Euspell: corrección ortográfica del euskera en software libre

Iñaki Alegria[1], Klara Ceberio[1], Nerea Ezeiza[1], Aitor Soroa[1], Gregorio Hernandez[2]

[1] Grupo Ixa, Euskal Herriko Unibertsitatea
i.alegria@ehu.es
[2] Eleka S.L.

Resumen. En este artículo se presenta la primera versión en software libre del corrector ortográfico para el euskera desarrollada entre el grupo de investigación IXA de la Universidad del País Vasco y la empresa Eleka. Debido a la complejidad morfológica del euskera las soluciones basadas en los programas libres *ispell*, *aspell* y *myspell* no son satisfactorias y ha habido que esperar a la difusión de *hunspell* para encontrar una herramienta de software libre idónea para idiomas de morfología compleja. La solución propietaria de nombre *Xuxen* ha tenido un gran éxito, por lo que esperamos que la distribución de *euspell* sea un éxito en un gran número de aplicaciones, sobre todo como complemento a los programas de ofimática (*OpenOffice*), edición para blogs (*Firefox2*) y correo electrónico. Como estos últimos aún no han adoptado *hunspell*, se ha desarrollado una versión derivada compatible con *myspell*.

Keywords: software libre, corrección ortográfica, ingeniería lingüística, morfología

1. Introducción

Desde 1992 está disponible de forma gratuita un corrector ortográfico para el euskera llamado *Xuxen*, que inicialmente fue desarrollado como producto independiente para PC y Mac reconociendo los formatos de *MSWord* y *WordPerfect* y posteriormente como plugin de *MSOffice* para PC y Mac. Adicionalmente se ha preparado una versión especializada para corrección en OCR. También hubo un desarrollo cerrado para *OpenOffice1*. Sin embargo el desarrollo de un software libre de corrección ortográfica para el euskera se ha demorado mucho por diversas razones. Las más importantes son las siguientes y están estrechamente relacionadas entre ellas:

1. el carácter aglutinante de la morfología vasca
2. la falta de un estándar adecuado en el entorno de software libre para lenguas de alta flexión.
3. la no disposición en software libre de un equivalente de la librería de Xerox[1] que se utilizaba en la mayoría las aplicaciones mencionadas anteriormente.

Hay que tener en cuenta que el corrector ortográfico base es muy complejo y eficiente.

1 www.xrce.xerox.com/competencies/content-analysis/fst/

Morfología del euskera

La morfología del euskera es especialmente rica sobre todo a nivel de flexión [1]. No existen las preposiciones y, sin embargo, hay un gran número de casos de declinación, que además varían con el número y la determinación. Los dos casos de genitivo (lugar y persona) son aglutinantes, es decir, tras el genitivo pueden aparecer cualquier otro caso.
El verbo auxiliar cambia además de en función del tiempo verbal y la persona del sujeto, en función de la persona del objeto y la del objeto indirecto.
La derivación también es bastante rica, sobre todo con respecto a la nominalización del verbo. Cualquier raíz verbal puede ser nominalizada y a partir de ahí tomar toda la flexión de los nombres. También los formas conjugadas del verbo auxiliar y principal pueden nominalizarse y encadenar todos los sufijos de nombre.
Por ejemplo, en el sintagma *alabaren etxean* (en la casa de la hija) los lemas son *alaba* (hija) y *etxe* (casa), siendo la *a* de los sufijos morfema de determinante, *ren* morfema de genitivo y *n* morfema de inesivo. Pero también se podría hacer una elipsis y decir o escribir *alabarenean* (en la de la hija). Incluso se podrían permitir dos elipsis, por ejemplo en *etxekoarena,* que se se podría traducir *el de la de la casa*, es una forma que en un buscador se puede encontrar. Una palabra no demasiado extraña como *egitekoarekin* (algo así como *con lo que se ha de hacer*) se compone de morfema: *egin* (hacer), *te* (nominalización), *ko* (genitivo singular), *aren* (genitivo singular) y *kin* (asociativo)[2].
Con estas morfología el número de formas que pueden ser generadas a partir de una raíz varían según la categoría, pero para los nombres, que es la categoría mas común, es más de mil. Por lo tanto una solución a la morfología o a la corrección ortográfica basada en una lista de palabras no es adecuada.
En Europa son idiomas de similares características morfológicas y con la misma problemática para la corrección ortográfica el finlandés, el húngaro y el turco.

Morfología de dos niveles y corrección automática

Para este tipo de idiomas la mejor manera de implementar un corrector ortográfico es basarse en un analizador morfológico [4]. Así, una palabra será presumiblemente correcta si a partir de ella es posible obtener algún análisis morfológico correcto. A la hora de hacer propuestas para las palabras incorrectas, se sigue el denominado método inverso, es decir, se generan posible palabras introduciendo cambios a partir de la original y se comprueba que existan en el idioma verificándolas morfológicamente. En consecuencia el análisis deberá ser rápido.
Para crear un analizador morfológico para este tipo de idiomas se ha demostrado que la morfología de dos niveles y sus mejoras [2][6] son los formalismos más adecuados. Las ventajas que ofrece son muy importantes:

1. Diferencian el léxico, la morfotáctica y la morfofonología con lo que la definición y el mantenimiento de toda la información es más sencillo.

2 Buscando en *google alabarenean* aparece 9 veces, *etxekoarena* 3 y *egitekoarekin* más de 100.

2. No se pone límites a ninguno de los fenómenos morfológicos nombrados, por lo que se pueden considerar formalismos universales.
3. Se compila en transductores de estados finitos, por lo que la eficiencia del analizador está garantizada.
4. Sirve tanto para análisis como para generación, por lo que el trabajo puede ser reutilizado para distintas aplicaciones (la generación se puede usar en sistemas de traducción automática por ejemplo).
5. Permite añadir y compilar conjuntamente léxicos y reglas no estándares, lo que permite corregir muy precisamente los errores de competencia léxica.

Debido a estas ventajas en 1993 adoptamos esta formalismo y definimos primero nuestro analizador/generador morfológico y, posteriormente, el corrector ortográfico. Estas herramientas son muy precisas pero tienen el inconveniente que usan una librería propietaria.

2. Software libre para corrección ortográfica

Desgraciadamente no existe software libre estandarizado para implementar morfología de dos niveles. Según nuestras referencias, solamente *mmoph*[3] aplica un formalismo de ese tipo, pero con un inconveniente, solo funciona para generación. Para el análisis se deben generar todas las formas posibles, lo cual es inviable para el euskera.

Con respecto a la familia de correctores ortográficos estándares clásicos para software libre (*aspell*, *ispell* y *myspell*), podemos decir que no ofrecen la expresividad necesaria para una fácil adaptación. Hay que tener en cuenta que siempre se han diseñado para el inglés y después se han intentado aplicar a otros idiomas.

ispell es el programa más antiguo de ellos y el que diseñó la división de los datos lingüísticos entre raíces y afijos. Adicionalmente permite reglas para eliminar y/o añadir caracteres cuando se encadena un lema con un afijo. Aunque esto le da algo de suficiente expresividad para el inglés y los idiomas latinos, es insuficiente para idiomas de las características del euskera, entre otras razones porque permite un número muy limitado de paradigmas[4].

aspell es el corrector estandar para GNU pero las únicas mejoras que aporta están dirgidas a mejorar las propuestas a los errores, y no a soportar idiomas morfológicamente más ricos.

myspell es una implementación en C++ parecida a *ispell* y orientada a integrarla en *OpenOffice* y aunque permite algo más de expresividad para la flexión, esta no es suficiente.

Las características y diferencias precisas entre estas herramientas pueden ser consultadas en la wikipedia (en inglés).

Hubiera sido sencillo prepara una lista de palabras integrable en *aspell*, o con más trabajo hacer una adaptación a los requisitos de *ispell*, pero esto habría presentado dos problemas que consideramos importantes:

---

3 packages.debian.org/unstable/misc/mmorph

4 Llamamos paradigma a cada conjunto diferenciado de sufijos que puede encadenarse tras un lema

1. falta de coherencia de los diagnósticos del corrector. Mientras ciertas declinaciones de un lema serían tomadas por correctas, otras algo menos frecuentes se darían por erróneas (falsos negativos). Dicho de otra forma, no se consigue una cobertura adecuada.
2. necesidad de mantenimiento de dos grandes y complejas bases de datos para el mismo problema.

Ante esta situación nuestro propósito fue solventar ambos problemas con el desarrollo de un software que posteriormente fuera liberado y presentado para su estandarización en el entorno del software libre. Sin embargo esta tarea demoró y fue más compleja de lo que esperábamos, y, además, coincidió con el desarrollo y estandarización de *hunspell* [7].

hunspell desarrollado inicialmente para el húngaro, soluciona la mayoría de los problemas señalados (salvo el de no obtener buenas propuestas para errores de competencia léxica), ya que permite mayor número de paradigmas y doble encadenamiento de sufijos y, por lo tanto, nos permite generar, a partir de la definición registrada en la base de datos, un corrector preciso y de gran cobertura. Además ya ha sido adoptado por *OpenOffice* y va a serlo por *Mozilla.*

3. Adaptación

Para hacer la adaptación de los datos lingüísticos del euskera se ha llevado a cabo un proceso de exportación/importación entre el sistema original, basado en la morfología de dos niveles y los transductores léxicos propuestos por Karttunen [6], a la especificación de *hunspell*. Los pasos necesarios (simplificados) han sido los siguientes:

1. Se han desarrollado en el sistema original reglas adicionales para restringir el poder generativo de la morfología vasca. Debido al gran poder generativo, ya descrito en la sección 1, la listas de afijos podía ser interminable. La decisión fue restringir el número de sufijos a tres, con ciertas excepciones que permiten sufijos adicionales (sufijos de plural, de nominalización, genitivo).
2. Añadiendo al sistema original esas reglas se ha construido un analizador/generador restringido cuyo funcionamiento será comparado con el del resultado final. Este sistema será usado también en las siguientes etapas.
3. Por un lado, de las reglas fonológicas se han obtenido las terminaciones de lema que pueden conllevar cambios al encadenarse con sufijos[5]. Por otro lado, se ha obtenido del léxico original un lema como representante de cada paradigma y terminación, agrupando las terminaciones que no producen cambios especiales en dos grupos, vocales y consonantes. Posteriormente, usando el generador construido en el paso anterior, se han generado todas las formas posibles para los lemas seleccionados.
4. A partir de las formas generadas por cada lema y el lema original se ha logrado separar las raíces y los afijos, generándose así un primer nivel de encadenamiento.

---

5 Por ejemplo la *a* final suele desaparecer, las terminaciones verbales *tz*, *ts* y *tx* al encadenarse con una consonante suelen perder la t, la r final suele doblarse delante de vocal, etc.

Esto podría ser suficiente para *myspell*, pero como se ha mencionado, aparecen dos problemas: demasiados paradigmas y listas de afijos demasiado largas.
5. En los sufijos correspondientes a cada paradigma se ha buscado la aparición del genitivo (ya que es el caso que multiplica el poder generativo) y se ha dividido el sufijo en dos partes, la anterior y la posterior. La perimera parte se ha mantenido en un primer nivel de encadenamiento y la segunda ha pasado a un segundo nivel, que es aceptado por *hunspell*.
6. Debido a que conjuntos de sufijos de segundo nivel pueden aparecer repetidos, se han identificado esos conjuntos repetidos minimando el segundo nuvel de sufijos.
7. Finalmente se han añadido todos los lemas al fichero de raíces, modificando, cuando era necesario, los últimos caracteres de la misma forma que se hizo con su representante en el paso 4.
8. Tras estos pasos se ha conseguido la información necesaria par su adaptación a *hunspell*. Con un simple cambio de formato se ha terminado la conversión
Tras diversas pruebas y correcciones, los ficheros se han liberado en enero de 2007, con una nueva versión en febrero.

4. (Re)Utilización

A continuación describimos dos productos adicionales obtenidos en este proceso y la utilización de las herramientas de corrección ortográfica.
Adicionalmente la descripción morfológica generada ha sido reutilizada en otras dos herramientas interesantes: versión *myspell* para el euskera y generador morfológico para un programa libre de traducción automática.

Datos del euskera para *myspell*

Debido a que *myspell* no puede gestionar todos los paradigmas generados, se han preparado los datos necesarios para *myspell* seleccionando los paradigmas más probables y añadiéndole a las raices una larga lista de palabras obtenidas de la siguiente forma:

1. Generación de formas más habituales a partir de los lemas más probables que no están en esos paradigmas previamente incluidos. Para hacer esta generación hemos utilizado el generador morfológico original restringiendo el poder generativo a un solo sufijo.
2. A partir de un gran corpus de euskera, hemos obtenido todas las formas correctas usando el corrector original y hemos detectado aquellas que *myspell* no podía reconocer, ni con los paradigmas más frecuentes, ni con la lista anterior. La lista obtenida se ha añadido.

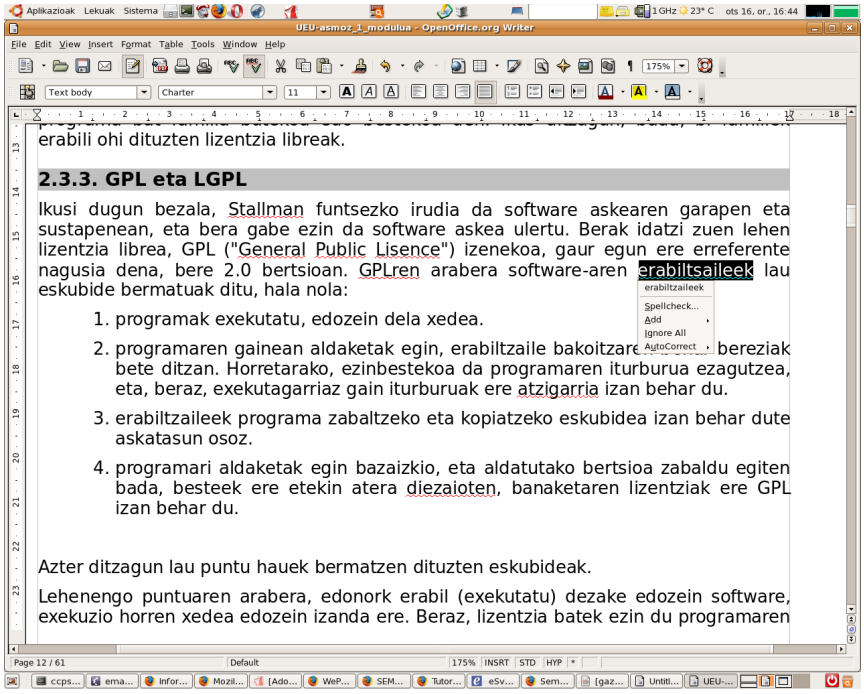
Esta versión de *myspell* la hemos hecho pública, ya que funciona bastante correctamente a pesar de que presenta la falta de coherencia comentada en el apartado 2. Hay que tener en cuenta que muchas herramientas, como las de *Mozilla*, no integran de momento *hunspell*.

Generación morfológica en *Matxin*

Además se ha usado la exportación conseguida, con información morfológica adicional, para integrarla en la generación morfológica de un ingenio libre de traducción automática de nombre *Matxin*[6] [3], que junto al ingenio de nombre *Apertium*[7] [5] (ver en estas actas) se han desarrollado dentro del proyecto *Opentrad*[8].
Aquí se demuestra lo conveniente de combinar corrección ortográfica con análisis/generación morfológica, ya que así a partir de un mismo desarrollo se pueden generar distintos productos. Esta misma idea está en el propio paquete *hunspell,* que desde su diseño ha apostado por la doble función.

Utilización

Los paquetes para el euskera en *hunspell* y *myspell* han sido desarrollados por el grupo IXA de Universidad del País Vasco y la empresa Eleka y se distribuyen bajo licencia GPL. Se pueden obtener en el sitio de descargas del Gobierno Vasco[9] y en los sitios correspondientes a las herramientas de software libre[10].
A continuación se presenta un ejemplo de la utilización en OpenOffice:



6 matxin.sourceforge.net
7 apertium.sourceforge.net
8 www.opentrad.org
9 http://www.euskara.euskadi.net/
10 http://www.librezale.org/mozilla/firefox/ entre otros

5. Mejoras y trabajo futuro

La versión generada para *hunspell* presenta dos limitaciones que pensamos deben ser mejoradas:

1. Al no haberse utilizado todas las potencialidades de *hunspell*, los datos todavía pueden mejorarse, sobre todo para conseguir una mayor compactación de los mismos y, por lo tanto, mayor eficiencia en su acceso. Para ello se pueden generar reglas fonológicas de eliminación/adición de caracteres al encadenar lemas y sufijos, con lo que se reduciría el número de paradigmas. Esto se puede realizar a muy corto plazo.
2. De todas formas, el uso de *hunspell* no resulta totalmente satisfactorio ya que no soporta hunspell más de dos encadenamientos de sufijos. Aunque esto lo hemos paliado alargando la lista de sufijos del segundo nivel, sería más conveniente permitir más niveles de encadenamiento de sufijos. Este es un trabajo a un plazo más largo.

También queremos poner en marcha un sitio de referencia para futuras mejoras y desarrollos relacionados.

En cualquier caso, de cara al futuro, sería conveniente, y creemos que este tema ya está planteado en la comunidad científica, la generación de un FLOSS de las mismas características expresivas de los transductores léxicos. Con esto se conseguirían procesadores morfológicos y correctores ortográficos de gran calidad y cobertura para un buen número de idiomas, a partir de los ya desarrollados utilizando este tipo de herramientas. Además, se permitirían definir reglas de corrección para errores de competencia léxica o reglas para errores en OCR, que harían la corrección más precisa.

Agradecimientos

El trabajo descrito en este artículo ha sido parcialmente subvencionado por Departamento de Política Lingüística del Gobierno Vasco y por el Ministerio de Industria dentro del programa PROFIT (FIT-350401-2006-5).

6. Referencias

1. Alegria I., Artola X., Sarasola K., Urkia M. Automatic morphological analysis of Basque. Literary & Linguistic Computing Vol. 11, No. 4, 193-203. Oxford University Press. Oxford. (1996)
2. Alegria I. Morfología de estados finitos Procesamiento del Lenguaje Natural (SEPLN) Revista no. 18, 1-26. Donostia (1996)
3. Alegria I., Díaz de Ilarraza A., Labaka G., Lersundi M., Mayor A., Sarasola K. Transfer-based MT from Spanish into Basque: reusability, standardization and open source. LNCS 4394. Cicling 2007. ISBN-10: 3-540-70938-X (2007)
4. Aldezabal I., Alegria I., Ansa O., Arriola J., Ezeiza N. Designing spelling correctors for inflected languages using lexical transducers. EACL'99 (1999)

5. Armentano-Oller C., Corbí-Bellot A., Forcada M., Ginestí-Rosell M., Bonev B., Ortiz-Rojas S., Pérez-Ortiz J., Ramírez-Sánchez G., Sánchez-Martínez F., "An open-source shallow-transfer machine translation toolbox: consequences of its release and availability", in Proceedings of OSMaTran: Open-Source Machine Translation, A workshop at Machine Translation Summit X (2005),
6. Beesley K.R. and Karttunen L. . *Finite State Morphology*. CSLI Publications, Palo Alto, CA. (2003)
7. Nemeth V., Tron P., Halacsy A., Kornai A., Rung I. Leveraging the open source ispell codebase for minority language analysis. Proceedings of SALTMIL (2004)