УДК 004.415.3

І.М. Кульчицький
Національний університет “Львівська політехніка”,
кафедра прикладної лінгвістики

ТЕХНІЧНІ АСПЕКТИ ОПРАЦЮВАННЯ КОМП’ЮТЕРОМ ПРИРОДНОМОВНОЇ ІНФОРМАЦІЇ

Розглянуто технічні проблеми опрацювання комп’ютером природномовної інформації, які спричинені наявністю багатьох стандартів кодування символів та недотриманням користувачами орфографічних та пунктуаційних правил. Обґрунтовано необхідність попереднього технічного опрацювання таких текстів перед їхнім використанням як у наукових дослідженнях, так і у різних інформаційних системах.

Ключові слова: інформаційний простір, інформаційне середовище, інформаційне суспільство, комп’ютерні технології, кодування символів, стандарти кодування.

The article deals with technical problems of natural language information processing by computer caused by the presence of multiple character encoding standards and non-compliance by users with spelling and punctuation rules. The necessity of previous technical processing of such texts before their use in scientific researches as well as in various information systems has been grounded.

Key words: information space, information environment, information society, computer technologies, character encoding, encoding standards.

Вступ

Сучасне світове суспільство прийнято називати інформаційним, тобто суспільством з високим рівнем опрацювання (створення, перероблення та використання) інформації [2, 41]. У такому разі місце її існування доцільно називати інформаційним простором чи інформаційним середовищем [4, 189; 24, 22]. Під останнім розумітимемо артефактне середовище, об’єкти якого – інформація, її матеріальні носії, суб’єкти, що її опрацьовують, та відповідні засоби її створення, перероблення, використання і передавання. Підґрунтя існування такого середовища – матеріальні носії інформації, адже саме завдяки їм ми маємо до неї доступ та можемо її використовувати. Загалом, інформація існує у найрізноманітніших формах, кожній з яких відповідають свої матеріальні носії. Нас цікавитиме її вербальна форма у електронному вигляді, яка сьогодні становить, насмілимось стверджувати, левову частку у засобах масової комунікації. Адже тепер навіть друковані тексти переважно спочатку створюють у електронному вигляді, а з раніше (до початку масового використання комп’ютерної техніки) виданої поліграфічної продукції створюють відповідні електронні копії.

Постановка проблеми

Дослідження проблем інформаційного простору, зокрема українського, має два аспекти. З одного (назвемо його “гуманітарним”) боку його досліджують [24, 22] фахівці з культурології, соціології, філософії, журналістики, філології та бібліотечної справи тощо, з іншого (“технічного”) – фахівці з соціальних комунікацій (на нашу думку у їхніх дослідженнях переважає технічна компонента) та інформатики. Галузь зацікавлень останніх – здебільшого внутрішнє подання інформації в електронному середовищі, її захист та передавання інформаційними каналами, технічна сторона інформаційного пошуку, створення необхідного лінгвістичного програмного забезпечення. Однак фахівці цих двох груп рідко контактують між собою на наукових конференціях (як правило, це або різні за тематикою конференції, або різні секції на спільній), читають різні наукові журнали і у різних джерелах публікують результати своїх досліджень. Як результат – кожна з груп має поверхневе, а часом і хибне уявлення про проблеми іншої, що, своєю чергою, аж

ніяк не сприяє ефективності функціонування інформаційного простору. **Мета** статті – частково усунути цей бар'єр.

Аналіз досліджень та публікацій

Дослідженнями інформаційного простору займались такі учені, як [4, с. 188–189] І. Вінічук, А. Манойло, С. Грицай, В. Ільганаєва, В. Гастинщиков, В. Карпенко, О. Романенко, Л. Білоус; [24, с. 22–23] В. Щербіна, О. Лобовікова, В. Карпенко та Ю. Бондар, О. Ліщинська, В. Малімон, А. Лобанова, О. Злобіна, М. Яковенко, О. Левченко [17], В. Широков [23], Н. А. Ахренова [1], М. Бергельсон [3], Т. Н. Галинська [5], Е. І. Горошко [9] та інші. Ці праці присвячені формулюванню та філософському уточненню змісту поняття "інформаційний простір", визначенню його ознак, дослідженню його соціально-політичних, соціально-психологічних, культурологічних, лінгвістичних аспектів, побудові формальних моделей для автоматизованого дослідження природномовної інформації. У всякому випадку у будь-яких вищевказаних дослідженнях об'єктом дослідження є природномовні тексти. Зрозуміло, що застосування до опрацювання останніх комп'ютерних технологій значно підвищує швидкість отримання та якість наукових результатів [23]. Однак відчутно бракує праць, які б висвітлювали проблеми впливу способів опрацювання текстів комп'ютером на структуру технології відповідного дослідження, склад та порядок використання її елементів. Про окремі з них, на нашу думку, базові, тобто такі, що стосуються будь-яких видів опрацювання природномовних текстів, йтиметься у подальшому викладі матеріалу.

Виклад основного матеріалу

Як зазначено вище, між дослідниками інформаційного простору гуманітарного та технічного напрямів існує певний бар'єр взаємонерозуміння, який, на нашу думку, зумовлений такими чинниками:

- Підручники та література з інформатики або переобтяжені інформацією, яка потребує фахових технічних знань, або у спрощеному варіанті вчать, "яку кнопку слід натиснути" чи "куди клацнути мишею", щоби виконати певну дію, абсолютно не пояснюючи базових засад комп'ютерного опрацювання природномовної інформації. Практично відсутня (або її дуже важко знайти) література, де б вони були пояснені зрозумілою для представників гуманітарного сектору мовою. У результаті інформатична компетентність [7; 8] останніх хибує розумінням того, що можна, а чого не можна вимагати від комп'ютера. Як результат у них формуються такі вимоги до комп'ютера, які людина ставить до іншої людини. При цьому неявно вважається, що ця "людина" має лише "штучний мозок", який мало чим різниться від натурального. Але будь-який комп'ютер може виконати лише те, що закладено у його програмах, і так, як це у них передбачено. А всяка програма – це лише матеріалізовані думки її розробника на предмет того, яким способом і у якій послідовності дій розв'язувати відповідну задачу. А це не завжди збігається (як правило, завжди не збігається) з баченням розв'язку цієї задачі користувачем, що цю програму використовує.
- Під впливом описів, наукових статей оглядового характеру та рекламних проспектів розробників програмного забезпечення у вищезгаданих користувачів здебільшого формується думка, що необхідно лише знайти "правильну програму" і всі проблеми будуть одномоментно вирішені. При цьому вони часто не враховують, що:
  - Основне завдання розробників програмного забезпечення – продати свої програми і тому описане у рекламі та у загальних описах не завжди у деталях збігається з тим, що реально може програма. Наприклад, більшість програм, які чудово працюють з англомовними текстами, не завжди дають такі самі результати для україномовних. Особливо це стосується програм, які здійснюють складне "інтелектуальне" опрацювання електронних текстів.
  - Так само як у рецептах для приготування страв явно не вказують, що перед цим реалізацією інгредієнти необхідно відповідно обробити (наприклад, почистити та помити), так і у настановах користувачеві з використання певної програми часто явно не вказують, що вхідну інформацію необхідно спочатку у певний спосіб приготувати. У результаті користувач часто або отримує хибний результат, що не завжди можна перевірити вручну, або не отримує його взагалі.

- Розробники лінгвістичного програмного забезпечення – фахові програмісти з технічною освітою. Зазвичай їхній рівень розуміння проблем тих наукових галузей, що досліджують різні аспекти природномовних текстів (лінгвістика, соціологія, психологія, філософія тощо) відповідає середньозагальному. І знову ж практично відсутня література, яка могла б їм ці проблеми пояснити зрозумілою для них мовою. Важко це зробити і фахівцям-гуманітаріям, коли вони замовляють відповідне програмне забезпечення. Тому часто у програму можуть бути не закладені саме ті можливості, які потрібні конкретному користувачу для вирішення конкретної проблеми у конкретному дослідженні.

Окрім того, творці електронних текстів часто не враховують тієї обставини, що їхні твори, крім основного вжитку, мають ще й додаткові. По-перше, їх використовують у різноманітних мовних дослідженнях (див., наприклад, у [1; 3; 5; 9; 17]), по-друге, вони – одиниці зберігання у різноманітних електронних сховищах даних. Відповідно це потребує певної уніфікації форми подання текстів, що, своєю чергою, вимагає дотримуватись певних технічних правил комп'ютерного набору текстів. Нехтування цими правилами знову ж таки або призводить до великих затрат на модифікацію текстів до необхідної форми, або до неправильних результатів під час їх дослідження. Наприклад, у середовищі редактора MS Word надзвичайно легко отримати деякі дані, скажімо – кількість абзаців, які використовують у квантитативній лінгвістиці [18]. Але якщо для утворення проміжків між певними абзацами використовувати не правила форматування MS Word, а натискати клавішу Enter, то буде отримано завідомо неправильний результат, позаяк кожен такий натиск утворює порожній абзац, який входить у підрахунки.

Усунення такого бар'єра взаємонерозуміння необхідно, на нашу думку, почати з глибшого розуміння базових принципів комп'ютерного опрацювання інформації представниками гуманітарного сектору інформаційного простору, позаяк до них належить більшість його дійових осіб. Розглянемо ці принципи детальніше. Відомо [21; 22], що опрацювання інформації комп'ютером має такі максими:

- Єдина активна складова комп'ютера, яка виконує всю роботу, – процесор. Своєю чергою, основна складова процесора – набір електронних схем, що у своїй сукупності фізично реалізовують скінченний набір операцій, які називають машинними командами, і лише їх вміє виконувати процесор.
- Процесор розуміє інформацію, яка записана лише послідовностями бітів, кожен з яких набуває двох значень – "0" та "1". Таке подання інформації для комп'ютера називають внутрішнім (на відміну від зовнішнього – для людей). Посередником між зовнішньою та внутрішньою формою подання інформації слугує програмне забезпечення (найчастіше – операційна система, наприклад Windows), яка, за необхідності, перетворює інформацію із однієї форми у іншу.
- Будь-яка програма – це послідовність машинних команд, які опрацьовують інформацію, закодовану бітами у вигляді певних структур. Під такою структурою розуміємо певну кількість бітів, де кожному біту чи групі бітів розробники програми приписують певне значення. Сукупність таких приписів називають форматом внутрішнього відображення інформації. Для кожного її типу існують свої формати.
- На апаратному рівні (існують відповідні машинні команди) процесор розрізняє лише числа (цілі, дійсні, десяткові (не завжди)) та поля (ланцюжки) бітів [20]. Це означає, що останніми необхідно відображати будь-яку, крім чисел, інформацію, зокрема і текстову.
- Який інформаційний зміст закодовано конкретними ланцюжками бітів, знає лише програма (реально – розробник цієї програми), яка цю інформацію записала. Це означає, що для прочитання раніше закодованої у електронному форматі інформації необхідна та програма, яка її у цій формі записувала. Якщо ж ми хочемо прочитати цю інформацію іншою програмою, то нам потрібні кодові таблиці, у яких вказана відповідність між ланцюжками бітів та звичними для людей засобами фіксації певного типу інформації.

У внутрішнє відображення вербальну інформацію зазвичай кодують посимвольно. Це означає, що для ефективного використання людиною електронних текстів їй необхідно чітко усвідомлювати такі речі:

- По-перше, для будь-якої мови необхідні таблиці взаємно однозначного відображення всіх її графічних символів (алфавіт, знаки пунктуації, цифри, спеціальні знаки тощо) і відповідних їм комп'ютерних кодів. Такі таблиці називають "кодовими таблицями" Треба пам'ятати, що різне, нехай і однотипне (наприклад текстові редактори), програмне забезпечення може працювати з різними кодовими таблицями (далі подано їх огляд).
- По-друге, користувачі бачать не коди символів, а відповідні їм графічні знаки завдяки програмному забезпеченню, яке ці ж кодові таблиці і використовує. Наприклад у Windows такої відповідності досягають за допомогою шрифтових файлів (рис. 1).
- По-третє – під час будь-якого опрацювання текстів комп'ютер (маємо на увазі процесор, який виконує відповідну програму) оперує не сукупністю графічних знаків, а сукупністю відповідних їм кодів.

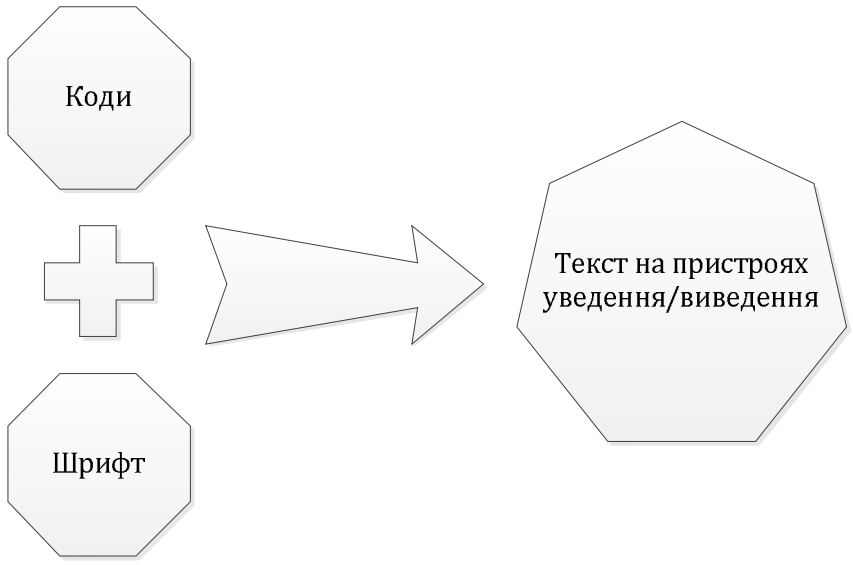

Рис. 1. Внутрішнє та зовнішнє відображення символьної інформації

Такий стан справ породжує певні проблеми, які необхідно вирішувати перед проведенням будь-яких досліджень з електронними текстами.

Проблема перша – різні таблиці кодування. За роки опрацювання текстів за допомогою комп'ютерів розробники апаратного та програмного забезпечення створили велику кількість кодових таблиць. Подамо загальну характеристику основних стандартних таблиць кодування символів за [13; 15; 16; 14; 19; 26]. Перші комп'ютери (США, 50-ті роки минулого сторіччя) були призначені лише для математичних розрахунків. Про відображення текстової інформації (спочатку роздруки програм, потім – природномовні тексти) мова зайшла лише через 10 років. Було прийнято доволі очевидне (за аналогією до телеграфу) рішення про окреме кодування кожного символу, що утворюють текст. У ті часи кожен виробник комп'ютерної техніки (головні відділення таких транснаціональних корпорацій розташовувалися переважно у США) пропонував власну систему кодування символів природних мов, несумісну з системою конкурентів. Так він намагався змусити покупців купувати лише його обладнання, в межах якого була забезпечена сумісність різних пристроїв. Це породило безліч способів кодування символьної інформації, що і не влаштовувало споживачів комп'ютерної техніки, і гальмувало розвиток інформаційних технологій в межах усієї держави, передусім США. Тому відповідному державному органу зі стандартизації (ASA – пізніше ANSI (American National Standards Institute)) було доручено розробити єдиний державний стандарт для внутрішнього відображення символів природномовних текстів. Такий стандарт, як державний для США, було прийнято ANSI у 1963 р., а його остаточну версію у 1968 р. Він отримав назву ASCII – American Standard Code for Information Interchange. В 1967 р. ISO (International Organization for Standardization – головна міжнародна організація зі стандартизації) опублікувала рекомендацію

ISO 646, за якою ASCII де-юре стає міжнародним стандартом. Сьогодні поширені декілька її синонімічних назв: ANSI_X3.4-1968 (канонічна назва); Iso-ir-6; ANSI_X3.4-1986; ISO_646.irv:1991; ASCII (варіанти ASCII-7 та ASCII-8); ISO646-US; US-ASCII (рекомендована назва); US (us); IBM367; cp367; csASCII.

ASCII – семибітний код, що сумарно дає можливість закодувати 128 різних символів. Але 32 коди стандарт вводить для керуючих послідовностей, тобто для кодування власне символів залишено, відповідно, 96. З них – 52 для великих та малих англійських букв, 10 – для арабських цифр, решта – для розділових знаків, пробілу та спецсимволів. Кодування символів інших мов (а їх налічують вже понад 2500, причому кількість символів у деяких, наприклад, японській, перевищує 65000) спочатку підтримували за допомогою технічних хитрувань – спеціальних керуючих (escape) послідовностей, які міняли стандартну таблицю кодування на одну з додаткових, що містила символи відповідної мови. Таких таблиць ISO затвердила загалом більш ніж 180. Згодом стандарт переглянули і він став восьмибітним, що дало змогу кодувати 256 символів. Перші 128 залишено без змін, а решту кодів віддано під інші необхідні символи. Це дозволило відмовитись від керуючих послідовностей, однак абсолютно не усунуло проблему одночасного кодування інформації різними мовами. Кількість вільних кодів була явно менша від кількості символів-претендентів. Тому ISO розробила групу стандартів 8859-X (де X – число від 1 до 15), якими регламентувала кодування символів тих чи інших національних алфавітів. У ньому для кирилиці відведено стандарт 8859-5, яким дуже незручно користуватись, оскільки відсутня буква “Ґ” та деякі інші необхідні символи.

З переходом до 8-бітного коду проблем у комп’ютерному відображенні символів не поменшало. До вищезгаданої проблеми одночасного відображення інформації декількома мовами додалось і те, що і виробники комп’ютерної техніки, і урядовці окремих держав повністю нехтували розробленими стандартами та створювали власні системи кодування і упроваджували їх у програмне забезпечення. Як результат – для однієї мови існувало декілька систем кодування її символів.

Особливо це стосується кирилиці. Так, Apple досі використовує свою таблицю X-Mac-Cyrillic, а на IBM PC-сумісних комп’ютерах ситуація така: у операційній системі DOS використовували кодову таблицю CP866, де для української мови були відсутні букви “І” та “Ґ”, з 1991 р. в Україні упроваджено стандарт RUSCII, у якому повноцінно функціонували білоруська, болгарська, російська та українська мови. Операційна система Windows запропонувала свою таблицю кодування кирилиці CP-1251, яка за кодами абсолютно не збігалась з попередніми. Одночасно в Інтернеті паралельно почала функціонувати застандартизована CPCP (а пізніше Росією) таблиця кодування кирилиці КОИ8, у якій відмінні українські літери були відсутні. Цю таблицю згодом доповнили відповідними літерами і, як результат, сьогодні існують дві подібні таблиці кодування – КОИ8-R і КОИ8-U.

Такий стан справ, особливо коли все більше людей почали одержувати доступ до Інтернету, став серйозною перешкодою для міжнаціонального інформаційного обміну. Назріла необхідність створення єдиної таблиці кодування. З цією метою некомерційна організація “Консорціум Юнікоду” (Unicode Consortium) [26] у 1991 р. запропонувала спосіб кодування символів, який потім було прийнято як стандарт Unicode, що дає змогу закодувати більшість символів існуючих писемностей. Перша його версія була системою кодування з фіксованим розміром символу в 16 бітів, тобто загальна кількість кодів – 2^{16} (65 536). Звідси походить практика позначення символів чотирма шістнадцятковим цифрами (наприклад, U+0410). При цьому в Unicode планувалося кодувати не всі наявні символи, а тільки ті, які необхідні в повсякденному житті. Рідковживані символи розміщено в “області символів для приватного використання” (U+ D800 – U+F8FF. Проте надалі вирішили кодувати всі символи і тому кодовий простір було значно розширено. Одночасно з цим коди символів почали розглядатися не як 16-бітні значення, а як абстрактні числа (кодові позиції), які в комп’ютері можна подати різними способами.

Сьогодні стандарт Unicode (остання версія – 6.3.0 [26]) – це не просто таблиця кодування символів. Основа його – два розділи: універсальний набір символів (UCS, Universal Character Set) та сім’я кодувань (UTF, Unicode Transformation Format). Універсальний набір символів задає

однозначну відповідність між символами та кодовими позиціями (додатні цілі числа). Нині існують дві версії: UCS-2 (одному символу відповідають два байти, застарілий) і UCS-4 (одному символу відповідають чотири байти, основний). Окрім того, UCS визначає різні характеристики символу: тип символу (велика чи мала буква, цифра, розділовий знак тощо), атрибути символу тощо.

Коди в стандарті Unicode розділені на кілька областей. Область з кодами від U+0000 до U+007F містить символи набору ASCII з відповідними кодами. Далі розташовані області знаків різних писемностей, знаки пунктуації та технічні символи. Частина кодів зарезервована для використання в майбутньому. Під символи кирилиці виділені коди від U+0400 до U+052F

Сім'я кодувань (UTF) визначає набір графічних символів і спосіб їх кодування для комп'ютерної обробки текстових даних. Графічні символи ділять на такі групи:

- літери, що належать хоча б одному з алфавітів, внесених у стандарт;
- цифри;
- знаки пунктуації;
- спеціальні знаки (математичні, технічні, ідеограми тощо);
- розділювачі.

Unicode – система для лінійного відображення тексту. Символи, що мають додаткові над- або підрядкові елементи, можуть бути подані або у вигляді побудованої за певними правилами послідовності кодів composite character), або у вигляді єдиного символу (precomposed character). Окрім того, графічні символи стандарту ділять на протяглі та непротяглі. Непротяглі символи, відображаючись, не займають місця у рядку тексту. До них належать, зокрема, знаки наголосу й інші діакритичні знаки. Як протяглі, так і непротяглі символи мають власні коди. Протяглі символи інакше називають базовими (base characters), непротяглі – модифікуючими (combining characters). Останні самостійно вживатись не можуть. Наприклад, символ "á" може бути поданий як послідовність базового символу "a" (U+0061) і модифікуючого символу " ´" (U+0301) або як монолітний символ "á" (U +00 C1).

Unicode має декілька форм кодування символів, що пов'язано з тим, що тексти у цьому форматі: мають більший обсяг; у разі передачі документів через Internet виникає проблема – стандарт несумісний з більшістю інтернет-протоколів. Причина в тому, що вони частину бітів двобайтового простору використовують як службові, а в Unicode цей самий діапазон призначений для іншого. Існують три системи кодування (UTF-8, UTF-16, UTF-32) і шість способів подання кодових позицій (UTF-8, UTF– EBCDIC, UTF-16BE, UTF-16LE, UTF-32BE, UTF-32LE). Сьогодні найпопулярніший – UTF-8. Він забезпечує найкращу сумісність зі старими системами, які використовували 8-бітові символи. Текст, що складається тільки з символів з номером, меншим за 128, у разі запису в UTF-8 перетворюється на звичайний текст ASCII. І навпаки, в тексті UTF-8 будь-який байт зі значенням менше за 128 зображує символ ASCII з тим самим кодом. Решту символів Unicode зображують послідовностями завдовжки від 2 до 6 байтів, в яких перший байт завжди має вид 11xxxxxx, а решта – 10xxxxxx.

Позаяк у Unicode один символ може бути подано різними способами (базовий, або базовий + модифікуючий), стандарт передбачає процеси нормалізації, призначені для приведення тексту до певного стандартному виду. Таких процесів визначено 4 [26]: канонічна декомпозиція; канонічна декомпозиція з подальшою канонічною композицією; сумісна декомпозиція; сумісна декомпозиція з подальшою канонічною композицією.

Однак і Unicode не позбавлений недоліків, хоча переважно вони пов'язані з можливостями обробників тексту, а не безпосередньо з принципом кодування:

- Деякі системи письма все ще не представлені належно в Unicode. Наприклад, відсутні деякі букви традиційної писемності церковнослов'янської мови. Ця писемність містить багато додаткових графічних елементів (такі як титла і виносні літери). Зображення "довгих" надрядкових символів, що простягаються над декількома буквами, досі не реалізовано.
- Тексти китайською, корейською та японською мовою мають традиційне написання зверху вниз, починаючи з правого верхнього кута. Перемикання горизонтального і вертикального

написання для цих мов не передбачено в Unicode – це мають робити засоби мов розмітки або внутрішні механізми текстових процесорів.

- Первісна версія Unicode припускала наявність великої кількості готових символів, потім віддали перевагу поєднанню букв з діакритичними модифікуючими знаками (Combining diacritics). Водночас багато символів мов з алфавітами на основі кирилиці не мають монолітних форм.
- Unicode передбачає можливість різних накреслень того самого символу залежно від мови. Так, китайські ієрогліфи можуть мати різні накреслення в китайській, японській (кандзі) і корейській (ханчча), Але при цьому в Unicode їх позначено тим самим символом (так звана CJK-уніфікація), хоча спрощені й повні ієрогліфи все ж мають різні коди. Часто виникають накладки, коли, наприклад, японський текст виглядає “по-китайськи”. Тому потрібно стежити за правильним належномовним маркуванням тексту.
- Файли з текстом в Unicode займають більше місця в пам’яті, оскільки один символ кодують не одним байтом, як у різних національних кодуваннях, а послідовністю байтів. Однак зі збільшенням потужності комп’ютерних систем і здешевленням пам’яті та дискового простору ця проблема стає все менш істотною.
- Хоча підтримка Unicode реалізована в найпоширеніших операційних системах, не все прикладне програмне забезпечення підтримує коректну роботу з ним. Ця проблема тимчасова.

Отже, наявність різних таблиць кодування символів та їхні внутрішні особливості вимагають обов’язкового приведення всіх задіяних у дослідженнях текстів до єдиної кодової таблиці (сьогодні – Unicode) та єдиного способу внутрішнього подання символів. Інакше говорити про правильність результатів порівняння рядків символів, різноманітних квантитативних досліджень (наприклад, підрахунку кількості символів у словах) тощо дуже проблематично.

Проблема друга. Відображення кодів за допомогою шрифтових файлів. Суть цієї проблеми така. Основне завдання шрифтових файлів – надати службовим програмам, які безпосередньо візуалізують коди символів на пристроях введення-виведення, інформацію про відповідність графічного і кодового подання символу. При цьому виникають такі колізії (рис. 2, 3).

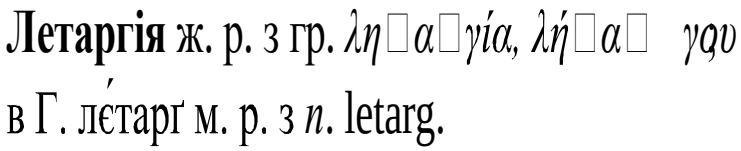

Рис. 2. Неповнота чи некоректність шрифтового файла

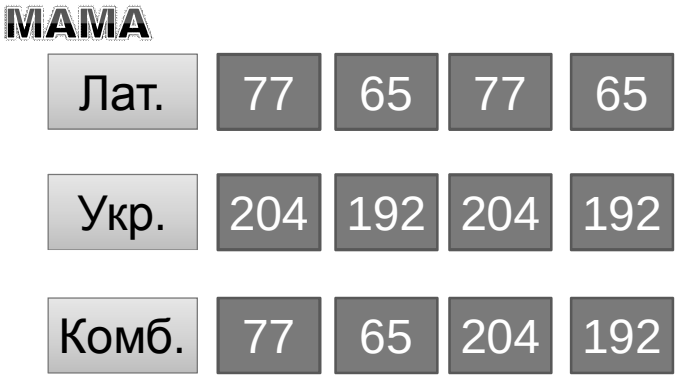

Рис. 3. Однакові за графікою букви – різні коди

По-перше, замість окремих символів користувач, у кращому випадку, побачить або прямокутники, або знаки запитання, або “кракозябри” – незрозумілі символи. Це означає, що у файлі шрифту або відсутні графічні зображення для певних кодів (прямокутники чи знаки питання), або графічне зображення не відповідає набору символів мови, якою написано текст.

По-друге, може трапитись такий випадок, коли графічно (для людини) слово (наприклад, МАМА) правильне, а за кодами (для комп’ютера) – це різні слова. Особливо така ситуація можлива у разі використання тих шрифтів, у яких однакові за виглядом букви різних алфавітів мають абсолютно однакову графіку. Це означає, що, окрім приведення всіх текстів до єдиної таблиці кодування, необхідні дві речі:

- наявність шрифтів, які би коректно і повною мірою відображали тексти;
- перевірка за допомогою спеціальних програм (візуально це майже неможливо зробити) кодування слів певної мови.

Проблема третя. Вибір текстового редактора. Очевидно, що для останнього етапу створення всякого тексту використовують текстовий редактор. Першу проблему породжує вибір користувачем текстового редактора. Якщо це нехай “запозичена”, але версія текстового редактора (MS Word, OpenOffice Word, редактор веб-переглядача Google Chrome) відомої фірми, тоді все гаразд. Такі програми акуратно працюють з кодовими таблицями, забезпечують безпомилкове перекодування тексту з однієї таблиці у іншу, коректно (згідно з чинними відповідними стандартами та домовленостями де-факто) запам’ятовують текст у вибраному користувачем форматі. Якщо ж вибраний редактор, нехай і безкоштовний, але доморощена розробка, то велика ймовірність того, що створений і записаний ним текст не зможуть правильно прочитати інші, не обов’язково текстові редактори, програми опрацювання текстів. До речі, це одна з причин того, чому часто ми не можемо прочитати отриману через Інтернет інформацію. Необхідно враховувати ще й те, що сучасні версії популярних текстових редакторів (MS Word) в процесі набору тексту замінюють одні символи іншими [18]. Така ситуація теж вимагає додаткових перевірок тексту спеціалізованими програмами перед їх подальшим використанням.

Проблема четверта. “Лінощі – джерело усіх хиб та ґанджів”. Полягає вона у звичайних лінощах користувачів, які умовно можна охарактеризувати фразою: “А!.. Яка різниця – і так зрозуміють…”. Зміст її такий.

По-перше, якщо для людини послідовність символів “ім’я” і “ім`я” – те саме слово, то для комп’ютера здебільшого (якщо у програмі не передбачено спеціального опрацювання таких випадків) це різні слова – графічним символам “’” і “`” у всякій кодовій таблиці відповідають різні коди.

По-друге, коли комп’ютерна техніка ще не набула поширення, то під час виготовлення поліграфічної продукції писані від руки чи надруковані на друкарській машинці тексти обов’язково перенабирали. Здійснювали цей перенабір фахівці, які чітко дотримувались чинних правил [6; 14]. Коли почали поширюватись комп’ютери та їх мережі, між авторами текстів та споживачами останніх здебільшого практично випала ланка фахового набору та форматування текстів. Це означає, що відповідають за фаховість вищевказаних компонент тексту автори. Однак аналіз україномовних текстів у інформаційному просторі показує, що стан справ з цією компонентою залишає бажати кращого. Найбільше помилок роблять, вживаючи апостроф, дефіс та тире. Про-аналізуємо ці помилки детальніше.

Позаяк апостроф відсутній у стандартній розкладці української клавіатури у середовищі Windows, то ліниві користувачі набирають замість нього або зворотний апостроф, або подвійні лапки, або знак зірочки. Таким користувачам скажемо: по-перше, у будь-якому редакторі апостроф можна набрати комбінацією “лівий Alt + 0039 на цифровій клавіатурі, увімкнувши її стані Num Lock”, по-друге, така заміна порушує пунктуаційні й набірні правила та утруднює чи унеможливлює якість та ефективність автоматизованого опрацювання текстів, по-третє, коли хтось вказує на такі тексти і стверджує, що українці безграмотні й не шанують своєї мови, то для серйозного опонування бракує аргументів.

Стосовно дефісу та тире, то у кодових таблицях для їх відображення наявні три різні за довжиною (і, відповідно, за кодами) знаки, які мають назви “дефіс-мінус”, “коротке тире” і “довге

тире”. Коли і який з них набирати, регламентовано у [14]. Але у автора склалося враження, що таких правил більшість користувачів комп’ютера не читала, а дехто з них просто навіть не знає, що такі правила існують. Результат – повний розгардіяш у використанні цих символів. Повчальним буде, наприклад, такий факт. У [14] вказано, що для позначення на письмі тире при наборі слід використовувати знак “довге тире”. Однак навіть у тексті ДСТУ ГОСТ 7.1-2006 [10] в описі знаків приписаної пунктуації для бібліографічного опису для зображення складеного знака, що має назву “точка і тире” замість комбінації “.–” (“довге тире”) використовують комбінацію “.–” (“коротке тире”). Принагідно вкажемо на певну неточність у формулюваннях цього стандарту [10, пункти 4.7.5–4.7.6]. Там вказано, що для розрізнення граматичної та приписаної пунктуації (тобто розділових знаків між зонами бібліографічного опису та їх елементами) застосовують проміжок в один друкований знак до і після приписаного знака. Виняток становлять: крапка і кома – проміжки ставлять тільки після них, а також квадратні і круглі дужки, які виділяються проміжками лише ззовні. Користувачі ж, складаючи список літератури, не дуже вникають у зміст цих пунктів і не враховують того, що комбінація “.–” єдиний приписаний пунктуаційний знак і, переживаючи, що статтю, монографію, звіт тощо не приймуть, вставляють між ними пробіл. Не дуже стежать за цією ситуацією і відповідальні за друк наукових, навчальних та науково-популярних видань. Прикладів довго шукати не потрібно. Достатньо уважно переглянути бібліографічні списки будь-якого наукового журналу, монографії чи підручника. Однак і ця вільність перешкоджає під час використання електронних текстів.

Висновки

Автоматизовано опрацьовуючи електронні тексти, комп’ютер має справу не з символами, а їх бітовими кодами, які організовані у спеціальні коди.

Немає сенсу зберігати тексти, систему кодування яких не визначено.

Існує велика кількість усталених та частково застандартизованих кодових таблиць. Тому сукупність текстів, яку отримано з різних джерел, яку заплановано чи використати у наукових дослідженнях, чи помістити у певне сховище, необхідно нормалізувати – звести всі тексти до однієї кодової таблиці, перевірити тексти на пунктуаційну коректність (наприклад, перевірити, чи знак апострофа всюди позначено одним символом), усунути зайві символи (наприклад, декілька прогалин підряд, порожні абзаци тощо), уніфікувати засоби та способи форматування текстів тощо.

Під час набору текстів необхідно користуватись поліграфічними правилами набору текстів. Для цього відповідні фахівці повинні ці правила узагальнити, гармонізувати їх з доступними у таблицях кодування (найкраще Unicode) і можливостями найпопулярніших текстових редакторів та довести їх до якнайширшого загалу.

1. Ахренова Н.А. Интернет-лингвистика: особенности аффиксации в языке интернета [Електронний ресурс] // Сибак. Научно-практические конференции ученых и студентов с дистанционным участием. Коллективные монографии: [сайт]. – 2014. – Режим доступу до статті: http://sibac.info/index.php/2009-07-01-10-21-16/1180-2012-02-10-08-02-01. 2. Бебик В. В. Глобальне інформаційне суспільство: поняття, структура, комунікації / В.М. Бебик // Інформація і право. – 2011. – № 1(1). – С. 41–49. 3. Бергельсон М. Языковые аспекты виртуальной коммуникации // Вестник Московского университета. Серия 19. Лингвистика и межкультурная коммуникация. – 2002. – № 1. – Режим доступу до статті: http://www.rik.ru/vculture/seminar/index.html. 4. Біловус Л. І. Український інформаційний простір: сьогодення та перспективи [Електронний ресурс]. – Режим доступу : http://ijimv.knukim.edu.ua/zbirnyk/1_1/bilovus_l_i_ukrayinskyy_informatsiynyy_prostir.pdf. 5. Галинская Т. Н. Контент-аналитическое исследование медийного образа российского политика: (на материале комментариев интернет-пользователей о Б. Немцове) [Електронний ресурс] / Т. Н. Галинская // Политическая лингвистика : науч. журн. – 2013. – N 4. – С. 91–98. – Библиогр.: С. 97–98. – Режим доступу до статті: http://cyberleninka.ru/article/n/kontent-analiticheskoe-issledovanie-mediynogo-obraza-rossiyskogo-politika-na-materiale-kommentariev-internet-polzovateley-o-b-nemtsove . 6. Гиленсон П. Г. Справочник художественного и технического редакторов /

П. Г. Гиленсон. – М.: Книга, 1988. – 526 с. 7. Головань М. С. Інформатична компетентність: сутність, структура та становлення / М. С. Головань // Інформатика та інформаційні технології в навчальних закладах: наук.-метод. журнал. – 2007. – № 4. – С. 62–69. 8. Головань М.С. Інформатична компетентність як об'єкт педагогічного дослідження / М. С. Головань // Проблеми інженерно-педагогічної освіти: зб. наук. праць. – Харків: УІПА, 2007. – № 16. – С. 314–324. 9. Горошко Е.И. Психолингвистика Интернет-коммуникаций [Електронний ресурс] // Текстология : [сайт]. – 1999 – 2011. – Режим доступу до статті: http://www.textology.ru/article.aspx?aId=198 10. ДСТУ ГОСТ 7.1-2006. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання : чинний з 2007-07-01. – К.: Держспоживстандарт України, 2007. – 47 с. (Система стандартів з інформації, бібліотечної та видавничої справи) (Національний стандарт України). 11. Дубас О. П. Інформаційно-комунікативний простір: поняття, сукупність, структура [Електронний ресурс]. – Режим доступу : http://dspace.nbuv.gov.ua/bitstream/handle/123456789/26693/22-Dubas.pdf?sequence=1. 12. Женченко М. І. Автоматизація редакційно-видавничого процесу в цифровому суспільстві: сегмент програмного забезпечення [Електронний ресурс]. – Режим доступу : http://ijimv.knukim.edu.ua/zbirnyk/1_1/zhenchenko.pdf 13. Кодовая суматоха [Електронний ресурс]. – Режим доступу : http://www.ht.ua/pub/40659.html 14. Крайнікова Т. С. Коректура: підручник / Т. С. Крайнікова.– К.: Наша культура і наука, 2005 . – 252 с. 15. Кульчицький І. М. Кодування символів української абетки 8-бітними кодами. РСТ УРСР 2018-91. Республіканський стандарт Української РСР / В. С. Костирко, І. М. Кульчицький, А. Ю. Мединець та ін. – К.: Міністерство економіки УРСР, 1991.– 14 с. 16. Кульчицький І. М. Розташування символів української абетки на клавіатурах. РСТ УРСР 2019-91. Республіканський стандарт Української РСР / В. С. Костирко, І. М. Кульчицький, А. Ю. Мединець та ін.– К.: Міністерство економіки УРСР, 1991 .– 18 с. 17. Левченко О. П. Лінгвокультурологія vs. інтернет / О. П. Левченко // Лінгвістика. – Луганськ: ЛНПУ, 2010. – № 2 (23). – С. 10–17. 18. Мирошниченко П. П. Новичок. Word 2010: создание и редактирование текстовых документов / П. П. Мирошниченко, А. И. Голицын, Р. Г. Прокди.– СПб.: Наука и Техника, 2010 .– 192 с. 19. Небольшой обзор кодировок кириллицы [Електронний ресурс]. – Режим доступу : http://segfault.kiev.ua/cyrillic-encodings/ 20. "Проблема кодировок": стечение обстоятельств или стратегический замысел? [Електронний ресурс]. – Режим доступу : http://bugtraq.ru/library/misc/encoding.html. 21. Таненбаум Э. Архитектура компьютера / Э. Таненбаум. – 5-е изд. (+CD) .– СПб: Питер, 2007. – 844 с. 22. Таненбаум Э. Операционные системы: разработка и реализация (+CD). Классика CS / Э. Таненбаум, А. Вудхалл.– СПб: Питер, 2006 .– 576 с. 23. Феноменологія лексикографічних систем: монография / В. А. Широков; НАН України. Укр. мов.-інформ. фонд. – К.: Наук. думка, 2004. – 327 с. 24. Яковенко М. Інформаційний простір: філософські аспекти формування поняття [Електронний ресурс]. – Режим доступу : http://ena.lp.edu.ua:8080/bitstream/ntb/10307/1/4.pdf. 25. Assembler: учебник для вузов / В. И. Юров. – 2-е изд. – СПб. : Питер, 2003. – 637 с. 26. The Unicode Consortium [Електронний ресурс]. – Режим доступу : http://www.unicode.org/