12. Tubaishat, M. Adaptive traffic light control with wireless sensor networks [Text] / M. Tubaishat, Y. Shang, H. Shi // Proceedings of IEEE Consumer Communications and Networking Conference, 2007. – P. 187–191. doi: 10.1109/ccnc.2007.44

13. Branston, D. Some factors affecting the capacity of signalized intersection [Text] / D. Branston. – TrafficEng. and Contr., 1979. – P. 390–396.

References

1. Gavrylov, E. V., Dmytrychenko, M. F., Dolja, V. K. et. al.; Dmytrychenko, M. F. (Ed.) (2007). Systemologija na transporti. Organizacija dorozhn'ogo ruhu. Kiev: Znannja Ukrai'ny, 452.

2. Dewar, R., Olsen, P. (2007). Human Factors in Traffic Safety. 2nd edition. Lawyers and Judges Publishing Company, Inc., 549.

3. Shinar, D. (2007). Traffic Safety and Human Behavior. Elsevier, 776.

4. Polishhuk, V. P., Dzjuba, O. P. (2008). Teorija transportnogo potoku: metody ta modeli organizacii' dorozhn'ogo ruhu. Kiev: Znannja Ukrai'ny, 175.

5. Leutzbach, W. (1988). Introduction to the theory of traffic flow. Berlin : Springer-Verlag, 204. doi: 10.1007/978-3-642-61353-1

6. Markowski, M. J. (2008). Modeling behavior in vehicular and pedestrian traffic flow. New York: Umi, 162.

7. Pugachev, I. N. (2004). Organizacija i bezopasnost' dvizhenija. Habarovsk: Izd–vo Habar. gos. tehn. un–ta, 232.

8. Jarkin, E. K., Harchenko, E. V. (2000). Planirovochnaja organizacija dvizhenija transporta v gorodah. Juzh.-Ros. gos. tehn. un-t. Novocherkassk: JuRGTU, 365.

9. Pal'chyk, A. M. (2010). Transportni potoky. Kiev: NTU, 171.

10. Glik, F. G. (1998). Obsledovanie transportnyh potokov i prognozirovanie nagruzki seti gorodskih ulic i dorog. Social'no-jekonomicheskie problemy razvitija transportnyh sistem gorodov. Ekaterinburg, 105.

11. Sacks, G., Rayner, M., Swinburn, B. (2009). Impact of front-of-pack 'traffic-light'nutrition labelling on consumer food purchases in the UK. Health promotion interna-tional, 24 (4), 344–352. doi: 10.1093/heapro/dap032

12. Tubaishat, M., Shang, Y., Shi, H. (2007). Adaptive traffic light control with wireless sensor networks. Proceedings of IEEE Consumer Communications and Networking Conference, 187–191. doi: 10.1109/ccnc.2007.44

13. Branston, D. (1979). Some factors affecting the capacity of signalized intersection. TrafficEng. and Contr., 390–396.

Рекомендовано до публікації д-р техн. наук Доля В.К.
Дата надходження рукопису 20.05.2015

Санько Ярослав Володимирович, кандидат технічних наук, доцент, кафедра транспортних систем і логістики, Харківський національний університет міського господарства ім. О. М. Бекетова, вул. Революції, 12, м. Харків, Україна, 61002
E-mail: yron08@rambler.ru

Музалевська Юлія Юріївна, аспірант, кафедра транспортних систем і логістики, Харківський національний університет міського господарства ім. О. М. Бекетова, вул. Революції, 12, м. Харків, Україна, 61002
E-mail: ulialu_02@mail.ru

Лепетюк Ярослав Олегович, Харківський національний університет міського господарства ім. О. М. Бекетова, вул. Революції, 12, м. Харків, Україна, 61002
E-mail: yaros-yaros@mail.ru

УДК 004.056
DOI: 10.15587/2313-8416.2015.44364

МЕТОД КЛАСТЕРИЗАЦІЇ ПОВІДОМЛЕНЬ ЗА ДОПОМОГОЮ АРХІВУЮЧОГО ПЕРЕТВОРЕННЯ

© О. О. Сірий

В даній статті представлено метод визначення характеристик текстів та їх класифікації за допомогою архівування. Використовуючи прямий зв'язок архівування за допомогою алгоритмів LZ77 і Хаффмана з ентропією, виділяються ознаки тексту, що дозволяють визначати мову його написання, стиль, авторство, кластеризувати масиви даних за їх належністю до певної тематики
Ключові слова:*архівація, ентропія, розпізнавання тексту, спам, фішинг, LZ77, алгоритм Хаффмана*

This article represents the method of the text's parameters identification and their classification with the help of archiving. Using the direct bond between the archiving with LZ77 and Huffman algorithm and entropy, the text's characteristics are identified, and they help to define its language, style, authorship, and cluster data files by their topic relevance
Keywords: *archiving, entropy, text recognition, spam, fishing, LZ77, Huffman algorithm*

1. Вступ

Важливе місце в спілкуванні за допомогою мережі Інтернет посідає обмін текстовими повідомлення. Спілкування з колегами, друзями, родиною, отримання інформації про актуальні події навколишнього світу, ознайомлення з новинами – все це представлено текстовою інформацією у повідомленнях електронної пошти, соціальних мереж, постах блогів і сайтів новин. Таке широке використання даного виду передачі інформації безсумнівно притягує в цю сферу тих, хто не завжди наслідує правила суспільства. Спам, фішинг, відмова від авторства, дезінформація – все це і багато іншого використовується шахраями для задоволення власних цілей. Виходячи з цього, постає проблема аналізу повідомлень на рівні їхнього змісту, стилю написання. Повний вербальний аналіз текстів з урахуванням рівня технологій, незважаючи на все, залишається важким завданням. Тому для цього можуть бути використані статистичні методи аналізу. Існують методи, що базуються на математичних властивостях інформації (зокрема, ентропії), що дозволяють виділяти спільні ознаки в близьких за походженням або авторством текстових повідомленнях.

2. Постановка проблеми

Проблема оптимального кодування тексту, зображення або будь-якого іншого виду інформації активно вивчалася в минулому столітті. Зокрема, американський науковець Клод Шенон виявив, що існує обмеження на можливість кодування послідовності. Це обмеження – це ентропія послідовність. Є багато еквівалентних визначень ентропії, але, ймовірно, найкраще визначення в даному контексті – це ентропія Хайтіна-Колмогорова: ентропія послідовності символів – це довжини (в бітах) найменшої програми, яка на виході дає цю послідовність. Це визначення є абстрактним. Зокрема, неможливо навіть теоретично знайти згадану програму. Проте є такі алгоритми, які найбільше наближаються до цієї теоретичної межі. Це компресори файлів або архіватори. Архіватор бере файл і намагається перетворити його на настільки короткий відповідник, наскільки це можливо без втрати змісту.

Вхідну послідовність символів можна розглядати як послідовність рядків, що містять довільну кількість символів. Ідея словникових методів полягає в заміні послідовностей символів на такі коди, що їх можна трактувати як індекси записів деякого словника. Записи, з яких складається словник, називаються фразами. При декодуванні здійснюється зворотна заміна індексу на відповідну їй фразу словника.

3. Літературний огляд

Зважаючи, що основною системою впровадження всіх вищеназваних атак є масові розсилки, основні зусилля спрямовуються на боротьбу зі спамом. На сьогоднішній день найбільш розвинутими системами захисту від спаму оснащені поштові агенти, адже такий вид спаму має найдовшу історію, а отже, і методи протидії йому почали знаходити найпершими.

Існує програмне забезпечення для автоматичного визначення спаму – фільтри [1]. Вони можуть застосовуватися кінцевими користувачами або на серверах. Є два основних методи роботи фільтрів.

Перший полягає в аналізі змісту листа, на основі чого робиться висновок, спам це чи ні. Якщо лист класифікований як спам, він може бути позначений, переміщений в іншу папку або навіть вилучений. Таке програмне забезпечення може працювати як на сервері, так і на комп'ютері клієнта. При такому підході ви не бачите відфільтрованого спаму, але сервер продовжує затрачати на його обробку ресурси.

Другий підхід базується на класифікації відправника як спамера, не торкаючись тексту листа. Для визначення застосовуються різні методи. Це програмне забезпечення може працювати тільки на сервері, який безпосередньо приймає пошту. При такому підході можна зменшити витрати ресурсу на обробку кожного листа, проте залишається необхідним аналізувати сервер-відправник і обмінюватись даними з базами, що уміщують перелік небезпечних серверів.

Однією з найпопулярніших систем фільтрації спаму є SpamAssassin [2]. Ця система заснована на взаємодії ключових компонентів — оцінюючого сервісу, транспортного агенту та бази шаблонів листів.

SpamAssassin поставляється з великою колекцією правил, які визначають, які листи віднести до спаму, а які ні. Більшість правил засновано на регулярних виразах, що співставляються тілу або заголовку листа, але SpamAssasisn також використовує й інші методики.

Кожне правило має певну вартість. Якщо повідомлення збігається з правилом, ця вартість додається до загального балу. Вартість може бути і додатною, і від'ємною: додатні значення називаються «spam», відємні – «ham». Повідомлення співставляється зі всіма правилами, підраховується загальний бал. Чим вищий бал, тим більша імовірність що це повідомлення є спамом.

SpamAssassin володіє межею, при перевищенні якої лист буде класифіковано як спам. Зазвичай межа встановлюється на такому рівні, щоб тільки листи, які співпадають за декількома правилами, потрапляли під класифікацію спаму, адже одного правила недостатньо, щоб вважати повідомлення спамом.

SpamAssassin, як і більшість програмного забезпечення такого виду, використовує для аналізу тексту баєсівську фільтрацію спаму [3] – метод для фільтрації спаму, заснований на використанні наївного баєсівського класифікатора, що використовує теорему Баєса.

В контексті спаму ця теорема використовується декілька разів:

- підрахунок імовірності того, що повідомлення є спамом, якщо в ньому з'являється це слово;
- підрахунок імовірності того, що повідомлення є спамом, враховуючи всі його слова або певні їх підмножини;
- інколи, коли зустрічаються повідомлення з рідкісними словами.

4. Архівуюче перетворення

Архівуюче перетворення – це процедура перекодування даних, що дозволяє встановити зв'язок між відносною ентропією та довжиною перекодованої послідовності, за рахунок використання ентропійного кодування.

Ентропійне кодування [4] – кодування послідовності значень з можливістю однозначного відновлення з метою зменшення обсягу даних (довжини послідовності) за допомогою усереднення ймовірностей появи елементів у закодованій послідовності. Коди використовують зіставлення кожному елементу вихідної послідовності різного числа елементів результуючої послідовності. Чим більше вірогідність появи вихідного елемента, тим коротше відповідна результуюча послідовність.

Таке кодування забезпечують потужний інструмент для вимірювання ентропії. Перший висновок, який можна зробити, – це можливість вимірювання ентропії методом простої архівації тексту.

Найпростіший спосіб зрозуміти, звідки походить наше визначення, – згадати поняття відносної ентропії, сутність якого легко зрозуміти на наступному прикладі.

Розглянемо два ергодичних джерела A і B, що генерують послідовності 0 і 1: A генерує 0 з імовірністю p і 1 з імовірністю $1-p$, в той час як B генерує 0 з імовірністю q і 1 з імовірністю $1-q$. Алгоритм стиснення при застосуванні до послідовності, згенерованої джерелом A, зможете закодувати послідовність майже оптимально, тобто 0 буде закодований $\log_2 p$ бітами і 1 $\log_2(1-p)$ бітами. Це оптимальне кодування не буде оптимальним для послідовності B. Зокрема, ентропія на символ послідовності B в кодуванні, оптимальному для A, буде рівна $-q\cdot\log_2 p-(1-q)\cdot\log_2(1-p)$, в той час як ентропія на символ послідовності B в її оптимальному кодуванні буде дорівнювати

$$-q\cdot\log_2 q-(1-q)\cdot\log_2(1-q).$$

Кількість бітів на символ витрачених, щоб закодувати послідовність B оптимальним кодуванням для A, буде відносною ентропією A і B.

$$S_{AB}=-q\cdot\log_2\left(\frac{p}{q}\right)-(1-q)\cdot\log_2\left(\frac{1-p}{1-q}\right).$$

Існує кілька способів вимірювання відносної ентропії. Однією з можливостей є, звичайно, використати приклад, описаний вище: використання оптимального кодування для даного джерела для кодування повідомлення іншого джерела. Ми використовуємо схожий алгоритм. Для того, щоб визначити відносну ентропію між двома джерелами A і B, ми генеруємо довгу послідовність A джерелом A, довгу послідовність B, а також коротку послідовність b джерелом B. Створюємо нову послідовність $A+b$ простим додаванням b після A. Після того архівуємо її, наприклад, за допомогою GZIP. Мірою довжини b в оптимальному кодуванні для A буде $\Delta_{Ab}=L_{A+b}-L_A$, де L_X – довжина в бітах архівованого файлу X. Відносна ентропія на символ S_A між A і B дорівнює

$$S_{AB}=\frac{(\Delta_{Ab}-\Delta_{Bb})}{|b|},$$

де $|b|$ – кількість символів в послідовності b і $\Delta_{Bb}=(L_{B+b}-L_B)/|b|$ – оцінка ентропії джерела B.

Методика кластеризації

Даний метод може бути використаний для визначення мови написання тексту, його стилю, належності певному автору, кластеризації масивів даних за їх належністю до певної тематики. Основний підхід включає в себе такі етапи:

1) Збір банку даних для поставленої задачі;

2) Порівняння цільового тексту з кожним елементом банку;

3) Аналіз результатів.

Розглянемо це на прикладі. Нехай ми маємо текст X, що написаний невідомою для нас мовою. В даному випадку банком даних буде виступати бібліотека текстів, написаних різними мовами, та їхні відповідників в архівованому вигляді. Порівняння текстів проходить за наступним алгоритмом:

1) Частина тексту X додається до оригінального тексту A_i з бібліотеки (кожного разу використовується той самий уривок тексту X);

2) Проводиться архівування, методом, аналогічним до того, що був застосований для архівування елементів бібліотеки;

3) Обраховується різниця між довжиною архівованого тексту A_i та тексту, отриманого в попередньому кроці.

Процедура повторюється для кожного тексту в бібліотеці. Найменше значення різниці буде свідчити про найбільше наближення мови даного тексту до мови тексту з бібліотеки.

Розглянемо чому попереднє твердження є вірним. Будемо використовувати алгоритм Хаффмана і вважатимемо, що тексти X та A_i в даному прикладі написані різними мовами. Коли архіватор розпочинає кодувати файл, він починає процес формування дерева кодів. З часом зміни в дереві відбуваються все рідше, за рахунок того, що всі символи мови вже записані в нього і їх розподіл є усталеним. Досягнувши кінця тексту A_i, ми отримуємо дерево кодів Хаффмана, налаштоване на символи і закономірності їх слідування для певної мови. Коли архіватор починає опрацьовувати текст іншої мови з'являються нові символи або змінюється їх статистичний розподіл. У випадку зміни набору символів у дереві починають з'являтись нові листи. У порівняння з листами початкового набору символів їх вага буде значно меншою, а отже, і довжина коду для такого символу буде більшою. Нові, більші коди вплинуть на розмір архівованого тексту більшою мірою, ніж якби були використані ті самі коди, що вже є в дереві. У випадку мови зі схожим набором символів, вплив нових символів не буде таким значним, хоча він все рівно буде зали-

шатись. У цьому випадку на темпи збільшення розміру файлу буде впливати розподіл символів. Коди Хаффмана вже не будуть оптимальними для цієї частини тексту, тому розмір шифрованого тексту буде не мінімально можливим.

Використання описаного методу для підтвердження належності повідомлення X певному автору має певні відмінності в підході. Для цієї ситуації банком даних буде виступати архів повідомлень користувача, що нас цікавить. Важливою умовою є те, що всі повідомлення повинні бути написані однаковою мовою, адже враховуючи попередній приклад – нові символи значною мірою впливають на процес. Розмір повідомлень не повинен бути однаковим для всіх елементів банку. Алгоритм буде виглядати таким чином:

1) З N повідомлень архіву генеруються (проста конкатенація) тексти A_i певної довжини, такої, що значно перевищує середню довжину повідомлення архіву. Кількість таких текстів залежить від розміру архіву, середньої довжини повідомлення та необхідної точності аналізу (еквівалентне затраті ресурсів);

2) Створюються архівовані копії текстів, згенерованих в першому кроці;

3) Повідомлення X додається до кожного оригінального тексту та архівуються способом, використаним в другому кроці;

4) Обраховується різниця між довжиною архівованого тексту A_i та тексту, отриманого в попередньому кроці;

5) Визначається коефіцієнт стиснення для повідомлення X, тобто відношення різниці довжин, отриманої в кроці 4, до довжини повідомлення X.

Попередньо аналогічна процедура повинна бути проведена для кожного повідомлення архіву, для визначення коефіцієнта стиснення для даного автора для усіх повідомлень. Усереднивши коефіцієнт стиснення, можна сформулювати нижню межу його значення, перехід якою дозволяє вважати повідомлення таким, що належить цьому автору. Це підтверджується тим, що найбільшого коефіцієнту стиснення можна досягти тільки при використанні оптимального кодування. Якщо ж текст написаний іншим автором, завдяки стильовим особливостям, розподіл символів і їх послідовностей буде змінюватись, а отже дерево кодів Хаффмана, сформоване під час архівування текстів A_i, вже не буде оптимальним.

5. Висновки

Метод, представлений у даній роботі, має широку сферу застосування – від простого розпізнавання мови до кластеризації масивів даних за ознаками авторства, стилю, тематики. Завдяки використанню перевірених і максимально ефективних алгоритмів архівування процес аналізу є легким і швидким. Це дозволяє його використання у системах моментального аналізу даних, таких як аналізатори спаму. Завдяки відсутності необхідності аналізувати сам сенс повідомлення, метод може бути використаний для будь-яких даних, навіть не обов'язково смислових текстів.

Література

1. Guzella, T. G. A review of machine learning approaches to Spam filtering [Text] / S. G. Thiago, M. C. Walmir // Expert Systems with Applications. – 2009. – Vol. 36, Issue 7 – P. 10206-10222. doi: 10.1016/j.eswa.2009.02.037

2. Schwarts, A. SpamAssasin [Text] / A. Schwarts. – O'Reilly Media, 2004. – 224 p.

3. Sahami, M. A Bayesian approach to filtering junk email [Text] / M. Sahami, S. Dumais, D. Heckerman, E. Horvitz // AAAI Workshop on Learning for Text Categorization, 1998. – WS-98-05.

4. Ватолин, Д. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео [Текст] / Д. Ватолин, А. Ратушняк, М. Смирнов, В. Юкин. – М.: Диалог-МИФИ, 2002. – 384 с.

5. Ziv, J. A Universal Algorithm for Sequential Data Compression [Text] / J. Ziv, A. Lempel // IEEE Transactions on Information Theory. – 1977. – Vol. IT-23, Issue 3 – P. 337–343.

6. Benedetto, D. Language Trees and Zipping [Text] / D. Benedetto, E. Caglioti, V. Loreto // Physical review letter. – 2002. – Vol. 88, Issue 4 – P. 1–4. doi: 10.1103/physrevlett.88.048702

7. Алгоритмы, методы, исходники [Электронный ресурс] / Режим доступа: http://algolist.manual.ru/compress/standard/huffman.php

References

1. Thiago, S. G., Walmir, M. C. (2009). A review of machine learning approaches to Spam filtering. Expert Systems with Applications, 36 (7), 10206–10222. doi: 10.1016/j.eswa.2009.02.037

2. Schwarts, A. (2004). SpamAssasin. O'Reilly Media, 224.

3. Sahami, M., Dumais, S., Heckerman, D., Horvitz, E. (1998). A Bayesian approach to filtering junk email. AAAI Technical Report, WS-98-05.

4. Vatolin, D., Ratushnyak, A., Smirnov, M., Yoockin, V. (2002). Data compression methods. Structure of archivers, image and video compression. Moscow, Russia: Dialog-MIFI, 384.

5. Ziv, J., Lempel, A. (1977). A Universal Algorithm for Sequential Data Compression. IEEE Transactions on Information Theory, IT-23 (3), 337–343.

6. Benedetto, D., Caglioti, E., Loreto, V. (2002). Language Trees and Zipping. Physical review letter, 88 (4), 1–4. doi: 10.1103/physrevlett.88.048702

7. Algorithms, methods, source codes. Available at: http://algolist.manual.ru/compress/standard/huffman.php

Рекомендовано до публікації д-р техн. наук, професор Качинскький А. Б.
Дата надходження рукопису 21.05.2015

Сірий Олексій Олександрович, кафедра захисту інформації, Фізико-технічний інститут, Київський національний університет України «Київський політехнічний інститут», пр. Перемоги, 37, м. Київ, Україна, 03056
E-mail: axelgreenkp@gmail.com