	pt-PT
Eslavos de sureste	Bosnio	bs
	Croata	hr
	Serbio	sr
Austronesio	Indonesio	id
	Malayo	my
Eslavos del oeste	Checo	cz
	Eslovaco	sk
Otros		xx

Tabla 6.1: Idiomas en el corpus DSLCC v.2.0.

La longitud de cada frase comprende entre 20 y 100 *tokens*. Para cada idioma o variedad, este corpus contiene 18.000 instancias para entrenamiento, 2.000 para validación y 1.000 para test. La Tabla 6.2 muestra un resumen del número total de instancias. El corpus está compuesto por dos conjuntos de test, A y B. Ambos contienen las mismas instancias, pero el test B fue procesado con un reconocedor de entidades nombradas o *Named Entities* (NE) para reemplazarlas por un marcador.

Training	Development	Test
252.000	28.000	14.000

Tabla 6.2: Número de instancias por conjunto.

Usamos el conjunto de entrenamiento para aprender las puntuaciones y los correspondientes modelos de clasificación. Probamos nuestros métodos con el conjunto de validación utilizando la GUI de Weka[2] [52]. Hemos construido una aplicación en Java para predecir documentos en conjunto de test utilizando los modelos previamente aprendidos con Weka. En las siguientes secciones explicamos nuestros enfoques para ambas tareas, abierta y cerrada. Presentamos resultados comparativos entre validación, test A y test B. También llevamos a cabo tests de significancia estadística entre ambos conjuntos de test. Utilizamos la siguiente notación para los niveles de confianza: * al 95 % y ** al 99 %.

[2] http://www.cs.waikato.ac.nz/ml/weka/

6.3.2 Modalidad Abierta

Para la modalidad abierta desarrollamos un sistema que consta de dos pasos. En primer lugar, utilizamos el detector de idioma `ldig` para obtener el grupo de idioma. El detector `ldig` fue entrenado con los ficheros xml de los *abstracts* de la Wikipedia. No especificamos ningún grupo de forma explicita. En su lugar, `ldig` detecta idiomas/variedades similares como un único idioma. Aprovechamos este hecho para establecer los grupos de idiomas. El *accuracy* obtenido en este paso para el conjunto de validación lo mostramos en la Tabla 6.3

Idiomas/Variedades	**Grupo de idiomas**	**Accuracy**
bg	bg	99,80
mk	mk	100,00
es-AR, es-ES	es	99,96
pt-BR, pt-PT	pt	99,72
hr, bs, sr	hr	99,73
id, my	id	99,92
cz	cz	99,63
sk	sk	99,65
otros idiomas	xx	99,90
	global	99,81

Tabla 6.3: Accuracies del detector `ldig` en el conjunto de validación.

En este paso, detectamos el búlgaro (*bg*), el checo (*cz*), el macedonio (*mk*) y el eslovaco (*sk*). En cuanto a las otras variedades, estas fueron detectadas como sigue: los idiomas eslavos del suroeste (croata, bosnio y serbio) fueron detectados como croata (*hr*). Los idiomas austronesios (indonesio y malayo) fueron detectados como indonesio (*id*) y las variedades españolas (argentino y peninsular) y portuguesas (europeo y brasileño) fueron clasificados en sus respectivos grupos (*es* y *pt*). Clasificamos como *xx* el resto. Una vez identificado el grupo, aplicamos el método LDR para detectar la variedad correspondiente.

Los resultados para validación, test, y el test sin NE los mostramos en la Tabla 6.4. Los resultados para los grupos con un solo idioma (*bg, mk, cz, sk*) muestran *accuracies* superiores al 99 % para validación y ambos tests. Los resultados para los grupos con más de una variedad son algo más bajos. Esto no se cumple en los idiomas austronesios (*id*) donde obtenemos resultados superiores al 99 % excepto para el indonesio (*id*) en el test sin NE. Los peores resultados los obtenemos para los idiomas eslavos del suroeste (*hr*), donde el clasificador debe discriminar entre tres clases. El test de significancia muestra que nuestro método es bastante robusto contra las NE eliminadas en el caso de los idiomas eslavos del suroeste (*bs*, *hr* y *sr*), malayo (*my*) y español argentino (*es-AR*).

Variedad	Accuracy		
	Validación	**Test A**	**Test B**
bg*	99,80	99,90	99,80
mk*	100,00	99,90	100,00
es-ES	88,00	84,70	79,50
es-AR*	87,50	88,00	87,70
pt-PT	88,60	87,40	94,00
pt-BR	90,10	90,03	68,50
bs*	78,35	78,00	74,40
hr*	86,15	85,80	85,40
sr**	86,40	86,40	82,70
id	99,40	99,40	92,90
my*	99,45	99,20	99,50
cz*	99,70	99,80	99,40
sk*	99,60	99,30	99,60
xx*	99,90	99,90	99,70
global	93,07	92,71	90,22

Tabla 6.4: Accuracies de las identificaciones para la modalidad abierta, para el conjunto de validación, test y test sin NE.

6.3.3 Modalidad Cerrada

Para la modalidad cerrada aplicamos el método LDR sin hacer ninguna separación previa en grupos de idiomas/variedades. Es decir, predecimos directamente la variedad de cada instancia entre las 14 posibles. Los resultados los resumimos en la Tabla 6.5. Podemos ver que los resultados globales para el test B (72,11 %) son mucho menores que para el test A (85,57 %) y para validación (86,08 %). En esta linea, los resultados para la mayoría de idiomas son significativamente diferentes, excepto para el argentino (*es-AR*), macedonio (*mk*) y el grupo otros (*xx*). Esto puede deberse a las probabilidades de los términos correspondientes a NE, que pueden causar confusión entre algunas variedades.

Variedad	Accuracy		
	Validación	**Test A**	**Test B**
bg	98,15	97,50	95,10
mk*	98,95	98,20	98,20
es-ES	87,55	84,80	48,70
es-AR**	67,05	70,00	74,10
pt-PT	82,15	81,20	58,30
pt-BR	72,45	72,50	65,90
bs	55,70	54,30	86,20
hr	80,85	78,88	13,10
sr	74,40	74,70	7,80
id	97,75	97,60	92,00
my	94,25	93,60	97,60
cz	98,45	98,40	94,40
sk	98,80	97,60	79,30
xx*	98,55	98,50	98,80
global	86,08	85,57	72,11

Tabla 6.5: Accuracies obtenidas en la modalidad cerrada, para validación, test y test sin NE.

6.3.4 Comparación entre Métodos

En la Tabla 6.6 mostramos los resultados comparativos entre las modalidades abierta y cerrada en el conjunto de validación. Merece la pena destacar que ambas aproximaciones obtienen resultados más bajos con los mismos grupos (*es*, *pt* y *hr*). En cuanto a los grupos con un solo idioma (*bg*, *mk*, *cz*, *sk*), ambas aproximaciones obtienen *accuracies* superiores al 95 %. El test de significancia indica que existen diferencias significativas entre ambas modalidades, para todos los sistemas. Por tanto, podemos concluir que el método en dos pasos para la modalidad abierta ofreció mejor resultado que tratar con todas las variedades juntas*.

* Encontramos un error en el código que disminuyó el *accuracy* notablemente. Lo corregimos tras la evaluación de la tarea y la publicación de este *ranking*. Los resultados y análisis mostrados en este capítulo corresponden a la versión corregida del sistema. El sistema para la modalidad cerrada ha quedado en la última posición de los siete sistemas que se presentaron. Para la modalidad abierta, el sistema ha alcanzado la segunda posición de tres sistemas presentados.

Grupo	Accuracy	
	Abierta	**Cerrada**
bg	99,80	98,15
mk	100,0	98,95
es	87,75	77,30
pt	89,35	77,30
hr	83,63	70,32
id	99,43	96,0
cz	99,70	98,45
sk	99,60	98,80
xx	99,90	98,55
global	93,07	86,08

Tabla 6.6: Accuracies en la identificación para las modalidades abierta y cerrada, en el conjunto de validación.

6.3.5 Conclusiones

En este capítulo hemos expuesto la participación del equipo NLEL_UPV_Autoritas en la tarea DSL 2015 en el taller LT4VarDial [43]. Participamos en ambas modalidades, abierta y cerrada, y para ambos conjuntos de test, normal y sin NE. Para la modalidad abierta, hemos desarrollado un sistema que funciona en dos pasos: en el primer paso detectamos el grupo del idioma y después la variedad específica. Para la modalidad cerrada, abordamos la tarea como un problema de clasificación multiclase con todas las variedades juntas.

Podemos concluir que abordar la tarea en dos pasos permite obtener mejores resultados que la identificación de todas las variedades juntas. Otros equipos enfocaron la tarea del año anterior con sistemas de clasificación en dos pasos, obteniendo buenos resultados. En esta línea, Goutte et al. [44] obtuvieron el *accuracy* más elevado (95,71 %) prediciendo primero el grupo del idioma con un clasificador probabilístico generativo, y después la variedad en ese grupo con una combinación de clasificadores por voto. En cuanto a las variedades, la predicción más difícil ha sido para los grupos de idiomas eslavos del suroeste, seguidos del español y el portugués. El grupo austronesio fue identificado con bastante éxito en ambas aproximaciones. Los grupos compuestos de un solo idioma obtuvieron *accuracies* elevados en ambas modalidades. Como trabajo futuro planeamos abordar la tarea con nuestro propio detector de idioma basado en LDR. Además, nos gustaría investigar cómo mejorar el *accuracy* en idiomas muy similares, como los idiomas eslavos del suroeste, español o portugués, y tratar de mejorar cuando no se dispone de entidades nombradas.

Capítulo 7

Conclusiones y Trabajo Futuro

En este trabajo hemos abordado el problema de la geolocalización de los usuarios de los medios sociales, con el fin de investigar posibles mejoras en la determinación de su localización geográfica. Hemos tratado de abordar este problema mediante la identificación de la variedad del idioma que emplean. Para ello hemos construido y liberado el corpus HispaTweets[1] con la herramienta que hemos desarrollado[2]. Esta permite descargar aquellos tweets que respondan a las claves de búsqueda especificadas y para las ubicaciones dadas. Con esta herramienta hemos descargado tweets de las ciudades más pobladas de siete países de habla hispana: Argentina, Chile, Colombia, España, México, Perú y Venezuela. Con los autores de los tweets obtenidos hemos elaborado un listado de usuarios categorizado por geografía, y a continuación hemos descargado los últimos 1.000 tweets de las *timelines* de dichos usuarios. Con el objetivo de obtener un corpus que fuera representativo de las variedades y tan equilibrado como fuera posible, hemos filtrado los tweets y usuarios obtenidos atendiendo a tres criterios: i) geográfico: para prescindir de los usuarios obtenidos en las ciudades frontera; ii) temporal: para que los tweets del corpus final estén acotados entre dos fechas y iii) por frecuencia: para garantizar la misma cantidad de información para cada país y que ésta se distribuya igual entre el número de usuarios. El resultado final de este proceso ha sido HispaTweets, un corpus equilibrado con 4.550 usuarios de Twitter (650 para cada uno de los países mencionados) y hasta los últimos 1.000 tweets cada uno de los usuarios, en total 3.938.090 tweets, todos ellos emitidos entre el 1 de enero de 2013 y el 1 de enero de 2016.

En primer lugar hemos evaluado el corpus mediante un algoritmo de localización por perfil. Este algoritmo utiliza la información del perfil del usuario (campo *location*) para predecir su ubicación, contrastando el texto de ese campo con el nombre de los países y las ciudades donde hemos realizado las búsquedas.

[1] https://github.com/autoritas/RD-Lab/tree/master/data/HispaTweets
[2] https://github.com/autoritas/RD-Lab/tree/master/src

El objetivo de este algoritmo era comparar los resultados que podíamos obtener únicamente con la información del perfil frente a los resultados obtenidos mediante el análisis del contenido de los tweets. Además, nos ha servido de referencia para comparar sus resultados con técnicas para LVI. Este algoritmo ha sido capaz de identificar correctamente el 60,02 % de los usuarios en total. Su mayor problema es el gran número de usuarios sin un contenido válido para el campo *location*, bien porque está vacío o bien porque no contiene una ubicación válida. Sin embargo, con aquellos usuarios que sí ha sido capaz de etiquetar obtiene resultados razonablemente buenos, con un *accuracy* del 95 %. Tras evaluar el corpus con el algoritmo de localización por perfil, lo evaluamos mediante técnicas de aprendizaje automático. Probamos distintas características actuales del estado del arte basadas en n-gramas de caracteres y n-gramas de palabras, como TF y TF-IDF. También aplicamos un método novedoso LDR para representar los documentos (usuarios) en un espacio de baja dimensionalidad. En nuestros experimentos hemos probado con valores de n de 1 a 7 y hemos considerado distintos tamaños de vocabulario, con 500, 1000 y 5000 términos. Como clasificadores hemos empleado Naïve Bayes, Árboles de Decisión y Máquinas de Vectores de Soporte lineales. Además, también hemos probado con distintas opciones de preproceso: eliminación de *stopwords*, *stemming* y la combinación de ambas. Todos nuestros experimentos con HispaTweets los llevamos a cabo bajo un esquema de validación cruzada en cinco bloques, y manteniendo siempre separados los usuarios de entrenamiento y validación durante todas las etapas, incluyendo el preproceso.

La configuración que mejores resultados ofrece es uni-gramas de palabras con el método LDR, sin preproceso y con SVM lineales como clasificador. Esta configuración da lugar a un *accuracy* del 96 %, mientras que empleando TF y TF-IDF sobre uni-gramas de palabras hemos obtenido *accuracies* del 91 % y 92 %, respectivamente. Para el resto de valores de n en los n-gramas de palabras obtenemos resultados más bajos. Los n-gramas de caracteres ofrecen peores resultados en general, solamente los 4-gramas con TF y SVM funcionan casi tan bien como con palabras, con un 91 % de *accuracy*. Esto lo atribuimos al hecho de que con valores de n iguales o superiores a 4, el sistema capta palabras cortas o sus partes más importantes, como los lexemas y los afijos. Este tipo de características suelen ser suficientes en tareas de identificación del idioma, porque entre idiomas distintos las diferencias morfológicas son más discriminativas. Por ejemplo, en los textos en inglés puede aparecer muchas veces el sufijo "*ing*" correspondiente al gerundio, mientras que en español aparecerá frecuentemente "*ndo*" para esta misma conjugación (como en 'andando' o 'corriendo'). Sin embargo, este tipo de características resulta insuficiente en LVI porque todas las variedades del español comparten estas características. Por ello, las características léxicas (palabras) ofrecen mejores resultados que las morfológicas (n-gramas de caracteres). No hemos encontrado verdaderamente útil ninguna de las opciones de preproceso. Solamente mejoran en un 1 % los resultados sobre uni-gramas de palabras con TF, siendo una mejora significativa al 95 % según el test estadístico aplicado. En el caso del TF-IDF y LDR, las opciones de preproceso no varían los resultados prácticamente.

También hemos evaluado el corpus HispaBlogs bajo la configuración que mejores resultados ha ofrecido [29, 30], consistente en la generación de características basadas en uni-gramas de palabras con LDR, y un clasificador multiclase basado en SVM para la clasificación. En HispaBlogs obtenemos un *accuracy* del 71 %, muy inferior al 96 % obtenido en HispaTweets. El principal motivo que podría explicar esta diferencia es que con HispaTweets disponemos de mucha más cantidad de información por usuario, lo cual podría facilitar su identificación a pesar de el registro sea más espontáneo.

Las variedades que mejor *precision* ofrecen en HispaTweets son Perú y Chile, y en el caso de HispaBlogs México y Perú. En ambos corpus Perú es uno de los países con la *precision* más elevada, lo cual indica que el clasificador no suele equivocarse al detectar la variedad peruana. En HispaTweets hemos obtenido el *recall* más alto para México, seguidos de España, Chile y Argentina. En el caso de HispaBlogs, los dos países con mayor *recall* han sido España y Chile. En ambos corpus obtenemos un *recall* bastante alto para España y Chile, lo que puede sugerir que estas variedades tienden a confundirse menos con las otras. Si analizamos la *F-score* para aquellas variedades compartidas por ambos corpus, en el caso de HispaTweets están muy igualadas, siendo Chile siendo la que mejor *F-score* ha obtenido con un 97 %, seguido de Argentina y México con un 96 %. En el caso de HispaBlogs, la *F-score* más alta la hemos conseguido con España, con un 74 %. Con Chile, México y Perú hemos obtenido una *F-score* del 71 %.

También hemos estudiado el problema de la discriminación de idiomas similares o DSL, que comparte la problemática de LVI, para lo cual hemos participado en la tarea compartida DSL 2015. Hemos abordado la tarea [43] con el método LDR y con dos aproximaciones distintas: i) En dos pasos: primero la detección del idioma o grupo de idiomas similares, y segundo la detección de la variedad dentro del idioma detectado; ii) En un paso: predicción directa de la variedad entre todas las posibles, mediante un clasificador multiclase. Concluimos que el enfoque en dos pasos, con el que obtenemos un *accuracy* global del 93.07 %, funciona mejor. Observamos que para el grupo compuesto por el bosnio, el serbio y el croata obtenemos un *accuracy* medio del 83.63 %, siendo así los idiomas que más cuestan de discriminar entre si. Las variedades del portugués (de Portugal y Brasil) también presentan dificultades, puesto que con ellas obtenemos un *accuracy* del 89.35 %. Estos resultados son similares a los del grupo de variedades del español (argentino y peninsular), que ofrecen un *accuracy* del 87.75 %. Para el resto de idiomas o variedades, las *accuracies* que obtenemos son superiores al 99 %.

Uno de los trabajos de investigación futuros que proponemos consiste en utilizar una cantidad menor de información por usuario y observar como varían los resultados. Podríamos por ejemplo identificar tweets en lugar de usuarios, siempre y cuando los tweets empleados en el entrenamiento y en el test pertenezcan a distintos usuarios. Esto sería necesario para seguir estudiando el problema desde la perspectiva del *Author Profiling*. También podemos limitar la cantidad de tweets por usuario en el entrenamiento para determinar como varían los resultados en función de la cantidad de información disponible. Este estudio es especialmente conveniente porque la recolección de tweets es muy costosa en

tiempo de descarga. Si con menos información pudiéramos obtener resultados similares, sería más fácil que surgiesen aplicaciones prácticas que pudieran sacar provecho a estas tecnologías.

Otra futura tarea consistiría en ampliar el corpus, incluyendo nuevos idiomas y variedades. La herramienta desarrollada permite definir las ubicaciones de las ciudades más pobladas de los países que escojamos, y descargar tweets que desde allí se emitan. Con ello podríamos, por ejemplo, recopilar tweets y usuarios de Portugal y Brasil para la identificación del portugués, o de Estados Unidos y Reino Unido para la identificación del inglés, entre otros. Las condiciones necesarias para poder llevar a cabo estas ampliaciones son, por un lado, que las distintas variedades de un idioma estén separadas geográficamente, y por el otro que consigamos recuperar suficiente volumen de usuarios y tweets de cada variedad.

Apéndice A

Análisis de la Construcción de HispaTweets

En este apéndice profundizamos en algunas de las estadísticas expuestas en el Capítulo 3, relativas a la construcción del corpus. Desglosamos algunas de las tablas de ese capítulo, mostrando los resultados a nivel de ciudad. La sección A.1 expone en detalle el proceso de la recolección geográfica de tweets, y los resultados de la misma. La sección A.2 profundiza en las estadísticas relativas al corpus tras la descarga de las *timelines*.

A.1 Recolección Geográfica

En esta sección explicamos en detalle algunos aspectos de la recolección geográfica. En la sección A.1.1 explicamos la configuración del sistema en relación a las ubicaciones. Los resultados de la búsqueda los mostramos en la sección A.1.2. Finalmente, en la sección A.1.3 analizamos los resultados de la búsqueda en cuanto a las palabras clave empleadas.

A.1.1 Configuración de la Búsqueda Geolocalizada

Como explicamos en la sección 3.2, nuestro sistema obtiene las ubicaciones de las ciudades a partir de la Wikipedia. También hemos explicado que hay algunas ciudades que no hemos podido añadir, debido a que su artículo no contenía las coordenadas geográficas. La Tabla A.1 muestra aquellas ciudades que no hemos podido añadir.

País	Ciudades ignoradas
Chile	La Serena-Coquimbo, Gran Iquique, Rancagua, Puerto Montt-Puerto Varas, Chillán, Quillota, San Antonio, Colina, Linares, Peñaflor, Lampa, Buin, San Felipe, San Carlos, Padre Hurtado
Colombia	Santa Marta
México	Saltillo, Atizapán de Zaragoza
Venezuela	Valencia, Santa Bárbara del Zulia

Tabla A.1: Listado de ciudades ignoradas para cada país.

La Tabla A.2 muestra la dirección de aquellos anexos en la Wikipedia de los cuales extraemos las URLs de las ciudades que hemos utilizado para la búsqueda, tal y como explicamos en el Capítulo 3.

País	URL al anexo de la Wikipedia
Argentina	https://es.wikipedia.org/wiki/Anexo:Ciudades_de_Argentina_por_poblaci%C3%B3n
Chile	https://es.wikipedia.org/wiki/Anexo:Ciudades_de_Chile
Colombia	https://es.wikipedia.org/wiki/Anexo:Ciudades_de_Colombia_por_poblaci%C3%B3n
España	https://es.wikipedia.org/wiki/Anexo:Municipios_de_Espa%C3%B1a_por_poblaci%C3%B3n
México	https://es.wikipedia.org/wiki/Anexo:Ciudades_de_M%C3%A9xico_m%C3%A1s_pobladas
Perú	https://es.wikipedia.org/wiki/Anexo:Ciudades_del_Per%C3%BA_por_poblaci%C3%B3n
Venezuela	https://es.wikipedia.org/wiki/Anexo:Ciudades_de_Venezuela

Tabla A.2: Enlaces a los anexos de la Wikipedia que contienen las URLs con los artículos de las ciudades.

Las Tablas A.3–A.9 muestran cada una de las ciudades, ubicaciones y radios empleados para aquellos países en los que llevamos a cabo la recolección de tweets por geografía. Las coordenadas de las ciudades son las que hemos extraído de la Wikipedia. El radio ha sido decidido de forma arbitraria. Consideramos que 8,5 km es un radio bastante grande como para captar una buena cantidad de tweets de las ciudades, evitando en lo posible invadir las ciudades colindantes. Si alguna de estas configuraciones se sale de su país correspondiente, esto no tendrá efecto en el resultado final puesto que estas ciudades son eliminadas manualmente con un filtrado geográfico (ver sección 3.4.1)

Argentina (15 ciudades)			
Ciudad	Latitud	Longitud	Radio
Buenos Aires	-34.599722222222	-58.381944444444	8,5
Córdoba	-31.416666666667	-64.183333333333	8,5
Rosario	-32.95	-60.65	8,5
Mendoza	-32.883333333333	-68.816666666667	8,5
La Plata	-34.933333333333	-57.95	8,5
Mar del Plata	-38.01667	-57.55	8,5
San Miguel de Tucumán	-26.816666666667	-65.216666666667	8,5
Salta	-24.788333333333	-65.410555555556	8,5
Santa Fe	-31.633333333333	-60.7	8,5
Corrientes	-27.483333333333	-58.816666666667	8,5
Bahía Blanca	-38.716666666667	-62.266666666667	8,5
Resistencia	-27.451388888889	-58.986666666667	8,5
Vicente López	-34.516666666667	-58.483333333333	8,5
Posadas	-27.366666666667	-55.896944444444	8,5
San Juan	-31.5375	-68.536388888889	8,5

Tabla A.3: Listado de las ubicaciones utilizadas para la recolección de tweets en Argentina: ciudad, latitud, longitud y radio.

Chile (25 ciudades)			
Ciudad	Latitud	Longitud	Radio
Santiago de Chile	-33.45	-70.666666666667	8,5
Gran Concepción	-36.783333333333	-73.116666666667	8,5
Gran Valparaíso	-33.04611111	-71.62222222	8,5
Gran Temuco	-38.75	-72.66666667	8,5
Antofagasta	-23.633333333333	-70.4	8,5
Talca	-35.433333333333	-71.666666666667	8,5
Arica	-18.475	-70.314444444444	8,5
Los Ángeles	-37.466666666667	-72.35	8,5
Copiapó	-27.366666666667	-70.316666666667	8,5
Valdivia	-39.8	-73.233333333333	8,5
Osorno	-40.566666666667	-73.15	8,5
Curicó	-34.983333333333	-71.233333333333	8,5
Calama	-22.466666666667	-68.916666666667	8,5
Punta Arenas	-53.15	-70.916666666667	8,5
Melipilla	-33.7	-71.216666666667	8,5
Ovalle	-30.583333333333	-71.2	8,5
San Fernando	-34.583333333333	-70.966666666667	8,5
Paine	-33.816666666667	-70.75	8,5
Talagante	-33.666666666667	-70.933333333333	8,5
Los Andes	-32.816666666667	-70.616666666667	8,5
Coyhaique	-45.566666666667	-72.066666666667	8,5

Rengo	-34.416666666667	-70.866666666667	8,5
Vallenar	-28.566666666667	-70.75	8,5
Villarrica	-39.266666666667	-72.216666666667	8,5
Angol	-37.8	-72.716666666667	8,5

Tabla A.4: Listado de las ubicaciones utilizadas para la recolección de tweets en Chile: ciudad, latitud, longitud y radio.

Colombia (49 ciudades)			
Ciudad	Latitud	Longitud	Radio
Bogotá	4.5988888888889	-74.080833333333	8,5
Medellín	6.2447472222222	-75.574827777778	8,5
Cali	3.44	-76.519722222222	8,5
Barranquilla	10.963888888889	-74.796388888889	8,5
Cartagena de Indias	10.423611111111	-75.525277777778	8,5
Cúcuta	7.9075	-72.504722222222	8,5
Soledad	10.909722222222	-74.785833333333	8,5
Ibagué	4.4377777777778	-75.200555555556	8,5
Bucaramanga	7.1186111111111	-73.116111111111	8,5
Soacha	4.5780555555556	-74.214444444444	8,5
Villavicencio	4.1425	-73.629444444444	8,5
Pereira	4.8142777777778	-75.694558333333	8,5
Bello	6.3319444444444	-75.558055555556	8,5
Valledupar	10.460277777778	-73.259722222222	8,5
Montería	8.7477777777778	-75.881388888889	8,5
San Juan de Pasto	1.21	-77.274722222222	8,5
Buenaventura	3.8772222222222	-77.026666666667	8,5
Manizales	5.0661111111111	-75.484722222222	8,5
Neiva	2.9275	-75.2875	8,5
Palmira	3.5347222222222	-76.295555555556	8,5
Armenia	4.5388888888889	-75.6725	8,5
Popayán	2.4591666666667	-76.600277777778	8,5
Sincelejo	9.2994444444444	-75.395833333333	8,5
Itaguí	6.1726	-75.6096	8,5
Riohacha	11.544166666667	-72.906944444444	8,5
Floridablanca	7.0697222222222	-73.097777777778	8,5
Envigado	6.1719444444444	-75.580277777778	8,5
Tuluá	4.0847222222222	-76.198611111111	8,5
San Andrés de Tumaco	1.8066666666667	-78.764722222222	8,5
Dosquebradas	4.8361111111111	-75.676111111111	8,5
Tunja	5.5402777777778	-73.361388888889	8,5
Barrancabermeja	7.0675	-73.847222222222	8,5
San Juan de Girón	7.0730555555556	-73.168055555556	8,5
Apartadó	7.8847222222222	-76635	8,5
Uribia	11.713888888889	-72.265833333333	8,5

Florencia	1.6141666666667	-75.611666666667	8,5
Turbo	8.0930555555556	-76.728333333333	8,5
Maicao	11.377777777778	-72.241388888889	8,5
Piedecuesta	6.9886111111111	-73.050277777778	8,5
Yopal	5.3305555555556	-72.390555555556	8,5
Ipiales	0.82888888888889	-77.640555555556	8,5
Fusagasugá	4.3372222222222	-74.364444444444	8,5
Facatativá	4.8147222222222	-74.355277777778	8,5
Cartago	4.7469444444444	-75.911944444444	8,5
Chía	4.8633333333333	-74.052777777778	8,5
Pitalito	1.8538888888889	-76.051388888889	8,5
Zipaquirá	5.0247222222222	-74.001388888889	8,5
Magangué	9.2466666666667	-74.759444444444	8,5
Malambo	10.860277777778	-74.778888888889	8,5

Tabla A.5: Listado de las ubicaciones utilizadas para la recolección de tweets en Colombia: ciudad, latitud, longitud y radio.

España (15 ciudades)			
Ciudad	Latitud	Longitud	Radio
Madrid	40.418888888889	-3.6919444444444	8,5
Barcelona	41.3825	2.1769444444444	8,5
Valencia	39.466666666667	-375	8,5
Sevilla	37.383333333333	-5.9833333333333	8,5
Zaragoza	41.65	-0.88333333333333	8,5
Málaga	36.716666666667	-4.4166666666667	8,5
Murcia	37.986111111111	-1.1302777777778	8,5
Mallorca	39.566666666667	2.6497222222222	8,5
Gran Canaria	28.127222222222	-15.431388888889	8,5
Bilbao	43.262222222222	-2.9533333333333	8,5
Alicante	38.345277777778	-0.48305555555556	8,5
Córdoba	37.883333333333	-4.7666666666667	8,5
Valladolid	41.651980555556	-4.7285611111111	8,5
Vigo	42.233333333333	-8.7166666666667	8,5
Gijón	43.533333333333	-5.7	8,5

Tabla A.6: Listado de las ubicaciones utilizadas para la recolección de tweets en España: ciudad, latitud, longitud y radio.

México (15 ciudades)			
Ciudad	Latitud	Longitud	Radio
México D. F.	19.419444444444	-99.145555555556	8,5
Ecatepec	19.6	-99.05	8,5
Guadalajara	20.666111111111	-103.35194444444	8,5
Puebla de Zaragoza	19.051388888889	-98.217777777778	8,5
Juárez	31.739444444444	-106.48694444444	8,5
Tijuana	32.530833333333	-117.02	8,5
León	21.119722222222	-101.68055555556	8,5
Zapopan	20.720277777778	-103.39194444444	8,5
Monterrey	25.671388888889	-100.30861111111	8,5
Nezahualcóyotl	19.412777777778	-99.041944444444	8,5
Chihuahua	28.635277777778	-106.08888888889	8,5
Naucalpan de Juárez	19.475277777778	-99.237777777778	8,5
Mérida	20.97	-89.62	8,5
San Luis Potosí	22.149722222222	-100975	8,5
Aguascalientes	21.880833333333	-102.29611111111	8,5

Tabla A.7: Listado de las ubicaciones utilizadas para la recolección de tweets en México: ciudad, latitud, longitud y radio.

Perú (51 ciudades)			
Ciudad	Latitud	Longitud	Radio
Lima	-12035	-77.018611111111	8,5
Arequipa	-16.398763888889	-71.536883333333	8,5
Trujillo	-8.1119444444444	-79.028888888889	8,5
Chiclayo	-6.7629611111111	-79.836613888889	8,5
Lambayeque	-6.7	-79.9	8,5
Iquitos	-3.7333333333333	-73.25	8,5
Piura	-5.2008333333333	-80.625277777778	8,5
Cusco	-13.518333	-71.978056	8,5
Chimbote	-9.0745444444444	-78.593572222222	8,5
Huancayo	-12.066666666667	-75.216666666667	8,5
Tacna	-18.055555555556	-70.248333333333	8,5
Juliaca	-15.490833333333	-70.126944444444	8,5
Ica	-14.066666666667	-75.733333333333	8,5
Cajamarca	-7.1644444444444	-78.510555555556	8,5
Pucallpa	-8.3833333333333	-74.55	8,5
Sullana	-4.9	-80.7	8,5
Ayacucho	-13.163055555556	-74.224444444444	8,5
Chincha Alta	-13.45	-76.133333333333	8,5
Huánuco	-9.9294444444444	-76.239722222222	8,5
Huacho	-11.1	-77.6	8,5
Tarapoto	-6.4833333333333	-76.366666666667	8,5

Puno	-15.843333333333	-70.023611111111	8,5
Paita	-5.0666666666667	-81.1	8,5
Huaraz	-9.5333333333333	-77.533333333333	8,5
Tumbes	-3.5666666666667	-80.45	8,5
Pisco	-13.709980555556	-76.203205555556	8,5
Huaral	-11.5	-77.216666666667	8,5
Moyobamba	-6.0333333333333	-76.966666666667	8,5
San Vicente de Cañete	-13.083333333333	-76.4	8,5
Puerto Maldonado	-12.6	-69.183333333333	8,5
Moquegua	-17.2	-70.933333	8,5
Cerro de Pasco	-10.686388888889	-76.2625	8,5
Barranca	-10.753888888889	-77.760994444444	8,5
Andahuaylas	-13.6575	-73.38333333	8,5
Yurimaguas	-5.9	-76.083333333333	8,5
Chancay	-11.568577777778	-77.269655555556	8,5
Talara	-4.5833333333333	-81.266666666667	8,5
Ilo	-17.648611111111	-71.330555555556	8,5
Abancay	-13.633333333333	-72.883333333333	8,5
Chulucanas	-5.0925	-80.1625	8,5
Tingo María	-9.2952777777778	-75.9975	8,5
Sicuani	-14.2700396	-71.231575	8,5
Mala	-12.6575	-76.62916667	8,5
Huancavelica	-12785	-74.971388888889	8,5
Ferreñafe	-6.6392169	-79.7879824	8,5
Chepén	-7.22713056	-79.42983611	8,5
Sechura	-5.55758333	-80.8223	8,5
Pacasmayo	-7.40027778	-79.57	8,5
Tarma	-11.4167	-75.6833	8,5
Bagua Grande	-5.75722222	-78.44527778	8,5
Guadalupe	-7.24686667	-79.47310278	8,5

Tabla A.8: Listado de las ubicaciones utilizadas para la recolección de tweets en Perú: ciudad, latitud, longitud y radio.

Venezuela (15 ciudades)			
Ciudad	Latitud	Longitud	Radio
Caracas	10.5	-66.933333333333	8,5
Maracaibo	10.633333333333	-71.633333333333	8,5
Barquisimeto	10.066666666667	-69.333333333333	8,5
Maracay	10.246944444444	-67.595833333333	8,5
Guayana	8.3596166666667	-62.651652777778	8,5
San Cristóbal	7.75	-72.216666666667	8,5
Barcelona	10.133333333333	-64.683333333333	8,5
Maturín	9.7136027777778	-63.187566666667	8,5
Bolívar	8.1166666666667	-63.55	8,5

Cumaná	10.45	-64.166666666667	8,5
Barinas	8.6333333333333	-70.216666666667	8,5
Cabimas	10.4	-71.433333333333	8,5
Punto Fijo	11.71666667	-70.18333333	8,5
Puerto la Cruz	10.216666666667	-64.616666666667	8,5
Guarenas	10.473888888889	-66.538333333333	8,5

Tabla A.9: Listado de las ubicaciones utilizadas para la recolección de tweets en Venezuela: ciudad, latitud, longitud y radio.

A.1.2 Resultados de la Búsqueda: Análisis Cuantitativo

Las Tablas A.10–A.16 muestran los resultados obtenidos tras la búsqueda geolocalizada a nivel de ciudad y para cada uno de los siete países, profundizando en los resultados de la sección 3.2. Más concretamente mostramos el número de usuarios obtenidos, el número de tweets, la población y la proporción entre la cantidad de usuarios recuperados y la población. Para cada país, sus ciudades están ordenadas descendentemente según su población. Como cabe esperar, aquellas ciudades para las que recuperamos más usuarios y tweets son las capitales. Vemos que la ciudad para la que más tweets hemos recuperado es Lima, seguida de México D.F. Sin embargo, recuperamos más usuarios de México D.F. que de Lima. Para las ciudades con menos de 500.000 habitantes obtenemos pocos usuarios en comparación con las capitales. En general recuperamos entre uno y dos tweets por usuario aproximadamente.

La proporción entre usuarios y población es mayor en las capitales, como cabe esperar. A nivel de país, Perú presenta las más baja de estas proporciones con un 0,0004 %, seguido de Colombia con un 0,0006 %. En España y Argentina esta penetración es mayor, del 0,01809 % y 0.02282 % respectivamente.

Argentina (15 ciudades)					
Ciudad	Tweets (búsqueda)	Usuarios	Tweets/ usuario	Población	Usuarios/ población (%)
Buenos Aires	1.393	830	1,68	2.890.151	0,02872
Córdoba	434	300	1,45	1.329.604	0,02256
Rosario	273	162	1,69	948.312	0,01708
Mendoza	142	89	1,6	937.154	0,00950
La Plata	319	183	1,74	740.369	0,02472
Mar del Plata	249	111	2,24	618.989	0,01793
San Miguel de Tucumán	138	67	2,06	549.163	0,01220
Salta	100	30	3,33	535.303	0,00560
Santa Fe	184	128	1,44	391.231	0,03272
Corrientes	69	37	1,86	314.546	0,01176

Bahía Blanca	139	78	1,78	299.101	0,02608
Resistencia	108	61	1,77	290.723	0,02098
Partido de Vicente López	288	185	1,56	274.082	0,06750
Posadas	153	85	1,8	252.981	0,03360
San Juan	62	47	1,32	112.778	0,04167
Total	4.051	2.393	1,69	10.484.487	0,02282

Tabla A.10: Resultados de la búsqueda geolocalizada en Argentina. Muestra para cada ciudad que ha generado resultados el número de tweets, usuarios, número de tweets por usuario, población y el porcentaje del número de usuarios frente a la población.

Chile (25 ciudades)					
Ciudad	Tweets (búsqueda)	Usuarios	Tweets/ usuario	Población	Usuarios/ población (%)
Santiago de Chile	1.760	972	1,81	6.158.080	0,01578
Gran Concepción	90	31	2,9	1.083.043	0,00286
Gran Valparaíso	161	117	1,38	1.066.893	0,01097
Gran Temuco	20	19	1,05	410.520	0,00463
Antofagasta	78	42	1,86	380.685	0,01103
Talca	77	41	1,88	253.743	0,01616
Arica	34	23	1,48	210.936	0,01090
Los Ángeles	10	8	1,25	187.494	0,00427
Copiapó	11	10	1,1	158.261	0,00632
Valdivia	28	25	1,12	154.445	0,01619
Osorno	19	11	1,73	153.797	0,00715
Curicó	41	24	1,71	141.017	0,01702
Calama	12	9	1,33	138.722	0,00649
Punta Arenas	31	22	1,41	130.704	0,01683
Melipilla	8	7	1,14	110.871	0,00631
Ovalle	5	4	1,25	105.252	0,00380
San Fernando	2	2	1	73.727	0,00271
Paine	0	0	-	66.238	-
Talagante	3	2	1,5	65.020	0,00308
Los Andes	6	6	1	63.055	0,00952

Coyhaique	2	2	1	58.659	0,00341
Rengo	0	0	-	56.188	-
Vallenar	0	0	-	52.099	-
Villarrica	1	1	1	51.511	0,00194
Angol	0	0	-	50.821	-
Total	2.399	1.378	1,74	11.381.781	0,01211

Tabla A.11: Resultados de la búsqueda geolocalizada en Chile. Muestra para cada ciudad que ha generado resultados el número de tweets, usuarios, número de tweets por usuario, población y el porcentaje del número de usuarios frente a la población.

Colombia (49 ciudades)					
Ciudad	Tweets (búsqueda)	Usuarios	Tweets/ usuario	Población	Usuarios/ población (%)
Bogotá	643	420	1,53	7.980.001	0,00526
Medellín	370	214	1,73	2.486.723	0,00861
Cali	218	159	1,37	2.394.870	0,00664
Barranquilla	261	157	1,66	1.223.967	0,01283
Cartagena de Indias	130	100	1,3	1.013.454	0,00987
Cúcuta	65	48	1,35	656.414	0,00731
Soledad	53	37	1,43	632.014	0,00585
Ibagué	76	56	1,36	558.815	0,01002
Bucaramanga	76	57	1,33	528.352	0,01079
Soacha	111	73	1,52	522.442	0,01397
Villavicencio	46	39	1,18	495.200	0,00788
Pereira	70	38	1,84	472.023	0,00805
Bello	32	24	1,33	464.560	0,00517
Valledupar	159	83	1,92	463.218	0,01792
Montería	62	36	1,72	447.716	0,00804
San Juan de Pasto	14	9	1,56	445.511	0,00202
Buenaventura	0	0	-	407.539	-
Manizales	13	10	1,3	397.488	0,00252
Neiva	12	8	1,5	344.130	0,00232
Palmira	5	4	1,25	306.727	0,00130
Armenia	17	14	1,21	298.197	0,00469
Popayán	5	5	1	280.107	0,00179
Sincelejo	12	10	1,2	279.027	0,00358
Itaguí	15	15	1	270.920	0,00554
Riohacha	4	3	1,33	268.758	0,00112
Floridablanca	9	7	1,29	266.102	0,00263
Envigado	15	13	1,15	227.599	0,00571

Tuluá	2	2	1	214.081	0,00093
San Andrés de Tumaco	0	0	-	203.971	-
Dosquebradas	7	7	1	200.829	0,00349
Tunja	10	7	1,43	191.878	0,00365
Barrancabermeja	10	9	1,11	191.704	0,00469
San Juan de Girón	22	15	1,47	185.248	0,00810
Apartadó	2	2	1	183.716	0,00109
Uribia	0	0	-	180.385	-
Florencia	0	0	-	175.395	-
Turbo	1	1	1	163.525	0,00061
Maicao	4	3	1,33	159.675	0,00188
Piedecuesta	8	7	1,14	152.665	0,00459
Yopal	2	2	1	142.982	0,00140
Ipiales	1	1	1	141.863	0,00070
Fusagasugá	2	2	1	137.164	0,00146
Facatativá	1	1	1	134.522	0,00074
Cartago	3	3	1	132.966	0,00226
Chía	10	9	1,11	129.652	0,00694
Pitalito	1	1	1	128.251	0,00078
Zipaquirá	11	8	1,38	124.376	0,00643
Magangué	0	0	-	123.833	-
Malambo	0	0	-	123.278	-
Total	2.590	1.719	1,51	27.653.833	0,00622

Tabla A.12: Resultados de la búsqueda geolocalizada en Colombia. Muestra para cada ciudad que ha generado resultados el número de tweets, usuarios, número de tweets por usuario, población y el porcentaje del número de usuarios frente a la población.

España (15 ciudades)					
Ciudad	Tweets (búsqueda)	Usuarios	Tweets/ usuario	Población	Usuarios/ población (%)
Madrid	951	737	1,29	3.141.991	0,02346
Barcelona	372	237	1,57	1.604.555	0,01477
Valencia	276	179	1,54	786.189	0,02277
Sevilla	253	195	1,3	693.878	0,02810
Zaragoza	187	64	2,92	664.953	0,00962
Málaga	216	118	1,83	569.130	0,02073
Murcia	72	51	1,41	439.889	0,01159
Mallorca	58	30	1,93	400.578	0,00749

Gran Canaria	43	29	1,48	379.766	0,00764
Bilbao	174	62	2,81	345.141	0,01796
Alicante	65	53	1,23	328.648	0,01613
Córdoba	76	56	1,36	327.362	0,01711
Valladolid	72	45	1,6	303.905	0,01481
Vigo	39	29	1,34	294.098	0,00986
Gijón	68	24	2,83	274.290	0,00875
Total	2.922	1.909	1,53	10.554.373	0,01809

Tabla A.13: Resultados de la búsqueda geolocalizada en España. Muestra para cada ciudad que ha generado resultados el número de tweets, usuarios, número de tweets por usuario, población y el porcentaje del número de usuarios frente a la población.

México (15 ciudades)					
Ciudad	Tweets (búsqueda)	Usuarios	Tweets/ usuario	Población	Usuarios/ población (%)
México D. F.	1.738	1.266	1,37	8.851.080	0,01430
Ecatepec	25	22	1,14	1.655.015	0,00133
Guadalajara	523	361	1,45	1.495.182	0,02414
Puebla de Zaragoza	326	157	2,08	1.434.062	0,01095
Juárez	13	10	1,3	1.321.004	0,00076
Tijuana	38	25	1,52	1.300.983	0,00192
León	100	66	1,52	1.238.962	0,00533
Zapopan	63	48	1,31	1.142.483	0,00420
Monterrey	270	200	1,35	1.135.512	0,01761
Nezahualcóyotl	86	75	1,15	1.104.585	0,00679
Chihuahua	29	21	1,38	809.232	0,00260
Naucalpan de Juárez	297	239	1,24	792.211	0,03017
Mérida	160	118	1,36	777.615	0,01517
San Luis Potosí	45	38	1,18	722.772	0,00526
Aguascalientes	60	38	1,58	722.250	0,00526
Total	3.773	2.684	1,41	24.502.948	0,01095

Tabla A.14: Resultados de la búsqueda geolocalizada en México. Muestra para cada ciudad que ha generado resultados el número de tweets, usuarios, número de tweets por usuario, población y el porcentaje del número de usuarios frente a la población.

Perú (51 ciudades)					
Ciudad	Tweets (búsqueda)	Usuarios	Tweets/ usuario	Población	Usuarios/ población (%)
Lima	1.587	815	1,95	9.866.647	0,00826
Arequipa	48	31	1,55	869.351	0,00357
Trujillo	108	72	1,5	799.550	0,00901
Chiclayo	45	27	1,67	600.440	0,00450
Lambayeque	5	5	1	600.440	0,00083
Iquitos	17	15	1,13	437.376	0,00343
Piura	18	17	1,06	436.440	0,00390
Cusco	36	30	1,2	427.218	0,00702
Chimbote	13	12	1,08	371.012	0,00323
Huancayo	9	9	1	364.725	0,00247
Tacna	15	9	1,67	293.119	0,00307
Juliaca	2	2	1	273.882	0,00073
Ica	13	13	1	244.390	0,00532
Cajamarca	6	4	1,5	226.031	0,00177
Pucallpa	2	1	2	211.651	0,00047
Sullana	4	4	1	201.302	0,00199
Ayacucho	1	1	1	180.766	0,00055
Chincha Alta	0	0	-	177.219	-
Huánuco	7	4	1,75	175.068	0,00228
Huacho	8	3	2,67	153.728	0,00195
Tarapoto	19	13	1,46	144.186	0,00902
Puno	2	2	1	140.839	0,00142
Paita	0	0	-	135.422	-
Huaraz	5	4	1,25	127.041	0,00315
Tumbes	2	2	1	111.595	0,00179
Pisco	3	3	1	104.656	0,00287
Huaral	0	0	-	96.468	-
Moyobamba	0	0	-	86.015	-
San Vicente de Cañete	4	4	1	85.533	0,00468
Puerto Maldonado	2	1	2	74.494	0,00134
Moquegua	0	0	-	67.428	-
Cerro de Pasco	0	0	-	66.272	-
Barranca	1	1	1	63.812	0,00157
Andahuaylas	0	0	-	63.654	-
Yurimaguas	0	0	-	63.427	-

Chancay	0	0	-	63.378	-
Talara	2	1	2	59.682	0,00168
Ilo	1	1	1	59.572	0,00168
Abancay	0	0	-	58.741	-
Chulucanas	54	1	54	57.380	0,00174
Tingo María	1	1	1	56.932	0,00176
Sicuani	0	0	-	55.000	-
Mala	1	1	1	53.532	0,00187
Huancavelica	0	0	-	47.866	-
Ferreñafe	0	0	-	47.087	-
Chepén	0	0	-	45.897	-
Sechura	0	0	-	44.103	-
Pacasmayo	5	1	5	43.356	0,00231
Tarma	2	2	1	42.569	0,00470
Bagua Grande	0	0	-	42.396	-
Guadalupe	1	1	1	42.177	0,00237
Total	2.049	1.113	1,84	19.160.865	0,00581

Tabla A.15: Resultados de la búsqueda geolocalizada en Perú. Muestra para cada ciudad que ha generado resultados el número de tweets, usuarios, número de tweets por usuario, población y el porcentaje del número de usuarios frente a la población.

Venezuela (15 ciudades)					
Ciudad	Tweets (búsqueda)	Usuarios	Tweets/ usuario	Población	Usuarios/ población (%)
Caracas	1.147	427	2,69	7.960.076	0,00536
Maracaibo	388	204	1,9	3.990.559	0,00511
Barquisimeto	299	133	2,25	2.008.733	0,00662
Maracay	334	137	2,44	1.725.606	0,00794
Guayana	52	32	1,62	1.250.080	0,00256
San Cristóbal	127	92	1,38	1.104.820	0,00833
Barcelona	73	45	1,62	965.989	0.00466
Maturín	184	69	2,67	821.955	0,00839
Bolívar	89	36	2,47	780.953	0,00461
Cumaná	65	33	1,97	774.706	0,00426
Barinas	120	40	3	655.413	0,00610
Cabimas	80	25	3,2	599.108	0,00417

Punto Fijo	86	39	2,21	477.017	0,00818
Puerto la Cruz	94	35	2,69	472.231	0,00741
Guarenas	109	53	2,06	399.290	0,01327
Total	3.247	1.400	2,32	23.986.536	0,00584

Tabla A.16: Resultados de la búsqueda geolocalizada en Venezuela. Muestra para cada ciudad que ha generado resultados el número de tweets, usuarios, número de tweets por usuario, población y el porcentaje del número de usuarios frente a la población.

A.1.3 Resultados de la Búsqueda: Palabras Clave

También realizamos un análisis de los resultados centrándonos en las palabras clave. Para la búsqueda geolocalizada hemos definido 35 palabras clave relacionadas con distintos temas: tecnología, cultura general, política/actualidad y salud/alimentación. Hemos tratado de seleccionar las palabras clave de forma que sean variadas y lo suficientemente genéricas como para evitar la introducción de sesgo por tema. La Tabla A.17 muestra las palabras clave empleadas para la búsqueda geolocalizada, agrupadas por tema.

Tema	Palabras clave
Tecnología	App, Ciencia, Informática, Internet, Tecnología
Cultura general	Arte, Autor, Canción, Cine, Gastronomía, Libro, Museo, Música, Serie
Política/Actualidad	Banco, Corrupción, Dinero, Economía, Empresa, Gobierno, Nación, Noticias, País, Política, Presidente, Televisión
Salud/Alimentación	Alimentación, Carne, Comida, Deporte, Ejercicio, Enfermedad, Médico, Pescado, Salud

Tabla A.17: Listado de palabras clave, agrupadas por tema.

En la Tabla A.18 mostramos los resultados obtenidos en cuanto a usuarios recuperados por país y palabra clave. Cada celda representa el número de usuarios cuyos tweets han respondido a la palabra clave asociada a su fila, en el país indicado por su columna. Hay usuarios cuyos tweets pueden haber respondido a más de una palabra clave, con lo cual estarán contados en más de una fila.

Observamos que la palabra clave que más usuarios recupera es 'Comida', con 1531 usuarios de los cuales la mayoría se concentran en México con 673 usuarios, seguido de España con 248. La siguiente palabra clave que más usuarios devuelve es 'Música' con 1474 usuarios. A continuación, 'País' y 'Salud' son las que más usuarios devuelven, con 1267 y 1031 respectivamente. En general,

podemos apreciar que las palabras clave que más usuarios han captado son aquellas relacionadas con el mundo de la cultura. Por ejemplo, 'Museo' y 'Arte' superan los 900 usuarios, y 'Canción', 'Cine' y 'Libro' superan los 600. Aquellas palabras relacionadas con la política o la actualidad también recuperan bastante volumen de usuarios. Por ejemplo la palabra 'Presidente' recupera 931 usuarios, 'Gobierno' recupera 669 y con 'Política' obtenemos 396.

Hay palabras clave que devuelven mayor concentración de usuarios en un país que en el resto. Por ejemplo, la palabra 'Comida' devuelve muchos más usuarios en México (673) que en Perú (78). Aunque esto pueda producir aparentemente un sesgo, en el siguiente paso del proceso descargamos los últimos 1000 tweets de las *timelines* de los usuarios obtenidos. Como los usuarios hablan de distintos temas a lo largo del tiempo, los posibles sesgos por tema se ven diluidos.

	AR	CH	CO	ES	ME	PE	VE	Total
App	13	5	16	15	8	23	20	100
Ciencia	10	11	4	21	5	8	4	63
Informática	12	8	1	3	4	4	1	33
Internet	47	25	14	19	28	28	65	226
Tecnología	20	11	11	7	49	10	11	119
Arte	149	134	89	215	234	70	42	933
Autor	7	7	11	8	13	2	3	51
Canción	175	51	152	77	67	79	59	660
Cine	105	132	115	128	95	89	32	696
Gastronomía	6	6	10	29	22	8	2	83
Libro	106	82	62	98	254	46	26	674
Museo	71	113	103	202	372	67	18	946
Música	439	140	293	241	151	109	101	1474
Serie	46	17	26	33	24	13	16	175
Banco	258	66	20	15	33	36	33	461
Corrupción	38	7	16	40	13	17	45	176
Dinero	33	25	58	65	72	47	57	357
Economía	37	49	14	29	36	25	33	223
Empresa	63	48	33	150	33	27	39	393
Gobierno	172	46	51	66	104	47	183	669
Nación	64	4	11	5	12	44	21	161
Noticias	56	34	53	52	36	52	42	325
País	247	70	166	133	64	104	483	1267
Política	81	26	29	85	23	56	96	396
Presidente	348	56	54	84	90	33	266	931
Televisión	20	42	48	18	29	62	20	239
Alimentación	1	5	6	8	4	2	11	37
Carne	41	27	53	33	54	13	31	252
Comida	156	126	141	248	673	109	78	1531
Deporte	40	44	40	44	22	20	19	229
Ejercicio	25	18	41	19	37	15	11	166
Enfermedad	22	6	9	12	14	4	11	78
Médico	52	34	23	12	93	18	11	243
Pescado	10	20	10	12	17	15	4	88
Salud	161	156	254	72	188	98	102	1031
Total	3131	1651	2037	2298	2973	1400	1996	15486

Tabla A.18: Número de usuarios obtenidos para cada palabra clave y para cada país: Argentina (AR), Chile (CH), Colombia (CO), México (ME), Perú (PE), Venezuela (VE).

A.2 Recuperación de Timelines

En esta sección analizamos el estado del corpus tras descargar hasta los últimos 1000 tweets de las *timelines* de los usuarios recuperados. En las siguientes secciones mostramos estadísticas para el corpus resultante desde dos puntos de vista: i) *cuantitativo*: relativo a la cantidad de tweets obtenidos y ii) *temporal*: relativo a sus fechas de publicación.

A.2.1 Análisis Cuantitativo de los Resultados

Las Tablas A.19–A.25 muestran como ha quedado el el corpus tras recuperar los 1000 tweets más recientes de cada usuario, para cada ciudad de cada país que ha generado resultados durante la búsqueda geolocalizada realizada previamente. Vemos que el país que más tweets recupera es Argentina (1.684.662) seguido de cerca por México (1.571.859). El resto de países obtienen una cantidad menor de tweets, en general alrededor de un millón. Perú es el país para el cual recuperamos menos tweets (676.623), seguido por Venezuela (715.987).

Argentina y Chile son los países que presentan un mayor número de tweets por usuario, 704,00 y 769,03 respectivamente. Para el resto de países recuperamos alrededor de 600 tweets por usuario. Esta proporción es algo más baja para Venezuela, con 511,42 tweets por usuario.

En cuanto a la longitud de los tweets en palabras, la mayoría de ciudades presentan una longitud entre 10 y 12 palabras por tweet. Aquellas ciudades en las que la longitud se desvía de este rango suele ser debido a que hemos obtenido pocos tweets y usuarios. En cuanto a la longitud en caracteres, vemos que la mayoría de ciudades oscila entre los 65 y 90 caracteres. Argentina presenta los tweets más cortos, con 65,18 caracteres de media. Los más largos vienen dados por Venezuela, con 90,33 caracteres por tweet de media.

Argentina (15 ciudades)					
Ciudad	Usuarios	Tweets (*timelines*)	Tweets/ usuario	Longitud media Palabras	Caracteres
Buenos Aires	830	587.730	708,11	11,59	72,23
Córdoba	300	204.274	680,91	9,67	57,90
Rosario	162	114.895	709,23	11,06	66,06
Mendoza	89	58.665	659,16	10,35	63,29
La Plata	183	130.606	713,69	10,36	61,63
Mar del Plata	111	87.184	785,44	10,23	60,67
San Miguel de Tucumán	67	44.045	657,39	9,89	58,41
Salta	30	18.312	610,40	11,99	76,80
Santa Fe	128	96.826	756,45	9,49	53,92
Corrientes	37	26.272	710,05	11,34	68,51
Bahía Blanca	78	57.823	741,32	9,92	56,69
Resistencia	61	38.474	630,72	9,69	59,39
Partido de Vicente López	185	127.155	687,32	11,12	69,36
Posadas	85	63.832	750,96	10,13	61,79
San Juan	47	28.569	607,85	9,16	54,68
Total	2.393	1.684.662	704,00	10,72	65,18

Tabla A.19: Número de usuarios, tweets y tweets por usuario obtenidos tras recuperar de las los últimos 1000 tweets de los usuarios de Argentina. También mostramos la longitud media de los tweets, en palabras y caracteres. Mostramos únicamente aquellas ciudades para las que hemos obtenido algún tweet en la búsqueda geolocalizada.

Chile (21 ciudades)					
Ciudad	Usuarios	Tweets (*timelines*)	Tweets/ usuario	Longitud media Palabras	Caracteres
Santiago de Chile	972	758.595	780,45	11,76	78,18
Gran Concepción	31	23.132	746,19	11,75	78,32
Gran Valparaíso	117	93.532	799,42	11,39	74,61
Gran Temuco	19	15.382	809,58	12,36	79,31
Antofagasta	42	30.614	728,90	11,47	75,83
Talca	41	29.701	724,41	11,38	72,41

Arica	23	16.113	700,57	12,06	78,13
Los Ángeles	8	6.211	776,38	10,54	67,90
Copiapó	10	7.099	709,90	12,09	76,28
Valdivia	25	18.431	737,24	11,29	73,48
Osorno	11	7.970	724,55	11,21	72,89
Curicó	24	19.602	816,75	12,02	76,89
Calama	9	5.153	572,56	12,44	81,64
Punta Arenas	22	16.146	733,91	11,25	74,57
Melipilla	7	3.289	469,86	12,11	76,97
Ovalle	4	1.721	430,25	15,84	107,63
San Fernando	2	1.341	670,50	9,96	69,28
Talagante	2	1.290	645,00	12,70	91,16
Los Andes	6	3.495	582,50	12,79	75,56
Coyhaique	2	2	1,00	14,50	95,00
Villarrica	1	905	905,00	13,53	96,03
Total	1.378	1.059.724	769,03	11,71	77,45

Tabla A.20: Número de usuarios, tweets y tweets por usuario obtenidos tras recuperar de las los últimos 1000 tweets de los usuarios de Chile. También mostramos la longitud media de los tweets, en palabras y caracteres. Mostramos únicamente aquellas ciudades para las que hemos obtenido algún tweet en la búsqueda geolocalizada.

Colombia (43 ciudades)					
Ciudad	Usuarios	Tweets (*timelines*)	Tweets/ usuario	Longitud media	
				Palabras	Caracteres
Bogotá	420	281.957	671,33	12,29	81,57
Medellín	214	144.926	677,22	12,51	83,82
Cali	159	97.605	613,87	12,57	84,96
Barranquilla	157	100.243	638,49	11,69	76,24
Cartagena de Indias	100	63.933	639,33	12,51	83,36
Cúcuta	48	29.190	608,12	11,94	80,36
Soledad	37	22.615	611,22	10,62	66,18
Ibagué	56	34.739	620,34	11,19	69,80
Bucaramanga	57	35.587	624,33	12,23	80,72
Soacha	73	36.546	500,63	10,82	63,28
Villavicencio	39	22.471	576,18	12,17	81,60
Pereira	38	24.631	648,18	11,08	70,78
Bello	24	13.429	559,54	11,44	72,97
Valledupar	83	53.999	650,59	12,33	83,71
Montería	36	19.745	548,47	12,56	85,26

San Juan de Pasto	9	4.578	508,67	13,19	87,14
Manizales	10	5.005	500,50	10,23	59,91
Neiva	8	5.782	722,75	12,06	74,83
Palmira	4	1.809	452,25	10,54	61,36
Armenia	14	6.423	458,79	12,89	82,88
Popayán	5	3.188	637,60	11,41	74,00
Sincelejo	10	5.457	545,70	12,69	79,21
Itaguí	15	8.637	575,80	11,09	75,51
Riohacha	3	1.848	616,00	13,60	83,77
Floridablanca	7	3.720	531,43	9,69	60,73
Envigado	13	9.519	732,23	12,60	85,04
Tuluá	2	32	16,00	14,28	89,72
Dosquebradas	7	5.917	845,29	9,05	54,78
Tunja	7	5.172	738,86	15,98	102,73
Barrancabermeja	9	3.816	424,00	10,78	66,86
San Juan de Girón	15	10.821	721,40	12,79	84,99
Apartadó	2	786	393,00	14,16	99,01
Turbo	1	563	563,00	20,02	123,07
Maicao	3	1.869	623,00	12,86	88,54
Piedecuesta	7	3.158	451,14	11,21	74,34
Yopal	2	836	418,00	11,70	81,09
Ipiales	1	840	840,00	15,11	104,51
Fusagasugá	2	1.030	515,00	11,52	65,56
Facatativá	1	1	1,00	8,00	72,00
Cartago	3	2.653	884,33	12,57	85,22
Chía	9	4.380	486,67	13,50	86,19
Pitalito	1	837	837,00	11,74	70,73
Zipaquirá	8	4.672	584,00	12,47	80,60
Total	1.719	1.084.965	631,16	12,13	79,96

Tabla A.21: Número de usuarios, tweets y tweets por usuario obtenidos tras recuperar de las los últimos 1000 tweets de los usuarios de Colombia. También mostramos la longitud media de los tweets, en palabras y caracteres. Mostramos únicamente aquellas ciudades para las que hemos obtenido algún tweet en la búsqueda geolocalizada.

España (15 ciudades)					
Ciudad	Usuarios	Tweets (*timelines*)	Tweets/ usuario	Longitud media Palabras	Caracteres
Madrid	737	446.238	605,48	12,75	83,80
Barcelona	237	137.317	579,40	13,43	91,26
Valencia	179	105.914	591,70	12,87	85,45
Sevilla	195	111.510	571,85	12,44	79,84
Zaragoza	64	43.088	673,25	13,20	87,80
Málaga	118	74.134	628,25	12,22	78,35
Murcia	51	28.757	563,86	12,45	82,20
Mallorca	30	19.573	652,43	13,12	88,80
Gran Canaria	29	19.246	663,66	12,96	83,28
Bilbao	62	31.468	507,55	12,02	79,23
Alicante	53	35.841	676,25	12,74	83,54
Córdoba	56	34.099	608,91	13,65	88,12
Valladolid	45	26.256	583,47	13,57	86,87
Vigo	29	19.091	658,31	10,89	67,98
Gijón	24	15.991	666,29	13,65	85,43
Total	1.909	1.148.523	601,64	12,81	84,12

Tabla A.22: Número de usuarios, tweets y tweets por usuario obtenidos tras recuperar de las los últimos 1000 tweets de los usuarios de España. También mostramos la longitud media de los tweets, en palabras y caracteres. Mostramos únicamente aquellas ciudades para las que hemos obtenido algún tweet en la búsqueda geolocalizada.

México (15 ciudades)					
Ciudad	Usuarios	Tweets (*timelines*)	Tweets/ usuario	Longitud media Palabras	Caracteres
México D. F.	1.266	737.290	582,38	11,85	77,11
Ecatepec	22	10.385	472,05	11,87	75,97
Guadalajara	361	211.429	585,68	11,92	77,85
Puebla de Zaragoza	157	99.150	631,53	11,85	76,49
Juárez	10	5.069	506,90	12,97	84,36
Tijuana	25	16.744	669,76	12,55	85,56
León	66	38.195	578,71	12,28	79,24
Zapopan	48	23.072	480,67	12,00	79,82
Monterrey	200	118.007	590,03	11,06	72,29
Nezahualcóyotl	75	42.412	565,49	12,25	77,54
Chihuahua	21	9.227	439,38	11,91	76,89
Naucalpan de Juárez	239	144.375	604,08	12,01	77,60

Mérida	118	68.926	584,12	11,37	74,48
San Luis Potosí	38	21.159	556,82	11,88	76,60
Aguascalientes	38	26.419	695,24	11,03	72,87
Total	2.684	1.571.859	585,64	11,82	76,87

Tabla A.23: Número de usuarios, tweets y tweets por usuario obtenidos tras recuperar de las los últimos 1000 tweets de los usuarios de México. También mostramos la longitud media de los tweets, en palabras y caracteres. Mostramos únicamente aquellas ciudades para las que hemos obtenido algún tweet en la búsqueda geolocalizada.

Perú (45 ciudades)					
Ciudad	Usuarios	Tweets (*timelines*)	Tweets/ usuario	Longitud media	
				Palabras	Caracteres
Lima	815	514.474	631,26	11,33	73,00
Arequipa	31	16.998	548,32	11,50	76,07
Trujillo	72	45.628	633,72	10,78	66,07
Chiclayo	27	13.021	482,26	11,79	75,76
Lambayeque	5	2.619	523,80	12,56	73,26
Iquitos	15	9.887	659,13	11,31	72,70
Piura	17	9.009	529,94	13,00	84,62
Cusco	30	19.577	652,57	11,26	74,66
Chimbote	12	2.888	240,67	11,95	76,82
Huancayo	9	4.304	478,22	10,48	65,20
Tacna	9	7.548	838,67	13,58	89,64
Juliaca	2	1.909	954,50	12,47	79,17
Ica	13	4.835	371,92	9,79	66,48
Cajamarca	4	2.554	638,50	12,78	83,60
Pucallpa	1	9	9,00	13,33	92,00
Sullana	4	1.317	329,25	9,36	58,16
Ayacucho	1	892	892,00	8,51	48,72
Huánuco	4	2.846	711,50	11,10	68,31
Huacho	3	2.599	866,33	10,04	66,44
Tarapoto	13	5.376	413,54	12,22	71,41
Puno	2	943	471,50	6,84	45,23
Huaraz	4	927	231,75	8,81	53,64
Tumbes	2	864	432,00	11,80	77,79
Pisco	3	3	1,00	9,00	72,33
San Vicente de Cañete	4	874	218,50	12,89	79,86
Puerto Maldonado	1	275	275,00	7,08	52,83
Barranca	1	460	460,00	12,93	72,69

Talara	1	23	23,00	19,96	129,30
Ilo	1	982	982,00	8,53	58,82
Chulucanas	1	175	175,00	18,90	127,57
Tingo María	1	10	10,00	8,60	52,90
Mala	1	940	940,00	12,19	82,99
Pacasmayo	1	886	886,00	13,92	96,74
Tarma	2	970	485,00	9,65	57,61
Guadalupe	1	1	1,00	7,00	44,00
Total	1.113	676.623	607,93	11,34	72,87

Tabla A.24: Número de usuarios, tweets y tweets por usuario obtenidos tras recuperar de las los últimos 1000 tweets de los usuarios de Perú. También mostramos la longitud media de los tweets, en palabras y caracteres. Mostramos únicamente aquellas ciudades para las que hemos obtenido algún tweet en la búsqueda geolocalizada.

Venezuela (15 ciudades)					
Ciudad	Usuarios	Tweets (*timelines*)	Tweets/ usuario	Longitud media	
				Palabras	Caracteres
Caracas	427	222.659	521,45	13,49	90,16
Maracaibo	204	132.011	647,11	13,47	91,40
Barquisimeto	133	66.569	500,52	14,20	94,83
Maracay	137	60.667	442,82	13,81	91,73
Guayana	32	13.836	432,38	12,75	85,10
San Cristóbal	92	49.262	535,46	13,33	86,36
Barcelona	45	15.992	355,38	13,91	92,84
Maturín	69	27.481	398,28	14,80	97,36
Bolívar	36	19.132	531,44	13,08	90,61
Cumaná	33	13.956	422,91	14,21	95,06
Barinas	40	16.696	417,40	14,64	98,45
Cabimas	25	13.945	557,80	10,95	68,65
Punto Fijo	39	18.571	476,18	11,40	71,85
Puerto la Cruz	35	20.558	587,37	13,81	90,44
Guarenas	53	24.652	465,13	13,56	89,73
Total	1.400	715.987	511,42	13,55	90,33

Tabla A.25: Número de usuarios, tweets y tweets por usuario obtenidos tras recuperar de las los últimos 1000 tweets de los usuarios de Venezuela. También mostramos la longitud media de los tweets, en palabras y caracteres. Mostramos únicamente aquellas ciudades para las que hemos obtenido algún tweet en la búsqueda geolocalizada.

A.2.2 Análisis Temporal

En esta sección analizamos el corpus obtenido desde el punto de vista temporal. El interés de este análisis es que el idioma varía con el paso del tiempo y cada vez más rápidamente. Por este motivo, para abordar esta tarea resulta muy conveniente que los tweets obtenidos se encuentren en el mismo intervalo temporal similar para todos los países. En las Tablas A.26–A.32 mostramos la fecha del primer y último tweet para cada ciudad de cada país, junto con la diferencia en meses entre ambos.

Observamos que los primeros tweets datan entre 2009 y 2011 en su mayoría, aunque en algunos casos algo más temprano (2007-2008) o más tarde (2012-2013). Por ejemplo, para España contamos con tweets desde 2007 y 2008 mientras que en Chile recuperamos tweets desde 2009. Las fechas finales comprenden desde finales de diciembre de 2015 hasta principios de enero de 2016. En las ciudades más pobladas la diferencia entre las fechas del primer y el último tweet ronda los 80 meses, a excepción de Madrid cuya diferencia es de casi 103 meses. En el resto de las ciudades esta cantidad fluctúa entre los 40 y 70 meses aproximadamente. Aquellas ciudades con pocos usuarios o tweets muestran una diferencia anómala respecto el resto. Por ejemplo, la ciudad de Ovalle (Chile) cuenta con 4 usuarios y una diferencia de 7,3 meses entre el primer y último tweet, o Pitalito (Colombia) con 1 usuario y una diferencia de 1,5 meses.

Argentina (15 ciudades)			
Ciudad	Primer tweet	Último tweet	Diferencia (meses)
Buenos Aires	03-03-2009	27-12-2015	82,97
Córdoba	08-04-2008	27-12-2015	93,93
Rosario	13-05-2010	26-12-2015	68,43
Mendoza	20-05-2010	26-12-2015	68,20
La Plata	05-06-2010	27-12-2015	67,67
Mar del Plata	17-05-2009	26-12-2015	80,47
San Miguel de Tucumán	04-06-2010	26-12-2015	67,67
Salta	29-09-2010	25-12-2015	63,77
Santa Fe	17-08-2010	26-12-2015	65,23
Corrientes	23-04-2011	26-12-2015	56,93
Bahía Blanca	27-07-2010	26-12-2015	65,93
Resistencia	01-06-2010	26-12-2015	67,77
Partido de Vicente López	17-09-2008	26-12-2015	88,53
Posadas	21-05-2010	26-12-2015	68,17
San Juan	02-04-2010	26-12-2015	69,80
Total	08-04-2008	27-12-2015	93,93

Tabla A.26: Estadísticas relativas a las fechas de los tweets a nivel de ciudad. Mostramos la fecha del primer y el último tweet, la diferencia en meses, y la media y la desviación típica de esta diferencia calculada para cada usuario de Argentina.

Chile (21 ciudaes)			
Ciudad	Primer tweet	Último tweet	Diferencia (meses)
Santiago de Chile	23-04-2009	04-01-2016	81,57
Gran Concepción	14-05-2012	04-01-2016	44,33
Gran Valparaíso	18-01-2010	04-01-2016	72,57
Gran Temuco	22-08-2011	02-01-2016	53,13
Antofagasta	30-10-2009	04-01-2016	75,20
Talca	09-10-2010	04-01-2016	63,73
Arica	05-01-2011	04-01-2016	60,80
Los Ángeles	14-04-2013	02-01-2016	33,10
Copiapó	03-06-2010	04-01-2016	68,00
Valdivia	28-06-2009	04-01-2016	79,37
Osorno	19-05-2012	04-01-2016	44,17
Curicó	28-08-2012	03-01-2016	40,77
Calama	25-02-2012	03-01-2016	46,93
Punta Arenas	01-03-2010	04-01-2016	71,17
Melipilla	05-05-2010	03-01-2016	68,93
Ovalle	26-05-2015	01-01-2016	7,30
San Fernando	12-05-2010	03-01-2016	68,70
Talagante	02-07-2012	03-01-2016	42,63
Los Andes	24-06-2014	03-01-2016	18,60
Coyhaique	26-12-2015	28-12-2015	0,03
Villarrica	13-12-2013	29-12-2015	24,83
Total	23-04-2009	04-01-2016	81,57

Tabla A.27: Estadísticas relativas a las fechas de los tweets a nivel de ciudad. Mostramos la fecha del primer y el último tweet, la diferencia en meses, y la media y la desviación típica de esta diferencia calculada para cada usuario de Chile.

Colombia (43 ciudades)			
Ciudad	Primer tweet	Último tweet	Diferencia (meses)
Bogotá	26-06-2009	05-01-2016	79,43
Medellín	16-08-2009	05-01-2016	77,73
Cali	22-04-2010	05-01-2016	69,43
Barranquilla	16-02-2010	05-01-2016	71,60
Cartagena de Indias	29-12-2009	05-01-2016	73,27
Cúcuta	18-04-2011	05-01-2016	57,40
Soledad	15-07-2010	05-01-2016	66,63
Ibagué	12-03-2012	04-01-2016	46,43
Bucaramanga	02-11-2011	05-01-2016	50,80
Soacha	04-04-2010	05-01-2016	70,03

Villavicencio	17-07-2010	05-01-2016	66,57
Pereira	25-08-2011	04-01-2016	53,07
Bello	02-09-2010	03-01-2016	64,97
Valledupar	20-06-2010	04-01-2016	67,47
Montería	03-03-2011	04-01-2016	58,93
San Juan de Pasto	03-03-2009	04-01-2016	83,27
Manizales	08-05-2011	04-01-2016	56,73
Neiva	13-04-2012	04-01-2016	45,37
Palmira	23-05-2015	05-01-2016	7,53
Armenia	18-02-2011	04-01-2016	59,37
Popayán	02-03-2012	05-01-2016	46,77
Sincelejo	21-09-2011	04-01-2016	52,20
Itaguí	16-02-2012	04-01-2016	47,27
Riohacha	04-02-2013	05-01-2016	35,47
Floridablanca	23-09-2012	04-01-2016	39,93
Envigado	21-07-2010	04-01-2016	66,40
Tuluá	15-10-2015	02-01-2016	2,63
Dosquebradas	18-12-2011	04-01-2016	49,27
Tunja	16-12-2012	05-01-2016	37,13
Barrancabermeja	24-05-2012	04-01-2016	43,97
San Juan de Girón	03-03-2013	04-01-2016	34,53
Apartadó	13-07-2010	01-01-2016	66,57
Turbo	05-03-2013	04-01-2016	34,50
Maicao	22-11-2011	04-01-2016	50,13
Piedecuesta	27-02-2013	04-01-2016	34,70
Yopal	04-10-2013	04-01-2016	27,40
Ipiales	14-05-2015	04-01-2016	7,83
Fusagasugá	23-07-2014	03-01-2016	17,60
Facatativá	30-12-2015	30-12-2015	0,00
Cartago	16-02-2013	04-01-2016	35,03
Chía	27-02-2013	05-01-2016	34,70
Pitalito	19-11-2015	04-01-2016	1,50
Zipaquirá	27-02-2013	05-01-2016	34,73
Total	03-03-2009	05-01-2016	83,30

Tabla A.28: Estadísticas relativas a las fechas de los tweets a nivel de ciudad. Mostramos la fecha del primer y el último tweet, la diferencia en meses, y la media y la desviación típica de esta diferencia calculada para cada usuario de Colombia.

España (15 ciudades)			
Ciudad	Primer tweet	Último tweet	Diferencia (meses)
Madrid	16-07-2007	29-12-2015	102,93
Barcelona	14-08-2009	29-12-2015	77,60

Valencia	19-02-2009	29-12-2015	83,43
Sevilla	30-11-2008	29-12-2015	86,13
Zaragoza	23-07-2010	29-12-2015	66,17
Málaga	01-06-2010	29-12-2015	67,90
Murcia	02-08-2011	29-12-2015	53,63
Mallorca	09-08-2011	29-12-2015	53,40
Gran Canaria	11-03-2009	29-12-2015	82,77
Bilbao	04-10-2010	28-12-2015	63,70
Alicante	17-09-2011	29-12-2015	52,10
Córdoba	04-11-2011	29-12-2015	50,50
Valladolid	05-08-2011	29-12-2015	53,53
Vigo	10-03-2011	29-12-2015	58,50
Gijón	08-03-2012	29-12-2015	46,33
Total	16-07-2007	29-12-2015	102,93

Tabla A.29: Estadísticas relativas a las fechas de los tweets a nivel de ciudad. Mostramos la fecha del primer y el último tweet, la diferencia en meses, y la media y la desviación típica de esta diferencia calculada para cada usuario de España.

México (15 ciudades)			
Ciudad	Primer tweet	Último tweet	Diferencia (meses)
México D. F.	08-07-2009	30-12-2015	78,87
Ecatepec	07-10-2012	30-12-2015	39,30
Guadalajara	03-07-2009	30-12-2015	79,00
Puebla de Zaragoza	19-11-2009	30-12-2015	74,40
Ciudad Juárez	30-03-2010	22-12-2015	69,73
Tijuana	09-04-2010	30-12-2015	69,70
León	28-04-2010	30-12-2015	69,07
Zapopan	22-07-2010	30-12-2015	66,20
Monterrey	14-09-2009	30-12-2015	76,60
Nezahualcóyotl	29-10-2009	30-12-2015	75,10
Chihuahua	22-06-2010	30-12-2015	67,20
Naucalpan de Juárez	27-03-2009	30-12-2015	82,30
Mérida	18-05-2010	30-12-2015	68,40
San Luis Potosí	01-06-2010	30-12-2015	67,90
Aguascalientes	14-01-2010	30-12-2015	72,53
Total	27-03-2009	30-12-2015	82,30

Tabla A.30: Estadísticas relativas a las fechas de los tweets a nivel de ciudad. Mostramos la fecha del primer y el último tweet, la diferencia en meses, y la media y la desviación típica de esta diferencia calculada para cada usuario de México.

Perú (35 ciudades)			
Ciudad	Primer tweet	Último tweet	Diferencia (meses)
Lima	01-04-2009	21-01-2016	82,83
Arequipa	21-09-2009	20-01-2016	77,07
Trujillo	03-09-2009	21-01-2016	77,70
Chiclayo	28-09-2011	19-01-2016	52,47
Lambayeque	06-12-2013	20-01-2016	25,80
Iquitos	05-09-2011	19-01-2016	53,20
Piura	09-06-2010	20-01-2016	68,37
Cusco	09-07-2009	19-01-2016	79,50
Chimbote	04-11-2013	18-01-2016	26,83
Huancayo	27-01-2011	14-01-2016	60,40
Tacna	10-10-2013	20-01-2016	27,73
Juliaca	03-10-2014	04-01-2016	15,27
Ica	03-08-2012	19-01-2016	42,10
Cajamarca	29-10-2014	13-01-2016	14,67
Pucallpa	10-03-2015	15-12-2015	9,33
Sullana	29-01-2013	20-01-2016	36,17
Ayacucho	13-07-2015	02-01-2016	5,73
Huánuco	18-05-2014	20-01-2016	20,40
Huacho	05-11-2013	18-01-2016	26,80
Tarapoto	25-02-2014	20-01-2016	23,10
Puno	06-11-2015	20-01-2016	2,50
Huaraz	23-09-2015	17-01-2016	3,87
Tumbes	21-03-2013	21-01-2016	34,53
Pisco	16-01-2016	18-01-2016	0,07
San Vicente de Cañete	04-08-2010	20-01-2016	66,47
Puerto Maldonado	24-10-2012	04-01-2016	38,87
Barranca	06-02-2012	20-01-2016	48,10
Talara	13-01-2016	20-01-2016	0,20
Ilo	03-10-2013	03-01-2016	27,37
Chulucanas	16-12-2015	19-01-2016	1,13
Tingo María	05-10-2015	05-01-2016	3,03
Mala	18-03-2015	20-01-2016	10,23
Pacasmayo	17-03-2015	16-01-2016	10,17
Tarma	14-11-2011	19-01-2016	50,90
Guadalupe	15-01-2016	15-01-2016	0,00
Total	01-04-2009	21-01-2016	82,83

Tabla A.31: Estadísticas relativas a las fechas de los tweets a nivel de ciudad. Mostramos la fecha del primer y el último tweet, la diferencia en meses, y la media y la desviación típica de esta diferencia calculada para cada usuario de Perú.

Venezuela (15 ciudades)			
Ciudad	Primer tweet	Último tweet	Diferencia (meses)
Caracas	03-09-2009	01-01-2016	77,03
Maracaibo	02-10-2009	31-12-2015	76,03
Barquisimeto	18-12-2009	31-12-2015	73,47
Maracay	01-12-2009	31-12-2015	74,03
Guayana	07-02-2011	31-12-2015	59,57
San Cristóbal	27-01-2010	31-12-2015	72,13
Barcelona	02-05-2010	31-12-2015	68,93
Maturín	13-11-2009	31-12-2015	74,60
Bolívar	20-06-2010	31-12-2015	67,33
Cumaná	25-03-2010	31-12-2015	70,20
Barinas	26-05-2010	31-12-2015	68,13
Cabimas	12-11-2010	31-12-2015	62,47
Punto Fijo	17-11-2011	31-12-2015	50,13
Puerto La Cruz	15-05-2010	31-12-2015	68,53
Guarenas	01-05-2010	31-12-2015	69,00
Total	03-09-2009	01-01-2016	77,03

Tabla A.32: Estadísticas relativas a las fechas de los tweets a nivel de ciudad. Mostramos la fecha del primer y el último tweet, la diferencia en meses, y la media y la desviación típica de esta diferencia calculada para cada usuario de Venezuela.

Además de analizar el intervalo temporal a nivel de ciudad independientemente del usuario, también resulta interesante estudiar la diferencia temporal entre el primer tweet y último tweet de cada usuario. Con esta finalidad exponemos las Tablas A.33–A.39 que muestran las medidas de posición para la diferencia (en meses) entre el primer y último tweet de cada usuario individualmente, para cada país y ciudad. Más concretamente mostramos el mínimo, el primer cuartil, la media, la desviación típica, la mediana, el tercer cuartil y el máximo

Observamos que las medidas presentan una alta variabilidad. Aunque el primer cuartil está ubicado entre los 0 y 5 meses aproximadamente, especialmente en las ciudades que obtenemos un número elevado de usuarios como en Argentina o España. Hay bastantes ciudades que se salen de estas estadísticas, como San Fernando (Chile) con el primer cuartil en 38,22 meses. Estos datos suelen presentarse en ciudades con pocos usuarios. En el otro extremo, el tercer cuartil suele situarse entre los 15 y 26 meses, aunque con mucha variabilidad. Vemos que por la forma en que se distribuyen los datos, la media y la mediana no suelen coincidir, siendo esta última menor en todos los casos. La mediana comprende valores entre los 6 y 10 meses, valores aproximados para los cuales se concentra la mayor cantidad de tweets. La media comprende de los 10 a los 20 meses aproximadamente, alrededor de 12-16 para las capitales. Las desviaciones

típicas toman valores similares, ligeramente superiores a las medias.

Argentina (15 ciudades)							
Ciudad	Min	1Q	Media	SDev	Mediana	3Q	Max
Buenos Aires	0,00	2,71	15,07	16,37	9,48	21,38	82,83
Córdoba	0,00	1,33	10,03	13,55	5,17	14,03	93,83
Rosario	0,00	2,53	12,92	14,69	6,28	19,25	68,03
Mendoza	0,00	2,70	14,39	17,63	5,23	21,50	68,13
La Plata	0,00	1,95	11,12	13,39	5,40	16,12	67,50
Mar del Plata	0,00	2,08	11,44	17,14	4,63	11,28	80,00
San Miguel de Tucumán	0,00	2,45	13,07	16,04	6,70	16,53	67,20
Salta	0,00	1,18	13,46	16,56	9,28	17,30	63,37
Santa Fe	0,00	2,68	10,22	13,20	5,12	11,46	64,90
Corrientes	0,00	4,87	14,38	13,88	12,13	16,63	56,23
Bahía Blanca	0,00	1,59	8,67	12,80	4,27	9,02	65,80
Resistencia	0,00	1,20	10,41	14,26	3,97	14,47	67,40
Partido de Vicente López	0,00	1,87	12,90	16,21	6,57	17,70	88,03
Posadas	0,00	2,40	11,28	12,92	6,47	17,10	67,43
San Juan	0,00	1,28	10,13	14,38	5,23	13,25	69,27
Total	0,00	2,13	12,73	15,38	6,53	17,60	93,83

Tabla A.33: Medidas de posición para la diferencia en meses entre el primer y último tweet de cada usuario individualmente, para cada ciudad de Argentina. Mostramos el mínimo, el primer cuartil (1Q), la media, la desviación típica, la mediana, el tercer cuartil (3Q) y el máximo.

Chile (21 ciudades)							
Ciudad	Min	1Q	Media	SDev	Mediana	3Q	Max
Santiago de Chile	0,00	3,33	16,32	17,41	10,07	23,37	81,23
Gran Concepción	0,00	1,23	11,15	12,19	6,37	16,57	43,73
Gran Valparaíso	0,00	3,60	14,35	15,34	9,33	19,30	72,23
Gran Temuco	0,00	7,42	17,94	13,42	16,13	25,20	52,47
Antofagasta	0,00	4,99	21,83	21,80	14,37	30,21	74,97
Talca	0,00	3,63	20,64	20,55	12,57	29,60	63,47
Arica	0,00	2,18	17,24	19,57	5,63	28,65	59,97
Los Ángeles	2,70	5,94	12,31	10,80	7,03	17,10	32,23
Copiapó	0,00	3,30	20,40	23,63	10,68	30,57	67,97

Valdivia	0,00	6,73	30,67	25,06	28,00	54,23	78,70
Osorno	0,00	3,02	15,16	16,37	8,73	25,83	44,10
Curicó	0,00	3,41	12,67	11,51	8,65	18,96	40,63
Calama	0,00	0,90	15,33	16,35	14,40	25,43	46,67
Punta Arenas	0,00	1,46	16,56	19,52	7,50	28,57	70,37
Melipilla	0,00	5,08	26,92	28,47	10,07	49,85	68,50
Ovalle	0,00	0,00	3,08	3,65	2,55	5,63	7,20
San Fernando	28,07	38,22	48,38	28,73	48,38	58,54	68,70
Talagante	10,83	18,75	26,67	22,39	26,67	34,58	42,50
Los Andes	0,00	0,84	5,99	7,21	3,57	8,62	18,57
Coyhaique	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Villarrica	24,83	24,83	24,83	NA	24,83	24,83	24,83
Total	0,00	3,30	16,58	17,70	10,07	24,42	81,23

Tabla A.34: Medidas de posición para la diferencia en meses entre el primer y último tweet de cada usuario individualmente, para cada ciudad de Chile. Mostramos el mínimo, el primer cuartil (1Q), la media, la desviación típica, la mediana, el tercer cuartil (3Q) y el máximo.

Colombia (43 ciudades)							
Ciudad	Min	1Q	Media	SDev	Mediana	3Q	Max
Bogotá	0,00	4,07	18,85	18,40	13,07	29,69	79,20
Medellín	0,00	5,22	20,42	18,65	15,43	31,58	77,67
Cali	0,00	4,18	20,74	18,81	16,87	33,05	69,17
Barranquilla	0,00	1,47	15,15	16,19	9,10	27,13	71,53
Cartagena de Indias	0,00	9,11	23,90	18,68	21,52	35,75	73,13
Cúcuta	0,00	4,72	18,70	16,85	13,73	29,58	56,73
Soledad	0,00	0,90	17,75	21,56	8,57	26,50	66,53
Ibagué	0,00	1,18	12,88	14,40	7,27	20,96	46,03
Bucaramanga	0,00	3,30	15,81	14,26	13,13	24,90	49,80
Soacha	0,00	0,63	11,29	15,14	5,40	18,03	70,00
Villavicencio	0,00	6,67	17,67	16,24	13,97	23,72	66,53
Pereira	0,00	6,72	17,89	15,25	15,23	26,98	53,03
Bello	0,00	2,49	20,03	20,53	12,38	32,66	64,80
Valledupar	0,00	3,92	17,51	16,45	11,50	31,98	67,37
Montería	0,00	0,98	14,80	16,50	7,57	31,08	58,63
San Juan de Pasto	0,00	2,97	22,39	30,65	7,87	33,07	83,17
Manizales	0,00	4,04	19,93	21,69	8,28	36,55	56,57
Neiva	0,00	11,27	20,14	13,49	21,38	25,21	45,27
Palmira	0,00	0,00	2,86	3,61	1,97	4,83	7,50
Armenia	0,00	0,00	15,73	21,46	2,08	33,58	59,23
Popayán	4,57	23,63	27,24	15,44	26,97	34,43	46,60

Sincelejo	0,00	2,38	20,34	18,03	21,38	33,48	52,10
Itaguí	0,00	0,37	14,87	15,91	7,97	25,67	47,20
Riohacha	0,00	10,75	18,98	17,85	21,50	28,47	35,43
Floridablanca	0,00	1,15	9,85	14,15	3,27	11,77	39,83
Envigado	0,00	4,43	18,27	20,04	13,47	24,07	66,30
Tuluá	0,00	0,66	1,32	1,86	1,32	1,98	2,63
Dosquebradas	0,97	7,78	15,54	15,81	10,60	16,22	49,20
Tunja	0,00	4,63	18,37	14,90	25,07	28,55	37,13
Barrancabermeja	0,00	0,17	11,90	16,62	6,20	12,30	43,93
San Juan de Girón	0,00	13,20	19,51	11,03	21,17	28,73	34,40
Apartadó	0,00	16,64	33,28	47,07	33,28	49,92	66,57
Turbo	34,50	34,50	34,50	-	34,50	34,50	34,50
Maicao	4,40	24,70	33,18	25,05	45,00	47,57	50,13
Piedecuesta	0,00	1,25	10,92	12,84	8,77	15,23	34,70
Yopal	0,00	6,85	13,70	19,37	13,70	20,55	27,40
Ipiales	7,83	7,83	7,83	-	7,83	7,83	7,83
Fusagasugá	12,10	13,48	14,85	3,89	14,85	16,23	17,60
Facatativá	0,00	0,00	0,00	-	0,00	0,00	0,00
Cartago	15,47	18,93	24,22	9,79	22,40	28,60	34,80
Chía	0,00	0,00	10,63	13,31	1,77	22,20	34,60
Pitalito	1,50	1,50	1,50	-	1,50	1,50	1,50
Zipaquirá	0,00	0,00	9,43	13,55	2,87	13,06	34,47
Total	0,00	3,12	18,11	17,80	12,67	29,15	83,17

Tabla A.35: Medidas de posición para la diferencia en meses entre el primer y último tweet de cada usuario individualmente, para cada ciudad de Colombia. Mostramos el mínimo, el primer cuartil (1Q), la media, la desviación típica, la mediana, el tercer cuartil (3Q) y el máximo.

España (15 ciudades)							
Ciudad	Min	1Q	Media	SDev	Mediana	3Q	Max
Madrid	0,00	0,93	15,70	16,76	10,17	25,97	102,40
Barcelona	0,00	0,93	17,09	19,34	8,70	26,60	76,80
Valencia	0,00	2,47	16,31	17,20	10,60	23,68	83,00
Sevilla	0,00	0,95	13,91	15,03	9,87	20,30	85,67
Zaragoza	0,00	1,93	14,50	14,62	11,12	20,51	65,77
Málaga	0,00	1,08	15,99	15,69	13,00	26,10	67,87
Murcia	0,00	0,38	13,15	14,24	10,20	17,75	53,20
Mallorca	0,00	1,90	15,36	14,41	14,22	26,90	53,10
Gran Canaria	0,00	2,23	17,41	20,71	12,13	22,33	82,57
Bilbao	0,00	0,00	12,78	15,60	5,80	21,94	63,00
Alicante	0,00	2,50	15,02	14,04	11,73	24,33	51,97
Córdoba	0,00	2,50	16,05	15,14	12,92	26,11	50,47

Valladolid	0,00	0,43	11,75	13,67	6,87	18,60	53,03
Vigo	0,00	1,70	13,21	13,11	9,70	19,00	57,87
Gijón	0,00	2,66	15,57	13,69	16,62	19,75	46,27
Total	0,00	1,10	15,44	16,51	10,17	24,50	102,40

Tabla A.36: Medidas de posición para la diferencia en meses entre el primer y último tweet de cada usuario individualmente, para cada ciudad de España. Mostramos el mínimo, el primer cuartil (1Q), la media, la desviación típica, la mediana, el tercer cuartil (3Q) y el máximo.

México (15 ciudades)							
Ciudad	Min	1Q	Media	SDev	Mediana	3Q	Max
México D. F.	0,00	0,00	14,16	16,91	7,57	22,22	78,07
Ecatepec	0,00	0,05	12,34	12,72	7,72	17,62	39,23
Guadalajara	0,00	0,43	16,40	19,15	7,97	26,83	78,50
Puebla de Zaragoza	0,00	2,60	15,54	17,40	10,20	21,23	74,27
Juárez	0,00	0,33	20,14	24,05	12,55	35,43	69,57
Tijuana	0,00	2,03	16,17	20,36	7,53	21,50	68,87
León	0,00	0,69	19,73	19,11	16,23	29,90	68,50
Zapopan	0,00	0,00	14,67	19,58	7,28	20,81	65,77
Monterrey	0,00	0,00	11,86	14,61	6,37	18,56	76,53
Nezahualcóyotl	0,00	0,00	14,16	16,49	8,50	22,00	74,67
Chihuahua	0,00	0,00	18,04	20,98	9,83	30,07	67,17
Naucalpan de Juárez	0,00	0,28	17,67	19,26	10,80	27,80	81,83
Mérida	0,00	0,00	13,84	16,34	8,42	20,47	67,67
San Luis Potosí	0,00	0,00	13,14	18,70	4,67	17,31	67,80
Aguascalientes	0,00	6,45	22,30	21,27	17,57	31,29	72,40
Total	0,00	0,00	14,97	17,61	7,95	23,71	81,83

Tabla A.37: Medidas de posición para la diferencia en meses entre el primer y último tweet de cada usuario individualmente, para cada ciudad de México. Mostramos el mínimo, el primer cuartil (1Q), la media, la desviación típica, la mediana, el tercer cuartil (3Q) y el máximo.

Perú (35 ciudades)							
Ciudad	Min	1Q	Media	SDev	Mediana	3Q	Max
Lima	0,00	2,90	17,53	19,06	10,83	26,00	79,87
Arequipa	0,00	3,80	24,38	21,94	21,60	32,00	75,87
Trujillo	0,00	4,66	18,92	18,90	13,63	29,52	77,20
Chiclayo	0,00	7,58	21,87	16,61	20,13	33,30	51,80
Lambayeque	0,00	0,00	9,60	10,47	11,20	11,43	25,37

Iquitos	0,00	4,57	20,31	18,59	14,00	39,85	52,37
Piura	0,00	0,33	22,66	22,86	16,40	38,30	68,23
Cusco	0,00	3,88	17,98	19,16	13,00	26,90	78,97
Chimbote	0,00	0,05	8,49	10,00	4,37	13,41	26,27
Huancayo	0,00	2,80	15,78	20,54	6,27	19,30	59,77
Tacna	0,00	2,43	10,99	10,55	5,70	18,30	27,00
Juliaca	12,63	13,28	13,93	1,84	13,93	14,58	15,23
Ica	0,00	0,00	16,17	17,23	11,27	32,80	41,27
Cajamarca	0,00	3,48	7,44	6,39	7,68	11,65	14,40
Pucallpa	9,33	9,33	9,33	-	9,33	9,33	9,33
Sullana	0,00	0,45	10,48	16,53	3,52	13,55	34,90
Ayacucho	5,73	5,73	5,73	-	5,73	5,73	5,73
Huánuco	5,53	6,91	12,36	6,99	12,00	17,45	19,90
Huacho	3,60	14,80	18,72	13,10	26,00	26,28	26,57
Tarapoto	0,00	0,00	5,84	8,24	0,17	7,50	22,33
Puno	0,00	0,63	1,25	1,77	1,25	1,88	2,50
Huaraz	0,00	0,00	0,90	1,80	0,00	0,90	3,60
Tumbes	9,03	15,32	21,60	17,77	21,60	27,88	34,17
Pisco	0,00	0,00	0,00	0,00	0,00	0,00	0,00
San Vicente de Cañete	0,00	0,00	21,62	31,27	10,08	31,70	66,30
Puerto Maldonado	38,87	38,87	38,87	-	38,87	38,87	38,87
Barranca	48,10	48,10	48,10	-	48,10	48,10	48,10
Talara	0,20	0,20	0,20	-	0,20	0,20	0,20
Ilo	27,37	27,37	27,37	-	27,37	27,37	27,37
Chulucanas	1,13	1,13	1,13	-	1,13	1,13	1,13
Tingo María	3,03	3,03	3,03	-	3,03	3,03	3,03
Mala	10,23	10,23	10,23	-	10,23	10,23	10,23
Pacasmayo	10,17	10,17	10,17	-	10,17	10,17	10,17
Tarma	0,00	12,72	25,45	35,99	25,45	38,17	50,90
Guadalupe	0,00	0,00	0,00	-	0,00	0,00	0,00
Total	0,00	2,80	17,47	18,86	10,87	26,33	79,87

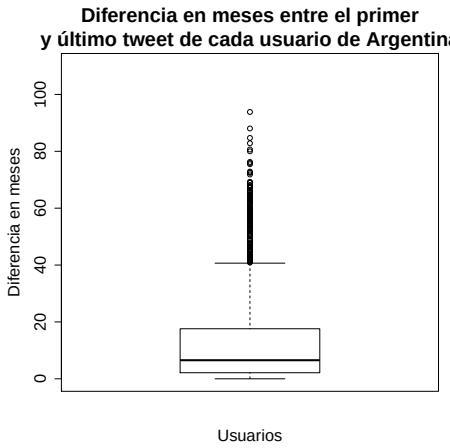
Tabla A.38: Medidas de posición para la diferencia en meses entre el primer y último tweet de cada usuario individualmente, para cada ciudad de Perú. Mostramos el mínimo, el primer cuartil (1Q), la media, la desviación típica, la mediana, el tercer cuartil (3Q) y el máximo.

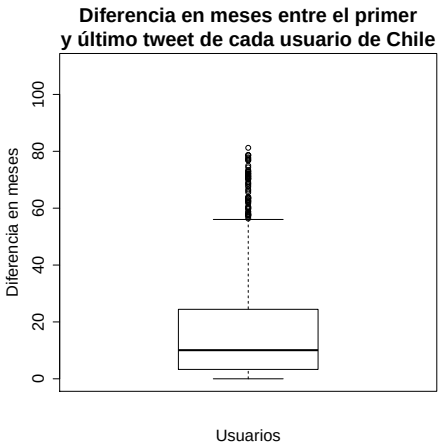
Venezuela (15 ciudades)							
Ciudad	Min	1Q	Media	SDev	Mediana	3Q	Max
Caracas	0,00	2,50	20,24	21,39	12,20	32,12	76,27
Maracaibo	0,00	4,32	20,37	19,55	15,15	28,68	75,63
Barquisimeto	0,00	3,80	19,53	19,67	14,43	30,83	73,00
Maracay	0,00	0,97	19,18	21,15	9,90	32,37	73,57

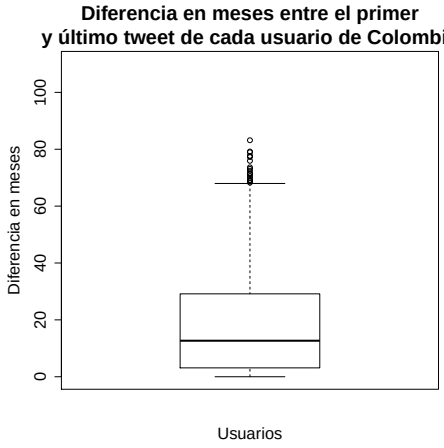
Guayana	0,00	1,07	22,63	20,00	22,25	39,54	58,97
San Cristóbal	0,00	10,12	26,38	20,81	22,45	36,75	71,47
Maturín	0,00	0,50	18,16	20,62	12,80	28,07	74,10
Bolívar	0,00	0,96	17,35	19,59	10,07	24,50	67,27
Cumaná	0,00	3,40	21,84	21,71	16,53	31,00	69,73
Barinas	0,00	2,04	20,60	18,63	17,43	32,99	67,63
Cabimas	0,00	1,00	17,70	20,80	7,57	27,13	62,03
Punto Fijo	0,00	0,85	10,30	11,61	6,67	16,30	49,97
Puerto La Cruz	0,00	2,27	19,23	22,43	9,47	26,88	68,03
Guarenas	0,00	3,33	20,19	19,79	13,10	31,80	68,60
Barcelona	0,00	0,37	16,15	18,24	11,47	23,53	68,13
Total	0,00	2,33	19,94	20,43	13,37	31,28	76,27

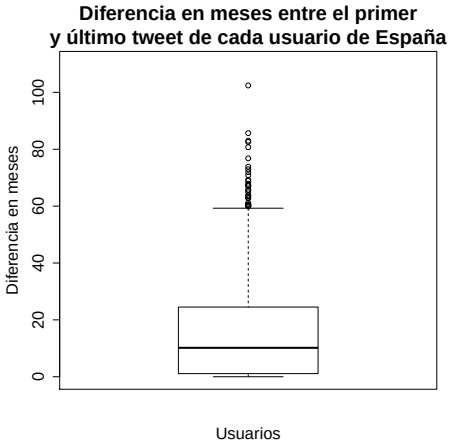
Tabla A.39: Medidas de posición para la diferencia en meses entre el primer y último tweet de cada usuario individualmente, para cada ciudad de Venezuela. Mostramos el mínimo, el primer cuartil (1Q), la media, la desviación típica, la mediana, el tercer cuartil (3Q) y el máximo.

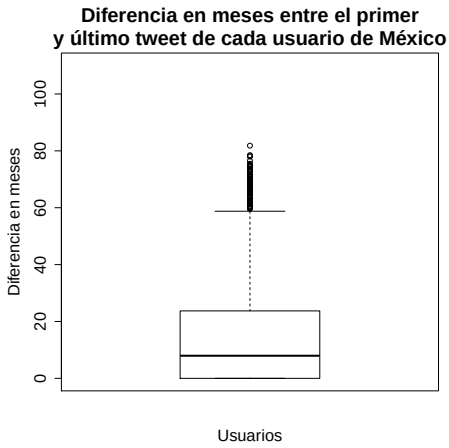
Para visualizar estos datos más claramente recurrimos a la Figura A.1. Estas figuras contienen un diagrama de caja por país, mostrando gráficamente los resultados por país de las Tablas A.33–A.39. Llama la atención que todos los países tienen su mínimo en cero. Esto ocurre porque en todos los países hay al menos un usuario con único tweet, para los cuales la diferencia entre las fechas de sus tweets es 0. Este caso se lleva al extremo en México, donde hay tantos usuarios con un solo tweet que el primer cuartil también toma valor 0. Las medianas toman valores entre los 6 y 15 meses aproximadamente, y siempre están ubicadas más cerca del primer cuartil que del tercero. Esto es una consecuencia de hecho de que los tweets se descargan en orden temporal desde los más recientes. Las fechas finales siempre se ubican cerca del cambio de año de 2015 a 2016. Las fechas iniciales están mucho más difusas, puesto que dependen de la cantidad de tweets que publique cada usuario y en qué intervalo de tiempo. Por ejemplo, hay usuarios que en pocos meses escriben más de 1000 tweets mientras que otros llevan años registrados y no llegan a esa cantidad. Esto provoca la aparición de datos atípicos en todos los países, a partir de los 60 o 70 meses. En el caso de Venezuela hay pocos datos anómalos.

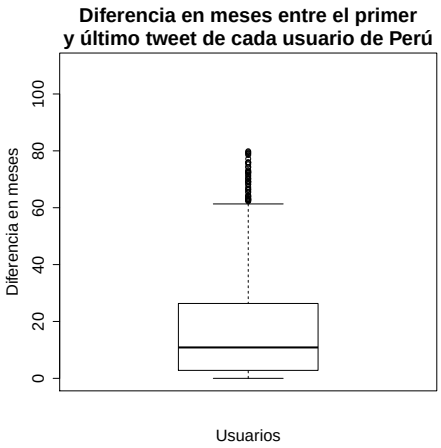

Diferencia en meses entre el primer
y último tweet de cada usuario de Argentina
Diferencia en meses
0
20
40
60
80
100
Usuarios


Diferencia en meses entre el primer
y último tweet de cada usuario de Chile
Diferencia en meses
0
20
40
60
80
100
Usuarios


Diferencia en meses entre el primer
y último tweet de cada usuario de Colombia
Diferencia en meses
0
20
40
60
80
100
Usuarios


Diferencia en meses entre el primer
y último tweet de cada usuario de España
Diferencia en meses
0
20
40
60
80
100
Usuarios


Diferencia en meses entre el primer
y último tweet de cada usuario de México
Diferencia en meses
0
20
40
60
80
100
Usuarios


Diferencia en meses entre el primer
y último tweet de cada usuario de Perú
Diferencia en meses
0
20
40
60
80
100
Usuarios

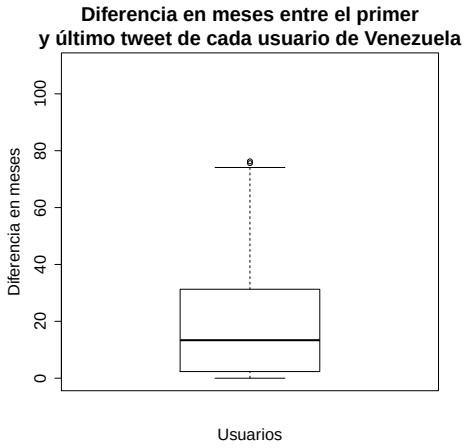


Figura A.1: Diferencia en meses entre el primer y el último tweet de cada usuario, para todos los países: Argentina, Chile, Colombia, España, México, Perú y Venezuela. Mostramos el mínimo, el máximo, la mediana, los cuartiles y los datos anómalos.

A.2.3 Corpus Final

Tras el proceso de recolección geográfica, la recuperación de *Timelines* y el proceso de refinamiento descrito en la sección 3.4, hemos escogido al azar 650 usuarios de cada país con entre 500 y 1.000 tweets publicados entre 2013 y 2016, excluyendo aquellos usuarios recuperados en ciudades cercanas a las fronteras. Este conjunto de usuarios (45.50) en total constituye nuestro corpus final, el cual evaluamos con la metodología descrita en el Capítulo 4. En esta sección desglosamos el corpus final a nivel de ciudad para analizarlo desde el punto de vista cuantitativo y temporal. Las Tablas A.40–A.46 muestran la cantidad de usuarios, tweets, y la proporción de tweets por usuario para cada ciudad. También mostramos la longitud media en palabras y caracteres del corpus final. Vemos que en general el número de tweets conservados para cada ciudad es proporcional al número de descargados. Obtenemos más usuarios de las capitales que del resto. Sin embargo, en la mayoría de países conservamos bastantes usuarios de distintas partes de la geografía. Por ejemplo, en Argentina 242 de 650 usuarios son de Buenos Aires (37,23 %), en España 247 de 650 son de Madrid (38,00 %), o en Colombia 180 de 650 son de Bogotá (27,69 %). En cuanto a México, el número de usuarios de la capital es de 307 de 650 (47,23), En el caso de Perú y Chile, la mayoría de tweets que conservamos sí que son de la capital, 497 (76,46 %) y 471 (72,46 %) respectivamente. El corpus cuenta con 215 de 650 usuarios de Caracas, siendo el 33,08 % de los usuarios de Venezuela. En cuanto a la proporción de tweets por usuario, observamos que en la mayoría de casos superamos los 800. Esta proporción sólo es menor en ciudades que recuperan pocos usuarios. Por ejemplo Apartadó (Colombia) tiene un único usuario con 600 tweets; o Gran Canaria, que con 13 usuarios, presenta una media de 832,54 tweets por usuario. En cuanto a las longitudes de los tweets no varían de forma significativa respecto a lo que hemos expuesto en las secciones anteriores.

Argentina (12 ciudades)					
País	Usuarios	Tweets (*timelines*)	Tweets/ usuario	Longitud media Palabras	Caracteres
Buenos Aires	242	212.308	877,31	11,61	71,77
Córdoba	87	75.016	862,25	9,58	56,91
Rosario	40	34.137	853,42	10,91	65,11
Mendoza	25	21.702	868,08	9,83	57,21
La Plata	57	50.062	878,28	10,15	60,78
Mar del Plata	39	33.849	867,92	9,78	58,80
San Miguel de Tucumán	18	15.121	840,06	9,08	50,51
Salta	10	8.448	844,80	12,47	83,37
Santa Fe	41	35.803	873,24	9,26	52,48
Bahía Blanca	24	21.510	896,25	9,48	52,48
Partido de Vicente López	54	46.868	867,93	11,31	71,14
San Juan	13	11.289	868,38	8,98	52,88
Total	650	566.113	870,94	10,63	64,32

Tabla A.40: Estadísticas del corpus final: número de usuarios, tweets, tweets por usuario y longitud media de los tweets en palabras y caracteres, para cada ciudad de Argentina.

Chile (18 ciudades)					
País	Usuarios	Tweets (*timelines*)	Tweets/ usuario	Longitud media Palabras	Caracteres
Santiago de Chile	471	413.159	877,20	11,73	78,24
Gran Concepción	13	10.876	836,62	11,89	78,05
Gran Valparaíso	60	53.295	888,25	11,30	73,15
Gran Temuco	11	9.355	850,45	12,30	79,66
Antofagasta	17	14.212	836,00	12,27	82,62
Talca	16	14.506	906,62	11,40	71,98
Los Ángeles	5	4.511	902,20	9,88	63,61
Copiapó	5	4.304	860,80	12,54	79,45
Valdivia	10	8.514	851,40	12,02	76,92
Osorno	5	4.135	827,00	10,95	70,59
Curicó	11	9.755	886,82	12,43	78,65

Calama	3	2.785	928,33	10,98	75,98
Punta Arenas	13	11.390	876,15	11,25	74,80
Melipilla	2	1.679	839,50	9,80	64,54
Ovalle	2	1.718	859,00	15,83	107,62
San Fernando	1	850	850,00	9,59	65,55
Talagante	1	839	839,00	11,54	87,48
Los Andes	4	3.307	826,75	12,78	75,87
Total	650	569.190	875,68	11,71	77,51

Tabla A.41: Estadísticas del corpus final: número de usuarios, tweets, tweets por usuario y longitud media de los tweets en palabras y caracteres, para cada ciudad de Chile.

Colombia (36 ciudades)					
País	Usuarios	Tweets (*timelines*)	Tweets/ usuario	Longitud media	
				Palabras	Caracteres
Bogotá	180	154.021	855,67	12,31	81,52
Medellín	96	81.850	852,60	12,68	85,12
Cali	59	50.801	861,03	12,42	83,41
Barranquilla	64	54.995	859,30	11,75	76,85
Cartagena de Indias	41	35.983	877,63	12,28	81,87
Soledad	15	12.619	841,27	11,37	71,32
Ibagué	23	19.909	865,61	11,44	71,74
Bucaramanga	19	17.089	899,42	12,18	80,01
Soacha	15	13.150	876,67	11,78	70,04
Villavicencio	11	9.176	834,18	12,65	84,96
Pereira	13	11.276	867,38	11,36	73,43
Bello	8	7.100	887,50	10,83	71,11
Valledupar	28	24.686	881,64	12,78	86,56
Montería	11	9.799	890,82	12,66	86,88
San Juan de Pasto	2	1.687	843,50	15,40	106,29
Manizales	3	2.503	834,33	8,28	50,97
Neiva	5	4.516	903,20	12,09	76,85
Armenia	3	2.450	816,67	13,05	84,50
Popayán	2	1.768	884,00	11,35	74,96
Sincelejo	5	4.427	885,40	12,03	73,56
Itaguí	8	6.897	862,12	11,43	78,09
Riohacha	1	956	956,00	14,62	93,10
Floridablanca	3	2.672	890,67	9,27	54,85
Envigado	8	6.929	866,12	11,92	82,31
Dosquebradas	5	4.194	838,80	8,91	52,62

Tunja	2	1.705	852,50	14,47	90,17
Barrancabermeja	2	1.481	740,50	7,70	45,83
San Juan de Girón	7	5.895	842,14	12,01	83,44
Apartadó	1	600	600,00	14,75	104,10
Turbo	1	557	557,00	20,07	123,43
Piedecuesta	2	1.697	848,50	9,84	66,53
Yopal	1	831	831,00	11,72	81,24
Fusagasugá	1	882	882,00	10,11	57,34
Cartago	1	910	910,00	12,25	70,29
Chía	2	1.725	862,50	14,10	90,31
Zipaquirá	2	1.883	941,50	11,36	69,84
Total	650	559.619	860,95	12,17	80,34

Tabla A.42: Estadísticas del corpus final: número de usuarios, tweets, tweets por usuario y longitud media de los tweets en palabras y caracteres, para cada ciudad de Colombia.

España (14 ciudades)					
País	Usuarios	Tweets (*timelines*)	Tweets/ usuario	Longitud media	
				Palabras	Caracteres
Madrid	247	213.038	862,50	12,71	83,67
Barcelona	82	71.185	868,11	13,51	90,37
Valencia	51	44.710	876,67	13,05	86,37
Sevilla	55	47.261	859,29	11,85	77,41
Zaragoza	27	23.861	883,74	13,21	89,54
Málaga	45	37.618	835,96	11,62	73,39
Murcia	22	18.620	846,36	12,38	82,94
Mallorca	13	11.234	864,15	12,39	84,82
Gran Canaria	13	10.823	832,54	12,45	81,63
Bilbao	20	16.815	840,75	11,58	75,86
Alicante	29	24.776	854,34	11,58	77,53
Córdoba	21	17.692	842,48	14,61	96,34
Valladolid	15	12.487	832,47	13,62	91,37
Gijón	10	8.133	813,30	14,66	91,85
Total	650	558.253	858,85	12,72	83,91

Tabla A.43: Estadísticas del corpus final: número de usuarios, tweets, tweets por usuario y longitud media de los tweets en palabras y caracteres, para cada ciudad de España.

México (13 ciudades)					
País	Usuarios	Tweets (*timelines*)	Tweets/ usuario	Longitud media Palabras	Caracteres
México D. F.	307	270.094	879,79	11,84	77,13
Ecatepec	6	5.374	895,67	11,08	68,4
Guadalajara	96	82.914	863,69	11,66	76,42
Puebla de Zaragoza	38	33.365	878,03	12,26	77,76
León	19	15.995	841,84	11,71	76,97
Zapopan	9	7.681	853,44	12,58	85,45
Monterrey	48	42.155	878,23	10,60	69,64
Nezahualcóyotl	23	19.722	857,48	11,92	75,00
Chihuahua	1	518	518,00	12,69	86,59
Naucalpan de Juárez	67	58.223	869,00	12,10	77,82
Mérida	22	19.315	877,95	11,06	73,52
San Luis Potosí	7	6.044	863,43	12,63	81,61
Aguascalientes	7	6.334	904,86	11,03	73,01
Total	650	567.734	873,44	11,75	76,42

Tabla A.44: Estadísticas del corpus final: número de usuarios, tweets, tweets por usuario y longitud media de los tweets en palabras y caracteres, para cada ciudad de México.

Perú (24 ciudades)					
País	Usuarios	Tweets (*timelines*)	Tweets/ usuario	Longitud media Palabras	Caracteres
Lima	497	433.776	872,79	11,27	72,56
Arequipa	16	13.822	863,88	11,54	75,98
Trujillo	47	39.083	831,55	10,83	66,78
Chiclayo	11	9.459	859,91	11,68	75,14
Lambayeque	3	2.550	850,00	12,56	73,06
Iquitos	11	9.404	854,91	11,25	72,40
Piura	9	7.470	830,00	13,10	85,62
Cusco	20	17.512	875,60	10,97	72,99
Chimbote	2	1.692	846,00	13,51	81,94
Huancayo	4	3.523	880,75	10,92	67,33
Juliaca	2	1.884	942,00	12,47	79,09
Ica	5	4.399	879,80	9,68	66,46
Cajamarca	3	2.544	848,00	12,77	83,59
Sullana	1	926	926,00	9,78	59,31

Ayacucho	1	891	891,00	8,51	48,70
Huánuco	3	2.592	864,00	10,94	66,96
Huacho	3	2.563	854,33	10,05	66,55
Tarapoto	6	5.326	887,67	12,24	71,46
Puno	1	616	616,00	7,23	46,82
Huaraz	1	904	904,00	8,75	53,17
San Vicente de Cañete	1	528	528,00	12,69	81,72
Ilo	1	981	981,00	8,52	58,76
Mala	1	877	877,00	12,15	82,77
Pacasmayo	1	881	881,00	13,92	96,64
Total	650	564.203	868,00	11,26	72,35

Tabla A.45: Estadísticas del corpus final: número de usuarios, tweets, tweets por usuario y longitud media de los tweets en palabras y caracteres, para cada ciudad de Perú.

Venezuela (14 ciudades)					
País	Usuarios	Tweets (*timelines*)	Tweets/ usuario	Longitud media	
				Palabras	Caracteres
Caracas	215	183.025	851,28	13,57	90,76
Maracaibo	136	116.824	859,00	13,43	91,75
Barquisimeto	67	55.591	829,72	14,35	96,26
Maracay	57	48.014	842,35	13,62	91,17
Guayana	13	11.302	869,38	12,38	83,83
Barcelona	15	12.771	851,40	14,08	95,49
Maturín	28	22.729	811,75	14,66	97,18
Bolívar	19	16.604	873,89	12,81	89,45
Cumaná	13	11.218	862,92	13,90	94,13
Barinas	15	12.855	857,00	14,53	99,75
Cabimas	14	11.960	854,29	10,88	67,26
Punto Fijo	18	16.204	900,22	11,15	70,63
Puerto la Cruz	19	15.903	837,00	13,99	92,53
Guarenas	21	17.978	856,10	13,45	89,59
Total	650	552.978	850,74	13,54	90,94

Tabla A.46: Estadísticas del corpus final: número de usuarios, tweets, tweets por usuario y longitud media de los tweets en palabras y caracteres, para cada ciudad de Venezuela.

Las Tablas A.47–A.53 muestran información relativa a las fechas de publicación de los tweets en el corpus final, desglosado para cada ciudad de cada país. Mostramos la fecha del primer tweet y último tweet, independientemente del usuario, y la diferencia en meses. Tras aplicar el filtrado temporal, observamos que para aquellas ciudades con suficiente cantidad de tweets la diferencia es de tres años, empezando en 2013 y terminando en 2016.

Argentina (12 ciudades)			
Ciudad	Primer tweet	Último tweet	Diferencia (meses)
Buenos Aires	02-01-2013	27-12-2015	36,30
Córdoba	01-01-2013	27-12-2015	36,30
Rosario	02-01-2013	26-12-2015	36,27
Mendoza	16-01-2013	26-12-2015	35,77
La Plata	23-03-2013	26-12-2015	33,60
Mar del Plata	01-01-2013	26-12-2015	36,27
San Miguel de Tucumán	01-09-2014	25-12-2015	16,00
Salta	15-07-2013	24-12-2015	29,70
Santa Fe	05-07-2013	25-12-2015	30,10
Bahía Blanca	01-09-2014	26-12-2015	16,00
Partido de Vicente López	02-01-2013	26-12-2015	36,27
San Juan	01-01-2013	25-12-2015	36,27
Total	01-01-2013	27-12-2015	36,30

Tabla A.47: Estadísticas relativas a las fechas de los tweets a nivel de ciudad, para Argentina. Mostramos la fecha del primer, del último tweet y la diferencia en meses.

Chile (18 ciudades)			
Ciudad	Primer tweet	Último tweet	Diferencia (meses)
Santiago de Chile	2013-01-01	2015-12-31	36,47
Gran Concepción	2013-01-04	2015-12-31	36,37
Gran Valparaíso	2013-01-01	2015-12-31	36,47
Gran Temuco	2013-01-01	2015-12-31	36,47
Antofagasta	2013-01-01	2015-12-31	36,47
Talca	2013-03-02	2015-12-31	34,47
Los Ángeles	2015-05-18	2015-12-31	7,57
Copiap ó	2013-09-11	2015-12-31	28,03
Valdiv ia	2013-01-01	2015-12-31	36,47
Osorno	2013-01-03	2015-12-31	36,37
Curicó	2013-09-02	2015-12-31	28,33
Calama	2013-07-11	2015-12-31	30,07
Punta Arenas	2013-01-01	2015-12-31	36,47
Melipilla	2015-02-24	2015-12-31	10,33

Ovalle	2015-05-26	2015-12-29	7,20
San Fernando	2013-09-06	2015-12-27	28,07
Talagante	2015-02-11	2015-12-31	10,77
Los Andes	2014-06-24	2015-12-31	18,50
Total	2013-01-01	2015-12-31	36,47

Tabla A.48: Estadísticas relativas a las fechas de los tweets a nivel de ciudad, para Chile. Mostramos la fecha del primer, del último tweet y la diferencia en meses.

Colombia (36 ciudades)			
Ciudad	Primer tweet	Último tweet	Diferencia (meses)
Bogotá	01-01-2013	31-12-2015	36,47
Medellín	01-01-2013	31-12-2015	36,47
Cali	01-01-2013	31-12-2015	36,47
Barranquilla	01-01-2013	31-12-2015	36,47
Cartagena de Indias	02-01-2013	31-12-2015	36,43
Soledad	06-01-2013	31-12-2015	36,30
Ibagué	06-01-2013	31-12-2015	36,30
Bucaramanga	01-01-2013	31-12-2015	36,47
Soacha	22-02-2013	31-12-2015	34,73
Villavicencio	02-01-2013	31-12-2015	36,43
Pereira	01-01-2013	31-12-2015	36,47
Bello	02-01-2013	30-12-2015	36,40
Valledupar	01-01-2013	31-12-2015	36,47
Montería	01-01-2013	31-12-2015	36,47
San Juan de Pasto	26-03-2015	31-12-2015	9,33
Manizales	04-12-2013	31-12-2015	25,23
Neiva	08-11-2013	31-12-2015	26,10
Armenia	01-01-2013	31-12-2015	36,47
Popayán	24-01-2014	31-12-2015	23,53
Sincelejo	11-02-2013	31-12-2015	35,10
Itaguí	14-05-2013	31-12-2015	32,03
Riohacha	04-02-2013	31-12-2015	35,30
Floridablanca	06-01-2015	31-12-2015	11,97
Envigado	22-05-2013	31-12-2015	31,77
Dosquebradas	02-07-2014	31-12-2015	18,23
Tunja	02-10-2013	31-12-2015	27,30
Barrancabermeja	02-07-2015	31-12-2015	6,07
San Juan de Girón	23-04-2013	31-12-2015	32,73
Apartadó	07-01-2013	30-12-2015	36,23
Turbo	05-03-2013	31-12-2015	34,37
Piedecuesta	02-04-2015	31-12-2015	9,10
Yopal	04-10-2013	31-12-2015	27,27

Fusagasugá	05-01-2015	31-12-2015	12,00
Cartago	03-03-2014	30-12-2015	22,23
Chía	08-03-2014	31-12-2015	22,10
Zipaquirá	27-02-2013	31-12-2015	34,57
Total	01-01-2013	31-12-2015	36,47

Tabla A.49: Estadísticas relativas a las fechas de los tweets a nivel de ciudad, para Colombia. Mostramos la fecha del primer, del último tweet y la diferencia en meses.

España (14 ciudades)			
Ciudad	Primer tweet	Último tweet	Diferencia (meses)
Madrid	01-01-2013	29-12-2015	36,40
Barcelona	01-01-2013	29-12-2015	36,40
Valencia	03-01-2013	29-12-2015	36,30
Sevilla	01-01-2013	29-12-2015	36,40
Zaragoza	19-01-2013	29-12-2015	35,77
Málaga	01-01-2013	29-12-2015	36,40
Murcia	01-01-2013	29-12-2015	36,37
Mallorca	02-01-2013	29-12-2015	36,37
Gran Canaria	01-01-2013	29-12-2015	36,40
Bilbao	04-01-2013	28-12-2015	36,27
Alicante	01-01-2013	29-12-2015	36,40
Córdoba	01-01-2013	28-12-2015	36,37
Valladolid	01-01-2013	28-12-2015	36,37
Gijón	01-01-2013	29-12-2015	36,40
Total	01-01-2013	29-12-2015	36,40

Tabla A.50: Estadísticas relativas a las fechas de los tweets a nivel de ciudad, para España. Mostramos la fecha del primer, del último tweet y la diferencia en meses.

México (13 ciudades)			
Ciudad	Primer tweet	Último tweet	Diferencia (meses)
México D. F.	01-01-2013	30-12-2015	36,43
Ecatepec	01-01-2013	30-12-2015	36,43
Guadalajara	01-01-2013	30-12-2015	36,43
Puebla de Zaragoza	06-01-2013	30-12-2015	36,23
León	12-02-2013	30-12-2015	35,03
Zapopan	04-01-2013	30-12-2015	36,33
Monterrey	01-01-2013	30-12-2015	36,43
Nezahualcóyotl	01-01-2013	30-12-2015	36,43

Chihuahua	18-02-2015	11-12-2015	9,83
Naucalpan de Juárez	01-01-2013	30-12-2015	36,43
Mérida	01-01-2013	30-12-2015	36,43
San Luis Potosí	03-01-2013	30-12-2015	36,37
Aguascalientes	19-05-2013	30-12-2015	31,83
Total	01-01-2013	30-12-2015	36,43

Tabla A.51: Estadísticas relativas a las fechas de los tweets a nivel de ciudad, para México. Mostramos la fecha del primer, del último tweet y la diferencia en meses.

Perú (24 ciudades)			
Ciudad	Primer tweet	Último tweet	Diferencia (meses)
Lima	01-01-2013	31-12-2015	36,47
Arequipa	01-01-2013	31-12-2015	36,47
Trujillo	01-01-2013	31-12-2015	36,47
Chiclayo	17-06-2013	31-12-2015	30,90
Lambayeque	06-12-2013	31-12-2015	25,17
Iquitos	04-01-2013	31-12-2015	36,37
Piura	01-01-2013	31-12-2015	36,47
Cusco	04-01-2013	31-12-2015	36,37
Chimbote	06-01-2015	31-12-2015	11,97
Huancayo	31-03-2015	31-12-2015	9,17
Juliaca	03-10-2014	31-12-2015	15,13
Ica	25-01-2013	31-12-2015	35,67
Cajamarca	29-10-2014	31-12-2015	14,23
Sullana	22-06-2015	31-12-2015	6,40
Ayacucho	13-07-2015	27-10-2015	3,53
Huánuco	08-09-2014	31-12-2015	15,97
Huacho	05-11-2013	31-12-2015	26,20
Tarapoto	25-02-2014	31-12-2015	22,47
Puno	06-11-2015	31-12-2015	1,83
Huaraz	23-09-2015	31-12-2015	3,30
San Vicente de Cañete	30-09-2013	19-10-2015	24,93
Ilo	03-10-2013	30-12-2015	27,27
Mala	18-03-2015	31-12-2015	9,57
Pacasmayo	17-03-2015	17-10-2015	7,10
Total	01-01-2013	31-12-2015	36,47

Tabla A.52: Estadísticas relativas a las fechas de los tweets a nivel de ciudad, para Perú. Mostramos la fecha del primer, del último tweet y la diferencia en meses.

Venezuela (14 ciudades)			
Ciudad	Primer tweet	Último tweet	Diferencia (meses)
Caracas	01-01-2013	31-12-2015	36,47
Maracaibo	01-01-2013	31-12-2015	36,47
Barquisimeto	01-01-2013	31-12-2015	36,47
Maracay	01-01-2013	31-12-2015	36,47
Guayana	26-01-2013	31-12-2015	35,60
Barcelona	01-01-2013	31-12-2015	36,47
Maturín	01-01-2013	31-12-2015	36,47
Bolívar	01-01-2013	31-12-2015	36,47
Cumaná	01-01-2013	31-12-2015	36,47
Barinas	07-01-2013	31-12-2015	36,27
Cabimas	15-04-2013	31-12-2015	32,97
Punto Fijo	27-02-2013	31-12-2015	34,53
Puerto La Cruz	01-01-2013	31-12-2015	36,47
Guarenas	03-01-2013	31-12-2015	36,40
Total	01-01-2013	31-12-2015	36,47

Tabla A.53: Estadísticas relativas a las fechas de los tweets a nivel de ciudad, para Venezuela. Mostramos la fecha del primer, del último tweet y la diferencia en meses.

En las Tablas A.54–A.60 mostramos las medidas de posición para la diferencia entre el primer y último tweet de cada usuario, para cada usuario individualmente. mostramos las medidas de posición para la diferencia entre el primer y último tweet de cada usuario, para cada usuario individualmente. Muestra la misma información que las Tablas A.33–A.39 pero sobre el corpus final. Comparando ambas tablas, vemos que el filtrado temporal ha reducido mucho la dispersión. Ahora el máximo se encuentra en los 36 meses, las medias por ciudad fluctúan menos y la desviación típica ahora es menor que la media. El mínimo ya no toma valor 0 en ningún caso, puesto que no conservamos ningún usuario con un único tweet. Los valores para el primer cuartil también son más estables. Éste oscila alrededor de 5 meses para las capitales, y varía de forma más aleatoria en las ciudades menos pobladas. El tercer cuartil ronda los 20 meses para las ciudades más pobladas.

Argentina (12 ciudades)							
Ciudad	Min	1Q	Media	SDev	Mediana	3Q	Max
Buenos Aires	0,30	4,93	12,88	10,06	10,62	18,31	36,20
Córdoba	0,80	2,55	8,97	8,36	6,30	13,13	36,03
Rosario	0,27	2,79	12,40	11,79	6,23	19,87	36,13
Mendoza	0,83	2,77	12,03	11,67	5,23	21,70	35,23
La Plata	0,33	2,10	8,83	8,64	6,23	12,60	33,27
Mar del Plata	0,40	2,05	7,97	10,64	3,53	8,27	36,20
San Miguel de Tucumán	1,47	3,67	7,75	4,56	7,60	10,97	15,90
Salta	1,33	11,57	15,46	8,35	14,92	18,63	29,37
Santa Fe	0,73	2,60	8,29	7,50	5,80	10,90	29,97
Bahía Blanca	0,63	1,83	4,60	4,08	3,30	5,03	16,00
Partido de Vicente López	0,67	4,08	10,95	9,18	8,18	14,64	36,17
San Juan	1,17	4,23	10,18	9,49	7,33	14,83	35,77
Total	0,27	3,07	10,73	9,64	7,45	15,38	36,20

Tabla A.54: Medidas de posición del corpus final, para la diferencia en meses entre el primer y último tweet de cada usuario individualmente, para cada ciudad de Argentina. Mostramos el mínimo, el primer cuartil (1Q), la media, la desviación típica, la mediana, el tercer cuartil (3Q) y el máximo.

Chile (18 ciudades)							
Ciudad	Min	1Q	Media	SDev	Mediana	3Q	Max
Santiago de Chile	0,23	4,67	13,73	10,85	11,13	20,30	36,43
Gran Concepción	0,53	0,90	8,35	10,00	3,40	13,83	35,70
Gran Valparaíso	0,27	3,08	12,03	10,69	9,00	17,58	36,23
Gran Temuco	4,53	8,22	17,95	10,80	16,80	25,15	36,47
Antofagasta	1,57	7,30	17,82	12,39	15,97	26,53	36,20
Talca	0,67	3,51	12,09	10,93	7,00	18,22	33,83
Los Ángeles	2,70	3,37	5,37	2,14	6,80	6,93	7,03
Copiapó	1,47	8,20	12,31	9,91	9,07	14,97	27,83
Valdivia	1,17	7,67	16,58	11,09	17,57	22,42	36,23
Osorno	4,13	8,63	20,16	14,42	18,27	33,40	36,37
Curicó	1,20	3,17	10,41	8,98	8,07	16,90	27,80
Calama	17,40	21,42	24,30	6,41	25,43	27,75	30,07
Punta Arenas	0,40	2,70	15,04	13,22	10,60	29,23	36,00
Melipilla	3,20	4,92	6,63	4,86	6,63	8,35	10,07

Ovalle	5,00	5,55	6,10	1,56	6,10	6,65	7,20
San Fernando	28,07	28,07	28,07	-	28,07	28,07	28,07
Talagante	10,77	10,77	10,77	-	10,77	10,77	10,77
Los Andes	3,27	3,59	8,91	7,13	6,93	12,25	18,50
Total	0,23	4,54	13,59	10,89	10,58	20,58	36,47

Tabla A.55: Medidas de posición del corpus final, para la diferencia en meses entre el primer y último tweet de cada usuario individualmente, para cada ciudad de Chile. Mostramos el mínimo, el primer cuartil (1Q), la media, la desviación típica, la mediana, el tercer cuartil (3Q) y el máximo.

Colombia (36 ciudades)							
Ciudad	Min	1Q	Media	SDev	Mediana	3Q	Max
Bogotá	0,10	5,06	15,80	12,04	13,37	27,23	36,40
Medellín	0,33	6,86	17,65	11,68	15,97	26,40	36,40
Cali	1,70	8,22	20,00	12,18	19,93	31,80	36,37
Barranquilla	1,23	6,36	17,11	12,14	13,38	29,41	36,37
Cartagena de Indias	0,63	8,73	16,49	9,87	17,53	23,73	36,27
Soledad	0,70	3,18	14,81	12,11	13,50	21,90	36,10
Ibagué	0,97	7,10	15,70	10,82	12,80	23,17	36,10
Bucaramanga	2,63	11,13	17,99	9,77	18,37	24,42	36,27
Soacha	0,60	2,22	10,35	9,89	7,63	17,05	34,50
Villavicencio	7,50	12,53	22,04	11,85	19,73	35,17	36,40
Pereira	0,53	13,30	19,56	10,82	17,77	28,63	36,23
Bello	2,30	4,18	13,28	12,84	8,57	17,74	36,23
Valledupar	1,83	7,26	15,15	11,34	11,48	18,52	36,43
Montería	0,63	1,38	11,97	13,29	8,03	15,00	36,47
San Juan de Pasto	7,73	8,13	8,52	1,11	8,52	8,91	9,30
Manizales	6,93	7,50	13,41	10,25	8,07	16,65	25,23
Neiva	10,47	19,10	20,79	6,35	23,43	24,90	26,07
Armenia	1,03	10,62	19,22	17,72	20,20	28,32	36,43
Popayán	4,43	9,20	13,97	13,48	13,97	18,73	23,50
Sincelejo	9,37	16,47	23,16	10,19	26,20	28,80	34,97
Itaguí	6,93	7,64	16,87	10,50	14,95	22,61	32,03
Riohacha	35,30	35,30	35,30	-	35,30	35,30	35,30
Floridablanca	2,20	2,67	5,77	5,39	3,13	7,55	11,97
Envigado	4,30	13,22	17,69	8,82	14,95	24,37	31,77
Dosquebradas	0,83	7,60	10,24	6,59	10,47	14,10	18,20
Tunja	2,23	8,48	14,73	17,68	14,73	20,98	27,23
Barrancabermeja	0,20	1,65	3,10	4,10	3,10	4,55	6,00
San Juan de Girón	5,57	13,13	20,09	10,69	16,90	29,68	32,53

Apartadó	36,23	36,23	36,23	-	36,23	36,23	36,23
Turbo	34,37	34,37	34,37	-	34,37	34,37	34,37
Piedecuesta	8,60	8,73	8,85	0,35	8,85	8,98	9,10
Yopal	27,27	27,27	27,27	-	27,27	27,27	27,27
Fusagasugá	12,00	12,00	12,00	-	12,00	12,00	12,00
Cartago	22,23	22,23	22,23	-	22,23	22,23	22,23
Chía	13,27	15,47	17,67	6,22	17,67	19,87	22,07
Zipaquirá	4,37	11,89	19,42	21,28	19,42	26,94	34,47
Total	0,10	6,26	16,73	11,62	14,52	26,92	36,47

Tabla A.56: Medidas de posición del corpus final, para la diferencia en meses entre el primer y último tweet de cada usuario individualmente, para cada ciudad de Colombia. Mostramos el mínimo, el primer cuartil (1Q), la media, la desviación típica, la mediana, el tercer cuartil (3Q) y el máximo.

España (14 ciudades)							
Ciudad	Min	1Q	Media	SDev	Mediana	3Q	Max
Madrid	0,20	6,78	16,52	11,15	14,43	26,07	36,37
Barcelona	0,73	3,71	14,25	11,61	11,95	23,28	36,33
Valencia	0,77	7,15	13,04	9,22	10,60	18,40	36,30
Sevilla	0,83	8,18	16,42	9,89	16,57	20,93	36,17
Zaragoza	1,23	2,43	12,27	10,29	10,00	19,48	35,73
Málaga	0,30	6,10	17,06	12,48	15,63	27,43	35,90
Murcia	0,50	8,82	17,02	10,22	13,92	24,22	36,30
Mallorca	1,80	5,27	17,52	13,09	17,80	29,87	36,03
Gran Canaria	2,50	9,33	17,99	11,37	14,90	22,33	36,33
Bilbao	0,67	2,14	15,92	13,11	13,53	27,67	35,83
Alicante	1,43	4,67	15,52	11,69	14,07	24,60	36,30
Córdoba	1,80	10,63	17,29	10,08	17,37	19,80	35,97
Valladolid	0,43	6,45	15,37	11,51	13,60	21,95	36,30
Gijón	3,30	10,77	17,56	11,45	16,65	21,67	36,30
Total	0,20	6,21	15,83	11,10	13,90	24,16	36,37

Tabla A.57: Medidas de posición del corpus final, para la diferencia en meses entre el primer y último tweet de cada usuario individualmente, para cada ciudad de España. Mostramos el mínimo, el primer cuartil (1Q), la media, la desviación típica, la mediana, el tercer cuartil (3Q) y el máximo.

México (13 ciudades)							
Ciudad	Min	1Q	Media	SDev	Mediana	3Q	Max
México D. F.	0,47	5,68	15,46	11,42	12,37	23,68	36,40
Ecatepec	3,23	7,58	15,64	14,27	7,72	25,38	36,40
Guadalajara	0,57	6,68	15,99	11,44	14,25	25,08	36,23

Puebla de Zaragoza	0,83	4,94	13,22	9,48	10,35	19,37	36,07
León	1,97	13,42	19,76	10,06	19,13	26,98	34,57
Zapopan	7,17	7,67	17,94	11,56	14,37	29,37	35,63
Monterrey	0,60	4,31	13,03	10,90	8,13	18,06	36,30
Nezahualcóyotl	1,47	7,03	16,70	10,77	15,43	26,75	36,33
Chihuahua	9,83	9,83	9,83	-	9,83	9,83	9,83
Naucalpan de Juárez	0,50	5,95	16,30	11,19	16,30	25,75	36,23
Mérida	2,20	9,42	16,73	11,25	14,70	19,92	36,43
San Luis Potosí	0,73	6,73	14,63	11,78	15,93	17,95	36,37
Aguascalientes	6,50	7,72	16,24	11,01	10,70	24,85	31,33
Total	0,47	5,83	15,56	11,19	13,07	24,67	36,43

Tabla A.58: Medidas de posición del corpus final, para la diferencia en meses entre el primer y último tweet de cada usuario individualmente, para cada ciudad de México. Mostramos el mínimo, el primer cuartil (1Q), la media, la desviación típica, la mediana, el tercer cuartil (3Q) y el máximo.

Perú (24 ciudades)							
Ciudad	Min	1Q	Media	SDev	Mediana	3Q	Max
Lima	0,20	4,73	13,54	10,34	11,27	20,63	36,43
Arequipa	2,67	4,36	19,05	12,07	23,30	28,92	36,00
Trujillo	0,47	5,75	15,39	11,12	13,90	19,92	36,43
Chiclayo	1,00	4,12	17,12	12,28	20,00	28,72	30,70
Lambayeque	9,80	10,17	15,16	8,65	10,53	17,83	25,13
Iquitos	3,60	5,23	17,66	13,93	14,00	32,40	36,17
Piura	3,17	8,87	19,61	12,71	17,57	32,57	36,47
Cusco	2,77	5,54	15,41	11,09	13,92	22,88	36,37
Chimbote	9,53	9,99	10,45	1,30	10,45	10,91	11,37
Huancayo	2,67	4,87	5,93	2,67	5,93	6,99	9,17
Juliaca	12,50	13,16	13,82	1,86	13,82	14,48	15,13
Ica	11,10	15,97	23,67	10,49	23,03	32,80	35,47
Cajamarca	4,63	7,53	8,96	3,81	10,43	11,12	11,80
Sullana	6,40	6,40	6,40	-	6,40	6,40	6,40
Ayacucho	3,53	3,53	3,53	-	3,53	3,53	3,53
Huánuco	4,90	5,80	9,19	5,94	6,70	11,33	15,97
Huacho	3,60	14,48	18,39	12,81	25,37	25,78	26,20
Tarapoto	2,80	4,90	11,17	8,08	9,23	17,29	22,33
Puno	1,83	1,83	1,83	-	1,83	1,83	1,83
Huaraz	3,30	3,30	3,30	-	3,30	3,30	3,30
San Vicente de Cañete	24,93	24,93	24,93	-	24,93	24,93	24,93

Ilo	27,27	27,27	27,27	-	27,27	27,27	27,27
Mala	9,57	9,57	9,57	-	9,57	9,57	9,57
Pacasmayo	7,10	7,10	7,10	-	7,10	7,10	7,10
Total	0,20	4,84	14,03	10,56	11,45	21,47	36,47

Tabla A.59: Medidas de posición del corpus final, para la diferencia en meses entre el primer y último tweet de cada usuario individualmente, para cada ciudad de Perú. Mostramos el mínimo, el primer cuartil (1Q), la media, la desviación típica, la mediana, el tercer cuartil (3Q) y el máximo.

Venezuela (14 ciudades)							
Ciudad	Min	1Q	Media	SDev	Mediana	3Q	Max
Caracas	0,30	5,73	16,57	12,04	14,43	27,08	36,37
Maracaibo	0,40	9,14	17,65	10,92	16,38	26,21	36,43
Barquisimeto	0,93	6,83	16,50	10,95	14,43	24,25	36,47
Maracay	0,30	5,47	17,26	12,60	13,80	29,90	36,00
Guayana	1,10	10,27	18,93	11,55	22,07	27,67	34,80
Barcelona	2,20	12,73	18,45	9,79	17,00	23,58	36,07
Maturín	0,43	10,81	19,16	11,77	16,80	31,11	35,97
Bolívar	0,87	7,45	16,16	12,48	12,77	26,67	35,93
Cumaná	2,57	12,70	16,79	10,16	15,73	18,00	36,00
Barinas	1,67	8,92	16,90	9,87	13,50	24,02	35,73
Cabimas	0,43	3,96	12,70	11,53	7,65	22,77	32,97
Punto Fijo	2,60	5,60	12,98	9,59	11,63	17,88	33,90
Puerto La Cruz	0,50	1,97	11,69	11,84	4,27	18,50	35,97
Guarenas	0,43	6,47	13,64	9,79	10,93	18,30	35,90
Total	0,30	6,42	16,63	11,47	14,98	26,17	36,47

Tabla A.60: Medidas de posición del corpus final, para la diferencia en meses entre el primer y último tweet de cada usuario individualmente, para cada ciudad de Venezuela. Mostramos el mínimo, el primer cuartil (1Q), la media, la desviación típica, la mediana, el tercer cuartil (3Q) y el máximo.

Bibliografía

[1] F. Rangel, P. Rosso, M. Moshe Koppel, E. Stamatatos, and G. Inches, "Overview of the author profiling task at PAN 2013," in *Forner P., Navigli R., Tufis D.(Eds.), Notebook Papers of CLEF 2013 LABs and Workshops. CEUR-WS.org*, vol. 1179, (2013).

[2] F. Rangel, P. Rosso, I. Chugur, M. Potthast, M. Trenkmann, B. Stein, B. Verhoeven, and W. Daelemans, "Overview of the 2nd author profiling task at PAN 2014," in *Cappellato L., Ferro N., Halvey M., Kraaij W. (Eds.) CLEF 2014 Labs and Workshops, Notebook Papers. CEUR-WS.org*, vol. 1180, (2014).

[3] F. Rangel, F. Celli, P. Rosso, M. Potthast, B. Stein, and W. Daelemans, "Overview of the 3rd author profiling task at PAN 2015," in *Cappellato L., Ferro N., Gareth J. and San Juan E. (Eds). CLEF 2015 Labs and Workshops, Notebook Papers. CEUR Workshop Proceedings, CEUR-WS.org*, vol. 1391, (2015).

[4] F. Rangel, P. Rosso, B. Verhoeven, W. Daelemans, M. Potthast, and B. Stein, "Overview of the 4th author profiling task at PAN 2016: cross-genre evaluations," *Working Notes Papers of the CLEF. CEUR-WS.org*, (2016).

[5] J. Tetreault, D. Blanchard, and A. Cahill, "A report on the first native language identification shared task," in *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications, BEA-8*, pp. 48–57, (2013).

[6] E. M. Gold, "Language identification in the limit," *Information and control*, vol. 10, no. 5, pp. 447–474, (1967).

[7] W. B. Cavnar, J. M. Trenkle, *et al.*, "N-gram-based text categorization," *Ann Arbor MI*, vol. 48113, no. 2, pp. 161–175, (1994).

[8] T. Dunning, "Statistical identification of language". *Computing Research Laboratory, New Mexico State University*, (1994).

[9] C. Kruengkrai, P. Srichaivattana, V. Sornlertlamvanich, and H. Isahara, "Language identification based on string kernels," in *IEEE International*

Symposium on Communications and Information Technology, 2005. ISCIT 2005, vol. 2, pp. 926–929, IEEE, (2005).

[10] M. Lui and T. Baldwin, "Cross-domain feature selection for language identification," in *In Proceedings of 5th International Joint Conference on Natural Language Processing*, pp. 553–561, (2011).

[11] B. Hughes, T. Baldwin, S. Bird, J. Nicholson, and A. Mackinlay, "Reconsidering language identification for written language resources," in *Proceedings of the 5th international conference on Language Resources and Evaluation, LREC 2006*, pp. 485–488, (2006).

[12] S. Carter, W. Weerkamp, and M. Tsagkias, "Microblog language identification: Overcoming the limitations of short, unedited and idiomatic text," *Language Resources and Evaluation*, vol. 47, no. 1, pp. 195–215, (2013).

[13] S. P. Corder, "The significance of learner's errors," *IRAL-International Review of Applied Linguistics in Language Teaching*, vol. 5, no. 1-4, pp. 161–170, (1967).

[14] M. Koppel, J. Schler, and K. Zigdon, "Determining an author's native language by mining a text for errors," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pp. 624–628, ACM, (2005).

[15] S. Granger, E. Dagneaux, F. Meunier, M. Paquot, *et al.*, "The international corpus of learner English. Handbook and CD-ROM," (2002).

[16] P. Tofighi, C. Köse, and L. Rouka, "Author's native language identification from web-based texts," *International Journal of Computer and Communication Engineering*, vol. 1, no. 1, pp. 47–50, (2012).

[17] D. Blanchard, J. Tetreault, D. Higgins, A. Cahill, and M. Chodorow, "TOEFL11: A corpus of non-native english," *ETS Research Report Series*, vol. 2013, no. 2, pp. i–15, (2013).

[18] J. Brooke and G. Hirst, "Using other learner corpora in the 2013 NLI shared task," in *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications, NAACL-HLT 2013*, pp. 188–196, (2013).

[19] S. Bykh and D. Meurers, "Exploring syntactic features for native language identification: A variationist perspective on feature encoding and ensemble optimization," in *the 25th International Conference on Computational Linguistics, COLING 2014*, pp. 1962–1973, (2014).

[20] C. Benavides, "La distribución del voseo en hispanoamérica," *Hispania*, vol. 96, no. 3, pp. 612–623, (2003).

[21] M. Zampieri and B. G. Gebre, "Automatic identification of language varieties: The case of Portuguese," in *The 11th conference on natural language processing, KONVENS 2012*, pp. 233–237, Österreichischen Gesellschaft für Artificial Intelligende (ÖGAI), (2012).

[22] F. Sadat, F. Kazemi, and A. Farzindar, "Automatic identification of arabic language varieties and dialects in social media," *Proceedings of the Second Workshop on Natural Language Processing for Social Media, SocialNLP*, pp. 22–27, (2014).

[23] W. Maier and C. Gómez-Rodríguez, "Language variety identification in spanish tweets," in *Proceedings of the EMNLP'2014 Workshop on Language Technology for Closely Related Languages and Language Variants*, pp. 25–35, (2014).

[24] J. Tiedemann and N. Ljubešić, "Efficient discrimination between closely related languages," in *Proceedings of COLING 2012*, pp. 2619–2634, (2012).

[25] M. Lui and T. Baldwin, "Langid.py: An off-the-shelf language identification tool," in *Proceedings of the ACL 2012 System Demonstrations*, ACL '12, pp. 25–30, Association for Computational Linguistics, (2012).

[26] N. Ljubešić and D. Kranjčić, "Discriminating between closely related languages on twitter," *Informatica*, vol. 39, no. 1, pp. 1–8, (2015).

[27] M. Zampieri, L. Tan, N. Ljubešić, and J. Tiedemann, "A report on the DSL shared task 2014," in *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects, VarDial*, pp. 58–67, Association for Computational Linguistics, (2014).

[28] M. Zampieri, L. Tan, N. Ljubešić, J. Tiedemann, and P. Nakov, "Overview of the DSL shared task 2015," in *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects. LT4VarDial*, (2015).

[29] M. Franco-Salvador, F. Rangel, P. Rosso, M. Taulé, and M. A. Martí, "Language variety identification using distributed representations of words and documents," in *Experimental IR meets multilinguality, multimodality, and interaction. CLEF 2015*, pp. 28–40, Springer-Verlag, LNCS(9283), (2015).

[30] F. Rangel, M. Franco-Salvador, and P. Rosso, "A low dimensionality representation for language variety identification," in *17th International Conference on Intelligent Text Processing and Computational Linguistics, CICLing 2016*, Springer-Verlag, LNCS(), (2016).

[31] B. Roark, M. Saraclar, and M. Collins, "Corrective language modeling for large vocabulary ASR with the perceptron algorithm," in *Acoustics, Speech, and Signal Processing, ICASSP'04. IEEE International Conference on*, vol. 1, pp. I–749, IEEE, (2004).

[32] J. B. Marino, R. E. Banchs, J. M. Crego, A. de Gispert, P. Lambert, J. A. Fonollosa, and M. R. Costa-Jussà, "N-gram-based machine translation," *Computational Linguistics*, vol. 32, no. 4, pp. 527–549, (2006).

[33] A. Tomović, P. Janičić, and V. Kešelj, "n-gram-based classification and unsupervised hierarchical clustering of genome sequences," *Computer methods and programs in biomedicine*, vol. 81, no. 2, pp. 137–153, (2006).

[34] S. Bird, E. Klein, and E. Loper, "Natural Language Processing with Python". *O'Reilly Media*, (2009).

[35] V. Metsis, I. Androutsopoulos, and G. Paliouras, "Spam filtering with naive bayes-which naive bayes?," in *Proceedings of the 3rd Conference on Email and Anti-Spam, CEAS 2006*, pp. 27–28, (2006).

[36] Y. Wang, J. Hodges, and B. Tang, "Classification of web documents using a naive bayes method," in *Proceedings on 15th IEEE International Conference on Tools with Artificial Intelligence*, pp. 560–564, IEEE, (2003).

[37] T. M. Mitchell, "Machine Learning". *McGraw-Hill, Inc.*, 1 ed., (1997).

[38] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, "Classification and regression trees". *CRC press*, (1984).

[39] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory, COLT92*, pp. 144–152, ACM, (1992).

[40] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, (2008).

[41] D. Torunoğlu, E. Çakirman, M. C. Ganiz, S. Akyokuş, and M. Z. Gürbüz, "Analysis of preprocessing methods on classification of turkish texts," in *International Symposium on Innovations in Intelligent Systems and Applications, INISTA 2011*, pp. 112–117, IEEE, (2011).

[42] J. W. Pennebaker, "The secret life of pronouns: What our words say about us," *New York: Bloomsbury Press*, (2011).

[43] R. Fabra-Boluda, F. Rangel, and P. Rosso, "NLEL_UPV_Autoritas participation at discrimination between similar languages (DSL) 2015 shared task," in *Proceedings of the RANLP Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects, LT4VarDial*, pp. 52–58, (2015).

[44] C. Goutte, S. Léger, and M. Carpuat, "The NRC system for discriminating similar languages," in *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects, COLING 2014*, pp. 139–145, Association for Computational Linguistics, (2014).

[45] J. Porta and J.-L. Sancho, "Using maximum entropy models to discriminate between similar languages and varieties," in *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*, (Dublin, Ireland), pp. 120–128, Association for Computational Linguistics, (2014).

[46] M. Purver, "A simple baseline for discriminating similar languages," in *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*, (Dublin, Ireland), pp. 155–160, Association for Computational Linguistics, (2014).

[47] B. King, D. Radev, and S. Abney, "Experiments in sentence language identification with groups of similar languages," in *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*, (Dublin, Ireland), pp. 146–154, Association for Computational Linguistics, (2014).

[48] M. Lui, N. Letcher, O. Adams, L. Duong, P. Cook, and T. Baldwin, "Exploring methods and resources for discriminating similar languages," in *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*, (Dublin, Ireland), pp. 129–138, Association for Computational Linguistics, (2014).

[49] M. Zampieri, L. Tan, N. Ljubešić, J. Tiedemann, and P. Nakov, "Overview of the DSL shared task 2015," in *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects, LT4VarDial*, (Hissar, Bulgaria), (2015).

[50] N. Shuyo, "Language detection library for java." http://code.google.com/p/language-detection/, (2010).

[51] L. Tan, M. Zampieri, N. Ljubešic, and J. Tiedemann, "Merging comparable data sources for the discrimination of similar languages: The DSL corpus collection," in *Proceedings of the 7th Workshop on Building and Using Comparable Corpora*, (Reykjavik, Iceland), pp. 11–15, (2014).

[52] I. H. Witten and E. Frank, "Data Mining: Practical machine learning tools and techniques". *Morgan Kaufmann*, (2005).

Glosario

AP *Author Profiling*. 1, 12, 13, 15, 46, 66

DSL *Discrimination between Similar Languages*. 13–15, 17, 18, 56, 66

LDR *Low-Dimensionality Representation*. 1, 35, 37, 42, 46, 48, 49, 52–54, 57, 58, 60, 61, 63, 65, 66

LI *Language Identification*. 12, 13, 18

LVI *Language Variety Identification*. 1, 12–20, 35, 56, 65, 66

NE *Named Entities*. 4, 59–63

NLI *Native Language Identification*. 13–16

PLN Procesamiento del Lenguaje Natural. 43

POS *Part of Speech*. Etiquetas las palabras con información asociada. Típicamente, categorías gramaticales. 15, 16

SVM *Support Vector Machines*. 4, 10, 14–16, 18, 43, 44, 46, 48–54, 57, 65, 66

TF *Term Frecuency*. 10, 37, 39, 46, 48–51, 53, 54, 65

TF-IDF *Term Frequency-Inverse Document Frecuency*. 1, 10, 35, 37, 41, 42, 46, 48–53, 65