GeoPerfil Profesional: una herramienta automática de información sobre profesionales

Emanuel Balcazar, Lucas Bobadilla, Waldo Fusiman, and Leo Ordinez

Laboratorio de Investigación en Informática (LINVI), FI - UNPSJB
Bvd. Brown 3051, Puerto Madryn, Argentina
{emanuelbalcazar13,lucasboba,wfusiman,leo.ordinez}@gmail.com

Resumen La Provincia del Chubut cuenta con características poblacionales de muy baja densidad general y alta concentración en pocas ciudades. A la vez, la actividad industrial y comercial requiere un agregado de valor diferencial para poder competir con los grandes centros urbanos o incluso lograr exportar sus productos. En este sentido, no se dispone de un registro de profesionales calificados que puedan aportar su conocimiento a dichos sectores. La presente propuesta busca contribuir al fortalecimiento del sector PyME de la provincia del Chubut mediante la construcción de una herramienta de extracción y procesamiento automático de información, para la consolidación de un mapa de perfiles profesionales de la provincia. Se pretende extraer información pública de medios digitales, analizarla en términos de campos disciplinares, identificar personas y proveer una forma de visualización georreferenciada.

Keywords: perfiles profesionales, extracción automática de información, medios digitales, NLP

1. Introducción

La Provincia del Chubut presenta una especialización productiva estrechamente ligada a la explotación y aprovechamiento de sus riquezas naturales. En términos generales, la economía de Chubut tiene una presencia notable de producción de *commodities* que, a su vez, posibilitan la existencia de un conjunto de industrias relacionadas[3][7].

Contexto productivo provincial. De acuerdo a datos del Observatorio PyME Regional Provincia del Chubut, el 43 % de las PyME industriales del Chubut se localiza en Comodoro Rivadavia, en tanto el 57 % restante reside en las localidades de Trelew, Puerto Madryn, Rawson y Gaiman [3]. Respecto a la contratación de personal, las principales dificultades de estas empresas parecen concentrarse en la contratación de operarios calificados y/o técnicos no universitarios, con el 80 % de las firmas con un grado de dificultad medio o alto. En segundo lugar, aparecen las dificultades en la contratación de universitarios, con el 73 % de

las firmas con dificultades medias o altas. Transversalmente debido a la característica de explotación de recursos naturales a gran escala se debe mencionar el desarrollo de base tecnológica para la industria metalmecánica.

La necesidad de formación de recursos humanos especializados, el mejoramiento de procesos productivos, la incorporación de nuevas tecnologías y el apoyo al desarrollo de nuevos productos, son algunos de los principales desafíos identificados. Se trata de problemáticas que comparten los proveedores de la industria petrolera, del aluminio y naval pesquera, y que deben a su vez ser atendidas en estrecha relación con las áreas vinculadas al desarrollo productivo de la Provincia. El principal motivo señalado por el cual los empresarios tienen dificultades para contratar personal es la escasa preparación o formación inadecuada para cubrir los puestos [3]. En este sentido, se ve la necesidad de aprovechar el limitado capital humano disponible en la Provincia, poniendo a disposición de los empresarios un mapa de perfiles profesionales calificados.

Trabajos relacionados. La extracción y el procesamiento automático de información mediante técnicas de *Procesamiento de Lenguaje Natural* (NLP) [4] tiene una gran variedad de campos de aplicación. En [9], los autores analizan artículos periodísticos a fin de detectar relatos dentro de narrativas más complejas y de ellos obtener información de los actores involucrados en esos relatos. En [5], los autores analizan la cobertura periodística respecto al bullying y cyberbullying en un período de seis años en diarios de Estados Unidos. En [6], se estudia y compara el desempeño de diferentes consultores económicos polacos en base a sus informes. Un caso cercano al propuesto en este trabajo, aunque para un estadio superior, es presentado en [2]. Allí se desarrolla una medida para cuantificar la preparación de los empleados que pertenecen a una gran empresa con respecto al paradigma de la Industria 4.0. El proceso permite la identificación de tecnologías, técnicas y habilidades relacionadas contenidas en las descripciones de trabajo. A partir de estos, se mide el impacto de la Industria 4.0 en cada perfil de trabajo. Por otro lado, en [1] se plantea un sistema de recomendación académica en base a los conocimientos previos manifestados por los candidatos, una clasificación de conocimiento generada a partir de Wikipedia.

Objetivo del trabajo. En virtud de la ausencia de un registro sistematizado, en particular, de graduados universitarios en la provincia[1], se planteó la posibilidad de construir un prototipo de software que realice la tarea de consolidar la información dispersa respecto a los mismos, mediante la búsqueda, extracción, clasificación y presentación de información de manera automática. En este proceso, el insumo principal serán notas periodísticas que den cuenta de nuevos graduados o colaciones de grado.

[1] En Chubut se encuentran dictando carreras presenciales la Universidad Nacional de la Patagonia San Juan Bosco, la Universidad Tecnológica Nacional y la Universidad del Chubut (provincial).

Organización. El resto del trabajo se organiza de la siguiente manera: la Sección 2 presenta el modelo de datos general de la propuesta; en la Sección 3 se expone el mecanismo de recuperación de la información de medios periodísticos; el mecanismo de procesamiento de la información obtenida se muestra en la Sección 4; una síntesis de los resultados obtenidos a partir del prototipo desarrollado se expone en la Sección 5; finalmente, la Sección 6 presenta las conclusiones y trabajos futuros.

2. Modelo de Datos

El esquema de datos del sistema se detalla en la Figura 1:

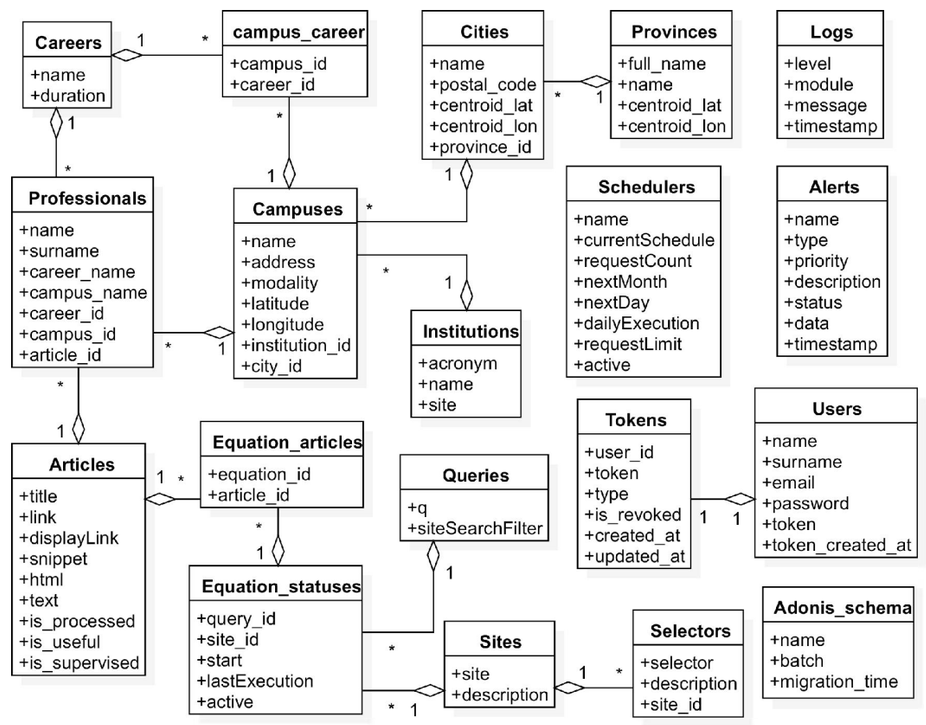

Figura 1. Esquema de datos

Las principales entidades del modelo son:

campuses: representa las sedes institucionales. Sus atributos permiten guardar su nombre, dirección, modalidad de cursado (presencial o virtual), su posición georeferenciada y posee además una *foreign key* a la institución a la cual pertenece.

institutions: representa las instituciones. Por lo general una institución posee una o varias sedes por lo cual se decidió separar la sede de la institución para diferenciarlos bien.

careers: representa las carreras, solo se guardan el nombre y su duración como información básica.

campus_careers: hace de intermediario entre las tablas campus y careers dado a que su relación se definió como un *muchos a muchos* debido a que muchas carreras pueden dictarse en múltiples sedes a su vez que una sede dicta múltiples carreras.

equations: representa una búsqueda de Google, en donde el atributo "q" almacena las palabras a utilizar en el motor.

equation_statuses: representa el estado de una ecuación, en la cual se guarda una referencia a la consulta de búsqueda y el sitio web utilizado, el último mes en el que fue ejecutada la ecuación y el último índice obtenido de la búsqueda.

equation_articles: hace de intermediario entre las entidades *equation_statuses* y *articles* para registrar qué ecuaciones se utilizaron para los artículos que fueron encontrados.

articles: se guardan los artículos obtenidos por el extractor en cada ejecución, es utilizada por el módulo NLP para obtener y procesar el contenido de los mismos.

sites: aquí se guardan los sitios webs de interés para ser utilizados por las ecuaciones de búsqueda, de esta forma es más sencillo agregar o quitar determinados sitios.

selectors: guarda los distintos selectores utilizados para cada sitio web, el cual la ecuación de búsqueda también hace uso para aplicarlos sobre los HTML y obtener así los textos de cada artículo.

users: se guardan los usuarios del sistema, con su correo electrónico y contraseñas encriptadas.

tokens: es creada por el framework AdonisJS por defecto.

schedulers: se guarda la configuración del planificador, sus atributos permiten guardar la periodicidad actual del mismo, cuál es la periodicidad a ejecutar por día y por mes, cuál es el límite de solicitudes por día que se pueden realizar y si el planificador debe estar o no activo al iniciar la aplicación.

adonis_schema: es usada por el framework AdonisJS.

3. Recuperación de la Información

3.1. Extracción de Artículos

Para la recuperación de los artículos que contengan información sobre profesionales graduados, se construyó un extractor de artículos que, en conjunto con el planificador, realizan todo el proceso de extracción y persistencia de los mismos. El esquema general del extractor se muestra en la Figura 2 y a continuación se describen sus componentes.

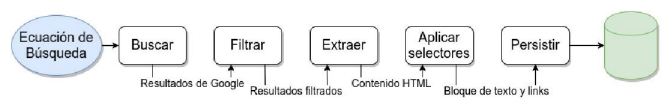

Figura 2. Esquema del proceso de extracción de artículos.

A. Ecuación de búsqueda: El extractor recibe como entrada una ecuación de búsqueda, se entiende por ecuación a un conjunto de parámetros que contienen los valores necesarios para que Google pueda realizar su búsqueda. El esquema general de una ecuación de búsqueda se muestra a continuación:

```
{"extractor": <nombre del extractor>
   "eq": {"q": <palabras a buscar>,
         "siteSearch": <sitio donde buscar>,
         "cx": <id del buscador personalizado>,
         "key": <key del usuario de Google>,
         "siteSearchFilter": "i"},
    "selectors": <array de selectores a aplicar>}}
```

extractor: permite manejar múltiples extractores.

q: consulta de búsqueda, Google lo utiliza como las palabras "clave" en su buscador.

siteSearch: sitio en donde Google realizará su búsqueda, esto permite filtrar los sitios webs y orientar las búsquedas en determinados lugares de interés (en este caso educativos y de noticias).

siteSearchFilter: un indicador de que el parámetro "siteSearch" debe tenerse en cuenta, por defecto su valor es siempre "i" (*include*).

start: un número que indica a partir de qué índice se debe comenzar la búsqueda, este índice se agrupa de 10 en 10 por cada página de los resultados de Google por lo cual haciendo cálculos se puede obtener resultados a partir de determinadas páginas.

selectores: un conjunto de valores que son utilizados para extraer efectivamente el texto incluido en el artículo. Cada página posee un conjunto de selectores que permiten obtener partes de la misma.

cx: ID del buscador personalizado. Éste se obtiene al crear un buscador en Google CSE en donde además se nos otorga una *key* para hace uso de la misma.

B. Búsqueda: Luego de recibir la ecuación, el extractor hace la llamada a la API de Google CSE para obtener los resultados, además se le provee de otros parámetros (como el id de cliente y la key) para poder hacer uso de la misma. La API retorna un JSON con los resultados obtenidos a causa de la búsqueda.

C. Filtro: Tras recibir los resultados de búsqueda, se filtran algunos enlaces de sitios webs que contengan determinadas palabras para evitar manipular información innecesaria. En nuestro caso, se filtran los enlaces que contengan la palabra "tags" ya que dichas páginas no brindan ningún tipo de información. Además se filtran los artículos que ya hayan sido obtenidos anteriormente para evitar procesar repetidas veces una misma información.

D. Extracción: En este paso se realiza la extracción iterando sobre cada resultado de búsqueda de Google y obteniendo todo el contenido de los enlaces *en crudo* para su posterior procesamiento.

La extracción se realiza accediendo secuencialmente a los "links" devueltos por Google sobre cada artículo encontrado en el resultado de búsqueda, mediante una librería se hace una llamada HTTP al enlace obteniendo así el sitio completo en formato HTML.

E. Aplicación de selectores: Se aplican sobre los HTML obtenidos en el paso anterior, quienes dividen y obtienen las porciones de texto de interés de cada artículo. Los selectores aplicados son diferentes para cada sitio por lo que cada resultado de búsqueda de Google devuelve todo lo recuperado en un determinado sitio para facilitar su procesamiento.

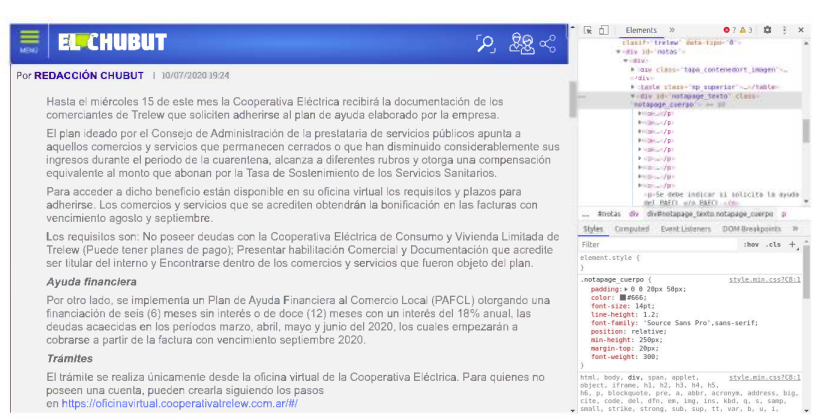

Figura 3. Ejemplo de uso de selectores Diario El Chubut

Los selectores se obtienen analizando la estructura HTML del sitio web que se desea observando en qué secciones se suelen ubicar los textos de los mismos. Para ilustrar el proceso se usan las noticias de la página `https://www.elchubut.com.ar` (ver Figura 3). Utilizando la herramienta de "inspeccionar elemento" del navegador, se puede identificar el bloque correspondiente al texto del artículo, en este caso con el atributo `class=notapage_cuerpo`, así que ése es el selector de dicha página.

Al aplicar el selector y guardarlo en una variable, podemos ver que posee una lista de un elemento con el bloque completo de texto, accediendo a ella se puede obtener el texto del mismo así como otra información (metadatos) que se almacenan en el sitio. Como último paso este selector se guarda en el sistema para su uso durante las extracciones automáticas. Este proceso debe ser realizado con cada página distinta ya que se suelen utilizar distintos selectores y la estructura HTML varia dependiendo el sitio.

F. Persistencia: Como último paso, se persisten los artículos extraídos, en particular los textos obtenidos así como el enlace de donde vino, la ecuación

utilizada, etc. Estos artículos se marcan con una bandera que indica que están listos para ser procesados por el NLP quien es el siguiente en actuar.

3.2. Planificación

Para poder realizar la extracción de artículos de forma automatizada, se tomó la decisión de implementar un planificador para la ejecución periódica del extractor y la obtención constante de información. Esta ejecución se ajusta automáticamente a los parámetros de ejecución preconfigurados al iniciar la aplicación y los mismos pueden ir variando dependiendo del estado de la ejecución como por ejemplo: se ejecutaron todas las ecuaciones o se llego al limite de búsquedas por día.

Esta decisión da como resultado la implementación de un planificador automatizado que se ejecuta en el servidor en intervalos configurables de tiempo. Esto permite que no se necesite de una persona involucrada en la búsqueda y extracción sino que se deja la responsabilidad a dicho proceso. En la Figura 4 se detalla como es el funcionamiento del planificador.

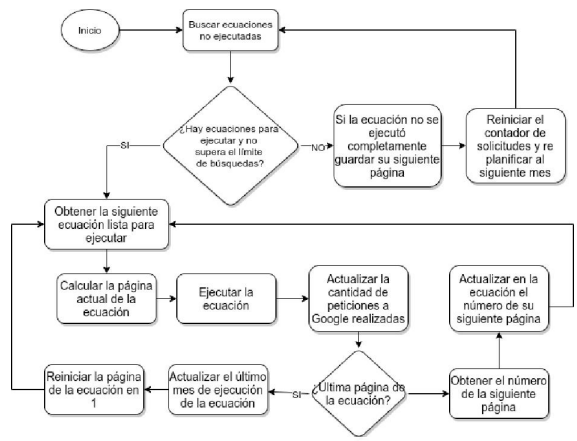

Figura 4. Esquema de funcionamiento del planificador

El planificador se implementó para acceder a tantos resultados de búsqueda como sea posible, esto significa que puede ocurrir una situación en la cual una búsqueda quede "por la mitad" y debe retomarse al día siguiente debido al límite de 100 búsquedas por día, impuesto por Google. Un inconveniente que posee la API de Google en su uso gratuito es que no se puede acceder más allá de la página 10 de los resultados, por lo que es un factor a tener en cuenta al momento de hacer la búsqueda.

Como periodicidad, se definió que el planificador deberá ejecutar una vez al día hasta llegar al límite de las 100 peticiones a la API de Google por mes

mientras aún haya ecuaciones sin ejecutar. En caso de que se hayan ejecutado todas las ecuaciones disponibles, el planificador se auto-replanifica para ser ejecutado al mes siguiente. Estos parámetros pueden ser modificados para ser utilizados por ejemplo en testeos del planificador, haciendo que su ejecución sea en intervalos de pocos minutos para asegurar su correcto funcionamiento.

4. Procesamiento de Artículos

Para el procesamiento del texto se optó por utilizar una librería llamada SpaCy [8], al ser una librería de Python el módulo completo de procesamiento de texto se realizó en este lenguaje. En la Figura 5 se muestra el esquema inicial que se planteó para este módulo.

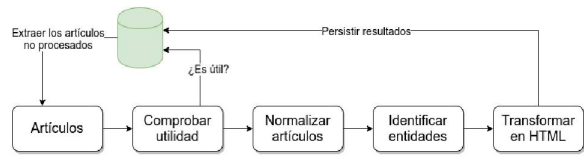

Figura 5. Esquema del procesamiento de artículos

Una vez que los artículos son extraídos y almacenados en la base de datos por el planificador, se dispone a ejecutar el módulo de NLP. Los pasos que se ejecutan en el módulo son:

1. Los artículos se guardan con una bandera `is_processed` que indica si el artículo ya fue o no procesado por el módulo NLP. Se obtiene de la base de datos todos los artículos hasta el momento sin procesar para comenzar con el análisis y procesamiento.
2. Se comprueba la utilidad del artículo, esto se hace simplemente observando el contenido del título del mismo, si presenta las palabras "colación", "graduados" u otras similares, se lo tiene en cuenta como un artículo de utilidad y se lo marca como un artículo útil en la base de datos. Esto se debe a que muchos artículos obtenidos por el extractor si bien pueden contener las palabras de búsqueda deseadas, quizás su contenido no es del todo preciso o es diferente a lo esperado. Por lo que realizar un análisis y filtro de los artículos basado en su contenido es importante para poder diferenciarlos.
3. Como tercer paso, se eliminan los caracteres que puedan perjudicar al NLP al momento de identificar las entidades. En este paso se detectó que en las distintas fuentes de los artículos, el carácter coma (,) generaban ruido al NLP, dado que si están mal ubicadas o hay mucha cantidad de ellas, puede generar confusión al momento de obtener un nombre y apellido de una persona.
4. Una vez quitado en el artículo los caracteres ruidosos, en este paso se ejecuta el NLP, para que realice la identificación de nombres y apellidos de profesionales.

Para la identificación de nombres y apellidos, se usa como base es el dataset de SpaCy `es_core_news_md`, donde en la configuración se le indica que identifique solamente las entidades personas `PER` y se le pasa un *template* para que cuando identifique a la persona genere el HTML correspondiente.

5. Cuando se detectan los apellidos y nombre, se transforma el artículo a un formato HTML, que resalta los nombres encontrados y permitirá desde la interfaz de usuario confirmar, y asignar una carrera e institución a cada uno de los profesionales.

5. Resultados

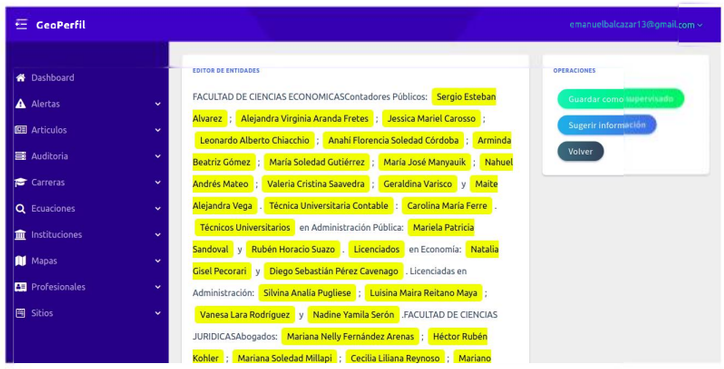

Figura 6. Interfaz de usuario para supervisión.

Supervisión de resultados. Como parte del procesamiento final de los artículos, se incluye una interfaz de usuario donde se muestran los resultados obtenidos por el procesamiento, y a partir de allí, mediante un asistente, se solicita la confirmación o corrección de la información de los profesionales, como así también la asignación de la carrera/título e institución de cada uno de ellos. Una captura se muestra en la Figura 6.

Desempeño. El funcionamiento de los extractores y el planificador obtuvieron buenos resultados así como la información que se lograba obtener al utilizar el módulo NLP con la librería SpaCy. La ejecución de una ecuación de búsqueda demoró aproximadamente entre 1 y 3 minutos, esto depende de varios factores como: la conexión a internet, la cantidad de páginas que devuelva la búsqueda (no siempre se puede alcanzar el límite de 10 páginas por ecuación) y la cantidad de veces en la que la ecuación sea ejecutada por encima de las demás debido a que no se ejecuta una ecuación completa de una sola vez sino que una vez extraída una página se cambia a la siguiente ecuación lista para ejecutar.

La cantidad de artículos obtenido por ecuación rondan entre los 30 y 100 artículos generalmente, y el número de profesionales puede ser muy disperso ya

que se han encontrado artículos con más de 200 profesionales y otros casos con 20 o 30.

6. Conclusiones

La implementación del proyecto, logró mostrar la factibilidad de construir una herramienta basada en técnicas de recuperación de información y procesamiento de lenguaje natural, para asistir a las PYMEs de Chubut en su mejora de competitividad.

Como trabajos futuros se propone la ampliación del espectro de búsquedas para identificar profesionales no a partir de la noticia de su graduación sino mediante sus menciones en noticias y la vinculación de dichos nombres con potenciales usuarios de redes sociales, así como la implementación de un mecanismo de *verificación (externa) y validación (propia) de identidad.*

Referencias

1. Amini, B., Ibrahim, R., Othman, M.S., Selamat, A.: Capturing scholar's knowledge from heterogeneous resources for profiling in recommender systems. Expert Systems with Applications 41(17), 7945 – 7957 (2014), http://www.sciencedirect.com/science/article/pii/S0957417414003807
2. Fareri, S., Fantoni, G., Chiarello, F., Coli, E., Binda, A.: Estimating industry 4.0 impact on job profiles and skills using text mining. Computers in Industry 118, 103222 (2020), http://www.sciencedirect.com/science/article/pii/S0166361519309327
3. Ibañez, J., Ball, F.: Industria manufacturera año 2010 : Observatorio pyme regional provincia del chubut. Tech. Rep. CDD 338.47, Fund. Observatorio Pyme; Bononiae Libris; GECSEA Patagonia - Universidad Nacional de la Patagonia San Juan Bosco, Buenos Aires (Sept 2011)
4. Jackson, P., Schilder, F.: Natural language processing: Overview. In: Brown, K. (ed.) Encyclopedia of Language Linguistics (Second Edition), pp. 503 – 518. Elsevier, Oxford, second edition edn. (2006), http://www.sciencedirect.com/science/article/pii/B0080448542009275
5. Moreno, M., Gower, A., Brittain, H., Vaillancourt, T.: Applying natural language processing to evaluate news media coverage of bullying and cyberbullying. Prevention Science 20 (08 2019)
6. Rybinski, K.: Ranking professional forecasters by the predictive power of their narratives. International Journal of Forecasting (2020), http://www.sciencedirect.com/science/article/pii/S0169207020300601
7. Selva, R.A.: Ventajas competitivas de la provincia del chubut, herramientas para la toma de decisiones. Tech. Rep. 49246, CONSEJO FEDERAL DE INVERSIONES, Buenos Aires (Nov 2011)
8. SpaCy: Industrial-Strength Natural Language Processing. https://spacy.io/ (2020), online; accessed 20 julio 2020
9. Zhang, H., Boons, F., Batista-Navarro, R.: Whose story is it anyway? automatic extraction of accounts from news articles. Information Processing Management 56(5), 1837 – 1848 (2019), http://www.sciencedirect.com/science/article/pii/S0306457318306101