

Minería de Direcciones Postales

Javier Arruego, Ester Llorente, José Carlos Cortizo, Diego Expósito
José Luis Medina

Sistemas Informáticos

Universidad Europea de Madrid

C/Tajo s/n, Villaviciosa de Odón

28670, Madrid

<http://www.esp.uem.es/jccortizo/fumas>
{jabo.arruego, pittu82, medina.uem}@gmail.com

AINetLab

Artificial Intelligence & Network Solutions S.L.

C/Bélgica 7 2B, Fuenlabrada

28943, Madrid

<http://www.ainetsolutions.com/jccp>
{jccp, deg}@ainetsolutions.com

Resumen

En este artículo se presenta FuMaS (Fuzzy Matching System), un sistema que permite la recuperación eficiente de direcciones postales a partir de consultas con ruido. La recuperación difusa de esta información tiene innumerables aplicaciones, desde encontrar/limpiar duplicados en bases de datos (registros electorales, encontrar nidos de fraude postal, etc.) hasta corregir las entradas de los usuarios en sistemas tales como callejeros o cualquier tipo de formulario dónde haya que introducir una dirección postal.

En este artículo se presenta la arquitectura del sistema, así como los experimentos que, hasta el momento, se han realizado sobre el mismo. Los resultados de estos experimentos muestran que FuMaS es una herramienta muy útil para recuperar direcciones postales a partir de consultas con ruido, siendo capaz de resolver cerca del 85% de las direcciones con errores introducidas al sistema, una eficacia un 15% mayor que cualquier otro sistema similar probado.

1. Motivación

La integración de información es un área muy importante de investigación dentro de los campos de las bases de datos y de la minería de datos [3], [12]. Integrar múltiples fuentes de

información distintas, permite obtener una visión más completa y precisa del mundo, así como obtener conocimiento adicional del mismo. Uno de los problemas más usuales a la hora de integrar grandes bases de datos es el problema de detectar (para posteriormente limpiar o integrar) fragmentos de varios registros que tratan de las mismas entidades. Detectar varias registros tratando sobre las mismas entidades puede ser una tarea simple si la información que identifica a las identidades es la misma (y está completa) en todos los registros, pero esto no es algo común, ya que no siempre se tienen los mismos identificadores en todas las fuentes de datos a integrar (dni en una base de datos y nombre en otra). Otra fuente de dificultades se debe a la existencia de los mismos identificadores pero presentando algún tipo de ruido que hace que la coincidencia entre los mismos no sea perfecta.

La expresión “record linkage” [8] [9] (vinculamiento de registros) se refiere precisamente al uso de técnicas algorítmicas para encontrar registros que, aunque no identifiquen exactamente de la misma forma a una entidad, si que se refieren a la misma. Dentro de la literatura, el proceso de linkado de registros se encuentra asociado a una gran variedad de nombres [9]: heterogeneidad de identidad [6], identificación de identidades [14], identificación de instancias [25], mezclar/purgar [11], reconciliación de entidades [7], lavado de listas y lim-

pieza de datos [5].

Las aplicaciones del vinculamiento de registros son innumerables, sobre todo en entornos administrativos: desde la gestión de relaciones con los clientes, detección del fraude, data warehousing, encontrar registros en diversas fuentes pertenecientes a un mismo paciente [20], etc.

El enfoque general de los algoritmos de vinculación de registros es determinar un coeficiente de similitud entre cada par de registros, lo cual es una tarea muy pesada, del orden de $O(n^2)$. Para poder determinar el coeficiente de similitud entre dos registros, usualmente, se han utilizado distancias de edición [17] como puedan ser la distancia de Levenhstein [13] o el algoritmo de Smith-Waterman [21]. Estas distancias permiten calcular el número mínimo de operaciones sobre los caracteres (sustitución, inserción o eliminación) necesarias para convertir una cadena en otra. Existe una gran variedad de algoritmos de cálculo de distancias de dicción [15] [1] que pueden ser utilizados tanto dentro del proceso de vinculación de registros como muchas otras aplicaciones tales como el procesamiento de señales con ruido [13], [22], [4], la recuperación de información con errores textuales [23], [24], [2], [16] y la biología computacional [18], [19], [10].

En este artículo se trata la vinculación de registros aplicada al problema del reconocimiento difuso de direcciones postales. Las direcciones postales pueden considerarse como información estructurada, ya que contienen varios campos bien conocidos y delimitados (tipo de vía, nombre de la vía, código postal, municipio, etc.). Ahora bien, una dirección postal puede escribirse de múltiples maneras, de hecho la siguiente lista de direcciones correctas representa la misma dirección postal (calle Santa María Magdalena, número 5, 4° B, 28900, Madrid):

- C/ Santa Maria Magdalena, n. 5, 4B, 28900, Madrid
- C/ Sta Maria Magdalena, n. 5, 4B, 28900, Madrid
- C/ Santa M. Magdalena, n. 5, 4B, 28900, Madrid

Los 3 ejemplos anteriores muestran la misma dirección postal pero escrita de 3 posibles maneras distintas. Esto es posible debido a la utilización de abreviaturas, sinónimos, contracciones y otras formas lingüísticas que permiten generar cierta ambigüedad o variedad de formas escritas para los mismos conceptos. Además de estas formas correctas, también se pueden encontrar formas incorrectas que, pese a referirse a la misma dirección, muestran ciertos errores o datos faltantes. Como ejemplos:

- C/ Santa Maria Magdalena, Madrid
- C/ Santa Maria Madalena, n. 5, 4B, 28900, Madrid
- C/ Santa Marai Magdalena, n. 5, 4B, 28900, Madrid
- C/ Santa Maria Magdalena, n. 54, B, 28900, Madrid
- AV Santa Maria Magdalena, Madrid

No solo se pueden encontrar varias formas correctas de escribir una dirección postal, ni siquiera se tiene que tener únicamente en cuenta el hecho de poder encontrar errores en las direcciones, sino que se pueden encontrar combinaciones de ambas, es decir, errores en direcciones que están ya utilizando formas alternativas:

- C/ Sta Marai Madalena, Madrid
- C/ S M Madalena, 5, 4B, 28900, Madrid

Todos estos ejemplos muestran la dificultad de, a partir de una dirección introducida por un usuario, encontrar la dirección real a la que se está refiriendo. Este artículo presenta un sistema que, teniendo en cuenta la posibilidad de existencia de estos errores y variantes a la hora de escribir las direcciones postales, trata de encontrar la forma normal y correcta de escribir la dirección postal dada.

La siguiente sección describe la arquitectura del sistema. La sección 3 describe los experimentos realizados y discute los resultados obtenidos. En la sección 4 se concluye el artículo y se muestran las líneas a seguir.

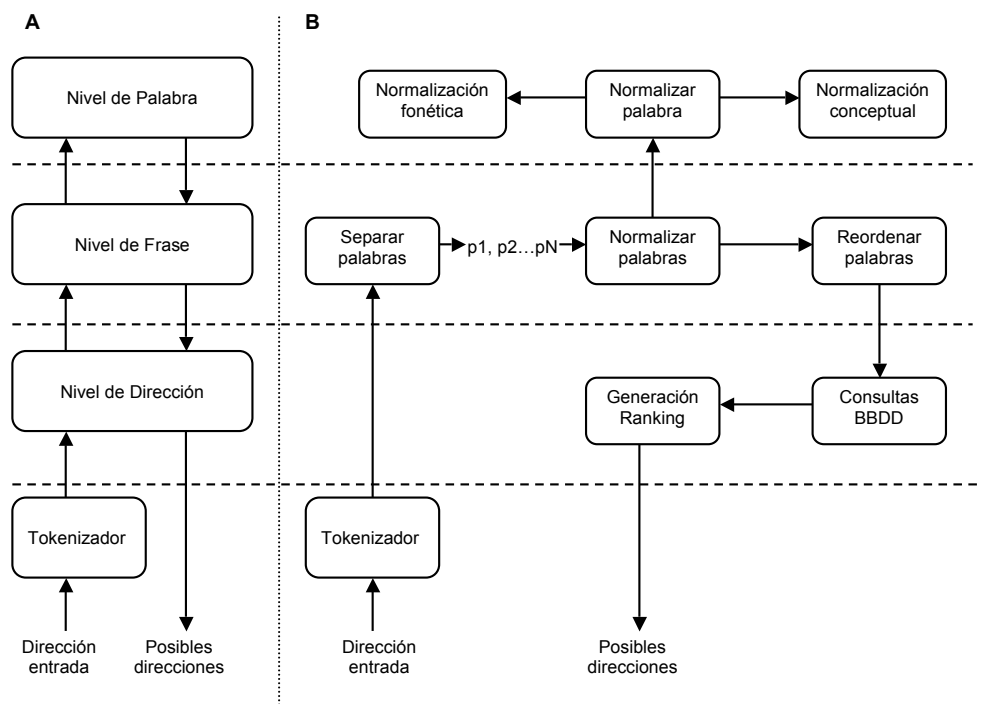


Figura 1: Arquitectura de FuMaS a nivel conceptual (subfigura A) y descomposición en módulos por nivel de abstracción (subfigura B). Como se puede ver, FuMaS, presenta una arquitectura en 3 capas de abstracción, cada una de las cuáles es experta en el manejo de un tipo de entidades (direcciones, frases o palabras). El nivel de palabra es experto en corregir los errores dentro de una palabra, tales como eliminaciones, inserciones o transposiciones de caracteres, así como de normalizar conceptualmente las palabras (abreviaturas, sinónimos, etc.). El nivel de frase se encarga de lidiar con los errores dentro de una frase, como puedan ser la eliminación, sustitución, inserción o transposición de palabras completas. El nivel de dirección se encarga de manejar las incoherencias entre todos los campos de una dirección postal, así como de estudiar cuáles de las soluciones propuestas son las más adecuadas en función de la entrada.

2. Descripción y Arquitectura de FuMaS

La arquitectura del sistema de FuMaS es una arquitectura dividida en 3 capas, cada una de ellas representativa de un nivel de abstracción. FuMaS recibe como entrada una dirección postal y devuelve, como salida, un ranking de posibles direcciones postales, cada una con una medida de la calidad de la solución, ordenadas de mayor a menor relevancia. Como se puede ver en la subfigura 1.A, una dirección que se introduzca como entrada del sistema pasará por los siguientes niveles:

- **Tokenización:** Se puede considerar como un paso de preprocesamiento, más que como un nivel de abstracción (por eso se contabilizan únicamente 3 capas). Este nivel se encarga de recoger un string conteniendo una dirección de entrada y separa la misma en los diferentes elementos constituyentes de la misma: tipo de vía, nombre de vía, número de portal, letra, escalera, código postal y municipio. Actualmente el tokenizador utilizado en FuMaS es un tokenizador 'ad-hoc' que espera los campos en un determinado orden, aunque, gracias a la modularidad del sistema, se puede sustituir por un tokenizador más inteligente capaz de lidiar con diversos tipos de estructuras de dirección postal.
- **Nivel de Dirección:** Constituye la capa de mayor abstracción, ya que es la capa encargada de estudiar las coherencias entre todos los campos de una dirección postal. Esto incluye las asociaciones entre códigos postales, municipios y nombres de calles, el número de portal asociado a un determinado domicilio, etc. Además, es el responsable de generar la salida del sistema: una lista de posibles direcciones postales, ordenadas de mayor a menor relevancia. Para poder generar el ranking de salidas del sistema, se utiliza un algoritmo de cálculo de distancia de edición entre dos strings (distancia de Levenshtein, q-gram, etc.), aunque la elección del algo-

ritmo concreto se ha dejado como un parámetro del sistema definible por el usuario.

- **Nivel de Frase:** Esta capa es la encargada de trabajar con las cadenas de palabras (frases). Principalmente se encarga de corregir los fallos debidos a la eliminación, inserción, sustitución o transposición de palabras dentro de una frase.
- **Nivel de Palabra:** Es la capa a más bajo nivel. Se encarga de corregir los fallos debidos a la eliminación, inserción, sustitución o transposición de letras dentro de una palabra, así como de normalizar las palabras, tanto a nivel fonético como conceptual. La normalización fonética se encarga de representar las palabras de una forma más cercana a como se pronuncian para corregir errores como la falta de una 'h' o la sustitución de una 'b' por una 'v'. La normalización conceptual se encarga de unificar distintas formas de escritura referentes a un mismo término (abreviaturas, sinónimos, etc.).

Esta arquitectura basada en niveles de abstracción permite un acercamiento más simplista a un problema bastante complejo de por sí. Cada una de las capas puede modificarse, añadiendo mayor o menor complejidad en función de si lo que se busca son resultados más precisos o bien menor tiempo de ejecución. Esto ha permitido tener, en un periodo de tiempo relativamente corto, un prototipo del sistema plenamente funcional (aunque muy mejorable), para poder evaluar el rendimiento del mismo y, por ende, evaluar las posibilidades de éxito de la arquitectura propuesta.

3. Experimentos y Resultados

Para poder evaluar el sistema, se ha construido una pequeña colección de 100 direcciones postales correctas y se han generado variantes incorrectas de la misma, consiguiendo una colección de 300 direcciones postales en total. Las direcciones postales incorrectas se han generado teniendo en cuenta distintos posibles

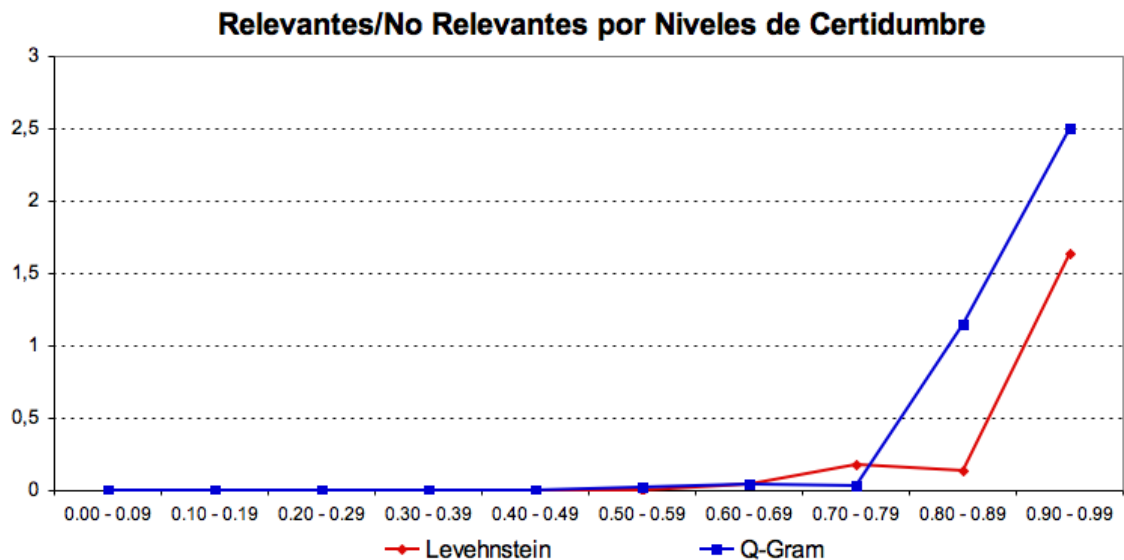


Figura 2: Ratios de direcciones recuperadas relevantes entre direcciones recuperadas no relevantes por niveles de certidumbre para las distancias de Q-gram y Levehnstein.

fallos a la hora de escribir una dirección postal:

- Fallos fonéticos: Sustituciones de 'b' por 'v', ausencia de 'h', etc.
- Fallos conceptuales: Utilización de palabras sinónimas, abreviaturas, etc.
- Incompletitud: Ausencia de algunas partes de la dirección postal (parte del nombre de la vía, tipo de vía, código postal, etc.).
- Fallos por sustitución: Aparición de un código postal, tipo de vía o algún otro elemento incorrecto.

Esta pequeña colección de direcciones postales se ha utilizado para evaluar distintos aspectos del sistema, así como el rendimiento global del mismo.

Los primeros experimentos realizados sobre el sistema han sido para evaluar la idoneidad de utilizar alguna distancia de edición. Concretamente se han estudiado las distancias de

Levehnstein y Q-Gram, para lo cuál se han calculado un ratio de direcciones relevantes entre direcciones no relevantes en función del grado de certidumbre de cada una de las métricas. La Figura 2 sintetiza los resultados, mostrando que la distancia de Q-Gram muestra, en general, un mejor ratio de relevantes recuperados por relevantes no recuperados que la distancia de Levehnstein, y que a partir de niveles de certidumbres superiores a 0.8, el número de direcciones no relevantes recuperadas es prácticamente despreciable.

Los otros experimentos realizados han sido para comparar a FuMaS con otros sistemas ya existentes. A este respecto, se ha seleccionado tanto el software de Uniserv (<http://www.uniserv.com>), que es una aplicación específicamente diseñada para la validación y corrección de direcciones postales, así como 3 sistemas de callejeros que realizan algún tipo de proceso para corregir las direcciones postales antes de mostrar el mapa adecuado:

- Guía Campsa

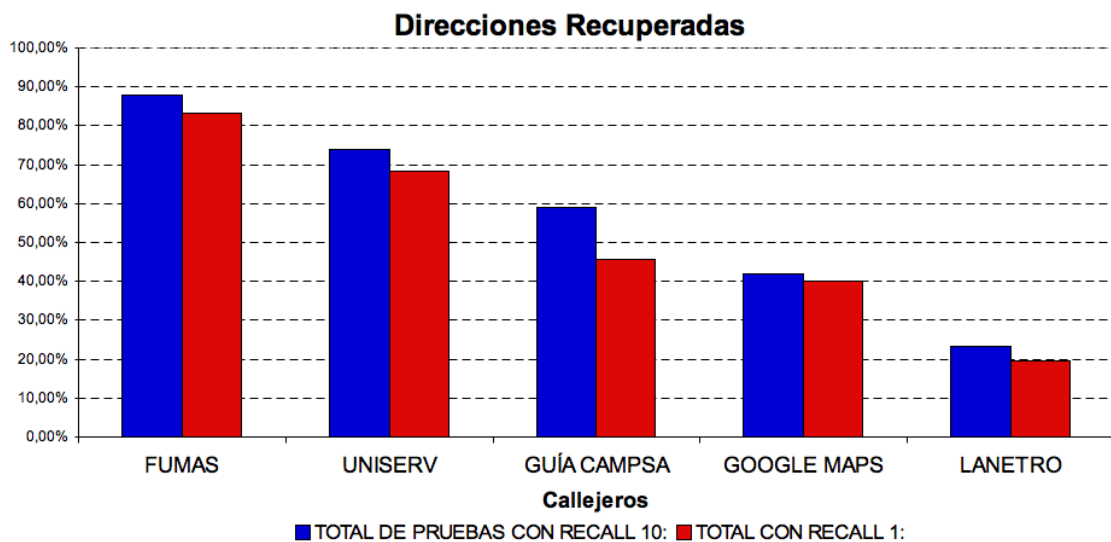


Figura 3: Porcentaje de direcciones relevantes recuperadas. Con recall 1 es el número de direcciones relevantes recuperadas cuándo solo se tiene en cuenta la primera dirección de la salida de cada sistema. Con recall a 10 es el número de direcciones relevantes recuperadas cuándo se tienen en cuenta las 10 primeras salidas del sistema.

(<http://www.guiacampsa.com>)

- Google Maps (<http://maps.google.com>)
- La Netro (<http://callejero.lanetro.com>)

La Figura 3 muestra los resultados de esta comparativa. Por cada sistema se ha calculado el porcentaje de direcciones relevantes recuperadas cuándo solo se tiene en cuenta la primera salida del sistema, es decir, la primera dirección que ofrece el sistema es la que se estaba buscando (Recall 1) así como el porcentaje de direcciones relevantes recuperadas cuándo se tienen en cuenta las 10 primeras direcciones que ofrece la salida de cada sistema, es decir, la dirección buscada está entre las 10 primeras.

Como se puede ver en la Figura 3, FuMaS consigue recuperar un mayor número de direcciones (R10: 87,85 %, R1: 83,18 %) que cualquiera de los otros sistemas (más de un 14 % de diferencia con su inmediato seguidor, el software de Uniserv). Además, el número de direcciones correctamente corregidas es bastante

elevado, por lo que es de suponer que con algunas mejoras en el sistema, se consigan ratios de corrección cercanos al 100 % de efectividad.

4. Conclusiones y Trabajo Futuro

En este artículo se ha presentado FuMaS, un sistema que permite la recuperación eficaz de direcciones postales con ruido. Los resultados experimentales muestran que FuMaS, a pesar de su pronto estado de gestación, es capaz de corregir un 85 % de las direcciones erróneas introducidas al sistema, superando por más de 15 puntos a cualquier otro software similar evaluado.

FuMaS está lejos de ser un sistema acabado y cerrado; por ahora muestra una arquitectura capaz de lidiar con el sistema y lo suficientemente modular como para permitir muchos grados de mejora. Por otra parte, los resultados presentados en este artículo presentan la piedra de apoyo de una nueva línea de investigación que pretende abordar el problema

de la recuperación aproximada de información estructurada tanto en el dominio de las direcciones postales como en otros dominios dónde la arquitectura de FuMaS pueda ser adaptada gracias a su gran flexibilidad. En este sentido, cabe señalar que de los 3 niveles de abstracción de FuMaS, dos de ellos son muy generales (el de frase y el de palabra) y pueden ser reutilizados en su práctica totalidad en otros ámbitos.

FuMaS abre muchas líneas futuras de actuación:

- Estudios de los errores más comunes a la hora de escribir direcciones postales para tomar las medidas necesarias para corregirlo
- Desarrollo de algoritmos de traducción fonética más avanzados y adecuados.
- Creación, reutilización e integración de recursos léxicos que permitan una normalización conceptual de fragmentos de direcciones.
- Desarrollo de tokenizadores más inteligentes capaces de extraer las partes de una dirección postal a pesar de que no esté correctamente estructurada.
- Desarrollo de métodos de indización para búsquedas aproximadas de strings.
- Desarrollo de algoritmos de sustitución de palabras/letras a partir del estudio de las frecuencias de aparición de bigramas, trigramas, etc.
- Adecuación de FuMaS a otros idiomas, ya que actualmente todos sus módulos están diseñados para el castellano.

FuMaS se muestra como un sistema eficaz a la hora de recuperar direcciones postales con ruido y está previsto que sea la plataforma que sustente una gran cantidad de investigaciones y líneas de actuación amparadas en los puntos anteriormente mencionados.

Referencias

- [1] A. Aho. *Handbook of Theoretical Computer Science: Algorithms and Complexity*, chapter Algorithms for finding patterns in string, pages 255–300. MIT Press, 1990.
- [2] R. Baeza-Yates and G. Navarro. A practical index for text retrieval allowing errors. *CLEI*, 1:273–282, 1997.
- [3] C. Batini, M. Lenzerini, and S. B. Navathe. A comparative analysis of methodologies for database schema integration. *ACM Comput. Surv.*, 18(4):323–364, 1986.
- [4] G. Chollet. Automatic speech and speaker recognition: overview, current issues and perspectives. pages 129–147, 1994.
- [5] P. Christen and T. Churches. febrl - freely extensible biomedical record linkage. Sourceforge.net, 2005.
- [6] D. Dey, S. Sarkar, and P. De. A probabilistic decision model for entity matching in heterogeneous databases. *Manage. Sci.*, 44(10):1379–1395, 1998.
- [7] D. Dey, S. Sarkar, and P. De. A distance-based approach to entity reconciliation in heterogeneous databases. *IEEE Transactions on Knowledge and Data Engineering*, 14(3):567–582, 2002.
- [8] I. P. Fellegi and A. B. Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210, 1969.
- [9] L. Gu, R. Baxter, D. Vickers, and C. Rainsford. Record linkage: Current practice and future directions.
- [10] D. Gusfield. *Algorithms on strings, trees, and sequences: computer science and computational biology*. Cambridge University Press, New York, NY, USA, 1997.
- [11] M. A. Hernandez and S. J. Stolfo. The merge/purge problem for large databases.

- In *SIGMOD Conference*, pages 127–138, 1995.
- [12] M. A. Hernandez and S. J. Stolfo. Real-world data is dirty: Data cleansing and the merge/purge problem. *Data Mining and Knowledge Discovery*, 2(1):9–37, 1998.
- [13] V. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10:707–+, Feb. 1966.
- [14] E.-P. Lim, J. Srivastava, S. Prabhakar, and J. Richardson. Entity identification in database integration. *Information Sciences*, 89(1):1–38, 1996.
- [15] G. Navarro. A guided tour to approximate string matching. *ACM Computing Surveys*, 33(1):31–88, 2001.
- [16] G. Navarro, R. Baeza-Yates, and J. Marcelo. Matchsimile: A flexible approximate matching tool for personal names searching. In *Proceedings of SBBD'01*, pages 228–242, 2001.
- [17] G. Navarro and M. Raffinot. *Flexible Pattern Matching in Strings – Practical online search algorithms for texts and biological sequences*. Cambridge University Press, 2002. ISBN 0-521-81307-7. 280 pages.
- [18] S. Needleman and C. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48:444–453, 1970.
- [19] J. B. Sankoff, David; Kruskal. *Time warps, string edits, and macromolecules: The theory and practice of sequence comparison*. Addison-Wesley, 1983.
- [20] E. A. Sauleau, J.-P. Paumier, and A. Bue-mi. Medical record linkage in health information systems by approximate string matching and clustering. *BMC Medical Information Decision Making*, 32(5):5–32, 2005.
- [21] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.
- [22] T. K. Vintsyuk. Speech discrimination by dynamic programming. *Cybernetics and System Analysis*, 4(1):52–57, 1968.
- [23] R. A. Wagner and M. J. Fischer. The string-to-string correction problem. *J. ACM*, 21(1):168–173, 1974.
- [24] R. A. Wagner and R. Lowrance. An extension of the string-to-string correction problem. *J. ACM*, 22(2):177–183, 1975.
- [25] Y. R. Wang and S. E. Madnick. The inter-database instance identification problem in integrating autonomous systems. In *Proceedings of the Fifth International Conference on Data Engineering*, pages 46–55, Washington, DC, USA, 1989. IEEE Computer Society.