Issue 7, 2020

https://doi.org/10.23939/sisn2020.07.051

УДК 004.415.3

УНОРМОВУВАННЯ ТЕКСТУ ПІД ЧАС ДОКОРПУСНОГО ОПРАЦЮВАННЯ: ДОСВІД ЗАСТОСУВАННЯ

Ігор Кульчицький

Національний університет "Львівська політехніка"
bis.kim@gmail.com, ORCID 0000-0001-9550-9739

Узагальнено досвід унормування текстів перед внесенням їх у корпус творів Наддністрянської України, створення якого розпочато на кафедрі прикладної лінгвістики Львівської політехніки. Йдеться про тексти художнього стилю. Під унормуванням розуміємо сукупність інформаційних процедур, що роблять текст придатним до внесення його в корпус: приведення всіх текстів до однієї кодової таблиці, перевірку їх на пунктуаційну коректність (однакові за смислом сутності мають бути позначені одним знаком), усунення зайвих символів (наприклад, порожні абзаци, декілька пробілів поспіль і т. ін.), уніфікацію засобів та способів форматування тощо. Як програмне середовище унормування запропоновано редактор MS Word, а для створення додаткового програмного інструментарію – мову програмування Python. Процес унормування текстів містить такі етапи: унормування кодування, унормування графіки, коректура тексту, технічне унормування пунктуації. Для кожного етапу подано його характеристику, вказано проблеми, які виникають при його реалізації та запропоновано шляхи їх подолання. Зроблено висновки.

Ключові слова: корпус текстів, унормування, кодові таблиці, графіка тексту, коректура тексту, пунктуація.

Постановка проблеми

Починаючи з 90-х років минулого століття все частіше основним інструментом лінгвістичних досліджень стають корпусні технології. Їхня основа – корпус текстів, під яким розуміємо електронну збірку текстів, яку споряджено фаховою лінгвістичною інформацією у придатній для опрацювання комп'ютером формі та, за необхідності, програмним знаряддям, яке спрощує доступ до цієї інформації [16]. Великою перевагою таких технологій є те, що дослідникам мови не доводиться покладатися на власну інтуїцію чи на інтуїцію носіїв мови або навіть на вигадані приклади. Вони можуть користуватися великою кількістю автентичних, природних лінгвальних даних, отриманих різними мовцями чи письменниками, щоб підтвердити або спростувати власні гіпотези щодо тих чи інших мовних явищ [2]. Збирання автентичних мовних даних з корпусу дає змогу описувати мову, починаючи з доказів, а не з нав'язування певної теоретичної моделі [5]. Позаяк в Україні корпусні технології перебувають на етапі розвитку та становлення, актуальними залишаються проблеми створення, наповнення та використання корпусів.

Аналіз останніх досліджень та публікацій

Дослідженнями в царині сучасної корпусної лінгвістики займаються такі науковці, як Адам Пшепьорковські (Adam Przepiórkowski, Польща), Барбара Левандовська-Томащик (Barbara Lewandowska-Tomaszczyk, Польща), Беатрікс Буссе (Beatrix Busse, Німеччина), Гевін Брукс (Gavin

Brookes, Великорбританія), Грем Кеннеді (Graham Kennedy, США), Джефрі Вільямс (Geoffrey Williams, Франція), Мікаела Мальберг (Michaela Mahlberg, Великорбританія), Дуглас Бібер (Douglas Biber, США), Лін Боуке (Lynne Bowker, Канада), Марек Лазінські (Marek Łaziński, Польща), Міослав Банько (Mirosław Bańko, Польща), Нік Еллін (Nick Ellis, Уельс), Перез Паредес (Perez Paredes, Іспанія), Пол Бейкер (Paul Baker, Великорбританія), Пьотр Пензік (Piotr Pęzik, Польща), Рафаль Горскі (Rafał L. Górski, Польща), Сільвія Бернардіні (Silvia Bernardini, Італія), Сюзан Конрад (Susan Conrad, США), Франциšек Черняк (František Čermák, Чехія), Хуанг Чу-Рен (Huang Chu-Ren, Китай),Штефан Гріс (Stefan Th. Gries, США), Яо Яо (Yao Yao, Китай), та багато інших. У сфері їхніх зацікавлень перебуває використання корпусів для дослідження за допомогою частотності появи, асоціативних мір, дисперсії, кореляції та інших статистичних показників асоціацій одних мовних одиниць (як правило, слова) з іншими (як правило, це також слово чи синтаксична конструкція) [3, 4]; для частотного аналізу n-грам [6]; у комп'ютерних дослідженнях природної мови, когнітивній лінгвістиці [1] тощо.

Серед українських вчених у галузі корпусної лінгвістики працюють такі вчені, як Т. В. Бобкова [8]; О. І. Ванівська [9]; І. Г. Данилюк [11]; Н. П. Дарчук [12], А. П. Загнітко [13, 14], І. М. Кульчицький [15, 16]; В. А. Широков [21] та ін. Їхні праці присвячені огляду дискусій про місце корпусної лінгвістики та корпусних технологій у сучасній лінгвістиці, теоретичному аналізу понять корпусу, його визначальним рисам, класифікації корпусів, галузей та методів їх використання, технічним проблемам залучення текстів у корпуси тощо.

Формулювання цілі статті

Перший етап створення корпусу – це збирання даних, що передбачає отримання текстів в електронній формі чи ручним складанням, чи розпізнаванням за допомогою програм OCR, чи як результат роботи текстового процесора, чи з PDF-файла тощо. Дані, отримані в електронному вигляді з інших джерел, майже завжди містять коди форматування та іншу інформацію, яку треба забрати або перетворити на придатну для комп'ютерного аналізу форму [5]. Такий процес називаємо внормовуванням тексту, під яким розуміємо сукупність інформаційних процедур, що роблять текст придатним до внесення його в корпус: приведення всіх текстів до однієї кодової таблиці, перевірку їх на пунктуаційну коректність (однакові за смислом сутності мають бути позначені одним знаком), усунення зайвих символів (наприклад, порожні абзаци, декілька прогалин підряд і т. ін.), уніфікацію засобів та способів форматування тощо [17].

Власний досвід доводить, що процес збирання та готування текстів до внесення їх у корпус є стилезалежним. Звичайно, багато дій є однаковими для всіх стилів тексту. Однак завжди у структурі тексту будуть особливості, що притаманні лише конкретному стилю (наприклад, поклики на джерела в науковому стилі). Такими особливостями можна нехтувати під час роботи з текстами одного стилю й обов'язково враховувати, опрацьовуючи тексти інших стилів.

Мета цього дослідження – осмислення досвіду унормування текстів перед внесенням їх у корпус творів Наддністрянської України, створення якого розпочато на кафедрі прикладної лінгвістики Національного університету "Львівська політехніка". Мова йтиме про тексти художнього стилю [19].

Виклад основного матеріалу

Для початку декілька підходів стосовно автоматизації процесу внормування текстів перед внесенням їх до корпусу. Звичайно, ідеальною би була ситуація, коли все реалізовувалося би автоматично, без участі людини. Однак сучасний стан програмного забезпечення, яке опрацьовує тексти природою мовою, зробити цього не дає змоги. З огляду на це, на мою думку, найкращим є варіант "трохи ручної роботи, трохи програмування, трохи сторонніх програм" (детальніше про це можна прочитати у [16]). Очевидно, що процес досягнення кінцевого результату уповільнюється. Проте інформаційна якість у такий спосіб укладеного фаховим персоналом корпусу зросте на порядок, що для мовознавчих студій має далеко не останнє значення.

Готування тексту до внесення його у корпус – процес етапний і певною мірою ітераційний: деякі етапи для досягнення успіху доводиться повторювати декілька разів. Насамперед необхідно роздобути сам текст в тій чи іншій формі. При цьому необхідно врахувати те, що в лінгвістиці для дослідження текстів творів автора науково достовірним вважають тексти останніх прижиттєвих видань. Згадані джерела не завжди вдається знайти. Тоді варто намагатися знайти одне з останніх прижиттєвих. Придатними для досліджень можна вважати академічні видання, хоча й вони не позбавлені недоліків [18]. У будь-якому випадку завжди треба фіксувати бібліографічний опис видання твору, текст якого буде додано в корпус. Після утворення збірки текстів, що складатимуть корпус, настають етапи їхнього унормування. При цьому необхідно вирішити низку проблем. Розгляньмо їх детальніше.

Вибір програмного забезпечення

Зрозуміло, що інформаційні процедури унормування тексту необхідно реалізовувати в певному програмному середовищі. Найкраще це, на мій погляд, реалізовувати в середовищі MS Word, а допоміжне програмне забезпечення створювати мовою Python. Цей вибір зумовлений такими чинниками.

- Редактор MS Word у своїй роботі використовує узагальнену програмно-об’єктну модель тексту. Доступ до об’єктів цієї моделі користувач має за допомогою макрокоманд, які написані об’єктно-орієнтованою мовою програмування VBA (Visual Basic for Applications). Це дає змогу створювати у програмному середовищі VBA достатньо серйозні програми обробітку природомовних текстів. Так, інші редактори середнього та складного класів теж мають можливість створювати макрокоманди. Проте підтримка редактором MS Word повноцінної мови програмування робить його однозначним лідером у цьому напрямі.
- Редактор MS Word вміє відкривати файли текстового формату у близько 140 різних системах кодування та у форматах інших текстових редакторів тощо. Це дає змогу достатньо легко привести всі наявні тексти до однієї системи кодування. Сьогодні для українських текстів це, мабуть, Unicode (Utf8).
- Позаяк у внутрішньому поданні MS Word текст — це набір об’єктів, кожний із яких представлений набором своїх атрибутів, то можна легко контролювати низку лінгвістичних характеристик складників тексту (наприклад, мову, до якої належить слово у тексті). Ці характеристики редактор візуалізує за допомогою спеціальних панелей чи вікон, що абсолютно не утруднює читання та сприймання тексту користувачем.
- Об’єктно-орієнтовану мову програмування Python спроектовано так, що її засобами доволі легко реалізовувати алгоритми опрацювання природомовних текстів. Реалізація її як інтерпретатора дозволяє легко створювати невеликі програми і тут же їх використовувати. З огляду на це, деякі інформаційні процедури унормування тексту доцільніше і швидше створювати в цьому програмному середовищі.

Такий програмний вищевказаними мовами комплекс і був створений на кафедрі прикладної лінгвістики для опрацювання текстів, що в майбутньому стануть інформаційною основою корпусу текстів Наддністрянської України. Програми мовою VBA створено у вигляді сукупності макрокоманд, які містяться в тілі документа MS Word. Передбачено, що ті файли, які будуть опрацьовані, розміщені в наперед відомій теці, місце й назву якої можна попередньо налаштувати. Ті процедури, які реалізовані у середовищі Python, оформлені у вигляді окремих програм, кожну з яких необхідно викликати окремо. У подальшому їх заплановано об’єднати у вигляді окремого проєкту.

Унормування кодування текстів

Позаяк у своєму внутрішньому форматі останні версії MS Word працюють у кодуванні Unicode, процедура зводиться до почергового відкривання файлів з текстами в середовищі MS Word з подальшим їх збереженням у форматі docx. Якщо кодування отриманих файлів невідоме, то процедуру доведеться виконувати вручну. Якщо MS Word не розпізнає формату чи кодування

отриманих файлів, то сьогодні існує єдина рекомендація – пошукати тексти в іншому прийнятному кодуванні. До сьогодні таких випадків не траплялося, що додатково підтверджує правильність вибору редактора.

Унормування графіки

Наступні дві процедури – унормування графіки [19] та коректуру тексту – можна виконувати як почергово, так і паралельно. Досвід показує, що розпочинати краще з унормування графіки – процедура, за допомогою якої з тексту забираємо всі непотрібні знаки, а для тих, що залишились, добиваємося, щоби графема тексту, що означає ту саму сутність, була закодована тим самим кодом Unicode. Зміст цього такий.

Множину графем тексту UG зазвичай розбивають на 5 підмножин:

$$UG = \bigcup_{i=1}^{6} G_i,$$

де G_1 – множина букв мови тексту (у нашому випадку – українська абетка); G_2 – множина неалфавітних орфографічних символів (дефіс, апостроф, наголос); G_3 – множина пунктуаційних знаків; G_4 – дужки та лапки; G_5 – множина значків (наприклад, № , §, ‰, ◊ тощо); G_6 – множина літер інших алфавітів, що можуть зустрітись у тексті.

Якщо з літерами алфавіту української мови та алфавітів будь-яких мов, які використовують в українських текстах у вигляді вкраплень, більш-менш все зрозуміло, то з нелітерними орфографічними та пунктуаційними символами не все так очевидно. Розгляньмо це на прикладі української мови.

Апостроф. Стандарт Unicode має 7, а з врахуванням назви і зовнішньої подібності – 9 кодів [UTF], що відповідають за виглядом значку апострофа. Тому дуже часто на місці апострофа стоять значки з різним кодуванням (про випадки вжитку на місці апострофа зірочки, лапок, зворотнього апострофа мова не йде взагалі). За допомогою клавіатури можна набрати один із них – U+0029. В епоху однобайтового кодування символів його використовували на письмі у будь-якому випадку (апостроф, прості лапки, модифікатор букви тощо). Натомість Unicode пропонує на місці апострофа вживати значок U+2019. Редактор MS Word автоматично вставляє його при наборі українського тексту. Починаючи детальний аналіз застосування всіх 9 варіантів, що потребує окремої розвідки, зазначу, що вищеописане програмне забезпечення нормалізує застосування апострофа так: для української мови використовує U+2019, для чужоземних мов – U+0039. Таке розрізнення, на мою думку, необхідне тому, що апострофи в українські і, наприклад, в англійській мові мають різне смислове навантаження. Зрештою, це значки, що належать до графіки різних мов.

Наголос. Відображати наголос можна двома способами: або ставити на місце наголошених букв букви з діакритичними знаками, або позначати спеціальним знаком (U+0301), який ставлять після наголошеної букви. Прийнятним є тільки другий спосіб, позаяк перший змішує у словах різні алфавіти та ускладнює алгоритми опрацювання тексту.

Дефіс та тире. У стандарті Unicode є близько 60-ти кодів [7], які у тексті відображені горизонтальним рисками. Деякі з них екзотичні, які більшість шрифтів не відображають. У наявних електронних текстах використання на місці дефісу та тире відповідних значків достатньо довільне. Переважно це пов’язане з тим, що українська й американська графіка мають різні традиції, а основний засіб набору тексту в Україні – MS Word, зрештою як і інші “запозичені” редактори текстів, налаштовані на автоматизацію вводу символів саме в американській традиції. Рідко хто вміє й хоче (а навіщо – і так зрозуміють) редактор перелаштовувати. Описане програмне забезпечення нормує кодування значків дефісу та тире так: дефіс – значком, який у стандарті має назву “дефіс-мінус” (U+002D), тире – значком “довге тире” (U+2014). Коротке тире (U+2019) дозволене тільки при заданні числових діапазонів. Знак “мінус” має свій окремий код – U+2212. У

подальшому заплановано перевести дефіс у код символа, який у стандарті має назву “дефіс” (U+2010), а тире у діалогах – на символ “тире, що використовується при цитуванні” (U+2015).

Лапки. Лапок є декілька різновидів та декілька національних традицій їх уживання [7]. Створене програмне забезпечення нормує вживання лапок згідно з чинним правописом [20]: основні – “(U+00AB) та” (U+00BB), допоміжні (якщо цитата в середині іншої цитати) – “(U+201E)” та “(U+201E)”.

Коректура тексту

Серед усіх процедур докорпусного опрацювання текстів ця процедура чи не найважче піддається автоматизації. У цьому випадку під коректурою розуміємо приведення електронного тексту в повну відповідність до паперового оригіналу. Виняток можна зробити тільки для явних технічних огріхів. Хоча в ідеалі про них тим чи іншим способом необхідно полишати слід. Практично весь обсяг робіт користувач вимушений робити власноруч (точніше – “власнооч”). А коректура тексту, навіть отримана за допомогою програм оптичного розпізнавання символів (OCR), дуже необхідна, якщо ми хочемо в майбутньому отримати якісні мовознавчі результати. Причина в цьому така: OCR-програми у своїй роботі використовують словники слів тієї мови, яку розпізнають. Тому дуже часто замість оригінального слова у розпізнаний текст потрапляє або йому подібне з іншою буквою, або декілька, із яких можна приблизно скласти це слово. Подам два приклади. Перший отримано під час розпізнавання новел М. Яцківа: було – “*Ти не **виділа** газди?*”, Fine Reader розпізнав – “*Ти не **ви діла** газди?*”. Наступні помилково розпізнані фрагменти отримано під час роботи з корпусом ГРАК [10]: “*У **Ш** тис. до н. е. рабів у Шумері було небагато , … тобто вони мали право **заво- дити** сім'ї і навіть викупити себе з неволі*”; “…на дві господарські **те- риторії** : Польщу “А” , до якої входили корінні польські землі , і Польщу “Б” , що складалася переважно із **захоп- лених українсіких** та білоруських земель . Дешевими **кре- дитами** та державними замовленнями промисловий **роз- виток** Польщі *А “ підтримувався і стимулювався , в **укра - їнських** же землях кредитування промислових **підпри- ємств** різко **обяежувалося**”. Ніхто не заперечує корисності й потрібності корпусу ГРАК, зрозумілі і причини такої ситуації – повне нехтування державою необхідністю фінансового забезпечення таких проєктів, але використання результатів, отриманих із такого корпусу, вимагає ще додаткового опрацювання й коригування. Тому цієї ручної процедури не оминути. Так, довго й дорого, але досконаліть вимагає жертв.

Технічне унормування пунктуації

Цією процедурою перевіряють дотримання правил використання пробілів зі знаками пунктуації. Відповідна програма вищеописаного програмного забезпечення забезпечує дотримання таких правил:

- у тексті не може бути двох пробілів підряд;
- у тексті не може бути порожніх абзаців;
- абзац не має починатися й закінчуватися пробілом;
- перед комою, крапкою, крапкою з комою, двокрапкою, знаком питання, знаком оклику, трикрапкою не має бути пробілу;
- після коми, крапки, крапки з комою, двокрапки, знаку питання, знаку оклику, трикрапки має бути пробіл;
- дефіс пробілами не обмежують;
- довге тире обмежують із двох сторін пробілами;
- між комою та тире пробіл не потрібний;
- якщо коротке тире стоїть між двома числами, заданими цифрами, то його з двох сторін пробілами не обмежують;
- якщо коротке тире стоїть між двома числами, заданими словами, то його обмежують із двох сторін пробілами;
- перед лівими лапками та дужками ставлять пробіл;

- після лівих лапок та дужок пробіл не ставлять;
- перед правими лапками та пробіл не ставлять;
- після правих лапок та дужок пробіл має бути, якщо наступний знак не пунктуаційний.

Опрацьований у такий спосіб текст готовий до наступного етапу — явного спорядження тим чи іншим способом фаховою лінгвістичною інформацією.

Висновки

Однією з процедур, що передує внесенню творів письменників у корпус, є нормування їхніх текстів – приведення всіх текстів до однієї кодової таблиці, перевірка їх на пунктуаційну коректність, усунення зайвих символів, уніфікація засобів та способів форматування тощо. Стан сучасного програмного забезпечення, що призначене для опрацювання природомовних текстів, не дає змоги зробити його повністю автоматичним. Нормування тексту доцільно виконувати в середовищі редактора MS Word, як додатковий засіб створення необхідного програмного інструментарію зручно використовувати середовище Python. Ці ідеї перевірено на кафедрі прикладної лінгвістики Національного університету "Львівська політехніка" при готуванні матеріалів для корпусу творів письменників Наддністрянської України.

Список літератури

1. Ellis N. C. (2012). Formulaic language and second language acquisition. *Zipfand the phrasal teddy bear'. Annual Review of Applied Linguistics, 32,* 17–44.

2. Friederike Müller & Birgit Waibel (n. d.) Corpus linguistics — an introduction. Retrieved January 15, 2020 from https://www.anglistik.uni-freiburg.de/seminar/abteilungen/sprachwissenschaft/ls_mair/corpus-linguistics.

3. Gries S. Th. (2013). *Statistics for Linguistics Using.* Berlin.

4. Gries Stefan Th. (2019). Some long overdue additions/corrections (to/of actually all sorts of corpus-linguistics measures). *International Journal of Corpus Linguistics, 24 (3),* 385–412.

5. Nancy Ide (2008). Preparation and Analysis of Linguistic Corpora. In S. Schreibman & R. Siemens & J. Unsworth (Eds.) *A Companion to Digital Humanities* (pp. 289-305). doi:10.1002/9780470999875.

6. Perez Paredes. (n. d.) All things corpus & applied linguistics Research methods: corpus linguistics. Retrieved January 15, 2020 from http://www.perezparedes.es/research-methods-corpus-linguistics/.

7. Unicode Standard Releases. (n. d.) Unicode – The World Standard for Text and Emoji. Retrieved January 15, 2020 from https://home.unicode.org.

8. Бобкова, Т. В. (2014). До визначення корпусної лінгвістики в сучасному мовознавстві. *Наукові записки Національного університету "Острозька академія", (45),* 3–6.

9. Ванівська, О. І. (2012). Основні підходи до аналізу мовних даних у корпусній лінгвістиці. *Наукові записки Національного університету "Острозька академія", 27,* 3–8.

10. ГРАК (n. d.) Генеральний регіонально анотований корпус української мови. Доступ 15/01/2020 http://uacorpus.org/

11. Данилюк, І. (2013). Корпус текстів для вивчення граматичної службовості. *Лінгвістичні студії, 26,* 224–229.

12. Дарчук, Н. (2010). Дослідницький корпус української мови: основні засади і перспективи. *Вісник Київського національного університету імені Тараса Шевченка, 21,* 45–49.

13. Загнітко, А. П. (2015). Встановлення функційної характерології та парадигмально-синтагмального вияву часток в експериментальному дослідницькому лінгвістичному корпусі службовості. In О. Левченко (Ed.) *Дані текстових корпусів у лінгвістичних дослідженнях* (pp. 46–64).

14. Загнітко, А. & Данилюк, І. (2013). Корпус текстів граматичної службовості. In *Прикладна лінгвістика та лінгвістичні технології* (pp. 102–112).

15. Кульчицький, І. М. (2015). Технологічні аспекти укладання корпусів текстів. In О. Левченко (Ed.) *Дані текстових корпусів у лінгвістичних дослідженнях* (pp. 29–45).

16. Кульчицький, І. (2016). Корпуси текстів як лінгвотехнологічне підґрунтя виявлення змін в українській мові. In А. Архангельська (Ed.) *XX–XXI століття: жанрово-стильові й лінгвістичні метаморфози в українській мові та літературі* (pp. 269–298).

17. Кульчицький І. М. (2014). Технічні аспекти опрацювання комп'ютером природномовної інформації. *Вісник Національного університету "Львівська політехніка", 783,* 344–353.

18. Друль Орест (2015). Поправлюваний Франко. Збруч. Отримано 16/01/2020 з https://zbruc.eu/node/35977

19. Русанівський В. М. & Тараненко О. О. & all. (2004). *Українська мова: Енциклопедія*. Видавництво "Українська енциклопедія ім. М. П. Бажана".

20. Український правопис 2019. (2019). Міністерство освіти і науки України. Отримано 15/01/2020 з https://mon.gov.ua/ua/osvita/zagalna-serednya-osvita/navchalni-programi/ukrayinskij-pravopis-2019

21. Широков В. А. & all (2005). *Корпусна лінгвістика*. Довіра.

References

1. Ellis N. C. 'Formulaic language and second language acquisition. Zipfand the phrasal teddy bear'. Annual Review of Applied Linguistics 32, 2012. 17–44.

2. Friederike Müller and Birgit Waibel, Corpus linguistics – an introduction, from https://www.anglistik.uni-freiburg.de/seminar/abteilungen/sprachwissenschaft/ls_mair/corpus-linguistics [FM].

3. Gries S. Th. Statistics for Linguistics Using R. 2nd edn. Berlin. De Gruyter Mouton, 2013. p. 179.

4. Gries Stefan Th. Some long overdue additions/corrections (to/of actually all sorts of corpus-linguistics measures). International Journal of Corpus Linguistics, Volume 24, Issue 3, Aug 2019, p. 385–412

5. Nancy Ide (2008). Preparation and Analysis of Linguistic Corpora. A Companion to Digital Humanities/Susan Schreibman, Ray Siemens, John Unsworth, John Wiley & Sons 640 p. [NI08].

6. Perez Paredes. All things corpus & applied linguistics Research methods: corpus linguistics, from http://www.perezparedes.es/research-methods-corpus-linguistics/

7. The Unicode Consortium, from http://www.unicode.org/ [UTF].

8. Bobkova, T. V (2014). Towards a definition of corpus linguistics in modern linguistics. Scientific Papers of Ostroh Academy National University, (45), 3–6.

9. Vanivska, O. I (2012). Basic approaches to the analysis of language data in corpus linguistics. Scientific Papers of Ostroh Academy National University, 27, 3–8.

10. GRAC (n. D.) General regionally annotated corpus of the Ukrainian language. Accessed 15/01/2020 http://uacorpus.org/

11. Danylyuk, I. (2013). A body of texts for the study of grammatical servitude. Linguistic Studies, 26, 224–229.

12. Darchuk, N. (2010). The research body of the Ukrainian language: basic principles and perspectives. Bulletin of Taras Shevchenko National University of Kyiv, 21, 45–49.

13. Zagnitko, A. P (2015). Establishment of Functional Characteristics and Paradigm-Syntagmal Particle Detection in the Experimental Research Linguistic Corps of Servitude. In O. Levchenko (Ed.) Data from text corpora in linguistic studies (pp. 46–64).

14. Zagnitko, A. & Danylyuk, I. (2013). A body of grammatical servitude texts. In Applied Linguistics and Linguistic Technologies (pp. 102–112).

15. Kulchytskyy, I. M. (2015). Technological aspects of text corpus laying. In O. Levchenko (Ed.) Text corpus data in linguistic research (pp. 29–45).

16. Kulchytskyi, I. (2016). Text Cases as a Linguistic and Technological Basis for Detecting Changes in the Ukrainian Language. In A. Arkhangelsk (Ed.) XX–XXI centuries: genre-style and linguistic metamorphoses in Ukrainian language and literature (pp. 269–298).

17. Kulchitsky I. M. (2014). Technical aspects of computer-generated natural language information. Bulletin of the National University of Lviv Polytechnic, 783, 344–353.

18. Drul Orestes (2015). Corrected by Franco. Collapsed. Retrieved 16/01/2020 from https://zbruc.eu/node/35977

19. Rusanovsky V. M & Taranenko OO & all. (2004). English language: Encyclopedia. Publishing House "Ukrainian Encyclopedia. MP Bazhan".

20. Ukrainian Spelling 2019. (2019). Ministry of Education and Science of Ukraine. Retrieved 15/01/2020 from https://mon.gov.ua/en/osvita/zagalna-serednya-osvita/navchalni-programi/ukrayinskij-pravopis-2019

21. Shirokov V. A & all (2005). Corpus linguistics. Trust.

TEXT NORMALIZATION DURING PRE-CORPUS PREPARATION: EXPERIENCE OF APPLICATION

Ihor Kulchytskyy

Lviv Polytechnic National University
bis.kim@gmail.com, ORCID 0000-0001-9550-9739

The article analyses the experience of normalization of texts before introduction into the corpus of literary works of Naddnistrian Ukraine. The creation of the corpus was started at the department of Applied Linguistics of Lviv Polytechnic National University. Normalization means a set of information procedures that make the texts suitable for insertion into the corpus: bringing all texts to one code table, checking them for punctuation correctness (sense-identical entities should be marked with one character), eliminating unnecessary characters (for example, blank paragraphs , several gaps in a row, etc.), unification of formatting tools and methods, and more. MS Word editor is offered as a standardization medium, and Python programming language is used to create additional programming tools. Text normalization process contains the following stages: normalization of coding, normalization of graphics, text proofreading, technical normalization of punctuation. Each stage characteristics are presented, problems that arise during their implementation are indicated, and ways to overcome them are suggested. The conclusions are drawn.

Key words: corpus of texts, normalization, code tables, text graphics, text correction, punctuation.