UCES Subjetividad y Procesos Cognitivos, Vol. 14, Nº 2, 2010
Pág. 20-31, ISSN impreso: 1666-244X, ISSN electrónico: 1852-7310

Insights lingüísticos relativos a la normalización léxica de contenidos generados por usuarios

Linguistic insights on the lexical normalization of user-generated content

Laura Alonso Alemany*

Resumen

Presentamos trabajo en progreso acerca de la normalización de palabras para contenidos generados por usuarios. El enfoque es simple y ayuda a reducir el volumen de anotaciones manuales características de enfoques más clásicos. Primero, agrupamos las variantes ortográficas de una palabra, mayormente las abreviaturas. De estos ejemplos agrupados manualmente aprendemos un clasificador automático que, dada una palabra no vista anteriormente, determina si es una variación ortográfica de una palabra conocida o si es una palabra totalmente nueva. Para lograr eso, calculamos la similitud entre la palabra no vista y todas las palabras conocidas, y clasificamos la nueva palabra como una variante ortográfica de su palabra más similar. El clasificador aplica una medida de similitud de secuencia de caracteres basada en la distancia de edición Levenshtein. Para mejorar la exactitud de esta medida, le asignamos a las operaciones de edición un costo basado en el error. Este esquema de asignación de costos apunta a maximizar la distancia entre secuencias similares que son variantes de diferentes palabras. Esta medida establecida de similitud alcanza una exactitud de .68, una importante mejoría si la comparamos con el .54 obtenido por la distancia Levenshtein.

Palabras clave: normalización de palabras, palabras no vistas, secuencias de caracteres, variantes ortográficas.

Summary

We present work in progress on word normalization for user-generated content. The approach is simple and helps in reducing the amount of manual annotation characteristic of more classical approaches. First, orthographic variants of a word, mostly abbreviations, are grouped together. From these manually grouped examples, we learn an automated classifier that, given a previously unseen word, determines whether it is an orthographic variant of a known word or an entirely new word. To do that, we calculate the similarity between the unseen word and all known words, and classify the new word as an orthographic variant of its most similar word. The

* NLP Group. FaMAF, UNC, Córdoba, Argentina. E-mail: alemany@famaf.unc.edu.ar

classifier applies a string similarity measure based on the Levenshtein edit distance. To improve the accuracy of this measure, we assign edit operations an error-based cost. This scheme of cost assigning aims to maximize the distance between similar strings that are variants of different words. This custom similarity measure achieves an accuracy of .68, an important improvement if we compare it with the .54 obtained by the Levenshtein distance.

Key words: word normalization, unseen words, strings, orthographic variants.

1. Introducción y motivación

En tiempos recientes ha habido un incremento significativo en textos informales generados por los usuarios, donde los usuarios utilizan en gran medida palabras personalizadas: abreviaturas, acrónimos, salteo de vocales, números en lugar de letras, etc. Buenos ejemplos de eso son los blogs comunitarios, los mensajes cortos de texto enviados por celular, los estados de los usuarios en las redes sociales, anuncios o listas de subastas.

Este contenido generado por los usuarios provee una creciente cantidad de información privilegiada para una variedad de metas, que oscilan entre reseñas de productos o servicios y vigilancia de epidemias. Entonces, ser capaz de procesar automáticamente este tipo de textos es tanto prometedor como necesario. Prometedor debido a la información rica que acarrea, usualmente es información que no puede encontrarse en otro lugar. Necesario porque la cantidad de textos generados por el usuario es grande y está en crecimiento, y también es diferente del lenguaje estándar y rápidamente cambiante, lo que hace que sea inviable tratarlo manualmente.

Grandes cantidades de textos son usualmente pre-procesadas con herramientas estándar de Procesamiento del Lenguaje Natural (PLN) para obtener alguna abstracción lingüística del texto crudo. No obstante, las herramientas estándar de PNL no pueden aplicarse directamente a textos generados por el usuario debido a que presentan muchas diferencias respecto al dialecto estándar de los lenguajes, en el nivel léxico, sintáctico e incluso semántico. Esto hace que sea muy difícil extraer información automáticamente de ellos o aplicar algún otro tipo de tratamiento automático más complejo, como traducción automática (AiTi et al., 2005) o alimentar una base de datos (Michelson y Knoblock, 2006).

Una estrategia usual para el tratamiento automático de estos mensajes es convertirlos (o "traducirlos") en sus equivalentes dentro del dialecto estándar. Esas versiones traducidas pueden ser tratadas exitosamente por herramientas estándar para el Procesamiento del Lenguaje Natural. Un valor agregado de esta estrategia es que provee información acerca de la relación entre el dialecto estándar y las próximas variantes. Además de esto, si utilizamos métodos de máquinas de aprendizaje no supervisado,

podemos incorporar automáticamente nuevos mensajes en la estrategia, y así seguirle el ritmo a la rápida evolución de las nuevas variantes.

La primera tarea necesaria para aplicar las herramientas de PLN es normalizar el vocabulario en estas aplicaciones. Con frecuencia, la normalización del vocabulario para este tipo de texto se hace con diccionarios manualizados de abreviaturas y variantes[1]. No obstante, la variación en este tipo de texto es muy alta, y nuevas variantes de palabras y nuevas maneras de expresión son producidas rápidamente. En este contexto, un enfoque más automatizado se vuelve necesario para mantenerse el ritmo de la evolución del lenguaje.

En este trabajo aplicamos un método basado en la distancia de edición para normalizar palabras no-estándar. Nuestra distancia de edición debe ser capaz de diferenciar entre operaciones de edición discriminativas y no discriminativas. No todas las operaciones de edición son iguales así como tampoco son iguales todos los contextos, para que tenga lugar una operación de edición. Por ejemplo, vocales comunes pueden ser eliminadas con bajo costo, o supresiones de consonantes son menos costosas si están en un grupo de consonantes que rodeadas de vocales, con un rol menos prominente en la formación de la sílaba. A pesar de que los insights lingüísticos pueden ser útiles para asignar peso a este tipo de fonema, su número, variedad y la rápida evolución de las formas de palabras hacen que un enfoque de aprendizaje mecánico sea más adecuado.

Desarrollamos un enfoque para paliar la intervención humana en el proceso de normalización del texto generado por el usuario. Sustituimos la normalización basada en el diccionario por conjuntos de palabras con el mismo tipo canónico. Luego, dada una nueva palabra, un clasificador automático encuentra la palabra más parecida en los conjuntos manualmente creados. Si la similitud entre estas dos secuencias de caracteres es sobre un umbral dado, la nueva palabra se clasifica junto a la más similar; si no, constituye un nuevo conjunto. En este encuadre, la tarea del anotador humano se reduce a crear una serie inicial de conjuntos y validar las asociaciones creadas por el clasificador.

Agrupar variantes de una misma palabra parece un buen enfoque para la normalización, en contraste con asignar una estructura canónica a todas las variantes de una palabra. Luego, la tarea de normalizar cambios cambia de encontrar similitudes entre una palabra desconocida y estructuras canónicas, a encontrar similitudes entre una palabra desconocida y todas las variantes conocidas de una palabra. Esto permite capturar variantes de variantes, que ocurren por lo general en el contexto dinámico

[1] Varios servicios *online* para normalización del lenguaje de los textos recurren a estrategias basadas en diccionarios, como http://transl8it.com/; http://www.lingo2word.com/translate.php; http://www.dtextrapp.com/

del contenido generado por los usuarios. En este trabajo nos centraremos en los avisos de los periódicos, pero esperamos que nuestro método pueda ser fácilmente transferido a otros dominios con similares fenómenos. La publicidad presenta la ventaja de proveer abundante, corpora situada en el tiempo, como opuesto a otros géneros como mensajes de texto cortos.

El resto del trabajo es organizado de la siguiente manera. Primero revisamos parte del trabajo previo sobre la normalización de textos muy abreviados y aprendizaje de distancias de edición. Luego describimos los datos y el método que utilizamos para normalizar textos de publicidad. En la Sección 4 recorremos los diferentes enfoques para modificar el costo de operaciones de edición en una distancia de edición de secuencias de caracteres para mejorar la exactitud de la medida de similitud entre palabras. Los experimentos y resultados son presentados en la Sección 5, seguido de un análisis de las implicaciones lingüísticas de los costos aprendidos para las distancias de edición. Concluimos delineando algunas ramificaciones para este trabajo.

2. Trabajos previos

La normalización de textos es una tarea crucial para muchas aplicaciones PLN, especialmente en géneros donde nuevas variantes de las palabras son producidas rápidamente, de manera que un lexicon se torna demasiado estático, fracasando en la cobertura de mayores proporciones de texto. Este es el caso de los sms y sus abreviaturas (AiTi et al., 2005; Choudhury et al., 2007; Acharyya et al., 2008; Kobus et al., 2008; Cook and Stevenson, 2009) o los textos médicos, con gran uso de acrónimos (Pakhomov, 2002; Torii et al., 2006; Okazaki et al., 2008; Stevenson et al., 2009). La normalización y los enfoques basados en la distancia de edición para la normalización pueden encontrarse también en áreas de aplicación como la corrección de ortografía o reconocimiento de discurso.

Sproat et al. (2001) presentan un extenso trabajo en normalización de palabras no estándar en diferentes géneros y aplicando técnicas tanto supervisadas como no supervisadas. Los enfoques que presentan dependen fuertemente de enfoques heurísticos a las más sistemáticas formas de variaciones de palabras, dejando el aprendizaje para las partes más libres. Este enfoque es exitoso, pero depende de una cantidad considerable de reglas hechas a mano que deben ser actualizadas regularmente si pretenden estar al día con los cambios en el contenido generado por el usuario. Nuestro objetivo es generar un método que reduzca el grado de intervención de seres humanos requerida.

Nuestra hipótesis consiste en que una medida conveniente de similitud entre palabras debería permitir agrupar palabras automáticamente. La distancia de edición de secuencias de caracteres parece ser especialmente adecuada para este propósito, porque estamos intentando evaluar si una secuencia de caracteres dada es una manera diferente de escribir otra secuencia de caracteres. No obstante, las distancias de propósito

general pierden precisión en secuencias de caracteres muy cortas, como en el caso de las abreviaturas. En este contexto, una distancia de edición más fina se torna necesaria, distancia donde el poder de discriminación de diferentes operaciones de edición no es homogéneo. Así, el problema de encontrar una distancia de edición más fina se convierte en el problema de encontrar los costos de las operaciones de edición que discriminan mejor.

Numerosos enfoques han sido propuestos para conocer los costos de las operaciones de edición para distancias de edición de secuencias de caracteres, desde transductores estocásticos (Ristad and Yanilos, 1998; Bilenko and Mooney, 2003; Oncina and Sebban, 2006) hasta campos condicionales aleatorios (McCallum et al., 2005), enfoques de máxima entropía (Pakhomov, 2002), modelos de canales ruidosos (AiTi et al., 2005; Cook and tevenson, 2009) o modelos de Markov ocultos (Durbin et al., 1998).

Proponemos un modelo simple que no posee el poder expresivo (y el enorme espacio de búsqueda correspondiente) de modelos más complejos como el de (McCallum et al., 2005), pero alcanza una buena performance en la tarea del momento, comparable con otros enfoques. Por ejemplo, (Choudhury et al., 2007) utilizan modelos de Markov ocultos entrenados en un corpus de SMS alineados manualmente con su transcripciones estándar al inglés, y alcanzan una precisión del 57,7%. Utilizando un enfoque no supervisado en una tarea comparable, (Cook and Stevenson, 2009) obtienen una exactitud de 59%.

3. Datos y método

Nuestro enfoque para la normalización de textos publicitarios es el que sigue. Primero, agrupamos manualmente las variaciones ortográficas de la misma palabra en una muestra del corpus. Luego, para una nueva palabra *w*, determinamos si es una variante ortográfica de una palabra conocida o si es una palabra nueva.

En lo que sigue detallamos ambos pasos.

3.1. Ejemplos clasificados manualmente

Estamos trabajando con un corpus de 1 millón de palabras de avisos en español clasificadas sobre inmuebles del periódico argentino local *La Voz del Interior*[2].

Hemos agrupado manualmente las variantes ortográficas de palabras encontradas en un solo día de avisos, en total 3359. Entre 55946 palabras tipo (Secuencias de caracteres separadas por espacios), 8727 elementos únicos fueron encontradas, sin contar los números. Clasificamos manualmente las palabras en 2824 grupos de los cuales 113 tenían 10 o más palabras y 1700 tenían un solo elemento. También creamos un corpus más pequeño, para llevar a cabo experimentos menores, con palabras que aparecían

[2] http://www.clasificadoslavoz.com.ar/

10 veces o más, en un total de 927 palabras, con 493 clases, de las cuales 333 eran clases de elemento único y 5 tenían 10 o más elementos.

Las variantes flexivas de la misma raíz son consideradas en la misma clase, porque tienden a abreviarse con el mismo significado (p. e., "*Ot. ot. ot ots Ot otr otra otros.*", con diferentes formas de "(y)otra" y abreviaturas que son válidas para todos ellos).

Las palabras no fueron separadas de la puntuación porque en este contexto la puntuación es ambigua, puede significar abreviatura pero también una relación entre dos partes de una misma palabra. No separamos las palabras en expresiones multi-palabra, muchas de las cuales se escriben juntas (p. e., "*3d*" para "*3 dormitorios*").

3.2. Clasificación de palabras no vistas previamente

Para la palabra *w* no vista previamente, encontramos la palabra *c* en la muestra manualmente recolectada que es más similar a *w*. Si la similitud entre *w* y *c* se encuentra sobre un determinado umbral, entonces *w* es asignada al mismo grupo que *c*, como una variante ortográfica de la misma palabra. Por el contrario, si *w* y *c* son demasiado disímiles, un nuevo grupo es creado con la nueva palabra *w*. Medimos la similitud entre las palabras utilizando la inversa de una distancia de edición de secuencia de caracteres: cuanto más corta la distancia, más grande la similitud.

La distancia básica de edición utilizada para calcular la similitud entre palabras es la distancia Levenshtein (Levenshtein, 1966). Esta distancia cuenta el número de cambios que son necesarios para transformar una secuencia de caracteres en otro. Cada cambio u operación de edición tiene un costo uniforme de 1, mientras que dejar el mismo carácter en ambas secuencias de caracteres tiene un costo de 0. Las operaciones de edición son aplicadas de izquierda a derecha en la secuencia de caracteres.

Hemos introducido algunas modificaciones en la distancia básica Levenshtein, tratando de mejorar su precisión para encontrar variantes ortográficas.

Primero, hemos sopesado los costos de las operaciones de edición, que son uniformes en las distancias Levenshtein. Como desarrollamos en la Sección 4, hemos explorado dos estrategias para determinar el costo de las operaciones de edición que reflejan su impacto para identificar variantes ortográficas de una palabra, como observamos en ejemplos manualmente clasificados.

Segundo, las operaciones de edición se han enriquecido con el contexto, teniendo en cuenta el carácter anterior y posterior al carácter en el que tiene lugar la operación. Entonces, diferentes pesos fueron asignados a la misma operación de edición si ocurría con diferentes caracteres antes o después de la misma.

Para evitar la ocurrencia escasa de datos que acompaña una mayor granularidad en las operaciones de edición, asignamos peso a las operaciones de edición con un contexto completo (con un carácter previo y un carácter posterior), con contenido parcial (con el carácter previo y el carácter posterior como instancias separadas) y sin contexto (sin caracteres que lo rodeen).

Luego, cuando encontramos la distancia de edición para un par de secuencias de caracteres, las operaciones de edición fueron aplicadas utilizando una estrategia de reculada /*back-off)*, en un orden específico de más a menos: si hemos aprendido un peso para la operación con el contexto izquierdo y derecho, aplicamos eso, si no, aplicamos el peso con el contexto parcial hacia la izquierda o hacia la derecha, si no tuvimos evidencia para ninguno de esos contextos, aplicamos el peso de la operación sin ningún contexto. En caso en que la operación de edición no haya sido vista en el corpus de entrenamiento, no tendría peso. En ese caso se aplica la distancia Levenshtein, con costo 0 en el caso de que haya coincidencia entre caracteres y 1 para no-coincidencia, inserción o supresión.

4. Aprendizaje de medidas de secuencias de caracteres basado en la distancia de edición

En el contexto de los avisos en periódicos, como es el caso en otros contextos con muchas abreviaturas, como sms o trabajos científicos, encontramos numerosas secuencias de caracteres cortas, posiblemente con grandes variantes entre ellas, por ejemplo: "*l/c liv/com. liv/com liv-com, liv/com, l/c. lc. lc lc, l/com. l/com L/c l/c*," para "living comedor". Con costos de edición uniforme, la mayoría de las secuencias de caracteres más cortas serán igualmente distantes a muchas palabras diferentes. Necesitamos nuestra distancia de edición para capturar grados en la importancia de operaciones de edición, evaluar cuáles significan una diferencia importante en secuencias de caracteres y cuáles no. Tomando la distancia Levenshtein como punto de partida, intentamos encontrar una medida de similitud que mejorara su performance al modificar el costo asociado a las diferentes operaciones de edición. Los costos fueron modificados al aprovechar la evidencia de nuestro corpus manualmente anotado, en dos diferentes maneras: por una combinación entre búsqueda aleatoria y búsqueda del mejor primero en el espacio de los costos asociados a cada opción de edición, y asociando los costos generados por errores a las operaciones de edición. Desarrollamos estos enfoques a continuación.

4.1. Búsqueda aleatoria del espacio de costos

Como un primer enfoque, hicimos una búsqueda aleatoria para encontrar una configuración de costos que podrían mejorar la línea de base de la distancia de edición. Debido a que el espacio de búsqueda es muy grande, exploramos cambiando los valores solo para aquellas operaciones que realmente ocurrieron cuando las palabras en el corpus fueron alineadas unas con otras. Utilizamos una combinación entre búsquedas aleatorias y búsquedas del primer mejor en el espacio de los costos de operaciones de edición.

El procedimiento fue el siguiente. Primero, tomamos 50 operaciones de edición al azar, y modificamos su costo inicial al agregar primero 1 y luego sustraer 1 al mismo. Para cada modificación en cada operación, evaluamos el impacto en la exactitud en una pequeña muestra aleatoria del corpus, y registramos la exactitud obtenida. Luego, ordenamos las operaciones en orden de exactitud decreciente, y evaluamos el impacto de la modificación en una muestra más grande del corpus.

Si la precisión en esta muestra más grande fue mayor que con la distancia Levenshtein, introdujimos las modificaciones en la presente serie de costos para operaciones de edición, y procedimos a evaluar la siguiente modificación. Dejamos de evaluar las modificaciones con muestras más grandes del corpus cuando la exactitud obtenida en la muestra más chica fue menor que la mejor exactitud obtenida hasta ese momento. Luego, comenzamos la búsqueda nuevamente explorando otras 50 operaciones aleatorias, hasta que una mejora significativa fue alcanzada.

4.2. Búsqueda dirigida por el error

Para mejorar la precisión de la distancia Levenshtein de una manera más eficiente que con la búsqueda aleatoria, aplicamos un esquema dirigido por el error para modificar los costos de la operación de edición.

Asignamos a las operaciones de edición un costo obtenido del número de veces que la operación de edición fue vista en la alineación de una palabra con la que se encontró como más parecida a ella, la que sería identificada como una variante ortográfica de la misma palabra. Las alineaciones fueron obtenidas aplicando la distancia Levenshtein a los pares de palabras. Luego, contamos las veces en que la operación de edición fue vista en una alineación de un par de palabras que eran en realidad variantes de la misma palabra (coincidencia), y palabras diferentes (no-coincidencia).

Luego, el valor asignado a cada operación de edición fue la proporción de veces que ocurrió en una no-coincidencia menos la proporción de veces que ocurrió en una coincidencia, de modo que las operaciones que ocurrían más seguido en los ajustes recibían costos menores, incluso negativos. Esto minimiza el costo de alineación de palabras con operaciones de edición que ocurrieron mayormente en las coincidencias, y maximiza el costo de las operaciones de edición que ocurrían más que nada en los no-coincidencias.

$$\text{Costo}_{oe} = \frac{\text{no-coincidencia}_{oe}}{\text{Noe}} - \frac{\text{coincidencia}_{oe}}{\text{Noe}}$$

Para evitar cálculos inexactos de eventos que ocurren demasiado pocas veces, solo tomamos en cuenta aquellas operaciones que ocurrían más de 10 veces en los ejemplos anotados manualmente. Además, tomamos 4 muestras del corpus y encontramos

el costo asignado a la operación en cada una de las 4 muestras, y solo incorporamos aquellos costos cuya desviación estándar en las cuatro muestras era menor a tres veces la media del costo. En esos casos, el costo de la operación de edición fue el costo promedio en las 4 muestras.

Se debe destacar que al encontrar una operación de edición para la cual no se había registrado ningún costo, se reculó (*backed-off*) a la distancia Levenshtein.

5. Experimentos y resultados

Evaluamos las modificaciones mencionadas anteriormente utilizando el corpus manualmente anotado que nombramos en la Sección 3.1. Evaluamos la exactitud como la proporción de palabras en el corpus de evaluación que coincidieron correctamente con una variante ortográfica de la misma palabra o destacadas como la única manera de escribir una palabra encontrada. El umbral para discriminar diferentes palabras fue llevado a 3, es decir, un nuevo grupo fue creado para aquellas palabras cuyo candidato más cercano estaba a una distancia de edición mayor a 3.

La distancia Levenshtein obtuvo una exactitud de .54 en el corpus de 900 palabras.

La búsqueda aleatoria en el espacio de costos de operación de edición no arrojó mejoras significativas respecto de la distancia Levenshtein. Efectivamente, para la mayoría de las muestras, la exactitud utilizando la distancia Levenshtein fue la misma que utilizando los costos modificados.

La modificación de costo dirigida por el error fue más exitosa que la búsqueda aleatoria. Cuando se entrenó en el corpus de 900 palabras, la exactitud creció a .58. No obstante, pocas (376) operaciones de edición fueron modificadas porque la mayoría de ellas no cumplían con las condiciones para ser tenidas en cuenta: o muy pocas ocurrencias o muy alta variabilidad entre las muestras. Cuando entrenado en el corpus de 8000 palabras, muchas más (6324) operaciones de edición cumplieron las condiciones, que arrojaron una exactitud del .68 en el corpus de 900 palabras.

6. Análisis cualitativo de los resultados

Inspeccionamos manualmente los pesos aprendidos para los parámetros de distancias de edición. Esto nos permitió encontrar regularidades para sostener hipótesis lingüísticas sobre cómo los usuarios introducen variaciones en las palabras, mientras que estas se mantienen entendibles para sus compañeros lectores.

Antes que nada, descubrimos que el enfoque completo estaba funcionando como se esperaba porque se asignaron, efectivamente, pesos más bajos a caracteres idénticos que a caracteres diferentes, y que a las variantes de minúsculas y mayúsculas del mismo caracter se les asignaron también valores más bajos. Con respecto a los caracteres

únicos, los paréntesis fueron los que más bajo costo de alineación tuvieron, junto con el signo +.

Por el contrario, los caracteres únicos a los que se asignó el costo más alto fueron los números (especialmente 2 y 3), y a la letra 's', especialmente en el contexto del final-de-la-palabra. Estos tres caracteres tienen la propiedad común de que expresan cardinalidad, que es un fragmento de significado muy importante en el contexto de textos referidos a inmobiliarias, y los usuarios no quieren arriesgar no ser comprendidos.

Los valores fueron más altos para las no-coincidencias con consonantes poco frecuentes, por ejemplo, entre *z* y otras consonantes, o entre letras de muy diferente sonido, como entre *t* y otras letras. Las inserciones y eliminaciones tuvieron valores más bajos que las no-coincidencias. Las inserciones más penalizadas fueron aquellas que recrearían una estructura silábica, es decir, la inserción de una vocal entre consonantes o de una consonante al final de la palabra.

7. Conclusiones y trabajo futuro

Hemos presentado trabajo en desarrollo orientado a la normalización de palabras en avisos clasificados. Nuestro enfoque requiere menos intervención humana que los enfoques basados en diccionarios. Las variantes ortográficas de la misma palabra son agrupadas, y la forma más larga es tomada como la canónica. Luego, los textos son normalizados substituyendo cada estructura por su estructura canónica.

Un clasificador automático asigna cada nueva palabra a un grupo pre-existente de variantes ortográficas o establece un nuevo grupo para la palabra nueva. La similitud entre palabras es calculada por una adaptación de la distancia Levenshtein, donde los costos de operaciones de edición son sopesados por su ocurrencia en errores y coincidencias del clasificador. Esta distancia de edición a medida, con costos de operaciones de edición aprendidos de ejemplos manualmente clasificados, mejora la distancia Levenshtein aumentando su exactitud de .54 a .68.

Como trabajo futuro planeamos comparar la performance de nuestra distancia aprendida con la distancia Jaro-Winkler (Winkler, 1999), que es especialmente adecuada para lidiar con abreviaturas, por su tratamiento particular de los prefijos.

Incrementaremos también el tamaño del contexto para tener en cuenta para que se aplique la distancia de edición, especialmente intentando capturar la proximidad al principio o final de la palabra.

Planeamos realizar una evaluación de larga escala de la dimensión temporal en la evolución de las abreviaturas. Compararemos la performance del clasificador si las nuevas palabras son provistas como aparecen en el periódico, incorporando palabras

en diariamente y luego utilizando todas las palabras incorporadas para calcular distancias con las palabras al día siguiente, o si intentamos clasificar todas las palabras en el corpus al mismo tiempo.

También queremos evaluar la performance de los costos de operaciones de edición aprendidas en el dominio de avisos para detectar variantes ortográficas de las palabras en otros dominios. Más concretamente, un enfoque similar para mensajes de texto cortos está en proceso, incluyendo la construcción de un corpus de mensajes en español, cubriendo el dialecto argentino y uruguayo.

Bibliografía

Acharyya, Sreangsu; Negi, Sumit; Subramaniam, L.V. and Shourya, Roy. 2008. Unsupervised learning of multilingual short message service (sms) dialect from noisy examples. In AND '08: Proceedings of the second workshop on Analytics for noisy unstructured text data, pages 67-74, New York, NY, USA. ACM.

Aw AiTi, Zhang Min, Yeo PohKhim, Fan ZhenZhen, and Su Jian. 2005. Input normalization for an English-Tochinese sms translation system. In The Tenth Machine Translation Summit.

Bilenko, Mikhail and Mooney, Raymond J. 2003. Adaptive duplicate detection using learnable string similarity measures. In Proceedings of the ninth ACM SIGKDD.

Choudhury, Monojit; Saraf, Rahul; Jain, Vijit; Mukherjee, Animesh; Sarkar, Sudeshna and Basu, Anupam. 2007. Investigation and modeling of the structure of texting language. Int. J. Doc. Anal. Recognit., 10(3):157-174.

Cook, Paul and Stevenson, Suzanne. 2009. An unsupervised model for text message normalization. In Workshop on Computational Approaches to Linguistic Creativity.

NAACL HLT 2009.

Durbin, R.; Eddy, S.; Drogh, A. and Mitchison, G. 1998. Biological sequence analysis: Probabilistic models of proteins and nucleic acids. Cambridge University Press.

Kobus, Catherine; Yvon, François and Damnati, Geraldine. 2008. Normalizing sms: are two metaphors better than one? In COLING '08: Proceedings of the 22nd International Conference on Computational Linguistics, pages 441-448, Morristown, NJ, USA. Association for Computational Linguistics.

Levenshtein, V.I. 1966. Binary codes capable of correcting deletions, insertions, and reversals. Cybernetics and Control Theory, 10(8):707-710. Original in Doklady Akademii Nauk SSSR 163(4): 845-848 (1965).

McCallum, Andrew; Bellare, Kedar and Pereira, Fernando. 2005. A conditional random field for discriminativelytrained finite-state string edit distance. In Proceedings of the Proceedings of the Twenty-First Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-05), pages 388–395, Arlington, Virginia. AUAI Press.

Michelson, Matthew and Knoblock, Craig A. 2006. Phoebus: a system for extracting and integrating data from unstructured and ungrammatical sources. In AAAI'06: proceedings of the 21st national conference on Artificial intelligence, pages 1947-1948. AAAI Press.

Okazaki, Naoaki; Ananiadou, Sophia and Tsujii, Jun'ichi. 2008. A discriminative alignment model for abbreviation recognition. In COLING '08: Proceedings of the 22nd International Conference on Computational Linguistics, pages 657-664, Morristown, NJ, USA. Association for Computational Linguistics. Jos´e Oncina and Marc Sebban. 2006. Learning stochastic edit distance: Application in handwritten character recognition. Pattern Recognition, 39(9):1575-1587.

Pakhomov, Serguei. 2002. Semi-supervised maximum entropy based approach to acronym and abbreviation normalization in medical texts. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), pages 160–167.

Ristad, E.S. and Yanilos, P.N. 1998. Learning string edit distance. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20:522-532.

Sproat, Richard; Black, Alan; Chen, Stanley; Kumar, Shankar; Ostendorf, Mari and Richards. Christopher 2001. Normalization of non-standard words. Computer Speech and Language, 15(3):287-333.

Stevenson, Mark; Guo, Yikun; Amri, Abdulaziz Al and Gaizauskas. Robert 2009. Disambiguation of biomedical abbreviations. In BioNLP '09: Proceedings of the Workshop on BioNLP, pages 71-79, Morristown, NJ, USA. Association for Computational Linguistics.

Torii, Manabu; Liu, Hongfang; Hu, Zhangzhi and Wu, Cathy. 2006. A comparison study of biomedical short form definition detection algorithms. In TMBIO '06: Proceedings of the 1st international workshop on Text mining in bioinformatics, pages 52-59, New York, NY, USA. ACM.

Winkler, W.E. 1999. The state of record linkage and current research problems. Technical Report Internal Revenue Service Publication R99/04, Statistics of Income Division.

Fecha de recepción: 15/12/09
Fecha de aceptación: 10/05/10