**Uniwersytet Warszawski**
Wydział Filozofii i Socjologii

**Paweł Lonca**
Nr albumu: 359794

# Analiza empiryczna metod automatycznego wykrywania i poprawiania błędów pisowni w tekstach polskojęzycznych w oparciu o kontekst

**Praca licencjacka**
**na kierunku KOGNITYWISTYKA**
**w zakresie przetwarzania języka naturalnego**

Praca wykonana pod kierunkiem
**dr Aliny Wróblewskiej**
Instytut Podstaw Informatyki
Polskiej Akademii Nauk

Listopad 2019

## Oświadczenie kierującego pracą

Oświadczam, że niniejsza praca została przygotowana pod moim kierunkiem i stwierdzam, że spełnia ona warunki do przedstawienia jej w postępowaniu o nadanie tytułu zawodowego.

Data Podpis kierującego pracą

## Oświadczenie autora (autorów) pracy

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną zamieszczoną w systemie (USOS, APD, UW)

Data Podpis autora pracy

## Streszczenie

W zależności od szacunków w przypadku języka angielskiego w wyniku wpisywania słów na klawiaturze komputerowej 1-15% z nich zawiera błędy pisowni [33]. Wychodząc od podziału na błędy w postaci nieistniejących słów (*non-word errors*) oraz istniejących słów (*real-word errors*) [57], implementuję model kanału z szumem (ang. *noisy channel model*) [59] dla języka polskiego, składający się z modelu językowego oraz modelu błędu. Pierwszy oparty jest o n-gramy stworzone na podstawie korpusu NKJP [42], które reprezentują zliczenia wystąpień poszczególnych par (2-gramy), trójek słów (3-gramy) itd., dając informację o najczęściej występujących zestawieniach wyrazów w tekście. Bazując na obserwacji, że dla języka angielskiego nawet połowa błędów pisowni typu *real-word* może zostać poprawiona za pomocą odwołania do kontekstu [1], implementuję model językowy w oparciu o hipotezę dystrybucyjną mówiącą, że znaczenie słowa zależy od kontekstu, w którym to słowo się znajduje [5]. Skuteczność modelu testuję na korpusie błędów języka polskiego powstałego na bazie analizy historii korekt artykułów Wikipedii [23]. Omawiając proponowany model zajmuję się problemem stosowania metod automatycznego wykrywania i poprawiania błędów pisowni za pomocą metod stworzonych dla języka angielskiego na gruncie języka polskiego, który wyróżnia się m.in. mnogością form morfologicznych.



## Abstract

According to different estimates between 1 and 15% of English words are typed with some kind of a spelling mistake [33]. With spelling errors being classified as either a *real-word error* (a spelling mistake that happens to be another existent word) or a *non-word error* (a spelling mistake that is a meaningless sequence of characters) as a starting point [57], I implement a *noisy channel model* [59] that is comprised of a language model and an error model. The former is based on n-grams that are generated based on NKJP corpus [42]. N-grams represent the frequency of certain pairs, triplets etc. of words and indicate what the most frequent words in corpora are. 50% of *real-word errors* in English can be corrected using the context which they occur in [1]. With that in mind, the language model is augmented with distributional hypothesis stating that the meaning of a word can be deduced from context that it occurs in [5]. The accuracy of the *noisy channel model* is evaluated on error corpora of Polish Wikipedia's revision history [23]. Automatic detection and correction techniques created specifically for errors in English are subject of discussion. I analyse their potential usage in Polish texts which, in contrast to those in English, are characterised by variety of the morphological forms.

## Słowa kluczowe

przetwarzanie języka naturalnego, n-gram, Bayes, model kanału z szumem, analiza morfologiczna, analiza składniowa, hipoteza dystrybucyjna

## Keywords

natural language processing, n-gram, Bayes, noisy channel model, morphological analysis, parsing, distributional hypothesis

## Empirical analysis of methods for automatic detection and correction of spelling errors in Polish-language texts based on context

## Dziedzina pracy

14.4 Kognitywistyka

# Spis treści

# Zawartość pracy

Poniższa praca składa się z sześciu rozdziałów. Rozdział 1 wprowadza czytelnika w tematykę przetwarzania języka naturalnego, a także przybliża rys historyczny metod korekty tekstu. Przedstawiona zostaje również hipoteza, wokół której skupiona jest dalsza część pracy.

W rozdziale 2, bazując na literaturze językoznawczej, definiuje się *błąd językowy* oraz przedstawia hierarchię błędów typowych dla języka polskiego. W nawiązaniu do osiągnięć psycholingwistyki, proponowany jest alternatywny podział błędów językowych.

Rozdział 3 wprowadza pojęcie *korpusu językowego*, które stanowi podstawę do zdefiniowania *korpusu błędów*. W rozdziale opisany jest sposób tworzenia egzemplarza tego drugiego oraz jego analiza w przypadku języka polskiego.

Rozdział 4 skupia się na korekcie tekstu w oparciu o metody słownikowe. Ponadto zawarte są w nim rozważania dotyczące transformacji ciągów znaków.

Rozdział 5 opisuje kontekstowe metody korekty tekstu bazujące m.in. na *n-gramowym modelu języka*, który jest w tym rozdziale definiowany. Poza tym rozdział wprowadza model kanału z szumem.

Rozdział 6, poświęcony implementacji modelu kanału z szumem, zawiera opis kroków potrzebnych od wczytania korpusu błędów poprzez oszacowanie modelu, aż wreszcie ewaluację tegoż na rzeczywistych danych językowych.

# Rozdział 1

# Wstęp

Przetwarzanie języka naturalnego (ang. *natural language processing*) to interdyscyplinarny obszar badań, korzystający z odkryć z dziedziny lingwistyki oraz informatyki, a skupiający się na rozpoznawaniu mowy, zrozumieniu i analizie języków używanych przez ludzi, a także generowaniu koherentnych próbek tychże języków. Za początek badań nad przetwarzaniem języka naturalnego uważa się rok 1950, kiedy Alan Turing w jednym ze swoich artykułów przestawił pomysł testu (później zwanego testem Turinga), w którym ogólnie rozumiany komputer miałby przejawiać inteligentne (ludzkie) zachowanie tak, aby sprawić, żeby rozmówca myślał, że rozmawia z drugim człowiekiem [2]. Jednym z aspektów tego testu jest nieodzowna konieczność posługiwania się językiem naturalnym, a więc odbioru, zrozumienia i generowania w nim komunikatów. Na przełomie lat 80. i 90. XX wieku nastąpił rozwój w obszarze przetwarzania języka naturalnego dzięki zastosowaniu metod uczenia maszynowego (ang. *machine learning*). Polega ono na zachodzeniu zmian w adaptujących się systemach, co z kolei pozwala tym systemom na bardziej efektywne rozwiązywanie takich samych lub podobnych zadań [3]. W zakresie używania języka naturalnego przez maszyny zaczęto posługiwać się modelami statystycznymi z szacowanymi parametrami.

Od czasu wynalezienia systemów komunikacji opartych na piśmie niedoskonała natura ludzka sprawiała, że w korespondencji, książkach, artykułach i innych tekstach pisanych pojawiały się różnego rodzaju błędy. Nie inaczej jest w erze cyfrowej, kiedy ludzie komunikują się za pomocą poczty elektronicznej oraz komunikatorów internetowych, do których wpisują wiadomości przy użyciu klawiatury komputerowej bądź też cyfrowego panelu swojego telefonu komórkowego.

Błędy pisowni w oficjalnych tekstach (książkach, artykułach, ustawach) mogą stanowić oznakę braku profesjonalizmu, dlatego w użyciu znajdują się zarówno (pół)automatyczne, jak i w całości manualne narzędzia do korekty tekstu. Abstrahując od osób profesjonalnie zajmujących się sprawdzeniem poprawności pisanych tekstów, można zauważyć, że podejście półautomatyczne rodzi pewne problemy, które polegają na obciążeniu użytkownika (osoby wpisującej tekst) obowiązkiem przejścia po zaznaczonych przez edytor tekstowy słowach i ewentualnego poprawienia błędów. Przykładem może być pakiet biurowy WPS Writer[1], wskazujący użytkownikom potencjalnie błędne słowa, które użytkownik może albo poprawić (wybierając słowo z proponowanej listy) albo zostawić w niezmienionej postaci.

Patrząc z perspektywy osoby zajmującej się wykrywaniem i poprawianiem błędów w tekście, przynajmniej część z nich może zostać zauważona na podstawie niespójnego komunikatu, jaki przekazuje zdanie (czy też cały tekst), w którym zawarte jest błędne słowo. Błędnie użyte słowo może mieć niewłaściwe znaczenie, które dla użytkownika danego języka jest łatwe do

[1] `https://www.wps.com/office-free`

wychwycenia. W przypadku rozwiązania automatycznego zrozumienie znaczenia komunikatu może być trudne lub wręcz niemożliwe. W tradycji filozoficznej XX wieku, do której powstania przyczynili się Ludwig Wittgenstein [4], J.R. Firth [5] oraz Zellig Harris [6], o znaczeniu słowa decyduje jego użycie w języku, tzn. dwa słowa, które pojawiają się w podobnych kontekstach przeważnie cechują się podobnym znaczeniem (teza dystrybucyjna).

Proces wprowadzania poprawek do tekstu mógłby zostać przyśpieszony dzięki wprowadzeniu automatycznego podejmowania decyzji co do poprawek, którymi tekst powinien zostać opatrzony. Dla języka angielskiego opracowano narzędzie, które pozwala na udoskonalenie procesu korekty tekstu pisanego. Jedna z wersji modelu kanału z szumem pozwoliła na automatyczną korektę 97,5% błędów w tekście [7]. Celem poniższej pracy jest analiza metod automatycznego wykrywania i poprawiania błędów pisowni w języku polskim. Poniższa praca stanowi, zgodnie z wiedzą autora, pierwszą udostępnioną publicznie implementację modelu kanału z szumem dla języka polskiego, który dodatkowo został rozbudowany, w myśl tezy dystrybucyjnej o znaczeniu słów, o obsługę kontekstu.

# Rozdział 2

# Typologia błędów językowych

Szukając korzeni pojęcia błędu językowego w rozróżnieniu na preskryptywne i deskryptywne podejście do języka naturalnego, przytaczam kompleksowy podział błędów dla języka polskiego. W końcowej części rozdziału zwracam uwagę na alternatywne podejścia do klasyfikacji błędów językowych.

## 2.1. Błąd językowy

Błąd językowy można postrzegać w kategorii niecelowego i niezamierzonego odejścia od wcześniej zdefiniowanych norm i wzorców językowych [13, s. 83]. Pojęcie błędu językowego może być wyróżnione w oparciu o preskryptywne podejście do języka naturalnego lub koncepcję rozróżnienia na kompetencję i performancję językową.

### 2.1.1. Preskryptywizm a deskryptywizm

Język naturalny jest nieustannie ewoluującym bytem. W językoznawstwie istnieją dwa konkurujące ze sobą nurty, podchodzące do zagadnienia języka zgoła odmiennie.

Podejście preskryptywne wychodzi z założenia, że język naturalny można opisać za pomocą norm oraz wzorców, które następnie narzucane użytkownikom skutkują stworzeniem neutralnej, normatywnej wersji języka [8]. Aspekty językowe takie jak fleksja, semantyka, wymowa, składnia, frazeologia, ortografia czy interpunkcja [9] mogą być regulowane w celu ustalenia jednolitego kodu językowego, który jednocześnie pozwala przekazać efektywnie i bez dwuznaczności komunikat językowy na stosunkowo dużym obszarze geograficznym [10, s. 80–81]. Szczególne znaczenie normy językowe wykazują w kontekstach formalnych, takich jak edukacja, działalność wydawnicza czy prawodawstwo, gdyż podmioty posługujące się jednym standardowym językiem są w stanie zrozumieć komunikat przekazywany im przez nadawcę (tu: nauczyciela, wydawcę, prawodawcę) [11, s. 139].

Podejście deskryptywne zarzuca pogląd, że można narzucić użytkownikom języka pewne normy, wymuszając stosowanie pożądanych wzorców. Językoznawca deskryptywny pozostaje niejako neutralnym obserwatorem wydarzeń w sferze danego języka, a jego zadaniem jest jedynie stworzyć opis, który nie zawiera sądów wartościujących [12, s. 47–48].

### 2.1.2. Kompetencja a performancja językowa

Wraz z pojawieniem się koncepcji gramatyki generatywnej, w której Chomsky zwrócił uwagę na rozróżnienie na kompetencję i performancję językową [29, s. 4], zaczęto postrzegać błąd językowy w kategorii ograniczeń stanu psychologicznego nadawcy (autora wypowiedzi).

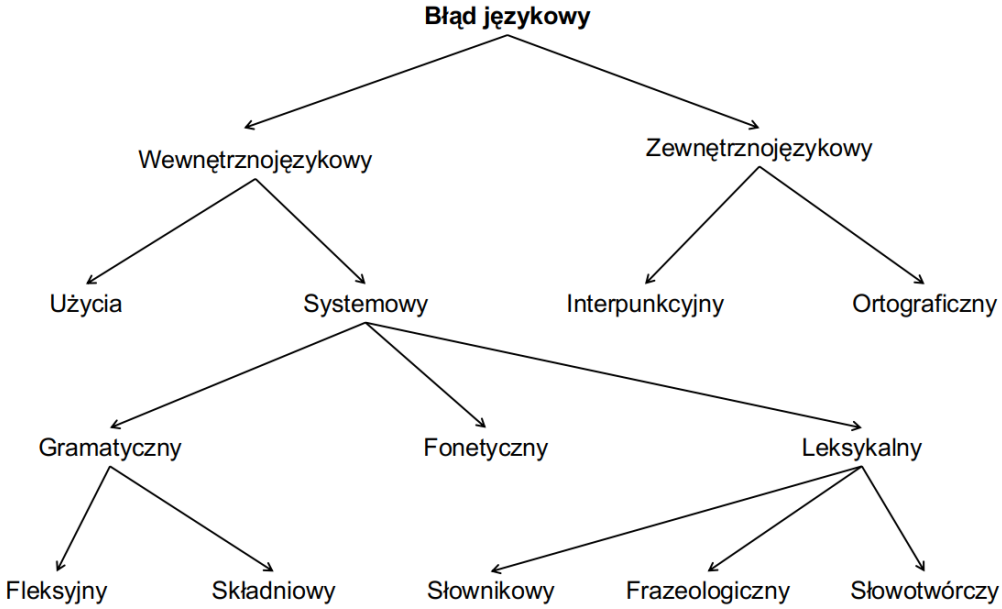
W ramach proponowanego programu błąd językowy należy do sfery performancji językowej, na jakość której wpływają nie tylko czynniki wewnętrzne (np. wykształcenie, środowisko ekonomiczno-społeczne), ale także czynniki zewnętrzne (np. kanał komunikacji).

Wychodząc od założenia, że każdy dorosły natywny użytkownik danego języka jest w stanie rozpoznać czy zakomunikowane mu zdanie (ewentualnie równoważnik zdania) do tego języka należy, teoria gramatyki generatywnej postuluje, że do zbioru produkcji językowych wspomnianych użytkowników należą przeważnie komunikaty poprawne, pozbawione błędów [31, s. 75]. Oczywiście dopuszczone są sytuacje, w których pojawiają się pomyłki, jednak nie mają one charakteru systematycznego [14, s. 2] i zaliczane są do błędów *sensu largo*. Miano błędów *sensu stricto* stosuje się do systematycznych uchybień językowych popełnianych przez nierodzimych użytkowników, dopiero uczących się danego języka [20, s. 700].

## 2.2. Klasyfikacja błędów językowych

Wyróżnia się dwie duże grupy błędów językowych: zewnętrznojęzykowe i wewnętrznojęzykowe [9, s. 1553]. Do pierwszej grupy należą błędy ortograficzne i interpunkcyjne. Mimo że są związane z językiem (np. stawianie znaków interpunkcyjnych ma podstawę w składni), to nie stanowią naruszenia reguł wewnętrznojęzykowych. Na błędy wewnętrznojęzykowe składają się błędy systemowe oraz błędy użycia (stylistyczne).

Rysunek 2.1: Podział błędów językowych. Opracowanie własne na podstawie [9, s. 1553-1555]




### 2.2.1. Błędy użycia (stylistyczne)

Błędy stylistyczne to w dużej mierze niewłaściwy dobór środków językowych w określonej wypowiedzi, niedostosowanie ich do charakteru i funkcji tej wypowiedzi:

- używanie elementów oficjalnych w wypowiedziach potocznych, np. **Dokonałem zakupu*

maszynki do golenia[1], **Skonsumowałeś* na obiad całą kartoflankę,

- używanie elementów potocznych w wypowiedzi o charakterze publicznym, np. W wyniku obserwacji ustalono, że **facet* przebywał w wymienionym obiekcie,
- stylizacja językowa niemająca uzasadnienia w treści i charakterze stylowym wypowiedzi, np. **Onegdaj* nasi zawodnicy odbyli tylko jeden trening,
- naruszenie zasad jasności, prostoty i zwięzłości stylu.

### 2.2.2. Błędy systemowe

- Błędy gramatyczne
  - Błędy fleksyjne przejawiają się w:
    - wyborze niewłaściwej postaci wyrazu, np. **wziąść* zamiast *wziąć*
    - wyborze niewłaściwego wzorca odmiany, np. *lubić* – **lubiał* zamiast *lubił*,
    - wyborze niewłaściwej postaci tematu fleksyjnego, np. **o gwiaździe* zamiast *o gwieździe*,
    - wyborze niewłaściwej końcówki fleksyjnej **imieniowi* zamiast *imieniu*,
    - nieodmienianiu wyrazu, który ma swój wzorzec deklinacyjny, np. **Jadę do Oksford* zamiast *Jadę do Oksfordu*,
    - odmianie wyrazu, któremu nie można przypisać wzorca odmiany, np. **wypić kubek kakaa* zamiast *wypić kubek kakao*.
  - Błędy składniowe polegają na niewłaściwym łączeniu form wyrazowych w jednostki tekstu:
    - błędy w zakresie związku zgody, np. **Rodzeństwo wyjechali za granicę* zamiast *Rodzeństwo wyjechało za granicę*,
    - błędy w zakresie związku rządu, np. **Podobny ojcu* zamiast *Podobny do ojca*,
    - błędy w używaniu przyimków **jechać do Ukrainy* zamiast *jechać na Ukrainę*,
    - błędy w zakresie używania wyrażeń przyimkowych, np. **Pytania odnośnie konstytucji* zamiast *Pytania co do konstytucji* albo *Pytania odnośnie do konstytucji*,
    - niepoprawne skróty składniowe **Organizuje i kieruje szajką złodziei samochodów* zamiast *Organizuje szajkę złodziei samochodów i kieruje nią*,
    - niepoprawne konstrukcje z imiesłowowym równoważnikiem zdania, np. **Zdając egzamin, został przyjęty do szkoły* zamiast *Zdawszy egzamin, został przyjęty do szkoły* albo *Po zdaniu egzaminu został przyjęty do szkoły*,
    - konstrukcje niepoprawne pod względem szyku, np. **Wystąpienie kulturalnego attaché ambasady USA* zamiast *Wystąpienie attaché kulturalnego ambasady USA*,
    - zbędne zapożyczenia składniowe, np. **Dzięki lekarzom zawdzięczam swoje życie* zamiast *Lekarzom zawdzięczam swoje życie*.
- Błędy fonetyczne

[1] W poniższych przykładach za pomocą poprzedzającej gwiazdki (*) oznaczono formy i wypowiedzi uznane w literaturze za błędne.

  - Niepoprawna wymowa głosek, np. *ą* jako [om] na końcu wyrazu (*[idom] zamiast [idą]), *ś* jako [śi] (np. *[śirodek] zamiast [środek]), przydechowe wymawianie głosek *k*, *t*, np. *[k(h)olejny], *[t(h)om],
  - Niepoprawna wymowa grup głoskowych, np. *[kielner] zamiast [kelner], *[nogie] zamiast [noge],
  - „Literowe" odczytywanie wyrazów, np. *[piętnaście] zamiast [pietnaście],
  - Redukcja głosek i grup głoskowych, np. *[tszea] zamiast [tszeba],
  - Niepoprawne akcentowanie wyrazów i form wyrazowych, np. *[**wi**zyta] zamiast [wi**zy**ta][2]

- Błędy leksykalne
  - Błędy słownikowe (językowe)
    - używanie wyrazów w niewłaściwym znaczeniu, czyli zbędna neosemantyzacja[3], np. *dywagacje* jako *ROZWAŻANIA[4] zamiast ODBIEGANIE OD TEMATU, ROZWLEKŁE MÓWIENIE LUB PISANIE NIE NA TEMAT,
    - mylenie znaczeń wyrazów podobnych brzmieniowo lub morfologicznie i ich niepoprawne wymienne używanie, np. *technika* i *technologia*, *efektowny* i *efektywny*, *adaptować* i *adoptować*,
    - posługiwanie się pleonazmami[5], np. **kontynuować dalej*, **akwen wodny*,
    - nadużywanie wyrazów modnych, takich jak *opcja*, *pakiet*, *lider*, *generować*, *strukturalny.*
  - Błędy frazeologiczne
    - zmiana formy frazeologizmów wskutek wymiany, redukcji lub uzupełnienia składu związku, np. **wywrzeć piętno* zamiast *wycisnąć piętno*, **utopić w przysłowiowej łyżce wody* zamiast *utopić w łyżce wody*,
    - zmiana formy frazeologizmu wskutek zmiany postaci gramatycznej jednego ze składników, np. **wiadomość wyssana z palców* zamiast *wiadomość wyssana z palca*,
    - zmiana znaczenia frazeologizmu, np. *potępiać, potępić (kogoś, coś) w czambuł* w znaczeniu *BARDZO ZGANIĆ, POTĘPIĆ (KOGOŚ, COŚ) zamiast POTĘPIĆ WSZYSTKO, WSZYSTKICH BEZ WYJĄTKU,
    - użycie frazeologizmu w niewłaściwym kontekście, powodującym odżycie znaczenia dosłownego, np. **Płodność Polaków i dzietność Polek schodzą na psy.*
  - Błędy słowotwórcze
    - używanie formacji zbudowanej niezgodnie z polskimi modelami słowotwórczymi, np. **speckomisja* zamiast *komisja specjalna*, **biznes informacje* zamiast *informacje o biznesie* czy *informacje biznesowe*,
    - zastosowanie niewłaściwego formantu, np. **głupość* zamiast *głupota*, **matkowy* zamiast *matczyny*,
    - wybór niewłaściwej podstawy słowotwórczej, np. **eurosejm* zamiast *europarlament* (*sejm* jest nazwą tylko parlamentu polskiego).

[2] Następujący [**za**pis] oznacza akcent na sylabę *za-*.

[3] Neosemantyzacja to nadanie istniejącemu już w języku wyrazowi nowego, dodatkowego znaczenia. Nowe znaczenie może pojawić się zarówno wskutek wewnętrznego rozwoju znaczeń wyrazów danego języka, jak i zapożyczeń znaczeń wyrazów z języków obcych.

[4] W zapisie używa się *słowa*, *JEGO BŁĘDNEGO SENSU oraz POPRAWNEGO SENSU.

[5] Pleonazm to wyrażenie złożone z wyrazów o takim samym lub bardzo podobnym znaczeniu.

### 2.2.3. Błędy interpunkcyjne

Do błędów interpunkcyjnych zalicza się:

- brak właściwego znaku interpunkcyjnego, zwłaszcza przecinka,
- zbędne użycie znaku interpunkcyjnego,
- użycie niewłaściwego znaku interpunkcyjnego.

### 2.2.4. Błędy ortograficzne

Do błędów ortograficznych zaliczane są:

- używanie niewłaściwych liter i połączeń literowych w zapisie, np. **gura* zamiast *góra*, **porzyczyć* zamiast *pożyczyć*,
- niewłaściwa pisownia łączna lub rozdzielna oraz niewłaściwe użycie łącznika, np. **dziko-rosnący* zamiast *dziko rosnący*, **wdal* zamiast *w dal*, **jasno zielony* zamiast *jasnozielony*,
- niewłaściwe używanie wielkich i małych liter na początku wyrazów, np. **jan kowalski* zamiast *Jan Kowalski*, **w ostatni Piątek* zamiast *w ostatni piątek*.

## 2.3. Klasyfikacja błędu a konsekwencje dla komunikatu

Abstrahując od klasyfikacji przedstawionej powyżej, błędy można rozróżnić także na poziomie zrozumienia komunikatu. Na tej podstawie wyróżnia się trzy typu błędów [9, s. 1555]:

1. **Błędy rażące** stanowią naruszenie podstawowych zasad poprawnościowych. Obecność tego typu błędu powoduje, że komunikat jest albo zupełnie niezrozumiały albo przekazuje informację inną niż zamierzona. Przykładem jest wyrażenie *dotknąć dziewczynę na ulicy* w znaczeniu *LEKKO TRĄCIĆ, które jednak zgodnie ze schematami składniowymi i ramami walencyjnymi[6] zostaje odebrane jako URAZIĆ DZIEWCZYNĘ NA ULICY. Wyrażenie fizycznej styczności może nastąpić jedynie przez konstrukcję *dotknąć dziewczyny*.
2. **Błędy pospolite** z zasady nie wiążą się z nieporozumieniem na poziomie semantycznym, lecz naruszają pewną normę panującą w danym środowisku, a osoba która je popełnia naraża się na negatywną ocenę ze strony odbiorców, np. brak odmiany nazwiska polskiego **Idę do pana Moniuszko* zamiast *Idę do pana Moniuszki*.
3. **Usterki językowe** naruszają normę tylko w niewielkim zakresie, np. używanie formy przestarzałej **To mój konfident* w znaczeniu *„osoba zaufana"*, przestawny szyk zdania.

## 2.4. Błędy mechaniczne i kognitywne

Inne rozróżnienie wskazuje na błędy mechaniczne i kognitywne [28, s. 644]. Do pierwszej grupy zaliczamy te błędy, które wynikają z czynników fizycznych, np. niewciśniętego prawego klawisza *Alt*. Sztandarowym przykładem dla języka polskiego jest brak znaku diakrytycznego, który może skutkować produkcją zarówno nieistniejącego słowa (**sroda* zamiast *środa*), jak i słowa istniejącego (**musze* zamiast *muszę*, gdzie słowo *musze* to celownik liczby pojedynczej

[6] http://walenty.ipipan.waw.pl/

od *mucha*)[7]. Do tej kategorii należą także pomyłki językowe polegające na zamianie znaków miejscami, o ile taka pomyłka nie występuje systematycznie w komunikatach nadawcy. Błędy mechaniczne związane są z performancją językową. Jako że błąd mechaniczny może skutkować produkcją słowa znajdującego się w słowniku, nie można osiągnąć pełnej dokładności w eliminacji tego typu błędów na zasadzie znajdowania w tekście słów nierozpoznanych przez słownik.

Błędy kognitywne są związane w szczególności z kompetencją językową, a raczej pewnymi jej brakami. Mowa tutaj o błędach rażących czy pospolitych (patrz podrozdział 2.3), które mogą naruszać normę językową w postaci błędów systemowych (patrz podrozdział 2.2.2). Biorąc pod uwagę fakt, że użytkownik popełniając tego typu błąd jest przekonany o poprawności nadawanego komunikatu, błędne słowo lub błędna konstrukcja (niezamierzone przez użytkownika) najprawdopodobniej należą do jego słownika.

[7]Druga sytuacja, w której mamy do czynienia z produkcją słowa istniejącego (znajdującego się w słowniku), w anglojęzycznej literaturze jest opisana mianem *real-word error*. W większych szczegółach traktuje o tym rozdział 4.

# Rozdział 3

# Korpus błędów językowych

Zaznajomiwszy się z wyzwaniami w zakresie tworzenia korpusów błędów, dokonuję opisu wykorzystywanego w pracy korpusu błędów. Odwołując się do rozdziału 2 zwracam uwagę na różnice w kategoryzacji błędów językowych podyktowane względami pragmatycznymi. Końcową część rozdziału poświęcam charakterystyce zasobu bazowego omawianego korpusu.

## 3.1. Korpus językowy

Korpus językowy to przechowywany w postaci cyfrowej zbiór autentycznych tekstów językowych, mówionych i pisanych, umożliwiający prowadzenie badań lingwistycznych. Ideą przyświecają tworzeniu korpusów językowych jest reprezentowanie typowych użyć słów oraz wyrażeń w zróżnicowanych pod względem odmiany, stylu czy typu tekstach [42, s. 12]. Na podobieństwo istniejących korpusów dla języka polskiego, np. Narodowego Korpusu Języka Polskiego [15] można stworzyć zbiór zdań zawierających typowe dla natywnych użytkowników języka polskiego błędy językowe.

## 3.2. Istniejące zasoby i źródła danych

W przeciwieństwie do klasycznych korpusów językowych, których zadaniem jest przedstawienie języka jakim jest, od korpusów błędów wymaga się, aby zawierały zarówno błędny fragment komunikatu językowego oraz jego poprawną wersję. Przykładem zasobu spełniającego te dwa wymagania jest relatywnie mały, liczący 10 000 segmentów *Słownik języka niby-polskiego, czyli błędy językowe w prasie* [17], który jednak nie jest dostępny w postaci cyfrowej[1].

W przypadku potrzeby większej ilości danych z błędami spełnienie wspomnianych wymogów wiąże się z koniecznością ręcznego opatrzenia komentarzem tak dużych zbiorów danych językowych, że koszt czy ilość czasu poświęcona na pozyskanie zasobu okazują się niewspółmierne do oczekiwanego rezultatu, co w ostateczności wstrzymuje proces powstawania zasobu [16, s. 632].

Alternatywą dla całościowego procesu pozyskiwania korpusu błędów, który rozpoczyna znalezienie reprezentatywnej dla danego języka próbki tekstów, jest pominięcie wstępnych kroków i odniesienie się do istniejących już zasobów. Podstawę korpusu mogą stanowić wyniki pracy korektorów tekstu [18, s. 69] czy wyniki pracy w ramach procesu poprawy jakości tłumaczeń obecnego w dużych agencjach tłumaczy. Problematyczny z praktycznego punktu

[1] Mowa tu oczywiście o dostępności w postaci innej niż zdjęcie czy skan książki, które wymagają zastosowania oprogramowania do optycznego rozpoznawania znaków (ang. *optical character recognition - OCR*).

widzenia okazuje się fakt, że dane uzyskane w taki sposób nie mogą być, zgodnie z wymogami klientów, udostępniane na zewnątrz zajmujących się nimi jednostek gospodarczych.

Innym sposobem na pozyskanie błędów językowych oraz ich poprawionych wersji jest skompletowanie wyników egzaminów pisanych przez uczniów w ramach procesu edukacji. Korzystnym z punktu widzenia liczebności oraz standaryzacji próbki wydaje się być egzamin *Programme for International Student Assessment* organizowany z ramienia Organizacji Współpracy Gospodarczej i Rozwoju [19]. Wadę tego rozwiązania stanowi zbyt niski wiek egzaminowanych oraz wynikająca z niego nie w pełni ukształtowana kompetencja językowa (patrz podrozdział 2.1.2).

## 3.3. Historia edycji Wikipedii jako korpus błędów

Wymienione problemy związane z procesem tworzenia korpusu błędów dla języka polskiego skłaniają do odwołania się do rozwiązań częściowo lub w pełni zautomatyzowanych. Przyjmując za prawdziwą hipotezę, że drobne poprawki tekstów (do kilku zmienionych słów) świadczą o obecności błędów składniowych, stylistycznych, ortograficznych czy fleksyjnych, historia edycji artykułów Wikipedii może stanowić cenne źródło błędów oraz ich poprawionych wersji [16, s. 631]. Następujące po sobie zmiany artykułu pozwalają na utworzenie bloków sprzed i po edycji, które odpowiadają błędowi oraz jego poprawionej wersji. Dodatkowymi atutami tak stworzonego korpusu są rozmiar, stały dostęp do źródła danych oraz udział znacznej ilości wolontariuszy w tworzeniu Wikipedii, które przyczyniają się do zwiększenia poziomu reprezentatywności korpusu [21, s. 7].

## 3.4. Analiza korpusu PlEWi

Propozycję rozwiązania problemu braku wystarczająco dużego korpusu błędów dla języka polskiego stanowi *Polish Language Errors from Wikipedia* (PlEWi) [23], który jest publicznie dostępny pod adresem `http://romang.home.amu.edu.pl/plewic/`.

### 3.4.1. Struktura korpusu PlEWi

Korpus ma postać 1348 plików, z których każdy ma format YAML (*YAML Ain't Markup Language*). Format pozwala na reprezentowanie danych w sposób ustrukturyzowany, zachowując jednocześnie czytelność dla odbiorcy[2].

Każdy plik składa się z kilkuset korekt, które mają następującą postać:

```
- :title: Krzew
  :user: Jozef-k
  :comment: lit.
  :revision: '3638234'
- !ruby/object:Plerrex::EditedText
  text: W rozwoju krzew powstaje przez rozwój pączków bocznych znajdujących
    sie u nasady pędu głównego, który zostaje szybko opanowany przez tworzące
    się liczne pędy boczne i wczesnie zamiera.
  new_text: W rozwoju krzew powstaje przez rozwój pączków bocznych, znajdujących
    się u nasady pędu głównego, który zostaje szybko opanowany przez tworzące
    się liczne pędy boczne i wcześnie zamiera.
```

[2] https://yaml.org/spec/current.html

```
attributes:
  :valid_sentence: true
errors:
- !ruby/object:Plerrex::ErrorCorrection
  error: bocznych
  correction: bocznych,
  position: 7
  attributes:
    :category: interpunkcja
- !ruby/object:Plerrex::ErrorCorrection
  error: sie
  correction: się
  position: 9
  attributes:
    :type: :nonword
    :distance: 1
    :category: znaki diakrytyczne
- !ruby/object:Plerrex::ErrorCorrection
  error: wczesnie
  correction: wcześnie
  position: 25
  attributes:
    :type: :nonword
    :distance: 1
    :category: znaki diakrytyczne
```

Znacznik `:title:` odwołuje się do tytułu artykułu, w którym dokonano korekty. `:user:` wskazuje użytkownika, który korekty dokonał. Jeżeli korekty dokonał niezarejestrowany (anonimowy) użytkownik, jako wartość atrybutu podany jest adres IP. `:comment:` to informacja pozostawiona przez osobę edytującą i często dotycząca typu zmiany oraz jej powodu. `:revision:` jednoznacznie identyfikuje korektę w historii edycji Wikipedii. `:text:` oznacza wersję fragmentu artykułu sprzed edycji, natomiast `:new_text:` to wersja zawierająca korektę. `:valid_sentence:` informuje, czy przedmiot korekty jest pełnym zdaniem. W ramach pracy nad korpusem za pełne zdanie uznawano każdy fragment tekstu, który zaczynał się od elementu takiego jak wielka litera, cyfra, cudzysłów, łącznik, natomiast kończył się jednym ze znaków `.;?!"`. Pod znacznikiem `errors:` umieszczono listę błędów. Każdy błąd oznaczony przez

```
!ruby/object:Plerrex::ErrorCorrection
```

składa się z pól:

- `error:` – wartość sprzed edycji,
- `correction:` – wartość po edycji,
- `position:` – który w kolejności token[3] (licząc od lewej i indeksując od 0) został poprawiony,
- `:type:` – czy ciąg znaków z błędem został rozpoznany jako słowo ze słownika (istniejące słowo); `:nonword` jeżeli nie, `:realword` jeżeli tak,

[3]Inaczej segment, ciąg znaków oddzielony od innych ciągów spacjami lub znakami interpunkcyjnymi.

| kategoria | liczba zliczeń |
|---|---|
| pisownia | 328779 |
| interpunkcja | 308508 |
| wielkość liter | 220341 |
| znaki diakrytyczne | 202135 |
| nierozpoznany | 156911 |
| fleksja | 121188 |
| przyimek | 56848 |
| semantyka | 50775 |
| znaki diakrytyczne/kontekst | 39488 |
| pisownia/prawdopodobnie | 27752 |
| styl | 24830 |
| spójnik | 24209 |
| fleksja/czas | 23552 |
| fleksja/liczba | 19741 |
| składnia | 19429 |
| notacja/skrót | 17350 |
| notacja/rok | 16146 |
| pisownia łączna i rozłączna | 13769 |
| semantyka/aspekt | 13609 |
| zaimek | 9709 |
| notacja/wiek | 5270 |
| semantyka/styl | 4744 |
| się | 3689 |
| ilość-liczba | 3448 |
| semantyka/stopień | 142 |

Tablica 3.1: Liczba wystąpień korekt ze względu na kategorię

- `:distance:` – jaki dystans edycji[4] dzieli słowo sprzed korekty i po korekcie,
- `:category:` – kategoria, do której zaliczono korektę.

### 3.4.2. Typy błędów poddane analizie

Analizowany zbiór plików stanowiący korpus PlEWi zawiera łącznie 25 kategorii, których liczebność została przedstawiona w tabeli 3.1. Zliczenia dokonano w oparciu o znacznik `:category:` występujący w każdej sekcji pliku YAML rozpoczynającej się od

```
!ruby/object:Plerrex::ErrorCorrection
```

W dalszych analizach zostaną wykorzystane jedynie niektóre z dostępnych kategorii: *pisownia*, *znaki diakrytyczne*, *znaki diakrytyczne/kontekst*, *składnia*, *fleksja/liczba*. Wybór jest podyktowany skorzystaniem z implementowanego w dalszej części pracy modelu (patrz podrozdział 5.1), który uwzględnia to, że słowa błędne i poprawne dzieli dystans edycji równy 1 (patrz przypis 11 na stronie 47).

[4]W szczegółach o dystansie edycji (odległości edycji) traktuje podrozdział 4.2.

## 3.5. Proces tworzenia korpusu błędów PlEWi

Podstawą do stworzenia PlEWi był plik XML z zawartością całej polskiej edycji Wikipedii [22], z którego uzyskiwano następujące po sobie edycje, korzystając z algorytmu najdłuższego wspólnego podciągu. Pary tak stworzonych wersji poszczególnych stron internetowej encyklopedii były poddawane lematyzacji oraz tokenizacji w celu wydobycia zdań składających się na każdą z wersji. Celem odfiltrowania niestosownych dla korpusu zdań użyto następującego testu [23, s. 132]:

1. Długość każdego ze zdań zawiera się w przedziale od 4 od 80 tokenów.

2. Różnica liczby tokenów dla zdań jest nie większa niż 4.

3. Stosunek słów rozpoznanych przez słownik do tych nierozpoznanych to co najmniej 75%.

4. Stosunek liczby nie-znaków[5] do wszystkich znaków to najwyżej 25%.

Jeżeli co najmniej jeden z warunków nie został spełniony, para zdań była wykluczana z dalszej analizy.

### 3.5.1. Rozpoznawanie błędów językowych

Na podstawie wstępnego filtrowania dla każdej pary zdań utworzono następujący ciąg:

$$((u_0, v_0), (u_1, v_1), ...)$$

gdzie każda para $(u, v)$ stanowi słowo (słowa) przed korektą oraz słowo (słowa) po korekcie. Biorąc pod uwagę obserwację, że duża liczba par (edycji) świadczy raczej o przeformułowaniu zdania aniżeli o drobnej poprawce, zdania dla których długość ciągu wynosiła więcej niż 4 zostały usunięte z dalszej analizy.

Mimo że klasyfikacja błędów językowych przedstawiona we wcześniejszym podrozdziale 2.2 stanowi całościowy przekrój błędów występujących w języku polskim, nie zwraca jednak uwagi na praktyczne zastosowanie w zakresie ich wykrywania. Biorąc pod uwagę wykorzystywane w przetwarzaniu języka naturalnego narzędzia, należy dokonać syntezy dotychczas wymienionych kategorii błędów i zdefiniować nowe, praktyczniejsze kategorie [25, s. 111-112].

#### Błędy proste

Do błędów prostych zaliczamy pary $(u, v)$, które charakteryzują się:

- brakującym znakiem interpunkcyjnym,

- brakiem lub nadmiarem spacji lub łącznika,

- niewłaściwą wielkością liter.

[5] Chodzi tu o znaki kontroli (ang. *control characters*), które nie są drukowane, ale mimo to wpływają na kształt tekstu, np. znak nowej linii `\n`.

**Błędy pisowni**

Jeżeli para $(u, v)$ nie została rozpoznana jako błąd prosty, przechodzi do weryfikacji przeprowadzonej przy użyciu narzędzia do sprawdzania pisowni (Hunspell [26]), które zawiera słownik $D$. W zależności od tego, z jakim typem słów w parze mamy do czynienia pary można traktować w odmienny sposób:

1. $u \notin D$ i $v \in D$, czyli słowo nieistniejące zostało poprawione na słowo istniejące. Tutaj dokonuje się rozbicia na pary, w których $u$ nie zawierało znaku diakrytycznego (natomiast $v$ zawierało) oraz takie, które stanowią inny błąd pisowni.

2. $u, v \in D$, czyli słowo istniejące zostało poprawione na istniejące. Pary z tej grupy są bezwarunkowo akceptowane jako odpowiedni element korpusu błędów.

3. $u \in D$ i $v \notin D$, czyli słowo istniejące zostało zmienione na słowo nieistniejące, co świadczy o akcie wandalizmu. Pary są bezwarunkowo eliminowane z dalszej analizy.

4. $u, v \notin D$, czyli słowo sprzed korekty i po korekcie nie należą do słownika. Biorąc pod uwagę obserwację dla języka angielskiego, że większość błędów pisowni nie skutkuje znacząco innym ciągiem znaków od poprawnego [27, s. 388], elementy pary $(u, v)$ oddalone o dystans edycji nie większy niż 4 są klasyfikowane jako „prawdopodobne błędy pisowni".

**Błędy gramatyki**

W tej grupie zwraca się uwagę w szczególności na błędy fleksyjne, które są częste w językach z bogatą morfologią. Pary $(u, v)$ należące do tej grupy charakteryzują się tym, że słowa $u$ i $v$ poddane procesowi lematyzacji stają się tym samym słowem, co oznacza wspólną formę podstawową tych słów.

## 3.6. Charakterystyka PlEWi – reprezentatywność

W procesie tworzenia korpusu błędów na podstawie swoistego korpusu językowego jakim jest Wikipedia dochodzi do propagacji ograniczeń wynikających z charakterystyki internetowej encyklopedii. W wątpliwość można poddawać twierdzenie o reprezentatywności błędów językowych zebranych w procesie tworzenia korpusu. Zgodnie ze stanem na rok 2018 [24] niecałe 2000 osób deklarowało się jako wikipedyści, czyli osoby edytujące Wikipedię (nawet okazjonalnie, bez długoterminowego zaangażowania), natomiast ok. 500 osób aktywnie uczestniczy w życiu społeczności (przyznawanie uprawnień użytkownikom, wybory do grup roboczych). Aż 70% spośród użytkowników, którzy zdecydowali się ujawnić swój wiek, urodziło się pomiędzy 1980 a 2000 r. Ponadto, podobnie jak anglojęzyczna wersja wolnej encyklopedii, polska Wikipedia charakteryzuje się tym, że jest edytowana głównie przez mężczyzn, ponieważ kobiety stanowią 10–15% społeczności. Wśród osób, które ujawniły poziom wykształcenia, przeważają te z wykształceniem wyższym (ukończone studia licencjackie, studia inżynierskie, studia magisterskie, studia podyplomowe) i stanowią 71% respondentów. Ponadto formalny charakter artykułów, niejednokrotnie wzorowany na pracach naukowych, skłania ku wnioskowi, że styl językowy Wikipedii można określić jako formalny. Biorąc pod uwagę wszystkie te obserwacje dotyczące społeczności wikipedystów, można wnioskować, że błędy znalezione w historii edycji mogą być niereprezentatywną próbką błędów pisowni popełnianych przez natywnego użytkownika języka polskiego.

# Rozdział 4

# Metody słownikowe

Rozpoczynając rozdział od krótkiego rysu historycznego, przechodzę do zasady działania narzędzi autokorekty opartych na słowniku. Zwracam uwagę na błędy typu *real-word*, a następnie omawiam możliwości mierzenia podobieństwa słów w oparciu o ich literowy zapis. Końcową część rozdziału poświęcam algorytmowi służącemu obliczeniu jednej z miar podobieństwa.

## 4.1. Rys historyczny i opis podejścia

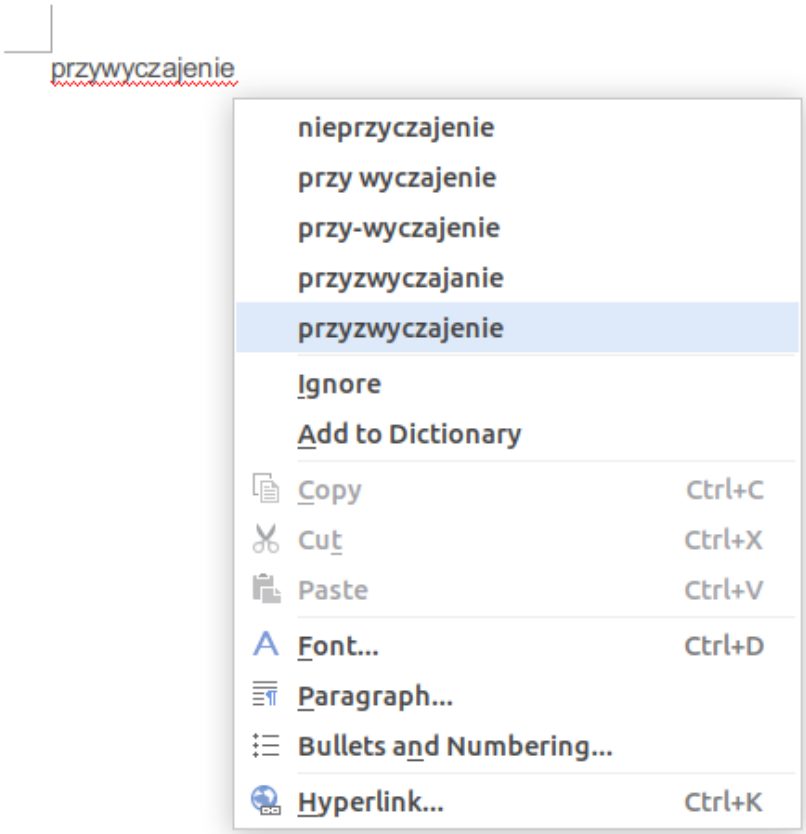
Programy komputerowe służące do korekty tekstu w języku angielskim istnieją od co najmniej 1960 roku, kiedy przedstawiono kompleksowy system korekty błędów korzystający m.in. z obserwacji dotyczącej częstości występowania w tekstach pisanych liter podlegającej prawu Zipf'a[1] oraz algorytmowi Soundex, który rozpoznaje słowa podobnie brzmiące przypisując im 4-znakowy kod [30]. Jako że do połowy lat 60. XX wieku warianty odległości edycji opisane w rozdziale 4.2.5 oraz 4.2.6 nie były znane, wspomniany system korzystał z faktu, że niektóre często występujące w języku angielskim litery mogą być usunięte z wyrazu, pozostawiając czteroliterową postać dla każdego co najmniej 5-literowego słowa. Jeżeli skrócona forma wyrazu błędnie zapisanego odpowiadała skróconej formie wyrazu poprawnego, wyraz błędnie zapisany zamieniany był na ten poprawny [54]. Później pojawiły się metody, które w sposób deterministyczny obliczały dystans edycji pomiędzy słowami i na tej podstawie mogły decydować o proponowanych poprawkach dla słowa [55].

Mianem metod słownikowych w korekcie tekstu określa się te metody, które dają odpowiedź na pytanie czy dany ciąg znaków jest poprawnym słowem na podstawie sprawdzenia czy słowo znajduje się w słowniku. Efektywność metody zależy jedynie od rozmiaru słownika: im więcej słów zawiera, tym mniej zachodzi fałszywych trafień (ang. *false positives*) oraz zwiększa się liczba prawdziwie pozytywnych obserwacji (ang. *true positives*).[2]

Opisywane metody, korzystając z dołączonego słownika poprawnie zapisanych słów, wyróżniają (np. czerwonym podkreśleniem) słowa potencjalnie błędne. Słowa zaznaczone niekoniecznie stanowią błąd, ponieważ możliwy jest przypadek, że poprawne słowo na tyle rzadko występuje w języku, że używany przez system korekty tekstu zbyt mały słownik tego słowa nie zawiera, co skutkuje wyróżnieniem słowa w interfejsie użytkownika. Przy pomocy metod

[1] W prawie Zipfa rozważa się listę rankigową słów korpusu, czyli listę słów uporządkowanych od największej do najmniejszej frekwencji. Jeżeli najczęściej występującemu słowu nadamy rangę 1, kolejnemu 2 itd., to w przybliżeniu częstość wystąpienia słowa będzie odwrotnie proporcjonalna do jego rangi.

[2] Przez fałszywe trafienie rozumiem sytuację, w której słowo poprawne zostało rozpoznane przez system korekty tekstu jako niepoprawne tylko dlatego, że nie znajdowało się w słowniku. Prawdziwie pozytywna obserwacja to sytuacja, w której poprawne słowo zawarte jest w słowniku i nie jest oznaczane jako błąd.




Rysunek 4.1: Interfejs programu autokorekty z perspektywy użytkownika pakietu WPS Office 2016 (nad słowem *przyzwyczajenie* znajdował się kursor kontrolowany przez użytkownika).

słownikowych nie da się na podstawie błędnego słowa wywnioskować jakie słowo użytkownik miał pierwotnie na myśli. Wydawałoby się, że w celu udoskonalenia narzędzi autokorekty tekstu należy zwiększać liczbę pozycji w słowniku, jednak takie podejście nie uwzględnia faktu, że jedyne co metody słownikowe oferują to lista propozycji, z których użytkownik wybiera słowo, które miał na myśli, jak na rysunku 4.1. Warto podkreślić, że metody słownikowe nie zaznaczą słowa, którego użytkownik nie miał na myśli podczas wprowadzania tekstu, a które znajduje się w słowniku, jak się zdarza w przypadku malapropizmów, np. **W biurze znajduje się nowy* ***czytelnik*** *kart magnetycznych* zamiast *W biurze znajduje się nowy* ***czytnik*** *kart magnetycznych.*

Malapropizmy, czyli omyłki słowne, które polegają na błędnym użyciu w wypowiedzi wyrazu podobnie brzmiącego, to podgrupa błędów polegających na zapisaniu rzeczywiście istniejącego słowa w miejsce słowa poprawnego (ang. *real-word error*). Według różnych szacunków odsetek tego typu błędów w tekstach anglojęzycznych może wahać się od 25% do 33% [56, s. 87]. W pracach pisemnych uczniów (rodzimych użytkowników języka angielskiego) znalazło się 40–50% błędów typu *real-word* [57, s. 497]. Dla języka polskiego trudno o podobne dane, jednak skalę problemu tego typu błędów można szacować na podstawie korpusu PlEWi opisanego w podrozdziale 3.4, gdzie błędy typu *real-word* stanowią 42% wszystkich błędów.

## 4.2. Odległość edycji

Jak stwierdzić czy dwa słowa są do siebie podobne? Wykluczając semantykę, a skupiając się na graficznej (znakowej) reprezentacji słów, można badać liczbę liter, które występują w obu słowach. Dla problemu korekty tekstu takie podejście jest niewystarczające, ponieważ nadawcy nie zapisują swoich komunikatów w postaci losowego ciągu liter – przed zapisaniem danego słowa mają pewien (nie)świadomy plan, w które klawisze należy uderzyć i w jakiej kolejności, aby wyprodukować zamierzone słowo. Nakładając na badanie liczby wspólnych liter warunek dotyczący tego, że znaczące dla wyniku badania litery muszą występować w takiej samej kolejności w słowach, można wyprowadzić metryki najdłuższego wspólnego podciągu (ang. *longest common subsequence*)(patrz podrozdział 4.2.2) oraz najdłuższego wspólnego podłańcucha (ang. *longest common substring*)(patrz podrozdział 4.2.3).

Druga grupa metryk uwzględnia liczbę wspólnych liter jedynie pośrednio. Odległość edycji (ang. *edit distance*) to miara wyrażająca podobieństwo dwóch ciągów znaków. Informuje o tym, ile (minimum) operacji należy wykonać, żeby zmienić jeden ciąg w drugi [33, s. 81]. Bardziej formalnie, mając ciągi $a$ oraz $b$ nad alfabetem $\Sigma$, dystans edycji $d(a, b)$ to ten spośród ciągów operacji, który ma najmniejszą sumaryczną wagę, a skutkuje przekształceniem $a$ w $b$. W zależności od dopuszczonych podstawowych operacji oraz założeń dotyczących wejściowych ciągów znaków można wyróżnić więcej niż jedną realizację idei odległości edycji.

### 4.2.1. Metryka

Funkcję, która określa odległość między każdą parą elementów niepustego zbioru[3] nazywamy metryką [32, s. 31]. Niech $X$ oznacza dowolny niepusty zbiór. Metryka w zbiorze $X$ to następująca funkcja:

$$d : X \times X \to \left[0, +\infty\right) \tag{4.1}$$

Dla dowolnych elementów *a, b, c* tego zbioru $d$ spełnia następujące warunki:

1. identyczność nierozróżnialnych

$$d(a, b) = 0 \iff a = b \tag{4.2}$$

2. symetria

$$d(a, b) = d(b, a) \tag{4.3}$$

3. nierówność trójkąta

$$d(a, b) \leq d(a, c) + d(c, b) \tag{4.4}$$

[3]W teorii mnogości zbiór jest pojęciem pierwotnym, definiowanym aksjomatycznie [61, s. 327]. Słowo *zbiór* przyjmuje znaczenie dystrybutywne, czyli abstrakcyjne. Oznacza to, że przedmioty, które tworzą zbiór są jego elementami. Określając zbiór w sensie dystrybutywnym zwracamy uwagę nie tylko na elementy zbioru, ale także to w jaki sposób do niego przynależą. Zgodnie z dominującą koncepcją przyjmuje się, że przedmiot jest lub nie jest elementem danego zbioru [60, s. 162].

### 4.2.2. Najdłuższy wspólny podciąg

Mianem najdłuższego wspólnego podciągu określa się najdłuższy podciąg znaków występujący w każdym z dwóch łańcuchów[4]. W przypadku tej miary liczy się jedynie to, aby elementy podciągu występowały w tej samej kolejności, ale niekoniecznie obok siebie [41, s. 39]. Przykładowo mając dwa łańcuchy znaków – ***polityka*** oraz ***polit****echni****ka*** – ich najdłuższy wspólny podciąg będzie miał długość 7, mimo że jedynie pierwszych pięć elementów leży w obu łańcuchach obok siebie, natomiast dwa ostatnie wspólne elementy są oddalone o różną ilość znaków od pierwszej części wspólnego podciągu. Założenie, że elementy najdłuższego wspólnego łańcucha nie muszą stać obok siebie sprawia, że podejście to nie nadaje się do analizy błędów pisowni.

### 4.2.3. Najdłuższy wspólny podłańcuch

W odróżnieniu od najdłuższego wspólnego podciągu najdłuższy wspólny podłańcuch nakłada obostrzenie w postaci wspólnych elementów występujących nie tylko w tej samej kolejności, ale obok siebie. Dla podanego wcześniej przykładu najdłuższy wspólny podłańcuch wynosi 5, ponieważ tylko 5 pierwszych elementów ciągów znaków występuje w tej samej kolejności i obok siebie. Sufiks *-ka* jest oddalony o różną liczbę liter od wspólnej początkowej części.

### 4.2.4. Odległość Hamminga

W tym podejściu zakłada się, że porównywane ciągi znaków są równej długości. Odległość Hamminga to metryka, która daje odpowiedź na pytanie, w ilu punktach dwa analizowane ciągi różnią się między sobą [34, s. 156]. Innymi słowy, informuje, w ilu miejscach ciągu $b$ należy zastąpić element odpowiadającym elementem z ciągu $a$.

Weźmy $a$ = chodzić oraz $b$ = płodzić. Odległość Hamminga $d_H(a, b)$ wyniesie 2, ponieważ ciągi znaków różnią się na dwóch miejscach:

chodzić
płodzić

### 4.2.5. Odległość Levenshteina

Odległość Levenshteina [35, s. 707-709] stanowi uogólnienie odległości Hamminga, ponieważ jako dane wejściowe przyjmuje dwa ciągi znaków o różnej lub tej samej długości, a operacje (każda o koszcie 1) to:

1. wstawienie nowego znaku do napisu, np. $ac \rightarrow abc$
2. usunięcie znaków z napisu, np. $abc \rightarrow ac$
3. zamiana znaku w napisie na inny znak $abc \rightarrow abd$[5]

[4] Problem najdłuższego wspólnego podciągu można generalizować na $n$ łańcuchów, jednak zgodnie z założeniami prezentowanego w dalszej części pracy modelu wystarczająca jest tutaj wersja dla dwóch łańcuchów znaków.

[5] Należy zaznaczyć, że chodzi tu nie o zmianę na dowolny znak, lecz o zmianę na znak na odpowiadającej pozycji w drugim analizowanym słowie. Zamiana na dowolny znak odpowiadałaby dwóm operacjom: usunięcia (koszt 1), a następnie wstawienia znaku (także koszt 1) wykonanym w operacji o jednostkowym koszcie. Więcej szczegółów znajduje się w podrozdziale 4.3.2.

Z punktu widzenia problemu korekty tekstu opisywana miara jest niedostateczna, ponieważ dyskryminuje błąd w ciągu znaków polegający na zamianie miejscami dwóch znaków. Weźmy ciąg znaków **peis* nienależący do słownika, a będący niepoprawnym zapisem słowa *pies*. Korzystając z odległości Levenshteina można przetransformować **peis* na *pies* za pomocą dwóch operacji:

$$peis \rightarrow pis \text{ (koszt 1)}$$

a następnie

$$pis \rightarrow pies \text{ (koszt 1)}$$

Intuicyjnie, rozumowanie, że zamiana miejscami dwóch sąsiadujących liter jest droższa (ma większą wagę) od wstawienia czy usunięcia pojedynczej litery wydaje się błędne.

### 4.2.6. Odległość Damerau-Levenshteina

Obserwacja, że zamiana sąsiednich liter miejscami powinna stanowić operację o jednostkowym koszcie doprowadziła do powstania odległości Damerau-Levenshteina [55], która do listy trzech podstawowych operacji wymienionych w podrozdziale 4.2.5 zalicza dodatkowo operację:

4. transpozycja, np. $abcd \rightarrow acdb$

Wspomniana powyżej zamiana **peis* na *pies* wymaga wykonania jedynie operacji transpozycji, co skutkuje kosztem zamiany błędnego słowa na poprawne równym 1.

Odległość Damerau-Levenshteina traktuje analizowane słowa jako ciągi nieokreślonych znaków, mimo że w rzeczywistości stanowią one zapis fonemów[6]. Rozważmy dwa przykłady:

1. **morze* zamiast *może*
   Aby dojść z błędnie wpisanego wyrazu do zamierzonego należy wykonać następujące operacje:

   $$morze \rightarrow moze \text{ (koszt 1)}$$

   $$moze \rightarrow może \text{ (koszt 1)}$$

2. **loże* zamiast może

   Wystarczy jedna operacja, która sprawia, że błąd zostaje wyeliminowany:

   $$loże \rightarrow może \text{ (koszt 1)}$$

Biorąc pod uwagę, że zasady ortograficzne stanowią zewnętrzny element języka, tzn. taki, który został narzucony sztucznie i ma charakter dyskusyjny, wydaje się nieintuicyjnym, że słowa w pierwszym przypadku dzieli dystans edycji równy dwa, podczas gdy w drugim wynosi on 1. W swojej podstawowej wersji dystans Damerau-Levenshteina zakłada, że każda z podstawowych operacji edycji ma takie samo prawdopodobieństwo zajścia. Poza tym nie zwraca uwagi na to, jakie litery są edytowane. Rozważmy inne dwa przykłady:

1. **rest* zamiast *test*

   Klawisze R oraz T są położone obok siebie na klawiaturze QWERTY przedstawionej na rysunku 4.2[7]. Dodatkowo obsługiwane są przez lewy palec wskazujący, co zwiększa prawdopodobieństwa pomylenia klawiszy ze sobą w wyniku błędu mechanicznego (patrz podrozdział 2.4).

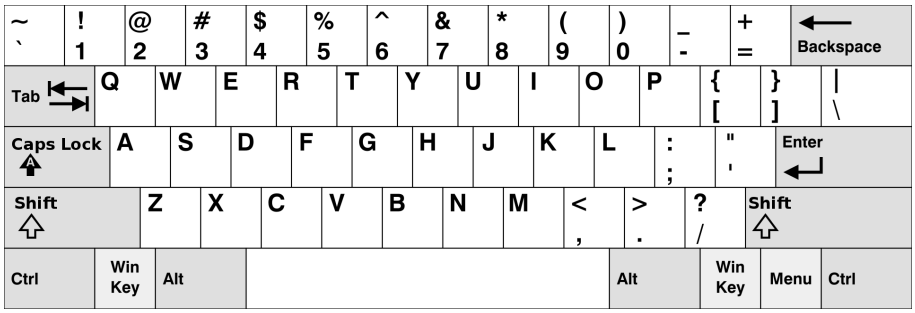
[6] Fonem to podstawowy, abstrakcyjny element struktury fonologicznej różnicujący znaczenie form wyrazowych, realizowany w mowie przez głoski [52, s. 238].

[7] `https://en.wikipedia.org/wiki/QWERTY#/media/File:KB_United_States.svg`

2. **lest* zamiast *test*

   Klawisze L oraz T są położone relatywnie daleko od siebie, do tego stopnia, że obsługiwane są przez dwie różne dłonie, odpowiednio przez palec wskazujący lewej ręki oraz palec serdeczny prawej ręki. Prawdopodobieństwo pomylenia klawiszy ze sobą w wyniku błędu mechanicznego jest niższe niż w przypadku pierwszym.

Dla obu tych rozważanych przypadków dystans Damerau-Levenshteina wynosi 1, gdzie jedyna wymagana operacja to zamiana $r$ na t oraz $l$ na $t$. Rozważania uwzględniające rozkład klawiszy na klawiaturze QWERTY wskazują na nieadekwatność dystansu w kontekście korekty teksty, jednak dystans ten możliwie całościowo odzwierciedla możliwe pomyłki prowadzące do błędnego zapisu słów.




Rysunek 4.2: Rozkład klawiatury QWERTY

## 4.3. Obliczanie dystansu Damerau-Levenshteina

### 4.3.1. Programowanie dynamiczne

Rozważamy przykład dotyczący przekształcenia słów, gdzie ciąg znaków tworzący słowo *kognitywistyka* chcemy zmienić na ciąg znaków tworzący słowo *kognitywista*, korzystając z czterech podstawowych operacji zdefiniowanych w metodzie obliczania odległości Damerau-Levenshteina (patrz podrozdział 4.2.6). Przykładowo istnieją dwie ścieżki:

1. Pierwsza ścieżka:

   $$kognitywistyka \rightarrow kognitiwistyka \text{ (koszt 1)}$$

   następnie

   $$kognitiwistyka \rightarrow kognitiwistya \text{ (koszt 1)}$$

   w kolejnym kroku

   $$kognitiwistya \rightarrow kognitiwista \text{ (koszt 1)}$$

   na koniec

   $$kognitiwista \rightarrow kognitywista \text{ (koszt 1)}$$

2. Druga ścieżka:

   $$kognitywistyka \rightarrow kognitywistya \text{ (koszt 1)}$$

   w kolejnym kroku

   $$kognitywistya \rightarrow kognitywista \text{ (koszt 1)}$$

Obie ścieżki doprowadzają do tego samego rozwiązania, ale posiadają różny sumaryczny koszt (cztery dla pierwszej ścieżki, dwa dla drugiej ścieżki). Można wyobrazić sobie podejście, które generuje wszystkie możliwe ścieżki, dla każdej obliczając sumaryczny koszt, a na

koniec wybiera tę, dla której koszt jest najmniejszy. Oczywiście podejście takie jest wysoce nieefektywne i dlatego może być udoskonalone. Narzędziem służącym udoskonaleniu jest programowanie dynamiczne. Programowanie dynamiczne to metoda rozwiązywania problemów optymalizacyjnych. Z problemem optymalizacyjnym mamy do czynienia, gdy od danych wejściowych do pożądanych danych wyjściowych możemy przejść wykonując różne ciągi operacji, co skutkuje różnym końcowym kosztem dla różnych ciągów operacji.

Zgodnie z koncepcją przedstawioną w literaturze [50, s. 362], proces projektowania algorytmu dynamicznego dzieli się na następujące etapy :

1. Scharakteryzowanie struktury optymalnego rozwiązania.
2. Rekurencyjne zdefiniowanie wartości (kosztu) optymalnego rozwiązania.
3. Obliczenie wartości optymalnego rozwiązania metodą wstępującą.
4. Skonstruowanie optymalnego rozwiązania na podstawie wyników wcześniejszych obliczeń.

Programowanie dynamiczne usprawnia rozwiązywanie problemów w sytuacji, gdy w toku obliczeń należy rozwiązać dany podproblem więcej niż raz. Korzystając z dodatkowej pamięci (kompromis między czasem a pamięcią), algorytm w duchu programowania dynamicznego zapisuje rozwiązanie podproblemu i w przypadku konieczności powtórzenia obliczeń odwołuje się do pamięci. Programowanie dynamiczne, w przeciwieństwie do podejścia *dziel i rządź* (ang. *divide and conquer*)[8], zalecane jest w sytuacjach, gdy podproblemy nie są niezależne.

Istnieją dwie równoważne implementacje podejścia dynamicznego.

#### Metoda zstępująca ze spamiętywaniem

Problem rozwiązywany jest rekurencyjnie z tym udoskonaleniem, że wyniki każdego rozwiązanego podproblemu są zapisywane w tablicy. Przed włączeniem procedury rekurencyjnej na danych wejściowych funkcja sprawdza, czy tablica nie zawiera wyników dla tych danych.

#### Metoda wstępująca

Zakłada się, że rozwiązanie problemu zależy jedynie od rozwiązania składających się na niego podproblemów. Rozmiar rozumiany jest tutaj w kategorii danych wejściowych, np. długości ciągu znaków czy liczby elementów macierzy. Procedura rozwiązywania problemu rozpoczyna się od sortowania podproblemów od najmniejszego do największego, a następnie rozwiązywania ich w tej kolejności, gdzie danymi wejściowymi dla większego podproblemu są m.in. rozwiązania składających się na niego mniejszych podproblemów.

### 4.3.2. Algorytm obliczający odległość Damerau-Levenshteina

Zdefiniowana poniżej funkcja zwraca dystans Damerau-Levenshteina dla podciągów $a$ oraz $b$ rozpoczynających się pierwszym znakiem $a$ oraz $b$, a kończących się odpowiednio w $i$-tym oraz $j$-tym znaku ciągów $a$ oraz $b$ [53, s. 11]. W szczególności, gdy $i$ równa się długości $a$ oraz $j$ równa się długości $b$ funkcja zwraca dystans edycji Damerau-Levenshteina dla $a$ i $b$.

[8]Podejście *dziel i rządź* w projektowaniu algorytmów zakłada podział problemu na niezależne podproblemy, które także, w zależności od potrzeb, można podzielić na kolejne podproblemy. Całościowe rozwiązanie problemu uzyskuje się przez połączenie rozwiązań dla podproblemów.

$$d_{a,b}(i{+}1, j{+}1) = min \begin{cases} 0 & \text{dla } i = j = 0 \quad (1) \\ d_{a,b}(i+1, j) + 1 & \text{dla } i > 0 \quad (2) \\ d_{a,b}(i, j+1) + 1 & \text{dla } j > 0 \quad (3) \\ d_{a,b}(i, j) + 1_{(a_{i+1} \neq b_{j+1})} & \text{dla } i, j > 0 \quad (4) \\ d_{a,b}(i-1, j-1) + 1 & \text{dla } i, j > 1 \wedge a[i+1] = b[j] \wedge a[i] = b[j+1] \quad (5) \end{cases} \tag{4.5}$$

$$1_{(a_i \neq b_j)} = \begin{cases} 1 & \text{dla } a[i] = b[j] \\ 0 & \text{w pozostałych przypadkach} \end{cases} \tag{4.6}$$

W równaniu 4.5 przypadek (1) to sytuacja, w której oba ciągi znaków są puste. Przypadek (2) opisuje operację usunięcia znaku, (3) wstawienia, (4) zamiany liter na odpowiadających sobie miejscach (koszt wynosi 0, gdy chodzi o tą samą literę), (5) to operacja transpozycji.

W przykładzie zastosowania algorytmu Damerau-Levenshteina zostanie użyta tabela wspomniana w podrozdziale 4.3.1, w której będą znajdowały się liczby informujące o minimalnym koszcie przekształcenia ciągu z kolumny A w ciąg w wierszu o numerze 1. Rozważany przykład dotyczy transformacji ciągu znaków *abcde* w ciąg *abdcfe*. Inicjalizacja tabeli 4.1 polega na wypełnieniu pierwszej kolumny i pierwszego wiersza. Pierwsza kolumna zawiera znak pustego wyrazu ($\#$) oraz litery tworzące słowo wejściowe (które zbierane kumulatywnie podczas przechodzenia z góry kolumny na jej dół utworzą słowo wejściowe). Do pierwszego wiersza trafia znak pustego wyrazu oraz litery tworzące słowo wyjściowe (docelowe). Zawartość drugiego wiersza dotyczy kosztu przekształcenia pustego ciągu w kolejne podciągi ciągu docelowego (stale rozpoczynające się od pierwszego znaku), czyli *a*, *ab*, *abd* itd. Kolumna B opisuje ten sam proces z tą różnicą, że pusty ciąg jest przekształcany w kolejne podciągi słowa wejściowego.

W nawiązaniu do podrozdziału 4.3.1 poświęconego idei rozbijania problemów na podproblemy, które nie są niezależne, rozważmy komórkę E4, która dotyczy przekształcenia ciągu *ab* w *abd* w kontekście pięciu operacji opisanych równaniem 4.5. Operacja (1) jest niemożliwa, ponieważ $i \neq 0$ oraz $j \neq 0$. Operacja (2), która opisuje usunięcie znaku, odwołuje się do komórki E3, co daje koszt 3. Operacja (3) opisująca wstawienie znaku i odwołująca się do komórki D4 daje koszt równy 1. Operacja (4) odwołująca się do D3 i dotycząca zamiany znaku na inny daje koszt 2. Transpozycja opisana operacją (5) jest niemożliwa do wykonania, ponieważ nie zachodzi warunek $a[i] = b[j+1]$, gdzie $a$ to ciąg wejściowy, a $b$ to ciąg pożądany. Z operacji wybieramy tą, która posiada najniższy koszt i zapisujemy go w komórce E4. Takim sposobem dochodzimy przez kolejne komórki do finalnej pozycji H7 zawierającej sumaryczny koszt przekształcenia *abcde* w *abdcfe*, który wynosi 2.

Wykonane operacje to:

$$abcde \rightarrow abdce \text{ (koszt 1)}$$

$$abdce \rightarrow abdcfe \text{ (koszt 1)}$$

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | | # | **a** | **b** | **d** | **c** | **f** | **e** |
| 2 | # | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| 3 | **a** | 1 | 0 | 1 | 2 | 3 | 4 | 5 |
| 4 | **b** | 2 | 1 | 0 | 1 | 2 | 3 | 4 |
| 5 | **c** | 3 | 2 | 1 | 1 | 1 | 2 | 3 |
| 6 | **d** | 4 | 3 | 2 | 1 | 1 | 2 | 3 |
| 7 | **e** | 5 | 4 | 3 | 2 | 2 | 1 | 2 |

Tablica 4.1: Tabelaryczne ujęcie podproblemów składających się na problem dystansu edycji pomiędzy ciągami znaków *abcde* a *abdcfe*

# Rozdział 5

# Metody kontekstowe

## 5.1. Model kanału z szumem

Narzędzia służące do autokorekty tekstu można podzielić na statystyczne (ang. *machine learning-based*) oraz oparte o reguły (ang. *rule-based*)[1]. Grupa metod statystycznych może zostać podzielona na następujące podgrupy:

1. metody statystyczne typu *positive-match*, które korzystają z dużych i reprezentatywnych korpusów błędów dla danego języka. Wykrywanie błędów w tekście polega na wyszukiwaniu takich sekwencji słów, które pojawiają się w korpusie i próbach zamiany elementów sekwencji w tekście na poprawną,

2. metody statystyczne typu *negative-match* oparte na wykorzystaniu dużych, przeważnie wolnych od błędów korpusów językowych i wskazujące jako błędy sekwencje słów, które występują rzadko w korpusie lub nie występują wcale.

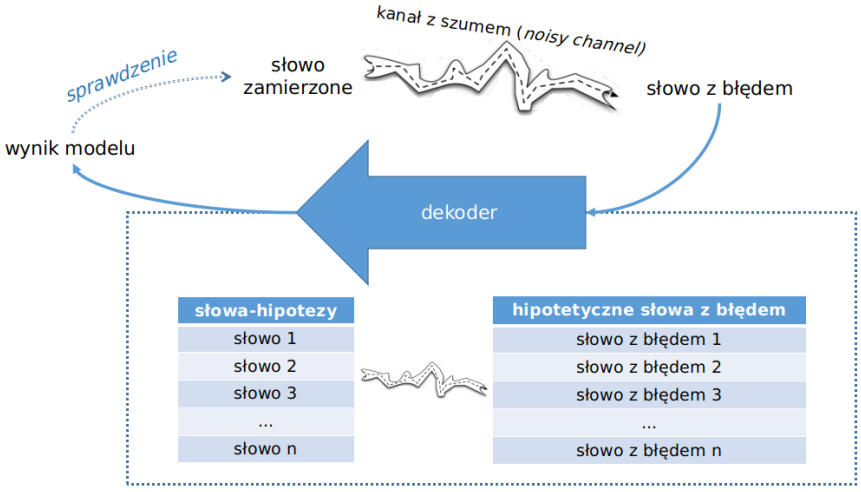
Model kanału z szumem (ang. *noisy channel model*) wykorzystuje metody statystyczne typu *negative-match*, ale korzysta także z elementów (tj. wykorzystanie korpusu błędów) charakterystycznych dla metod statystycznych typu *positive-match*.

Omawiany model można rozumieć intuicyjnie jako sytuację, gdzie błędne słowo w tekście stanowi produkt przejścia zamierzonego przez użytkownika słowa przez pewien kanał zniekształcający. Abstrahując od realizacji kanału, wprowadza on do pierwotnie poprawnie zapisanego słowa jedną lub więcej zmian polegających na wykonaniu podstawowych operacji zdefiniowanych w podrozdziale 4.2.5 poświęconym odległości Levenshteina oraz podrozdziale 4.2.6 poświęconym odległości Damerau-Levenshteina. Aby uzyskać ze zniekształconego słowa jego pierwotną, poprawną wersję, przez kanał z szumem należy przepuścić poprawne słowa zawarte w słowniku[2], a następnie wybrać to, którego zniekształcona forma jest równa słowu z błędem [33, s. 43].

Model kanału z szumem stanowi przykład zastosowania wnioskowania bayesowskiego. Załóżmy, że obserwujemy słowo $O$, które zawiera błąd pisowni. Zadanie polega na znalezieniu

[1] Zgodnie z propozycją przedstawioną w [58] metody regułowe można podzielić na dwie grupy. Pierwsza grupa to metody regułowe typu *positive-match*, które wykorzystują obserwacje autorów systemów dotyczące częstych błędów w języku, na podstawie których definiowane są reguły. Podejście tego rodzaju wykorzystywane jest w narzędziu autokorekty tekstu LanguageTool (`https://languagetool.org/`). Druga grupa to metody regułowe typu *negative-match*, które w oparciu o gramatyki formalne opisują poprawne wzorce użycia słów czy wyrażeń.

[2] Najogólniej rzecz ujmując słownik może stanowić zbiór wszystkich słów języka uznanych za poprawne, chociaż można listę słów (słownik) zawężać do kilku lub jednej pozycji w miarę potrzeb.




Rysunek 5.1: Schematyczne ujęcie modelu kanału z szumem

takiego słowa $w$ należącego do słownika $V$, że prawdopodobieństwo warunkowe[3] $P(w|O)$ jest największe:

$$\hat{w} = \underset{w \in V}{\operatorname{argmax}} P(w|O) \tag{5.1}$$

Korzystając ze wzoru Bayesa[4], równanie 5.1 może zostać przekształcone do następującej postaci:

$$\hat{w} = \underset{w \in V}{\operatorname{argmax}} \frac{P(O|w)P(w)}{P(O)} \tag{5.2}$$

gdzie $P(O)$ to prawdopodobieństwo, że obserwacja zaszła, $P(w)$ to prawdopodobieństwo wystąpienia słowa $w$, natomiast $P(O|w)$ oznacza prawdopodobieństwo zajścia obserwacji pod warunkiem słowa $w$.

Dla każdego proponowanego jako pierwotnie poprawne słowa $w \in V$ prawdopodobieństwo zaobserwowania słowa z błędem $O$ jest stałe, dlatego we wzorze 5.2 można pominąć $P(O)$, co

[3]Prawdopodobieństwo warunkowe to model zajścia zdarzenia A pod warunkiem, że zaszło zdarzenie B [40, s. 48]:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

[4]Wzór na prawdopodobieństwo całkowite pozwala na obliczenie prawdopodobieństwa zdarzeń, które mogą zajść w wyniku realizacji innych zdarzeń. Rozbicie zbioru zdarzeń elementarnych $\Omega$ to rodzina zdarzeń $\{H_i\}_{i \in I}$, które parami wykluczają się, a ich suma jest równa $\Omega$ [40, s. 55]. W dalszych rozważaniach zakładam, że rozbicie jest przeliczalne (skończone). Jeżeli $\{H_1, H_2, ..., H_n\}$ jest rozbiciem $\Omega$ na zdarzenia o dodatnich prawdopodobieństwach, to dla dowolnego zdarzenia A zachodzi

$$P(A) = \sum_{i=1}^{n} P(A|H_i)P(H_i).$$

Mając prawdopodobieństwo $P(H_j|A)$ można wyprowadzić wzór Bayesa. Jeśli $\{H_i\}_{i \in I}$ jest przeliczalnym rozbiciem $\Omega$ na zdarzenia o dodatnich prawdopodobieństwach oraz $P(A) > 0$, to dla dowolnego $j \in I$ mamy:

$$P(H_j|A) = \frac{P(A|H_j)P(H_j)}{\sum_{i \in I} P(A|H_i)P(H_i)}.$$

Omawiany przypadek jest przypadkiem szczególnym, gdzie mamy do czynienia z jednoelementową rodziną zdarzeń.

skutkuje ostateczną definicją modelu kanału z szumem [33, s. 43]:

$$\hat{w} = \underset{w \in V}{\operatorname{argmax}} P(O|w)P(w) \tag{5.3}$$

Wracając do intuicji stojącej za modelem można powiedzieć, że pewne słowo $w$ jest przez kanał z szumem zniekształcone do postaci $O$. $P(O|w)$ to część modelująca prawdopodobieństwo błędu (patrz podrozdział 5.2), natomiast $P(w)$ to model językowy (patrz podrozdział 5.3) operujący na rozkładach możliwych słów $w_i$, z których $O$ mogło powstać.

## 5.2. Model błędu

W nawiązaniu do równania 5.3, jego pierwszą część, czyli $P(O|w)$, można określić mianem modelu błędu (ang. *error model, channel model*). Część ta odpowiedzialna jest za modelowanie prawdopodobieństwa obserwacji pod warunkiem poprawnego słowa. Gdybyśmy chcieli w pełni modelować to zjawisko, należałoby się odwołać do wszelkich okoliczności procesu zapisywania błędnego wyrazu. Przykładowo wpływ na niepoprawność zapisu wyrazu miałoby to, czy nadawca jest prawo- czy leworęczny, jakim narzędziem zapisuje komunikat, czy wreszcie, czy posiada dysleksję. Oczywiście lista czynników, które mogłyby wpływać na poprawność komunikatu jest zbyt długa, dlatego w praktyce stosuje się ograniczenie listy czynników do poszczególnych liter obu analizowanych słów (tzn. obserwacji, czyli błędnego słowa oraz poprawnego słowa). Model błędu szacowany jest na podstawie tzw. macierzy współwystępowania po sobie liter. W zgodzie z ideą, że wyróżniamy cztery podstawowe typy operacji (które zostały opisane w podrozdziałach 4.2.5 oraz 4.2.6) na potrzeby szacowania modelu z błędem tworzone są cztery macierze – jedna dla każdej z operacji. Dodatkowo, w celu uwspólnienia mianownika[5] stosuje się macierz znaków, która dla dwugramów znakowych określa częstotliwość danej litery występującej przed następującą po niej literą. We wzorze 5.4 $O$ oznacza obserwację (słowo błędne), $w$ słowo-kandydata, natomiast $p$ to indeks pojedynczego znaku w słowie[6]. Szacowanie prawdopodobieństwa obserwacji pod warunkiem słowa-kandydata na podstawie częstości występowania liter w dwugramach znakowych zachodzi na podstawie równania 5.4 [59, s. 206].

$$P(O|w) \approx \begin{cases} \frac{del[w_{p-1},\ w_p]}{count[w_{p-1},\ w_p]}, \text{ gdy usunięcie znaku} \\ \frac{add[w_{p-1},\ O_p]}{count[w_{p-1}]}, \text{ gdy wstawienie znaku} \\ \frac{sub[O_p,\ w_p]}{count[w_p]}, \text{ gdy zamiana znaku} \\ \frac{trans[w_p,\ w_{p+1}]}{count[w_p,\ w_{p+1}]}, \text{ gdy transpozycja znaków} \end{cases} \tag{5.4}$$

[5] Jest to swego rodzaju normalizacja. Załóżmy, że litery $a$ i $b$ są mylone (trzeba je przestawić) 2000 razy, natomiast występują obok siebie 2500 razy. Mamy także parę $c$ i $d$, którą trzeba zamienić 21 razy, a wystąpiły obok siebie 25 razy. W przypadku modelu błędu korzystającego jedynie z licznika, preferowane są (model zwraca większe prawdopodobieństwo) dla kandydatów, którzy są transformacją obserwacji przy użyciu pary $a$, $b$. Gdy weźmiemy pod uwagę mianownik (liczbę razy gdy litery wystąpiły obok siebie), preferowana jest para $c$, $d$.

[6] Indeksy oznaczone tak samo jak w artykule, w którym zaproponowano model kanału z szumem [59].

## 5.3. Model językowy

Wychodząc od tak zwanego zadania Shannona (ang. *Shannon's task*) [36, s. 50] polegającego na przewidzeniu kolejnej litery mając $n$ poprzednich liter, możemy mówić o modelowaniu języka w kategoriach określenia prawdopodobieństwa wystąpienia słowa w sytuacji, gdy znamy $n$ poprzednich słów. Kolokacja (łac. *col-locatio*, czyli umieszczenie razem) to związek łączliwych semantycznie wyrazów. W badaniach [38, s. 2-3] zwraca się uwagę na trzy cechy kolokacji:

1. brak kompozycyjności (ang. *non-compositionality*), który przejawia się w braku jednoznacznego wynikania znaczenia frazy ze znaczenia jej elementów składowych. Może to być skrajna niekompozycyjność jak w przypadku idiomów, np. *piąte koło u wozu* lub jedynie pewne odchylenie od sumy znaczeń poszczególnych elementów jak na przykład w przypadku *białego wina*, które rzeczywiście jest koloru żółtego.

2. niezastępowalność (ang. *non-substitutability*) polega na niemożności zastąpienia elementu kolokacji innym słowem, mimo że to słowo stanowi synonim analizowanego elementu. Rozważmy dwie kolokacje: *silny wiatr* oraz *mocna kawa*. Zgodnie z definicjami podanymi w słowniku [39], słowa *silny* – „odznaczający się intensywnością, nasileniem, silnie działający, intensywny" oraz *mocny* – „silnie działający; intensywny, skondensowany, znaczny" mają znaczeniową część wspólną. Nie można jednak powiedzieć o kawie, że jest silna, a o wietrze, że jest mocny.

3. niemodyfikowalność (ang. *non-modifiability*) informująca, że kolokacji nie można modyfikować przez wstawianie lub usuwanie ich elementów. Rozważmy przykład związku frazeologicznego *flaki z olejem* (coś wyjątkowo nudnego). Gdy dodamy element doprecyzowujący rodzaj oleju *roślinny*, uzyskana fraza *flaki z olejem roślinnym* zupełnie traci znaczenie przenośne i oznacza teraz potrawę na bazie mięsa i oleju roślinnego.

Zagadnienie modelowania językowego pojawia się w ramach prac nad optycznym rozpoznawaniem znaków (ang. *OCR - optical character recognition*), poprawiania błędów pisowni (ang. *spell checking*) oraz tłumaczenia maszynowego [37, s. 191].

Nawiązując do list frekwencyjnych[7] (patrz podrozdział 6.1.3), które dla wymienionych dalej fraz wskazują częstotliwość ich wystąpienia, można zauważyć, że pewne frazy (np. *mocna kawa* – 19, *piąte koło u wozu* – 78 oraz *silny wiatr* – 561) występują w języku częściej niż inne (np. odpowiednio *silna kawa* – 0, *szóste koło u wozu* – 0, *mocny wiatr* – 28). Obserwacja stanowi istotną część modelowania języka, ponieważ ogranicza zbiór potencjalnych słów, które mogą wystąpić po zaobserwowanym łańcuchu $n$-słów.

Zadaniem modelu językowego jest oszacowanie prawdopodobieństwa wystąpienia pewnego słowa $w$ w tekście pod warunkiem wystąpienia słów wcześniejszych (kontekstu) $h$, co można zapisać jako $P(w|h)$. Ogólnie rzecz ujmując prawdopodobieństwo może zostać obliczone na podstawie ilorazu zliczeń, gdzie w liczniku wystąpiło zliczenie całego wyrażenia[8] (składającego się z $h$ oraz $w$), a w mianowniku wystąpiło zliczenie wystąpień wyrażenia $h$.

Załóżmy, że istnieje ciąg słów $w$ powstały ze słów z leksykonu $W$. Ciąg $w$ składa się z $k$ słów $w_1, w_2, ..., w_k$. Przez $P(w)$ oznaczamy zdarzenie, że ciąg $w$ zaszedł (innymi słowy, że $w_1, w_2, ..., w_k$ występują w dokładnie takiej, a nie innej kolejności). Zgodnie z równaniem wzoru łańcuchowego[9]:

[7] Modele w postaci archiwów dostępne są pod adresem `http://zil.ipipan.waw.pl/NKJPNGrams`

[8] W ogólnym przypadku mowa o ciągu znaków (w tym spacji), który w ramach języka naturalnego może być nieznaczący. Ze względu na tematykę dalszych rozważań analizowany jest tu przypadek szczególny, czyli ciągi znaczące, a więc wyrażenia.

[9] Jeżeli zdarzenia $A_1, A_2, ..., A_k$ spełniają warunek

$$P(w) = P(w_1)P(w_2|w_1)P(w_3|w_1, w_2) \cdot ... \cdot P(w_k|w_1, ..., w_{k-1})$$

Zakładając, że przeciętny użytkownik języka używa 20000 unikalnych słów, liczbę parametrów modelu językowego do oszacowania na podstawie 2-, 3- oraz 4-gramów przedstawia tabela 5.1.

| Model językowy | Liczba parametrów |
|---|---|
| 2-gram | $3.9998 \cdot 10^8$ |
| 3-gram | $7,9996 \cdot 10^{12}$ |
| 4-gram | $1,59992 \cdot 10^{17}$ |

Tablica 5.1: Liczba parametrów koniecznych do oszacowania dla poszczególnych modeli n-gramowych

Celem zmniejszenia liczby parametrów koniecznych do oszacowania stosuje się założenie upraszczające (założenie Markowa [33, s. 98]) wskazujące na to, że rozważany stan (tutaj $w_n$) zależy jedynie od $n$ poprzednich stanów ($n < k$) [12, s. 193]. Innymi słowy użycie modelu n-gramowego w celu oszacowania prawdopodobieństwa wystąpienia danego słowa zakłada użycie $n-1$ słów poprzedzających jako kontekstu (szczególnym przypadkiem jest model 1-gramowy, w którym liczba elementów kontekstu jest równa 0, a więc korzysta się jedynie z samego słowa, którego prawdopodobieństwo jest szacowane).

W przypadku modelu 1-gramowego ustalono wartość domyślną, która używana jest w miejsce zera w przypadku prawdopodobieństwa słów nieznajdujących się w korpusie, na podstawie którego stworzony został model. Ze względu na zastosowane wygładzanie Laplace'a wartość domyślna została ustalona jako

$$\frac{1}{m + \sum_{i=1}^{m} c_i} \tag{5.5}$$

gdzie $m$ to liczba unikalnych słów korpusu, a $c_i$ to częstość (frekwencja) $i$-tego słowa. Prawdopodobieństwo każdego z elementów 1-gramowego modelu językowego zostało oszacowane przy pomocy następującego wzoru [33, s. 99]:

$$P(w_i) = \frac{c_i + 1}{m + \sum_{i=1}^{m} c_i}$$

W modelu 2-gramowym prawdopodobieństwa określone są za pomocą następującego wzoru:

$$P(w_n|w_{n-1}) = \frac{c(w_{n-1}w_n)}{c(w_{n-1})}$$

gdzie zapis $w_{n-1}w_n$ oznacza, że słowo $n-1$ wystąpiło przed słowem $n$-tym, natomiast $c$ oznacza sumę zliczeń 2-gramów oraz 1-gramów (odpowiednio w liczniku oraz mianowniku równania). W przypadku tego modelu nie było koniecznym definiowanie wartości domyślnej, ponieważ w

$$P(A_1 \cap A_2 \cap ... \cap A_k) > 0,$$

to wzór łańcuchowy można wyrazić następującym wzorem:

$$P(A_1 \cap A_2 \cap ... \cap A_k) = P(A_1)P(A_2|A_1) \cdot ... \cdot P(A_k|A_1 \cap A_2 \cap ... \cap A_{k-1}).$$

przypadku braku szukanego 2-gramu następowało odwołanie do modelu 1-gramowego. Oszacowane na zasadzie interpolacji[10] prawdopodobieństwo modelu 2-gramowego było równe [33, s. 114]:

$$\hat{P}(w_n|w_{n-1}) = \lambda * P(w_n|w_{n-1}) + (1 - \lambda) * P(w_n) \tag{5.6}$$

W dalszych rozważaniach używane są eksperymentalnie następujące wartości współczynnika $\lambda$: 0,2, 0,33, 0,5, 0,66, 0,75, 0,9, 0,95 oraz 1.

[10]Metoda interpolacji jest obok metody *backoff* rozwiązaniem problemu zerowych prawdopodobieństw dla niektórych n-gramów. Interpolacja zakłada, że prawdopodobieństwo każdego n-gramu może być oszacowane jako ważona suma jego samego oraz n-gramów niższego stopnia, np. dla 3-gramu należy zsumować prawdopodobieństwo 3-gramu, 2-gramu oraz 1-gramu. Proces ważenia składników następuje przy użyciu nieujemnych parametrów $\lambda_i$, które są wagami $i$-tych składników równania oraz sumują się do 1. Przypadek szczególny stanowi interpolacja 2-gramu, gdzie pojedynczy parametr lambda odpowiada za to, jak bardzo 2-gram jest bliski 1-gramowi, tzn. im mniejszy parametr lambda, tym 2-gramowi bliżej do 1-gramu. Różnicą pomiędzy interpolacją a metodą *backoff* jest to, że ta pierwsza wymusza użycie ważonej sumy nawet w przypadku niezerowego prawdopodobieństwa analizowanego n-gramu, natomiast druga odwołuje się do ważonej sumy jedynie w przypadku zerowego prawdopodobieństwa wystąpienia n-gramu. Jeżeli nie wystąpił 1-gram obie metody odwołują się do wartości domyślnej modelu opisanej równaniem 5.5.

# Rozdział 6

# Implementacja modelu kanału z szumem

Wychodząc od list frekwencyjnych stworzonych na podstawie Narodowego Korpusu Języka Polskiego, implementuję n-gramowy model językowy, jednocześnie poruszając kwestie jego estymacji oraz przechowywania. Dokonawszy podziału danych korpusu PlEWi na zestaw treningowy i testowy, szacuję niezbędne składniki modelu błędu. Finalnie, przez złączenie obu oszacowanych modeli tworzę model kanału z szumem, który następnie jest ewaluowany na zestawie testowym. Implementacja modelu kanału z szumem w języku Python (wraz z opisem i wykonaniem poszczególnych kroków) znajduje się na stronie `https://github.com/balkon16/noisy_channel_model_thesis`.

## 6.1. n-gramowy model językowy

Wychodząc od charakterystyki Narodowego Korpusu Języka Polskiego jako reprezentatywnego zbioru, na podstawie list frekwencyjnych 1- oraz 2-gramów budowany jest n-gramowy model języka polskiego. W podrozdziale zostały opisane kwestie dotyczące wstępnego czyszczenia danych językowych oraz efektywny sposób przechowywania finalnych modeli n-gramowych.

### 6.1.1. Korpus NKJP

Wcześniej wspomniana idea utworzenia korpusu językowego dla języka polskiego znalazła odzwierciedlenie w Narodowym Korpusie Języka Polskiego [42]. Nawiązując do definicji korpusu językowego podanej w podrozdziale 3.1, NKJP spełnia wymagania dotyczące reprezentatywności zawartej w nim próbki tekstów.

#### Reprezentatywność

W językoznawstwie korpusowym dokonuje się rozróżnienia na reprezentatywność (ang. *representativeness*) oraz zrównoważenie (ang. *balance*). Reprezentatywność to cecha, która ma zagwarantować, że obserwacja dokonana na podstawie analizy korpusowej może zostać uznana za twierdzenie o języku w ogólności [43]. Zwraca się także uwagę na korpus jako próbkę języka, która w pełni odzwierciedla zróżnicowanie analizowanego bytu [44, s. 243].

Zrównoważenie korpusu stanowi raczej przykład dobrych intencji twórców aniżeli falsyfikowalne twierdzenie o obecności pewnej cechy [46, s. 19], dlatego na potrzeby dalszych analiz skupiono się jedynie na kwestii reprezentatywności.

| Typ tekstu | Udział procentowy w korpusie |
|---|---|
| Publicystyka i krótkie wiadomości prasowe | 50,0 % |
| Literatura piękna | 16,0 % |
| Literatura faktu | 5,5 % |
| Typ informacyjno-poradnikowy | 5,5 % |
| Typ naukowo-dydaktyczny | 2,0 % |
| Inne teksty pisane | 3,0 % |
| Książka niebeletrystyczna nieklasyfikowana | 1,0 % |
| Teksty konwersacyjne, mówione medialne i quasi mówione razem | 10,0 % |
| Teksty internetowe statyczne i dynamiczne razem | 7,0 % |

Tablica 6.1: Reprezentacja gatunków w NKJP [42, s. 41]

Dobór tekstów to przedsięwzięcie, które może być uwarunkowane na dwa sposoby. Pierwszym z nich są kryteria wewnętrzne, na podstawie których do składu korpusu dopuszcza się jedynie teksty o wcześniej założonych cechach językowych (często jest to liczba wystąpień pewnych słów) [45]. Drugie podejście stanowi skupienie się na kryteriach zewnętrznych, w których znaczącą rolę odgrywa gatunek czy kanał przekazywania proponowanego tekstu [46, s. 16].

Przy pracy nad stworzeniem Narodowego Korpusu Języka Polskiego jego twórcy skupili się na kryteriach zewnętrznych, dodatkowo wprowadzając założenia do prac, które pozwoliły uwzględnić to, jakie kanały oraz gatunki językowe spotyka na co dzień statystyczny użytkownik języka polskiego [42, s. 35-41]. Wynikiem pracy jest 300-milionowy zrównoważony korpus języka polskiego (NKJP 300M), który charakteryzuje się budową opisaną w tabeli 6.1.

### Typologia tekstów w NKJP

W ramach prac nad Narodowym Korpusem Języka Polskiego każdy tekst składowy zaklasyfikowano do jednej z następujących kategorii [42, s. 22]:

1. literatura piękna
   (a) proza,
   (b) poezja,
   (c) dramat,
2. literatura faktu,
3. publicystyka i krótkie wiadomości prasowe,
4. typ naukowo-dydaktyczny,
5. typ informacyjno-poradnikowy,
6. książka niebeletrystyczna niesklasyfikowana,
7. inne teksty pisane
   (a) typ urzędowo-kancelaryjny,
   (b) teksty perswazyjne (ogłoszenia, reklamy, propaganda polityczna),
   (c) krótkie teksty instruktażowe
8. listy,

9. Internet
    (a) interaktywne strony WWW (fora, chaty, listy dyskusyjne itp.),
    (b) statyczne strony WWW,
10. teksty mówione konwersacyjne,
11. teksty mówione medialne,
12. teksty quasi-mówione.

**Kanał**

Kanał określa w jaki sposób tekst zostaje przekazany od nadawcy do odbiorcy. W omawianym korpusie wyróżnia się następujące kanały [42, s. 24]:

1. prasa
    (a) prasa - dziennik,
    (b) prasa - tygodnik,
    (c) prasa - miesięcznik,
    (d) prasa - inne
2. książka,
3. Internet,
4. mówiony,
5. ulotka,
6. napis,
7. rękopis.

### 6.1.2. Określenie wielkości kontekstu

Proces estymacji n-gramowego modelu języka rozpoczyna się od wyboru $n$, czyli liczby określającej długość ciągów słów, które będą brane pod uwagę przy liczeniu prawdopodobieństwa danego słowa. Bazując na wspomnianym wcześniej korpusie NKJP 300M dla języka polskiego, który cechuje się bogactwem form morfologicznych, można utworzyć modele 1- i 2-gramowe [47, s. 256]. W celu stworzenia dobrego modelu wyższego rzędu (tzn. model 3-gramowy i bardziej rozbudowane) należałoby posłużyć się większą ilością danych tekstowych, aby zapobiec problemowi rzadkości danych (ang. *data sparsity*), który w językach z bogatą morfologią jest skutkiem istnienia względnie dużego zbioru unikalnych form tego samego lematu [48, s. 90].

| Typ n-gramu | Liczba n-gramów | Liczba n-gramów po czyszczeniu | Waga plików (GB) | Przykład n-gramu |
|---|---|---|---|---|
| 1-gram | 5 364 398 | 2 260 310 | 0,1 | izolacyjnych |
| 2-gram | 75 395 184 | 12 068 407 | 1,9 | w moskwie. |
| 3-gram | 170 180 746 | n.d. | 5,0 | już mi pan |
| 4-gram | 217 586 930 | n.d. | 7,7 | by nie uwierzył, że |
| 5-gram | 232 439 967 | n.d. | 9,8 | komisja wnosi o odrzucenie tej |

Tablica 6.2: Charakterystyka plików zawierających n-gramy dla języka polskiego

| 1-gram pierwotnie | 1-gram po wyczyszczeniu | Wyjaśnienie |
|---|---|---|
| ■ | | Elementy listy zawierające znaki niealfanumeryczne zostały zignorowane. |
| się, | się | Znaki interpunkcyjne zostały usunięte. |
| 1 | | Elementy listy składające się wyłącznie z cyfr zostały zignorowane. |
| ? | | Elementy listy zawierające wyłącznie znaki interpunkcyjne nie zostały wzięte pod uwagę. |
| ul. | ul | W przypadku skrótów usunięto końcowe kropki. |
| k.p.k. | k.p.k. | Skróty zawierające więcej niż jedną kropkę potraktowano jako wyjątek i w ich przypadku nie doszło do usunięcia kropek. |
| (marfet@bigfoot.com) | | Elementy listy zawierające znaki alfanumeryczne oraz niealfanumeryczne nie zostały dopuszczone do dalszej analizy. |
| (jakość | jakość | Jeżeli wyraz był otoczony przez jeden lub więcej znaków niealfanumerycznych, to znaki te zostały usunięte, a wyraz trafił do etapu dalszych analiz. |

Tablica 6.3: Przykłady czyszczenia surowych elementów listy frekwencyjnej zawierającej 1-gramy. Pusta komórka oznacza pusty ciąg znaków.

### 6.1.3. Przygotowanie danych na podstawie list frekwencyjnych NKJP

Na podstawie zbalansowanego korpusu NKJP 300M utworzono 1-, 2-, 3-, 4- oraz 5-gramowe listy frekwencyjne dla języka polskiego, które są publicznie i za darmo dostępne[1]. Listy mają formę plików bez rozszerzenia i mogą być oglądane oraz edytowane w dowolnym edytorze tekstowym. Jedna linijka stanowi jedną obserwację unigramu, dwugramu itd. i ma postać:

*liczba wystąpień n-gram* .

W tabeli 6.2 znajduje się podsumowanie cech każdego z plików.

Mimo że dostępne są listy frekwencyjne aż z 5 elementami (potencjalne 5-gramy), do oszacowania modeli językowych wykorzystano listy 1-gramów oraz 2-gramów, co było podyktowane obserwacją dotyczącą jakości modeli tworzonych w oparciu o ograniczoną ilość danych tekstowych (patrz 6.1.2).

Wskazane pliki poza literami alfabetu łacińskiego rozszerzonego o znaki diakrytyczne i cyfry zawierają szereg innych znaków, z których zostały wyczyszczone. Używając wyrażeń regularnych (ang. *regular expressions*) przeprowadzono czyszczenie elementów list z niepożądanych znaków (patrz tabela 6.3).

W przypadku listy frekwencyjnej zawierającej 2-gramy zastosowano taką samą metodę eliminacji niepożądanych znaków jak w przypadku 1-gramów. Obrana strategia czyszczenia skutkowała w niektórych przypadkach unieważnieniem elementu listy jako 2-gramu (patrz tabela 6.4). Liczba unikalnych n-gramów w ostatecznej, wyczyszczonej wersji znacząco różni się od liczby surowych nieoczyszczonych elementów list frekwencyjnych (patrz trzecia kolumna

[1] http://zil.ipipan.waw.pl/NKJPNGrams

| 2-gram pierwotnie | Pośredni ciąg znaków | 2-gram ostateczny | Wyjaśnienie |
|---|---|---|---|
| - mówi | mówi | | Po usunięciu myślnika (znaku niealfanumerycznego) element listy zawierał jedno słowo, a więc nie mógł zostać uznany za 2-gram. |
| panie marszałku! | panie marszałku | panie marszałku | Usunięciu podlegały znaki interpunkcyjne. |
| ust. 1 | ust | | Po usunięciu cyfr i znaków interpunkcyjnych pozostało jedno słowo, które nie mogło być uznane za dwugram. |
| § 2 | | | Elementy listy zawierające wyłącznie znaki niealfanumeryczne oraz cyfry zostały zignorowane. |

Tablica 6.4: Przykłady czyszczenia surowych elementów listy frekwencyjnej zawierającej 2-gramy. Pusta komórka oznacza pusty ciąg znaków.

w tabeli 6.2).

### 6.1.4. Przechowywanie n-gramowej listy frekwencyjnej

Opisane wcześniej listy są posortowane malejąco według liczby wystąpień n-gramów (par słów w przypadku 2-gramów). Dla implementowanego modelu jest to podejście niepraktyczne, ponieważ koszt dostępu do często występujących n-gramów jest minimalizowany (są na samym początku przeszukiwanej listy), natomiast w przypadku wyrazów występujących rzadziej trzeba przejść proporcjonalnie (do rzadkości występowania) pozycji listy więcej, aby znaleźć n-gram i jego frekwencję.

Przed zaimplementowaniem modelu językowego opartego na n-gramach konieczna jest zmiana formy przechowywania 1- i 2-gramów. Jako że model został zaimplementowany w języku programowania Python, rozważania dotyczące struktury przechowywanych danych skupiają się na rozwiązaniach w nim zaimplementowanych.

Listy frekwencyjne będą przechowywane w tablicy asocjacyjnej[2] używającej funkcji haszującej[3] do przechowywania par klucz-wartość. W języku programowania Python taka struktura

[2]Tablica asocjacyjna (ang. *associative array*) to abstrakcyjna struktura danych opierającą się na przechowywaniu wartości pod polami oznaczonymi przez unikalne klucze [51, s. 403-404]. Zadaniem struktury jest utrzymanie dostępu do danych podczas wykonywania podstawowych operacji wyszukania klucza, usunięcia klucza (i odpowiadającej mu wartości) oraz dodania nowego klucza (i odpowiadającej mu wartości). Z podstawowych operacji można tworzyć operacje wyższego rzędu, takie jak przypisanie innej wartości do klucza czy nadanie określonym wartościom innych niż dotychczas kluczy. Elementem koniecznym dla opisywanej struktury danych jest także możliwość iterowania po parach klucz-wartość, tzn. przechodzenia po kolejnych (niekoniecznie zawsze w tej samej kolejności przy każdej instancji iteratora) parach.

Implementacja tablicy asocjacyjnej może przyjmować różną postać w zależności od liczby elementów, które tablica miałaby przechowywać. Dla małej liczby elementów można zastosować rozwiązanie, gdzie za klucz służy pozycja w tablicy, ponieważ liniowy koszt wyszukania $O(n)$ dla odpowiednio małych $n$ jest zaniedbywalnie mały. Mowa wtedy o adresowaniu bezpośrednim, gdzie element o kluczu $k$ zostaje umieszczony na pozycji o indeksie $k$. W przypadku większych wartości $n$ potrzebne jest bardziej efektywne kosztowo rozwiązanie.

[3]Innym podejściem do umieszczania elementów w tablicy jest funkcja haszująca $h$, która odwzorowuje zbiór kluczy $U$, do którego należy wspomniany wcześniej klucz $k$ w zbiór pozycji tablicy z haszowaniem [50, s. 254]:

$$h : U \to \{0, 1, ..., m-1\}$$

gdzie rozmiar $m$ tablicy z haszowaniem jest zazwyczaj znacznie mniejszy niż liczba elementów $U$.

Ideą stojąca za stosowaniem tablic z haszowaniem jest ograniczenie rozmiaru pamięci potrzebnej do przechowywania obiektów w pamięci oraz zmniejszenie czasu dostępu do nich. W kontekście tablicy z haszowaniem mówimy o kluczu $k$, że jest haszowany na pozycję $h(k)$, a o $h(k)$, że jest wartością haszującą klucza k. Zważywszy na fakt, że $|U| > m$ w $U$ znajdują się takie dwa klucze $k_1$ i $k_2$, że $h(k_1) = h(k_2)$. Przypadek, gdy funkcja haszująca dla dwóch różnych argumentów zwraca tą samą wartość nazywamy kolizją.

danych określona jest mianem słownika i średni czas dostępu do elementu wynosi w niej $O(1)$[4] (jest stały[5]), dlatego została ona wybrana jako sposób przechowywania n-gramowego modelu języka.

Traktując każdy wyczyszczony i akceptowalny ciąg znaków (patrz podrozdział 6.1.3) jako klucz słownika[6], stworzono dwie struktury słownikowe[7] przechowujące 1-gramy oraz 2-gramy w postaci słowników przedstawionych na rysunkach[8] 6.1 oraz 6.2.

### 6.1.5. Implementacja modeli językowych

Na potrzeby analizy skuteczności działania modelu kanału z szumem zaimplementowano 1- i 2-gramowy[9] model językowy na podstawie przetworzonych list frekwencyjnych n-gramów (patrz podrozdział 6.1.3).

Do oszacowania modelu językowego składającego się na analizowany model kanału z szu-

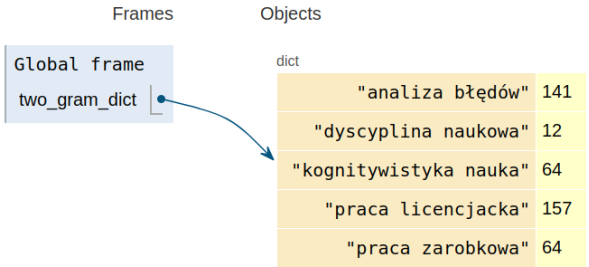
[4]Notacja $O$ używana jest do określenia asymptotycznego ograniczenia górnego funkcji. Dla danej funkcji $g(n)$ przez $O(g(n))$ oznaczamy zbiór funkcji [50, s. 47]:

$$\mathrm{O(g(n))} = \left\{ \mathrm{f(n)}\colon \exists c, n_0 \forall n \geqslant n_0 : 0 \leqslant f(n) \leqslant cg(n) \right\}$$

W rozważaniach dotyczących notacji $O$ abstrahuje się od konkretnej postaci danych wejściowej, skupiając się jedynie na ich rozmiarze $n$.

[5]`https://wiki.python.org/moin/TimeComplexity`

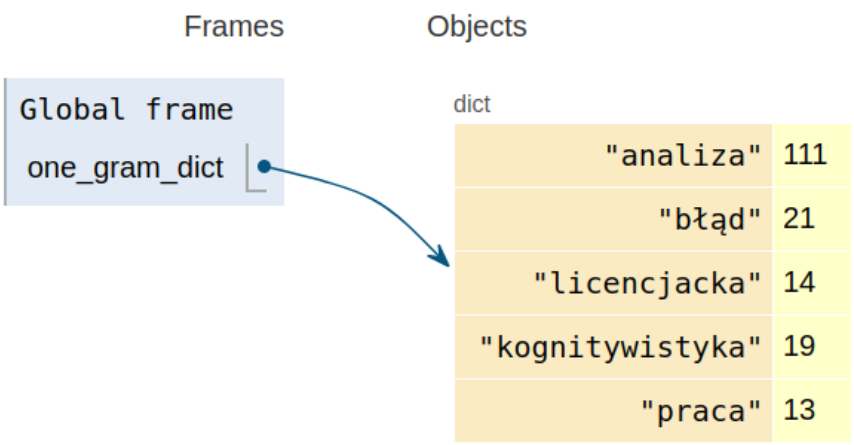
[6]W ostatecznej formie słowników przechowujących modele 1- i 2-gramowe klucz stanowi ciąg znaków bez spacji, a więc kluczem nie może być dwugram. Poniższy rysunek pokazuje przejściową postać zliczeń 2-gramów, które w następnym kroku zostały zapisane w formie podwójnych kluczy, których złączenie (z użyciem spacji) oznacza 2-gram.



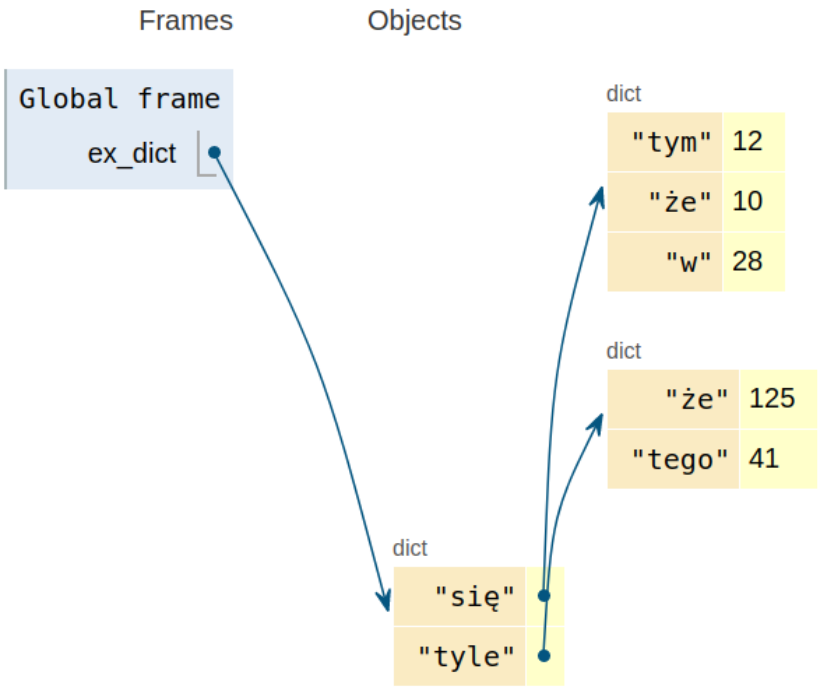

[7]Lista i krotka, które mogłyby stanowić struktury alternatywne w stosunku do wykorzystywanej, to typy sekwencyjne języka Python [49, s. 26]. Sekwencje stanowią uporządkowane zbiory obiektów indeksowanych przy użyciu nieujemnych liczb całkowitych. Lista (ang. *list*) to implementacja teoretycznej koncepcji tablicy (ang. *array*). Lista umożliwia przechowywanie elementów, które mogą być modyfikowane. W ramach języka Python lista to tablica i nie powinna być mylona ze strukturą danych zwaną listą, którą często określa się mianem listy z dowiązaniami (ang. *linked list*). W przeciwieństwie do kierunku wyznaczonego przez indeksy tablicy (listy w języku Python) kierunek w liście z dowiązaniami wyznaczają wskaźniki związane z każdym elementem listy [50, s. 234]. Krotka (ang. *tuple*) to struktura podobna do listy, ale taka, która nie umożliwia modyfikowania swoich elementów. Dla potrzeb przechowywania danego słowa i obliczonej wcześniej częstości jego występowania wydaje się być odpowiednia. Zgodnie z dokumentacją techniczą języka Python (dostępnej pod adresem `https://wiki.python.org/moin/TimeComplexity` znając indeks, czyli miejsce występowania w liście szukanego elementu, czas dostępu do niego jest słały $O(1)$. Problematyczne w przypadku korzystania z implementacji listy krotek jest fakt, że indeks szukanego słowa nie jest znany, co wymaga odwiedzenia średnio więcej niż jednego elementu listy. Ułatwieniem w przeszukiwaniu listy krotek jest alfabetyczne ustawienie pierwszych elementów krotek (zliczanych słów), co skutkuje tym, że algorytm wyszukania binarnego zwróci indeks szukanego słowa w czasie $O(\ln n)$ [50, s. 819-820].

[8]Rysunki zostały stworzone w oparciu o wizualizator kodu Python `http://pythontutor.com/`.

[9]Listy n-gramów wyższego rzędu nie zostały wykorzystane do obliczania modeli językowych, ponieważ zajmowałyby zbyt dużo pamięci RAM, w której na czas obliczeń trzymane były słowniki (tablice asocjacyjne) przechowujące listy n-gramów.




Rysunek 6.1: Model zliczeń 1-gramów jako słownik




Rysunek 6.2: Model 2-gramowy z podwójnym kluczem

mem potrzebne są prawdopodobieństwa, natomiast do tej pory wspomniane struktury słownikowe przechowują liczby naturalne świadczące o częstości wystąpienia 1-gramu czy 2-gramu. Prawdopodobieństwa wystąpienia 1-gramów i 2-gramów zostały oszacowane za pomocą funkcji zawartych w skryptach `create_n_gram_models_with_probs.py` oraz `bigram_language_model_multiprocessing.py` zgodnie z metodologią przedstawioną w podrozdziale 5.3).

## 6.2. Model błędu

### 6.2.1. Zestaw danych

#### Konwersja danych PlEWi do formatu DataFrame

Opisana w podrozdziale 3.4.1 struktura korpusu PlEWi, który ma formę plików tekstowych, gdzie informacje ułożone są linijka po linijce, sprawiała trudności zarówno w kontekście bycia argumentem wejściowym modelu kanału z szumem oraz dalszej ręcznej analizy, która pozwoliła na pominięcie niektórych elementów korpusu. W jednym z kroków wstępnego przetworzenia danych korpusu błędów zamieniono jego strukturę na widoczną na rysunku 6.3. Zamiana struktury polegała na przekształceniu linijek w pary klucz-wartość, a następnie wstawienie ich do tablicy asocjacyjnej (słownika).

```
{   'errors': [   {   'attributes': {   'category': 'semantyka',
                                        'distance': '1',
                                        'type': 'realword'},
                      'correction': 'bulteriera,',
                      'error': 'bullteriera,',
                      'position': '12'},
                  {   'attributes': {   'category': 'wielkość liter',
                                        'distance': '1',
                                        'type': 'realword'},
                      'correction': 'dr',
                      'error': 'Dr.',
                      'position': '24'},
                  {   'attributes': {   'category': 'wielkość liter',
                                        'distance': '2',
                                        'type': 'multiword'},
                      'correction': 'prof. dr',
                      'error': 'Prof. Dr.',
                      'position': '28'}],
    'new_text': "! 'Można by cytować liczne przykłady, że James Hinks wcale "
                'nie był wynalazcą bulteriera, a wiedzę tę zaczerpnęłam z '
                'książek takich autorytetów jak: Tom Homer, dr Dieter Fleig i '
                "prof. dr Hans Raber.'",
    'text': "! 'Można by cytować liczne przykłady, że James Hinks wcale nie "
            'był wynalazcą bullteriera, a wiedzę tę zaczerpnęłam z książek '
            'takich autorytetów jak: Tom Homer, Dr. Dieter Fleig i Prof. Dr. '
            "Hans Raber.'"}
```

Rysunek 6.3: Zmieniona w stosunku do pierwotnej struktura pojedynczego wpisu w korpusie błędów PlEWi

Od tego momentu każda z obserwacji (tzn. błąd oraz poprawiona wersja) mogła zostać wyodrębniona wraz z towarzyszącymi cechami, czyli zdaniem, w którym błąd wystąpił, kategorią błędu czy typem słowa, które uznano za błędne. Pokazana na rysunku 6.3 postać korpusu błędów została następnie przekształcona do ramki danych[10]. W ramce danych znalazło się 609 572 obserwacji.

[10]Ramka danych to występująca m.in. w języku programowania Python (konieczne jest zainstalowanie

Dystans edycji, składający się na cechy opisujące każdy z wpisów korpusu błędów (patrz podrozdział 3.4.1) był jednym z czynników decydujących o tym, czy obserwacja trafi do ostatecznego zestawu danych, na podstawie którego został wyestymowany model kanału z szumem. Jako że jednym z założeń modelu kanału z szumem jest dystans edycji równy 1 dzielący obserwację (błąd) a poprawną wersję tej obserwacji[11], z ramki danych wykluczono obserwacje o dystansie większym niż 1. Liczba obserwacji zmniejszyła się do 548 953. Kolejnym krokiem obróbki danych było stworzenie kolumny z informacją dotyczącą typu podstawowej operacji niezbędnej do przekształcenia błędu w poprawną wersję słowa. W ramce danych znalazła się także kolumna wskazująca na indeksy liter (gdzie napis jest traktowany jako ciąg znaków indeksowany liczbami naturalnymi, począwszy od zera), których należy użyć w procesie transformacji. Z dalszej analizy wykluczono obserwacje, których typ był inny niż *nonword* lub *realword*. Ostatecznie ramka danych zawierała 538 751 obserwacji.

W ostatecznym zbiorze (patrz rys. 6.4) znalazły się następujące zmienne (kolumny) opisujące każdą z obserwacji:

- `text_with_error` - tekst zawierający błąd,
- `corrected_text` - poprawiony tekst,
- `error` - obserwacja, słowo z błędem,
- `correction` - słowo poprawione, wolne od błędu,
- `type` - zmienna określająca, czy wartość z pola `error` została uznana za słowo (*realword*) czy nie-słowo (*nonword*),
- `category` - określona w danych źródłowych kategoria błędu,
- `file` - nazwa pliku tekstowego, w którym pierwotnie umieszczona była obserwacja,
- `basic_type_operation` - typ operacji niezbędnej do przeprowadzenia transformacji kandydata (słowa potencjalnie poprawnego, które generowane jest na potrzeby modelu - patrz podrozdział 6.3) w słowo z błędem,
- `nums` - para liczb określająca, które litery należy poddać transformacji, aby z kandydata otrzymać słowo z błędem.

### Podział danych na zbiór treningowy i testowy

Ograniczony do kategorii omawianych w rozdziale 3.4.2 zestaw danych potrzebny do stworzenia modelu kanału z szumem dla języka polskiego zawiera łącznie 538 751 obserwacji (wierszy). Przed stworzeniem modelu błędu (patrz rozdział 5.2) należy dokonać podziału danych

pakietu *pandas*) struktura danych, która ma postać dwuwymiarowej tabeli. Dla każdej obserwacji rezerwuje się jeden wiersz, natomiast w kolumnach przechowywane są wartości poszczególnych zmiennych.

[11] Założenie to nie jest *stricte* wymaganiem, lecz raczej zaleceniem wynikającym z liczby słów-kandydatów. Zakładając, że przy użyciu 32 miniskuł polskiego alfabetu chcemy ze słowa o długości $n$ wygenerować kandydatów odległych o dystans edycji równy 1 musimy liczyć się z $n$ możliwymi operacjami usunięcia, $n-1$ możliwymi operacjami przestawień znaków, $32(n+1)$ możliwymi operacjami wstawienia litery oraz $32n$ możliwymi operacjami zamiany liter. Dla dystansu edycji równego 1 słowo długości $n$ posiada $66n+31$ kandydatów, natomiast dla dystansu edycji równego 2 liczba kandydatów wzrasta do $4356n^2+4092n+961$. Dodatkowo obserwacja [55, s. 171], że 80% błędów popełnianych przez użytkowników może zostać zaklasyfikowana w ramach dystansu edycji równego 1, wskazywałaby na nieefektywność gwałtownego zwiększania liczby kandydatów w celu analizy jedynie 20% niewyjaśnionych błędów.

| | text_with_error | corrected_text | error | type | dist | category | file | basic_type_operation | nums | correction |
|---|---|---|---|---|---|---|---|---|---|---|
| **119** | Pod koniec tego samego roku przszło na świat d... | Pod koniec tego samego roku przyszło na świat ... | przszło | nonword | 1 | pisownia | plewic.01.0427.yaml | delete | (3, 3) | przyszło |
| **296** | Zna powiązania Eliasza z mesjanistycznymi ocze... | Zna powiązania Eliasza z mesjanistycznymi ocze... | wię, | nonword | 1 | znaki diakrytyczne | plewic.01.0040.yaml | replace | (2, 2) | wie, |
| **441** | Na początku cala okolica byla osobną wioską, s... | Na początku cała okolica była osobną wioską, s... | cala | realword | 1 | znaki diakrytyczne/kontekst | plewic.01.0026.yaml | replace | (2, 2) | cała |
| **440** | Była więc to działalnośc analogiczna do tej, k... | Była więc to działalność analogiczna do tej, k... | działalnośc | nonword | 1 | znaki diakrytyczne | plewic.04.0067.yaml | replace | (10, 10) | działalność |
| **26** | Kościół posiada nawę z dwoma podcieniami i cen... | Kościół posiada nawę z dwoma podcieniami i cen... | podwyzszonym | nonword | 1 | znaki diakrytyczne | plewic.01.0036.yaml | replace | (5, 5) | podwyższonym |

Rysunek 6.4: Pięć pierwszych wierszy ramki danych zawierających dane treningowe.

| | Kategoria | | | | | Typ | |
|---|---|---|---|---|---|---|---|
| | Pisownia | Znaki diakrytyczne | Znaki diakrytyczne/ kontekst | Składnia | Fleksja/liczby | nonword | realword |
| Zestaw treningowy | 54% | 34% | 7,3% | 2,4% | 2,3% | 88% | 12% |
| Zestaw testowy | 53,9% | 34,1% | 7,4% | 2,4% | 2,2% | 88% | 12% |

Tablica 6.5: Rozkład obserwacji według kategorii oraz typu w danych treningowych i testowych

na zestaw treningowy i testowy. W części eksperymentalnej omawianej pracy dokonano arbitralnego podziału danych: 75% obserwacji trafiło do zestawu treningowego, a 25% do zestawu testowego. Dodatkowo podział został dokonany w oparciu o losowanie warstwowe, które pozwoliło na zachowanie podobnego rozkładu obserwacji w obu zestawach ze względu na zmienne *category* oraz *type*. Dokładny rozkład pokazuje tabela 6.5. Dwie zmienne przedstawione w tabeli zostały wzięte pod uwagę, ponieważ różnicują one błędy na te łatwiejsze oraz trudniejsze. Przykładowo, za błąd łatwiejszy można uznać brak znaku diakrytycznego, co powoduje, że słowo jest nieznaczącym ciągiem znaków, natomiast za błąd trudniejszy do naprawienia przez model można uznać rzeczywiście istniejące słowo, jednak niewłaściwie odmienione w rozważanym kontekście.

### 6.2.2. Macierze znaków

Na potrzeby oszacowania prawdopodobieństwa obserwacji (błędnego słowa) pod warunkiem poprawnego słowa (korzystając ze wzoru 5.4) na podstawie informacji o otoczeniu błędu, błędzie, poprawnej wersji słowa z błędem, typie operacji oraz indeksach niezbędnych liter opracowano macierze błędów, korzystając jedynie z zestawu treningowego [59, s. 206]. Zestaw testowy nie został w tym kroku użyty celem sprawdzenia zdolności generalizowania modelu na wcześniej niewidziane pary liter umieszczone w macierzy. Stworzenie macierzy zostało poprzedzone wyczyszczeniem zarówno zdania zawierającego błąd, jak i samego błędu. Ponadto macierze służące do szacowania modelu błędu zawierają litery $q$, $v$ oraz $x$, które nie należą do polskiego alfabetu, co było podyktowane niektórymi błędami zawierającymi właśnie te litery.

Każda z macierzy odpowiada jednemu licznikowi z czterech podrównań równania 5.4. Macierz oznaczona jako *add* (tablica 6.6) zawiera informację, ile razy po literze z wiersza należało wstawić literę z kolumny (operacja *insert*), aby uzyskać zaobserwowane słowo z błędem, mając do dyspozycji pewne słowo-kandydata. Macierz *del* (tablica 6.7) odwołuje się do sytuacji, w której niezbędne było usunięcie litery w kolumnie, gdy ta wystąpiła po literze z wiersza (operacja *delete*). Macierz *sub* (tablica 6.8) wskazuje frekwencję operacji polegającej na zamianie litery z wiersza na literę z kolumny celem ukończenia transformacji kandydata na błąd (operacja *replace*). Macierz *trans* (tablica 6.9) informuje, ile razy doszło do transpozycji pary liter (litera wierszowa, litera kolumnowa) na parę liter (litera kolumnowa, litera wierszowa), co opisane jest operacją *transpose*. Macierz *count* (tablica 6.10) występująca w mianownikach podrównań to macierz zliczeń pierwszego i drugiego elementu dwugramów znakowych, które wystąpiły w wartościach zmiennej `text_with_error` zestawu treningowego.

W macierzach 6.6 oraz 6.7 krzyżyk (#) oznacza, że dodawany lub usuwany znak (litera w kolumnie) był pierwszym w ciągu znaków (dlatego nie miał poprzednika oznaczonego przez litery w wierszach macierzy). We wszystkich omawianych macierzach występują także znaki spoza alfabetu polskiego, ponieważ w pojedynczych przypadkach występowały one w obserwacjach. Celem uniknięcia zerowych wartości (w szczególności mianowników w równaniu 5.4) użyto wygładzania Laplace'a polegającego na dodaniu 1 dla każdego elementu macierzy.

| | a | ą | b | c | ć | d | e | ę | f | g | h | i | j | k | l | ł | m | n | ń | o | ó | p | r | s | ś | t | u | w | y | z | ź | ż | q | x | v |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **a** | 645 | 7 | 33 | 463 | 3 | 67 | 48 | 8 | 14 | 20 | 25 | 79 | 183 | 147 | 147 | 29 | 88 | 358 | 6 | 71 | 8 | 32 | 121 | 220 | 4 | 131 | 39 | 160 | 30 | 164 | 4 | 7 | 1 | 1 | 1 |
| **ą** | 34 | 25 | 3 | 12 | 4 | 10 | 1 | 1 | 2 | 6 | 1 | 9 | 9 | 1 | 3 | 24 | 14 | 15 | 3 | 4 | 1 | 7 | 6 | 5 | 137 | 4 | 4 | 4 | 4 | 8 | 12 | 9 | 1 | 1 | 1 |
| **b** | 87 | 1 | 73 | 5 | 1 | 7 | 78 | 6 | 1 | 3 | 7 | 121 | 7 | 10 | 36 | 3 | 3 | 54 | 1 | 71 | 1 | 3 | 35 | 13 | 1 | 7 | 23 | 5 | 33 | 3 | 1 | 1 | 1 | 1 | 1 |
| **c** | 70 | 4 | 4 | 185 | 1 | 15 | 83 | 2 | 6 | 9 | 62 | 119 | 90 | 50 | 12 | 3 | 4 | 29 | 1 | 22 | 4 | 6 | 5 | 16 | 1 | 10 | 26 | 10 | 91 | 349 | 1 | 1 | 1 | 1 | 1 |
| **ć** | 3 | 2 | 2 | 19 | 4 | 2 | 1 | 1 | 2 | 1 | 1 | 10 | 1 | 1 | 4 | 1 | 2 | 5 | 1 | 4 | 1 | 1 | 1 | 6 | 2 | 1 | 1 | 3 | 1 | 6 | 2 | 2 | 1 | 1 | 1 |
| **d** | 161 | 1 | 18 | 32 | 3 | 277 | 285 | 1 | 26 | 10 | 2 | 77 | 8 | 53 | 19 | 8 | 19 | 241 | 1 | 88 | 3 | 21 | 33 | 47 | 2 | 27 | 24 | 36 | 93 | 590 | 1 | 4 | 1 | 1 | 1 |
| **e** | 150 | 5 | 22 | 185 | 3 | 107 | 343 | 7 | 15 | 24 | 27 | 65 | 633 | 80 | 60 | 5 | 135 | 194 | 3 | 74 | 1 | 17 | 300 | 199 | 2 | 77 | 41 | 278 | 20 | 130 | 2 | 18 | 1 | 1 | 1 |
| **ę** | 9 | 3 | 4 | 22 | 5 | 44 | 34 | 11 | 2 | 4 | 2 | 4 | 4 | 4 | 4 | 7 | 7 | 15 | 2 | 5 | 1 | 1 | 6 | 27 | 4 | 6 | 1 | 13 | 3 | 11 | 1 | 2 | 1 | 1 | 1 |
| **f** | 11 | 1 | 1 | 1 | 1 | 2 | 17 | 1 | 72 | 4 | 4 | 26 | 1 | 3 | 10 | 2 | 3 | 1 | 1 | 9 | 1 | 1 | 27 | 7 | 1 | 6 | 4 | 2 | 4 | 1 | 1 | 1 | 1 | 1 | 1 |
| **g** | 92 | 1 | 6 | 3 | 1 | 15 | 67 | 1 | 15 | 131 | 59 | 179 | 2 | 4 | 28 | 15 | 5 | 29 | 1 | 74 | 5 | 10 | 62 | 16 | 2 | 17 | 24 | 6 | 5 | 2 | 1 | 1 | 1 | 1 | 1 |
| **h** | 173 | 1 | 3 | 44 | 1 | 5 | 33 | 3 | 1 | 23 | 45 | 44 | 18 | 6 | 3 | 4 | 19 | 15 | 1 | 44 | 1 | 12 | 10 | 12 | 1 | 20 | 14 | 21 | 27 | 12 | 1 | 2 | 1 | 1 | 1 |
| **i** | 473 | 20 | 10 | 112 | 5 | 20 | 1242 | 5 | 6 | 12 | 9 | 4540 | 169 | 61 | 67 | 16 | 66 | 234 | 2 | 435 | 11 | 9 | 25 | 80 | 11 | 32 | 206 | 109 | 19 | 42 | 3 | 2 | 1 | 1 | 1 |
| **j** | 128 | 3 | 3 | 21 | 1 | 18 | 154 | 5 | 1 | 2 | 19 | 153 | 86 | 51 | 9 | 2 | 18 | 36 | 1 | 30 | 2 | 6 | 5 | 25 | 1 | 2 | 50 | 20 | 21 | 10 | 2 | 2 | 1 | 1 | 1 |
| **k** | 375 | 9 | 2 | 21 | 1 | 5 | 49 | 4 | 1 | 3 | 5 | 568 | 51 | 218 | 107 | 23 | 16 | 21 | 1 | 118 | 5 | 6 | 33 | 59 | 1 | 97 | 23 | 199 | 9 | 9 | 1 | 2 | 1 | 1 | 1 |
| **l** | 170 | 2 | 7 | 32 | 1 | 12 | 220 | 1 | 5 | 3 | 1 | 446 | 28 | 101 | 518 | 3 | 5 | 86 | 2 | 114 | 2 | 2 | 2 | 26 | 1 | 26 | 36 | 11 | 25 | 6 | 1 | 1 | 1 | 1 | 1 |
| **ł** | 151 | 14 | 13 | 12 | 1 | 6 | 99 | 2 | 4 | 6 | 2 | 8 | 8 | 14 | 41 | 39 | 15 | 10 | 1 | 81 | 23 | 18 | 4 | 15 | 2 | 15 | 39 | 39 | 51 | 17 | 1 | 1 | 1 | 1 | 1 |
| **m** | 210 | 3 | 3 | 20 | 3 | 6 | 70 | 3 | 3 | 4 | 8 | 341 | 20 | 27 | 11 | 7 | 357 | 337 | 1 | 103 | 1 | 15 | 7 | 20 | 3 | 12 | 43 | 29 | 54 | 8 | 1 | 2 | 1 | 1 | 1 |
| **n** | 385 | 3 | 61 | 72 | 1 | 58 | 300 | 1 | 4 | 29 | 18 | 1229 | 19 | 76 | 8 | 5 | 101 | 1086 | 1 | 121 | 2 | 9 | 13 | 78 | 2 | 110 | 104 | 27 | 230 | 18 | 2 | 1 | 1 | 1 | 1 |
| **ń** | 14 | 1 | 2 | 2 | 1 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | 1 | 1 | 2 | 16 | 7 | 1 | 1 | 1 | 1 | 32 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| **o** | 164 | 6 | 43 | 248 | 5 | 152 | 72 | 1 | 11 | 20 | 28 | 198 | 124 | 32 | 65 | 6 | 67 | 93 | 2 | 549 | 4 | 284 | 71 | 209 | 8 | 46 | 71 | 547 | 43 | 98 | 2 | 38 | 1 | 1 | 1 |
| **ó** | 2 | 2 | 1 | 3 | 1 | 3 | 7 | 1 | 2 | 1 | 2 | 3 | 1 | 4 | 5 | 17 | 1 | 4 | 1 | 27 | 20 | 6 | 4 | 1 | 2 | 12 | 2 | 36 | 1 | 7 | 2 | 4 | 1 | 1 | 1 |
| **p** | 96 | 1 | 2 | 9 | 1 | 8 | 59 | 1 | 1 | 3 | 10 | 132 | 3 | 7 | 19 | 14 | 7 | 38 | 2 | 315 | 11 | 266 | 96 | 12 | 1 | 34 | 19 | 13 | 55 | 9 | 1 | 1 | 1 | 1 | 1 |
| **r** | 384 | 3 | 7 | 68 | 2 | 38 | 314 | 2 | 16 | 47 | 15 | 106 | 7 | 57 | 19 | 3 | 18 | 44 | 1 | 160 | 8 | 18 | 363 | 149 | 1 | 269 | 39 | 43 | 156 | 314 | 1 | 1 | 1 | 1 | 1 |
| **s** | 213 | 3 | 5 | 139 | 1 | 61 | 69 | 3 | 4 | 2 | 34 | 86 | 27 | 104 | 20 | 16 | 11 | 38 | 1 | 100 | 2 | 37 | 31 | 719 | 1 | 228 | 40 | 66 | 123 | 395 | 2 | 1 | 1 | 1 | 1 |
| **ś** | 6 | 7 | 2 | 34 | 20 | 1 | 2 | 1 | 1 | 1 | 5 | 8 | 2 | 1 | 2 | 1 | 1 | 5 | 1 | 4 | 1 | 2 | 2 | 19 | 9 | 7 | 1 | 5 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| **t** | 501 | 4 | 4 | 15 | 2 | 7 | 248 | 4 | 2 | 8 | 155 | 97 | 6 | 65 | 17 | 4 | 13 | 48 | 1 | 147 | 7 | 19 | 304 | 51 | 1 | 372 | 37 | 92 | 294 | 23 | 1 | 1 | 1 | 1 | 1 |
| **u** | 64 | 1 | 12 | 66 | 3 | 14 | 38 | 2 | 1 | 5 | 4 | 62 | 30 | 47 | 32 | 207 | 24 | 70 | 1 | 17 | 5 | 12 | 33 | 53 | 2 | 47 | 305 | 25 | 52 | 29 | 1 | 8 | 1 | 1 | 1 |
| **w** | 743 | 1 | 12 | 33 | 1 | 33 | 197 | 3 | 1 | 5 | 6 | 384 | 11 | 40 | 14 | 8 | 28 | 299 | 1 | 151 | 2 | 13 | 17 | 93 | 2 | 19 | 33 | 261 | 127 | 18 | 1 | 4 | 1 | 1 | 1 |
| **y** | 83 | 2 | 18 | 245 | 2 | 31 | 33 | 1 | 7 | 5 | 17 | 22 | 254 | 26 | 26 | 11 | 55 | 47 | 3 | 23 | 2 | 8 | 21 | 186 | 4 | 134 | 84 | 71 | 178 | 67 | 4 | 3 | 1 | 1 | 1 |
| **z** | 489 | 4 | 16 | 52 | 1 | 38 | 365 | 3 | 1 | 9 | 5 | 186 | 30 | 68 | 10 | 9 | 41 | 364 | 1 | 103 | 3 | 6 | 46 | 97 | 11 | 55 | 30 | 98 | 220 | 296 | 1 | 5 | 1 | 1 | 1 |
| **ź** | 1 | 1 | 1 | 1 | 4 | 5 | 2 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 2 | 1 | 4 | 1 | 1 | 1 | 1 | 2 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 3 | 2 | 2 | 1 | 1 | 1 |
| **ż** | 41 | 6 | 6 | 4 | 1 | 6 | 88 | 1 | 1 | 1 | 1 | 6 | 1 | 7 | 3 | 2 | 4 | 57 | 1 | 14 | 1 | 1 | 3 | 3 | 6 | 3 | 6 | 11 | 27 | 22 | 4 | 12 | 1 | 1 | 1 |
| **q** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| **x** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| **v** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| **#** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Tablica 6.6: Macierz znaków: wstawienie nowego znaku do napisu

| | a | ą | b | c | ć | d | e | ę | f | g | h | i | j | k | l | ł | m | n | ń | o | ó | p | r | s | ś | t | u | w | y | z | ź | ż | q | x | v |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **a** | 151 | 1 | 37 | 445 | 46 | 224 | 14 | 1 | 20 | 50 | 19 | 39 | 493 | 364 | 488 | 212 | 134 | 985 | 40 | 56 | 1 | 106 | 636 | 561 | 29 | 524 | 133 | 373 | 1 | 210 | 13 | 24 | 1 | 1 | 1 |
| **ą** | 5 | 1 | 3 | 113 | 3 | 66 | 1 | 1 | 1 | 38 | 1 | 2 | 3 | 4 | 1 | 109 | 8 | 6 | 1 | 6 | 1 | 20 | 2 | 44 | 2 | 83 | 3 | 7 | 1 | 51 | 1 | 23 | 1 | 1 | 1 |
| **b** | 74 | 5 | 29 | 19 | 1 | 3 | 54 | 3 | 2 | 4 | 3 | 223 | 7 | 13 | 95 | 23 | 2 | 132 | 1 | 99 | 3 | 2 | 211 | 65 | 3 | 1 | 25 | 2 | 67 | 2 | 1 | 1 | 1 | 1 | 1 |
| **c** | 74 | 3 | 1 | 38 | 1 | 1 | 115 | 11 | 1 | 1 | 464 | 505 | 197 | 107 | 10 | 7 | 2 | 97 | 1 | 38 | 2 | 3 | 2 | 1 | 1 | 28 | 15 | 2 | 164 | 837 | 2 | 1 | 1 | 1 | 1 |
| **ć** | 1 | 1 | 4 | 7 | 1 | 4 | 1 | 1 | 6 | 1 | 1 | 3 | 1 | 1 | 1 | 1 | 3 | 4 | 1 | 8 | 1 | 9 | 1 | 6 | 1 | 2 | 4 | 8 | 1 | 6 | 1 | 1 | 1 | 1 | 1 |
| **d** | 291 | 10 | 23 | 52 | 1 | 840 | 168 | 6 | 1 | 18 | 10 | 59 | 3 | 71 | 23 | 55 | 13 | 448 | 1 | 399 | 2 | 28 | 208 | 151 | 1 | 31 | 47 | 85 | 176 | 713 | 17 | 8 | 1 | 1 | 1 |
| **e** | 153 | 1 | 39 | 628 | 5 | 544 | 62 | 1 | 15 | 175 | 10 | 54 | 604 | 247 | 241 | 29 | 295 | 864 | 38 | 188 | 1 | 134 | 1085 | 573 | 53 | 153 | 90 | 253 | 4 | 834 | 10 | 36 | 1 | 1 | 1 |
| **ę** | 3 | 1 | 25 | 41 | 73 | 72 | 1 | 1 | 1 | 13 | 3 | 2 | 1 | 85 | 5 | 5 | 2 | 2 | 1 | 6 | 1 | 155 | 2 | 69 | 64 | 42 | 1 | 3 | 1 | 20 | 1 | 9 | 1 | 1 | 1 |
| **f** | 17 | 2 | 1 | 2 | 1 | 1 | 30 | 1 | 24 | 3 | 1 | 119 | 2 | 2 | 24 | 1 | 1 | 2 | 1 | 14 | 1 | 1 | 94 | 2 | 1 | 13 | 2 | 1 | 4 | 1 | 1 | 1 | 1 | 1 | 1 |
| **g** | 115 | 2 | 2 | 1 | 1 | 63 | 53 | 5 | 5 | 26 | 60 | 278 | 1 | 2 | 66 | 125 | 3 | 53 | 1 | 239 | 20 | 2 | 296 | 10 | 1 | 2 | 32 | 11 | 1 | 11 | 1 | 1 | 1 | 1 | 1 |
| **h** | 198 | 2 | 2 | 56 | 1 | 3 | 65 | 4 | 1 | 2 | 15 | 93 | 1 | 9 | 5 | 22 | 13 | 73 | 1 | 246 | 4 | 10 | 51 | 11 | 2 | 33 | 8 | 58 | 31 | 7 | 1 | 2 | 1 | 1 | 1 |
| **i** | 1125 | 48 | 30 | 214 | 4 | 28 | 3463 | 92 | 5 | 20 | 5 | 5083 | 158 | 183 | 164 | 67 | 55 | 482 | 16 | 655 | 15 | 41 | 50 | 304 | 14 | 168 | 138 | 133 | 1 | 93 | 1 | 18 | 1 | 1 | 1 |
| **j** | 181 | 109 | 12 | 12 | 1 | 50 | 306 | 41 | 1 | 6 | 3 | 140 | 11 | 15 | 20 | 1 | 33 | 151 | 1 | 52 | 3 | 17 | 10 | 437 | 16 | 13 | 14 | 19 | 1 | 7 | 1 | 1 | 1 | 1 | 1 |
| **k** | 483 | 17 | 1 | 86 | 1 | 1 | 23 | 10 | 6 | 1 | 6 | 954 | 3 | 57 | 66 | 71 | 2 | 24 | 1 | 345 | 8 | 5 | 231 | 210 | 1 | 533 | 66 | 54 | 1 | 3 | 1 | 8 | 1 | 1 | 1 |
| **l** | 258 | 6 | 16 | 13 | 1 | 20 | 391 | 19 | 7 | 13 | 2 | 795 | 1 | 135 | 381 | 1 | 26 | 307 | 1 | 249 | 2 | 2 | 3 | 87 | 2 | 64 | 32 | 9 | 5 | 2 | 1 | 3 | 1 | 1 | 1 |
| **ł** | 306 | 12 | 14 | 13 | 1 | 3 | 34 | 9 | 2 | 16 | 1 | 3 | 1 | 97 | 2 | 22 | 16 | 22 | 1 | 187 | 13 | 36 | 1 | 11 | 1 | 117 | 73 | 9 | 184 | 13 | 1 | 6 | 1 | 1 | 1 |
| **m** | 292 | 4 | 48 | 11 | 1 | 3 | 145 | 4 | 7 | 1 | 2 | 602 | 5 | 21 | 4 | 20 | 46 | 387 | 1 | 129 | 6 | 253 | 5 | 26 | 2 | 19 | 58 | 17 | 92 | 6 | 1 | 2 | 1 | 1 | 1 |
| **n** | 752 | 30 | 3 | 150 | 1 | 221 | 296 | 17 | 14 | 112 | 1 | 2109 | 4 | 323 | 1 | 1 | 2 | 766 | 1 | 224 | 7 | 1 | 2 | 283 | 1 | 409 | 62 | 5 | 406 | 5 | 1 | 5 | 1 | 1 | 1 |
| **ń** | 1 | 1 | 1 | 31 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 222 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| **o** | 70 | 1 | 191 | 128 | 6 | 517 | 22 | 1 | 14 | 70 | 12 | 26 | 55 | 88 | 224 | 71 | 127 | 523 | 26 | 131 | 1 | 259 | 446 | 337 | 125 | 186 | 42 | 478 | 2 | 704 | 2 | 38 | 1 | 1 | 1 |
| **ó** | 1 | 1 | 34 | 5 | 1 | 100 | 1 | 1 | 1 | 12 | 1 | 1 | 58 | 3 | 39 | 150 | 1 | 1 | 1 | 1 | 1 | 1 | 825 | 13 | 1 | 76 | 1 | 956 | 1 | 5 | 10 | 15 | 1 | 1 | 1 |
| **p** | 247 | 2 | 3 | 21 | 1 | 1 | 100 | 11 | 2 | 2 | 11 | 288 | 1 | 7 | 86 | 23 | 1 | 35 | 1 | 719 | 26 | 54 | 483 | 19 | 1 | 16 | 41 | 3 | 38 | 1 | 1 | 1 | 1 | 1 | 1 |
| **r** | 681 | 7 | 19 | 132 | 5 | 114 | 542 | 9 | 10 | 42 | 10 | 222 | 1 | 47 | 21 | 9 | 39 | 124 | 1 | 531 | 42 | 39 | 175 | 421 | 3 | 406 | 55 | 695 | 769 | 1004 | 1 | 2 | 1 | 1 | 1 |
| **s** | 220 | 2 | 12 | 295 | 1 | 2 | 163 | 2 | 13 | 1 | 12 | 285 | 29 | 371 | 8 | 74 | 8 | 38 | 1 | 180 | 9 | 178 | 4 | 132 | 1 | 1491 | 32 | 91 | 121 | 901 | 1 | 1 | 1 | 1 | 1 |
| **ś** | 1 | 1 | 4 | 211 | 182 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 50 | 1 | 8 | 27 | 2 | 1 | 1 | 3 | 47 | 1 | 1 | 1 | 1 | 51 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| **t** | 1011 | 12 | 1 | 13 | 1 | 4 | 347 | 43 | 3 | 4 | 54 | 73 | 1 | 216 | 14 | 10 | 5 | 172 | 1 | 357 | 48 | 9 | 1248 | 29 | 1 | 136 | 133 | 543 | 791 | 13 | 1 | 1 | 1 | 1 | 1 |
| **u** | 64 | 1 | 62 | 64 | 1 | 84 | 21 | 1 | 4 | 64 | 1 | 16 | 79 | 100 | 53 | 234 | 62 | 280 | 2 | 40 | 1 | 36 | 136 | 159 | 4 | 113 | 84 | 36 | 1 | 45 | 1 | 13 | 1 | 1 | 1 |
| **w** | 1485 | 5 | 6 | 47 | 1 | 21 | 331 | 11 | 1 | 4 | 5 | 803 | 3 | 26 | 10 | 33 | 2 | 307 | 1 | 336 | 20 | 34 | 52 | 540 | 14 | 6 | 9 | 53 | 344 | 277 | 1 | 1 | 1 | 1 | 1 |
| **y** | 12 | 1 | 15 | 298 | 35 | 52 | 10 | 1 | 13 | 30 | 2 | 5 | 203 | 108 | 75 | 101 | 128 | 141 | 6 | 49 | 2 | 61 | 25 | 856 | 17 | 124 | 11 | 137 | 1 | 179 | 2 | 13 | 1 | 1 | 1 |
| **z** | 2074 | 59 | 82 | 71 | 1 | 166 | 1775 | 44 | 1 | 55 | 10 | 662 | 23 | 184 | 16 | 58 | 59 | 944 | 1 | 344 | 2 | 73 | 35 | 146 | 2 | 216 | 40 | 219 | 891 | 17 | 1 | 1 | 1 | 1 | 1 |
| **ź** | 1 | 1 | 4 | 3 | 3 | 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 8 | 1 | 3 | 1 | 1 | 9 | 16 | 1 | 1 | 1 | 4 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| **ż** | 71 | 76 | 24 | 6 | 1 | 37 | 109 | 7 | 1 | 1 | 1 | 2 | 4 | 11 | 10 | 1 | 2 | 105 | 1 | 125 | 1 | 2 | 7 | 23 | 1 | 1 | 5 | 3 | 97 | 1 | 1 | 1 | 1 | 1 | 1 |
| **q** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| **x** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| **v** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| **#** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Tablica 6.7: Macierz znaków: usunięcie znaku z napisu

| | a | ą | b | c | ć | d | e | ę | f | g | h | i | j | k | l | ł | m | n | ń | o | ó | p | r | s | ś | t | u | w | y | z | ź | ż | q | x | v |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **a** | 1 | 5360 | 6 | 13 | 2 | 2 | 4924 | 30 | 6 | 2 | 2 | 245 | 6 | 6 | 12 | 4 | 2 | 14 | 1 | 1641 | 18 | 42 | 3 | 265 | 1 | 3 | 338 | 48 | 650 | 211 | 1 | 5 | 1 | 1 | 1 |
| **ą** | 29262 | 1 | 1 | 4 | 9 | 2 | 51 | 730 | 1 | 1 | 1 | 5 | 3 | 1 | 1 | 7 | 1 | 1 | 2 | 455 | 15 | 1 | 1 | 2 | 31 | 1 | 4 | 4 | 47 | 1 | 1 | 43 | 1 | 1 | 1 |
| **b** | 3 | 1 | 1 | 17 | 1 | 111 | 1 | 1 | 3 | 66 | 12 | 1 | 1 | 14 | 9 | 2 | 40 | 309 | 1 | 2 | 1 | 174 | 6 | 5 | 1 | 5 | 1 | 83 | 1 | 3 | 1 | 1 | 1 | 1 | 1 |
| **c** | 4 | 1 | 9 | 1 | 1787 | 74 | 19 | 1 | 2 | 21 | 7 | 3 | 10 | 271 | 17 | 2 | 5 | 84 | 1 | 13 | 1 | 5 | 22 | 450 | 2 | 64 | 3 | 11 | 1 | 245 | 1 | 3 | 1 | 1 | 1 |
| **ć** | 2 | 2 | 1 | 7101 | 1 | 4 | 2 | 2 | 1 | 2 | 1 | 1 | 2 | 4 | 1 | 1 | 3 | 1 | 6 | 1 | 7 | 1 | 1 | 3 | 116 | 2 | 1 | 1 | 1 | 2 | 8 | 38 | 1 | 1 | 1 |
| **d** | 2 | 1 | 77 | 91 | 1 | 1 | 10 | 1 | 89 | 222 | 4 | 5 | 3 | 24 | 8 | 3 | 4 | 52 | 1 | 37 | 1 | 14 | 76 | 364 | 1 | 525 | 2 | 54 | 3 | 26 | 1 | 1 | 1 | 1 | 1 |
| **e** | 4445 | 18 | 3 | 23 | 2 | 10 | 1 | 3610 | 2 | 2 | 2 | 124 | 3 | 7 | 5 | 1 | 7 | 16 | 1 | 541 | 2 | 6 | 285 | 28 | 1 | 12 | 171 | 312 | 403 | 8 | 1 | 2 | 1 | 1 | 1 |
| **ę** | 25 | 623 | 1 | 2 | 4 | 1 | 45632 | 1 | 1 | 1 | 1 | 44 | 1 | 1 | 1 | 11 | 1 | 2 | 1 | 5 | 1 | 1 | 4 | 1 | 5 | 1 | 10 | 6 | 124 | 1 | 5 | 2 | 1 | 1 | 1 |
| **f** | 1 | 1 | 4 | 4 | 1 | 43 | 2 | 1 | 1 | 53 | 5 | 3 | 11 | 11 | 2 | 7 | 3 | 6 | 1 | 1 | 1 | 27 | 8 | 9 | 1 | 13 | 1 | 159 | 1 | 3 | 1 | 1 | 1 | 1 | 1 |
| **g** | 3 | 1 | 51 | 19 | 1 | 243 | 4 | 1 | 79 | 1 | 388 | 1 | 16 | 136 | 2 | 1 | 3 | 6 | 1 | 5 | 1 | 15 | 8 | 8 | 1 | 11 | 1 | 7 | 1 | 6 | 1 | 11 | 1 | 1 | 1 |
| **h** | 2 | 1 | 10 | 4 | 1 | 1 | 1 | 1 | 12 | 110 | 1 | 4 | 125 | 64 | 5 | 1 | 4 | 15 | 1 | 1 | 1 | 3 | 1 | 2 | 1 | 3 | 4 | 3 | 10 | 170 | 1 | 1 | 1 | 1 | 1 |
| **i** | 380 | 3 | 1 | 3 | 1 | 6 | 195 | 83 | 10 | 4 | 25 | 1 | 865 | 19 | 91 | 8 | 16 | 103 | 1 | 851 | 3 | 2 | 19 | 9 | 1 | 20 | 634 | 6 | 703 | 22 | 1 | 3 | 1 | 1 | 1 |
| **j** | 3 | 2 | 2 | 3 | 1 | 2 | 5 | 1 | 10 | 18 | 138 | 882 | 1 | 221 | 67 | 15 | 50 | 41 | 28 | 1 | 1 | 1 | 16 | 5 | 5 | 12 | 6 | 9 | 228 | 8 | 2 | 12 | 1 | 1 | 1 |
| **k** | 2 | 1 | 12 | 723 | 3 | 17 | 2 | 1 | 5 | 162 | 23 | 10 | 222 | 1 | 223 | 9 | 51 | 94 | 1 | 4 | 1 | 33 | 11 | 2 | 1 | 82 | 2 | 14 | 1 | 8 | 1 | 1 | 1 | 1 | 1 |
| **l** | 5 | 1 | 3 | 7 | 1 | 5 | 6 | 1 | 2 | 7 | 3 | 80 | 92 | 167 | 1 | 848 | 11 | 48 | 1 | 21 | 1 | 10 | 90 | 6 | 1 | 38 | 2 | 5 | 2 | 11 | 1 | 1 | 1 | 1 | 1 |
| **ł** | 8 | 15 | 6 | 4 | 1 | 2 | 3 | 2 | 6 | 1 | 2 | 13 | 22 | 12 | 14663 | 1 | 12 | 15 | 3 | 2 | 10 | 3 | 8 | 4 | 1 | 56 | 16 | 36 | 2 | 1 | 1 | 5 | 1 | 1 | 1 |
| **m** | 3 | 1 | 45 | 5 | 1 | 4 | 4 | 1 | 2 | 5 | 3 | 3 | 54 | 70 | 14 | 11 | 1 | 892 | 9 | 2 | 1 | 42 | 6 | 4 | 1 | 4 | 7 | 83 | 1 | 3 | 1 | 1 | 1 | 1 | 1 |
| **n** | 7 | 1 | 261 | 70 | 1 | 63 | 14 | 6 | 2 | 6 | 13 | 50 | 42 | 51 | 62 | 14 | 1096 | 1 | 1371 | 6 | 1 | 10 | 66 | 31 | 1 | 43 | 9 | 64 | 12 | 61 | 1 | 2 | 1 | 1 | 1 |
| **ń** | 1 | 2 | 2 | 1 | 33 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 33 | 1 | 1 | 1 | 1 | 3174 | 1 | 3 | 1 | 1 | 2 | 2 | 8 | 1 | 1 | 3 | 1 | 1 | 4 | 7 | 1 | 1 | 1 |
| **o** | 2085 | 201 | 2 | 3 | 1 | 22 | 706 | 7 | 2 | 4 | 1 | 936 | 3 | 6 | 9 | 4 | 5 | 7 | 1 | 1 | 1659 | 438 | 5 | 11 | 2 | 9 | 535 | 4 | 154 | 6 | 1 | 1 | 1 | 1 | 1 |
| **ó** | 39 | 13 | 1 | 2 | 2 | 1 | 9 | 6 | 1 | 1 | 1 | 21 | 1 | 1 | 1 | 3 | 1 | 1 | 2 | 9777 | 1 | 4 | 1 | 1 | 7 | 1 | 1595 | 1 | 2 | 1 | 2 | 3 | 1 | 1 | 1 |
| **p** | 2 | 1 | 424 | 2 | 1 | 12 | 3 | 1 | 8 | 6 | 1 | 1 | 5 | 28 | 20 | 3 | 44 | 7 | 1 | 143 | 1 | 1 | 22 | 4 | 1 | 24 | 1 | 17 | 3 | 6 | 1 | 1 | 1 | 1 | 1 |
| **r** | 8 | 1 | 9 | 20 | 1 | 59 | 166 | 1 | 29 | 16 | 2 | 14 | 14 | 17 | 237 | 15 | 10 | 57 | 1 | 8 | 1 | 23 | 1 | 220 | 1 | 700 | 5 | 105 | 4 | 37 | 1 | 1 | 1 | 1 | 1 |
| **s** | 141 | 1 | 9 | 433 | 1 | 229 | 20 | 1 | 10 | 4 | 2 | 5 | 9 | 10 | 6 | 2 | 2 | 34 | 1 | 4 | 1 | 4 | 593 | 1 | 1411 | 20 | 7 | 50 | 1 | 1047 | 1 | 7 | 1 | 1 | 1 |
| **ś** | 2 | 23 | 1 | 3 | 44 | 2 | 2 | 7 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 5 | 1 | 1 | 7 | 2 | 6 | 2 | 1 | 12258 | 1 | 1 | 1 | 1 | 1 | 12 | 37 | 37 | 1 | 1 | 1 |
| **t** | 2 | 1 | 9 | 28 | 4 | 384 | 4 | 1 | 27 | 9 | 6 | 23 | 13 | 95 | 27 | 12 | 7 | 48 | 1 | 8 | 1 | 25 | 617 | 15 | 1 | 1 | 2 | 26 | 228 | 11 | 1 | 1 | 1 | 1 | 1 |
| **u** | 508 | 11 | 2 | 3 | 1 | 1 | 106 | 3 | 1 | 2 | 3 | 608 | 3 | 3 | 2 | 69 | 2 | 15 | 1 | 572 | 1469 | 2 | 5 | 4 | 1 | 3 | 1 | 35 | 1337 | 6 | 1 | 1 | 1 | 1 | 1 |
| **w** | 15 | 1 | 29 | 12 | 1 | 45 | 153 | 1 | 149 | 2 | 3 | 1 | 7 | 13 | 9 | 47 | 66 | 74 | 1 | 4 | 1 | 11 | 69 | 23 | 1 | 17 | 14 | 1 | 1 | 81 | 1 | 21 | 1 | 1 | 1 |
| **y** | 998 | 20 | 1 | 2 | 1 | 1 | 654 | 170 | 1 | 2 | 7 | 1074 | 24 | 3 | 3 | 2 | 4 | 6 | 1 | 138 | 6 | 3 | 6 | 9 | 1 | 290 | 2018 | 6 | 1 | 138 | 1 | 1 | 1 | 1 | 1 |
| **z** | 83 | 1 | 3 | 274 | 2 | 32 | 5 | 1 | 1 | 4 | 292 | 17 | 11 | 5 | 2 | 1 | 6 | 43 | 2 | 3 | 1 | 3 | 34 | 869 | 5 | 10 | 8 | 26 | 104 | 1 | 875 | 702 | 1 | 1 | 1 |
| **ź** | 1 | 10 | 1 | 1 | 9 | 2 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 5 | 1 | 1 | 5 | 1 | 1 | 1 | 1 | 3 | 248 | 1 | 1 | 1 | 1 | 1381 | 1 | 3979 | 1 | 1 | 1 |
| **ż** | 3 | 24 | 1 | 4 | 14 | 1 | 1 | 3 | 1 | 6 | 1 | 9 | 10 | 2 | 1 | 8 | 2 | 2 | 4 | 1 | 1 | 3 | 3 | 7 | 27 | 2 | 1 | 3 | 2 | 15851 | 713 | 1 | 1 | 1 | 1 |
| **q** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| **x** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| **v** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Tablica 6.8: Macierz znaków: zamiana znaku na inny

| | a | ą | b | c | ć | d | e | ę | f | g | h | i | j | k | l | ł | m | n | ń | o | ó | p | r | s | ś | t | u | w | y | z | ź | ż | q | x | v |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **a** | 1 | 1 | 11 | 28 | 1 | 59 | 75 | 1 | 5 | 13 | 7 | 18 | 106 | 266 | 515 | 61 | 100 | 354 | 4 | 9 | 1 | 40 | 525 | 87 | 1 | 102 | 24 | 225 | 1 | 117 | 1 | 1 | 1 | 1 | 1 |
| **ą** | 1 | 1 | 1 | 2 | 1 | 2 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 22 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 4 | 1 | 1 | 1 | 1 | 1 | 3 | 1 | 1 | 1 |
| **b** | 41 | 1 | 1 | 3 | 1 | 6 | 7 | 1 | 1 | 1 | 1 | 13 | 2 | 1 | 22 | 4 | 1 | 3 | 1 | 22 | 2 | 8 | 29 | 15 | 1 | 1 | 11 | 2 | 4 | 1 | 1 | 1 | 1 | 1 | 1 |
| **c** | 15 | 2 | 1 | 1 | 1 | 1 | 24 | 1 | 1 | 1 | 75 | 66 | 81 | 22 | 2 | 1 | 1 | 10 | 1 | 9 | 1 | 1 | 2 | 8 | 1 | 4 | 5 | 1 | 22 | 143 | 1 | 1 | 1 | 1 | 1 |
| **ć** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| **d** | 58 | 1 | 16 | 2 | 1 | 1 | 19 | 1 | 1 | 25 | 1 | 6 | 3 | 4 | 23 | 3 | 3 | 134 | 1 | 85 | 1 | 10 | 53 | 6 | 1 | 5 | 21 | 14 | 41 | 83 | 1 | 3 | 1 | 1 | 1 |
| **e** | 26 | 1 | 5 | 44 | 1 | 38 | 1 | 1 | 1 | 25 | 5 | 26 | 212 | 123 | 220 | 17 | 63 | 218 | 1 | 35 | 1 | 92 | 230 | 41 | 1 | 44 | 47 | 133 | 2 | 37 | 1 | 4 | 1 | 1 | 1 |
| **ę** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 8 | 1 | 1 | 1 | 3 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 1 |
| **f** | 2 | 1 | 1 | 1 | 1 | 1 | 3 | 1 | 1 | 2 | 1 | 6 | 1 | 1 | 3 | 1 | 1 | 1 | 1 | 6 | 1 | 1 | 6 | 1 | 1 | 5 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| **g** | 78 | 1 | 1 | 1 | 1 | 25 | 10 | 1 | 1 | 1 | 1 | 9 | 1 | 2 | 19 | 3 | 1 | 16 | 1 | 57 | 3 | 1 | 70 | 1 | 1 | 1 | 11 | 1 | 1 | 11 | 1 | 3 | 1 | 1 | 1 |
| **h** | 31 | 1 | 1 | 3 | 1 | 2 | 7 | 1 | 1 | 1 | 1 | 2 | 2 | 1 | 2 | 1 | 1 | 6 | 1 | 7 | 1 | 1 | 10 | 1 | 1 | 24 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 |
| **i** | 533 | 2 | 8 | 63 | 1 | 20 | 949 | 2 | 3 | 3 | 1 | 1 | 2 | 46 | 155 | 1 | 33 | 94 | 1 | 69 | 1 | 14 | 39 | 127 | 1 | 17 | 28 | 75 | 1 | 57 | 1 | 2 | 1 | 1 | 1 |
| **j** | 80 | 2 | 1 | 14 | 1 | 15 | 67 | 2 | 1 | 1 | 1 | 26 | 1 | 1 | 6 | 1 | 2 | 8 | 1 | 6 | 1 | 2 | 2 | 299 | 2 | 1 | 6 | 4 | 1 | 2 | 1 | 1 | 1 | 1 | 1 |
| **k** | 260 | 2 | 1 | 187 | 1 | 1 | 4 | 1 | 2 | 1 | 4 | 22 | 1 | 1 | 74 | 3 | 1 | 10 | 1 | 33 | 2 | 2 | 37 | 192 | 1 | 91 | 17 | 27 | 1 | 1 | 1 | 6 | 1 | 1 | 1 |
| **l** | 414 | 1 | 8 | 4 | 1 | 12 | 133 | 3 | 7 | 5 | 16 | 106 | 2 | 23 | 1 | 1 | 5 | 31 | 1 | 75 | 1 | 3 | 38 | 139 | 1 | 10 | 41 | 7 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| **ł** | 119 | 6 | 3 | 1 | 1 | 1 | 7 | 2 | 1 | 4 | 1 | 1 | 1 | 4 | 1 | 1 | 1 | 2 | 1 | 14 | 14 | 3 | 1 | 2 | 1 | 10 | 4 | 1 | 8 | 2 | 1 | 2 | 1 | 1 | 1 |
| **m** | 205 | 1 | 22 | 1 | 1 | 1 | 54 | 1 | 1 | 1 | 1 | 25 | 1 | 13 | 5 | 1 | 1 | 8 | 1 | 26 | 1 | 36 | 2 | 14 | 1 | 1 | 15 | 6 | 4 | 1 | 1 | 1 | 1 | 1 | 1 |
| **n** | 508 | 2 | 1 | 33 | 1 | 138 | 247 | 1 | 10 | 23 | 12 | 121 | 1 | 61 | 2 | 1 | 1 | 1 | 1 | 43 | 1 | 1 | 5 | 53 | 1 | 85 | 4 | 4 | 39 | 23 | 1 | 4 | 1 | 1 | 1 |
| **ń** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 4 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| **o** | 8 | 1 | 19 | 11 | 1 | 153 | 18 | 1 | 7 | 36 | 5 | 28 | 3 | 55 | 68 | 3 | 22 | 39 | 1 | 1 | 1 | 88 | 279 | 149 | 1 | 44 | 9 | 261 | 5 | 45 | 1 | 4 | 1 | 1 | 1 |
| **ó** | 1 | 1 | 2 | 2 | 1 | 4 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 22 | 1 | 1 | 1 | 1 | 1 | 1 | 30 | 3 | 1 | 5 | 1 | 12 | 1 | 1 | 2 | 1 | 1 | 1 | 1 |
| **p** | 76 | 1 | 4 | 11 | 1 | 1 | 41 | 1 | 1 | 1 | 3 | 34 | 1 | 2 | 12 | 3 | 4 | 1 | 1 | 93 | 2 | 1 | 62 | 31 | 1 | 5 | 15 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 |
| **r** | 458 | 1 | 8 | 8 | 1 | 36 | 178 | 2 | 6 | 141 | 1 | 37 | 1 | 24 | 6 | 2 | 22 | 39 | 1 | 495 | 48 | 67 | 1 | 41 | 1 | 145 | 46 | 76 | 31 | 722 | 1 | 1 | 1 | 1 | 1 |
| **s** | 34 | 1 | 1 | 10 | 1 | 2 | 29 | 1 | 3 | 1 | 2 | 57 | 26 | 197 | 6 | 16 | 6 | 17 | 1 | 48 | 4 | 63 | 1 | 1 | 1 | 414 | 17 | 7 | 65 | 28 | 1 | 1 | 1 | 1 | 1 |
| **ś** | 1 | 1 | 1 | 2 | 10 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 5 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| **t** | 343 | 2 | 1 | 6 | 1 | 3 | 69 | 1 | 4 | 1 | 7 | 20 | 2 | 56 | 33 | 8 | 3 | 40 | 1 | 166 | 44 | 4 | 195 | 14 | 1 | 1 | 56 | 118 | 75 | 9 | 1 | 1 | 1 | 1 | 1 |
| **u** | 23 | 1 | 10 | 8 | 1 | 22 | 11 | 1 | 1 | 7 | 3 | 30 | 9 | 22 | 25 | 30 | 22 | 14 | 1 | 21 | 1 | 16 | 51 | 56 | 1 | 19 | 1 | 2 | 1 | 10 | 1 | 1 | 1 | 1 | 1 |
| **w** | 135 | 1 | 1 | 8 | 1 | 13 | 76 | 1 | 1 | 1 | 1 | 337 | 1 | 5 | 3 | 3 | 2 | 26 | 1 | 528 | 38 | 1 | 22 | 15 | 1 | 3 | 4 | 1 | 55 | 91 | 1 | 1 | 1 | 1 | 1 |
| **y** | 3 | 1 | 4 | 200 | 1 | 14 | 1 | 1 | 4 | 4 | 1 | 2 | 14 | 1 | 11 | 4 | 20 | 29 | 1 | 1 | 1 | 9 | 8 | 170 | 1 | 46 | 1 | 107 | 1 | 10 | 1 | 2 | 1 | 1 | 1 |
| **z** | 57 | 1 | 1 | 9 | 1 | 23 | 108 | 1 | 1 | 6 | 1 | 147 | 6 | 31 | 1 | 3 | 9 | 56 | 1 | 143 | 1 | 23 | 1 | 3 | 1 | 5 | 8 | 44 | 163 | 1 | 1 | 1 | 1 | 1 | 1 |
| **ź** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| **ż** | 2 | 2 | 1 | 2 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 |
| **q** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| **x** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| **v** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Tablica 6.9: Macierz znaków: transpozycja znaków w napisie

## 6.3. Przykład obliczenia modelu kanału z szumem

Dla zadanej obserwacji model generuje wszystkich kandydatów, tzn. słowa oddalone od obserwacji o odległość edycji równą 1, uwzględniając wszystkie wymienione do tej pory podstawowe operacje edycji. Pewna część zbioru kandydatów to nieznaczące ciągi słów, które są usuwane[12]. Tabele 6.11 oraz 6.12 pokazują wartości kluczowe dla oszacowania modelu błędu oraz modelu językowego, których iloczyn stanowi model kanału z szumem, dla trzech kandydatów na poprawną wersję błędu **powacający*. Kluczowe w oszacowaniu modelu błędu są rodzaj operacji oraz dwugram znakowy. Załóżmy, że rozważamy kandydata *powracający*, którego można przekształcić w zaobserwowany błąd **powacający*:

$$powracający \rightarrow powacający$$

Dwugram znakowy, na podstawie którego należy oszacować prawdopodobieństwo modelu błędu to *wr* (litera *r* jest usuwana po literze *w*). Korzystając z równania 5.4 oraz macierzy odpowiadającej operacji usunięcia znaku (*delete*, tabela 6.7) możliwe jest oszacowanie modelu błędu.

Warto zwrócić uwagę na oszacowania modeli językowych w wariancie 1-gramowym (tabela 6.11) oraz 2-gramowym (tabela 6.12), które w różnym stopniu uwypuklają różnice w prawdopodobieństwie wystąpienia w języku omawianego kandydata *powracający* w porównaniu z pozostałymi kandydatami (*poważający* oraz *powalający*). W modelu 2-gramowym, pod względem językowym, pozostali kandydaci jawią się jako o rząd wielkości mniej prawdopodobni niż omawiany kandydat. Model kanału z szumem poprawnie wskazuje na słowo *powracający* jako korektę niepoprawnego zdania

**Pięciobój zimowy (odmiana Pięcioboju nowoczesnego, sport pokazowy) Skeleton (sport powacający do programu IO po 20 latach).*

W ogólnym przypadku model kanału z szumem zwraca słowo, które otrzymało najwyższe prawdopodobieństwo bycia słowem zamierzonym przez użytkownika.

## 6.4. Omówienie wyników

### 6.4.1. Zwracane wartości

W przypadku 6,8% obserwacji zestawu testowego model kanału z szumem nie zwrócił wartości, co było spowodowane pustą listą kandydatów zwróconą przez funkcję ich generującą. Takiemu stanowi rzeczy można zaradzić rozszerzając wspomnianą w podrozdziale 6.3 listę poprawnych słów języka polskiego, co jednak wiązałoby się z manualną i kosztowną pracą. W dalszej części analizy działania modelu rozważane są dwa jego warianty, które różnią się zastosowanym modelem językowym. Jeden z wariantów stworzony został w oparciu o model 1-gramowy, natomiast drugi wariant korzystał z modelu 2-gramowego. Ze względu na wykorzystanie różnych wartości współczynnika $\lambda$ wariant 2-gramowy modelu błędu można podzielić na podwarianty różniące się wartością tego współczynnika (patrz równanie 5.6). Model kanału z szumem w wariancie 2-gramowym nie zwrócił wartości w dwóch przypadkach więcej niż model 1-gramowy, ponieważ słowo oznaczone jako błąd nie znajdowało się w błędnie zapisanym zdaniu, co uniemożliwiło zdefiniowanie kontekstu, w którym błąd wystąpił.

[12]Lista słów, które zostały uznane za znaczące ciągi znaków (prawdziwe słowa) została zaczerpnięta ze strony `https://sjp.pl/slownik/growy/`

### 6.4.2. Skuteczność modelu

Dla obserwacji, dla których model zwrócił wartości, zastosowano miarę skuteczności opierającą się na porównaniu zwróconej wartości oraz pochodzącej z korpusu błędów prawidłowej poprawki (tzw. *gold standard*). Należy zaznaczyć, że w dalszej analizie wyników nie została zastosowana żadna metoda dająca porównanie dla procentowego poziomu wyników (tzw. *baseline*). Taka sytuacja podyktowana jest charakterem modelu kanału z szumem, który automatycznie zwraca poprawkę dla błędu. Istniejące dla języka polskiego narzędzia, jak wskazano w podrozdziale 4.1 poświęconym metodom słownikowym, zapewniają jedynie półautomatyczny tryb działania, gdzie to końcowy użytkownik narzędzia korekty tekstu wybiera poprawkę dla błędu.

W ogólnym przypadku model kanału z szumem w wariancie 1-gramowym osiągnął skuteczność 86,8%, natomiast model w wariancie 2-gramowym wskazał poprawnie 88-89% koniecznych poprawek (szczegóły w tabeli 6.13). W większości przypadków model kanału z szumem oparty o model 2-gramowy radził sobie lepiej, z wyjątkiem parametru lambda równego jeden (jest to „czysty" model 2-gramowy, który nie odwołuje się do modelu 1-gramowego). Kolumny *1-gram lepszy* oraz *2-gram lepszy* wskazują na liczbę obserwacji, odpowiednio, gdzie model 1-gramowy wskazał poprawną korektę, a 2-gramowy błędną oraz model 2-gramowy wskazał poprawną korektę, a model 1-gramowy błędną. Dodatkowo analiza została rozszerzona o podział ze względu na typ (zmienna `type`), kategorię (zmienna `category`) oraz rodzaj niezbędnej operacji (zmienna `basic_type_operation`).

#### Typ błędu

Biorąc pod uwagę podział błędów na nieznaczące ciągi słów (wiersz *nonword*) oraz ciągi oznaczające słowa (wiersz *realword*), model kanału z szumem w wariancie 2-gramowym okazał się skuteczniejszy (z wyjątkiem modelu 2-gramowego z parametrem $\lambda$ równym 1). Najskuteczniejsza korekta błędów miała miejsce w przypadku hybrydy modelów 1-gramowego (waga 5%) oraz 2-gramowego (waga 95%), co pokazuje tabela 6.14. Zauważalna jest niższa skuteczność (o około 13-14 p.p.) w kategorii *realword* w porównaniu do kategorii *nonword* w przypadku wszystkich wariantów i podwariantów modeli. Taki stan rzeczy spowodowany jest faktem, że jak pokazuje tabela 6.15 błędy uznane za nie-słowa w przeważającej części należą do kategorii związanych ze znakami diakrytycznymi lub błędną pisownią. Przykładowo następujące błędy *czyyny** z kategorii pisownia czy *stala** z kategorii znaków diakrytycznych można uznać za łatwe dla modelu ze względu na ograniczony zbiór kandydatów.

#### Kategoria błędu

Jak zwrócono uwagę powyżej warianty modelu osiągają najlepsze wyniki dla stosunkowo łatwych do poprawienia błędów tzn. tych należących do kategorii związanych ze znakami diakrytycznymi. Nieco gorsze rezultaty osiągają dla kategorii *pisownia*, która obejmuje m.in. błędy *obronnę** zamiast *obronę* czy *poczynianiami** zamiast *poczynaniami*. W przypadku kategorii poświęconej fleksji oraz liczbie model radzi sobie słabiej ze względu na przynależność obserwacji tej kategorii do grupy rzeczywistych słów (patrz tabela 6.15). Przykładowo problematyczne dla modelu były *klinik** zamiast *kliniki* (model wskazał *klonik*) czy *analiz** zamiast *analizy* (gdzie model wskazał *analiza*). W porównaniu do dotychczas zaprezentowanych wyników model (bez względu na wariant) osiągnął niską skuteczność w kategorii błędów składniowych, która w całości należy do typu rzeczywistych słów, np. *palny** zamiast *plany*, gdzie model wskazał *palnę*. Różnica prawie 30 p.p. w tej kategorii pomiędzy modelem w wariancie 1-gramowym a 2-gramowym świadczy o konieczności używania kontekstu w celu lepszego

zrozumienia otoczenia błędu i jego skuteczniejszej naprawy (patrz tabela 6.16).

### Rodzaj operacji

Zauważono znaczące różnice w skuteczności modelu w kontekście podziału obserwacji na typ operacji wymaganej do przeprowadzenia transformacji z kandydata na błąd. Nawiązując do tabeli 6.17, część różnic można wyjaśnić przez nadreprezentację operacji typu *replace* (zamiana znaku na inny) w grupie względnie łatwych błędów należących do kategorii *nonword* (wg tabeli 6.18 operacja *replace* występuje w dużej mierze w kontekście znaków diakrytycznych). Przyczyną zróżnicowania skuteczności modelu w obrębie rodzaju operacji mogą być także macierze znaków, które niewystarczająco dobrze generalizują zestaw treningowy na zestaw testowy (w szczególności w przypadku operacji *delete*, czyli usunięcia znaku).

## 6.5. Podsumowanie

Cel pracy polegający na przeanalizowaniu zakresu, w jakim możliwe jest automatyczne wykrywanie i poprawianie błędów pisowni języka polskiego został osiągnięty. Powyższa praca stanowi pierwszą publicznie udostępnioną implementację modelu kanału z szumem dla języka polskiego. Nawiązując do osiągnięć modelu dla języka angielskiego, gdzie poprawianych jest 97,5% błędów [7], należy stwierdzić, że w przypadku języka polskiego model jest mniej skuteczny. Jedną z przyczyn może być znacząco rozbudowana struktura morfologiczna języka polskiego w porównaniu do języka angielskiego.

Zgodnie z hipotezą postawioną na początku pracy, a nawiązującej do tezy dystrybucyjnej o języku, model 2-gramowy wykazał się większą skutecznością niż model 1-gramowy pozbawiony obsługi kontekstu. Warto zwrócić uwagę na to, że to hybryda tych dwóch modeli osiągała najlepsze rezultaty.

Podział zbioru testowego na kategorie pozwolił zwrócić uwagę na niedoskonałości modelu językowego, które sprawiły, że model kanału z szumem osiągał słabe rezultaty w przypadku kategorii silnie opartych o kontekst, ale w rozumieniu uzgodnienia części zdania, czyli w przypadku kategorii *składnia*.

Co do modelu błędu można mieć zastrzeżenia w kontekście podziału danych testowych na typ niezbędnej operacji. Innymi słowy, wymagana jest dalsza analiza dotycząca macierzy wystąpień znaków przy operacji usunięcia.

Pomysłem na dalsze analizy pod kątem udoskonalania modelu kanału z szumem jest zaimplementowanie w macierzach współwystąpień znaków obserwacji dotyczącej tego, że niektóre litery mogą być ze sobą częściej mylone z powodu ich bliższego występowania na klawiaturze komputerowej. Inna obserwacja dotyczy zbioru-słów kandydatów, który mógłby zostać ograniczony na podstawie uzgodnień poszczególnych części mowy w zdaniu. Model kanału z szumem musiałby implementować system wykrywania części mowy (*ang. part-of-speech tagging*), który pozwoliłby na ograniczenie zbioru kandydatów do tych, którzy swoimi częściami mowy pasują do kontekstu. Ponadto, przedstawiona w pracy implementacja modelu kanału z szumem może stanowić punkt odniesienia (ang. *baseline*) w dalszych badaniach nad metodami automatycznej korekty tekstu dla języka polskiego.

| | a | ą | b | c | ć | d | e | ę | f | g | h | i | j | k | l | ł | m | n | ń | o | ó | p | r | s | ś | t | u | w | y | z | ź | ż | # | q | x | v |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **a** | 4301 | 164 | 36923 | 238956 | 28600 | 168914 | 4715 | 46 | 15474 | 42639 | 5076 | 15316 | 144470 | 159793 | 198175 | 212185 | 134747 | 428429 | 35223 | 4065 | 88 | 43172 | 271903 | table 147827 | 10827 | 177229 | 28683 | 129180 | 5326 | 106059 | 5576 | 31891 | 4084 | 311 | 983 | 4652 |
| **ą** | 78 | 31 | 1415 | 99575 | 3089 | 26585 | 44 | 5 | 16 | 9778 | 3 | 33 | 192 | 1241 | 163 | 8460 | 66 | 160 | 45 | 22 | 2 | 5011 | 60 | 6346 | 549 | 18515 | 9 | 575 | 122 | 16245 | 118 | 10583 | 561 | 2 | 2 | 7 |
| **b** | 51945 | 4426 | 1293 | 2662 | 12 | 676 | 41904 | 9111 | 318 | 55 | 341 | 76057 | 3209 | 4508 | 25704 | 3986 | 158 | 12659 | 2 | 46969 | 5134 | 323 | 50328 | 13257 | 3 | 315 | 38665 | 1939 | 83851 | 138 | 2 | 14 | 124 | 1 | 4 | 36 |
| **c** | 65043 | 12322 | 133 | 1878 | 23 | 210 | 118994 | 4279 | 42 | 165 | 412613 | 239022 | 110845 | 42176 | 1634 | 2028 | 2387 | 16878 | 6 | 44036 | 12827 | 202 | 1730 | 557 | 6 | 10283 | 16124 | 250 | 98020 | 410214 | 1 | 107 | 181 | 216 | 49 | 53 |
| **ć** | 95 | 12 | 171 | 95 | 12 | 489 | 38 | 2 | 189 | 1 | 14 | 1524 | 5 | 69 | 9 | 4 | 434 | 26 | 1 | 24 | 14 | 11 | 3 | 29 | 15 | 9 | 24 | 688 | 16 | 84 | 3 | 2 | 160 | 1 | 1 | 2 |
| **d** | 118475 | 7804 | 7593 | 20563 | 23 | 10024 | 81091 | 4197 | 331 | 2918 | 1179 | 29399 | 3480 | 21522 | 29776 | 22961 | 10190 | 104169 | 15 | 227210 | 9727 | 10480 | 43549 | 17626 | 197 | 3521 | 56473 | 25727 | 83876 | 231296 | 5157 | 3552 | 179 | 22 | 38 | 251 |
| **e** | 19039 | 163 | 24451 | 141450 | 3811 | 144367 | 5947 | 212 | 13903 | 220220 | 2389 | 12638 | 265265 | 119712 | 131917 | 19527 | 195138 | 258709 | 20924 | 16338 | 177 | 39985 | 296860 | 190653 | 33387 | 76161 | 12318 | 61770 | 4901 | 94103 | 3813 | 41611 | 2707 | 338 | 1948 | 2987 |
| **ę** | 18 | 21 | 6084 | 29657 | 2464 | 33762 | 53 | 16 | 9 | 8473 | 5 | 58 | 86 | 20933 | 1500 | 5727 | 112 | 272 | 73 | 62 | 8 | 20888 | 118 | 12906 | 14458 | 23692 | 54 | 283 | 48 | 9589 | 604 | 9280 | 529 | 2 | 5 | 6 |
| **f** | 18548 | 142 | 40 | 163 | 1 | 31 | 15742 | 197 | 2647 | 463 | 42 | 47982 | 94 | 171 | 4712 | 27 | 141 | 403 | 1 | 24127 | 478 | 30 | 11218 | 348 | 1 | 1846 | 8831 | 42 | 1256 | 33 | 1 | 3 | 60 | 2 | 7 | 11 |
| **g** | 60054 | 5162 | 258 | 125 | 2 | 18000 | 26433 | 3575 | 165 | 1114 | 3181 | 59358 | 147 | 217 | 19904 | 30050 | 6712 | 15945 | 8 | 220278 | 16472 | 193 | 71100 | 1416 | 12 | 897 | 20945 | 4132 | 749 | 1887 | 2 | 38 | 184 | 17 | 31 | 42 |
| **h** | 45686 | 604 | 366 | 4478 | 4 | 396 | 26640 | 1626 | 99 | 86 | 174 | 28198 | 133 | 282 | 2401 | 3111 | 2411 | 12867 | 2 | 61538 | 4326 | 160 | 12858 | 1038 | 328 | 4291 | 9165 | 4521 | 6276 | 64 | 1 | 71 | 1381 | 20 | 3 | 50 |
| **i** | 333764 | 61192 | 9916 | 189282 | 12819 | 24916 | 1005171 | 175542 | 4343 | 19830 | 965 | 52379 | 13330 | 82224 | 63487 | 50753 | 68375 | 168265 | 7829 | 100695 | 9960 | 13756 | 19505 | 123337 | 7820 | 56928 | 52920 | 35862 | 557 | 38351 | 822 | 11955 | 3141 | 398 | 1454 | 3656 |
| **j** | 121694 | 104693 | 3264 | 10599 | 21 | 13698 | 210872 | 29985 | 118 | 676 | 155 | 63263 | 244 | 3758 | 3112 | 206 | 11212 | 39919 | 12 | 21164 | 2903 | 2713 | 1881 | 57683 | 5331 | 862 | 20573 | 8278 | 118 | 414 | 14 | 330 | 1140 | 2 | 2 | 13 |
| **k** | 207808 | 22310 | 398 | 24573 | 9 | 127 | 10365 | 10720 | 419 | 550 | 925 | 300380 | 527 | 2457 | 25929 | 25539 | 4064 | 5741 | 95 | 259425 | 32416 | 546 | 79402 | 49152 | 330 | 114668 | 124857 | 12968 | 1448 | 986 | 19 | 12837 | 466 | 10 | 13 | 37 |
| **l** | 146494 | 9729 | 8771 | 7771 | 13 | 7398 | 178812 | 10431 | 3201 | 3243 | 1337 | 220688 | 361 | 36489 | 20264 | 84 | 9488 | 81760 | 61 | 84749 | 1716 | 1762 | 443 | 38289 | 58 | 11416 | 63027 | 3945 | 5064 | 738 | 19 | 793 | 369 | 30 | 16 | 1093 |
| **ł** | 163946 | 17944 | 1269 | 6829 | 29 | 1846 | 25478 | 5939 | 295 | 2074 | 32 | 264 | 35 | 13193 | 127 | 557 | 1471 | 17118 | 18 | 122297 | 17919 | 2752 | 363 | 1198 | 81 | 5231 | 44044 | 1109 | 66328 | 561 | 4 | 1625 | 134 | 3 | 1 | 1 |
| **m** | 127474 | 2355 | 10477 | 4470 | 15 | 749 | 67287 | 4732 | 1281 | 506 | 136 | 282338 | 290 | 5008 | 2101 | 3992 | 5456 | 20249 | 16 | 101012 | 12434 | 19985 | 1089 | 6942 | 415 | 1602 | 55615 | 890 | 28899 | 184 | 2 | 342 | 1259 | 11 | 7 | 21 |
| **n** | 471701 | 42144 | 2134 | 42842 | 27 | 48809 | 241716 | 13565 | 9123 | 30761 | 1840 | 746464 | 1292 | 50037 | 1440 | 110 | 899 | 47873 | 25 | 168857 | 11132 | 7043 | 4690 | 37153 | 92 | 89224 | 27544 | 3130 | 243183 | 3602 | 8 | 1427 | 669 | 161 | 106 | 432 |
| **ń** | 56 | 6 | 41 | 21578 | 61 | 24 | 9 | 2 | 1 | 6 | 1 | 72 | 6 | 561 | 1 | 1 | 24 | 35 | 7 | 8 | 19 | 1 | 1 | 44780 | 241 | 88 | 9 | 6 | 13 | 17 | 1 | 7 | 114 | 1 | 3 | 5 |
| **o** | 5615 | 140 | 108403 | 97909 | 2174 | 283978 | 5032 | 35 | 20991 | 60357 | 6108 | 10609 | 65600 | 123305 | 152442 | 56830 | 106260 | 276076 | 15637 | 7656 | 144 | 84664 | 235695 | 196400 | 100322 | 90162 | 9996 | 480096 | 2458 | 83369 | 1786 | 41827 | 1588 | 101 | 1001 | 3881 |
| **ó** | 90 | 17 | 10293 | 9773 | 60 | 21299 | 401 | 2 | 20 | 2871 | 11 | 34 | 8115 | 857 | 20328 | 27289 | 135 | 974 | 15 | 57 | 24 | 350 | 81168 | 2818 | 256 | 4732 | 10 | 164469 | 380 | 5650 | 6622 | 13688 | 14 | 7 | 174 | 14 |
| **p** | 90839 | 871 | 177 | 8999 | 8 | 507 | 49060 | 4503 | 206 | 84 | 2641 | 101511 | 175 | 1745 | 22868 | 18423 | 317 | 16526 | 45 | 380014 | 31001 | 2753 | 296028 | 10789 | 24 | 7966 | 38768 | 913 | 10525 | 797 | 17 | 11 | 99 | 7 | 205 | 23 |
| **r** | 353869 | 11391 | 9344 | 38316 | 1252 | 36751 | 172773 | 18341 | 2936 | 25071 | 921 | 65848 | 665 | 26023 | 9818 | 4505 | 35069 | 43866 | 48 | 328656 | 78705 | 9877 | 6627 | 55997 | 884 | 54353 | 80130 | 41728 | 135123 | 380743 | 15 | 1829 | 424 | 151 | 49 | 1276 |
| **s** | 71668 | 17581 | 2494 | 45871 | 758 | 779 | 47497 | 1564 | 3677 | 379 | 5435 | 219128 | 15946 | 226374 | 6816 | 39345 | 7742 | 16660 | 20 | 56468 | 12081 | 86111 | 3308 | 10301 | 28 | 477893 | 33702 | 21307 | 47153 | 240299 | 2 | 75 | 423 | 206 | 10 | 216 |
| **ś** | 88 | 32 | 346 | 92456 | 35358 | 88 | 67 | 7 | 5 | 5 | 4 | 648 | 16 | 374 | 19145 | 187 | 6817 | 18786 | 191 | 61 | 39 | 1502 | 14696 | 38 | 46 | 172 | 28 | 25095 | 43 | 39 | 2 | 7 | 30 | 1 | 3 | 10 |
| **t** | 303844 | 12454 | 587 | 3921 | 18 | 786 | 207284 | 27569 | 992 | 562 | 10876 | 24201 | 949 | 46783 | 8187 | 4773 | 3515 | 47073 | 24 | 204079 | 81094 | 2815 | 152707 | 5576 | 19 | 8678 | 89172 | 78346 | 181377 | 6953 | 2 | 9 | 405 | 13 | 33 | 184 |
| **u** | 11090 | 98 | 40058 | 42962 | 267 | 64070 | 5850 | 46 | 2909 | 26403 | 1221 | 4926 | 69535 | 42077 | 35228 | 14882 | 44143 | 52070 | 1235 | 3144 | 46 | 27208 | 76760 | 70758 | 2086 | 44255 | 889 | 10150 | 641 | 25493 | 692 | 30521 | 1674 | 77 | 818 | 626 |
| **w** | 322905 | 17604 | 783 | 28033 | 56 | 8033 | 118346 | 9655 | 50 | 1611 | 274 | 308206 | 646 | 11379 | 3633 | 21975 | 454 | 77716 | 17 | 157244 | 24036 | 8758 | 21336 | 91693 | 2214 | 4024 | 8891 | 1558 | 203450 | 17316 | 7 | 85 | 1013 | 44 | 8 | 6 |
| **y** | 3490 | 74 | 23891 | 245114 | 12494 | 31909 | 2532 | 49 | 6493 | 15585 | 818 | 1270 | 41791 | 68036 | 32757 | 61066 | 148634 | 73281 | 5467 | 2451 | 73 | 26655 | 14636 | 100641 | 7910 | 54829 | 908 | 69268 | 314 | 17946 | 716 | 12646 | 1887 | 8 | 92 | 99 |
| **z** | 305647 | 50533 | 23766 | 41188 | 48 | 25507 | 395924 | 36259 | 212 | 15875 | 362 | 178390 | 14952 | 45094 | 7798 | 20939 | 31307 | 165727 | 116 | 117671 | 5143 | 20682 | 11165 | 4290 | 117 | 26519 | 35524 | 62143 | 292067 | 1873 | 11 | 76 | 372 | 25 | 51 | 20 |
| **ź** | 123 | 7 | 1221 | 628 | 751 | 4334 | 107 | 7 | 2 | 10 | 3 | 251 | 4 | 40 | 678 | 4 | 1151 | 12060 | 131 | 47 | 24 | 3 | 2841 | 67 | 2 | 35 | 38 | 1914 | 96 | 16 | 3 | 9 | 13 | 1 | 3 | 1 |
| **ż** | 19990 | 6856 | 3020 | 2159 | 150 | 6600 | 69254 | 1963 | 1 | 83 | 10 | 517 | 26 | 4712 | 6300 | 220 | 243 | 31707 | 43 | 19338 | 1169 | 28 | 582 | 7122 | 12 | 47 | 4302 | 554 | 44352 | 61 | 11 | 17 | 34 | 1 | 1 | 1 |
| **#** | 426 | 4 | 500 | 499 | 8 | 697 | 198 | 4 | 206 | 473 | 138 | 399 | 389 | 847 | 294 | 58 | 625 | 765 | 1 | 860 | 6 | 1720 | 503 | 949 | 96 | 558 | 406 | 1564 | 7 | 895 | 25 | 77 | 5 | 3 | 11 | 27 |
| **q** | 95 | 1 | 13 | 10 | 3 | 9 | 23 | 1 | 3 | 1 | 3 | 71 | 4 | 9 | 6 | 11 | 9 | 17 | 3 | 18 | 3 | 6 | 52 | 26 | 2 | 6 | 2137 | 34 | 8 | 8 | 2 | 5 | 2 | 4 | 1 | 14 |
| **x** | 688 | 5 | 33 | 121 | 2 | 34 | 439 | 3 | 83 | 2 | 81 | 699 | 5 | 11 | 78 | 4 | 32 | 226 | 1 | 230 | 20 | 387 | 23 | 59 | 1 | 433 | 138 | 73 | 170 | 37 | 1 | 1 | 86 | 3 | 103 | 12 |
| **v** | 4689 | 66 | 10 | 72 | 1 | 55 | 8436 | 8 | 2 | 35 | 63 | 5943 | 9 | 30 | 88 | 4 | 29 | 105 | 1 | 2883 | 7 | 16 | 310 | 219 | 1 | 21 | 230 | 6 | 630 | 39 | 1 | 1 | 7 | 1 | 2 | 11 |

Tablica 6.10: Macierz znaków: współwystępowanie znaków w dwugramach znakowych

| **Słowo-kandydat** | **Operacja** | **Litera** | | **Model błędu** | | **Modele** | | |
|---|---|---|---|---|---|---|---|---|
| | | *wiersz* | *kolumna* | *licznik* | *mianownik* | *błędu* | *językowy* | *kanału* |
| powracający | delete | w | r | 52 | 21336 | 2437,195 | 1,588 | 3869,772 |
| poważający | replace | c | ż | 3 | 107 | 28037,383 | 0,013 | 352,383 |
| powalający | replace | c | l | 17 | 1634 | 10403,917 | 0,084 | 871,731 |

Tablica 6.11: Składniki szacowania modelu kanału z szumem w wariancie 1-gramowym. W celu zwiększenia czytelności tabel prawdopodobieństwa modelów błędu oraz językowego zostały pomnożone przez $10^6$, natomiast model kanału z szumem został przeskalowany o $10^{12}$.

| **Słowo-kandydat** | **Operacja** | **Litera** | | **Model błędu** | | **Model językowy (słowo)** | | **Modele** | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | *wiersz* | *kolumna* | *licznik* | *mianownik* | *poprzednie* | *następne* | *błędu* | *językowy* | *kanału* |
| powracający | delete | w | r | 52 | 21336 | sport | do | 2437,195 | 1,592 | 3880,745 |
| poważający | replace | c | ż | 3 | 107 | sport | do | 28037,383 | 0,000000166 | 0,002 |
| powalający | replace | c | l | 17 | 1634 | sport | do | 10403,917 | 0,000000025 | 0,001 |

Tablica 6.12: Składniki szacowania modelu kanału z szumem w wariancie 2-gramowym. W celu zwiększenia czytelności tabel prawdopodobieństwa modelów błędu oraz językowego zostały pomnożone przez $10^6$, natomiast model kanału z szumem został przeskalowany o $10^{12}$.

| lambda | skuteczność | 1-gram lepszy | 2-gram lepszy |
|---|---|---|---|
| 0,20 | 88,37% | 1727 | 3714 |
| 0,33 | 88,56% | 1763 | 3977 |
| 0,5 | 88,70% | 1794 | 4193 |
| 0,66 | 88,87% | 1819 | 4427 |
| 0,75 | 88,99% | 1825 | 4578 |
| 0,90 | 89,24% | 1837 | 4906 |
| 0,95 | **89,39%** | 1848 | 5108 |
| 1,0 | 74,37% | 21267 | 5677 |

Tablica 6.13: Skuteczność modelu kanału z szumem w wariancie 2-gramowym

| type | unigram | lambda | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | *0,20* | *0,33* | *0,5* | *0,66* | *0,75* | *0,90* | *0,95* | *1,0* |
| nonword | 88,64 | 90,04 | 90,20 | 90,34 | 90,48 | 90,59 | 90,79 | 90,93 | 77,98 |
| realword | 74,05 | 76,91 | 77,21 | 77,43 | 77,74 | 77,93 | 78,54 | 78,76 | 49,52 |

Tablica 6.14: Skuteczność modelu kanału z szumem w wariancie 1- i 2-gramowym z podziałem na typ błędu. Wartości w tabeli dotyczącą procentów, jednak znak procentu został usunięty celem zwiększenia czytelności tabeli.

| type | category | count |
|---|---|---|
| nonword | pisownia | 66639 |
| | znaki diakrytyczne | 42990 |
| realword | znaki diakrytyczne/kontekst | 9668 |
| | fleksja/liczba | 3569 |
| | składnia | 2664 |

Tablica 6.15: Zliczenie obserwacji zestawu testowego z podziałem na typ oraz kategorię błędu

| category | unigram | lambda | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | *0,20* | *0,33* | *0,5* | *0,66* | *0,75* | *0,90* | *0,95* | *1,0* |
| fleksja/liczba | 49,17 | 55,93 | 56,74 | 57,33 | 58,03 | 58,56 | 59,99 | 60,44 | 41,02 |
| pisownia | 82,21 | 84,66 | 84,95 | 85,21 | 85,49 | 85,67 | 86,04 | 86,30 | 77,26 |
| składnia | **35,21** | 53,30 | 55,22 | 56,72 | 58,41 | 59,35 | 62,65 | **64,15** | 55,78 |
| znaki diakrytyczne | 98,61 | 98,38 | 98,33 | 98,28 | 98,23 | 98,21 | 98,16 | 98,11 | 79,10 |
| znaki diakrytyczne/ kontekst | 93,94 | 91,17 | 90,83 | 90,56 | 90,34 | 90,20 | 89,76 | 89,55 | 50,93 |

Tablica 6.16: Skuteczność modelu kanału z szumem w wariancie 1- i 2-gramowym z podziałem na kategorię błędu. Wartości w tabeli dotyczącą procentów, jednak znak procentu został usunięty celem zwiększenia czytelności tabeli.

| basic_type_operation | unigram | lambda | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | *0,20* | *0,33* | *0,5* | *0,66* | *0,75* | *0,90* | *0,95* | *1,0* |
| delete | **75,08** | 79,47 | 79,90 | 80,35 | 80,83 | 81,11 | 81,81 | 82,26 | 72,31 |
| insert | 80,21 | 82,51 | 82,92 | 83,24 | 83,53 | 83,75 | 84,19 | 84,51 | 78,53 |
| replace | **93,22** | 93,37 | 93,39 | 93,37 | 93,37 | 93,39 | 93,40 | 93,38 | 73,09 |
| transpose | 87,83 | 89,97 | 90,21 | 90,35 | 90,56 | 90,65 | 91,06 | 91,29 | 87,35 |

Tablica 6.17: Skuteczność modelu kanału z szumem w wariancie 1- i 2-gramowym z podziałem na operację. Wartości w tabeli dotyczącą procentów, jednak znak procentu został usunięty celem zwiększenia czytelności tabeli.

| type | category | basic_type_operation | count |
|---|---|---|---|
| nonword | znaki diakrytyczne | replace | 42943 |
| | pisownia | delete | 28664 |
| | | replace | 16176 |
| | | insert | 14994 |
| realword | znaki diakrytyczne/kontekst | replace | 9553 |
| nonword | pisownia | transpose | 6805 |
| realword | fleksja/liczba | replace | 2380 |
| | składnia | delete | 1470 |
| | fleksja/liczba | delete | 676 |
| | składnia | replace | 653 |
| | fleksja/liczba | insert | 510 |
| | składnia | insert | 482 |
| | znaki diakrytyczne/kontekst | delete | 102 |
| | składnia | transpose | 59 |
| nonword | znaki diakrytyczne | delete | 45 |
| realword | znaki diakrytyczne/kontekst | insert | 13 |
| | fleksja/liczba | transpose | 3 |
| nonword | znaki diakrytyczne | insert | 2 |

Tablica 6.18: Zliczenie obserwacji zestawu testowego z podziałem na typ i kategorię błędu oraz rodzaj niezbędnej operacji

# Spis rysunków

# Spis tablic

# Bibliografia


[1] Golding, A. R., & Roth, D. (1999). A winnow-based approach to context-sensitive spelling correction. Machine learning, 34(1-3), 107-130.

[2] Turing, A. M. (2009). Computing machinery and intelligence. In Parsing the Turing Test (pp. 23-65). Springer, Dordrecht.

[3] Simon, H. (1990). Reason in human affairs. Stanford University Press.

[4] Wittgenstein, L. (1953). Philosophische Untersuchungen. Frankfurt am Main: Suhrkamp Verlag.

[5] Firth, J. R. (1957). A synopsis of linguistic theory 1930-55. , 1952-59, 1-32.

[6] Zellig S. Harris (1954) Distributional Structure, WORD, 10:2-3, 146-162.

[7] Brill, E., & Moore, R. C. (2000, October). An improved error model for noisy channel spelling correction. In Proceedings of the 38th Annual Meeting on Association for Computational Linguistics (pp. 286-293). Association for Computational Linguistics.

[8] Jarosz, M., & Adamski, M. (2001). Słownik wyrazów obcych. Wrocław: Europa.

[9] Markowski, A. (2017). Wielki słownik poprawnej polszczyzny. Poznań: Wiedza.

[10] Lívia, N. F., & Jenõ, B. (2006). From theoretical to pedagogical grammar: Reinterpreting the role of grammar in English language teaching. Pannon University, Veszprém.

[11] Eggins, S. (2010). An introduction to systemic functional linguistics. London: Continuum.

[12] Trask, R. L. (1999). Key concepts in language and linguistics. Psychology Press.

[13] Rozprawy Komisji Językowej, vol. 33, Państwowe Wydawnictwo Naukowe, 2006

[14] Gilquin, G., & De Cock, S. (2011). Errors and disfluencies in spoken corpora: Setting the scene. International Journal of Corpus Linguistics, 16(2), 141.

[15] `http://www.nkjp.pl/`

[16] Miłkowski, M. (2007). Automated building of error corpora of Polish. Corpus Linguistics, Computer Tools, and Applications–State of the Art. PALC, 631-639.

[17] Pisarek, W. (1978). Słownik języka niby-polskiego, czyli błędy językowe w prasie. Wrocław, Warszawa, Kraków, Gdańsk: Zakład Narodowy im. Ossolińskich.

[18] Sågvall Hein, A. (1998). A Chart-Based Framework for Grammar Checking Initial Studies. In NODALIDA'98 (Vol. 11, p. 13). Center for Sprogteknologi, Denmark.

[19] Ray, A., & Margaret, W. (Eds.). (2003). PISA Programme for international student assessment (PISA) PISA 2000 technical report: PISA 2000 technical report. oecd Publishing.

[20] Ellis, R., & Ellis, R. R. (1994). The study of second language acquisition. Oxford University.

[21] Zeng, H., Alhossaini, M. A., Ding, L., Fikes, R., & McGuinness, D. L. (2006). Computing trust from revision history. Stanford Univ Ca Knowledge Systems LAB.

[22] https://dumps.wikimedia.org/plwiki/

[23] Grundkiewicz, R. (2013, September). Automatic extraction of Polish language errors from text edition history. In International Conference on Text, Speech and Dialogue (pp. 129-136). Springer, Berlin, Heidelberg.

[24] https://pl.wikipedia.org/wiki/Wikipedia:Spo%C5%82eczno%C5%9B%C4%87_wikipedyst%C3%B3w

[25] Bušta, J., Hlaváčková, D., Jakubícek, M., & Pala, K. (2009). Classification of errors in text. RASLAN 2009 Recent Advances in Slavonic Natural Language Processing, 109.

[26] http://hunspell.sourceforge.net/

[27] Kukich, K. (1992). Techniques for automatically correcting words in text. Acm Computing Surveys (CSUR), 24(4), 377-439.

[28] Shaalan, K. F. (2005). Arabic GramCheck: A grammar checker for Arabic. Software: Practice and Experience, 35(7), 643-665.

[29] Chomsky, N. (2014). Aspects of the Theory of Syntax (Vol. 11). MIT press.

[30] https://www.archives.gov/research/census/soundex.html

[31] Saniei, A. (2011). Who is an ideal native speaker. International Proceedings of Economics Development and Research26, 74-78.

[32] Kołodziej, W. (2012). Analiza matematyczna. Warszawa: Wydawnictwo Naukowe PWN.

[33] Jurafsky, D., & Martin, J. H. (2014). Speech and language processing (Vol. 3). London: Pearson.

[34] Hamming, R. W. (1950). Error detecting and error correcting codes. The Bell system technical journal, 29(2), 147-160.

[35] Levenshtein, V. I. (1966, February). Binary codes capable of correcting deletions, insertions, and reversals. In Soviet physics doklady (Vol. 10, No. 8, pp. 707-710).

[36] Shannon, C. E. (1951). Prediction and entropy of printed English. Bell system technical journal, 30(1), 50-64.

[37] Manning, C. D., Manning, C. D., & Schütze, H. (1999). Foundations of statistical natural language processing. MIT press.

[38] Benson, M. (1989). The structure of the collocational dictionary. International Journal of Lexicography, 2(1), 1-14.

[39] Drabik, L., Kubiak-Sokół, A., Sobol, E., Wiśniakowska, L., & Stankiewicz, A. (2018). Słownik języka polskiego Pwn. Warszawa: Wydawnictwo Naukowe PWN SA.

[40] Jakubowski, J., & Sztencel, R. (2017). Rachunek prawdopodobieństwa dla (prawie) każdego. Warszawa: Script.

[41] Bergroth, L., Hakonen, H., & Raita, T. (2000). A survey of longest common subsequence algorithms. In Proceedings Seventh International Symposium on String Processing and Information Retrieval. SPIRE 2000 (pp. 39-48). IEEE.

[42] Przepiórkowski Adam. (2012). Narodowy Korpus Języka Polskiego. Warszawa: Wydawnictwo Naukowe PWN.

[43] Leech, G. (2014). The state of the art in corpus linguistics. In English corpus linguistics (pp. 20-41). Routledge.

[44] Biber, D. (1993). Representativeness in corpus design. Literary and linguistic computing, 8(4), 243-257.

[45] Otlogetswe, T. (2001). The BNC design as a Model for a Setswana language corpus. extraction, 1.

[46] McEnery, T., Xiao, R., & Tono, Y. (2006). Corpus-based language studies: An advanced resource book. Taylor & Francis.

[47] Ziółko Bartosz, & Ziółko Mariusz. (2011). Przetwarzanie mowy. Kraków: Wydawnictwa AGH.

[48] Whittaker, E. W., & Woodland, P. C. (2003). Language modelling for Russian and English using words and classes. Computer speech & language, 17(1), 87-104.

[49] Beazley, D. M. (2009). Python Essential Reference, Fourth Edition. Addison-Wesley Professional.

[50] Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C., Diks, K., Malinowski, A., Rytter, W. (2018). Wprowadzenie do algorytmów. Warszawa: PWN.

[51] Goodrich, M. T., Tamassia, R., & Goldwasser, M. H. (2014). Data structures & algorithms in java. John Wiley & Sons, Inc.

[52] Szypra-Kozłowska, J. (2002). Wprowadzenie do współczesnej fonologii. Lublin: Wydawnictwo Uniwersytetu Marii Curie-Skłodowskiej.

[53] Boytsov, L. (2011). Indexing methods for approximate dictionary searching: Comparative analysis. Journal of Experimental Algorithmics (JEA), 16, 1-1.

[54] Blair, C. R. (1960). A program for correcting spelling errors. Information and Control, 3(1), 60-67.

[55] Damerau, F. J. (1964). A technique for computer detection and correction of spelling errors. Communications of the ACM, 7(3), 171-176.

[56] Hirst, G., & Budanitsky, A. (2005). Correcting real-word spelling errors by restoring lexical cohesion. Natural Language Engineering, 11(1), 87-111.

[57] Mitton, R. (1987). Spelling checkers, spelling correctors and the misspellings of poor spellers. Information processing & management, 23(5), 495-505.

[58] Miłkowski, M. (2010). Developing an open-source, rule-based proofreading tool. Software: Practice and Experience, 40(7), 543-566.

[59] Kernighan, M. D., Church, K. W., & Gale, W. A. (1990, August). A spelling correction program based on a noisy channel model. In Proceedings of the 13th conference on Computational linguistics-Volume 2 (pp. 205-210). Association for Computational Linguistics.

[60] Trzęsicki K. (2004). Elementy logiki i teorii mnogości. Białystok: Wyższa Szkoła Finansów i Zarządzania.

[61] Waliszewski, W. (1978). Encyklopedia Szkolna. Warszawa: WSiP.