УДК 004.91

# ПЕРЕВІРКА ПРАВОПИСУ ВВЕДЕНОГО ТЕКСТУ НА ОСНОВІ МОДЕЛІ NOISY CHANNEL

**Кулибаба П.О., Якименко Д.О., Рідкокаша А.А.**
Черкаський державний технологічний університет

У роботі представлено алгоритм для перевірки орфографії в тексті оснований на моделі Noisy Channel. Проаналізовано роботу алгоритму. Описано найбільш відомі методи та алгоритми для нечіткого пошуку слів, які можна застосувати до моделі Noisy Channel. Запропоновано варіанти удосконалення та спрощення роботи алгоритму. Запропоновано варіанти поліпшення програмної реалізації алгоритму для перевірки орфографії основаного на моделі Noisy Channel.
**Ключові слова:** правопис, орфографія, виправлення помилок, noisy channel, spelling correction, fuzzy string search.

**Постановка проблеми.** У наш час проблема орфографічної грамотності набуває все більшої актуальності. Адже без знання орфографічних правил неможливо написати лист, який-небудь документ, статтю або просто записку близьким. Для перевірки орфографії все більше використовують комп'ютери. Існує безліч онлайн сервісів та комп'ютерних програм для перевірки орфографії. Тому задача побудови простого і швидкого алгоритму пошуку помилок у тексті є актуальною.

Оцінка частоти орфографічних помилок у текстах, що друкуються людиною, варіюються від 1–2% (ретельний перезапис вже друкованого тексту) до 10–15% (для веб-запитів) [1].

Виправлення орфографічних помилок є невід'ємною частиною написання в сучасному світі, незалежно від того, чи це написання є частиною текстових повідомлень на телефоні, надсиланням електронної пошти, написанням довших документів або пошуку інформації в Інтернеті.

Сучасні орфографічні коректори не є досконалими, але вони практично повсюдно використовуються будь-яким програмним забезпеченням, яке залежить від введення даних з клавіатури [1].

**Аналіз останніх досліджень і публікацій.** Модель Noisy Channel (Shannon 1948) була успішно застосована до широкого кола проблем, в тому числі до корекції орфографії. Ця модель складається з двох компонентів: вихідна модель (source model) і модель каналу (channel model). Люди присвятили багато зусиль на поліпшення обох компонентів, що призвело до поліпшення загальної точності системи. Тим неменше, відносно мало досліджень пішло на поліпшення моделі каналу для корекції орфографії [2].

Великий внесок у теорію і практику коригування помилок в текстах внесли L. Philips, E. Brill, O. Kolak, E. Mays, D. Fossati, K. Kukich, M. Reynaert і інші зарубіжні вчені [3].

Натомість у вітчизняній лінгвістиці об'єктом аналізу є обробка української мови (Н. П. Дарчук, Н. Г. Чейлитко). Разом з тим, напрацювання в царині граматики англійської мови, зокрема генеративної граматики (І. Р. Буніятова, М. В. Полховська, І. Є. Снісаренко) [4].

**Виділення невирішених раніше частин загальної проблеми.** Перевірка граматики тексту є однією з найбільш важливих задач обробки природної мови (ОПМ). Перевірка орфографії та простих граматичних явищ в текстових процесорах вже стала звичною. Однак перевірка більш складних мовних явищ досі представляє проблему. Так, граматична перевірка, що здійснюється за допомогою пакетів програмного забезпечення, таких як Microsoft Office, може вирішувати лише обмежене коло граматичних помилок, наприклад: узгодження і керування. Але вони далекі від того, щоб знайти всі помилки.

Очевидно, у природних мовах помилки при суб'єктно-предикатному узгодженні належать до числа найбільш грубих. Окрім того в кожній мові є «власні» найбільш поширені помилки. На сьогодні автоматична перевірка граматики не може усунути значної кількості таких помилок. Складністю є той факт, що велика кількість граматичних правил включають в себе не лише морфо-синтаксичні, але і семантичні та прагматичні аспекти, що робить їх формалізацію для автоматичної перевірки проблематичною.

Так, відомий приклад (речення: «One morning I shot an elephant in my pajamas», переклад: «Одного ранку я вистрілив у слона в моїй піжамі») при автоматичній перевірці не міститиме помилок.

Тому на сьогодні повний парсинг при граматичній перевірці використовується рідко, натомість використовується підхід, що передбачає врахування найбільш поширених помилок або полегшеної модифікації повних синтаксичних формалізмів [4].

**Мета статті.** Проаналізувати роботу алгоритму для перевірки орфографії в тексті основаного на моделі Noisy Channel. Описати найбільш відомі алгоритми для нечіткого пошуку слів, які можна застосувати до моделі Noisy Channel. Запропонувати варіанти удосконалення та спрощення алгоритму для перевірки орфографії в тексті основаного на моделі Noisy Channel.

**Виклад основного матеріалу.** Модель Noisy Channel була застосована до задачі корекції орфографії приблизно в той же час дослідниками з AT&T Bell Laboratories (Браян Керніган і ін., 1990, Черч і Гейл, 1991) та IBM Watson Research (Мейс і ін., 1991) [1].

У моделі Noisy Channel ми уявляємо, що поверхня, яку ми бачимо, насправді є «спотвореною» формою оригінального слова, пропущеного через шумний канал (Noisy Channel). Декодер передає кожну гіпотезу через модель цього каналу і вибирає слово, яке найкраще відповідає шумному слову поверхні.

Суть моделі Noisy Channel (див. рис. 1) полягає в тому, щоб трактувати неправильно написане слово так, наче слово, що правильно написане, було «спотворене» пропуском через шумний канал зв'язку. Цей канал представляє «шум» у вигляді замін або інших змін літер, що ускладнює розпізнавання «справжнього» слова. З огляду на цю модель, ми потім знаходимо справжнє слово, пропускаючи кожне слово мови через нашу модель Noisy Channel і бачимо, яке з них наближається до неправильно написаного слова. Ця модель Noisy Channel є свого роду баєсівським виведенням (наївним баєсовим класифікатором) [1].

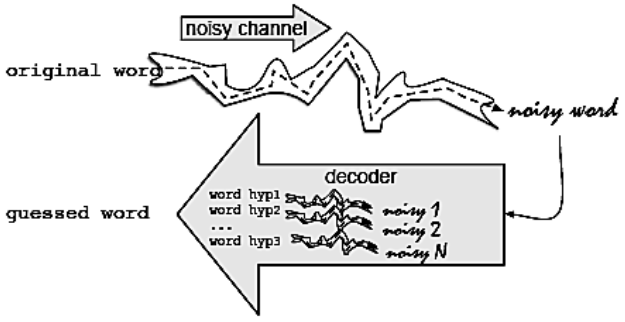


**Рис. 1. Модель Noisy Channel**

*Джерело: [1]*

Ми бачимо спостереження $x$ (слово з помилками), і наша робота полягає в тому, щоб знайти слово $w$, яке спричинило це неправильно написане слово. З усіх можливих слів у словнику $V$ ми хочемо знайти слово $w$ таке, щоб $P(w|x)$ було найвищим. Ми використовуємо позначення $\hat{}$, щоб позначити «нашу оцінку правильного слова».

$$\hat{w} = \arg\max_{w \in V} P(w \mid x). \quad (1)$$

Функція $\text{argmax}_x f(x)$ означає «$x$ такий, що $f(x)$ досягає максимуму». Формула 1 означає, що з усіх слів в словнику, ми хочемо конкретне слово, яке максимізує праву частину $P(w|x)$. Суть класифікації Баєса – це використання правила Баєса для перетворення формули 1 в набір інших ймовірностей. Правило Баєса представлено у формулі 2. Це дає нам можливість розбити будь-яку умовну ймовірність $P(a|b)$ на три інші ймовірності:

$$P(a|b) = \frac{P(b \mid a) \cdot P(a)}{P(b)} \quad (2)$$

Тоді ми можемо застосувати формулу 2 до формули 1 для отримання формули 3:

$$\hat{w} = \arg\max_{w \in V} \frac{P(x \mid w) \cdot P(w)}{P(x)}. \quad (3)$$

Ми можемо зручно спростити формулу 3, відкинувши знаменник $P(x)$. Оскільки ми вибираємо з усіх слів потенційно коректне слово, ми будемо обчислювати для кожного слова $\frac{P(x \mid w) \cdot P(w)}{P(x)}$. Але $P(x)$ не змінюється для кожного слова, ми завжди питаємо про найбільш ймовірне слово для тієї ж поширеної помилки $x$, яка повинна мати таку ж ймовірність $P(x)$. Таким чином, ми можемо вибрати слово, яке максимізується цією більш простою формулою:

$$\hat{w} = \arg\max_{w \in V} P(x \mid w) \cdot P(w) \quad (4)$$

Отже, модель Noisy Channel говорить, що у нас є якесь справжнє основне слово $w$, і у нас є Noisy Channel (шумний канал), який модифікує слово в якусь можливу орфографічну поверхневу форму. Ймовірність (likelihood) або модель каналу (channel model) Noisy Channel, що створює яку-небудь конкретну послідовність спостережень $x$, позначається, як $P(x|w)$. Апріорна ймовірність (prior probability) прихованого слова позначається, як $P(w)$. Ми можемо обчислити найбільш вірогідне слово $w$, враховуючи те, що ми спостерігали деякі спостережувані помилки $x$, шляхом множення апріорної ймовірності $P(w)$ та ймовірності $P(x|w)$ і вибору слова, для якого цей продукт найбільший.

Ми застосовуємо підхід Noisy Channel для виправлення несловесних (non-word) помилок правопису, приймаючи будь-яке слово не з нашого словника, створюючи список кандидатів (candidate words), класифікуючи їх відповідно до формули 4, і вибираємо найбільш ранговий. Ми можемо змінити формулу 4, щоб послатися на цей список кандидатів, а не на повний словниковий запас $V$ наступним чином:

$$\hat{w} = \underset{w \in C}{\text{argmax}} \overbrace{P(x|w)}^{\text{channel model}} \cdot \overbrace{P(w)}^{\text{prior}} \quad (5)$$

Алгоритм Noisy Channel показаний на рис. 2.

```
function NOISY CHANNEL SPELLING(word x, dict D, lm, editprob) returns correction
  if x ∉ D
    candidates, edits ← All strings at edit distance 1 from x that are ∈ D, and their edit
    for each c, e in candidates, edits
      channel ← editprob(e)
      prior ← lm(x)
      score[c] = log channel + log prior
    return argmax_c score[c]
```

**Рис. 2. Алгоритм Noisy Channel псевдомовою**

*Джерело: [1]*

Щоб побачити деталі розрахунку ймовірності та апріорної ймовірності (так же, мовної моделі (language model)), давайте розберемо приклад, застосувавши алгоритм до неправильно написаного слова *acress*. Перший етап алгоритму передбачає коригування кандидатів, знаходячи слова, які мають аналогічне написання для вхідного слова. Аналіз даних про помилку правопису показує, що більшість орфографічних помилок складаються з зміни однієї літери і тому ми часто робимо спрощений погляд на те, що ці кандидати мають відстань редагування 1 від слова помилки. Щоб знайти цей список кандидатів ми будемо використовувати мінімальний алгоритм редагування відстані, але розширимо його так, що на додаток до вставки, видалення і заміни, ми додамо четвертий тип редагування, транспозицію, в якому дві букви міняються місцями. Цей алгоритм називається відстанню Дамерау-Левенштейна. Всі такі поодинокі перетворення до acress дають список кандидатів зображених на рис. 3.

Після того, як у нас є набір кандидатів, оцінимо кожен, використовуючи формулу 5, щоб обчислити апріорну ймовірність та модель каналу. Апріорною ймовірністю кожної корекції $P(w)$ є ймовірність мовної моделі слова $w$ у контексті, яку можна обчислити за допомогою будь-якої

мовної моделі від unigram до trigram або 4-gram. Для цього прикладу давайте почнемо з наступного рисунку, використавши мовну модель unigram. Ми обчислили мовну модель зі слів у корпусі сучасної англійської мови (Corpus of Contemporary American English (COCA)).

| Помилка | Корекція | Вірна літера | Помилкова літера | Позиція | Тип операції |
|---|---|---|---|---|---|
| acress | actress | t | - | 2 | видалення |
| acress | cress | - | a | 0 | вставка |
| acress | caress | ca | ac | 0 | транспозиція |
| acress | access | c | r | 2 | заміна |
| acress | across | o | e | 3 | заміна |
| acress | acres | - | s | 5 | вставка |
| acress | acres | - | s | 4 | вставка |

**Рис. 3. Список кандидатів до помилково написаного слова acress**

*Джерело: розроблено авторами за даними [1]*

| w | count(w) | p(w) |
|---|---|---|
| actress | 9,321 | 0.0000231 |
| cress | 220 | 0.000000544 |
| caress | 686 | 0.00000170 |
| access | 37,038 | 0.0000916 |
| across | 120,844 | 0.000299 |
| acres | 12,874 | 0.0000318 |

**Рис. 4. Апріорна ймовірність кандидатів у мовній моделі unigram**

*Джерело: розроблено авторами за даними [1]*

Тепер потрібно обчислити ймовірність $P(x|w)$, яку також називають моделлю каналу (channel model) або моделлю помилок (error model). На щастя, ми можемо отримати досить розумну оцінку $P(x|w)$, просто, дивлячись на локальний контекст: тотожність самого правильного листа, орфографічні помилки та сусідні літери. Наприклад, букви *m* та *n* часто замінюються один на одного, це частково є фактом їхньої ідентичності (ці дві букви вимовляються однаково, вони розташовані поруч один з одним на клавіатурі), а частково і фактом контексту (тому, що вони вимовляються аналогічно і вони зустрічаються у подібних контекстах). Проста модель може оцінити, наприклад, $p(acress|across)$, просто використовуючи кількість разів, коли буква *e* була замінена буквою *o* у деякому великому корпусі помилок. Для того, щоб обчислити ймовірність кожного редагування таким чином, нам потрібна матриця помилок, яка містить кількість помилок. Взагалі, матриця помилок вказує кількість разів, коли одна річ була заплутана з іншою. За Керніганом, Черчом і Гейлом ми будемо використовувати чотири суперечливі матриці [5].

*del [x;y]: count(xy набрано як x)*
*ins [x;y]: count(x набрано як xy)*
*sub [x;y]: count(x набрано як y)*
*trans [x;y]: count(xy набрано як yx)*

Ці суперечливі матриці можна отримати зі списків орфографічних помилок, таких як:

***additional:*** *addional, additonal*
***environments:*** *enviornments, enviorments, enviroments*
***preceded:*** *preceeded*
...

Є списки, доступні в Вікіпедії, є списки Роджера Міттона [7] та Пітера Норвіга [8]. Норвіг також дає підрахунки для кожного редагування одного символу, який можна використовувати для прямого створення вірогідності моделі помилок.

Альтернативний підхід, використаний Керніганом та співавторами – це обчислення матриці ітераційно, використовуючи саме сам алгоритм виправлення правопису. Ітеративний алгоритм спочатку ініціалізує матриці з однаковими значеннями, таким чином, будь-який символ буде однаково вилучений, однаково може бути замінений будь-яким іншим символом і т. д. Потім алгоритм виправлення помилки орфографії виконується набором орфографічних помилок. Враховуючи набір помилок у поєднанні з їх передбачуваними виправленнями, можуть бути перекомпільовані матриці плутанин, алгоритм орфографії знову виконується, тощо. Цей ітеративний алгоритм є прикладом алгоритму EM (Демпстер і ін., 1977).

Після того, як у нас є суперечливі матриці, ми можемо оцінити $P(x|w)$ наступним чином (де $w_i$ – $i$-й символ правильного слова $w$, а $x_i$ – $i$-й символ помилки $x$):

$$P(x|w) = \begin{cases} \frac{del[x_{i-1}, w_i]}{count[x_{i-1} \cdot w_i]}, if\ deletion \\ \frac{ins[x_{i-1}, w_i]}{count[w_{i-1}]}, if\ insertion \\ \frac{sub[x_i, w_i]}{count[w_i]}, if\ substitution \\ \frac{trans[w_i, w_{i+1}]}{count[w_i \cdot w_{i+1}]}, if\ transposition \end{cases} \quad (6)$$

Використані дані від Кернігана та ін. призводять до ймовірностей моделі помилок для acress, показаної на рисунку 5 [6].

| Кандидати | Вірна літера | Помилкова літера | x\|w | P(x\|w) |
|---|---|---|---|---|
| actress | t | - | c\|ct | 0.000117 |
| cress | - | a | a\|# | 0.00000144 |
| caress | ca | ac | ac\|ca | 0.00000164 |
| access | c | r | r\|c | 0.000000209 |
| across | o | e | e\|o | 0.0000093 |
| acres | - | s | es\|e | 0.0000321 |
| acres | - | s | ss\|s | 0.0000342 |

**Рис. 5. Модель каналу для acress**

*Джерело: розроблено авторами за даними [1, 5]*

Ймовірності взяті з матриць заміщень *del []*, *ins []*, *sub []* і *trans []*, як показано Керніганом і ін. [5].

Рисунок 6 показує кінцеві вірогідності кожного з потенційних виправлень, апріорну ймовірність помножену на ймовірність (обчислену за допомогою формули 6 та матриці з плутанинами). У заключному стовпчику показані значення, помножені на $10^9$ для зручності читання.

Розрахунки на рис. 6 показують, що наша реалізація моделі Noisy Channel вибирає across, як найкращу корекцію, а *actress* – як друге ймовірне слово.

Намір письменника стає зрозумілий з контексту: *«stellar and versatile acress whose combination of beauty and glamour has defined he…»* («видатна і різнобічна acress, чия комбінація привабливості і гламуру визначила її»). Навколишні слова дають зрозуміти, що *actress*, а не *across* було наміченим слово.

З цієї причини важливо використовувати більші мовні моделі, ніж юніграми (unigrams). Наприклад, якщо ми використовуємо корпус сучасної

англійської мови (Corpus of Contemporary American English) для розрахунку ймовірностей bigram для слів *actress* та *across* через їх контекст, використовуючи згладжування, ми отримуємо такі ймовірності:

$P(actress|versatile) = 0.000021$

$P(across|versatile) = 0.000021$

$P(whose|actress) = 0.0010$

$P(whose|across) = 0.000006$

Перемножуючи ці результати, ми оцінюємо мовну модель для двох кандидатів у контексті:

$P(«versatile\ actress\ whose») = 0.000021 \cdot 0.0010 = 210 \cdot 10^{-10}$

$P(«versatile\ across\ whose») = 0.000021 \cdot 0.000006 = 1 \cdot 10^{-10}$

Поєднуючи мовну модель з моделлю помилок на рисунку 6, модель bigram шумного каналу (Noisy Channel) вибирає правильне слово *actress*.

| Кандидати | Вірна літера | Помилкова літера | x\|w | P(x\|w) | P(w) | P(x\|w) * P(w) * $10^9$ |
|---|---|---|---|---|---|---|
| actress | t | - | c\|ct | 0,000117 | 0,0000231 | 2,7 |
| cress | - | a | a\|# | 0,00000144 | 0,000000544 | 0,00078 |
| caress | ca | ac | ac\|ca | 0,00000164 | 0,00000170 | 0,0028 |
| access | c | r | r\|c | 0.000000209 | 0,0000916 | 0,019 |
| across | o | e | e\|o | 0,0000093 | 0,000299 | 2,8 |
| acres | - | s | es\|e | 0,0000321 | 0,0000318 | 1.0 |
| acres | - | s | ss\|s | 0,0000342 | 0,0000318 | 1.0 |

**Рис. 6. Обчислення рейтингу для кожної корекції кандидата**

*Джерело: розроблено авторами за даними [1, 6]*

Оцінювання алгоритмів корекції орфографії зазвичай проводиться шляхом викладання набору тренувань, розробки та тестування списків помилок, таких як на сайтах Міттона та Норвіга [7, 8].

Крім відстані Дамерау-Левенштейна, яка була використана в реалізації моделі Noisy Channel існують і інші алгоритми та методи для обчислення міри подібності двох слів. Найбільш відомі: відстань Левенштейна, метод N-грамм, Soundex, Metaphone.

Відстань Левенштейна: мінімальне число операцій вставки, видалення і заміни символів, яке необхідно провести для того, щоб перетворити один рядок в інший. Метод розроблений у 1965 році радянським математиком Левенштейном В. Й. і названий його іменем. У реалізації моделі Noisy Channel була використана розширена версія відстані Левенштейна, яка називається відстань Дамерау-Левенштейна. В якій на додачу до операцій вставки, видалення і заміни додається операція перестановки двох сусідніх символів.

N-грами на рівні символів: символьна n-грама являє собою послідовність з n символів. Відношення кількості n-грам, які містяться в обох словах, і унікальної кількості всіх n-грам. N-грами можуть бути використані в якості запобіжного визначення схожості слів.

Soundex: фонетичний алгоритм для індексації назв за звучанням, в англійській мові. Він встановлює однакове представлення омофонів, що спрощує їхній пошук, незважаючи на неточності в написанні. Алгоритм переважно кодує приголосні звуки, голосні опускаються, крім першої букви. Soundex – найвідоміший зі всіх фонетичних алгоритмів (частково, через те, що є стандартною особливістю популярних СКБД, таких як DB2, PostgreSQL, MySQL, Ingres, MS SQL і Oracle).

Metaphone: фонетичний алгоритм, опублікований 1990 року для індексації слів в англійській вимові. Алгоритм із змінною довжиною ключа, на відміну від Soundex зі фіксованою довжиною ключів. Metaphone був розроблений Лоуренс Філіпсом, як відповідь на недоліки в алгоритмі Soundex. Він використовує більший набір правил англійської вимови. Metaphone доступний, як вбудований оператор у низці систем, зокрема, у останніх версіях PHP. Пізніше створено нову версію алгоритму, Double Metaphone, яка виробляє точніші результати, ніж початковий алгоритм. Всупереч оригінальному алгоритму, який обмежений лише англійською мовою, ця версія враховує орфографічні особливості ряду інших мов. У 2009 році компанія Lawrence Philips випустила третю версію, названу Metaphone 3, яка досягає точності приблизно 99% для англійських слів.

При реалізації моделі Noisy Channel можна використати перечисленні вище алгоритми для знаходження міри подібності двох слів. А як же бути зі швидкістю роботи алгоритму, адже нам доведеться проходити по всій базі слів, а це трудомісткий процес, а так же обчислювати для всіх слів ймовірності помилки? Тут приходить на допомогу наступний статистичний факт: 80% всіх друкованих помилок знаходяться в межах 1 операції редагування, тобто відстані Дамерау-Левенштейна рівним одиниці, майже всі друковані помилки знаходяться в межах 2 операцій редагування.

Збільшення швидкості вибірки досягається через вибірку слів по його ключу (хешу). Пошук здійснюється шляхом обчислення хеша помилкового слова і пошуку слів у словнику з таким же значенням хеша. Для цього можна використати методи Soundex та Metaphone.

Якщо нам потрібні тільки слова не більші ніж в t операцій редагування від поточного слова, то їх довжина відрізняється від поточного не більше ніж на t.

Можна створювати хеш-таблицю, в якій ключами є довжини слів, а значеннями – множини слів цієї довжини, це дозволяє значно скорочувати простір пошуку.

І нарешті можна поліпшити саму програмну реалізацію алгоритму, можна реалізувати її на компільованій мові програмування, а не на інтерпретованій. Можна використовувати інший контейнер для рядків, спеціалізований під зберігання і порівняння рядків. Можна кешувати результати обчислень, так щоб не повторювати їх кілька разів.

**Висновки і пропозиції.** Було проаналізовано роботу алгоритму для перевірки орфографії в тексті основаного на моделі Noisy Channel, описано найбільш відомі алгоритми для нечіткого пошуку слів, які можна застосувати до моделі Noisy Channel. Запропоновано варіанти удосконалення та спрощення роботи алгоритму для перевірки орфографії основаного на моделі Noisy Channel.

**Список літератури:**

1. Daniel Jurafsky, James H. Martin. Speech and Language Processing, 2nd Edition / Publisher: Prentice Hall. – 2008. – 1024 p.
2. Brill E. An Improved Error Model for Noisy Channel Spelling Correction / E. Brill, R. Moore // Proceedings of the 38th Annual Meeting on Association for Computational Linguistics (ACL '00). – 2000. – Pp. 286–293.
3. Смирнов С. В. Технология и система автоматической корректировки результатов при распознавании архивных документов: диссертация кандидата технических наук: 05.13.11 / Смирнов С. В. [Место защиты: Федеральное государственное бюджетное учреждение науки Санкт-Петербургский институт информатики и автоматизации Российской академии наук, www.spiiras.nw.ru/dissovet]. – Санкт-Петербург, 2015. – 130 с.
4. Гирин О. В. Автоматичний синтаксичний аналіз англійського речення: застосування та перспективи. Вісник Житомирського державного університету імені Івана Франка (1(85)). – 2017. – Pp. 26–30.
5. Kernighan M. D., Church K. W. and Gale W. A. (1990). A spelling correction program base on a noisy channel model. In COLING-90, Helsinki, Vol. II, Pp. 205–211.
6. Кулибаба П. О. Перевірка правопису введеного тексту на основі моделі Noisy Channel [Текст] / П. О. Кулибаба, Д. О. Якименко // «Інноваційний розвиток науки нового тисячоліття» (м. Ужгород, 21–22 квітня 2017 р.). – Херсон: Видавничий дім «Гельветика», 2017.
7. Corpora of misspellings for download. – [Електронний ресурс]. – Режим доступу: http://www.dcs.bbk.ac.uk/~ROGER/corpora.html
8. Natural Language Corpus Data: Beautiful Data. – [Електронний ресурс]. – Режим доступу: http://norvig.com/ngrams/

**Кулибаба П.А., Якименко Д.О., Ридкокаша А.А.**
Черкасский государственный технологический университет

## ПРОВЕРКА ПРАВОПИСАНИЯ ВВЕДЕННОГО ТЕКСТА НА ОСНОВЕ МОДЕЛИ NOISY CHANNEL

**Аннотация**
В работе представлен алгоритм для проверки орфографии в тексте основан на модели Noisy Channel. Проанализирована работа алгоритма. Описаны наиболее известные методы и алгоритмы для нечеткого поиска слов, которые можно применить к модели Noisy Channel. Предложены варианты усовершенствования и упрощения работы алгоритма. Предложены варианты улучшения программной реализации алгоритма для проверки орфографии основанного на модели Noisy Channel.
**Ключевые слова:** правописание, орфография, исправления ошибок, noisy channel, spelling correction, fuzzy string search.

**Kulybaba P.O., Yakimenko D.O., Ridkokasha A.A.**
Cherkasy State Technological University

## SPELL CHECKING OF THE ENTERED TEXT BASED ON THE MODEL NOISY CHANNEL

**Summary**
The paper presents an algorithm for checking spelling in the text based on the Noisy Channel model. The work of the algorithm is analyzed. The most well-known methods and algorithms for fuzzy word search that can be applied to the Noisy Channel model are described. The variants of improvement and simplification of work of the algorithm are offered. There are offered variants of improvement of program realization of the algorithm for checking the spelling based on the model Noisy Channel.
**Keywords:** spelling, orthography, error correction, noisy channel, spelling correction, fuzzy string search.