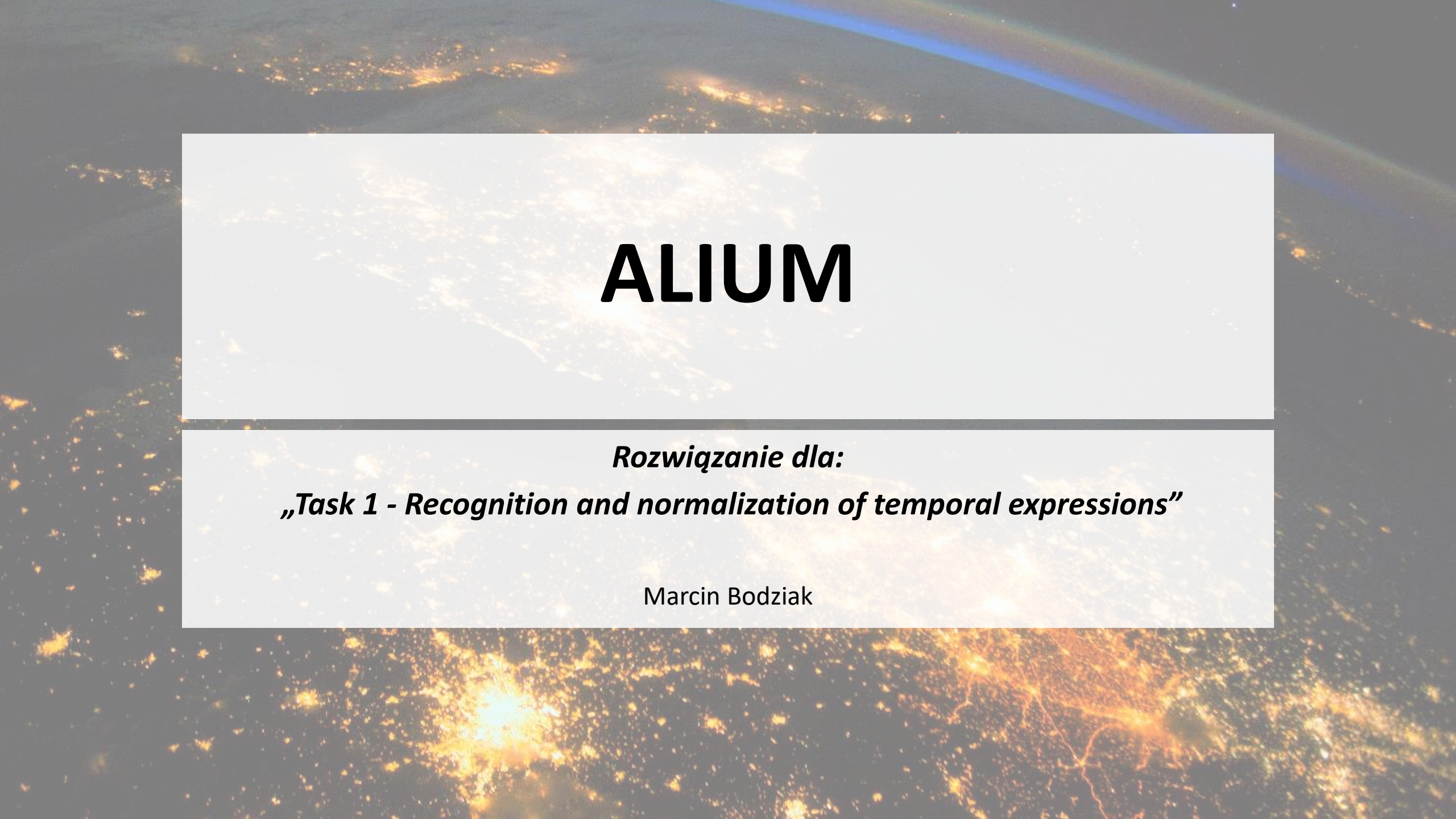




ALIUM

***Rozwiązanie dla:
„Task 1 - Recognition and normalization of temporal expressions”***

Marcin Bodziak

Przykłady wyrażeń temporalnych – 1/2

Tekst	Wyrażenie znormalizowane
13 marca 1893	type="DATE" value="1893-03-13"
dniu 1.09.2011 r.	type="DATE" value="2011-09-01"
1 poł. III w.	type="DATE" value="02-H1"
XIV-wiecznego	type="DATE" value="13"
trzy lata	type="DURATION" value="P3Y"
dwie godzinki	type="DURATION" value="PT2H"
w środę 11 stycznia 2006 r. o godz. 17:00	type="TIME" value="2006-01-11T17:00"
dwa razy do roku	type="SET" value="VAGUE"

Przykłady wyrażeń temporalnych – 2/2

Tekst	Wyrażenie znormalizowane
późnym wieczorem	type="TIME" value="2005-07-07TEV"
przyszły wtorek	type="DATE" value="1995-01-03"
tegorocznych	type="DATE" value="2004"
poprzedniego dnia rano	type="TIME" value="2005-03-20TMO"
w niedzielę o godzinie 17:15	type="TIME" value="2006-07-02T17:15"
piętnasta czterdzieści coś	type="TIME" value="2007-03-18T15:4x"

O czym chciałbym powiedzieć ...

[illegible]

Architektura rozwiązania

Silnik regułowy Alium i postprocessing:



Struktura reguły

Reguła

Atrybuty

Warunki filtrowania

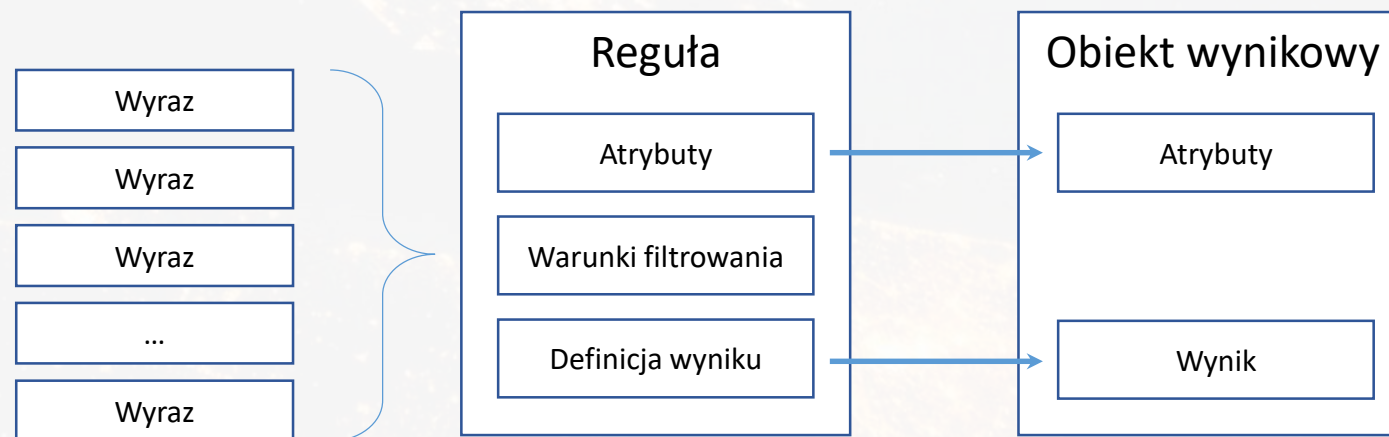
Definicja wyniku

```
#####  
# lrzym  
#####  
  
%rule:  
%position: non-absolute  
%final: no  
%name: lrzym  
  
%filter  
m1:r,rr,rrr,rrrr,rrrrr  
  
%map  
t(wartość) - f(liczba|w1)
```

```
#####  
# wyrażenie_wiek - X wiek  
#####  
  
%rule:  
%position: non-absolute  
%frame: no  
%final: no  
%name: wyrażenie_wiek  
  
%filter  
r1:lrzym  
w2:wiek,wieku,stulecie,stulecia  
  
%map  
t(date) - t(wiek) - r1:fmap(wartość)
```

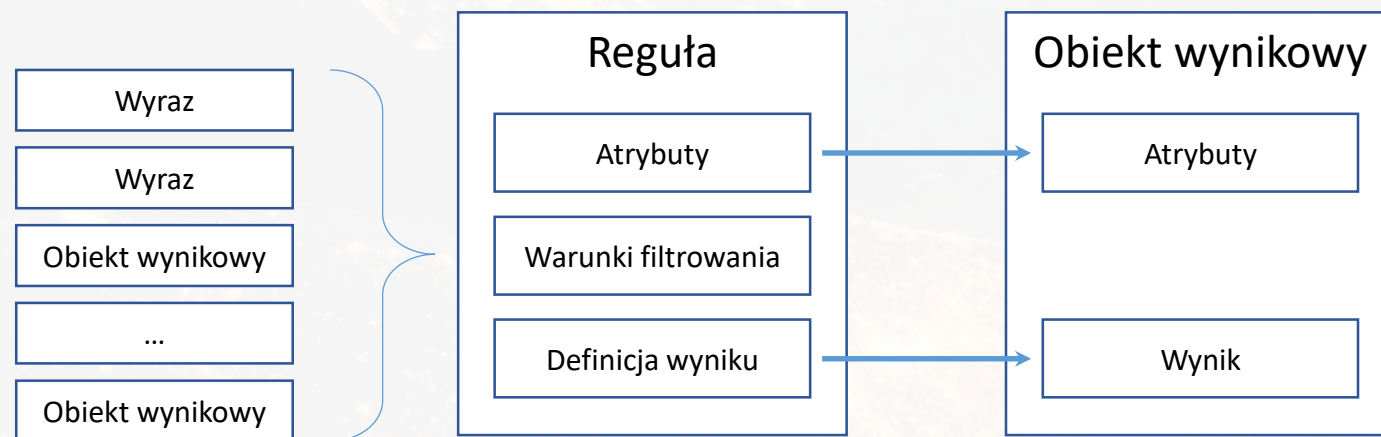

Zasada działania

- reguły proste



Zasada działania

- reguły złożone



Inne

- Hierarchia reguł – dowolnie głęboka
- Zaawansowany pruning reguł przy parsowaniu tekstu
- Możliwość korzystania z wyników tagowania (wyraz, forma bazowa, oznaczenia morfosyntaktyczne)
- Możliwość filtrowania jednostek łączących się tym samym przypadkiem, liczbą, rodzajem
- Możliwość dołączania własnych słowników, innych zbiorów (WordNet)
- Silnik – językowo agnostyczny

Podsumowanie rozwiązania dla Zadania 1

- 420 reguł, w hierarchii 6 poziomów
- prosta tokenizacja tekstu (bez taggowania)
- 14 osobo-dni pracy (1 md silnik, 8 md reguły, 5 md postprocessing)
- przetworzenie tekstu (1-10 kB) – średnio 1.2 sek

- F1 (Relaxed Match) – 86.49

Co można zrobić lepiej ...

- klasyfikacja tekstów – różne zestawy reguł
- wykorzystanie taggowania – mniejsza liczba reguł
- rozbudowanie funkcjonalności silnika – mniejsza liczba reguł, większa skuteczność reguł



DZIĘKUJĘ

Marcin Bodziak, mbodziak70@gmail.com