Korpusomat

Wydanie 0.1

IPI PAN

10 cze 2022

Spis treści

Korpusomat.pl jest aplikacją webową służącą do tworzenia korpusów tekstów, z których można korzystać za pomocą wyszukiwarki MTAS. Aplikacja w zasadzie nie stanowi nowego narzędzia informatycznego, a jedynie łączy istniejące narzędzia, które powstały i wciąż są rozwijane w Zespole Inżynierii Lingwistycznej Instytutu Podstaw Informatyki PAN, a także w innych jednostkach naukowych zajmujących się przetwarzaniem języka polskiego. Zasadniczym celem Korpusomatu jest udostępnienie wyników działań tych narzędzi bez konieczności szczegółowego poznawania technicznej strony ich działania.

Na działanie Korpusomatu składają się:

- analizator morfologiczny Morfeusz stworzony w oparciu o dane lingwistyczne Słownika gramatycznego języka polskiego,
- tager Concraft,
- Liner 2 wykorzystany do rozpoznawania nazw własnych,
- parser COMBO,
- TermoPL,
- wyszukiwarka korpusowa MTAS.

Dwa pierwsze z programów, czyli Morfeusz i Concraft, są wciąż rozwijane i ich nowsze wersje będą sukcesywnie włączane do Korpusomatu. Liner2 to narzędzie do rozpoznawania i znakowania nazw własnych, wyrażeń temporalnych i opisów sytuacji w tekście – w Korpusomacie aktualnie wykorzystywany jest jedynie do znakowania nazw własnych, ale w przyszłości zostaną włączone również inne jego moduły. COMBO jest analizatorem składniowym budującym zależnościowe drzewa analiz składniowych. TermoPL to narzędzie do ekstrakcji terminologii z podanych tekstów – jest używane na ekranie statystyk korpusu. MTAS jest wyszukiwarką korpusową stworzoną przez holenderski Meertens Instituut w ramach projektu Clarin.

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 stosowanym dla języka polskiego kodowaniu, np. ISO-8859-2 czy CP-1250, 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 tutaj.

ROZDZIAŁ 1

Dokumentacja

1.1 Korzystanie z aplikacji

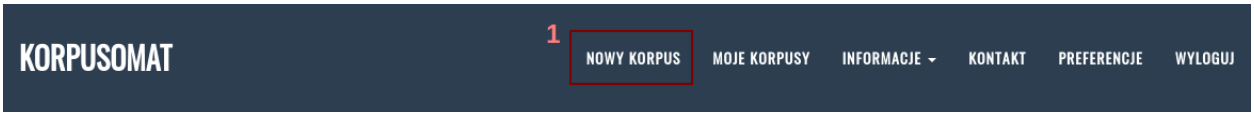
1.1.1 Tworzenie konta

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.

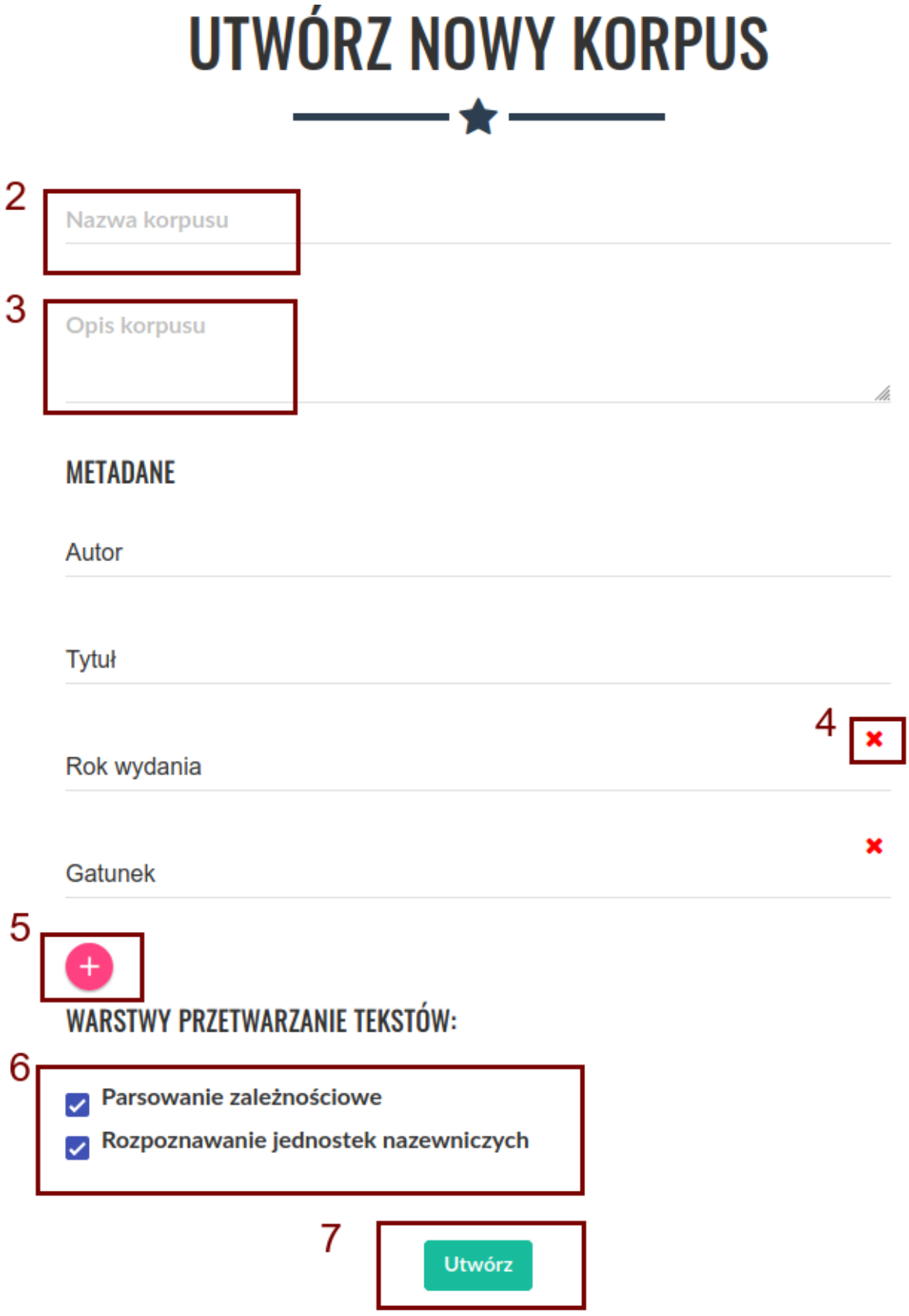
Konto można stworzyć, klikając tutaj lub w przycisk w menu w prawym górnym rogu.

1.1.2 Tworzenie korpusu

Aby utworzyć nowy korpus (po uprzednim zalogowaniu się), należy kliknąć odnośnik „Nowy korpus” (1) z górnego menu.



Następnie należy wybrać nazwę dla korpusu (2) oraz warstwy przetwarzania tekstów (6). W tym miejscu można też dodawać (5) i usuwać (4) nowe metadane dla wszystkich tekstów w korpusie. Autor i Tytuł to metadane obowiązkowe. Dodatkowo można dodać opis korpusu (3). Aby zapisać korpus, należy kliknąć przycisk „Utwórz” (7).


UTWÓRZ NOWY KORPUS
2
Nazwa korpusu
3
Opis korpusu
METADANE
Autor
Tytuł
4
Rok wydania
Gatunek
5
WARSTWY PRZETWARZANIE TEKSTÓW:
6
Parsowanie zależnościowe
Rozpoznawanie jednostek nazewniczych
7
Utwórz

Po utworzeniu korpusu zostaniemy przeniesieni do ekranu „Moje korpusy językowe”. Aby rozpocząć dodawanie tekstów do nowo utworzonego korpusu, należy kliknąć jego nazwę (8) na liście korpusów. Na tym ekranie wyświetlane są także dodatkowe informacje o korpusach, jest tu również możliwość usunięcia niepotrzebnego korpusu oraz edycji istniejących korpusów (9, 10).

Aby dodać nowy tekst do korpusu, należy następnie kliknąć ikonę „+” (11) w prawym dolnym rogu ekranu.

Po kliknięciu zostaniemy przeniesieni do ekranu dodawania tekstu. Lista dozwolonych formatów znajduje się tutaj. Dodać teksty możemy na dwa sposoby.

Pierwszym jest kliknięcie górnego przycisku „+ Dodaj pliki” (12), który pozwala na dodawanie plików z lokalnego dysku. Po kliknięciu pojawi się okno wyboru plików, w którym możemy wskazać jeden lub wiele plików jednocześnie do dodania do korpusu.

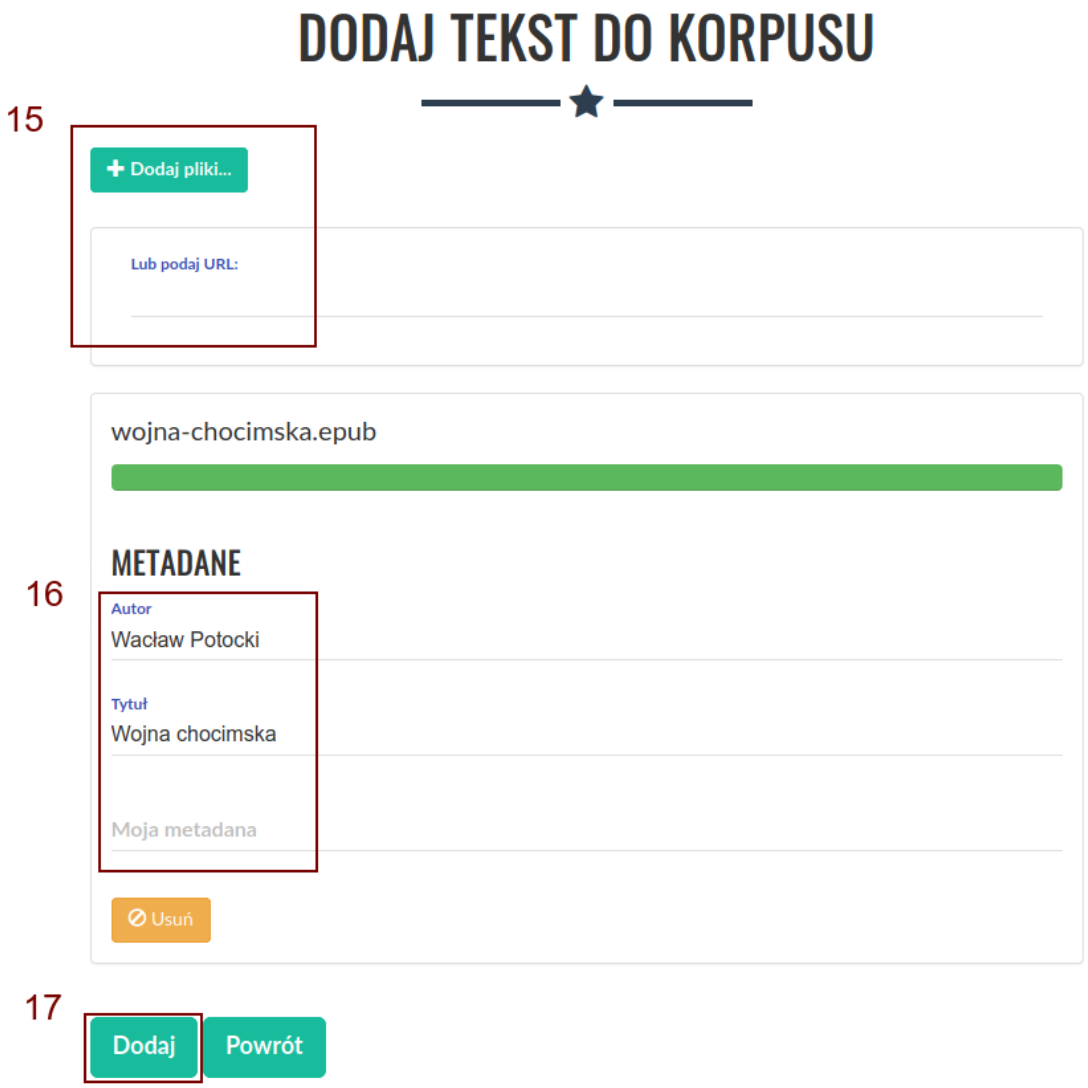
Drugim sposobem jest podanie bezpośrednio linku do tekstu w polu tekstowym „Lub podaj URL:” (13), a następnie kliknięcie przycisku „Pobierz” (14). Korpusomat pobierze wtedy plik automatycznie i przetworzy go. W takim przypadku możliwe jest również podanie linku do artykułu (np z portalu internetowego), z którego zostanie wydobyta treść i przetworzona do pliku txt.

Po przetworzeniu wybranych tekstów istnieje możliwość edycji metadanych (16) lub dodania kolejnych tekstów (15). Korpusomat automatycznie próbuje uzyskać metadane z dodanego pliku, jednak nie zawsze jest to możliwe. Automatyczne rozpoznawanie metadanych zakłada, że nazwy plików są 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". Powyższe dotyczy plików w formatach, które nie zawierają odpowiednich pól przechowujących metadane — nie dotyczy np. plików epub, z których metadane zostaną wyciągnięte wprost z pliku, a nie z jego nazwy.

Przed zatwierdzeniem istnieje możliwość ręcznej edycji metadanych.

Do dodawania kolejnych tekstów służą przyciski na górze (15). Metoda dodawania jest identyczna jak w przypadku pierwszego tekstu.

Gdy wszystkie teksty są już dodane, a ich metadane są poprawnie ustawione, należy kliknąć przycisk „Dodaj" (17) na dole ekranu, aby dodać wybrane teksty do korpusu.



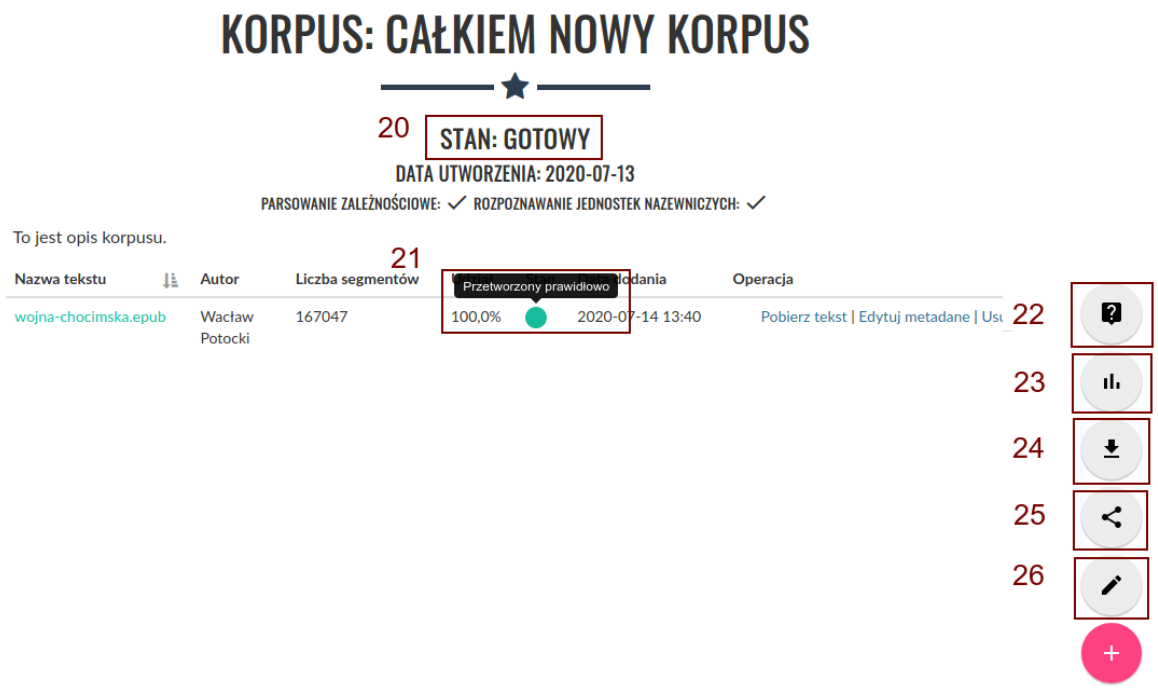
Po dodaniu tekstów zostaniemy przeniesieni do ekranu korpusu, a Korpusomat zacznie przetwarzanie tekstów w wybranych warstwach anotacji. Przy nazwie korpusu pojawi się stan korpusu, data jego utworzenia oraz informacje o warstwach przetwarzania tekstów (18). Przy każdym z tekstów będzie wyświetlony status przetwarzania (19). Podczas analizy będzie to „Trwa przetwarzanie". Czas przetwarzania przeciętnej wielkości książki o objętości ok. 80-100 tys. słów powinien wynieść około 4-5 minut, choć zależy to również od aktualnego obciążenia serwera oraz wybranych warstw anotacji. Obecnie maksymalny czas przetwarzania jednego pliku wynosi 10 minut – zadania dłuższe zakończą się niepowodzeniem. Podczas przetwarzania tekstów można nadal dodawać następne teksty za pomocą przycisku (11).

Gdy wszystkie teksty zostaną przetworzone, a ich status zostanie oznaczony jako „Przetworzony prawidłowo” (21), status całego korpusu zostanie również automatycznie zaktualizowany do wartości „Gotowy” (20). Na tym etapie zostaną odblokowane przyciski u dołu ekranu i będzie można przystąpić do dalszej pracy z korpusem. Możliwe dalsze czynności to:

- Przeszukiwanie korpusu — przycisk (22)
- Podsumowanie statystyczne korpusu — przycisk (23)
- Pobieranie przetworzonych plików XML — przycisk (24)
- Udostępnienie korpusu innym użytkownikom (25)
- Edycja korpusu (26)

Kliknięcie przycisku (23) spowoduje przeniesienie do ekranu podsumowania statystycznego oraz rozpoczęcie ich generowania. Opis poszczególnych części podsumowania znajduje się tutaj.

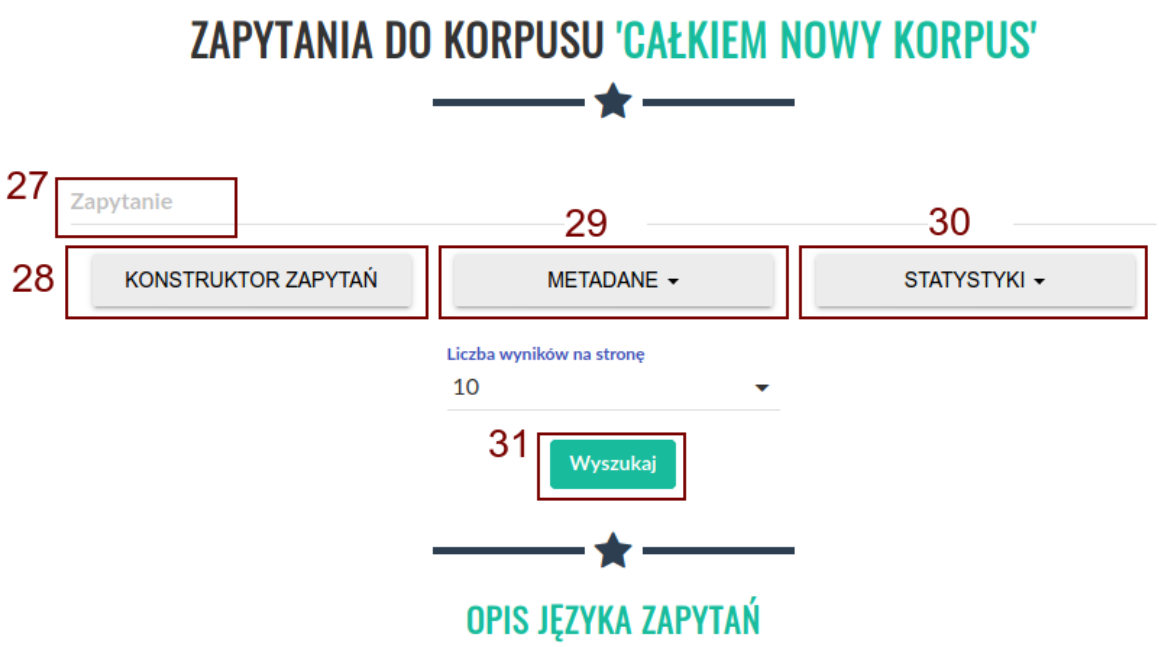
Kliknięcie przycisku (24) spowoduje pobranie archiwum z przetworzonymi plikami XML tekstów w korpusie. Pliki te są w formacie zgodnym ze specyfikacją CCL.



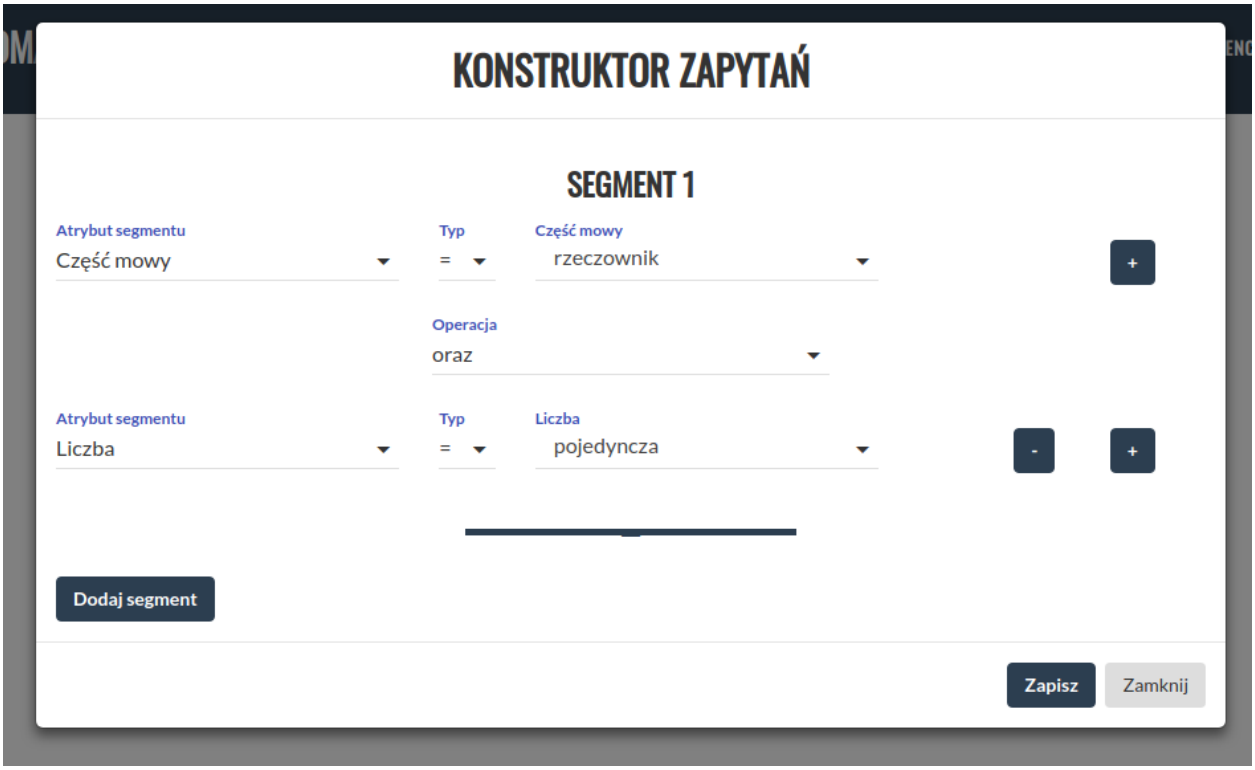
Na tym etapie nadal można edytować korpus. Dodawanie oraz usuwanie tekstów spowoduje automatyczne uruchomienie procesu przetwarzania, po zakończeniu którego korpus z powrotem otrzyma status „Gotowy”.

1.1.3 Wyszukiwanie w korpusie

Kliknięcie przycisku (22) spowoduje przeniesienie do ekranu wyszukiwania. W polu „Zapytanie” (27) należy wpisać zapytanie, które chcemy wykonać, a następnie wcisnąć przycisk „Wyszukaj” (31). Opis języka zapytań dostępny jest w kolejnej części instrukcji. Przycisk (28) uruchamia graficzny konstruktor zapytań. Przycisk (29) rozwija menu ograniczenia wyszukiwania do tekstów o konkretnych metadanych. Przycisk (30) pozwala na wygenerowanie z wyników zapytania prostej listy frekwencyjnej w oparciu o wybrane kryterium.



Kliknięcie przycisku (28) spowoduje otwarcie ekranu konstruktora zapytań. Pozwala on na zbudowanie interesującego zapytania poprzez wybranie cech segmentów z rozwijanych list. Należy jednak pamiętać, że konstruktor jest ograniczony jedynie do anotacji fleksyjnej. Po wybraniu wszystkich cech należy kliknąć przycisk „Zapisz", aby powrócić do ekranu wyszukiwania. W polu zapytanie pojawi się wtedy interesujące nas zapytanie przetworzone na język zapytań wyszukiwarki.

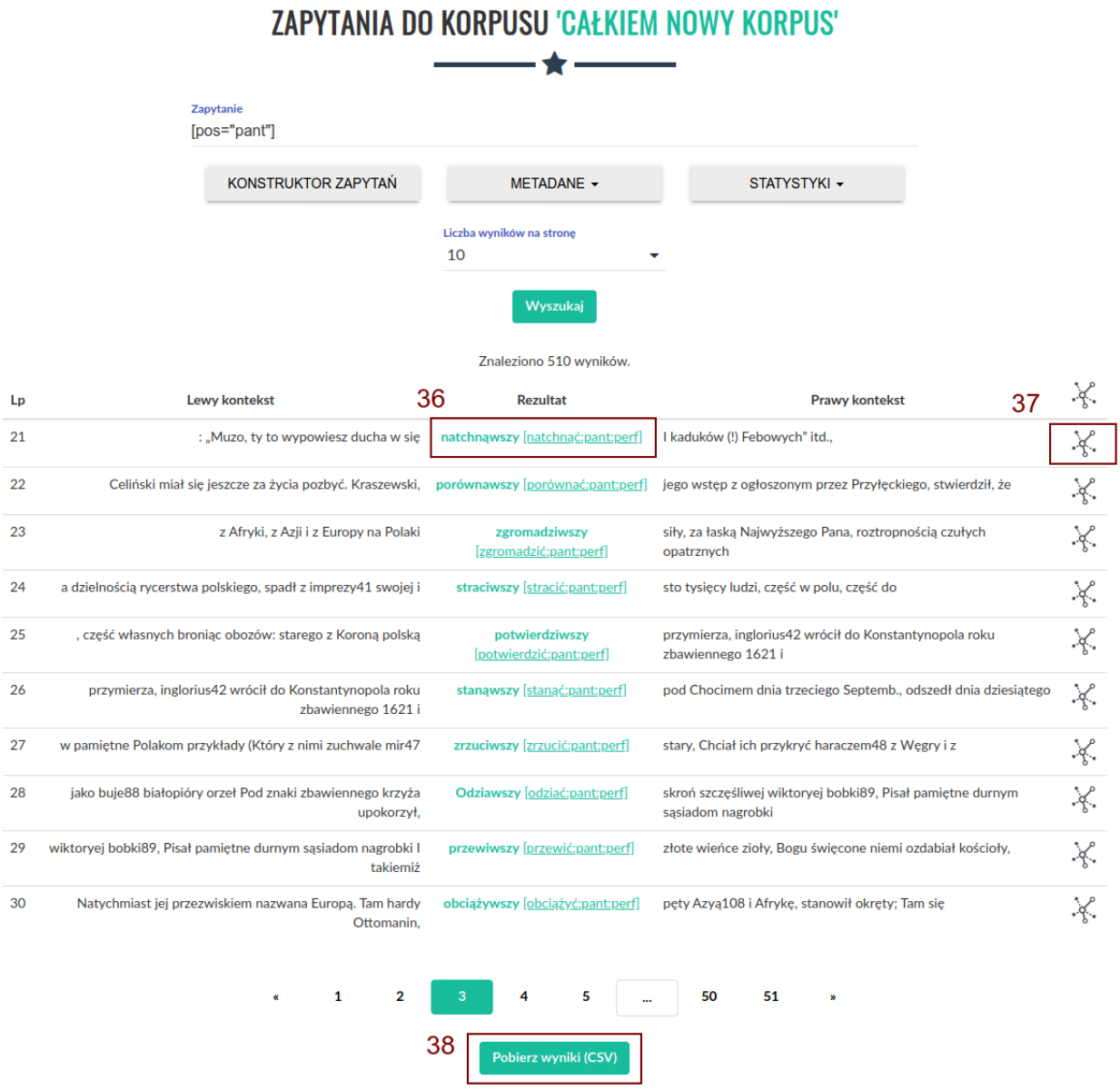


Kliknięcie przycisku (32) spowoduje rozwinięcie menu metadanych (33). Możemy tutaj ograniczyć wyniki wyszukiwania jedynie do tekstów, które spełniają wyspecyfikowane kryteria.

Kliknięcie przycisku (34) spowoduje rozwinięcie menu statystyk (35). Możemy tutaj dołączyć do wyników wyszukiwania pewne proste dane frekwencyjne. Przykładowo, możemy wyświetlić listę frekwencyjną wyników według konkretnego atrybutu segmentu lub wykres pokazujący rozkład wyników ze względu na wybrane metadane.

Po wykonaniu zapytania zostaniemy przeniesieni do strony z wynikami, które możemy przeglądać. Możemy też wyświetlić dodatkowe informacje o kontekście znalezionego wyniku, klikając na niego (36), lub pobrać całą listę wyników

w formie csv (37). Aby wyświetlić wizualizację rozbiorów składniowych wypowiedzień w korpusach, w których opcja przetwarzania zależnościowego została zaznaczona, należy kliknąć w ikonę po prawej stronie tabeli (37).



1.2 Tworzenie zapytań do korpusu

1.2.1 Wprowadzenie

Niniejszy dokument powstał w oparciu o Ściągawkę do Narodowego Korpusu Języka Polskiego, której autorem jest Adam Przepiórkowski i którą następnie poprawiali i rozszerzali Jakub Wilk i Aleksander Buczyński. *Ściągawka* stanowi instrukcję użytkowania wyszukiwarki Poliqarp z Narodowym Korpusem Języka Polskiego. Jej pełna wersja znajduje się w repozytorium wyszukiwarki Poliqarp.

Niniejszy dokument został przygotowany przez Witolda Kierasia i opisuje sposób użytkowania wyszukiwarki MTAS, niepowiązanej z Poliqarpem, ale wykorzystującej podobny język zapytań znany pod nazwą *Corpus Query Language* (CQL). Modyfikacje wprowadzone do pierwotnej wersji instrukcji uwzględniają różnice w języku zapytań oraz w tagsecie stosowanym w Korpusomacie. Za zgodą wszystkich wyżej wymienionych autorów niniejsza wersja dokumentu zostaje udostępniona na zasadach licencji Creative Commons BY-SA.

1.2.2 Segmentacja

Znaczniki morfosyntaktyczne, tzw. tagi, przypisane są segmentom (tokenom, w przybliżeniu słowom). Segmenty nie są dłuższe niż słowa ortograficzne (słowa 'od spacji do spacji'), ale w niektórych wypadkach segmenty mogą być krótsze niż takie słowa:

- Jako odrębne segmenty traktowane są formy aglutynacyjne leksemu być, a zatem następujące słowa reprezentują po dwa segmenty: *[łgał][eś]*, *[długo][śmy]*, *[tak][em]*.
- Za odrębne segmenty uznane są partykuły *by*, *-ż(e)* i *-li*, a zatem następujące słowa reprezentują po kilka segmentów: *[przyszedł][by]*, *[napisała][by][m]*, *[chodź][że]*, *[potrzebował][że][by][ś]*, *[znasz][li]*.
- Odrębnym segmentem jest poprzyimkowa nieakcentowana forma zaimka *-ń*: *[do][ń]*, *[ze][ń]*.
- Dzielone na segmenty są niektóre słowa zawierające łącznik, a mianowicie:
  - słowa typu *[polsko][-][niemiecki]*,
  - podwójne nazwiska, np. *[Kowalska][-][Nowakowska]*.

Nie są natomiast dzielone skrótowce zawierające łącznik sygnalizujący odmianę, np. *PRL-u*.

Dzielone na segmenty są także występujące na końcu zdania formy kończące się kropką, np. skróty typu *itd.*, *itp.*, liczby pisane cyframi w znaczeniu porządkowym i inicjały, np. *[itp][.]*, *[George] [W][.]* itp. Dzielenie form z kropką kończących zdanie jest uzasadnione podwójną rolą kropki w takiej pozycji: jest ona częścią formy i jednocześnie sygnalizuje koniec zdania. W wypadku, gdy takie formy nie występują na końcu zdania, są one uznawane za pojedyncze segmenty.

Poniżej znajduje się przykładowe zdanie i jego segmentacja zgodna z opisanymi w tej części zasadami.

> Pojechalibyśmy z Janem M. Rokitą i Janem Nowakiem-Jeziorańskim na sesję polsko-amerykańską, gdyby nas zaprosił George W. Byłaby to nasza już 2. doń podróż od czasów PRL-u, a może i 3., czy nawet 4.

```
[Pojechali][by][śmy] [z] [Janem] [M.] [Rokitą] [i] [Janem]
[Nowakiem][-][Jeziorańskim] [na] [sesję]
[polsko][-][amerykańską][,] [gdyby] [nas] [zaprosił] [George]
[W][.] [Była][by] [to] [nasza] [już] [2.] [do][ń] [podróż] [od]
[czasów] [PRL-u][,] [a] [może] [i] [3.][,] [czy] [nawet] [4][.]
```

1.2.3 Zestaw znaczników morfosyntaktycznych

Każdy znacznik morfosyntaktyczny jest ciągiem wartości rozdzielonych dwukropkami, np.: *subst:sg:nom:m1* dla segmentu *chłopiec*. Pierwsza wartość, np. *subst*, określa klasę gramatyczną (por. p. *2.2*), następne zaś, np. *sg*, *nom* i *m1* wartości odpowiednich dla tej klasy kategorii gramatycznych (por. p *2.1*).

Kategorie gramatyczne

Tabela *1* przedstawia repertuar kategorii gramatycznych używanych w znakowaniu tekstów w Korpusomacie. Repertuar kategorii pochodzi z tagsetu analizatora morfologicznego Morfeusz, który jest podobny do tagsetu NKJP, ale zawiera też kilka modyfikacji. Najważniejszą cechą tagsetu Morfeusza w stosunku do tagsetu NKJP jest wprowadzenie obok kategorii rodzaju opcjonalnej tzw. kategorii przyrodzaju o wartościach `pt`, `col` i `ncol` przysługują one wyłącznie klasom `subst` i `num`. W wypadku `subst` informują o łączliwości danego rzeczownika rodzaju nijakiego z liczebnikami głównymi (`ncol`, np. okno) lub zbiorowymi (`col`, np. dziecko) oraz o braku w paradygmacie rzeczownika form liczby pojedynczej (`pt`, np. skrzypce). W wypadku klasy `num` wartość kategorii przyrodzaju informuje o tym, że dana forma jest zbiorowa (`col`, np. *dwoje*) lub niezbiorowa (`ncol`, np. *dwa*). Konsekwencją wprowadzenia kategorii przyrodzaju w klasie `num` było usunięcie z tagsetu klasę `numcol`. Inne różnice to m.in.:

- ograniczenie klasy przymiotników poprzyimkowych *adjp* wyłącznie do form dopełniacza (np. *(z) wolna*) i celownika (np. *(po) polsku*) dawnej odmiany niezłożonej przymiotników oraz dodanie im wartości przypadka;
- przemianowanie klas `qub` i `burk` odpowiednio na `part` i `frag`.

Klasy gramatyczne

Zasięg tradycyjnych części mowy, takich jak czasownik, rzeczownik, liczebnik czy zaimek, jest nieostry i przez to kontrowersyjny: czy tzw. odsłowniki, tj. formy typu *picie* i *palenie*, to czasowniki (posiadają kategorię aspektu, są regularnie powiązane z formami czasownikowymi typu *pić* i *palić*), czy też rzeczowniki (odmieniają się przez przypadek, posiadają słownikową kategorię rodzaju)?, czy *piąty* to liczebnik (na to wskazuje semantyka), czy też przymiotnik (na to wskazuje odmiana)?, czy *taki* to zaimek (semantyka), czy przymiotnik (odmiana)?

W Korpusomacie klasy gramatyczne rozumiane są morfosyntaktycznie, są one oparte na pojęciu fleksemu, będącym pojęciem węższym od terminu leksem.

Tabela *2* zawiera przybliżoną charakterystykę morfosyntaktyczną wszystkich klas fleksyjnych przyjmowanych w niniejszym tagsecie. Symbol $\oplus$ oznacza, że dla danej klasy fleksyjnej dana kategoria gramatyczna jest morfologiczna (fleksemy należące to tej klasy zwykle „odmieniają się" przez tę kategorię), zaś symbol $\odot$ oznacza, że dana kategoria jest słownikowa (wszystkie formy dowolnego fleksemu należącego do tej klasy mają tę samą wartość tej kategorii, choć mogą to być różne wartości dla różnych fleksemów, jak w wypadku rodzaju rzeczowników).

Tabela *3* zawiera informacje o formach podstawowych dla poszczególnych klas fleksyjnych, a także skróty nazw klas fleksyjnych używane w korpusie.

1.2.4 Język zapytań

Składnia zapytań w programie MTAS została oparta na języku zapytań o nazwie Corpus Query Language (CQL), wykorzystywanym w wielu innych tego typu programach, m.in. w programie Sketch Engine, ale też w znanym z NKJP Poliqarpie (wykorzystywanym w starej wersji Korpusomatu). Należy jednak zwrócić uwagę na drobne różnice, ponieważ mogą one wpływać na poprawność formułowanych zapytań. Niniejszy rozdział omawia składnię zapytań wyszukiwarki MTAS i ilustruje ją wieloma przykładami.

MTAS jest uniwersalną wyszukiwarką pozwalającą na przeszukiwanie korpusów zawierających wiele warstw anotacyjnych, np. warstwę morfosyntaktyczną, warstwę składniową, warstwę nazw własnych, warstwę sensu słów itp. Niniejsza instrukcja dotyczy przeszukiwania korpusów tekstów polskich w postaci indeksowanej przez Korpusomat, który tworzy aktualnie trzy warstwy znakowania: warstwę morfosyntaktyczną i składniową oraz warstwę jednostek nazewniczych. Z tego powodu instrukcja języka zapytań ogranicza się tylko do tych warstw i nie uwzględnia możliwości wyszukiwarki zastosowanej do innych korpusów. Nie należy jej zatem traktować jako ogólnej instrukcji użytkowania wyszukiwarki MTAS. Podstawowa dokumentacja wyszukiwarki 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=".z.z..?"]`

   segmenty składające się z pięciu lub sześciu znaków, w których 2. i 4. znak to *z*, np. *czczą*, *rzezi* i *szczyt*,

8. `[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.,

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

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

10. `[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*,

11. `[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*,

12. `[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*, [lala]

13. `[orth="[bcćdfghjklłmnńprsśtwzźż]{4,}[aąeęioóuy]"]`

    segmenty składające się z co najmniej 4 liter spółgłoskowych i dokładnie jednej litery samogłoskowej, np. *źdźbła*, *drzwi* i *czczą*; wyrażenie `[bcćdfghjklłmnńprsśtwzźż]{4,}` oznacza co najmniej czterokrotne powtórzenie znaku pasującego do `[bcćdfghjklłmnńprsśtwzźż]`, tj. co najmniej cztery wystąpienia litery spółgłoskowej (niekoniecznie tej samej),

14. `[orth="([bcćdfghjklłmnńprsśtwzźż]{3}[aąeęioóuy]){2,}"]`

    segmenty składające się z co najmniej dwukrotnego powtórzenia wzorca CCCV, gdzie C to litera spółgłoskowa, a V to litera samogłoskowa, np. *wszystko*, *przykrzejszy* i *szlachta*; konstrukcja typu `n` oznacza dokładnie `n` wystąpień znaku lub ujętego w nawiasy okrągłe wyrażenia bezpośrednio przed nią,

15. `[orth="(pod|na|za)jecha.*"]`

    segmenty zaczynające się od *podjecha*, *najecha* i *zajecha*, np. *podjechał*, *zajechawszy*.

Specyfikacje segmentów podane powyżej muszą pasować do całych segmentów stąd konieczność umieszczenia po obu stronach ciągu `(la){3,}` w zapytaniu `[orth=".*(la){3,}.*"]` wyrażenia `.*`, pasującego do dowolnego ciągu znaków.

Zapytania z innymi atrybutami

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

```
[base="korpus"]
```

Atrybut `base` jest jednym z wielu możliwych atrybutów, jakie mogą pojawić się w zapytaniu. Wartością tego atrybutu powinna być specyfikacja formy podstawowej (hasłowej), a zatem zapytanie `[base="pisać"]` może być użyte do znalezienia form typu *pisać*, *piszę*, *pisała*, *piszcie*, *pisanie*, *pisano*, *pisane* itp.

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

```
[base="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" & base="mina"]
```

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

```
[orth="minę" & !base="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:

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

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

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

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

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

  segment *pora* nie będący ani formą leksemu por, ani formą leksemu pora.

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

```
[orth="minę" & base="mina"]
[orth="minę" | base="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"][][][base="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}[base="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}[base="bać"]
| [base="bać"][base="on|ja|ty|my|wy"]?[orth="się"]
```

Zapytania o znaczniki morfosyntaktyczne

Powyższe zapytanie można uprościć poprzez zastąpienie warunku `orth!="[.!?,:]"` bezpośrednim odwołaniem do „klasy gramatycznej” `interp`:

```
[orth="się"][!pos="interp"]{0,5}[base="bać"]
| [base="bać"][base="on|ja|ty|my|wy"]?[orth="się"]
```

Ogólniej, wartościami atrybutu `pos` (ang. *part of speech* ‘część mowy’) są skróty nazw klas gramatycznych omówionych w p. *2.2* (por. tabela *2*). Na przykład zapytanie o sekwencję dwóch form rzeczownikowych rozpoczynających się na *a* może być sformułowane w sposób następujący:

```
[pos="subst" & 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. Na przykład, zważywszy na to, że zaimki osobowe należą do klasy zaimków trzecioosobowych *ppron3* i do klasy zaimków nietrzecioosobowych *ppron12*, poniższe zapytania mogą posłużyć do znalezienia dowolnych form dowolnych zaimków osobowych:

```
[pos="ppron12" | pos="ppron3"]
[pos="ppron12|ppron3"]
[pos="ppron(12|3)"]
[pos="ppron[123]+"]
[pos="ppron.+"]
```

A zatem zapytanie o formy *bać się* może zostać jeszcze bardziej uproszczone do następującego zapytania:

```
[orth="się"][!pos="interp"]{0,5}[base="bać"]
| [base="bać"][pos="ppron.+"]?[orth="się"]
```

W zapytaniach można określić nie tylko postać ortograficzną segmentu (za pomocą atrybutu `orth`), formę hasłową (za pomocą `base`) i klasę gramatyczną (za pomocą `pos`), ale także wartości poszczególnych kategorii gramatycznych, np. przypadka czy rodzaju. Służą do tego następujące atrybuty (por. p.*2.1*):

atrybut	kategoria	możliwe wartości
`number`	liczba	`sg pl`
`case`	przypadek	`nom gen dat acc inst loc voc`
`gender`	rodzaj	`m1 m2 m3 f n`
`subgender`	przyrodzaj	`col ncol pt`
`person`	osoba	`pri sec ter`
`degree`	stopień	`pos comp sup`
`aspect`	aspekt	`imperf perf`
`negation`	zanegowanie	`aff neg`
`accentability`	akcentowość	`akc nakc`
`post-prepositionality`	poprzyimkowość	`npraep praep`
`agglutination`	aglutynacyjność	`agl nagl`
`vocalicity`	wokaliczność	`nwok wok`
`fullstoppedness`	kropkowalność	`pun npun`

A zatem możliwe jest zadanie na przykład następujących zapytań:

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

   znalezione zostaną wszystkie formy w liczbie pojedynczej,

2. ```
   [pos="subst" & number="sg"]
   ```

   znalezione zostaną formy rzeczownikowe w liczbie pojedynczej,

3. ```
   [pos="subst" & !gender="f"]
   ```

   formy rzeczownikowe rodzaju męskiego lub nijakiego,

4. ```
   [number="sg" & case="nom|acc" & gender="m[123]"]
   ```

   pojedyncze mianownikowe lub biernikowe formy męskie.

O klasy gramatyczne i kategorie gramatyczne można także pytać łącznie, używając do tego atrybutu `tag`. Na przykład, aby znaleźć wszystkie rzeczowniki żeńskie w mianowniku o pojedynczej wartości liczby, można zadać następujące zapytanie:

```
[tag="subst:sg:nom:f"]
```

Wartości atrybutu `tag` mają postać `kl:kat1:kat2:...:katn`, gdzie `kl` to nazwa klasy gramatycznej, a `kati` to wartości kategorii przysługujących tej klasie w kolejności, w jakiej zostały podane w tabeli *2*.

Jak w wypadku innych atrybutów, specyfikacja atrybutu `tag` może być zadana wyrażeniem regularnym, np.:

```
[tag=".*:sg:(nom|acc):m[123].*"]
```

Ponieważ nazwy wartości poszczególnych kategorii są rozłączne, można również stosować zbiorczą kategorię `feat` (ang. *feature* 'cecha') w zastępstwie każdej innej. Ujednoznacznienie dokona się przez odpowiednią wartość. Dlatego następujące dwa zapytania zwrócą te same wyniki:

- `[pos="subst" & case="acc" & number="pl" & gender="f"]`
- `[pos="subst" & feat="acc" & feat="pl" & feat="f"]`

Interpretacje spoza słownika

Interpretacje fleksyjne w znakowaniu morfosyntaktycznym Korpusomatu pochodzą z analizatora Morfeusz 2 i tagera Concraft 2 — analizator zwraca wszystkie możliwe interpretacje dla danego słowa, a tager wybiera najbardziej prawdopodobną ze względu na swój model statystyczny. Interpretacje Morfeusza pochodzą ze Słownika gramatycznego języka polskiego (SGJP). Jeśli danego słowa nie da się w żaden sposób zinterpretować jako formy wyrazowej leksemu zanotowanego w SGJP, to Morfeusz nie zwraca żadnej interpretacji. Wówczas tager „zgaduje" znacznik morfosyntaktyczny, czyli wybiera taki, który zgodnie z jego modelem jest najbardziej prawdopodobny. Skuteczność zgadywania jest w oczywisty sposób dużo niższa niż skuteczność wybierania spośród gotowych interpretacji z Morfeusza, dlatego użytkownik może uznać za przydatną możliwość sterowania tym parametrem w swoich wyszukaniach, np. w wypadku słownictwa najnowszego, nienotowanego w słownikach. Segmenty, którym Morfeusz nie przypisał żadnej interpretacji, mają dodatkowy parametr postaci `[ign="true"]`. Poniższe przykładowe zapytanie odnajdzie w korpusie wszystkie słowa, które zaczynają się od „tofu" i nie są znane Morfeuszowi:

```
[orth="tofu.*" & ign="true"]
```

Analogicznie można usunąć z wyszukiwania interpretacje zgadywane, np.:

```
[pos="subst" & !ign="true"]
```

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, postać segmentu w obu warstwach tekstowych, formę hasłową, a także wartości wszystkich kategorii gramatycznych opisanych w tabeli *1*. 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:

```
[base="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 być w czasie przyszłym złożonym ograniczone do zdań zawierających formę bezokolicznika:

```
[pos="bedzie"] within (<s/> containing [pos="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 formy pseudoimiesłowu, a bezokolicznik pełni w zdaniu inną funkcję gramatyczną. Można też sformułować zapytanie odwrotnie i zapytać o zdania, w których forma pseudoimiesłowu w ogóle nie występuje:

```
[pos="bedzie"] within (<s/> !containing [pos="praet"])
```

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> [pos="num"]
```

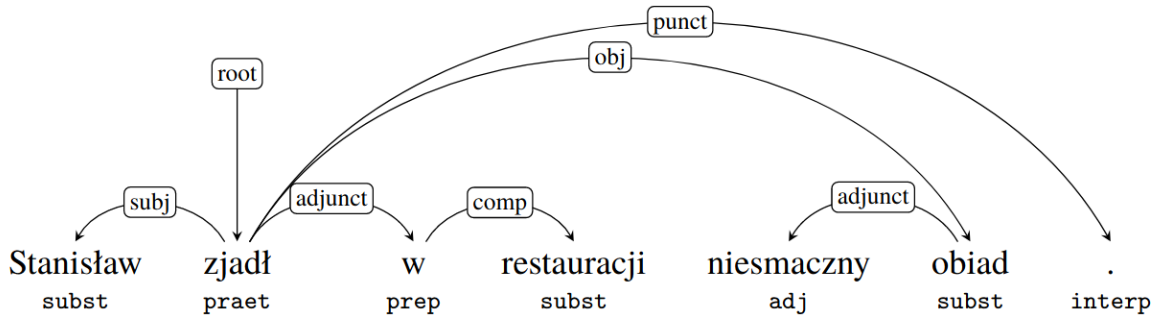
odnajdzie wszystkie wystąpienia liczebnika stojącego na początku zdania. Analogicznie zapytanie:

```
[pos="num"][pos="interp"]</s>
```

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

Warstwa składniowa

W Korpusomacie jest również wbudowany parser zależnościowy Combo. Wprowadzony przez użytkownika tekst jest automatycznie dzielony na wypowiedzenia, które z kolei są poddawane pełnej analizie składniowej w aparacie zależnościowym według zasad przyjętych w Polskim Banku Drzew Zależnościowych. Przykład takiej analizy znajduje się na poniższym rysunku.



Rysunek 1. Rozbiór składniowy przykładowego zdania.

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 jeden z 28 typów zależności przewidzianych w Polskim Banku Drzew Zależnościowych,
- `head.pos` — klasa fleksyjna bezpośredniego nadrzędnika segmentu (tabela *2*),
- `head.base` — forma hasłowa 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 użyte w funkcji dopełnienia bliższego konkretnego czasownika:

```
[pos="subst" & deprel="obj" & head.base="kupić"]
```

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

```
[deprel="obj.*" & head.pos="(fin|praet|ppas|pact|ger|impt|imps)" & base="betel"]
```

Należy jednak zwrócić uwagę, że w powyższym przykładzie wynikiem zapytania będą wystąpienia rzeczownika betel, 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. Wartością atrybutu `deprel` jest wyrażenie regularne, do którego dopasowują się dwa możliwe typy relacji zależności: *obj* i *obj_th* opisane w dokumentacji Polskiego Banku Drzew Zależnościowych.

Podobne wyszukanie możliwe jest również w wypadku wymagań czasownika innych niż nominalne. Na przykład za pomocą zapytania:

```
[deprel="comp" & head.pos="(fin|praet|imps|impt|ppas|pact)" & base="o" & case="loc"]
```

można znaleźć czasowniki wymagające frazy przyimkowej miejscownikowej z przyimkiem o.

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

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

lub dopełnienia bliższego przed orzeczeniem:

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

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

```
[pos="adj" & deprel="adjunct" & head.base="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.pos="praet" & !head.distance="1"]
```

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

Jeszcze jednym praktycznym przykładem wykorzystania anotacji składniowej jest możliwość wyszukania analitycznych form fleksyjnych, których poszczególne fleksemy nie są oznaczane w warstwie morfosyntaktycznej jako elementy takiej formy. Dotyczy to np. form czasu przyszłego niedokonanego (utworzonych z formami bezokolicznika lub pseudoimiesłowu lub w obu wariantach):

```
[pos="bedzie" & deprel="aux" & head.pos="(inf|praet)"]
```

czy analitycznych form stopnia wyższego i najwyższego przymiotników:

```
[deprel="adjunct" & base="bardzo" & degree="(com|sup)" & head.pos="adj"]
```

Podobnie w wypadku konstrukcji biernej:

```
[base="(być|zostać)" & deprel="aux" & head.pos="ppas"]
```

Warstwa jednostek nazewniczych

Korpusy indeksowane przez Korpusomat zawierają również warstwę znakowania jednostek nazewniczych (ang. *named entities*). Są to jednostki tekstowe jedno- lub wielowyrazowe nazywające osoby, miejsca, instytucje czy momenty czasowe. Automatycznym klasyfikowaniem takich jednostek tekstowych zajmuje się wbudowany w Korpusomat program Liner2, który określa początek i koniec danej jednostki nazewniczej oraz przydziela jej odpowiednią etykietę. Liner2 opiera się na wzorcowej anotacji jednostek nazewniczych przygotowanej w ramach projektu NKJP, której szczegóły zostały opisane w rozdziale *Anotacja jednostek nazewniczych* (str. 129-167) książki Narodowy Korpus Języka Polskiego. Niniejsza instrukcja ogranicza się jedynie do opisania sposobu korzystania z tej klasyfikacji w wyszukiwarce Korpusomatu.

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 osób:

```
<ne="persName" />
```

Ta kategoria jednostek ma swoją dodatkową podkategorię klasyfikującą człony nazwy osobyc: imię, nazwisko, itp. Następujące zapytanie ograniczy wyniki jedynie do nazwisk:

```
<ne="persName.surname" />
```

Pełny repertuar wartości klasyfikacji jednostek nazewniczych to:

- `persName` (nazwy osób) z podtypami: `forename` (imiona), `surname` (nazwiska) i `addName` (pseudonimy, przydomki itp.),

- `orgName` (nazwy organizacji),
- `geogName` (nazwy geograficzne),
- `placeName` (nazwy miejsc czy też tzw. nazwy geopolityczne) z podtypami: `district` (jednostki administracyjne miast, np. *Mokotów*), `settlement` (miasta, wioski, osady, np. *Warszawa*), `region` (jednostki administracyjne większe niż miasto, np. *województwo mazowieckie*), `country` (państwa, kraje, wspólnoty, kolonie, np. *Polska*, *Gujana Francuska*), `bloc` (organizacje polityczne obejmujące co najmniej dwa państwa, np. *Unia Europejska*, *Grupa Wyszehradzka*); **uwaga**: w przypadku typu `placeName` zapytanie ogólne nie zwraca wyników anotowanych podtypami szczegółowymi,
- wyrażenia czasowe: `date` (daty kalendarzowe, np. 13 sierpnia 2018 r.) oraz `time` (określenia czasu w postaci godzin, minut i sekund, np. *ósma wieczorem*).

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:

```
[pos="conj" & base="i"] within <ne="orgName" />
```

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

```
<ne="persName" /> !containing <ne="persName.forename" />
```

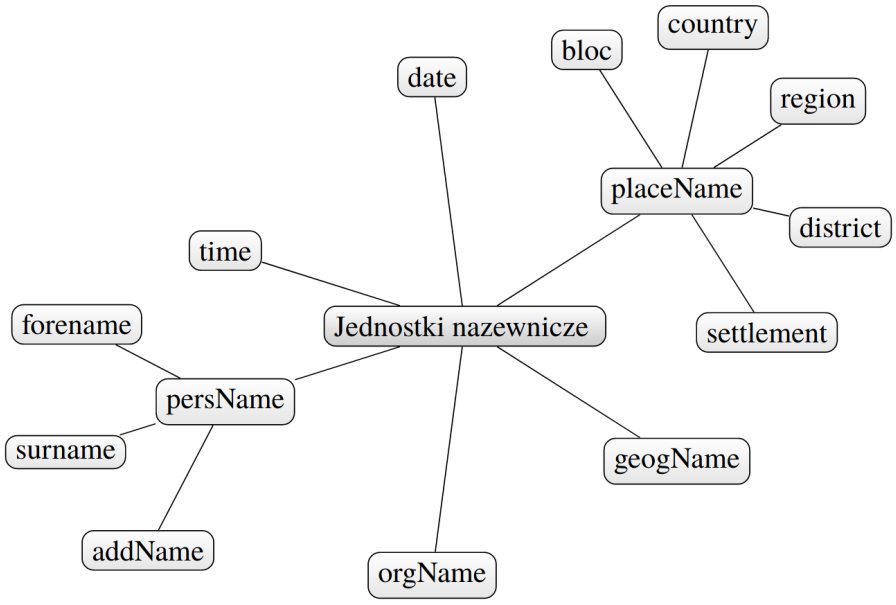
— wszystkie jednostki nazywające osoby, których składową nie jest imię,

```
<ne="geogName" /> [pos="conj"] <ne="geogName" />
```

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

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

— 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*.



Rysunek 2. Hierarchia typów jednostek nazewniczych w NKJP

Warstwa znakowania wydźwięku emocjonalnego

Znakowanie tekstów w Korpusomacie można również wzbogacić o oznaczenie wydźwięku emocjonalnego słów. Jest to znakowanie wyłącznie słownikowe, opierające się na zbiorze 2902 polskich rzeczowników, przymiotników i czasowników zebranych w bazie NAWL (*Nencki Affective Word List*) stworzonej w ramach projektu prowadzonego w Instytucie Biologii Doświadczalnej im. M. Nenckiego PAN. W oparciu o badania ankietowe w słowniku sklasyfikowano słowa ze względu na kojarzące się z nimi podstawowe emocje: szczęście (*happiness*), złość (*anger*), smutek (*sadness*), strach (*fear*), wstręt (*disgust*) oraz słowa neutralne emocjonalnie (*neutral*) oraz takie, dla których wskazania były niejednoznaczne i nie umożliwiały zaklasyfikowania (*unclassified*). Poszczególnym klasom odpowiadają etykiety będące pierwszymi literami ich angielskich odpowiedników, czyli H, A, S, F, D, N, U. Etykiety są wartościami atrybutu `sentiment.nawl`, którego można użyć w zapytaniach korpusowych. Przykładowo, zapytanie postaci:

```
[sentiment.nawl="A"]
```

odnajdzie wszystkie wystąpienia słów oznaczonych w słowniku NAWL jako kojarzące się ze złością. Tego typu zapytania można łączyć z warunkami dotyczącymi innych warstw znakowania (o ile zostały one wybrane przez użytkownika w trakcie tworzenia korpusu), na przykład można ograniczyć wyniki do określonych części mowy:

```
[sentiment.nawl="A" & pos="adj"]
```

czy do postaci hasłowej składniowego nadrzędnika w strukturze zależnościowej wypowiedzenia:

```
[sentiment.nawl="H" & head.base="Polak"]
```

Oczywiście należy pamiętać, że słownik NAWL jest stosunkowo niewielki, zatem zdecydowana większość słów w korpusie nie będzie miała przypisanych żadnych wartości wydźwięku emocjonalnego.

W oryginalnej bazie danych słownika NAWL każde słowo zostało przypisane tylko do jednej kategorii. W zaimplementowanej w Korpusomacie wersji rozszerzonej tego słownika słowo może mieć przypisaną więcej niż jedną etykietę kategorii emocji, jeśli te emocje uzyskały w bazie odpowiednio wysoki wskaźnik. Na przykład rzeczownik wojna

w słowniku rozszerzonym ma przypisane dwie etykiety: strach (F) i smutek (S). Zapytanie o każdą z tych emocji zwróci wystąpienia rzeczownika wojna w korpusie (o ile oczywiście to słowo się w nim znajduje). Jednak w oryginalnym słowniku ten sam rzeczownik jest przypisany do kategorii U, czyli słów niesklasyfikowanych ze względu na niejednoznaczne wskazania ankietowe. Obie wersje tego słownika są dostępne w Korpusomacie. Wyniki dla wersji rozszerzonej dostępne są pod atrybutem `sentiment.nawl`, dla oryginalnej wersji zaś — pod atrybutem `sentiment.nawl_org`. W wypadku korzystania wersji oryginalnej należy pamiętać, że w wynikach znacznie więcej słów będzie przypisanych do kategorii U.

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.

1.2.5 Tabele

Kategorie gramatyczne

Tabela 1: Kategorie gramatyczne

Liczba: (2 wartości)		
pojedyncza	*sg*	*oko*
mnoga	*pl*	*oczy*
Przypadek: (7 wartości)		
mianownik	*nom*	*woda*
dopełniacz	*gen*	*wody*
celownik	*dat*	*wodzie*
biernik	*acc*	*wodę*
narzędnik	*inst*	*wodą*
miejscownik	*loc*	*wodzie*
wołacz	*voc*	*wodo*
Rodzaj: (5 wartości)		
męski osobowy	*m1*	*papież*, *kto*, *wujostwo*
męski zwierzęcy	*m2*	*baranek*, *walc*, *babsztyl*
męski rzeczowy	*m3*	*stół*
żeński	*f*	*stuła*
nijaki	*n*	*dziecko*, *okno*, *co*, *skrzypce*, *spodnie*
Przyrodzaj: (3 wartości)		
przymnogi	*pt*	*wujostwo*, *skrzypce*, *spodnie*
zbiorowy	*col*	*dziecko*
niezbiorowy	*ncol*	*okno*
Osoba: (3 wartości)		
pierwsza	*pri*	*bredzę*
druga	*sec*	*bredzisz*
trzecia	*ter*	*bredzi*
Stopień: (3 wartości)		
równy	*pos*	*cudny*

ciąg dalszy na następnej stronie

Tabela 1 – kontynuacja poprzedniej strony

Liczba: (2 wartości)		
wyższy	*com*	*cudniejszy*
najwyższy	*sup*	*najcudniejszy*
Aspekt: (2 wartości)		
niedokonany	*imperf*	*iść*
dokonany	*perf*	*zajść*
Zanegowanie: (2 wartości)		
niezanegowana	*aff*	*pisanie*, *czytanego*
zanegowana	*neg*	*niepisanie*, *nieczytanego*
Akcentowość: (3 wartości)		
akcentowana	*akc*	*niego*, *jego*, *tobie*
nieakcentowana	*nakc*	*go*, *-ń*, *ci*
zneutralizowana	*neut*	*one*, *im*, *je*
Poprzyimkowość: (2 wartości) *		
poprzyimkowa	*praep*	*niego*, *-ń*
niepoprzyimkowa	*npraep*	*jego*, *go*
Akomodacyjność: (2 wartości)		
uzgadniająca	*congr*	*dwaj*, *pięcioma*
rządząca	*rec*	*dwóch*, *dwu*, *pięciorgiem*
Aglutynacyjność: (2 wartości)		
nieaglutynacyjna	*nagl*	*niósł*
aglutynacyjna	*agl*	*niosł-*
Wokaliczność: (2 wartości)		
wokaliczna	*wok*	*-em*
niewokaliczna	*nwok*	*-m*
Kropkowalność: (2 wartości)		
z następującą kropką	*pun*	*tzn*
bez następującej kropki	*npun*	*wg*

Klasy gramatyczne

Tabela 2: Klasy gramatyczne

	liczba	przypadek	rodzaj	przyrodz.	osoba	stopień	aspekt	zaneg.	akcent.	pop
rzeczownik										
rzeczownik deprecjatywny										
liczebnik główny										
liczebnik złoż.										
przymiotnik										
przymiotnik przyprzym.										
przymiotnik poprzyim.										
przysłówek										
zaimek nietrzecioosobowy										
zaimek trzecioosobowy										
zaimek siebie										
forma nieprzeszła										
forma przyszła być										
aglutynant być										
pseudoimiesłów										
rozkaźnik										

Tabela 2 – kontynuacja poprzedniej strony

	liczba	przypadek	rodzaj	przyrodz.	osoba	stopień	aspekt	zaneg.	akcent.	pop
bezosobnik										
bezokolicznik										
im. przys. współczesny										
im. przys. uprzedni										
odsłownik										
im. przym. czynny										
im. przym. bierny										
winien										
predykatyw										
przyimek										
spójnik współrz.										
spójnik podrz.										
partykuła										
skrót										
człon wyrażenia										
wykrzyknik										
znak interpunkcyjny										
ciało obce										

Skróty nazw klas gramatycznych oraz ich formy hasłowe

Tabela 3: Skróty nazw klas gramatycznych oraz ich formy hasłowe.

fleksem	**skrót**	**forma podstawowa**	**przykład**
rzeczownik	*subst*	mianownik l. poj.	*doktor*
rzeczownik deprecjatywny	*depr*	mianownik l. poj. rzeczownika	*doktor*
liczebnik główny	*num*	mianownik rodz. m3	*pięć*, *dwa*
liczebnik złoż.	*numcomp*	mianownik rodz. m3	*pięć*, *dwa*
przymiotnik	*adj*	mianownik l. poj. rodzaju męskiego st. równego	*polski*
przymiotnik przyprzyim.	*adja*	mianownik l. poj. rodz. męskiego przymiotnika w st. równym	*polski*
przymiotnik poprzyim.	*adjp*	mianownik l. poj. rodz. męskiego przymiotnika w st. równym	*polski*
przysłówek	*adv*	forma stopnia równego	*dobrze*, *bardzo*
zaimek nietrzecioosobowy	*ppron12*	mianownik l. poj.	*ja*
zaimek trzecioosobowy	*ppron3*	mianownik l. poj.	*on*
zaimek siebie	*siebie*	biernik	*siebie*
forma nieprzeszła	*fin*	bezokolicznik	*czytać*
forma przyszła być	*bedzie*	bezokolicznik	*być*
aglutynant być	*aglt*	bezokolicznik	*być*
pseudoimiesłów	*praet*	bezokolicznik	*czytać*
rozkaźnik	*impt*	bezokolicznik	*czytać*
bezosobnik	*imps*	bezokolicznik	*czytać*
bezokolicznik	*inf*	bezokolicznik	*czytać*
im. przys. współczesny	*pcon*	bezokolicznik	*czytać*
im. przys. uprzedni	*pant*	bezokolicznik	*czytać*
odsłownik	*ger*	bezokolicznik	*czytać*
im. przym. czynny	*pact*	bezokolicznik	*czytać*
im. przym. bierny	*ppas*	bezokolicznik	*czytać*
winien	*winien*	forma męska l. poj.	*winien*, *rad*
predykatyw	*pred*	jedyna forma fleksemu	*warto*

ciąg dalszy na następnej stronie

Tabela 3 – kontynuacja poprzedniej strony

fleksem	skrót	forma podstawowa	przykład
przyimek	*prep*	niewokaliczna forma fleksemu	*na*, *przez*, *w*
spójnik współrz.	*conj*	jedyna forma fleksemu	*oraz*
spójnik podrz.	*comp*	jedyna forma fleksemu	*że*
partykuła	*part*	jedyna forma fleksemu	*nie*, *-li*, *się*
skrót	*brev*	forma hasłowa rozwinięcia skrótu	*rok*, *i_tak_dalej*
człon wyrażenia	*frag*	jedyna forma fleksemu	*wskroś*, *dala*
wykrzyknik	*interj*	jedyna forma fleksemu	*laboga*, *pst*
znak interpunkcyjny	*interp*	jedyna forma fleksemu	*;*, *!*, *?*
ciało obce	*xxx*	jedyna forma fleksemu	*wsio*, *revolutionibus*

1.3 Profile słów

1.3.1 Wprowadzenie

Profile słów polegają na odnalezieniu w tekście słownictwa, które często łączy się ze wskazanym słowem w związki syntaktyczne 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 związku koordynacji z rzeczownikiem *kot*. Otrzymane kolokacje charakteryzują język korpusu, tj. w korpusie reprezentatywnym dla standardowego języka polskiego, będą się głównie pojawiać 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.

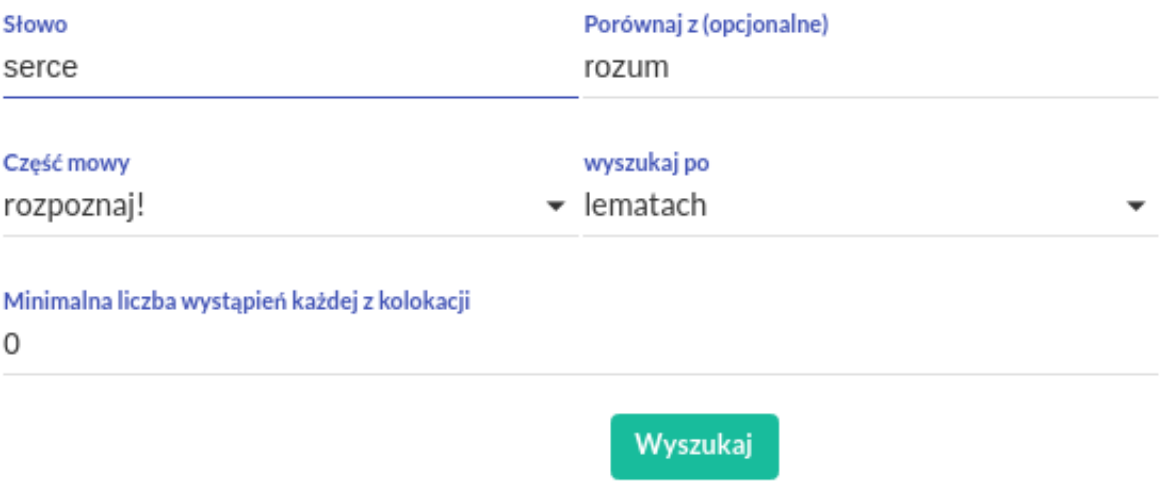
Informacja: Profile słów są dostępne wyłącznie dla korpusów posiadających warstwę anotacji zależnościowej.

Należy zauważyć, że ze względów statystycznych funkcjonalność profili słów 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 obliczyć. Tworząc profil danego słowa, możemy wybrać, czy interesują nas wszystkie jego wystąpienia, niezależnie od formy w tekście (i.e. szukamy po lemacie), 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 nie powtarzają się, a uzyskały wysoki wynik ze względu na ich rzadkość 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, więc uporządkowanie danych w wiersze w tabeli, nie ma znaczenia.

Aplikacja umożliwia także tworzenie profili porównawczych. W tym celu należy wpisać do pola *porównaj z* drugie z interesujących nas słów. 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
SŁOWA TE POJAWIAJĄ SIĘ W KORPUSIE ODPOWIEDNIO 99 I 421 RAZY

Szukaj:

	słowa których podmiotem jest "serce" vs. "rozum"	słowa których dopełnieniem bliższym jest "serce" vs. "rozum"	słowa których dopełnieniem dalszym jest "serce" vs. "rozum"	przyimki "serce" vs. "rozum"	apozycje "serce" vs. "rozum"	przymiotniki/imiesłowy przymiotnikowe modyfikujące "serce" vs. "rozum"	modyfikatory rzeczownikowe "serce" vs. "rozum"	słowa które modyfikuje "serce" vs. "rozum"	słowa z którymi "serce" vs. "rozum" występuje w koordynacji	podmioty, dla których "serce" vs. "rozum" jest orzeczeniem imiennym	orzeczenia imienne, dla których "serce" vs. "rozum" jest podmiotem
1	wezbrać VERB 8.356	znaleźć VERB 8.371	brak OTHER 9.155	w OTHER 2.635		lekki ADJ 8.945	jeleń NOUN 8.342	zachcianka NOUN 8.313	zapał NOUN 9.155		koń NOUN 8.476
2	uciszać VERB 8.356	ucieszyć VERB 8.356	wskrzeszony PPAS 8.356	mimo OTHER 1.278		wierzący ADJ 8.356	uczeń NOUN 7.724	odruch NOUN 8.206	wiara NOUN 8.625		
3	rozradować VERB 8.356	zdziczenie NOUN 8.356	lgnąć VERB 8.342			gołębi ADJ 8.356	Agaton NOUN 7.356	rys NOUN 7.608	nerki NOUN 8.356		
4	mówić VERB 0.255	być OTHER -3.987	mieć VERB -1.96	u OTHER 0.566	* OTHER -5.22	taki ADJ 0.199	Sokrates NOUN 1.128	sprawa NOUN 0.763	żądza NOUN 2.201	i OTHER -0.83	, OTHER -0.649
5	być VERB -0.727	powiadać VERB -4.034	człowiek NOUN -3.485	przez OTHER 0.563	rozsądek NOUN -6.096	ten ADJ -2.259	człowiek NOUN -0.832	; OTHER -3.638	nie OTHER 1.662	on OTHER -2.684	on OTHER -1.684
6	mieć VERB -3.019	mówić VERB -4.989	trzeba OTHER -4.978	do OTHER -0.733	núus NOUN -6.275	który ADJ -3.89	istota NOUN -5.409	coś NOUN -4.241	pożądliwość NOUN -0.17	jeden ADJ -4.098	co NOUN -2.93
7	tracić VERB -7.151	nabywanie NOUN -7.265	nazywać VERB -7.673	jak OTHER -6.776		stroniący PACT -6.279		wszechwładztwo NOUN -6.279	umysł NOUN -8.871	panowanie NOUN -6.193	czcigodny ADJ -6.232
8	decydować VERB -7.215	wołać VERB -7.823	poznawać VERB -7.715	niż OTHER -7.252		własny ADJ -6.351	núus NOUN -6.275	praca NOUN -6.897	rozsądek NOUN -8.903	oczyszczenie NOUN -6.248	iskra NOUN -6.272
9	panować VERB -7.666	słuchać VERB -8.39	poddany PPAS -8.252	bez OTHER -7.797	fronesis NOUN -6.279	prawdziwy ADJ -6.5	rozdział NOUN -6.762	skutek NOUN -7.078	rozkosz NOUN -9.023	dzielność NOUN -6.486	przyczyna NOUN -6.75

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.

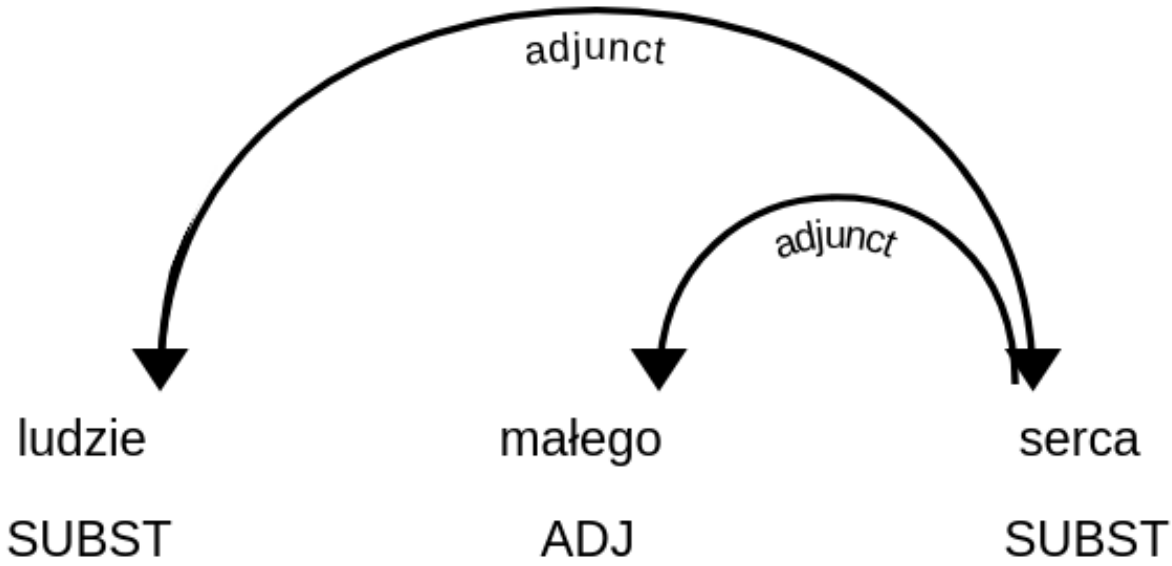
1.3.3 Wykorzystane miary

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 słowa zadanego 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 nie jest też zależna od wielkości korpusu (można więc porównywać wartości otrzymane dla różnych korpusów).

1.3.4 Podstawa lingwistyczna

Profile słów są obliczane na podstawie anotacji morfologicznej, i wyników parsowania zależnościowego, jest więc funkcją dostępną wyłącznie dla korpusów posiadających warstwę anotacji zależnościowej. Dla każdej z obsługiwanych części mowy (rzeczowniki, czasowniki, przymiotniki, przysłówki, imiesłowy przymiotnikowe czynne oraz bierne i przysłówkowe uprzednie oraz współczesne) 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).



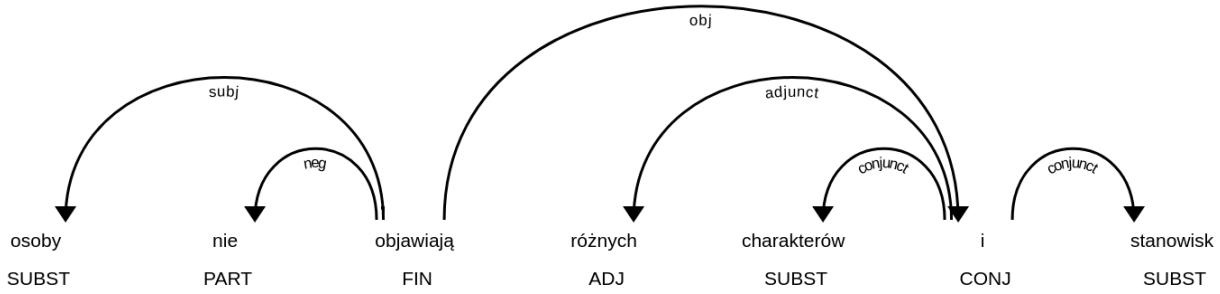
W obliczaniu profili słów, wykorzystywane są drzewa zależnościowe, takie jak fragment przedstawiony na obrazku. Jeżeli interesuje nas słowo *serce*, do kolumny *przymiotniki modyfikujące „serce”* trafi słowo *mały*, zaś do kolumny *rzeczowniki modyfikowane przez „serce”*, słowo *człowiek*.

W wykazie kolokacji, wystąpienia pojawiają się w formie zlematyzowanej, jednak stosujemy tu inne zasady lematyzacji, niż te, których wyniki są zapisane w plikach korpusu (i dostępne dla odpytywania korpusu), w szczególności imiesłowy oraz gerundia nie są lematyzowane do form czasownikowych.

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 głową jest spójnik, zaś liścmi człony koordynacji. Te ostatnie są oetykietowane relacjami *conjunct*, zaś sam spójnik łączy się ze swoim nadrzędnikiem relacją, którą pełniłby każdy z członów gdyby wystąpił osobno. Na przykład w zdaniu *Pies i kot śpią.* słowa *Pies* oraz *kot* są podrzędnikami spójnika *i*, jako człony koordynacji mają więc etykiety *conjunct*, podczas gdy spójnik *i* przyjmuje etykietę *subj*, tak jakby to spójnik (a tak naprawdę cała konstrukcja) pełnił rolę podmiotu, względem czasownika *śpią*. W naszym systemie ekstrakcji kolokatów, traktujemy spójniki koordynujące w sposób szczególny, niejako przeskakując przez nie w drzewie. Analizując wskazany wyżej przykład, w poczet podmiotów czasownika *spać* (a więc idąc w dół drzewa) nie zostanie zaliczony spójnik *i* (jak trzeba by zrobić aplikując reguły do drzewa takiego, jakim jest) tylko wszystkie człony które koordynuje, i.e. *pies* oraz *kot*. Obliczając zaś listę czasowników których podmiotem jest słowo *pies* (tj. idąc w górę drzewa), przejdziemy w drzewie dwa kroki, zaliczając wystąpienie czasownika *spać* zamiast pomijać go.

Należy zwrócić uwagę na to, że kolokacje nie są liczone w sposób uwzględniający negacje. Wystąpienia danego słowa, będą zaliczane w poczet tego samego kolokatu niezależnie od tego, czy są w zasięgu słowa *nie*, spójnika takiego jak *ani*, modyfikatora o leksykalnym charakterze zbliżonym do negacji (jak np. *mało* w *mało przystojny*), albo wreszcie

same są formą zanegowaną (np. imiesłów *niepoinformowany*).



Reprezentacja koordynacji, oraz negacji, w zastosowanym formalizmie składniowym. Ponieważ bezpośrednim podrzędnikiem relacji *obj* jest spójnik *i*, podczas obliczeń, przeskakujemy o poziom niżej, po relacjach z etykietą *conjunct*, aby zaliczyć w poczet *dopełnień bliższych „objawiać”* słowa *charakter* i *stanowisko*, zamiast słowa *i*. Profile słów nie są wrażliwe na to, czy więc we wskazanym przykładzie, licząc profil słowa *osoba*, czasownik *objawiać* trafi do kolumny *czasowniki których podmiotem jest „osoba”*.

1.4 Ekran statystyk

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, natomiast zawierająca pierwszy 1000 najczęściej występujących lematów.

1.4.2 Słownictwo charakterystyczne

Słownictwo charakterystyczne generowane jest przez aplikację TermoPL. Opis jej działania, instrukcja i dodatkowe informacje dostępne są na tej stronie.

W polu na stronie informacje ograniczone są do formy bazowej, wartości C-value oraz liczby wystąpień, posortowane wg C-value. Po kliknięciu przycisku „Pobierz” pobrany zostaje plik zawierający wszystkie dane wygenerowane przez TermoPL w formacie wyjściowym aplikacji.

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.

1.4.3 Wykres wg metadanych

W tym polu wyświetlany jest wykres przedstawiający podział segmentów w korpusie ze względu na wybraną metadaną, wybraną przez użytkownika.

1.4.4 Kolokacje

To pole zawiera wyznaczone najbardziej prawdopodobne związki wyrazowe dla zadanego schematu. Domyślnie wyświetlone są formy bazowe każdego ze słów składających się na dane wystąpienie, liczba wystąpień oraz symetryczne prawdopodobieństwo warunkowe. Wyświetlanych jest 50 związków posortowanych wg prawdopodobieństwa.

W pliku dostępnym do pobrania znajdują się wszystkie związki spełniające dane kryterium wyszukiwania oraz kilka miar je charakteryzujących.

Te miary to:

- Liczba wystąpień danego związku
- Prawdopodobieństwo warunkowe symetryczne.
- Maksymalne prawdopodobieństwo warunkowe.
- Miara Dice.
- Prawdopodobieństwo warunkowe obliczone dla każdego ze składników związku.
- Liczba wystąpień każdego ze składników związku.

1.4.5 Słowa kluczowe

Słowa kluczowe obliczone są na podstawie porównania listy frekwencyjnej korpusu z listą frekwencyjną korpusu referencyjnego. Korpusem referencyjnym w tym przypadku jest milionowy korpus NKJP.

1.4.6 Rozkład słów kluczowych

Wizualizacja przedstawia występowanie słów kluczowych korpusu w ramach każdego dokumentu tekstowego. Położenie i wielkość są oparte na sumach wystąpień poszczególnych słów kluczowych w kolejnych zdaniach dokumentu. Regulacja zagęszczenia pozwala na sumowanie wystąpień słów z kilku zdań do jednego punktu na wizualizacji, domyślnie ustawiona wartość zagęszczenia ma na celu zwiększenie czytelności wizualizacji.

1.5 Wykorzystane narzędzia

Korpusomat jest aplikacją agregującą już istniejące narzędzia.

Wykorzystane narzędzia to, między innymi:

- Witold Kieraś and Marcin Woliński. Morfeusz 2 – analizator i generator fleksyjny dla języka polskiego. Język Polski, XCVII(1):75–83, 2017.
- Waszczuk J., Kieraś W., Woliński M. (2018) Morphosyntactic Disambiguation and Segmentation for Historical Polish with Graph-Based Conditional Random Fields. In: Sojka P., Horák A., Kopeček I., Pala K. (eds) Text, Speech, and Dialogue. TSD 2018. Lecture Notes in Computer Science, vol 11107. Springer, Cham
- Marcińczuk, Michał; Kocoń, Jan; Gawor, Michał. Recognition of Named Entities for Polish-Comparison of Deep Learning and Conditional Random Fields Approaches In: Ogrodniczuk, Maciej; Kobyliński, Łukasz (Eds.): Proceedings of the PolEval 2018 Workshop, pp. 63-73, Institute of Computer Science, Polish Academy of Science, Warszawa, 2018.
- 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.
- Piotr Rybak and Alina Wróblewska. Semi-supervised neural system for tagging, parsing and lematization. In Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, pages 45–54. Association for Computational Linguistics, 2018.

1.6 Publikacje i wystąpienia

Szczegóły działania są dostępne w publikacjach oraz archiwalnych wystąpieniach prezentujących aplikację.

Lista materiałów:

- PlatonTV - DARIAH-PL: Sesja 3a, Łukasz Kobyliński „Korpusomat — narzędzie do tworzenia przeszukiwalnych korpusów języka polskiego”.
- Seminarium ZIL, z którego dostępne są także slajdy.

1.7 Cytowanie

W przypadku użycia w pracy naukowej użytkownik proszony jest o zacytowanie jednego z artykułów:

- Witold Kieraś, Łukasz Kobyliński. Korpusomat – stan obecny i przyszłość projektu. Język Polski, CI(2):49–58, 2021.
- Witold Kieraś, Łukasz Kobyliński, and Maciej Ogrodniczuk. Korpusomat — a tool for creating searchable morphosyntactically tagged corpora. Computational Methods in Science and Technology, 24(1):21–27, 2018.