Hunspell tutorial

Repetimos que esta página no va de ciencias sociales, sin embargo somos fanáticos de la ortografía y en esta oportunidad os presentamos una herramienta para corregir nuestros escritos, que ya son bastantes, por demás. Ahora es el punto donde nos diréis "ea, tío, ¿de qué váis si eso está integrado en 'todos' los programas?" Pues bueno preparaos a sorprenderos con los que os vamos a contar, ¡vamos a aprender a programar para corregir nuestra ortografía!

Introducción.

Ya tocamos el tema del idioma inglés anteriormente, su semántica y ortografía, un toque leve pues, nada profundo. Ahora os presentamos una herramienta llamada HunSpell para ayudarnos rápidamente a corregir archivos completos de una manera automatizada con dos grandes ventajas: **se puede usar la línea de comandos (scripts bash) y con Python3.**

Historia.

HunSpell deriva de MySpell.

MySpell.

MySpell fue desarrollado por Kevin Hendricks quien buscaba integrar varias herramientas de software libre para la suite de oficina Apache OpenOffice.org pero no fue sino con la ayuda de Kevin Atkinson (creador de Aspell -sucesor de Ispell- y Pspell) y un arduo trabajo en C++ que nace MySpell el cual soporta "Affix compression" el cual tiene una capital importancia para nosotros los programadores y ya veremos cómo cuando toquemos el tema con Python (Python3 en específico).

HunSpell.

¿Por qué crean HunSpell si ya existía MySpell? Porque este mundo es grande y ancho y para el idioma húngaro se necesitaba un corrector ortográfico que pudiera manejar UTF-8 y por ende los idiomas del mundo entero. Nosotros por estos lares siempre indicamos -y luchamos- para que esta codificación en bases de datos, lenguajes de programación y lenguajes de marcado sean presentados correctamente tanto por pantalla como por impresoras (aunque estas útlimas se están dejando de usar para ahorrar papel, tinta, dinero y recursos naturales). **De allí proviene el "HUN-"del idioma húngaro,** "Spell" es deletrear en inglés, de allí la palabra compuesta.

La última versión estable data del 2011 y al igual que su predecesor MySpell, HunSpell está escrito en lenguaje C++. En su repositorio bien reza "la librería de corrección ortográfica más popular" y al momento de escribir estas líneas tenía 7 horas de haber sido actualizado dicho repositorio ¡está más viva que nunca HunSpell!

Programas que utilizan HunSpell.

Y he aquí la sorpresa: HunSpell es el corrector ortográfico de fuente abierta para una gran cantidad de programas que usamos diariamente, por nombrar unos cuantos:

- Apache OpenOfiice.
- **¡LibreOffice!**
- Thunderbid.
- **Mozilla Firefox** ¡desde el año 2005!
- Google Chrome.
- Google Chromium.
- SeaMonkey.
- WinShell.
- Opera.

Lenguajes de programación y HunSpell.

Ya os dijimos que HunSpell está escrito en lenguaje **C++** pero al menos hay dos lenguajes en los cuales podemos imprtar estas librerías. Uno es **Python**, en el cual estamos vivamente interesado gracias a su gran versatilidad y plataformas que lo soportan. El otro es el **lenguaje R** que ha tomado auge recientemente y que, por ahora, no tocaremos su uso. En nuestra sección de fuentes consultadas en idioma inglés os dejamos unos enlaces con ejemplos que nos parecieron bastantes sencillo por si queréis ahondar en ese tema.

Instalando HunSpell en Ubuntu.

Como en las distribuciones basadas en Debian, podemos utilizar **apt** para instalar por una ventana de comandos desde nuestros repositorios predeterminados. Para ello con una simple línea -con derechos de administrador **root**-, escribimos lo siguiente:

```
sudo apt-get install hunspell
```

Previo a este comando es recomendable usar **apt-get update** para sincronizar el catálogo de aplicaciones entre nuestro ordenador y los repositorios que tengamos predefinidos. Una vez hayamos instalado podemos consultar la versión que hayamos instalada:

```
hunspell -v
```

Como podéis constatar aparecen las atribuciones debidas hacia **Ispell**, **László Németh**, **MySpell**, **Kevin Hendricks** y **openOffice.org**: así es el software libre, se debe heredar la licencia y hacer econocimiento expreso a los autores a fin de evitar el patentado y que devenga en software privativo.

Uso por línea de comandos de HunSpell.

Uso de Hunspell de manera interactiva.

Una vez tengamos instalado HunSpell podemos, sin más, abrir una ventana de comando y comenzar a usarlo. Aunque su uso principal es para revisar archivos o ficheros de texto, también podemos hacerlo interactivamente. *Primero debemos conocer dónde estamos parados, **conocer cuáles diccionarios tenemos instalados**, y por ello usaremos el siguiente comando -el cual devuelve una extensa respuesta-:*

```
hunspell -D
```

Este parámetro nos devolverá, esencialmente, tres secciones:

1. **Rutas de búsqueda**: dependiendo de la cantidad de programas que tengamos instalados, cada uno de ellos -si utilizan MySpell o HunSpell- tienen sus propios diccionarios. Dada la filosofía del software libre, tenemos la opción de usarlos, descargar otros e incluso crear nuestros propios diccionarios -¡avanzado!-.
2. **Diccionarios disponibles**: la "segunda" ubicación es la del propio HunSpell, *generalmente está ubicada en la carpeta* **"/usr/share/hunspell"** (esta ruta puede variar según vuestra distribución linux utilizada). En estas rutas **también** puede aparecer la ruta "**/usr/share/myspell**" así que si tenemos más opciones, pues mejor. Los diccionarios allí listados son simples ficheros cuyo nombre está compuesta por dos letras minúsculas, el guión bajo "_" y dos letras mayúsculas. Las primeras dos letras corresponden al idioma y las segundas dos letras corresponden a la distribución regional. Por ejemplo "en_GB" representa "english Great Britain", "fr_FR" francés de Francia, "fr_CA" francés de Canadá y así sucesivamente. Para nuestro país -aunque lo identifican mal- nos corresponde "es_VE" osea español de Venezuela (cuando en realidad hablamos es castellano). Importante destacar que si queremos cargar cualquiera de los diccionarios en esta sección listada *no es necesaria especificar su ruta ya que HunSpell sabe muy bien donde están ubicados.*
3. **Diccionario cargado**: dependiendo de la configuración regional que tengamos en nuestro sistema operativo, será cargado un diccionario de manera predeterminada, por lo cual, como dijimos, podemos usar sin más a HunSpell. Para nuestro caso tenemos el fichero

"**es_VE.dic**" y su contraparte -ya veremos su uso- "**es_VE.aff**".

Vamos, pues, a comenzar a usar a HunSpell. Ya sabemos cual diccionario tenemos cargado por defecto, pero acostumbremonos a penar internacionalmente, que es de cultura general saber o al menos tener nociones de varios idiomas modernos:

```
hunspell -d es_VE
```

Acá notamos que usamos el parámetro "**-d**" para cargar un diccionario *que se encuentra en la ruta de HunSpell, por lo que no es necesario especificar su ubicación, HunSpell sabe ya donde está.* A continucación presionamos intro o enter y escribimos la plabra que deseamos revisar y presionamos INTRO. Si la palabra está bien escrita (HunSpell la busca en el diccionario y consigue una conicidencia exacta) nos devolverá ya sea asterisco "*****", signo de suma "**+**" o un signo de resta "**-**". Por ahora nos conformamos en saber que la palabra es correcta. *De estar mal escrita la palabra (no se consigue en el diccionario) HunSpell nos devuelve una lista de palabras aproximadas donde, generalmente, la primera que aparece es la correcta. Aquí es muy importante el archivo "es_VE.aff", que utiliza el "affix compression" -fijaos la extensión del fichero-: hay palabras raíces y con prefijos, sufijos o ambos, encontramos su aproximación a la palabra* correcta. Es decir, según unas reglas predeterminadas (palabras raíces, palabras derivadas) podemos corregir y lograr la ortografía.

Muchas veces devolverá pocas y a veces varias, veamos la imagen siguiente:

Notamos que si escribimos mal el nombre de nuestro país, HunSpell nos devuelve en primer lugar la aproximación más cercana y luego una segunda palabra que medio se le parece. Hasta aquí todo bien pero si escribimos la palabra "**vurro**" (sic) lo primero que nos devolverá es "**curro**" -palabra no muy común en Venezuela- y en segunda opción la buscada, el animal muy útil en las granjas aún hoy en día. Luego nos muestra palabras menos comunes aún. ***Es por ello que debemos aprender a crear o modificar nuestros propios diccionarios de acuerdo al uso más común que damos a nuestra habla cotidiana.*** Es evidente que este diccionario proviene de España ¡NO HAY PROBLEMA CON ESO! Allá se originó nuestro castellano, de la región abundante en castillos, Castilla, que era más fortificada y pudo someter militarmente al resto de la península ibérica e inclusó expulsó a los moros que por 600 años estuvieron allí construyendo ciudades y civilizando a la población con los últimos avances científicos de la época.

Pero seamos sinceros, las diferencias son evidentes entre el castellano de España y el castellano de Venezuela:

- coche -> carro.
- móvil -> celular (teléfono).
- piso -> apartamento (departamento, vivienda).
- ordenador -> computadora.
- contadores -> medidores (electricidad).
- Y paren ustedes de contar...

Uso de Hunspell sobre archivos de texto.

Cuando le indicamos a Hunspell que trabaje sobre un archivo de texto no es completamente automático su comportamiento, en cierto modo es interactivo con nosotros. Pero primero veamos el comando a utilizar:

```
hunspell -d es_VE archivo_de_texto.txt
```

Una vez comenzamos la tarea, Hunspell cambia sin preguntar las palabras mal escritas *que encuentre en coincidencia en el diccionario especificado.* Pero si no consigue una coincidencia clara nos pregunta a nosotros con las siguientes opciones:

- Tecla "R": reemplaza la palabra mal escrita completamente.
- Tecla "Espacio": Acepta la palabra solamente por esta vez.
- Tecla "A": Acepta la palabra para el resto de esta sesión.
- Tecla "I": Acepta la palabra, y el ingreso en su diccionario privado.
- Tecla "U": Acepta y añade la versión minúscula en el diccionario privado.
- Tecla "S": Pedir una raíz y una palabra modelo para almacenarlas en el diccionario personal.
- Números: cuando son pocas las opciones uno puede pulsar un solo número pero a veces la lista es extensa y hay que pulsar dos, como por ejemplo doble cero.

Uso práctico de Hunspell.

Pues si ya ustedes son lectores habituales de este vuestro humilde blog -y sino pues ahora lo sabrán- **somos terriblemente pragmáticos con el software, nos gusta, amamos y nos encanta darle uso práctico a las herramientas informáticas.** Para este caso estamos usando el Hunspell con los archivos de texto generados con el *reconocimiento óptico de caracteres, al* cual le dedicamos una entrada correspondiente*. A esas imágenes que capturamos con un aumento de 200%* (*zoom* le dicen las personas que hablan inglés) y con ayuda de el maravilloso Shutter capturamos por párrafos y luego los corregimos de manera rápida con Hunspell.

Fuentes consultadas.

En idioma castellano.

- "HunSpell" en Wikipedia.

En idioma inglés.

- "About HunSpell".
- "HunSpell" repository at GitHub.
- "Affix compression".
- "Hunspell Tutorial" By Xah Lee.
- "HunSpell tutorial for v1.4.3 R language".
- "Hunspell: Spell Checker and Text Parser for R".
- "Stemming and Spell Checking in R".