Algoritmos eficientes para detección temprana de errores y clasificación idiomática para uso en procesamiento de lenguaje natural y texto.

Andres T. Hohendahl[1] *José F. Zelasco*[1,2]

1 Laboratorio de Estereología y Mecánica Inteligente. Dto. de Ing. Mecánica Facultad de Ingeniería, Universidad de Buenos Aires, Paseo Colón 850, (1065), Buenos Aires, Argentina.

2 INTIA, Facultad de Ciencias Exactas, UNCPBA, Universidad del Centro de la Provincia de Buenos Aires, Campus Universitario, Paraje Arroyo Seco (B7000) Tandil, Prov. de Buenos Aires, Argentina.

Resumen

La temprana clasificación de idiomas y detección de errores gramaticales, juegan un rol fundamental tanto en el procesamiento de texto en lenguaje natural (Natural Language Processing - NLP) como en procesadores de texto convencionales. Para procesar una palabra gramaticalmente se requiere primero clasificarla, lo cual implica búsquedas en varios diccionarios para el caso de texto multilingual. Esto implica el uso de recursos importantes, en especial cuando las palabras no se encuentran, por tener errores de algún tipo.
Se ha sintetizado un conjunto de algoritmos sencillos, que utilizan las propiedades de las distribuciones de pares de letras de cada idioma. Logramos simultáneamente tanto la clasificación estadística como la detección temprana de errores gramaticales, ahorrando recursos en etapas posteriores.
Estos mecanismos proveen un rechazo estadístico de errores y poseen la ventaja de requerir escasos recursos de procesamiento, datos y memoria.
El sistema es apto para ser aplicado en las etapas iniciales de procesamiento de texto, mitigando la pesada tarea de búsquedas innecesarias y clasificaciones estériles, en etapas posteriores.

Antecedentes

Los textos en idiomas escritos alfabéticamente representan secuencias de fonemas reales, los cuales deben de ser pronunciables en esa lengua. Las lenguas tienen en general un conjunto acotado de fonemas y esto demarca la manera que estos idiomas *"suenan"*. Este hecho acota las posibles secuencias de fonemas presentes en uno y otro idioma.

También es conocido el hecho que ciertos fonemas no son posibles sino como transición entre otros bien determinados, tal es el caso de los fonemas fricativos, oclusivos y explosivos que suelen estar ligados o rodeados por los vocálicos y/o líquidos. Un ejemplo claro es que en español no se puede pronunciar una palabra sin vocales.

El hecho que los textos escritos sean mayormente una trascripción de sonidos reales producidos por el aparato fono-articulador, hace que subyazca una lógica de secuencia natural. El universo de palabras resultantes no contiene la combinación de todas las letras posibles en cualquier orden, sino solamente un subconjunto limitado de éstas, basado en las reglas del idioma en cuestión.

Simultáneamente, los errores de trascripción, ortográficos o de un sistema de conversión[7] de *"voz a texto"* suelen contener errores cuya distribución es predecible y muchas veces muy específica, como el caso de errores de digitación (por la distribución de letras en un teclado).

Estos errores inherentes o de método, generan palabras que muy frecuentemente caen fuera del universo de palabras existentes en ese idioma, siendo de este modo fácilmente detectables y corregibles con algoritmos estadísticos.

Solución propuesta

Se procedió a estudiar la distribución de pares de letras de tres idiomas: español, alemán e inglés, para poder crear uno a más algoritmos simples capaces de detectar en forma estadística y simultánea el idioma, posibles errores gramaticales y palabras agramaticales en un mismo paso.

Desarrollo del Trabajo

Se utilizó un diccionario español con alrededor de 48.000 palabras únicas[3], y se las clasificó conforme a las secuencias de 2 letras[1], hallando un interesante espectro de lugares 'vacantes'. Luego se analizó un diccionario similar en inglés y otro en alemán, con un número similar de palabras.

Analizando la frecuencia de pares de letras "*diletras*" en cada idioma, se halló que el universo de las existentes respecto a todas las posibles combinaciones es de solamente entre el 40 y el 60 % del total de combinaciones posibles. El análisis de estas frecuencias y sus distribuciones finalmente fueron candidatos para ser utilizadas en determinar si una palabra o frase

pertenece probablemente a un idioma o a ninguno.
Si tomamos el total de pares de letras "*diletras*" y lo llamamos ***d-es*** para el español, ***d-en*** para el inglés y ***d-de*** para el alemán, realizamos un análisis sobre un conjunto grande de palabras únicas en cada idioma, podemos hallar la frecuencia con que aparecen en cada diccionario. En este caso prescindimos el ponderar con la frecuencia promedio que esas palabras aparecen en el idioma.

Comparando las matrices de frecuencias ***d-es*** ***d-de*** nos hallamos con una zona en donde hay letras que se dan en ambos idiomas y otras que son exclusivas del español, como la "eñe" y otras del alemán, como la "ä". Igualmente se da la situación para el inglés que carece de acentos escritos. Conjuntamente, se halló que hay secuencias de letras que nunca se dan en español, otras que nunca se dan en ingles pero si en español, otras son únicas del alemán y así sucesivamente.

Esto permite crear un estimador que determine el idioma más probable al que pertenezca esa palabra y si es gramatical o no. Si la palabra desconocida contiene *diletras* imposibles para todos los idiomas considerados, puede ser etiquetada como no gramatical, ahorrando un importante proceso de búsqueda inútil en diccionarios o reconstrucción morfológica posterior. También es importante utilizar como significativo el caso de que ciertas letras nunca están presentes en ciertos idiomas, como las acentuadas en el inglés.

Desarrollo del Algoritmo

El funcionamiento del algoritmo, toma las palabras de entrada determinando el total de pares de letras y confeccionando un vector de frecuencias. Para esto solamente utiliza adición de enteros de 32 bits, por utilizar plataforma .NET (C#, MSIL), esto no genera errores de redondeo, para longitudes de frase de entrada de hasta varios cientos de palabras.

Se realiza el producto interno con los vectores característicos de cada idioma, normalizándolo luego, en punto flotante.

Finalmente se obtienen los valores de punto flotante (uno para cada idioma) en donde el mayor corresponde al del idioma más probable, el que le sigue al siguiente y así sucesivamente.

Se han incorporado algunas variantes en el algoritmo a fin de poder determinar en forma temprana la discordancia de palabras basado en un "umbral" de frecuencias mínimo, haciendo un producto interno no lineal, lo cual permite descartar palabras en forma inmediata, evitando el restante cálculo y proveyendo valores de significación más precisos a la hora de descartar una palabra.

Se incorporaron rutinas adicionales de separación en palabras, utilizando espacios y signos de puntuación comunes a los idiomas utilizados, para poder tratar frases de varias palabras.

Ponderando las frecuencias de *"diletras"* con la frecuencia de aparición de las letras individuales sobre un corpus de cada lenguaje, se obtiene un refuerzo en la selectividad del mecanismo, constituyendo un índice adicional para el caso de determinación del idioma.
Este mecanismo no resulta necesario para determinar si una palabra está mal escrita, ya que dentro de la creación del vector de frecuencias, se prevé una opción que detecta la ausencia o caída bajo un determinado umbral de frecuencia de alguna letra o secuencia, resultando en un inmediato rechazo de la misma.

Resultados

El algoritmo desarrollado, y sus variantes, finalmente permiten con apenas 30-40 sumas, comparaciones y multiplicaciones enteras más una de punto flotante, el obtener un valor que indica la posibilidad de que la palabra sea española, alemana o inglesa y/o ninguna de ellas, pasando a ser candidata a un trato no gramatical. El algoritmo entrega un valor extra, que indica el grado de significación de la muestra, basado en su diferenciación estadística.

Este algoritmo se incorporó como componente a un librería C#, utilizada por nosotros para numerosos procesos de NLP y fue bautizado ***PreSpell*** por su aplicación temprana al análisis ortográfico.

Trabajos Similares

Luego de revisar numerosa literatura, no se han hallado documentos que indican la utilización de este tipo de algoritmos para la reducción del uso de recursos para procesamiento tanto de texto como de lenguaje natural.
Hay trabajos[4] del área los cuales no responden a las técnicas empleadas en el presente trabajo pero sirven como referencia para el estado del arte de estas técnicas.
Los principales trabajos están inmersos en la problemática de trascripción de voz en texto (reconocimiento de voz) [5,6,7].

Se han hallado algunas referencias[2] a uso de *"vectores"* para detectar e indexar palabras, pero

con un uso y resultados diferentes a los presentados en el presente trabajo.

Conclusiones

El sistema propuesto ha resultado de suma utilidad ya que a pesar de que la selección realizada no es perfecta y posee los errores clásicos de toda estadística, las etapas posteriores de procesamiento, (en nuestro caso NLP), al tener en cuenta el etiquetado previo producido por el sistema, pueden tomar las decisiones más apropiadas. Tal es el caso de las abreviaturas que suelen ser detectadas como no gramaticales, pudiendo en este caso tratárselas de manera acorde, limitando la búsqueda a un diccionario especial y acotado para tales fines, ahorrando numerosos recursos.

El sistema trabaja con caracteres unicode. Se trabajó en especial con el subset Latin-1 (ISO 8851-1) por hallarse los idiomas utilizados completamente representados con este subconjunto de caracteres.

El hecho de trabajar en C# y .NET hace que el sistema sea extremadamente flexible y utilizable desde múltiples lenguajes de computación de la plataforma .NET, y posiblemente permita la portación de la misma a GNU MONO para unix/linux.

El actual trabajo, probó ser útil para la detección temprana de errores y clasificación de palabras brindando como resultado una importante disminución de procesamiento en nuestros sistemas de NLP, en especial recibiendo texto libre desde un teclado o desde la web.

Los archivos necesarios pesan apenas 2-3kbytes por idioma y el uso de cpu es extremadamente bajo, aunque su medición resulta difícil por lo corto de los procesos y las indeterminaciones del sistema, dado que el MSIL, realiza una pre-compilación inicial del código intermedio sobre las clases instanciadas.

Próximos Pasos

Se estudiará la posibilidad de potenciar los algoritmos con la utilización de datos adicionales teniendo en cuenta y ponderando la posición de los pares de letras, dentro de las palabras para determinar si mejora la calidad del reconocimiento.

También se realizarán mediciones de la eficacia del mismo, en un ambiente controlado, para determinar su capacidad de separar palabras erradas utilizando corpus conocidos en los tres idiomas considerados.

Resulta imprescindible poseer los derechos del uso de estos corpus, puesto que mucho de ellos son de carácter restringido y no son de libre acceso.

Estimamos que se puede mejorar el desempeño del algoritmo principal, utilizando las frecuencias de *diletras* pesadas por la frecuencia de aparición de ellas en un corpus general del idioma y no en un diccionario, o tal vez pesando cada conjunto de palabras por la frecuencia de aparición de la misma en textos generales, esto es candidato a una continuación del presente trabajo.

Comentarios

Hay claras evidencias de donde estos algoritmos son necesarios, tal es el caso de procesadores de textos populares como el MS-Word®, el cual cada vez que se edita texto o cambia algo, suele pasar un importante tiempo consumiendo hasta el 100% de cpu, marcando *(subrayando)* errores de gramática y ortografía e identificando idiomas de las palabras sus algoritmos internos, tornando lento y poco responsivo al sistema. Esto empeora significativamente, para documentos largos o si en él, abundan palabras y siglas desconocidas o no gramaticales, reiniciándose cada vez que hay un cambio o inserción.

Referencias Bibliográficas

[1] Se descartaron las palabras de una sola letra, por carecer de importancia y no representar problemas en búsquedas de diccionarios. Además suelen ser reemplazadas por búsquedas en memoria ya que las posibles palabras jamás exceden el número de letras del idioma.

[2] 2002, San José,CA, USA.*A Composite Approach to Language/Encoding Detection* Shanjian Li, & Katsuhiko Momoi, Global Customization Group, Netscape Communications, USA

[3] Los diccionarios utilizados fueron obtenidos de GNU Open Office, pueden ser descargados libremente de: http://lingucomponent.openoffice.org/spell_dic.html

[4] Huerst, W., Yang, J. & Waibel, A. (1998), Interactive Error Repair for an Online Handwriting Interface, in Proceedings of ACM CHI'98 (Poster), pp.353-354.

[5] Suhm, B. (1997), Empirical Evaluation of Interactive Multimodal Error Correction, in IEEE Workshop on Speech recognition and understanding, IEEE, Santa Barbara.

[6] Ainsworth, W. A. & Pratt, S. R. (1992), ``Feedback Strategies for Error Correction in Speech Recognition Systems'', International Journal of Man-Machine Studies 36(6), 833-842.

[7] Mankoff, J. and Abowd, G. D. Error Correction Techniques for Handwriting, Speech, and other ambiguous or error prone systems. GVU TechReport GIT-GVU-99-18. June 1999.