DANIEL CASTRO CASTRO*
ROCÍO LANNES LOSADA*
AURORA PONS PORRATA**
YUNIOR RAMÍREZ CRUZ**
Diseño de Aplicaciones, Tecnología y Sistemas (DATYS-SC)*
Centro de Estudios de Reconocimiento de Patrones y Minería de Datos**
Santiago de Cuba
Cuba
{daniel.castro,rocio.lannes}@sc.datys.co.cu, {aurora,yunior}@cerpamid.co.cu

Construcción de un corrector ortográfico automático

1 Introducción

Los errores ortográficos son diversos, en los documentos digitales se presentan la mayoría de los errores cometidos por una persona en una escritura en papel, pero a estos se agregan nuevos como resultado del uso de la computadora. Además si el texto no fue escrito y se empleó por ejemplo un scanner para digitalizarlo (Reconocimiento Óptico de Caracteres, OCR) entonces estaremos en presencia de otros errores. De manera general los errores se agrupan en dos conjuntos:

- Errores de contexto (Real Word)
- Palabras incorrectas (Non Word)

Los errores de contexto son los que generalmente encontramos en documentos reales donde por ejemplo vemos la no concordancia entre número o género (La niño, Los niño, etc.). Las palabras incorrectas son aquellas en las que falta un conocimiento, ejemplo, la no acentuación (camion, camión), la sustitución de letras (cambio de b por v, s por c) entre otros; pero a estos se suman otros. Estos errores ortográficos de palabras incorrectas los podemos dividir en 4 grupos:

- Sustitución
- Inserción
- Eliminación
- Transposición

Los errores se pueden dividir en más o menos grupos. Los de sustitución son por ejemplo los mencionados anteriormente; en los de inserción tenemos a aquellas palabras donde al teclearlas presionamos una tecla de más (cammión, camión); en los de eliminación lo contrario a la inserción (cmión, camión) y en los de transposición el intercambio de dos letras (cmaión, camión).

Todos estos podemos decir que son errores humanos, porque los comete una persona al redactar un documento. Algunos de estos se presentan en mayor o menor medida si se usan técnicas de OCR y se pueden sumar otros propios del proceso. Estos es solo una ilustración del gran conjunto de errores con los que se trabaja.

Además de los errores antes mencionados, para el desarrollo de una aplicación de corrección, se requiere conocer el ámbito en el que se va a emplear y los requisitos que queremos que se traten y se cumplan. Si queremos que solo se detecten, o ya detectados, se muestre un listado de sugerencias correctas para sustituir por la palabra incorrecta, o que se haga una corrección automática del error. En la figura 1 mostramos un resumen donde para cada etapa se puede construir una aplicación, o realizar una que cumpla con cada una.

En el presente artículo exponemos el desarrollo de un Corrector Ortográfico Automático como una herramienta integrante de una suite de Procesamiento del Lenguaje Natural (PLN) en desarrollo.

Como parte de esta suite se encuentran otras herramientas:

- Analizador Morfológico
- Analizador Sintáctico
- Reconocedor de Entidades
- Etiquetador
- Otros

Debido a la idea de la suite, nos damos a la tarea de construir un corrector propio capaz de integrarse a las demás aplicaciones y con desarrollo modular para que pueda incluir o no diferentes técnicas de detección y corrección. Debe permitir que las opciones de detección y corrección sean configurables para el uso de las herramientas de la suite u otras de terceros, la inclusión de aplicaciones con el mínimo de esfuerzo de programación, o el uso o no de las posibilidades que brinda en función de incrementar el tiempo de ejecución o el uso de la memoria.

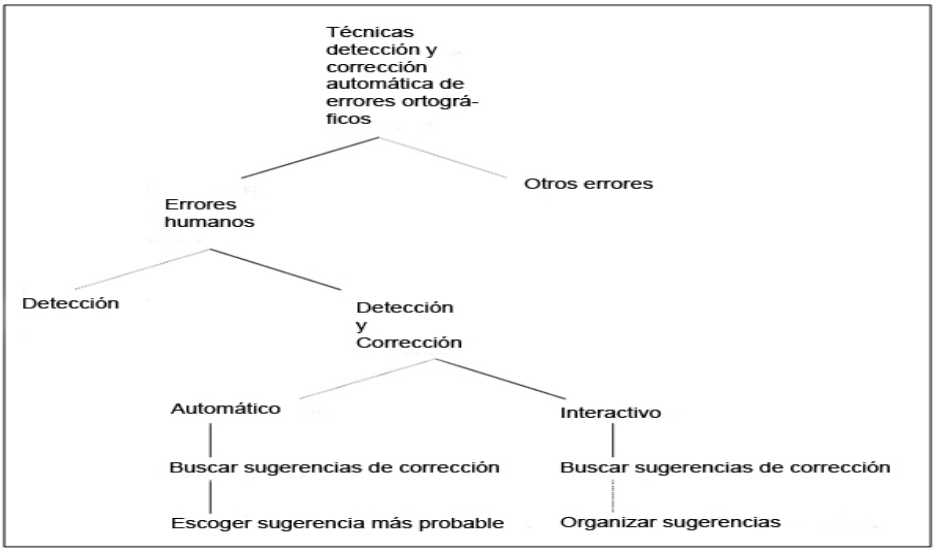

Fig. 1. Técnicas de detección y corrección.

La idea inicial planteada es la corrección ortográfica teniendo en cuenta el contexto. Como producto terminado contamos con la versión de corrección sin información de contexto que cumple con los requerimientos expuestos y que detallaremos en el transcurso del artículo. Expondremos además algunos resultados obtenidos y los experimentos realizados, teniendo en cuenta que realice análisis para diferentes idiomas, inicialmente el español, inglés, francés, italiano y portugués.

Este artículo se estructura de la siguiente manera: en la sección 2 la explicación del modelo de nuestro corrector y sus funcionalidades, con sub-secciones para cada módulo funcional del sistema y las técnicas de PLN empleadas. En la sección 3 se exponen los resultados y experimentos, para finalizar con las conclusiones y las referencias bibliográficas en las secciones 4 y 5 respectivamente.

2 Modelo del Corrector

Nuestro corrector se divide en dos partes fundamentales figura 2:

- Pre-procesamiento
- Corrección

En el Pre-procesamiento es donde se analiza el texto de entrada, para conocer su estructura y las partes que lo componen, se realiza una división por oraciones y estas por palabras. El análisis se realiza por palabras, ya finalizada esta fase, se procede a la detección.

En la Corrección, ya contamos con la fase inicial de detección y solo nos concentramos en aquellas palabras que no han sido reconocidas.

De manera general, contamos con 5 módulos independientes para permitir el uso de cada uno de estos cuando se requiera, a su vez en cada módulo se delimitan las diferentes funcionalidades con el objetivo de hacer configurables su uso. En el Pre-procesamiento tenemos los módulos de Procesamiento y Análisis-Detección, en la Corrección los módulos: Extra-Análisis, Sugerencias y Probabilidad.

Como entrada del corrector, podemos recibir la dirección de un documento o el texto plano; y como salida brindamos el texto corregido automáticamente o un listado con las sugerencias ordenadas por probabilidad para cada palabra no detectada.

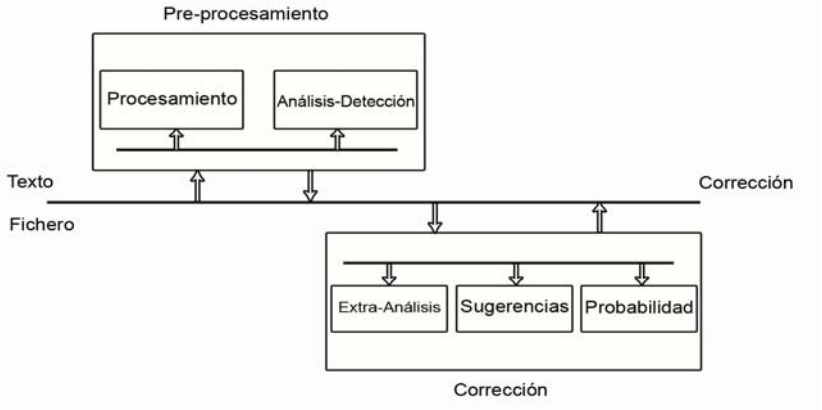

Fig. 2. Flujo y módulos del trabajo del corrector.

2.1 Módulo Análisis-Detección

En este módulo se realiza un análisis por palabras en cuanto a la detección de números, fechas y símbolos, búsqueda en el diccionario, reconocimiento de entidades, análisis de sufijos, abreviaturas y signos. En esta fase se determinan para cada palabra todas sus posibles categorías gramaticales de estar contenidas en el diccionario o en el análisis de sufijos.

2.2 Módulo Extra-Análisis

Teniendo en cuenta que cada idioma tiene sus particularidades e información que se obtiene o surge a diario, se hace necesario este análisis extra, en el cual toda la información es cargada en memoria, la misma es temporal y definida por el usuario y puede tenerse en cuenta a incluir en alguna nueva versión del sistema, para formar parte del módulo anterior.

2.3 Módulo Sugerencias

En este módulo hay tres fases:

- Detección de un error común
- Generación de sugerencias para un solo error
- Generación de sugerencias para más de un error

Inicialmente se analiza si la palabra no reconocida constituye un error común del idioma y en caso de ser positivo ya tendremos la sugerencia correcta. El listado de errores comunes es obtenido gracias al estudio de lingüistas, con la sugerencia correcta para cada uno de ellos.

De no ser un error común, se presupone que la palabra correcta se puede generar realizando un cambio en la no reconocida. Para esto se generan las sugerencias en cuatro etapas:

- Inserción
- Eliminación
- Sustitución
- Transposición

Permitiendo que sea configurable el orden en que se generan las sugerencias e incluso la no ejecución de etapas escogidas.

Cada sugerencia es filtrada por el análisis de trigramas que se van construyendo con el cambio. Si se genera un trigrama no probable en el idioma, entonces es rechazada la sugerencia.

Cada palabra se descompone en todos los posibles grupos de tres letras consecutivas que la componen, ejemplo:
camión: _ca, cam, ami, mió, ión, ón_ (el _ es el espacio vacío al principio o fin de palabra que nos permite identificar trigramas válidos en estas posiciones).

En cada uno de estos trigramas se realiza el cambio para cada letra una a la vez, si en la palabra cxmión hacemos la sustitución de la x por la s, tratando de encontrar una palabra correcta, entonces se generarían los siguientes trigramas:
csmión: _cs, csm, smi, mió, ión, ón_ .

Al analizarlos, se encuentra que el trigrama csm es no probable en nuestro idioma y de esa forma la palabra que se genera (csmión) no es considerada en las sugerencias.

Si se requiere de un análisis más detallado, se incluye la generación de sugerencias considerando que la palabra incorrecta es producto de la presencia de más de un error, para lo que se realiza un proceso parecido al anterior.

2.4 Módulo Probabilidad

Finalmente se agrupan las sugerencias y se analiza cuales de estas son palabras válidas. Para cada palabra válida se analiza su probabilidad de ser la sugerencia correcta según el cambio por el cual fue generada. Todas estas sugerencias son ordenadas en una lista por orden de probabilidad, donde se asume que la de mayor probabilidad es la escogida para sustituir a la incorrecta.

En el caso de la probabilidad para la Inserción, Eliminación y Transposición, fue necesario el estudio y cálculo de probabilidades de trigramas a partir de un conjunto de documentos variados que no presentaran errores ortográficos. Para el caso de la sustitución, se analiza la probabilidad de que la sustitución sea producto a un error de conocimiento, a una tecla de adyacencia o a una tecla cualquiera. Siendo esta probabilidad fijada gracias al criterio de un experto debido a que no contamos con un CORPUS de textos etiquetados con errores ortográficos.

3 Resultados y Experimentos

La aplicación se desarrolla con portabilidad tanto para Windows como Linux y con un mínimo de requerimiento de hardware.

Como parte de las pruebas realizadas hasta el momento, se han analizado textos en inglés, español e italiano, por contar con los diccionarios y los trigramas de los respectivos idiomas.

Para el caso de las pruebas en español, estas han sido más detalladas, con el análisis de la precisión y otras medidas en la detección para un total de 10 documentos con 5300 palabras entre todos.

Las medidas que se usan son precisión, relevancia (o recuperación) y F1.

La precisión mide la cantidad de errores detectados por el sistema que realmente son errores, o sea:

Prec=ErrDetReal/ErrDet

- ErrDetReal: Cantidad de errores detectados que realmente son errores
- ErrDet: Cantidad de errores detectados

La relevancia mide la cantidad de los errores existentes que el sistema pudo detectar, o sea:

Rel=ErrDetReal/ErrReal

- ErrReal: Cantidad de errores reales existentes y F1 es la media armónica de la precisión y la relevancia, o sea, si las dos son altas la F1 es alta pero si alguna de ellas baja F1 baja aunque la otra se mantenga alta.

F1 = (2 * Prec * Rel) / (Prec + Rel)

La precisión obtenida al analizar los documentos fue de un 87% debido fundamentalmente a la riqueza de los diccionarios por la presencia de palabras correctas que no pudieron ser detectadas, la relevancia de un 100% con lo que fueron detectados todos los errores que realmente lo son, para una media armónica F1 de 93.

Comparamos estos resultados con los obtenidos con la detección realizada sobre el mismo CORPUS por el corrector Hunspell, en este corrector, la precisión fue de un 68%, una relevancia del 98% y una armonía F1 de 82.

El caso de la detección es dependiente fundamentalmente de la riqueza del diccionario y las reglas de sufijos con las que se cuenta.

4 Conclusiones

En este artículo se ha presentado un recorrido por la estructura y funcionamiento de nuestro corrector, así como algunas pruebas realizadas. El corrector se puede emplear de dos maneras distintas: se ha desarrollado una librería para que se pueda incluir en otros proyectos, y como un servicio para posibilitar el uso de este por otras aplicaciones que no lo requieran como parte activa o sean de otra plataforma. Y realizamos estudios para paralelizar las funcionalidades sometidas a una mayor carga de ejecución.

A esto se le sumará el análisis de los errores de contexto, tan diverso y amplio como se pueda, el reconocimiento de textos multilingües tanto en un mismo documento como la entrada de un flujo de documentos de distintos idiomas entre sí, y el análisis de idiomas de carácter aglutinantes.

5 Referencias

Golding, A. R. Schabes, Y. Combining Trigram-based and Feature-based Methods for Context-Sensitive Spelling Correction. *Mitsubishi Electric Research Laboratories, 201 Broadway Cambridge, MA 2139.*

Golding, A. R. (1995). A Bayesian hybrid method for context-sensitive spelling correction. *In Proceedings of the Third Workshop on Very Large Corpora*. Boston. pp. 39-53,

Kukich, K. (1992). Techniques for automatically correcting words in text. *ACM Computing Surveys*. pp. 377-439.

Gale, W. A. Kenneth W. C. Yarowsky, D. (1993). A method for disambiguating word senses in a large corpus. *Computers and the Humanities*. pp. 415-439.

Verberne, S. (2002). Context-sensitive spell checking based on word trigram probabilities. *Master thesis Taal, Spraak & Informatica University of Nijmegen.*

Molina, A. (2004). Desambiguación en procesamiento del lenguaje natural mediante técnicas de aprendizaje automático. *Tesis doctoral. Universidad Politécnica de Valencia, España.*

Fossati, D. Eugenio, B. D. (2007). A Mixed Trigrams Approach for Context Sensitive Spell Checking. *CICLing 2007*. pp. 623-633.

Zribi, C. B. O. Mejri, H. Ahmed, M. B. (2007). Combining Methods for Detecting and Correcting Semantic Hidden Errors in Arabic Texts. *CICLing 2007*. pp. 634-645.