Procesamiento de Lenguaje Natural Robusto

Andrés T. Hohendahl [1,2]
andres.hohendahl@fi.uba.ar

1 *Laboratorio de Estereología y Mecánica Inteligente, Facultad de Ingeniería, UBA.*
2 *IIBM (Instituto de Ingeniería Bio Médica), Facultad de Ingeniería, UBA.*
Personal-Blog: http://web.fi.uba.ar/~ahohenda

INTRODUCCIÓN

El ingreso de texto en lenguaje natural ha resultado en una de las principales actividades masivas en el ingreso de datos de las últimas décadas; impulsado recientemente por la conectividad masiva y la creciente interacción de los usuarios en la web 2.0, foros, blogs, redes sociales, chat y mensajes cortos móviles; coincidiendo con electrónica de consumo cada vez más sofisticada. Los sistemas de PLN (**P**rocesamiento de **L**enguaje **N**atural) llamado NLP (*en inglés*) procesan ese texto "plano" generalmente producido por humanos. Hoy más que nunca con la globalización e informatización este texto ingresado posee una extensa terminología no acotada y suele en consecuencia contener innumerables errores de todo tipo y etiología. A la vez la constante miniaturización de la electrónica, acompañada por su baja de costos y el aumento paulatino de su potencia de cómputo (*ley de Moore*) permitieron a la industria crear nuevos dispositivos electrónicos cada vez mas inteligentes para toda clase de usos, requiriendo texto como entrada preferida.

Un creciente sinnúmero de nuevas funciones de estos dispositivos electrónico-informáticos requiere del ingreso de órdenes cada vez más complejas, muchas de las cuales se realizan hoy en forma mecánica y usando sensores especializados (*mouse/trackball/tacto/luz*), mientras que el habla directa con conversión voz a texto aún no es lo suficientemente robusto para ser usado masivamente, queda por último el teclado alfanumérico como el elemento más usual y robusto para el ingreso de datos y posibles órdenes precisas y complejas en forma de texto.

La difusión de estas nuevas tecnologías y el compromiso que presentan por su reducido tamaño, el ingreso de texto en estos dispositivos resulta cada vez más engorroso y problemático, multiplicando el número de errores que aparecen asociados. Por esto se requiere de nuevas estrategias y métodos para abordar los complejos problemas derivados de los errores de escritura. Esto se hace más destacado cuando se procesa lenguaje natural, pues torna potencialmente problemáticos los errores de escritura, como un ejemplo simple: los comandos de sistemas operativos y lenguajes de computación tradicionales, no toleran el más mínimo error.

Sin duda habrá una mayor necesidad de estas técnicas cuando la electrónica de consumo pueda dialogar plenamente con el usuario en lenguaje natural, habiendo numerosas y marcadas evidencias de esta tendencia en la industria.

PROCESAR TEXTO

El propósito de los sistemas de PLN es obtener alguna información específica o datos a partir de texto plano (sin características de formato). Ese texto básicamente es esperado ser ingresado desde un humano, he aquí la situación actual: los humanos se comunican

mediante palabras ensambladas en estructuras, existiendo dentro de éstas una cantidad de elementos de ligadura como la sintaxis, coexistiendo con una fuerte semántica subyacente para dar sentido a nuestros actos ilocutorios en forma de texto escrito.

Si bien el estado del arte del PLN permite hoy procesar un texto en forma decente (*no preciso ni profundizo pues daría para un libro completo*), éste estado del arte choca contra un patrón muy frecuente culturalmente cuando se trata de escribir texto: los errores de escritura de los que estamos hablando al amparo de la enorme variedad de texto viable, con inclusiones multiculturales, acrónimos, fórmulas y palabras prestadas de múltiples idiomas (‘*loan words*’) además de una sinnúmero de términos nuevos o parasintéticos de uso muy frecuente.

Estos errores son muy difíciles de detectar y pueden ser de muchos tipos y de hecho los más frecuentes son los involuntarios y los que ocurren por falta de conocimiento o base cultural, como ser el saber si un determinado nombre propio se escribe con cierta combinación extraña de letras no frecuentes en nuestro idioma habitual.

Otros errores se centran en fenómenos como la mecánica de escritura, por ejemplo teclados demasiado chicos como para el dedo ‘demasiado grande para la teclita’ de un usuario promedio. Algunos fabricantes están usando mecanismos tan sutiles como los teclados ‘táctiles’ capacitivos que procesan por válido un mero roce imperceptible por el desafortunado escritor.

Esto en su conjunto da como resultado una enorme cantidad de fuentes de error coexistentes y difíciles de tipificar, creando una dificultad creciente en el intento de interpretación por PLN de ese texto virtualmente ‘sucio’ y sin control alguno.

Un sistema de ingreso de datos para texto debiera de ser tolerante e inteligente, en especial si está orientado a lenguaje natural. Demostrado está el éxito y difusión de técnicas predictivas como el T9, usadas para el ingreso de texto en teclados numéricos de celulares, controles remotos, televisores inteligentes, reproductores y cámaras de foto.

Sin embargo la popularidad de micro-teclados completos tipo QWERTY en los dispositivos modernos como los teléfonos inteligentes o *smartphones*: *IPhone, Blackberry y Android* (*entre muchos otros*) ha multiplicado la cantidad de fuentes de error para el texto, y de hecho dado que el humano es muy robusto en su lectura, el error pasa por alto y termina impactando negativamente en la calidad del texto creado, creando una obvia dificultad creciente de su procesamiento automático.

TRATAR DE ENTENDER ‘ESO’

Se busca que la reconstrucción de palabras resulte similar a la de un humano promedio sin contexto, semántica ni conocimientos gramaticales. Es sabido que la lectura es un proceso muy robusto, apoyado por la comprensión semántica, el conocimiento previo y estimación contextual. De hecho una persona puede leer fácil y rápidamente un texto con un asombroso porcentaje de letras mal escritas, permutadas u omitidas. Esto último evidencia que en el lenguaje humano debe existir uno o varios patrones subyacentes con la redundancia suficiente para permitir la creación de mecanismos exitosos para la correcta reconstrucción de datos escritos, eliminando o acotando los errores.

Veamos la problemática en detalle: para entender y procesar algo escrito en lengua, es necesario clasificar las palabras y grupos de símbolos conforme algún patrón de formas escritas. En Lingüística la clasificación morfológica (CM) es un patrón primordial y útil para realizar tareas de NLP como ser: estadísticas, análisis sintáctico, semántico, desambiguación de sentidos (WSD) hasta comprensión artificial (NLU), pasando por retórica y pragmática.

Ahora, es sabido que el éxito o fracaso de toda tarea encadenada depende del eslabón más débil, y como la CM es uno de los eslabones obligados y más débiles por los errores y la cantidad de términos y palabras fuera de vocabulario (OOV), términos extranjeros, siglas, abreviaturas, acrónimos, fórmulas, etc.

Esto es bastante complejo especialmente para idiomas altamente flexivos como el español y muchos de los idiomas derivados del latín en general. Ni hablar de idiomas con declinaciones como el alemán y el latín, en donde se conjugan más elementos de derivación que generan ambigüedades y dificultad de un adecuado reconocimiento.

La CM involucra la tarea inicial de segmentación del texto (*tokenización*) con reconocimiento de entidades usuales en escritura y formatos de números y fechas, esto conlleva muchos problemas que a menudo se menosprecian, dado que el humano los resuelve instantáneamente y casi sin darse cuenta.

Solo a la hora de tratar de hacerlos con un programa de computación es que se vuelven tangibles y reales como problema, siendo extremadamente duros de resolver pues la mayoría resultan de orden combinatorios y parecen irresolubles óptimamente en un principio.

El desafío de hoy es poder procesar este texto “sucio” y obtener algo útil a partir de él, éste desafío no parece para nada trivial y de hecho no lo es.

ANTECEDENTES E INTENTOS DE SOLUCION

Si analizamos la literatura, *David Yarowsky* en sus trabajos de lingüística [13], plantea enfáticamente la necesidad de reconstrucción ortográfica. Por otro lado *Martín Reynaert* desarrolla en el 2004 [14] un sistema con esta finalidad para inglés y francés, obteniendo un resultado útil pero escaso en performance y requiriendo una cantidad de recursos importantes.

Grupos de España (UPC) crearon la librería *Freeling* [15] que contiene un analizador morfológico básico poco robusto que usa muchos recursos, siendo el único desarrollo abierto aplicado al español y que anda decentemente, mas no bien.

El gigante *Microsoft* embebió en el procesador de textos más famoso: *Word* un corrector ortográfico propietario del cual no hay especificación cierta, ni mediciones de calidad ni documentación alguna disponible.

Queda claro que aún hacen falta estándares claros y corpus anotados, para la medición de la bondad y el adecuado contraste entre todos estos sistemas de corrección y los por venir.

NUESTRA APROXIMACION

Hemos atacado este problema obteniendo un sistema interesante, capaz de reconocer cientos de millones de palabras inclusive inexistentes (*parasintéticas*) para lenguajes altamente flexivos como el español, con la capacidad de reconocer OOV y lidiar con numerosos errores de todo tipo. Un aspecto importante de este enfoque es la utilidad práctica: mientras muchos experimentos y desarrollos se hacen en condiciones controladas (vocabulario acotado y ningún error) la verdadera utilidad de un sistema de NLP se verá en el campo, es decir lidiando con errores y términos desconocidos.

Nuestra tarea ha sido la de "humanizar" la clasificación, dotando a este sistema algo de la robustez humana en la capacidad de no solo realizar una clasificación morfológica completa incluyendo OOV, sino hasta discernir entre palabras impronunciables, errores deliberados de tipeo, siglas alfanuméricas y palabras extranjeras, estimando de paso el idioma, extrayendo números, fórmulas matemáticas y reconocimiento de entidades simples frecuentes, unidades, abreviaturas, locuciones, números en palabras, etc.

REPARACION DE ERRORES

La problemática a resolver es de tipo NP dura, de naturaleza combinatoria, lo cual plantea un formidable y duro desafío pues están incluidos entre los problemas decididamente irresolubles por definición, agravados por la restricción de recursos en un sistema electrónico de cómputo de costo moderado y uso masivo del tipo de una PC.

Yendo al problema, etiquetar un texto es sencillo si está bien escrito, aunque reparar un error parece fácil y trivial, y demostraremos que no lo es con un simple ejercicio de razonamiento lógico (omitiremos el detalle matemático): Tenemos una palabra de N letras, la cual no está presente en el diccionario que tenemos al nuestro alcance como una base de datos, asumiendo por un instante que hay sólo 35 letras diferentes para cada una de las posibles posiciones, y que podríamos haber errado una letra, o haberla omitido, o tal vez puesto una de más o hasta invertido un par de ellas, si nos limitamos esto y la palabra es de 7 letras el número de opciones para restaurarlas asciende a mas de 462 intentos de una sola letra, pero si asumimos que el error puedan ser 2 caracteres, el número asciende rápidamente a 200 mil, y en el caso de querer probar arreglar muchas más letras, el número de pruebas asciende tanto que por más de tener la más veloz computadora disponible, ésta requeriría desde un par de años hasta algunos milenios dedicada su cálculo, con el agravante que muchas de esas permutaciones de letras intentadas, seguramente resulten en una palabra existente en nuestros diccionarios.. pero una seria pregunta subyacente sigue en pié: ¿es ésta la permutación que el humano quiso escribir?.

Para poder contestar o intentar resolver esto se necesita una métrica 'humana' y eso es lo que precisamente hicimos en un previo trabajo [17], en donde diseñamos una métrica aplicable a la similitud de formas escritas, la cual simplifica enormemente al convergencia hacia una solución al error de modo mucho más viable, en términos humanos.

TEORIAS INVOLUCRADAS

El modelo utilizó métricas y reglas derivadas del idioma español [2], el cual resulta altamente flexivo y derivativo, formando un sinnúmero de palabras, incluyendo un

importante número de palabras ‘inventadas’ a diario, llamadas parasintéticas [12], resultando en una enorme variabilidad y riqueza expresiva, compleja de procesar.

Para esto se usaron modelos extractados de los diccionarios y las reglas de escritura [3], también se usó la formación de palabras conforme a sus teorías básicas [2].

Las teorías utilizadas además de las lingüísticas, fueron la de mediciones y errores en comunicaciones del tipo de símbolos en electrónica y su estadística. Utilizamos fuertemente el reconocimiento de patrones de símbolos como los caracteres [11], luego se utilizaron métodos de aprendizaje automático supervisado para entrenar clasificadores, usados frecuentemente en inteligencia artificial [9].

Las teorías más usadas fueron las Markovianas y especialmente las de análisis de frecuencias de bigramas y trigramas, conjuntamente con estimación Bayesiana, junto con las teorías de grafos, aportaron algoritmos para optimización en búsquedas [6]. Estas teorías se usaron para conformar los módulos de reconocimiento y estimación del idiomas de palabras desconocidas, basado en su escritura por discriminación con los rasgos fonéticos y grafémicos, sin tener un diccionario y solamente basado en la evidencia de su estructura de frecuencias de N-gramas respecto a un corpus. Se usaron en consecuencia métricas basadas en el modelo planteado y su estadística ponderando la pertenencia de un texto a un idioma [16]. Se desarrollarán métricas alternativas para poder medir y estimar el error de un texto, basado en parámetros fonéticos [1] y las diversas secuencias de 2 y 3 grafemas.

Luego hemos hecho una extensiva búsqueda y pruebas sobre estructuras óptimas y eficientes en uso de memoria y tiempo de proceso, usando a la vez algoritmos eficientes para la búsqueda y almacenamiento del tipo descripto en [5] y [6], técnicas de inteligencia artificial [9], infraestructura para ingeniería de textos [10] y modelos de lenguaje [8], usados en procesamiento del habla con las teorías lingüísticas de etiquetado y toda la base de lenguajes formales y gramáticas libres de contexto basados en las teorías de *Noam Chomsky*,

En las estrategias y algoritmos se tuvo en cuenta el hecho de que los recursos son limitados y se usaron técnicas de compresión de afijos descriptas en [7] con estructuras óptimas compactas como árboles de prefijos y ternarios [5] permitiendo de este modo que estos modelos y algoritmos sean óptimos para los sistemas modernos.

Finalmente usamos la modelización del lenguaje como código de comunicación robusta, lematización y flexión [4], conjuntamente con una teoría de generación y percepción humana del habla basada en el aparato fonoarticulatorio para la evaluación y/o creación de métricas para estimar similitud perceptiva de formas escritas [17].

Para el etiquetado morfológico se utilizó una variante de etiquetas llamadas EAGLES 2.0, ampliada para permitir etiquetado semántico. [19]

Se utilizaron lenguajes de programación modernos que posen compiladores tanto para plataformas de cómputo de servidores como de escritorio, móviles y embebidos.

RESULTADOS

El sistema resultante está programado en C# y sus diversos módulos forman parte de un servidor léxico multilingüe. Fue aplicado en la creación de agentes inteligentes de diálogo en lenguaje natural, algunos diccionarios flexivos, traductores y asistentes matemáticos que ya funcionan en la Argentina contra números cortos móviles.

Se han realizado pruebas de lematización robusta sobre texto proveniente de blogs de inherente mala calidad, logrando un 97.5% de reconocimiento, el 2.5% restante eran palabras que no tenían sentido alguno incluyendo puntuaciones agrupadas, direcciones web (url's) y de correo electrónico las cuales no responden a formato lingüístico alguno.

Como ejemplo se mostrará en l a **Tabla 1** muestra la lematización real de un texto decididamente mal escrito, mostrando que las alternativas escogidas por el sistema coinciden en un 100% con lo que un humano hubiese decidido al tratar de entender este texto. Las etiquetas son Eagles 2.0, el lema o raíz morfológica se halla precedida de un asterisco entre signos de mayor y menor, ej.: <*venir>los números como ser ~0.99 indica la verosimilitud (-1, 0, +1) de la forma analizada, mientras que p:0.036 indica la probabilidad de esa forma calculada por su frecuencia relativa de aparición en un corpus balanceado. Si ay mas de una alternativa, éstas se separan con el carácter | 'pipe' usado como 'o' exclusiva.

Tabla 1

vimo hoi ezpe kavrom kon eza vayema ozpitalaria ke nempekapo salu2 klhdrdzkuio

vimo vino_VPIS3SM<*venir>~0.98 p:0.00000019 | vimo_Xes<*vimo>~0.83 p:0.036
vino_VIIS3SM<*venir>~0.98 p:0.00000019 | vino_NCMS~0.98 p:1
hoi hoy_RT~0.99 p:0.0000048 | ohm_NCMS0h~0.95 p:0.0000048 | Ho_NPMS~0.55 p:0.9 |
hot_AQ0MS~0.89 p:0.0000048 | hoi_Xen<*hoi>~0.84 p:0.9
ezpe éste_PD0MS~0.98 p:0.03 | este_DD0MS~0.98 p:0.9 | este_NCMS~0.98 p:0.0075 |
este_PD0MS~0.98 p:0.03 | Xes<*ezpe>~0.93 p:0.0075
kavrom cabrón_AQ0MS~0.96 p:0.76 | cabrón_NCMS~0.96 p:0.23 | Xde<*kavrom>~0.84 p:0.076
kon_con_SPSMS<%mode>~1 p:0.08 | kon_Xde<*kon>~0.9]~0,997
eza esa_PD0FS<*eso> p:0.076 | esa_DD0FS<*ese> p:0.9 | Xes<*eza>~0.93 p:0.076
vayema [ballena_NCFS0z~0.99 p:0.19 | ballena_AQ0FSL<*balleno>~0.99 p:0.62 |
ballena_NCFS0L<*balleno>~0.99 p:0.19 | vayema_Xes<*vayema>~0.88 p:0.062
ozpitalaria [hospitalaria_AQ0FS<*hospitalario>~1 p:0.039 | ozpitalaria_Xes<*ozpitalaria>~0.88]~0,995
ke _que_CS~0.99 p:0.011 | qué_PE0NS~0.99 p:0.032 | qué_DE0CN~0.94 p:0.000017 |
qué_PT0CNN~0.94 p:0.0015 | que_PR0CNN~0.94 p:0.009 | Xes<*ke>~0.55
nempekapo mentecato_AQ0MS~0.95 p:0.76 | mentecato_NCMS~0.95 p:0.23 |
Xen<*nempekapo>~0.789 p:0.076
salu2 saludos_NCMP<*saludo> | p:0.09 Ka<*salu2> p:0.9
klhdrdzkuio _B(BadWord)~0,875

DESEMPEÑO y USO DE RECURSOS

Respecto al uso de recursos de procesador y memoria, el sistema utiliza un máximo de 40 megabytes para 82 mil palabras raíz en español, con 15 mil nombres propios, conformando un universo de palabras reconocibles exactas de más de 5.2 millones, mientras que el número de palabras combinables, derivadas y fuera de vocabulario es prácticamente indeterminado, con una cota según alguna literatura en el orden de 3000 millones de palabras [18].

La velocidad de etiquetado es dependiente del tipo de palabra, si la palabra está o no en su diccionario y en caso contrario la dificultad en su reconstrucción. Contando con palabras existentes la velocidad es de 2500 palabras por segundo [20] mientras que cuando las palabras son inexistentes o poseen errores la velocidad baja a 15 por segundo en promedio. Debido a que el sistema posee un mecanismo de memoria de proceso llamada "cache" la velocidad promedio de un sistema usado regularmente mejora con el tiempo pues almacena palabras etiquetadas de uso frecuente, a medida que le son solicitadas.

CONCLUSIONES y TRABAJO FUTURO

El analizador léxico descripto responde a una variedad de palabras y formatos, incluyendo números romanos, formatos numéricos científicos, números dichos con palabras, ordinales, fraccionales y multiplicativos en español e inglés. Si bien la potencia de flexión del inglés es escasa, se pudieron expresar apenas 34 reglas de sufijación contra las más de 3600 del español, se está trabajando en agregar nuevos prefijos para permitir etiquetado semántico por presunción de conformación derivativo-semántica.

El presente trabajo conformará una librería léxica para ser utilizada en sistemas de procesamiento de lenguaje natural multilingüe, para ser usado en sistemas de diálogo.

Se ampliará con un tesauro español (ya está bajo pruebas) y un sistema de flexión y derivación artificial morfológico, permitiendo no solo lematizar sino flexionar palabras incluyendo la posibilidad de crear palabras parasintéticas, el problema subyacente es que la derivación y flexión rigurosa debe ser específica para cada tipo de palabra gramaticalmente hablando, mientras que esta librería podría virtualmente transformar palabras y lograr crear por ejemplo un adverbio a partir de un verbo.

BIBLIOGRAFIA

[1] "Lawrence Phillips, Hanging on the Metaphone", Computer Language, vol. 7, no. 12, pp. 39-43, 1990.

[2] Relaciones morfoléxicas prefijales del español. Santana, O.; Carreras, F.; Pérez, J.; Rodríguez, G. Boletín de Lingüística, Vol. 22. ISSN: 0798-9709. Jul/Dic, 2004. 79/123.

[3] "Diccionario de la Lengua Españols, 22ª edición" , http://buscon.rae.es/draeI/, 6/2011

[4] FLANOM: Flexionador y lematizador automático de formas nominales. Santana, O.; Pérez, J.; Carreras, F.; Duque, J.; Hernández, Z.; Rodríguez, G. Lingüística Española Actual XXI, 2, 1999. Ed. Arco/Libros, S.L. 253/297

[5] "Ternary Search Tree" http://xlinux.nist.gov/dads//HTML/ternarySearchTree.html, 6/2011

[6] Mehlhorn, K. Dynamic Binary Search. SIAM Journal on Computing 8, 2 (May 1979), 175-198.

[7] "ASPELL", http://aspell.sourceforge.net/man-html/Affix-Compression.html

[8] "The Noam Chomsky Website", http://www.chomsky.info/, 6/2011

[9] "Inteligencia Artificial Modelos Técnicas y Áreas de Aplicación", Francisco Escoleano, Miguel angel Cazorla, María Isabel Alfonso, Otto Colomina, Miguel Ángel Lozano, Thompson Editores Spain Paraninfo s.a., 2003, ISBN 84-9732-183-9

[10] "GATE: General Architecture for Text Engineering, http://gate.ac.uk/, 6/2011

[11] "JLEX: Automatic Lexer Generator for Java", www.cs.princeton.edu/~appel/modern/java/JLex/, 6/2011

[12] "Parasynthetic Morpholexical Relationships Of The Spanish: Lexical Search Beyond The Lexicographical Regularity" Octavio Santana Suárez , Francisco J. Carreras Riudavets , Jose R. Pérez Aguiar , Juan Carlos , Rodríguez Pino; http://acceda.ulpgc.es/bitstream/10553/471/1/3689.pdf , 06/2011

[13] David Yarowsky (publicaciones) http://www.cs.jhu.edu/~yarowsky/pubs.html, 06/2011
[14] Martin Reynaert, 'Proceedings of the 20th international conference on Computational Linguistics' 2004
[15] Freeling http://nlp.lsi.upc.edu/freeling/ , 06/2011
[16] "Algoritmos eficientes para detección temprana de errores y clasificación idiomática para uso en procesamiento de lenguaje natural y texto", Hohendahl, A.T.; Zelasco J. F., WICC2006 - ISBN 950-9474-35-5
[17] "Desarrollo de un algoritmo para la medición del grado de similitud fonológica entre formas escritas" , Hohendahl, A.T.; Zanutto, B. S.; Wainselboim, A. J.; SLAN2007. X Congreso Latinoamericano de Neuropsicología 2007, Buenos Aires, Argentina
[18] "Parasynthetic Morpholexical Relationships Of The Spanish: Lexical Search Beyond The Lexicographical Regularity" Octavio Santana Suárez , Francisco J. Carreras Riudavets , Jose R. Pérez Aguiar , Juan Carlos , Rodríguez Pino;
http://acceda.ulpgc.es/bitstream/10553/471/1/3689.pdf , 06/2011
[19] "Etiquetas EAGLES", www.lsi.upc.es/~nlp/tools/parole-sp.html, 6/2011
[20] Sistema de Referencia es un CPU Intel x386 de escritorio E5300 a 2.54 GHz, con 4Gb de RAM corriendo XP Pro de 32 bits, usando un solo procesador (el sistema no es multiproceso).