Korpusomat EU

Wydanie 0.1

IPI PAN

19 sie 2023

Spis treści

Korpusomat jest aplikacją webową służącą do tworzenia wielowarstwowo oznakowanych korpusów tekstów, z których można korzystać za pomocą wyszukiwarki MTAS. Do znakowania wykorzystywane są nowoczesne wielojęzyczne zestawy narzędzi programistycznych spaCy i Stanza. Zasadniczym celem Korpusomatu jest udostępnienie badaczom wyników działań tych narzędzi na dowolnych tekstach bez konieczności szczegółowej znajomości technicznej strony ich działania.

Korpusomat przetwarza pliki tekstowe (txt) oraz większość innych formatów służących do przechowywania danych tekstowych (np. epub, mobi, doc, rtf czy pdf – pełna lista możliwych formatów dostępna jest pod adresem http://tika.apache.org/1.17/formats.html). Narzędzia, z których korzysta, wymagają stosowania kodowania UTF-8, jeśli jednak użytkownik prześle plik w innym kodowaniu, np. ISO-8859-2 czy CP-1250 (w wypadku języka polskiego), Korpusomat automatycznie skonwertuje je do kodowania UTF-8 na swój wewnętrzny użytek.

Korpusomat pozwala również na dodawanie artykułów ze stron internetowych. W takim przypadku wskazana strona zostaje przetworzona za pomocą biblioteki newspaper, której opis dostępny jest pod adresem: https://newspaper.readthedocs.io/.

ROZDZIAŁ 1

Dokumentacja

1.1 Korzystanie z aplikacji

1.1.1 Tworzenie konta

Korzystanie z Korpusomatu należy rozpocząć od rejestracji, czyli założenia konta użytkownika, w ramach którego będzie można zarządzać tworzonymi korpusami. Do założenia konta wystarczy podanie adresu e-mail i hasła użytkownika.

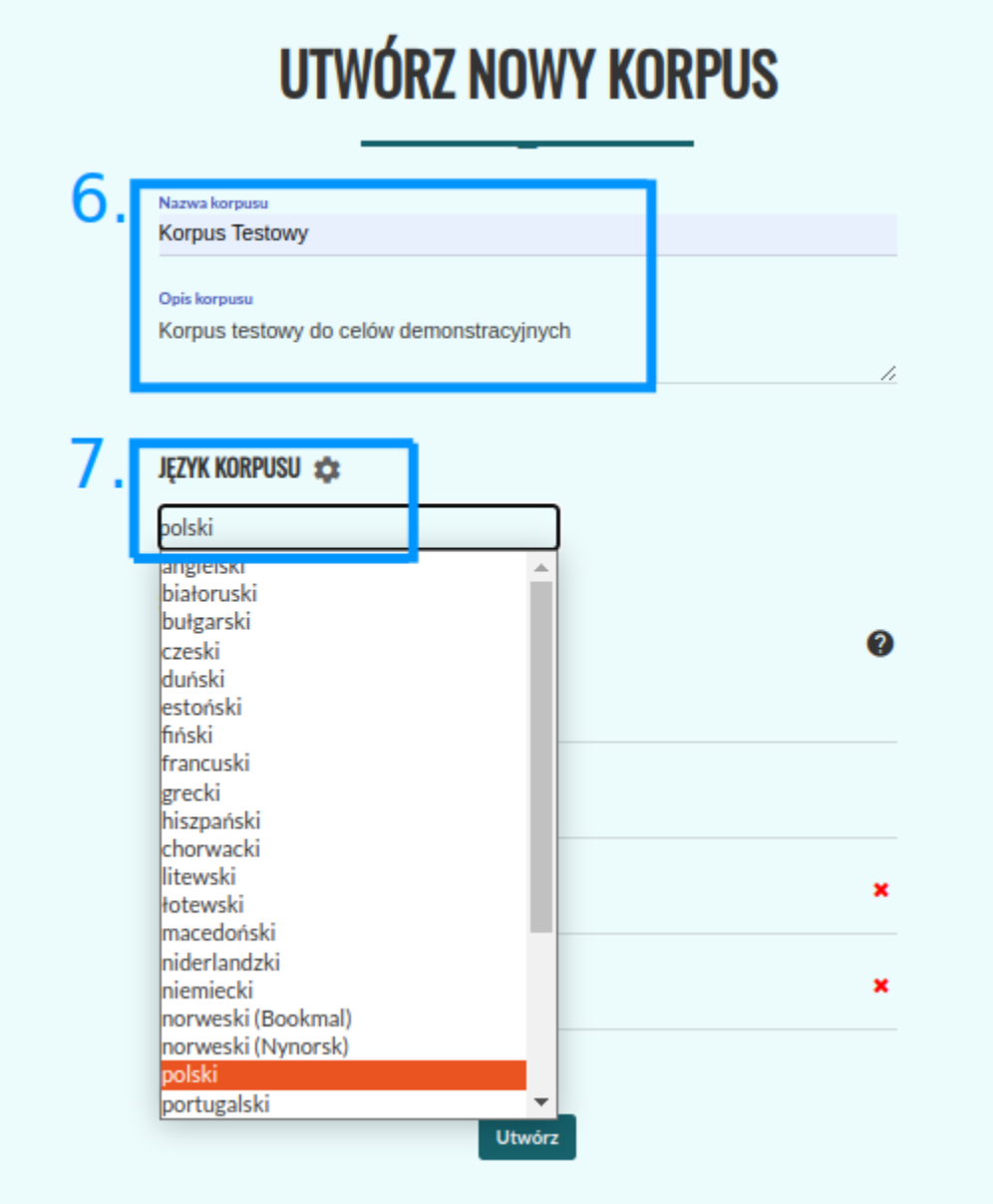
Konto można stworzyć, klikając w przycisk „Login/Rejestracja” w menu w prawym górnym rogu.

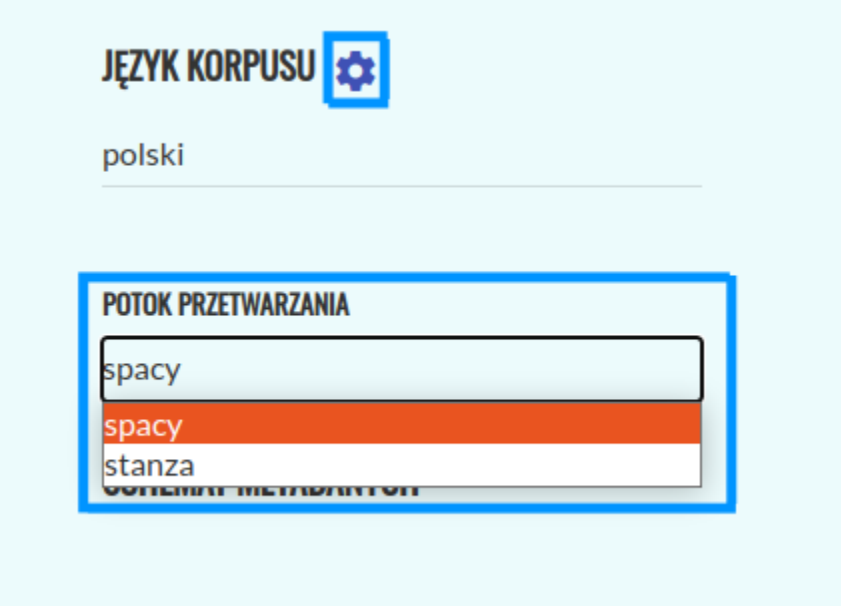

ZALOGUJ SIĘ
E-mail:
Hasło:
Pamiętaj mnie
Zaloguj się
Utwórz nowe konto
2.
Zapomniałeś hasła?

1.1.2 Tworzenie korpusu

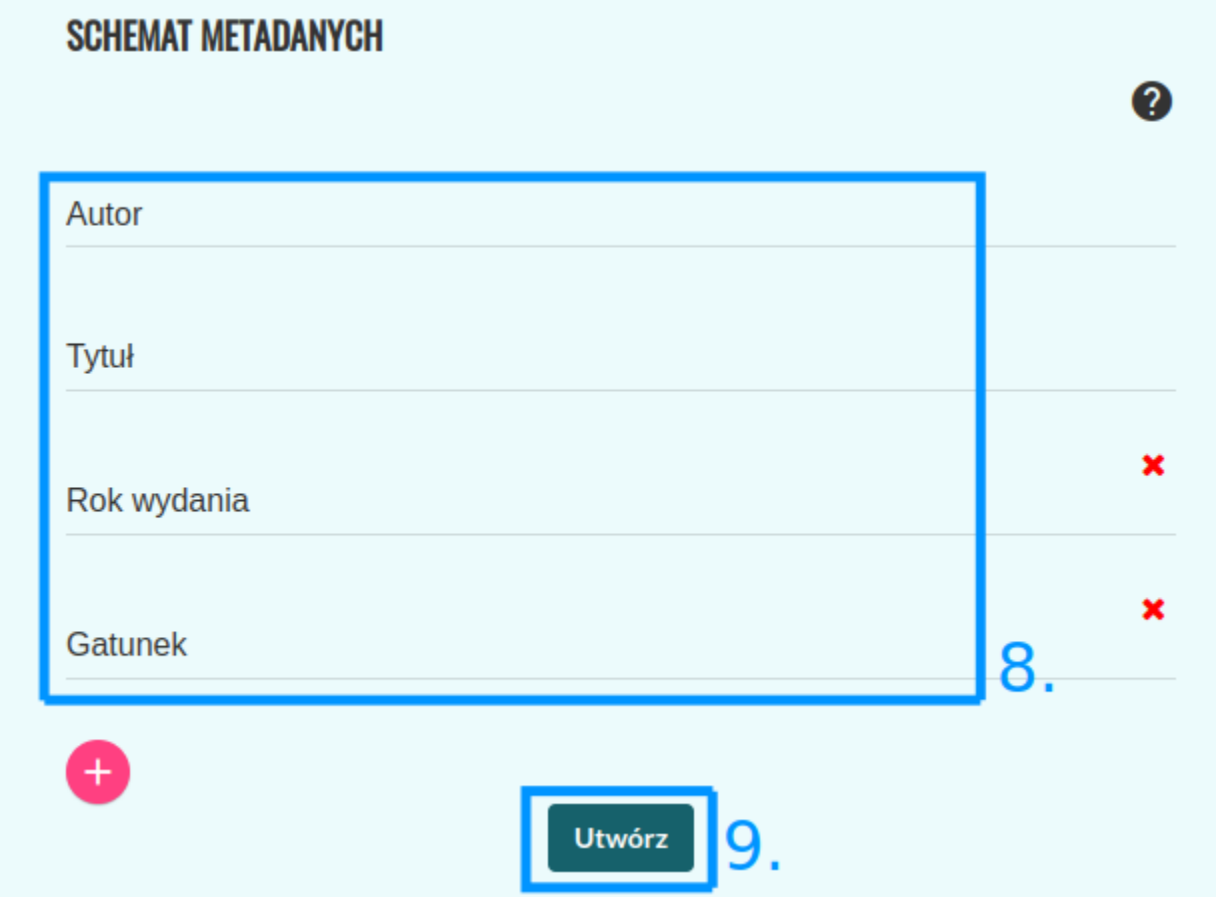
Aby utworzyć nowy korpus, należy kliknąć „Nowy korpus” w menu głównym.

Następnie należy wprowadzić nazwę i opis korpusu (6) oraz wybrać język korpusu (7).


UTWÓRZ NOWY KORPUS
6.
Nazwa korpusu
Korpus Testowy
Opis korpusu
Korpus testowy do celów demonstracyjnych
7.
JĘZYK KORPUSU
polski
angielski
białoruski
bułgarski
czeski
duński
estoński
fiński
francuski
grecki
hiszpański
chorwacki
litewski
łotewski
macedoński
niderlandzki
niemiecki
norweski (Bookmal)
norweski (Nynorsk)
polski
portugalski
Utwórz

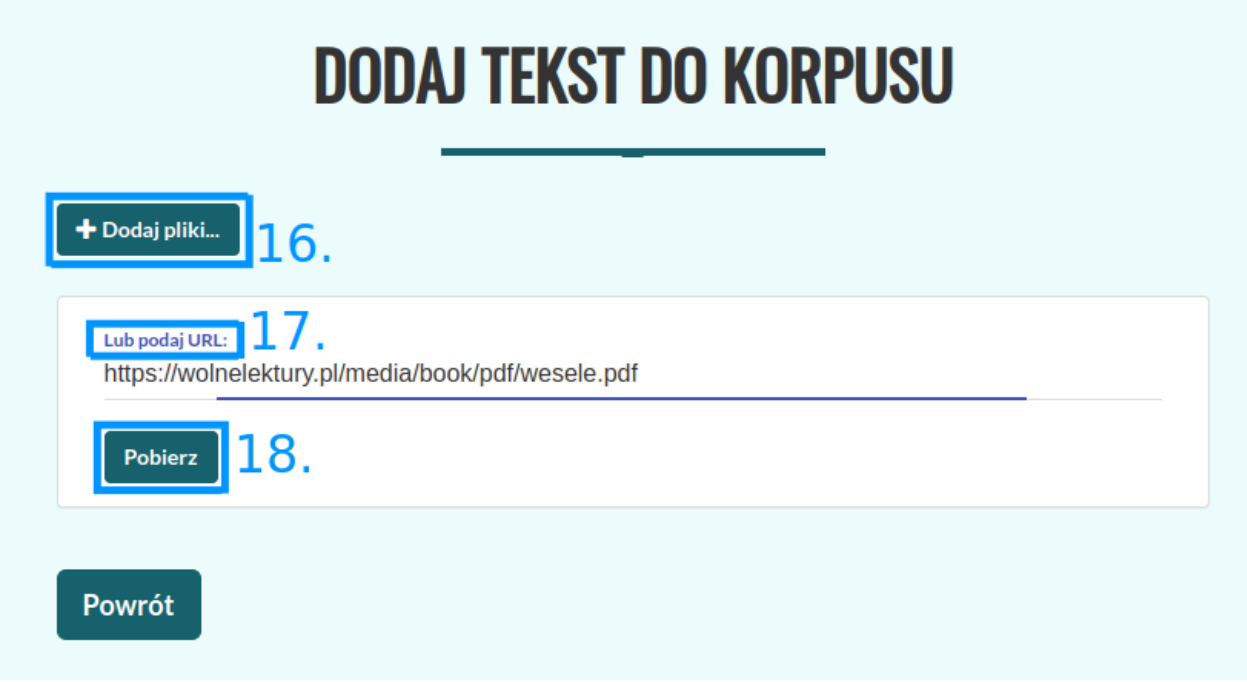


Po naciśnięciu ikony ustawień obok pola (7) możliwe jest wybranie preferowanego potoku przetwarzania. Jeśli dla danego języka dostępne są oba potoki przetwarzania, domyślnym potokiem jest Spacy. W polach (8) można wprowadzić metadane dla wszystkich tekstów w korpusie (możliwe jest ich zmodyfikowanie po utworzeniu korpusu). Aby zapisać korpus, należy kliknąć przycisk „Utwórz” (9).



Użytkownik zostanie przekierowany do widoku utworzonego korpusu. Po zmianie stanu korpusu na „Gotowy” (10) można dodać pierwszy tekst za pomocą przycisku „+” (11).

Po kliknięciu nastąpi przekierowanie do ekranu dodawania tekstu. Lista dozwolonych formatów znajduje się tutaj. Teksty można dodać na dwa sposoby.

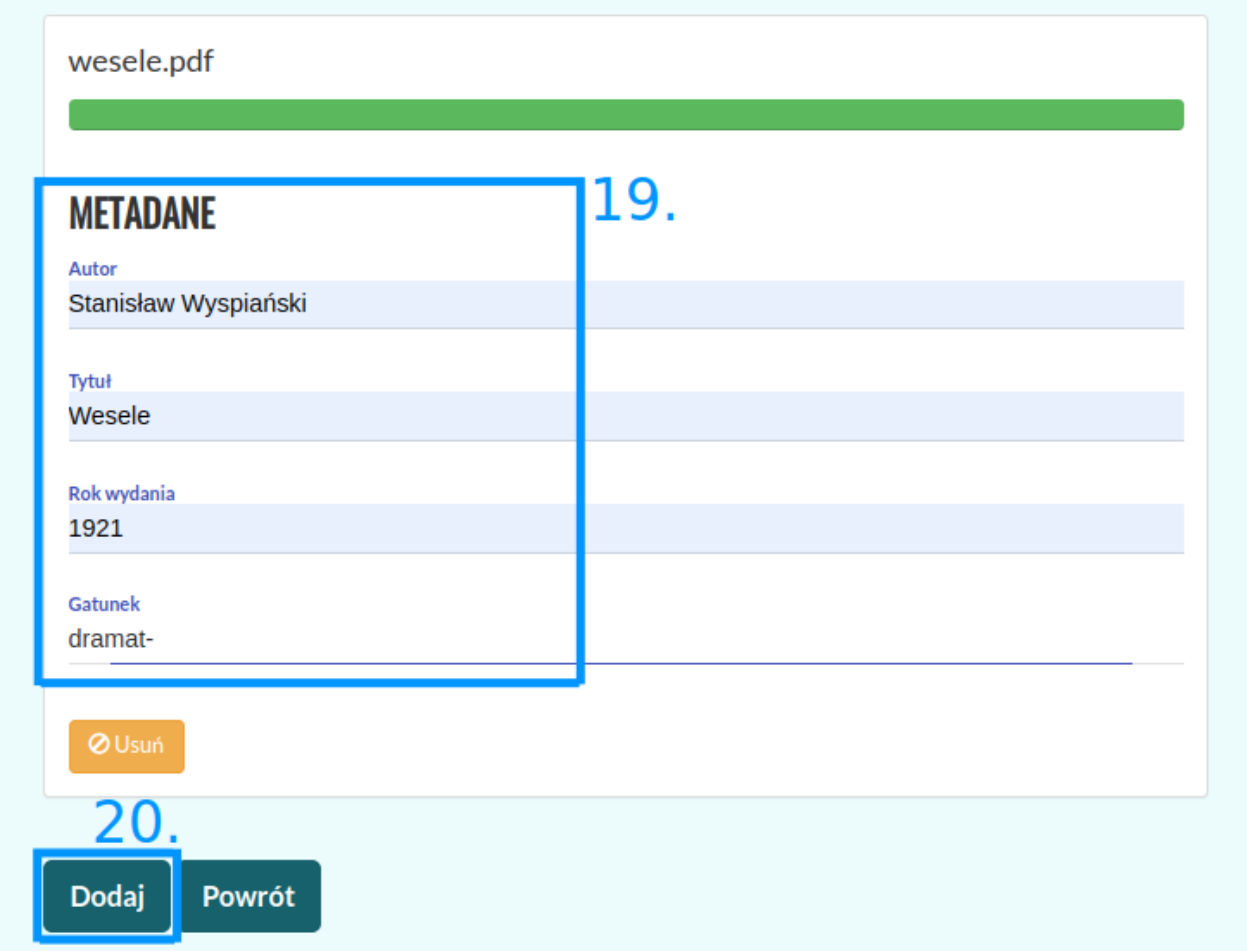


Pierwszym jest kliknięcie przycisku „+ Dodaj pliki” (16), który pozwala na dodawanie plików z lokalnego dysku. Po kliknięciu pojawi się okno wyboru plików, w którym można wskazać jeden lub wiele plików jednocześnie. Drugim

sposobem jest podanie bezpośrednio linku do tekstu w polu tekstowym „Lub podaj URL:" (17), a następnie kliknięcie przycisku „Pobierz" (18). Korpusomat pobierze wtedy plik automatycznie i przetworzy go. W takim wypadku możliwe jest również podanie linku do artykułu (np. z portalu internetowego), z którego zostanie wydobyta treść i przetworzona do pliku tekstowego.

Po załadowaniu treści można wprowadzić lub zmodyfikować metadane (19). Korpusomat automatycznie spróbuje wydobyć metadane z dodanego pliku, jednak nie zawsze jest to możliwe. W wypadku tekstów w formatach EPUB i MOBI oraz stron internetowych Korpusomat spróbuje wydobyć metadane z nagłówków dokumentów. W wypadku plików tekstowych automatyczne rozpoznawanie metadanych wymaga tego, by nazwy plików zapisane były w następującym formacie: „autor - tytuł (miejsce, rok)". Przykładowo, aby Korpusomat automatycznie rozpoznał metadane „Pana Tadeusza" z nazwy pliku, dodany plik powinien nazywać się „Adam Mickiewicz - Pan Tadeusz (Paryż, 1834).txt".

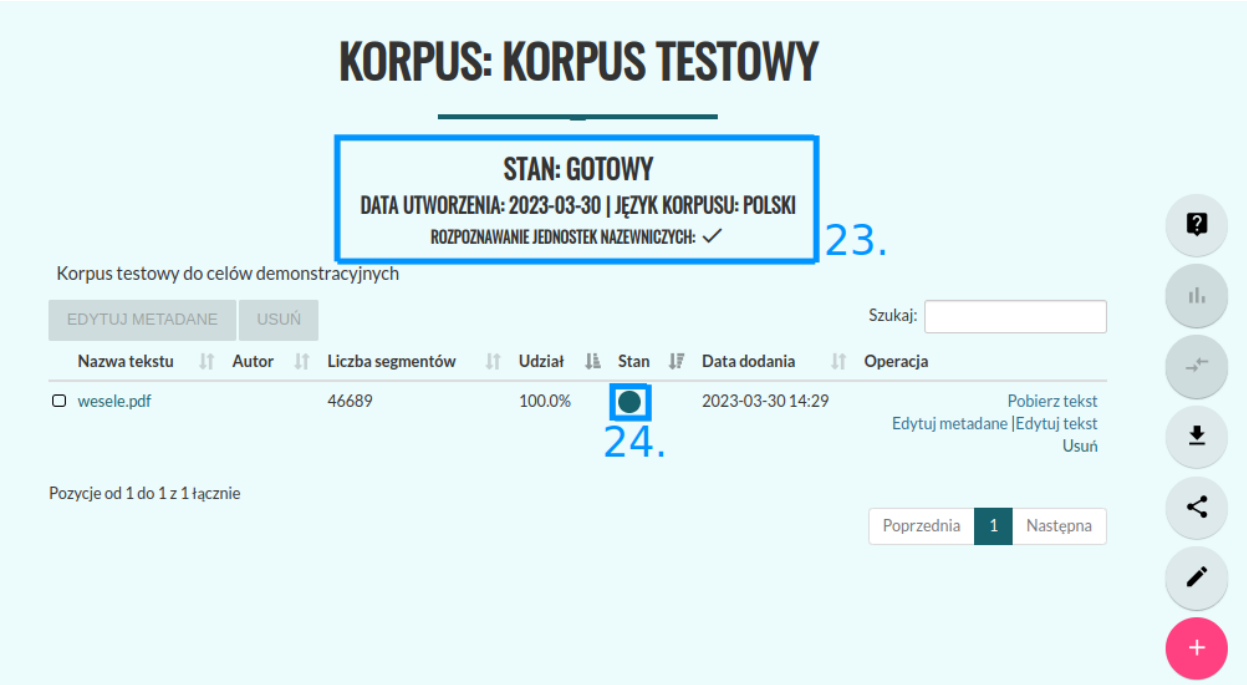
Aby dodać więcej tekstów, należy ponownie użyć przycisku „+ Dodaj pliki…" (16), czy pola „Lub podaj URL" (17).



Aby przejść dalej, należy kliknąć przycisk „Dodaj" (20).

Po dodaniu tekstów zostaniemy przeniesieni do ekranu korpusu, a Korpusomat zacznie przetwarzać teksty. Przy nazwie korpusu pojawi się informacja o stanie jego przetwarzania oraz data jego utworzenia (21). Przy każdym z tekstów będzie wyświetlony status przetwarzania (22). Podczas analizy będzie to „Trwa przetwarzanie" (22). Przetworzanie książki o objętości ok. 80-100 tys. słów powinno potrwać około 4-5 minut, choć zależy to również od aktualnego obciążenia serwera oraz wybranych warstw znakowania. Obecnie maksymalny czas przetwarzania jednego pliku wynosi 10 minut – zadania dłuższe zakończą się niepowodzeniem. Podczas przetwarzania tekstów można nadal dodawać kolejne teksty za pomocą przycisku (11).

Gdy wszystkie teksty zostaną przetworzone a ich status zostanie oznaczony jako „Przetworzony prawidłowo”, status całego korpusu zostanie również automatycznie zmieniony na „Gotowy” (23).

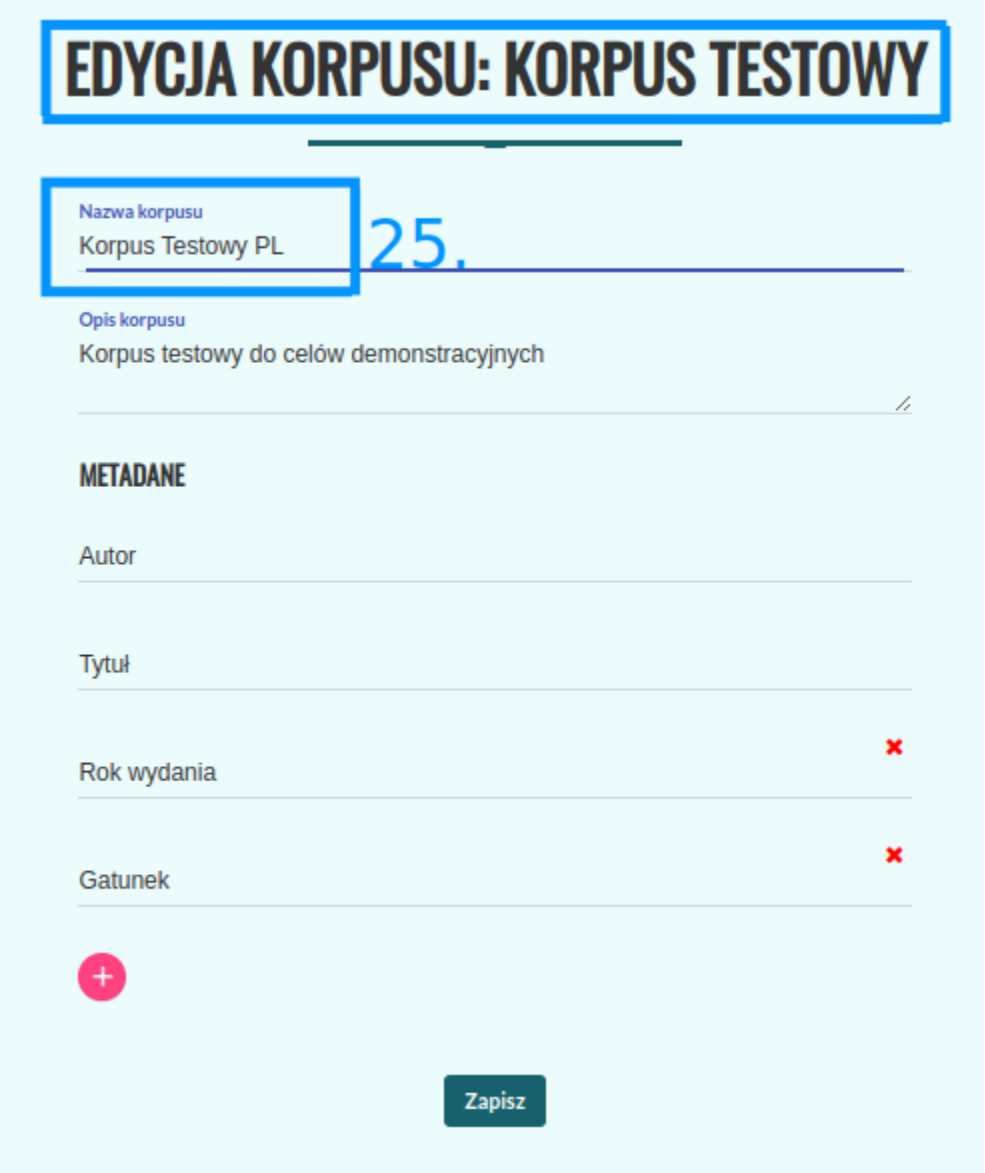


Na tym etapie będzie można przystąpić do dalszej pracy z korpusem. Możliwe dalsze czynności to:

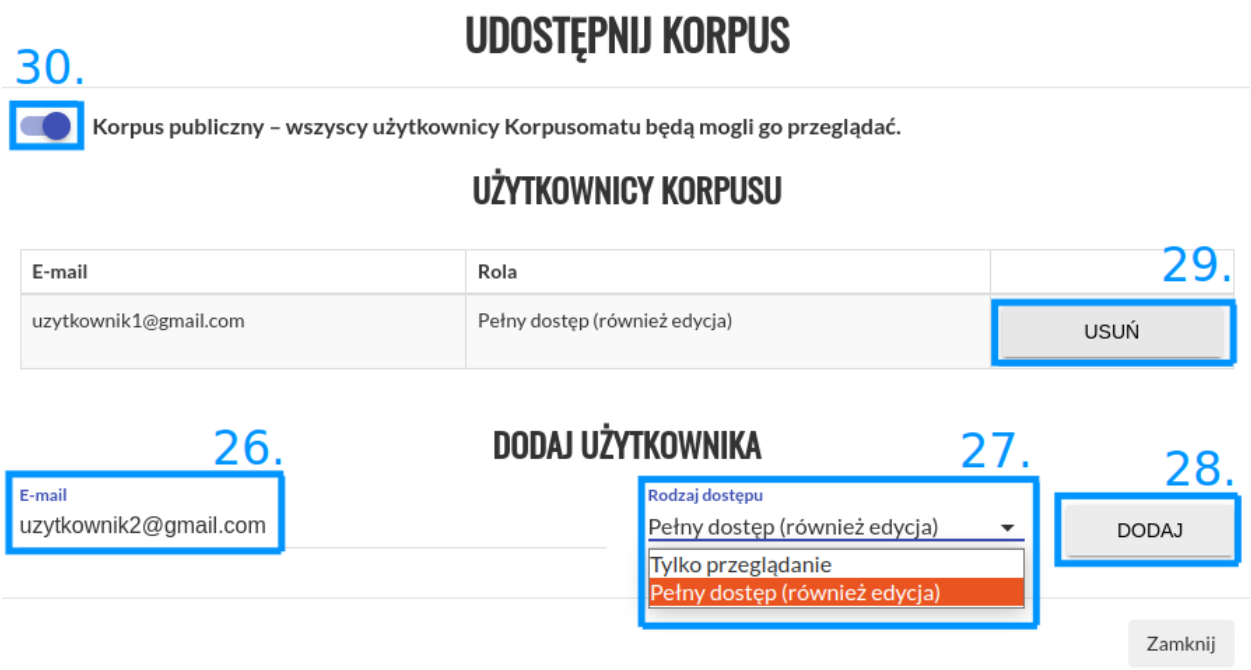
1. Edycja korpusu (12)
2. Udostępnienie korpusu innym użytkownikom (13)

3. Pobieranie przetworzonych plików XML — przycisk (14)
4. Przeszukiwanie korpusu (15)

Ad 1. Po wciśnięciu przycisku (12) możliwa jest edycja nazwy i opisu korpusu, a także dodanie lub zmiana metadanych (25).



Ad 2. Po wciśnięciu przycisku (13) możliwe jest udostępnienie utworzonego korpusu innym użytkownikom Korpusomatu. W polu (26) należy wpisać adres email użytkownika, któremu udostępnia się utworzony korpus, następnie wybiera się rodzaj dostępu jaki ma być przydzielony (27). Po zatwierdzeniu zmian przyciskiem „Dodaj" (28) nowy użytkownik korpusu zostaje dodany. Możemy go usunąć przyciskiem (29). W celu udostępnienia korpusu wszystkim użytkownikom Korpusomatu należy przełączyć przycisk (30).



Ad 3. Kliknięcie przycisku (14) spowoduje pobranie archiwum z przetworzonymi plikami XML tekstów w korpusie. Pliki te są w formacie zgodnym ze specyfikacją CCL.

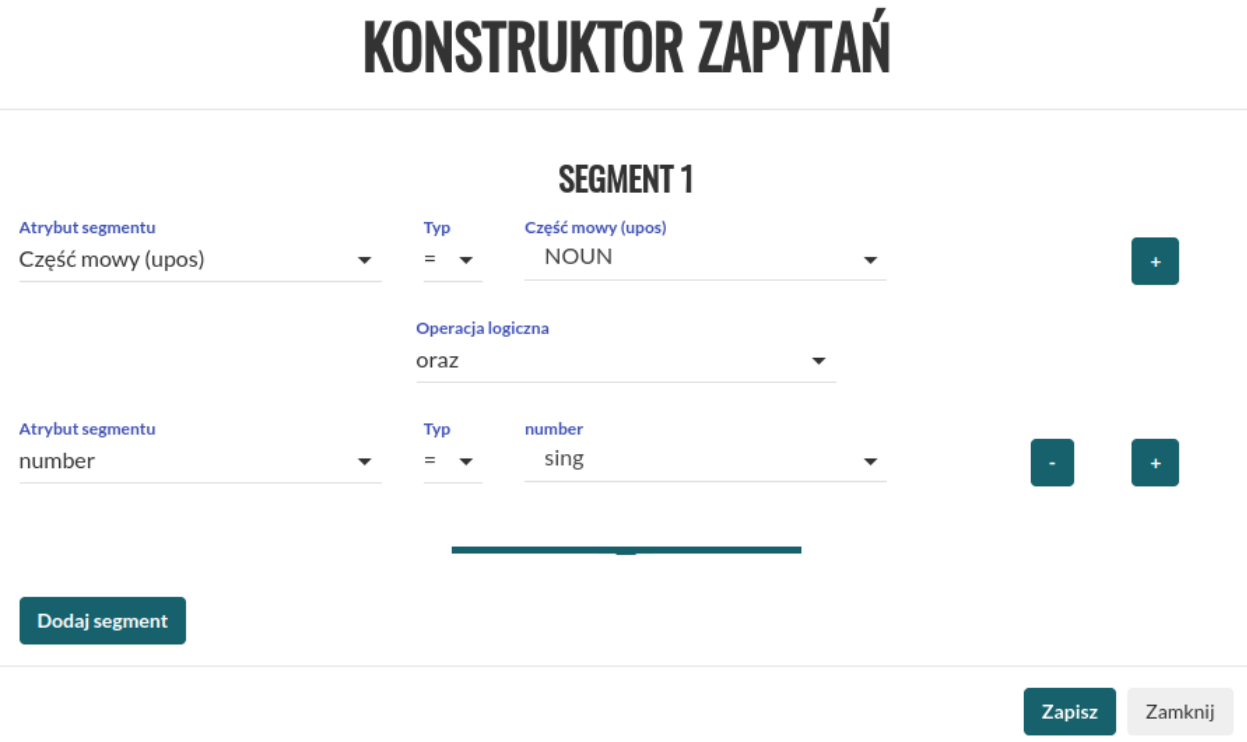
Na tym etapie nadal można edytować korpus. Dodawanie oraz usuwanie tekstów spowoduje automatyczne uruchomienie procesu przetwarzania, po zakończeniu którego korpus z powrotem otrzyma status „Gotowy".

Ad 4. Kliknięcie przycisku (15) spowoduje przeniesienie do ekranu wyszukiwania. Sposób tworzenia zapytań został opisany w dalszej części instrukcji.

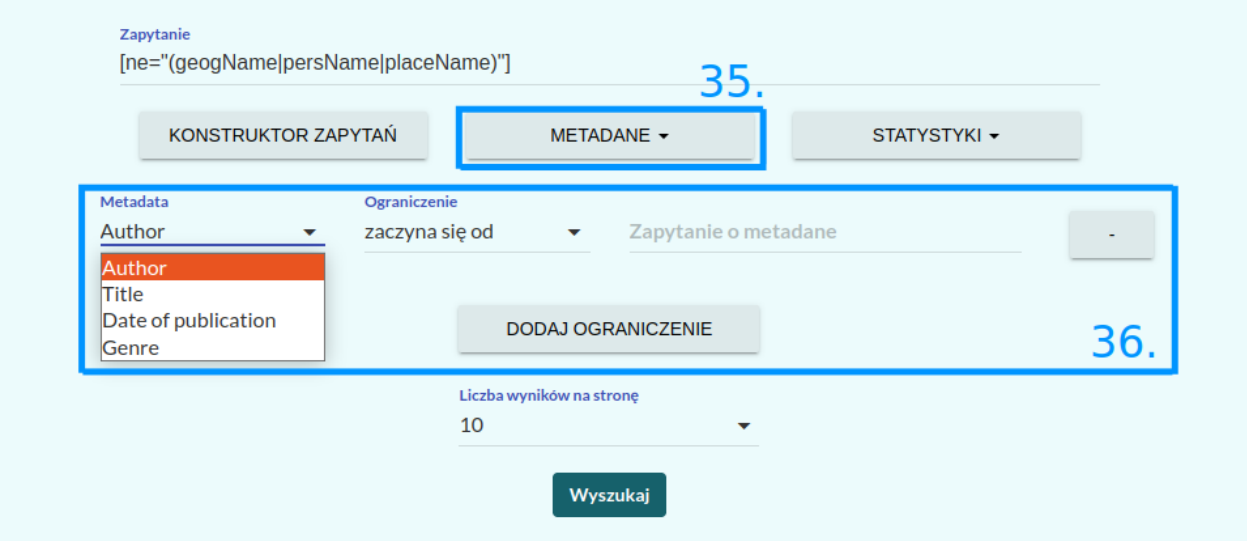
1.1.3 Wyszukiwanie w korpusie

W polu „Zapytanie" (31) należy wpisać zapytanie, które chcemy wykonać, a następnie wcisnąć przycisk „Wyszukaj" (34). Opis języka zapytań dostępny jest w kolejnej części instrukcji. Przycisk (32) uruchamia graficzny konstruktor zapytań. Przycisk (33) rozwija menu ograniczenia wyszukiwania do tekstów o konkretnych metadanych.

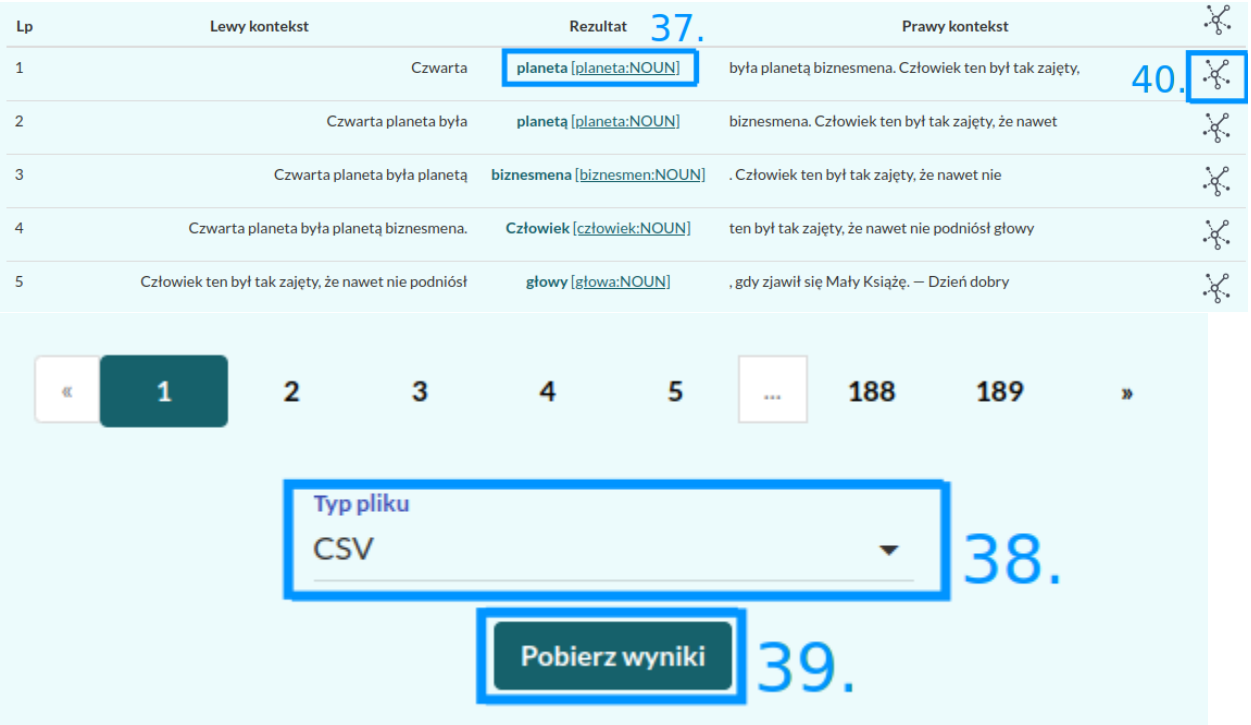
Kliknięcie przycisku (32) spowoduje otwarcie ekranu konstruktora zapytań. Pozwala on na zbudowanie interesującego zapytania poprzez wybranie cech segmentów z rozwijanych list. Po wybraniu wszystkich cech należy kliknąć przycisk „Zapisz", aby powrócić do ekranu wyszukiwania. W polu zapytanie pojawi się zapytanie przetworzone na język zapytań wyszukiwarki (CQL – Corpus Query Language).



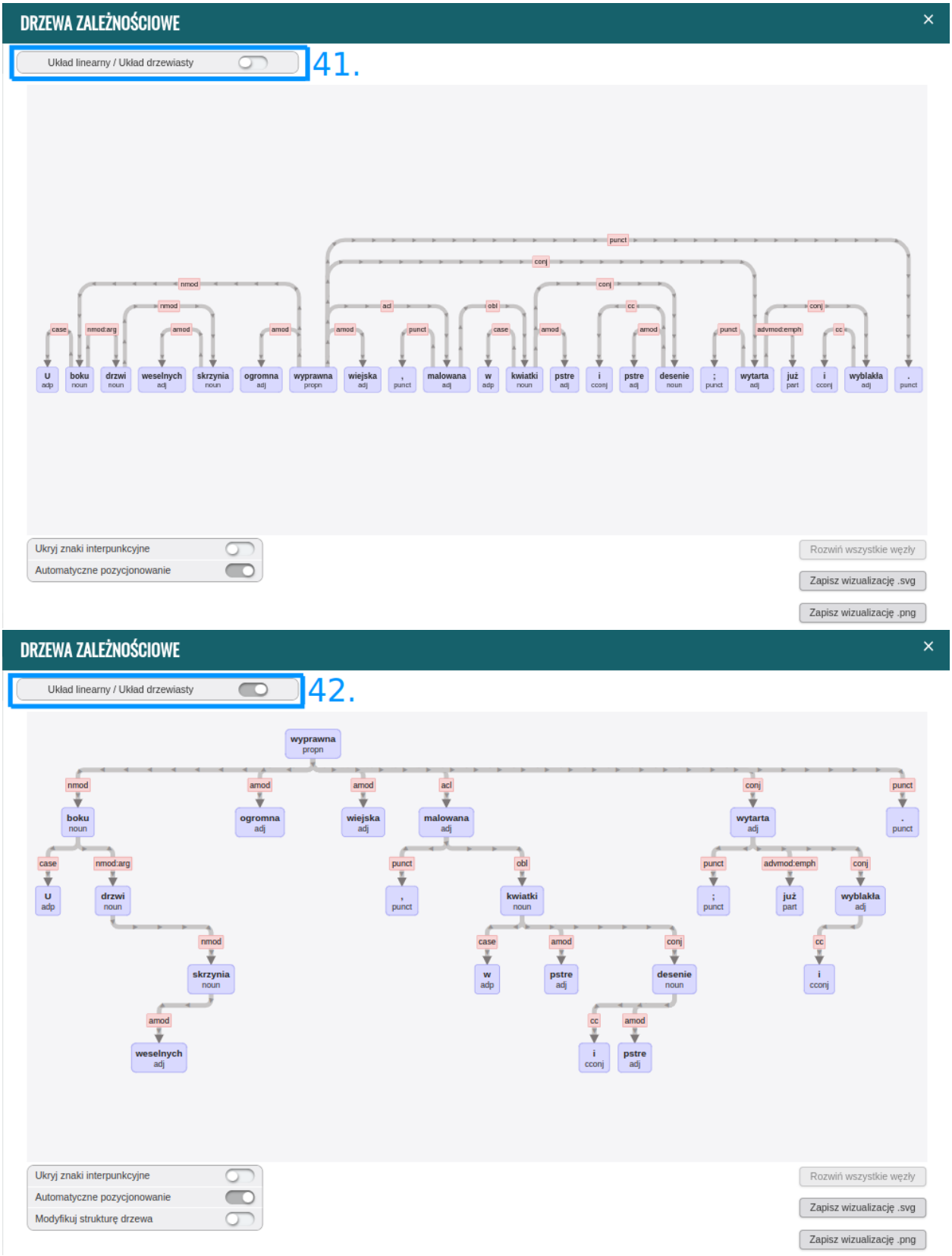
Kliknięcie przycisku (35) spowoduje rozwinięcie menu metadanych (36). Można tutaj ograniczyć wyniki wyszukiwania jedynie do tekstów, które spełniają określone kryteria.



Po wykonaniu zapytania użytkownik zostanie przeniesiony do strony z wynikami w postaci konkordancji. Szerszy kontekst znalezionego wyrazu lub wyrażenia oraz metadane tekstu, z którego pochodzi, można obejrzeć, klikając na niego (37). Aby pobrać całą listę wyników w formie pliku CSV lub XLS, należy wybrać typ pliku (38) i wcisnąć przycisk „Pobierz wyniki” (39).

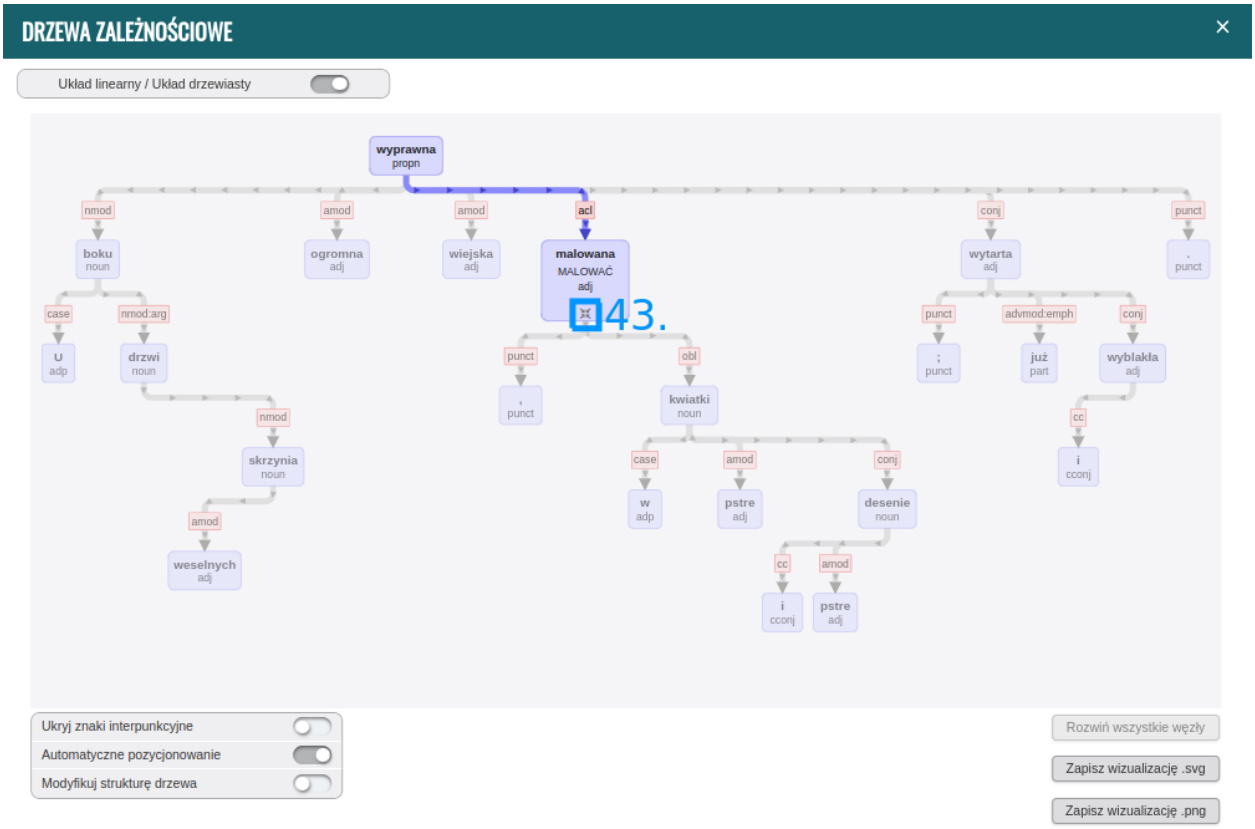


Po kliknięciu ikony (40) użytkownik zostaje przekierowany do strony z wizualizacją drzew zależnościowych. Wizualizacji podlega pełne wypowiedzenie, zawierające konkretny przykład wyszukanych rezultatów zapytania. Użytkownik może się przełączać między dwoma układami: linearnym (41), bądź drzewiastym (42).



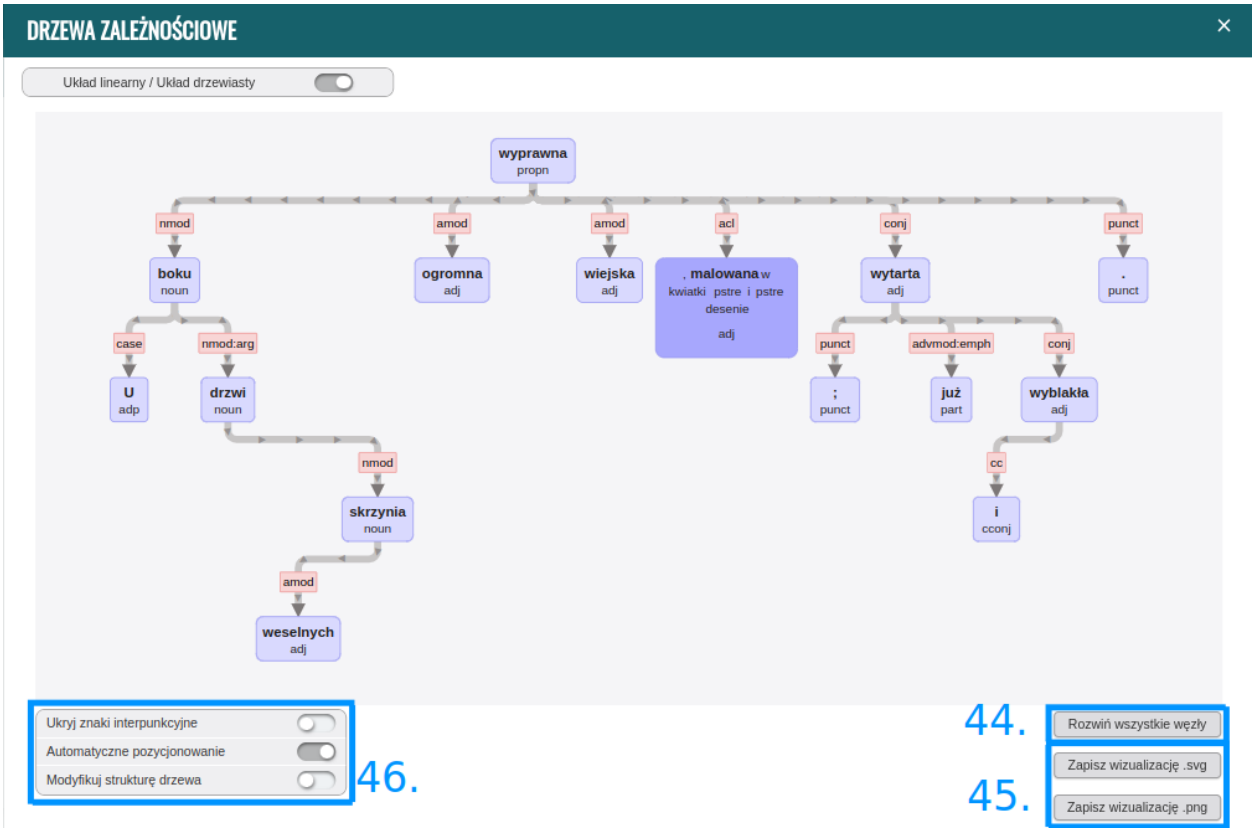
Użytkownik może zwinąć wybrane węzły. W tym celu należy najechać myszą na wybrany węzeł i kliknąć ikonę (43).

W ten sam sposób można rozwinąć zwinięty wcześniej węzeł lub użyć przycisku (44), który rozwinie wszystkie węzły w drzewie.



Wizualizację drzewa można zapisać w formacie SVG lub PNG (45). w polach (46) dostępne są dodatkowe opcje:

- ukrycia znaków interpunkcyjnych w analizowanym zdaniu dla zwiększenia jego czytelności,
- automatycznego pozycjonowania drzewa (po wybraniu tego przycisku analizowane drzewo zostanie usytuowane w centralnej części okna modalnego),
- modyfikacji struktury drzewa (poszczególne węzły można przesuwać w obrębie drzewa).



1.1.4 Preferencje

Po wciśnięciu przycisku „Preferencje” w głównym menu można ustawić postać niektórych elementów wyświetlanych na stronie wyników zapytań. W części „Informacja w rezultatach” można wybrać rodzaj znacznika, który zostanie wyświetlony w głównej kolumnie wyników (xpos lub upos). W części „Informacja w etykietach” można wybrać warstwy znakowania, których informacje będą się wyświetlać w dymkach widocznych po najechaniu myszą na wynik wyszukania korpusowego.

W części „Kolumny eksportowane podczas generowania pliku z wynikami zapytania do korpusu” można określić, które informacje mają znaleźć się w pliku eksportu wyników wyszukiwania.

Dodatkowo można zmienić hasło oraz wyrazić zgodę na otrzymywanie treści o nowościach w Korpusomacie lub taką zgodę cofnąć.


PREFERENCJE
Informacja w rezultatach:
xpos
upos
Informacja w etykietach:
Części mowy
Jednostki nazewnicze
Informacje składniowe
Własności morfologiczne
Kolumny eksportowane podczas generowania pliku z wynikami zapytania do korpusu:
Lewy kontekst
Wynik
Prawy kontekst
Interpretacja
Nazwa własna
Relacja zależnościowa
Lemat nadrzędnika
Autor
Tytuł
Data
Miejsce
Gatunek
Wiadomości e-mail:
Chcę otrzymywać wiadomości e-mail z informacjami o nowościach w Korpusomacie.
Zmiana hasła
Zapisz

1.2 Tworzenie zapytań do korpusu

Niniejsza część instrukcji opisuje język zapytań CQL w odniesieniu do warstw znakowania dostępnych w Korpusomacie.

1.2.1 Segmentacja

Znaczniki morfosyntaktyczne, tzw. tagi, przypisane są segmentom (tokenom, w przybliżeniu słowom). Segmenty nie są dłuższe niż słowa ortograficzne (słowa 'od spacji do spacji' z oddzieleniem znaków interpunkcyjnych), ale w niektórych wypadkach segmenty mogą być krótsze niż takie słowa. Szczegółowe zasady segmentacji dla poszczególnych języków mogą się różnić i zależą od decyzji podjętych przez twórców zasobów jezykowych dla danego języka (głównie twórców banków drzew zależnościowych) oraz przez twórców konkretnych narzędzi programistycznych. Przykładowo, w korpusach języka polskiego (w tym m.in. w Narodowym Korpusie Języka Polskiego) na etapie segmentacji zwykło się oddzielać od form przeszłych czasowników tzw. aglutynant (wykładnik osoby i liczby) a także partykułę *by* będącą wykładnikiem trybu warunkowego. W efekcie jedno słowo tekstowe zostaje rozbite odpowiednio na dwa lub trzy segmenty i każdemu z nich jest przypisana osobna interpretacja fleksyjna: np. *[pisał][eś]*, *[jedli][by][śmy]*. Jednak w Korpusomacie ta decyzja jest dodatkowo zależna od wybranego potoku przetwarzania, bowiem twórcy tych potoków podjęli w tej sprawie różne decyzje: Stanza stosuje segmentację taką, jak w Narodowym Korpusie Języka Polskiego, czyli oddziela aglutynant i partykułę *by*, ale spaCy uznaje formy czasu przeszłego oraz trybu warunkowego za pojedyncze segmenty i nie rozbija ich. To jednak rzadkie i skrajne przypadki wynikające ze specyfiki wykorzystanych narzędzi — w większości wypadków teksty w poszczególnych językach powinny być segmentowane w obu potokach tak samo i zgodnie z typową dla danego języka segmentacją stosowaną w bankach drzew składniowych oraz korpusach narodowych danych języków.

1.2.2 Znaczniki morfosyntaktyczne

Wszystkie korpusy w Korpusomacie zawierają warstwę informacji morfosyntaktycznej zgodną ze specyfikacją Universal Dependencies. Informacja ta jest rozdzielona na dwie składowe: oznaczenie części mowy (tzw. UPOS — *universal part of speech*) oraz cechy morfosyntaktyczne (tzw. UFEATS — *universal features*). Obie te składowe (nazwy części mowy, nazwy cech morfoskładniowych i listy ich możliwych wartości) są opisane w dokumentacji na stronie projektu UD. Ponieważ z zasady jest to opis uniwersalny, każdy z konkretnych języków korzysta tylko z podzbioru cech morfologicznych i ich wartości.

Oprócz informacji morfosyntaktycznej zgodnej ze specyfikacją UD w większości korpusów dostępny jest również dodatkowy znacznik, tzn. XPOS, który przechowuje informację morfosyntaktyczną zgodną z tagsetem stosowanym w zasobach dla danego języka. Oba potoki przetwarzania znakują teksty również znacznikami XPOS, ale ich konkretna postać zależy zarówno od twórców narzędzi, jak i twórców banku drzew UD. W szczególności od twórców banku drzew zależy, jaka postać znacznika znajdzie się w polu XPOS — zarówno pod względem szczegółowości opisu morfosyntaktycznego, jak i technicznego opisu samego tagsetu, dlatego nie mają one ustandaryzowanej wspólnej postaci. Najczęściej są to jednak systemy znaczników stosowane w narodowych korpusach tych języków. W wypadku gdy twórcy banku drzew UD nie umieścili w nim znaczników XPOS, Korpusomat również nie umożliwa korzystania z nich — tak jest wypadku języka rosyjskiego (w obu potokach przetwarzania). W niektórych wypadkach postać znacznika XPOS może się różnić w zależności od potoku przetwarzania, np. w wypadku języka polskiego Stanza zwraca pełne znaczniki morfosyntaktyczne (w tagsecie stosowanym w polskich korpusach), spaCy zaś ogranicza się tylko do pierwszej części takiego znacznika oznaczającej przynależność słowa do klasy gramatycznej.

1.2.3 Język zapytań

Składnia zapytań w programie MTAS została oparta na języku zapytań o nazwie Corpus Query Language (CQL), który jest powszechnie znany i stosowany w wyszukiwarkach korpusowych. Niniejszy rozdział omawia składnię CQL w wariancie zastowanym w Korpusomacie.

MTAS jest uniwersalną wyszukiwarką pozwalającą na przeszukiwanie korpusów zawierających wiele warstw anotacyjnych. Niniejsza instrukcja dotyczy przeszukiwania korpusów postaci indeksowanej przez Korpusomat, który tworzy aktualnie trzy warstwy znakowania: warstwę morfosyntaktyczną i składniową oraz warstwę jednostek nazewniczych. Ogólna podstawowa dokumentacja wyszukiwarki MTAS znajduje się na jej stronie internetowej.

Zapytania o segmenty

Podstawową jednostką wyszukiwaną w korpusie jest segment. Segmenty w zapytaniach są ograniczone nawiasami kwadratowymi, wewnątrz których można określać konkretne cechy, które segment ma spełniać. W najprostszym przypadku jest to kształt tekstowy (napis). Do zapytań o tę postać ortograficzną segmentu służy atrybut `orth`, można też jednak ograniczyć się do wpisania w oknie wyszukiwarki poszukiwanego słowa (lub słów). Zatem poniższe zapytanie o dwa sąsiadujące ze sobą segmenty:

```
[orth="komisja"][orth="szkolna"]
```

można zadać również w prostszy sposób:

```
komisja szkolna
```

Domyślnie rozróżniana jest kasztowość (wielkość) liter, a zatem poniższe dwa zapytania dadzą różne wyniki:

- `przyszedł`
- `Przyszedł`

Dostępny jest jednak dodatkowy atrybut pomocniczy `orth_lc` (`lc` od ang. *lower case*) przechowujący postać ortograficzną segmentu z zamienionymi literami wielkimi na małe. Dzięki temu można wyszukiwać słowa zapisane w różny sposób bez konieczności odwoływania się do wyrażeń regularnych. Na przykład zapytanie `[orth_lc="przyszedł"]` zwróci wystąpienia słów postaci *przyszedł* i *Przyszedł*, jak również *PRZYSZEDŁ* czy *PRzySZedŁ*.

W zapytaniach o segmenty mogą wystąpić standardowe wyrażenia regularne wykorzystujące następujące znaki specjalne: `?`, `*`, `+`, `.`, `,`, `|`, `,`, `[`, `]`, `(`, `)` oraz liczby naturalne pisane cyframi arabskimi, np. `0` czy `21`. Ponieważ formalny opis wyrażeń regularnych wykracza poza ramy niniejszej instrukcji, ograniczymy się tutaj do kilku przykładów, które powinny pozwolić użytkownikowi na szybkie przyswojenie składni i znaczenia takich wyrażeń.

1. `[orth="(Ala|Ela)"]`

   znak `|` oznacza alternatywę dwóch wyrażeń (całość należy dodatkowo ująć w nawiasy okrągłe), a zatem zapytanie to może zostać użyte do znalezienia wszystkich wystąpień segmentów *Ala* lub *Ela*,

2. `[orth="[AE]la"]`

   nawiasy kwadratowe oznaczają alternatywę znaków, a zatem zapytanie to może zostać użyte do znalezienia tych segmentów, których pierwszy znak to *A* lub *E*, po którym następuje ciąg znaków postaci *la*, tj. zapytanie to jest równoważne poprzedniemu,

3. `[orth="beza?"]`

   znak zapytania oznacza opcjonalność znaku (tutaj ostatniego *a*) lub ujętego w nawiasy okrągłe wyrażenia bezpośrednio poprzedzającego znak `?`, a zatem w wyniku zadania tego zapytania znalezione zostaną segmenty *bez* i *beza*,

4. `[orth="bez."]`

   kropka oznacza dowolny znak, a zatem wynikiem tego zapytania będą segmenty *beza*, *bezy*, *bezą* itp., ale nie *bez* czy *bezami*,

5. `[orth="bez.?"]`

   bez, *beza*, *bezy*, *bezą* itp., ale nie *bezami*,

6. `[orth=".z.z."]`

   segmenty pięcioznakowe, w których 2. i 4. znak to *z* (np. *czczą* i *rzezi*),

7. `[orth="a*by"]`

   gwiazdka oznacza dowolną liczbę wystąpień znaku lub wyrażenia bezpośrednio przed nią, a zatem zapytanie to może posłużyć do znalezienia segmentów składających się z dowolnej liczby liter *a*, po których następuje ciąg *by*, np. *by* (zero wystąpień *a*), *aby*, *aaaaby* itp.,

8. `[orth="Ala.*"]`

   segmenty zaczynające się na *Ala*, np. *Ala* i *Alabama*,

9. `[orth=".*al+"]`

   plus ma działanie podobne do gwiazdki i oznacza dowolną większą od zera liczbę wystąpień znaku lub wyrażenia bezpośrednio przed nim, a zatem wynikiem tego zapytania będzie znalezienie segmentów kończących się na *al*, *all*, *alll* itd., ale nie na *a*, np. *dal*, *robal* i *Gall*,

10. `[orth="a{1,3}b.*"]`

    konstrukcja typu `n,m` oznacza od `n` do `m` wystąpień znaku lub wyrażenia bezpośrednio przed nią, a zatem zapytanie to pomoże znaleźć segmenty zaczynające się od ciągu od 1 do 3 liter a, po którym następuje litera b, a następnie dowolny ciąg znaków (por. `.*`), np. *aby*, *aaaby*, *absolutnie*,

11. `[orth=".*(la){3,}.*"]`

    konstrukcja typu `n,` oznacza co najmniej `n` wystąpień znaku lub ujętego w nawiasy okrągłe wyrażenia bezpośrednio przed nią, a zatem zapytanie to może posłużyć do znalezienia segmentów, w których ciąg *la* występuje przynajmniej 3 razy z rzędu, np. *tralalala*, *sialalala*,

Zapytania z innymi atrybutami

Aby znaleźć wszystkie formy leksemu *korpus*, można użyć następującego zapytania:

```
[lemma="korpus"]
```

Atrybut `lemma` jest jednym z wielu, jakie mogą pojawić się w zapytaniu. Wartością tego atrybutu powinna być forma podstawowa (hasłowa), a zatem zapytanie `[lemma="pisać"]` może być użyte do znalezienia form typu *pisać*, *piszę*, *pisała*, *piszcie*, *pisanie*, *pisano*, *pisane* itp.

Podobnie jak w wypadku atrybutu `orth` wartościami atrybutu `lemma` mogą być wyrażenia regularne, np:

```
[lemma="komit[ae]t"]
```

znalezione zostaną wszystkie segmenty, których forma hasłowa ma postać komitet lub komitat.

Zapytania o różne atrybuty segmentów można łączyć. Na przykład, aby znaleźć wszystkie wystąpienia segmentu *minę* rozumianego jako forma leksemu mina (a nie na przykład leksemu minąć), można zadać następujące zapytanie:

```
[orth="minę" & lemma="mina"]
```

Podobne znaczenie ma następujące zapytanie o te wystąpienia segmentu *minę*, które nie są interpretowane jako formy leksemu minąć.

```
[orth="minę" & !lemma="minąć"]
```

W powyższych zapytaniach operator `&` spełnia rolę logicznej koniunkcji. Operatorem do niego dualnym jest operator `|`, spełniający rolę logicznej alternatywy. Oto kilka przykładów użycia tego operatora:

- `[lemma="on" | lemma="ja"]`

  wszystkie formy zaimków on i ja, równoważne zapytaniu `[lemma="on|ja"]`,

- `[lemma="on" | orth="mnie" | orth="ciebie"]`

  wszystkie formy zaimka on, a także segmenty *mnie* i *ciebie*,

- `[orth="pora" & !(lemma="por" | lemma="pora")]`

  segment *pora* niebędący ani formą leksemu por, ani formą leksemu pora.

Aby lepiej zrozumieć różnicę pomiędzy operatorami `&` i `|`, porównajmy następujące dwa zapytania:

```
[orth="minę" & lemma="mina"]
[orth="minę" | lemma="mina"]
```

W wyniku zadania pierwszego zapytania znalezione zostaną te segmenty, które są jednocześnie (koniunkcja) segmentem *minę* i formą leksemu mina, a więc wyłącznie te wystąpienia segmentu *minę*, które są interpretowane jako formy leksemu mina. W wyniku zadania drugiego zapytania znalezione natomiast zostaną te segmenty, które są albo dowolnie interpretowanym segmentem *minę*, albo formą leksemu mina (alternatywa), czyli wszystkie wystąpienia zarówno segmentu *minę*, jak i segmentów *mina*, *miny*, *minami* itp. interpretowanych jako formy leksemu mina.

Specyfikacje pozycji w korpusie, ujęte w nawiasy kwadratowe, mogą zawierać dowolną liczbę warunków typu `atrybut="wartość"` (na przykład `orth="nie"`) połączonych operatorami `!`, `&` i `|`, tak jak pokazują to powyższe przykłady. Możliwe jest także całkowite pominięcie jakichkolwiek warunków. Poniższe zapytanie mogłoby posłużyć do znalezienia wszystkich segmentów w korpusie.

```
[]
```

Taka „pusta” specyfikacja pozycji w korpusie, pasująca do dowolnego segmentu, może posłużyć na przykład do znalezienia dwóch form oddzielonych od siebie dowolnymi dwoma segmentami, np.:

```
[orth="się"][][][lemma="bać"]
```

W wyniku tego zapytania zostaną znalezione ciągi takie jak *się mnie też bać* czy *się nie chcę bać*.

Dla wielu zastosowań ciekawsza byłaby możliwość zapytania na przykład o formy oddalone od siebie o najwyżej pięć pozycji. MTAS umożliwia zadawanie takich pytań, gdyż pozwala na formułowanie wyrażeń regularnych także na poziomie pozycji korpusu. Na przykład zapytanie o formę leksemu bać występującą dwie, trzy lub cztery pozycje dalej niż forma *się* może wyglądać następująco:

```
[orth="się"][]{2,4}[lemma="bać"]
```

W wyniku tego zapytania zostaną znalezione ciągi uzyskane w wyniku poprzedniego zapytania, a także na przykład ciąg *się pani niczego nie boi*.

Zapewne nieco bardziej precyzyjnym zapytaniem o różne wystąpienia form tzw. czasownika zwrotnego bać się byłoby zapytanie o *się* w pewnej odległości przed formą leksemu bać , ale bez znaku interpunkcyjnego pomiędzy tymi formami, lub bezpośrednio za taką formą, ewentualnie oddzielone od formy bać zaimkiem osobowym:

```
[orth="się"][!orth="[.!?,:]"]{0,5}[lemma="bać"] | [lemma="bać"][lemma="on|ja|ty|my|wy"]?
↪[orth="się"]
```

Zapytania o znaczniki morfosyntaktyczne

Powyższe zapytanie można uprościć poprzez zastąpienie warunku `orth!="[.!?,:]"` bezpośrednim odwołaniem do „części mowy” `PUNCT`:

```
[orth="się"][!upos="PUNCT"]{0,5}[lemma="bać"] | [lemma="bać"][lemma="on|ja|ty|my|wy"]?
↪[orth="się"]
```

Ogólniej, wartościami atrybutu `upos` (*universal part of speech*) są skróty nazw klas gramatycznych omówionych w dokumentacji Universal Dependencies. Na przykład zapytanie o sekwencję dwóch form rzeczownikowych rozpoczynających się na *a* może być sformułowane w sposób następujący:

```
[upos="NOUN" & orth="a.*"]{2}
```

Podobnie jak to miało miejsce w wypadku specyfikacji form obu warstw tekstowych i form hasłowych, także specyfikacje klas gramatycznych mogą zawierać wyrażenia regularne.

Dodatkowo za pomocą atrybutu `xpos` można odwołać się w zapytaniu do znacznika specyficznego dla języka. Specyfikacja tego atrybutu również może zawierać wyrażenia regularne. Na przykład w korpusie stworzony w języku czeskim następujące zapytanie:

```
[xpos="NNNS1.*"]
```

wyszuka wszystkie rzeczowniki w rodzaju nijakim w mianowniku liczby pojedynczej. Rzeczowniki o tych samych cechach w polskim korpusie (w potoku Stanzy) znajdzie zapytanie:

```
[xpos="subst:sg:nom:n.*"]
```

W obu wypadkach wartość atrybutu `xpos` jest zakończona wyrażeniem `.*`, ponieważ po wartościach części mowy, liczby, rodzaju i przypadka mogą pojawić się jeszcze wartości innych kategorii uwzględnionych w obu tagsetach.

W zapytaniach można określić nie tylko postać ortograficzną segmentu (za pomocą atrybutu `orth`), formę hasłową (za pomocą `lemma`) i klasę gramatyczną (za pomocą `upos` lub ewentualnie `xpos`), ale także wartości poszczególnych kategorii gramatycznych, np. przypadka czy rodzaju — o ile te kategorie występują w danym języku. W korpusach danego języka można używać atrybutów o nazwie kategorii obecnych w banku drzew zależnościowych w warstwie cech morfosyntaktycznych (UFEATS) dla tego języka. Listę wszystkich kategorii można znaleźć na stronie Universal Dependencies.

A zatem w korpusach dla języków posiadających liczbę gramatyczną możliwe jest zadanie na przykład następujących zapytań:

1. ```
   [number="sing"]
   ```

   znalezione zostaną wszystkie formy w liczbie pojedynczej,
   ```

2. `[upos="NOUN" & number="sing"]`

   znalezione zostaną formy rzeczowników pospolitych w liczbie pojedynczej,

3. `[upos="NOUN" & !gender="fem"]`

   formy rzeczowników pospolitych w rodzaju innym niż żeński (czyli np. dla polskiego, czeskiego czy ukraińskiego: w rodzaju męskim lub nijakim),

4. `[number="sing" & case="(nom|acc)" & gender="masc"]`

   pojedyncze mianownikowe lub biernikowe formy męskie (jeśli w języku są kategorie liczby, przypadka i rodzaju).

Można również stosować zbiorczy atrybut `ufeat` w zastępstwie każdej innej nazwy kategorii. Ujednoznacznienie dokona się przez odpowiednią wartość. Dlatego następujące dwa zapytania zwrócą te same wyniki:

```
[upos="NOUN" & case="acc" & number="plur" & gender="fem"]
```

```
[upos="NOUN" & ufeat="acc" & ufeat="plur" & ufeat="fem"]
```

Graficzny konstruktor zapytań

Do tworzenia podstawowych zapytań o sekwencje segmentów można użyć prostego graficznego konstruktora. W oknie konstruktora można definiować warunki określające cechy kolejnych segmentów zapytania, np. część mowy (UPOS), postać segmentu w obu warstwach tekstowych, formę hasłową, a także wartości wszystkich kategorii gramatycznych opisanych w dokumentacji UD. Poszczególne warunki w obrębie segmentu mogą być łączone operatorami *oraz* (koniunkcja) i *lub* (alternatywa). Po zdefiniowaniu wszystkich segmentów zapytania należy wcisnąć przycisk *Zapisz*, następnie określić dodatkowe parametry wyszukania, np. ograniczenia za pomocą metadanych, i rozpocząć wyszukiwanie. Zbudowane za pomocą konstruktora zapytania pojawi się w pasku wyszukiwania, dzięki czemu można dodatkowo zweryfikować jego poprawność.

Ograniczenie zapytania do zdania lub akapitu

Jednostkami organizacji tekstu w korpusach indeksowanych przez Korpusomat są zdania i akapity. Podział ten można wykorzystać w zapytaniach, na przykład ograniczając dopasowanie do jednego zdania.

Aby ograniczyć zasięg zapytania, należy dopisać do zapytania słowo kluczowe `within`, a po nim `<s/>` lub `<p/>`, w zależności od tego, czy zasięg ma być ograniczony do zdania (ang. *sentence*) czy do akapitu (ang. *paragraph*). Ilustruje to następujący przykład zapytania o zdania, w których forma *się* występuje za formą leksemu być, w odległości co najmniej jednego i nie więcej niż dziesięciu segmentów:

```
[lemma="bać"][!orth="się"]{1,10}[orth="się"] within <s/>
```

Dodatkowo można również na elementy `<s/>` i `<p/>` nałożyć pewne warunki dotyczące tego, czy zawierają segmenty innego typu. Przykładowo, za pomocą następującego zapytania można znaleźć wszystkie wystąpienia czasownika pomocniczego być w czasie przyszłym ograniczone do zdań zawierających formę bezokolicznika:

```
[upos="AUX" & lemma="być" & tense="fut"] within (<s/> containing [verbform="inf"])
```

Intencją takiego zapytania jest odnalezienie (w przybliżeniu) wszystkich wystąpień konstrukcji czasu przyszłego złożonego, w których pojawia się bezokolicznik. Wśród wyników będą oczywiście również takie zdania, w których czas przyszły został utworzony z użyciem formy przeszłej czasownika, a bezokolicznik pełni w zdaniu inną funkcję gramatyczną. Można też sformułować zapytanie odwrotnie i zapytać o zdania, w których forma przeszła w ogóle nie występuje:

```
[upos="AUX" & lemma="być" & tense="fut"] within (<s/> !containing [tense="past"])
```

Pełną listę słów kluczowych, które mogą się pojawić w zapytaniach wyszukiwarki MTAS, można znaleźć w jej dokumentacji, nie wszystkie jednak będą miały sensowne zastosowanie w Korpusomacie.

Oprócz znaczników odnoszących się do elementów struktury tekstu (np. <s/>) istnieją również znaczniki odnoszące się do ich początku i końca. W wypadku <s/> będą to odpowiednio: <s> i </s>. Ich dopasowaniem nie jest żaden segment, ale mogą być użyte w połączeniu z warunkami definiującymi inne segmenty, np. zapytanie:

```
<s> [upos="NUM"]
```

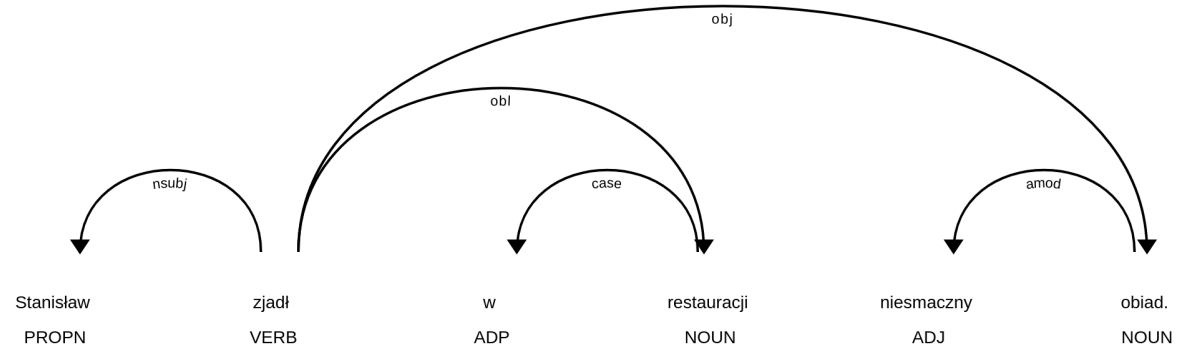
odnajdzie wszystkie wystąpienia liczebnika stojącego na początku zdania. Analogicznie zapytanie:

```
[upos="NUM"][upos="PUNCT"]</s>
```

odnajdzie wszystkie wystąpienia ciągu składającego się z liczebnika i znaku interpunkcyjnego stojących na końcu zdania.

Warstwa składniowa

Kolejną warstwą znakowania w Korpusomacie jest parsowanie zależnościowe. Wprowadzony przez użytkownika tekst jest automatycznie dzielony na wypowiedzenia, które z kolei są poddawane pełnej analizie składniowej w aparacie zależnościowym według zasad przyjętych w projekcie Universal Dependencies. Przykład takiej analizy znajduje się na poniższym rysunku.



MTAS nie jest wyszukiwarką struktur składniowych, nie pozwala zatem na indeksowanie i przeszukiwanie pełnych rozbiorów zdań. Jednak na poziomie każdego segmentu w tekście Korpusomat indeksuje informację o jego bezpośrednim nadrzędniku składniowym (tzn. jego formie hasłowej i klasie fleksyjnej) oraz o typie relacji zależności łączącej oba te elementy w wypowiedzeniu. Ponadto indeksuje również ich położenie względem siebie w wypowiedzeniu: kolejność w porządku linearnym oraz odległość (liczoną w segmentach). Pozwala to na łatwe wyszukanie w korpusie prostszych konstrukcji składniowych oraz analitycznych nieciągłych form fleksyjnych.

W warstwie znakowania składniowego dostępne są następujące atrybuty:

- `deprel` — typ zależności, jaką dany segment jest związany ze swoim bezpośrednim nadrzędnikiem składniowym; wartością tego atrybutu może być jedna z 65 relacji zależności przewidzianych w specyfikacji Universal Dependencies (nie wszystkie muszą jednak wystąpić w rozbiorach zdań w każdym języku),
- `head.upos` — część mowy (UPOS) bezpośredniego nadrzędnika segmentu,

- `head.lemma` — forma hasłowa bezpośredniego nadrzędnika segmentu,
- `head.ufeat` — wartość dowolnej cechy morfologicznej bezpośredniego nadrzędnika segmentu,
- `head.distance` — odległość bezpośredniego nadrzędnika segmentu,
- `head.position` — położenie (lewo- lub prawostronne) bezpośredniego nadrzędnika względem segmentu w porządku linearnym wypowiedzenia.

Dzięki rozszerzeniu języka zapytań o powyższe atrybuty można np. łatwo znaleźć wszystkie rzeczowniki pospolite użyte w funkcji dopełnienia bliższego konkretnego czasownika:

```
[upos="NOUN" & deprel="obj" & head.lemma="kupić"]
```

Możliwe jest również odwrotne wyszukanie odpowiadające na pytanie, przy jakich czasownikach w roli dopełnienia występuje w korpusie konkretny rzeczownik:

```
[deprel="obj" & head.upos="VERB" & lemma="osoba"]
```

Należy jednak zwrócić uwagę, że w powyższym przykładzie wynikiem zapytania będą wystąpienia rzeczownika osoba, nadrzędne względem nich formy czasownikowe (finitywne i niefinitywne) będą się zaś znajdowały w lewym lub prawym kontekście wyników wyróżnione pismem pogrubionym. Można je jednak zgrupować i posortować względem ich częstości dzięki opcjom Statystyk.

Dzięki atrybutowi kodującemu lewo- i prawostronną pozycję nadrzędnika względem segmentu można znaleźć przykłady niekanonicznego szyku zdania, np. podmiotu po orzeczeniu:

```
[deprel="nsubj" & head.position="left"]
```

lub dopełnienia bliższego przed orzeczeniem:

```
[deprel="obj" & head.position="right"]
```

Podobnie w wypadku innych konstrukcji — brak określenia pozycji nadrzędnika w zapytaniu:

```
[upos="ADJ" & deprel="amod" & head.lemma="zupa"]
```

zwróci wszystkie przymiotnikowe określenia rzeczownika zupa. Dodanie parametru pozycji pozwoli ograniczyć wyszukanie do określeń lewostronnych (np. *gorąca zupa*) lub prawostronnych (np. *zupa pomidorowa*).

Częściowa anotacja składniowa pozwala na odnalezienie elementów wypowiedzenia połączonych ze sobą bezpośrednią relacją zależności bez względu na to, czy sąsiadują one ze sobą w porządku linearnym, czy też są przedzielone innymi elementami wypowiedzenia. Atrybut odległości pozwala np. na ograniczenie wyników tylko do takich przypadków, w których elementy nie sąsiadują ze sobą:

```
[deprel="obj" & head.upos="VERB" & tense="past" & !head.distance="1"]
```

Powyższe przykładowe zapytanie wyszuka dopełnienia bliższe orzeczenia w czasie przeszłym, które są oddzielone od tego orzeczenia co najmniej jednym elementem.

Innym przykładem użycia znakowania składniowego w korpusie może być zapytanie wyszukujące konstrukcje w stronie biernej:

```
[upos="AUX" & deprel="aux:pass" & head.upos="ADJ"]
```

którego dopasowaniem są słowa posiłkowe konstrukcji biernej połączone z formą imiesłowu biernego (oznaczoną jako przymiotnik) relacją `aux:pass`.

Warstwa jednostek nazewniczych

Ostatnią warstwą informacji znakowaną w Korpusomacie jest warstwa jednostek nazewniczych (ang. *named entities*). Są to jednostki tekstowe jedno- lub wielowyrazowe nazywające osoby, miejsca, instytucje czy momenty czasowe. Ponieważ nie istnieje międzynarodowy standard i wielojęzyczny zestaw danych oznakowanych, w których oznakowano w spójny sposób jednostki nazewnicze, zbiór wartości i ich zakres różni się w poszczególnych potokach przetwarzania oraz może być różny dla różnych języków w obrębie tych potoków. Ponadto, nie dla wszystkich języków istnieją odpowiednie modele do oznaczania jednostek nazewniczych.

Najprostszy i dość często stosowany zestaw etykiet jednostek nazewniczych składa się tylko z czterech elementów: `PER` (osoba), `LOC` (miejsce), `ORG` (organizacja) i `MISC` (inne), ale dla niektórych języków istnieją bardziej szczegółowe klasyfikacje, np. języki chiński i angielski w potoku Stanzy mają 18 wartości klasyfikacji jednostek nezewniczych. W poniższych przykładach stosuje się powyższą najprostszą klasyfikację, która dostępna jest np. w potoku Stanzy dla języków hiszpańskiego, francuskiego, rosyjskiego czy ukraińskiego. Pełną listę wartości klasyfikacji dla danego korpusu użytkownik znajdzie w graficznym konstruktorze zapytań.

Jednostki nazewnicze, podobnie jak opisane wyżej zdania i akapity, przekraczają granicę segmentu, więc można się do nich odnosić w zapytaniach korpusowych tak samo jak do zdań, za pomocą znacznika `<ne />`. Obowiązują również te same zasady dotyczące znaku ukośnika wewnątrz znacznika:

- `<ne>` oznacza początek ciągu opisanego jako jednostka nazewnicza,
- `</ne>` oznacza koniec ciągu opisanego jako jednostka nazewnicza.

Najprostsze możliwe zapytanie tego typu ma postać:

```
<ne />
```

i zwróci wszystkie jednostki nazewnicze wszystkich typów odnalezione w korpusie. Wyszukanie można ograniczyć do konkretnego typu nazw np. nazw miejsc:

```
<ne="LOC" />
```

Podobnie jak w wypadku zdań i akapitów, zapytania o jednostki nazewnicze można łączyć z cechami ortograficznymi i morfosyntaktycznymi segmentów, z których są one zbudowane lub klasyfikacją nazewniczą ich elementów składowych. Oto kilka przykładów takich zapytań:

```
[upos="CCONJ"] within <ne="PER" />
```

— wszystkie nazwy organizacji zawierające spójnik współrzędny, np. *Krajowa Rada Radiofonii i Telewizji* czy *Instytut Meteorologii i Gospodarki Wodnej*,

```
<ne="LOC" /> [upos="CCONJ"] <ne="LOC" />
```

— wystąpienia dwóch nazw geograficznych połączonych spójnikiem współrzędnym, np. *Europa Zachodnia lub Skandynawia*,

```
[orth="A.*"][orth="M.*"] fullyalignedwith <ne="PER" />
```

— dwa kolejne segmenty, z których pierwszy zaczyna się od *A*, drugi zaś od *M* i które w całości w tekście występują jako nazwa osoby, np. *Adam Michnik*, *Antoni Macierewicz*.

Ponadto indeksowany jest również atrybut zawierający informację o długości jednostki nazewniczej mierzonej w segmentach. Zapytanie:

```
<ne.len="3" />
```

odnajdzie wszystkie takie jednostki składające się z dokładnie trzech segmentów.

Ograniczenie zapytania za pomocą metadanych

Teksty wprowadzane przez użytkownika do Korpusomatu są domyślnie opatrywane czterema polami metadanych o etykietach: autor, tytuł, rok wydania, gatunek. Od użytkownika zależy to, w jaki sposób zostaną one wypełnione, w szczególności mogą pozostać puste. Użytkownik może też zdefiniować własne pola o dowolnych etykietach.

Pól metadanych można użyć następnie do ograniczenia zasięgu zapytań w wyszukaniach korpusowych. Służy do tego przycisk metadane, pod którym można zdefiniować takie ograniczenia. Można nałożyć wiele ograniczeń jednocześnie, dodając je za pomocą przycisku dodaj ograniczenie.

1.3 Profile słów

1.3.1 Wprowadzenie

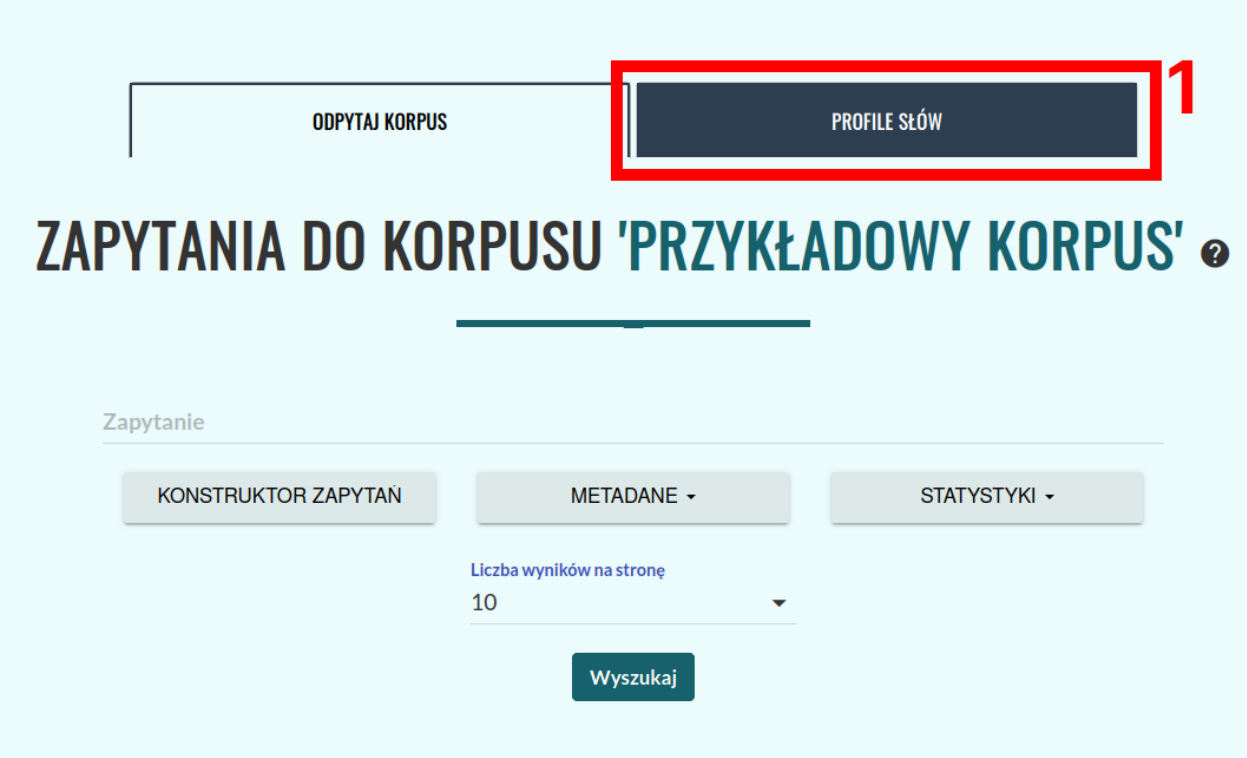
Profile słów umożliwiają odnalezienie w tekście słownictwa, które często łączy się ze wskazanym słowem w związki składniowe określonego rodzaju. Na przykład rzeczownik *oczy* często jest modyfikowany przez przymiotnik *niebieskie* i często jest dopełnieniem bliższym czasownika *zamknąć*. Z kolei rzeczownik *pies* często pojawia się w koordynacji z rzeczownikiem *kot*. Otrzymane kolokacje charakteryzują język korpusu, tj. w korpusie reprezentatywnym dla standardowego języka polskiego będą się pojawiały głównie związki wynikające z ogólnych zależności semantycznych lub frazeologii, natomiast w korpusie dziedzinowym — związki wywodzące się z języka danej dziedziny, związki charakteryzujące styl autora lub jego sposób myślenia. Na przykład w korpusie ogólnym słowo *funkcja* będzie często określane przymiotnikiem *podstawowa*, zaś w korpusie matematycznym częściej pojawi się przymiotnik *ciągła* lub *różnowartościowa*. Można się też spodziewać, że przymiotnik *robotniczy* będzie występował z innymi kolokatami w korpusie z czasów PRL, a z innymi w korpusie współczesnym.

Informacja: Profile słów są dostępne wyłącznie dla korpusów posiadających warstwę anotacji zależnościowej.

Należy zauważyć, że ze względów statystycznych funkcja ta najlepiej działa dla korpusów stosunkowo dużych (od 1 miliona segmentów), oraz słów pojawiających się w danym korpusie relatywnie często.

Obliczenie profilu danego słowa może potrwać od kilku do kilkudziesięciu sekund, w zależności od wielkości korpusu i częstości słowa.

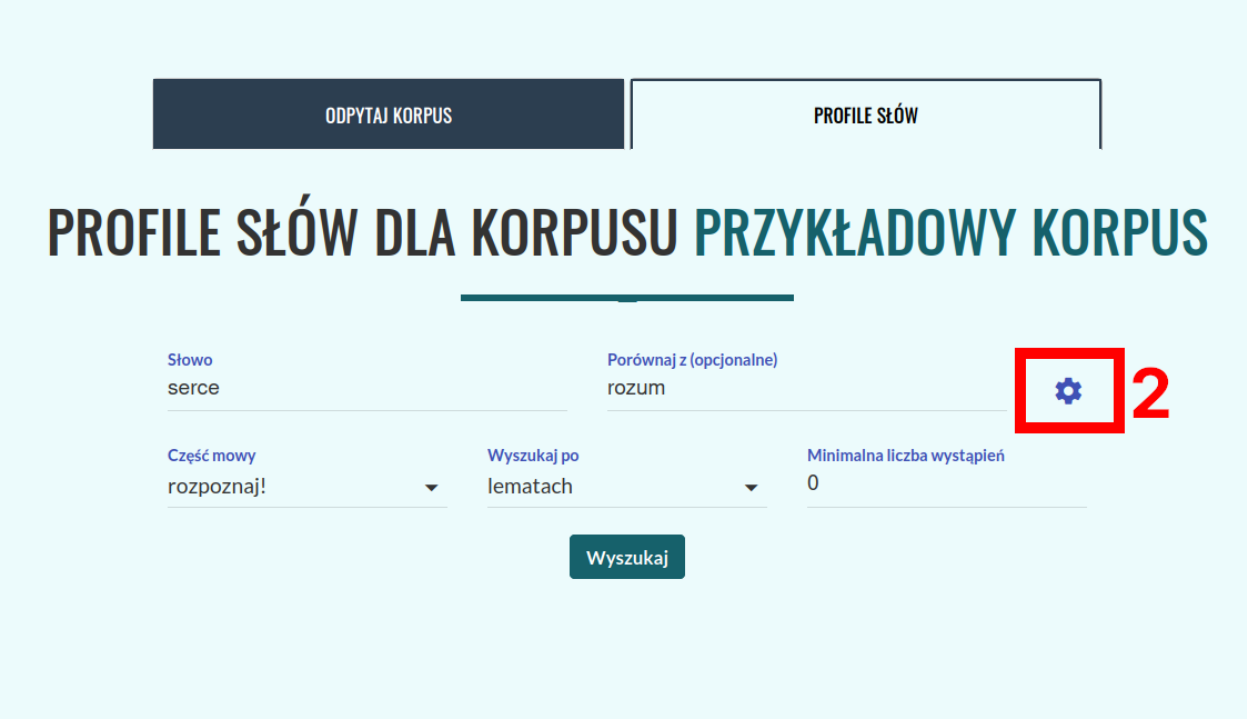
1.3.2 Korzystanie



Profile słów są dostępne z poziomu ekranu *Odpytaj korpus*, karta *Profile słów* (**1** na obrazku). W pole *Słowo* należy wpisać słowo, którego profil chcemy wyliczyć.

Zaawansowane opcje wyszukiwania

Po kliknięciu w ikonę koła zębatego (**2**) dostępne są również zaawansowane opcje wyszukiwania. Tworząc profil danego słowa, możemy wybrać, czy interesują nas wszystkie jego wystąpienia, niezależnie od formy w tekście (i.e. szukamy wg lematu), czy też chcemy zobaczyć jedynie kolokaty określonej formy danego leksemu (np. rzeczownika *psy*, a więc słowa w liczbie mnogiej, i mianowniku lub bierniku). Możemy też odfiltrować kolokaty, ustawiając minimalną liczbę wspólnych wystąpień w korpusie, ta funkcja pozwala ominąć pary, które rzadko się powtarzają, a uzyskały wysoki wynik ze względu na rzadkość ich poszczególnych elementów składowych w korpusie.



Formularz pozwalający doprecyzować parametry profilu słów: narzucić określoną interpretację pod względem klasy gramatycznej, określić, czy interesują nas wystąpienia wskazanej formy czy wszystkich form przynależących do danego leksemu, zastosować filtrowanie frekwencyjne lub słowo kontrastowe.

W wyniku otrzymujemy tabelę, z której każda kolumna odpowiada jednemu z typów związków syntaktycznych w jakie może wchodzić wskazane słowo. Dane w każdej z kolumn reprezentują ranking kolokatów, każdy z takich rankingów jest niezależny od pozostałych.

Informacja: Domyślna wartość minimalnej liczby wystąpień kolokatów zależy od wielkości korpusu – dla korpusów mających co najmniej 100 tysięcy segmentów wynosi 3, natomiast dla mniejszych korpusów – 0.

Profile porównawcze

Aplikacja umożliwia także tworzenie profili porównawczych. W tym celu należy wpisać do pola *porównaj z* drugie z interesujących nas słów. Wyszukiwanie porównawcze zakłada, że zadane słowa należą do tej samej klasy gramatycznej. Przygotowując tabelę, aplikacja weźmie pod uwagę różnicę wartości **logDice** słowa podstawowego, oraz słowa porównawczego dla każdego z kolokatów. Tabela jest automatycznie skracana do postaci w której ekstrahowane są trzy sekcje. Kolokaty wyraźnie preferujące pierwsze słowo, kolokaty neutralne (o wartościach różnicy logDice najbliższych 0) oraz kolokaty wyraźnie preferujące słowo porównawcze. Indeksy wierszy wpadających do każdej z tych sekcji są oznaczone innym kolorem.

WYNIKI WYSZUKIWANIA PORÓWNAWCZEGO DLA SŁÓW SERCE I ROZUM JAKO RZECZOWNIK POSPOLITY (NOUN)
SŁOWA TE POJAWIAJĄ SIĘ W KORPUSIE ODPOWIEDNIO 24 I 229 RAZY

Widoczne kolumny

Szukaj:

	słowa których podmiotem jest "serce" vs. "rozum"	słowa których dopełnieniem bliższym jest "serce" vs. "rozum"	słowa których dopełnieniem dalszym jest "serce" vs. "rozum"	słowa których modyfikatorem jest "serce" vs. "rozum"	apozycje "serce" vs. "rozum"	słowa z którymi "serce" vs. "rozum" występuje w koordynacji	modyfikatory przymiotnikowe słowa "serce" vs. "rozum"	czasownikowe modyfikatory słowa "serce" vs. "rozum"	modyfikatory nominalne słowa "serce" vs. "rozum"	słowa których modyfikatorem nominalnym jest "serce" vs. "rozum"	określniki słowa "serce" vs. "rozum"
1			sądzić VERB 7.956	unosić ADJ 10.3		znosić VERB 10.272	złoty ADJ 9.715	żyć VERB 8.0		wykonawca NOUN 9.574	
2	umieścić VERB 10.093	ofuknąć VERB 11.3	przychylić VERB 10.356	słuchać VERB 8.492		strzymać VERB 10.3	gorący ADJ 10.0	ścisnąć ADJ 10.245		ręka NOUN 8.591	
3	bronić VERB 9.3	obłożyć VERB 10.245	brak VERB 11.046	prowadzić VERB 9.023		skóra NOUN 9.642	dumny ADJ 10.356	znaleźć VERB 7.923	uczeń NOUN 9.046	odruch NOUN 10.093	
4	taki DET -4.992	rodzaj NOUN -6.043	mieć VERB -2.926	dojść VERB 1.974		to PRON -0.505	ludzki ADJ 1.295	mieć VERB -4.297	on PRON -3.983	człowiek NOUN 0.303	który DET -3.909
5	musieć VERB -5.937	rozumieć VERB -6.578	mówić VERB -4.56	robić VERB -5.989		jeden ADJ -4.919	mały ADJ -0.007	wydawać VERB -5.942	nic PRON -5.915	Sokrates PROPN -3.746	ten DET -5.294
6	móc VERB -6.282	przyjąć VERB -6.665	robić VERB -5.989	iść VERB -6.385		rzecz NOUN -5.11	własny ADJ -6.382	zło NOUN -6.536	rozkosz NOUN -6.008	taki DET -4.992	wszystek DET -5.719
7	dusza NOUN -5.382	kochać VERB -6.972	dostać VERB -7.907	donieść VERB -7.142	przyjaciel NOUN -6.578	cześć NOUN -6.945	jeden ADJ -4.919	należeć VERB -6.382	drugi ADJ -5.287	cel NOUN -6.565	pewien DET -7.096
8	dostać VERB -6.907	dowodzić VERB -7.081	dodać VERB -7.063	bać VERB -6.795	pierwiastek NOUN -6.855	ależ INTJ -6.891	drugi ADJ -5.287	dobro NOUN -5.929	domieszka NOUN -7.099	brak NOUN -8.492	jakiś DET -6.658
9	być VERB -6.329	chronić VERB -7.13	dobry ADJ -5.385	bardzo ADV -5.636		Sokrates PROPN -4.746	cudzy ADJ -7.099	ciemny ADJ -7.117	człowiek NOUN -4.919	akt NOUN -7.075	czyj DET -7.117

Pozycje od 1 do 9 z 9 łącznie

Tabela wynikowa, dla profilu porównawczego: *serce* vs. *rozum* w korpusie dialogów Platona.

Kliknięcie każdego z kolokatów, wygeneruje wyrażenie wyszukiwawcze które pozwoli odnaleźć wszystkie wspólne wystąpienia obu terminów w korpusie.

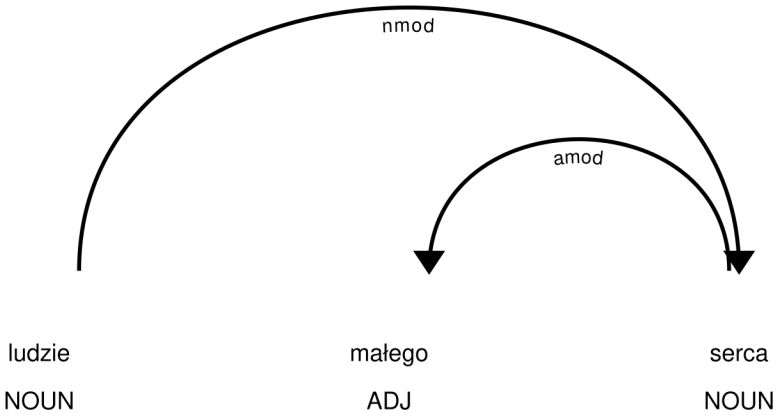
1.3.3 Wykorzystane miary

Profile słów przedstawiają słownictwo często współwystępujące ze wskazanym słowem. Znaczenie słowa *często* jest tutaj formalizowane za pomocą miary **logDice** (i to te wartości są widoczne w tabeli). Miara ta przypisuje każdej z badanych par słów wynik będący – w pewnym uproszczeniu – stosunkiem liczby wystąpień w korpusie razem do sumy wystąpień w korpusie w ogóle (razem lub osobno) każdego ze słów. W ten sposób odfiltrowujemy takie słowa, które pojawiają się obok zadanego słowa często w wyniku tego, że same są bardzo częste (np. czasownik *mieć*, w odróżnieniu od czasownika *zamykać*).

Miara **logDice**, w odróżnieniu od innych miar stosowanych do ekstrakcji kolokacji, jest interpretowalna: maksymalnie osiąga wartość 14 (gdy słowa współwystępują zawsze), zaś różnica między wartościami wielkości 1 oznacza, że jedna z kolokacji jest dwukrotnie częstsza niż druga. Wartość **logDice** jest też niezależna od wielkości korpusu (można więc porównywać wartości otrzymane dla różnych korpusów).

1.3.4 Podstawa lingwistyczna

Profile słów są obliczane na podstawie znakowania morfologicznego i składniowego (zależnościowego), dlatego funkcja ta dostępna jest wyłącznie dla korpusów posiadających warstwę anotacji zależnościowej. Dla każdej z obsługiwanych części mowy przygotowano ręcznie zestaw reguł pozwalających odnaleźć potencjalne kolokaty danego słowa. Na przykład dla rzeczowników reguły odnajdują w korpusie czasowniki, których dany rzeczownik jest podmiotem (*pracownik wykonuje*), dopełnieniem bliższym (*zwolnił pracownika*), lub rzeczowniki modyfikowane przez dany rzeczownik (*rynek pracownika*). Zestaw reguł jest domyślnie dobierany na podstawie klasy morfosyntaktycznej zadanego słowa, rozpoznanej przez aplikację automatycznie, natomiast możliwe jest także narzucenie określonej interpretacji (np. słowu *wieść* jako rzeczownik, a nie czasownik). W wykazie kolokacji wystąpienia pojawiają się w formie zlematyzowanej.

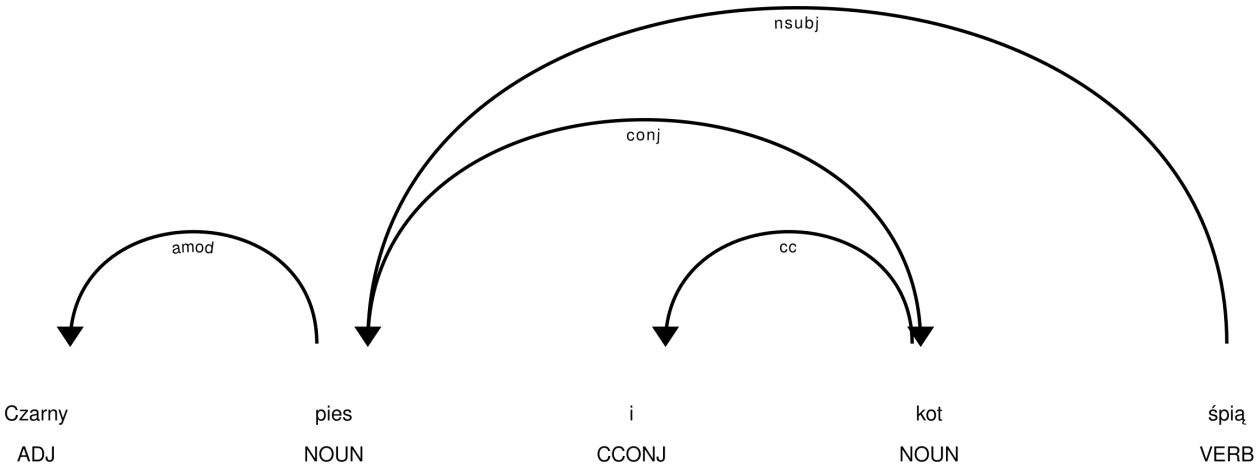


W obliczaniu profili słów wykorzystywane są drzewa zależnościowe, takie jak fragment przedstawiony na obrazku. Jeżeli interesuje nas słowo *serce*, do kolumny *przymiotniki modyfikujące „serce”* trafi słowo *mały*, zaś do kolumny *słowa których modyfikatorem nominalnym jest „serce”*, słowo *człowiek*.

Należy zwrócić uwagę na to, że kolokacje nie są liczone w sposób uwzględniający negacje. Wystąpienia danego słowa będą zaliczane do tego samego kolokatu niezależnie od tego, czy są w zasięgu modyfikatorów negujących (np. słowo *nie*), spójników takich jak *ani*, modyfikatorów leksykalnych o charakterze zbliżonym do negacji (jak np. *mało* w *mało przystojny*), albo wreszcie same są formą zanegowaną (np. imiesłów *niepoinformowany*).

Związki koordynacji

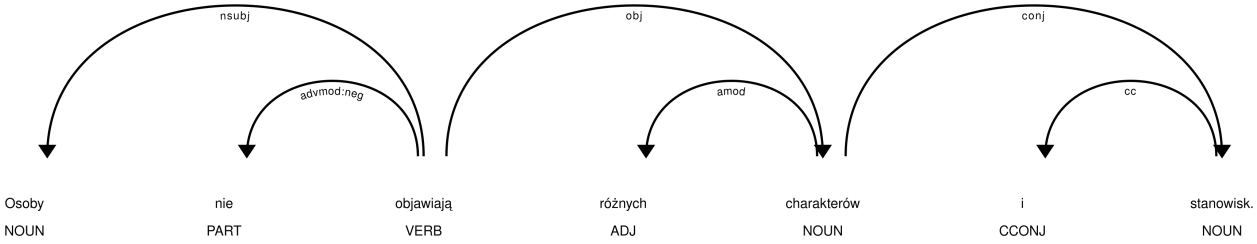
Jedną z najistotniejszych relacji, które można wziąć pod uwagę, są związki o charakterze współrzędnym — koordynacja. W wykorzystanym formalizmie gramatycznym koordynację reprezentuje się jako poddrzewo, którego nadrzędnikiem jest pierwszy z członów koordynacji, liśćmi zaś — pozostałe człony. Nadrzędnik poddrzewa łączy się ze swoim nadrzędnikiem relacją, którą pełniłby, gdyby występował jako pojedyncze wyrażenie, pozostałe człony natomiast są opatrzone etykietami relacji *conj*. Ewentualne spójniki połączone są z liśćmi poddrzewa relacją *cc*. Na przykład w zdaniu *Czarny pies i kot śpią.* słowo *pies* łączy się z czasownikiem *śpią* relacją *nsubj*, słowo *kot* jest podrzędnikiem słowa *pies* i słowa te łączy relacja o etykiecie *conj*, natomiast *i* łączy się ze słowem *kot* jako spójnik łączący, przyjmuje więc etykietę *cc*, jak w poniższym przykładzie.



Aby umożliwić rozpoznawanie słów występujących jako drugi lub kolejny człon koordynacji, w zastosowanym systemie ekstrakcji kolokatów traktujemy poddrzewa koordynacji w sposób szczególny. W ramach przeszukiwania drzewa niejako przeskakuje się przez pierwszy z członów koordynacji, czyli korzeń poddrzewa. Analizując wskazany wyżej przykład, w poczet podmiotów czasownika *spać* zostaną zaliczone rzeczowniki *pies* i *kot*. Obliczając zaś listę czasowników, których podmiotem jest słowo *kot* (tj. idąc w górę drzewa), przejdziemy w drzewie dwa kroki, dzięki czemu zaliczymy wystąpienie czasownika *spać*.

Warto zaznaczyć, że mechanizmem tym objęte są jedynie niektóre z relacji. W powyższym przykładzie słowo *czarny* nie zostanie uwzględnione jako modyfikator przymiotnikowy słowa *kot*. Następujące relacje uwzględniają wyżej opisany mechanizm rozszerzania koordynacji:

- dla rzeczowników (NOUN/PROPN):
  - *słowa których podmiotem jest (…)*,
  - *słowa których dopełnieniem bliższym jest (…)*,
  - *słowa których dopełnieniem dalszym jest (…)*.
- dla czasowników (VERB):
  - *słowa które są podmiotem (…)*,
  - *słowa które są dopełnieniem bliższym (…)*,
  - *słowa które są dopełnieniem dalszym (…)*,
  - *słowa które są podmiotem zdaniowym (…)*,
  - *słowa których podmiotem zdaniowym jest (…)*,
  - *słowa które są dopełnieniem zdaniowym (…)*,
  - *słowa których dopełnieniem zdaniowym jest (…)*.



Reprezentacja koordynacji oraz negacji w zastosowanym formalizmie składniowym. Ponieważ bezpośrednim podrzędnikiem relacji *obj* jest pierwszy z członów koordynacji, słowo *charakter* zostanie włączone w poczet

dopełnień bliższych „objawiać”. Podczas obliczeń przeskakujemy dodatkowo o poziom niżej po relacjach z etykietą *conj*, aby uwzględnić również słowo *stanowisko*.

1.4 Statystyki korpusu

1.4.1 Lista frekwencyjna

W tym polu wyświetlana jest lista frekwencyjna ze względu na lematy słów. Wstępnie odfiltrowana jest do słów pełnoznacznych, jednak za pomocą przycisku na górze można ją odfiltrować do dowolnie wybranych części mowy. Po kliknięciu przycisku „Pobierz” zostanie pobrana pełna lista frekwencyjna bez zastosowanego filtrowania.

Miary dyspersji i skorygowanej frekwencji

Oprócz standardowej listy frekwencyjnej w polu „Lista frekwencyjna” wyświetlane są również miary dyspersji oraz skorygowanej frekwencji słów.

Poniższa lista przedstawia wszystkie dostępne miary. W podanych wzorach przyjęto następujące oznaczenia:

- N - liczba tekstów w korpusie.
- N_t - liczba tekstów, w których występuje dane słowo.
- L – liczba słów w korpusie.
- L_i – liczba słów w i-tym tekście korpusu.
- F - liczba wystąpień słowa w całym korpusie.
- F_i – liczba wystąpień słowa w i-tym tekście korpusu.
- $\bar{F}$ - średnia liczba wystąpień danego słowa w tekście korpusu: $\bar{F} = \frac{\sum_{i=1}^{N} F_i}{N}$.
- P_i – stosunek liczby wystąpień danego słowa do liczby wszystkich słów w i-tym tekście korpusu.
- $\bar{P}$ - średnia częstość wystąpień danego słowa w tekście korpusu: $\bar{P} = \frac{\sum_{i=1}^{N} P_i}{N}$.
- S_i – stosunek liczby słów w i-tym tekście korpusu do liczby słów we wszystkich tekstach korpusu: $S_i = \frac{L_i}{L}$

Miary dyspersji i skorygowanej frekwencji:

- Frekwencja – liczba wystąpień danego słowa w korpusie.

$$F$$

- *IDF* – miara ilości informacji wiążącej się z użyciem danego słowa. Przyjmuje wartość 0, jeśli słowo występuje we wszystkich tekstach korpusu, i wartości tym większe im mniejsza jest liczba tekstów, w których występuje słowo.

$$log_2 \frac{N}{N_t}$$

- Współczynnik zmienności (*vc*).

$$\sigma_F = \sqrt{\frac{\sum_{i=1}^{N}(F_i - \bar{F})^2}{N}}$$

$$vc = \frac{\sigma_F}{\bar{F}}$$

- *D* i *U* Juillanda – miara dyspersji (*D*) i skorygowanej frekwencji (*U*). Miary obliczane są przy pomocy skorygowanej formuły, która uwzględnia różne objętości tekstów należących do korpusu. *D* przyjmuje wartości z przedziału $[0, 1]$, natomiast *U* – z przedziału $[0, F]$.

$$\sigma_P = \sqrt{\frac{\sum_{i=1}^{N}(P_i - \bar{P})^2}{N}}$$

$$D_{adj} = 1 - \frac{\sigma_P}{\bar{P}\sqrt{N-1}}$$

$$U = D_{adj} \cdot F$$

Uwaga: odchylenie standardowe wykorzystane do obliczania D i U Juillanda obliczane jest przy pomocy innej formuły, niż w przypadku współczynnika zmienności.

- *DP* Griesa (odwrotne *DP*, standaryzowane *DP*) – miary dyspersji. Oprócz standardowej miary *DP* obliczane są wartości odwrotnego *DP* (DP_{rev}), które łatwiej można porównać z wartościami innych metryk dyspersji, a także standarydowanego *DP* (DP_{norm}), wartości którego można porównać z korpusami składającymi się z innej liczby tekstów. Wyższe wartości *DP* i standaryzowanego *DP* (lub niższe wartości odwróconego *DP*) reprezentują bardziej nierównomierne występowanie słowa w korpusie. *DP* przyjmuje wartości z przedziału $[0, 1 - min\,\{S_i\}_{i=1}^{N}]$, standaryzowane *DP* – z przedziału $[0, 1]$, a odwrotne *DP* – z przedziału $[min\,\{S_i\}_{i=1}^{N}, 1]$.

$$DP = \frac{\sum_{i=1}^{N} |\frac{F_i}{F} - S_i|}{2}$$

$$DP_{rev} = 1 - DP$$

$$DP_{norm} = \frac{DP}{1 - \frac{1}{N}}$$

- *D2* i *Um* Carolla – miara dyspersji (*D2*) i skorygowanej frekwencji (*Um*). Miara *D2* przyjmuje wartości z przedziału $[0, 1]$, przy czym wyższe wartości *D2* świadczą o bardziej równomiernej dystrybuji słowa w korpusie. *Um* przyjmuje wartości z przedziału $[\frac{F}{N}, F]$.

$$D_2 = \frac{-\sum_{i=1}^{N}\left(\frac{P_i}{\sum_{i=1}^{N} P_i} \cdot \log_2 \frac{P_i}{\sum_{i=1}^{N} P_i}\right)}{\log_2 N}$$

$$U_m = F \cdot D_2 + (1 - D_2) \cdot \frac{F}{N}$$

- Rozbieżność KL (*KLD*) – niesymetryczna miara różnic w rozkładach prawdopodonieństwa, używana jako miara dyspersji. Przy obliczaniu wartości tej metryki przyjmuje się, że
$log_2 0 = 0$. Miara przyjmuje nieujemne wartości – tym wyższe, im mniej równomiernie dane słowo występuje w korpusie.

$$KLD = \sum_{i=1}^{N}\left(\frac{F_i}{F} \cdot \log_2\left(\frac{F_i}{F} \cdot \frac{1}{S_i}\right)\right)$$

- *S* i *AF* Roesengrena – miara dyspersji (*S*) oraz frekwencji (*AF*). Miary obliczane są przy pomocy skorygowanej formuły, która uwzględnia różnie objętości tekstów należących do korpusu. *S* przyjmuje wartości z przedziału $[\frac{1}{N}, 1]$, przy czym wyższe wartości odzwierciedlają bardziej równomierną dystrybuję słowa. AF przyjmuje natomiast wartości z przedziału $[\frac{F}{N}, F]$.

$$S_{adj} = \frac{1}{F}\left(\sum_{i=1}^{N}\sqrt{F_i \cdot S_i}\right)^2$$

$$AF = F \cdot S_{adj}$$

- *ARF* – miara zmodyfikowanej frekwencji oparta o odległości pomiędzy wystąpieniami danego słowa. W poniższym wzorze zmienna d_j oznacza dystans między j-tym i j+1-szym wystąpienie słowa (a dla $j = F$ – odległość pomiędzy pierwszym i ostatnim wystąpieniem słowa zakładając, że pierwsze i ostatnie słowo korpusu sąsiadują ze sobą). Miara przyjmuje wartości z przedziału $[1, F]$, tym wyższe, im bardziej równomierna jest dystrybuja słowa w korpusie.

$$ARF = \frac{F}{L} \sum_{i=1}^{F} min \left\{ d_i, \frac{L}{F} \right\}$$

Dla korpusów składających się tylko z jednego tekstu obliczane są jedynie frekwencja oraz ARF.

Dostępne miary zostały opisane w artykule „Dispersions and adjusted frequencies in corpora" (Gries, 2008) oraz rozdziale 5. podręcznika „A Practical Handbook of Corpus Linguistics" (Gries, 2021).

1.4.2 Terminologia

Lista terminów generowana jest przez aplikację TermoPL. Opis jej działania, instrukcja i dodatkowe informacje dostępne są na stronie aplikacji.

W zakładce Terminologii informacje ograniczone są do formy bazowej, wartości C-value oraz liczby wystąpień, posortowane wg C-value. Po kliknięciu przycisku „Pobierz" pobrany zostaje plik txt zawierający wszystkie dane wygenerowane przez TermoPL.

Pliki wygenerowane przez Korpusomat są kompatybilne z aplikacją TermoPL – po pobraniu „plików źródłowych korpusu" (przycisk dostępny na ekranie korpusu) można samodzielnie uruchomić aplikację TermoPL z wybranymi przez siebie opcjami.

1.5 Wykorzystane narzędzia

Korpusomat korzysta z z dwóch wysokopoziomowych bibliotek programistycznych do przetwarzania języków naturalnych spaCy oraz Stanza oraz zbudowanych przez ich twórców modeli dla poszczególnych języków.

Inne wykorzystane narzędzia i zasoby:

- Universal Dependencies,
- Marciniak, M., Mykowiecka, A., & Rychlik, P. (2016). TermoPL - a Flexible Tool for Terminology Extraction. LREC.
- Matthijs Brouwer, Hennie Brugman and Marc Kemps-Snijders 2017. MTAS: A Solr/Lucene based multi tier annotation search solution. Selected papers from the CLARIN Annual Conference 2016. Linköping Electronic Conference Proceedings 136: 19–37.

1.6 Licencja

Niniejsza instrukcję przygotowali Witold Kieraś, Karol Saputa, Łukasz Kobyliński i Ryszard Tuora. Instrukcja jest dostępna na warunkach licencji BY-SA.

Część pt. „Tworzenie zapytań do korpusu" stanowi pochodną „Ściągawki do Narodowego Korpusu Języka Polskiego" (dostępnej na warunkach licencji Creative Commons BY-SA), której autorem jest Adam Przepiórkowski i którą następnie poprawiali i rozszerzali Jakub Wilk i Aleksander Buczyński.

1.7 Cytowanie

W przypadku użycia w pracy naukowej prosimy o zacytowanie artykułu:

Karol Saputa, Aleksandra Tomaszewska, Natalia Zawadzka-Paluektau, Witold Kieraś, and Łukasz Kobyliński. **Korpusomat.eu: A multilingual platform for building and analysing linguistic corpora.** In Jiří Mikyška, Clélia de Mulatier, Maciej Paszynski, Valeria V. Krzhizhanovskaya, Jack J. Dongarra, and Peter M.A. Sloot, editors, Computational Science – ICCS 2023. 23rd International Conference, Prague, Czech Republic, July 3–5, 2023, Proceedings, Part II, number 14074 in Lecture Notes in Computer Science, pages 230–237, Cham, 2023. Springer Nature Switzerland. bibtex doi

1.8 Autorzy

1.8.1 Zespół

Witold Kieraś
Opiekun merytoryczny

Łukasz Kobyliński
Kierownik projektu, autor wersji oryginalnej

Karol Saputa
Główny programista

Sandra Penno
Programista

Filip Koselski
Programista

1.8.2 Dotychczasowi współpracownicy

- Zbigniew Gawłowicz
- Michał Wasiluk
- Agnieszka Olech
- Filip Karpiński
- Marcel Kawski
- Michał Modzelewski
- Kacper Mirowski
- Ryszard Tuora