Warstwowy tager pamięciowy i kwestia oceny tagerów morfosyntaktycznych

Adam Radziszewski

Politechnika Wrocławska

6 maja 2013

O czym będzie mowa

Definicja zadania (1)

Segment (*token*)

Wystąpienie w tekście wyrazu, znaku interpunkcyjnego, ciągu cyfr lub symboli. Niekiedy za segment można uznać fragment wyrazu. Segmenty są ciągłe oraz rozłączne.

Tag, znacznik morfosyntaktyczny (ang. *morpho-syntactic tag*, *MSD tag*)

Symbol, który można przypisać segmentowi, określający jego własności morfologiczno-składniowe. Może określać:

1. przybliżenie *części mowy* segmentu,
2. własności o charakterze fleksyjnym (przypadek rzeczownika),
3. składniowym (z jakim przypadkiem łączy się dany przyimek),
4. czasem semantycznym (np. że dana forma jest nazwą własną).

Definicja zadania (2)

Problem znakowania ciągu (ang. *sequence labelling problem*)

Zadanie klasyfikacji polegające na przypisanie ciągowi (a_n) o elementach ze zbioru A ciągu (b_n) tej samej długości o elementach ze zbioru B (przekształcenie typu $\mathbf{L} : A^* \rightarrow B^*$).

Znakowanie morfosyntaktyczne (ang. *morpho-syntactic tagging*, *MSD tagging*)

Problem znakowania ciągu, gdzie segmentom przypisywane są tagi (albo *interpretacje morfosyntaktyczne*, tj. pary $(tag, lemat)$).

Przykład

Używam tagsetu NKJP

(1)

Kazał	*kurze*	*ścierać*	*kurze*
kazać	kura	ścierać	kurz
`praet:sg:m1:perf`	`subst:sg:dat:f`	`inf:imperf`	`subst:pl:acc:m3`

Forma *kurze*:

- w roli rzeczownika (`subst`) rodzaju żeńskiego (`f`) *kura*,
- w roli rzeczownika rodzaju męskiego nieożywionego (`m3`) *kurz*.

Tagset NKJP

prep : *gen* : *nwok*

- *prep*: klasa gramatyczna — przyimek
- *gen*: wartość wymaganego atrybutu — przypadek
- *nwok*: wartość opcjonalnego atrybutu — wokaliczność

1. Zbiór 36 symboli określających klasy gramatyczne
2. Zbiór 13 atrybutów (kategorii gramatycznych)
3. Zbiór 36 wartości atrybutów
4. Przypisanie klasie gramatycznej podzbioru atrybutów wymaganych
5. Przypisanie klasie gramatycznej podzbioru atrybutów opcjonalnych
6. Przypisanie atrybutowi zbioru wartości (zbiory są rozłączne)
7. Składnia tekstowej reprezentacji tagów

Podkorpus milionowy NKJP

Podkorpus Milionowy Narodowego Korpusu Języka Polskiego

- 1,2 mln segmentów
- znakowany ręcznie
- dostępny publicznie, otwarta licencja (GNU GPL 3.0)

Segmentów	1 215 513
Zdań	85 663
Akapitów	18 484

Tablica: Statystyki podkorpusu milionowego NKJP 1.0.

Trafność tagera

Najczęściej stosowana metoda: porównanie wyjścia tagera z korpusem wzorcowym oznakowanym ręcznie.

Trafność (ang. *accuracy*)

Procent segmentów, którym tager przypisał prawidłowe tagi.

$$Acc = \frac{|\{i : tag(i) = ref(i), 0 < i \leq N\}|}{N} \quad (1)$$

- N — liczba segmentów w korpusie, na którym przeprowadzamy ocenę
- *tag* — przyporządkowanie numerom segmentów tagów wykonane przez tager
- *ref* — przyporządkowanie tagów z korpusu wzorcowego

TaKIPI i PANTERA

TaKIPI (Politechnika Wrocławska)

- Niewielki zbiór reguł napisanych ręcznie
- Klasyfikacja kolejnych segmentów za pomocą drzew decyzyjnych
- Uczenie: algorytm C4.5

PANTERA (IPI PAN)

- Zmodyfikowany algorytm Brilla, dostosowany do języków słowiańskich
- Indukcja reguł zmieniających oznakowanie
- Zmiany na poziomie wartości pojedynczych atrybutów

TaKIPI i PANTERA: cechy wspólne

Analiza morfologiczna

- W pierwszym etapie każdemu segmentowi przypisywane są zbiory „możliwych" tagów
- Analizator Morfeusz (+ Odgadywacz)
- Drugim etapem jest ujednoznacznianie — kontekstowe wykreślanie tagów z przypisanych zbiorów
- Nie ma możliwości dodania tagu spoza zbioru

Znakowanie warstwowe (ang. *tiered tagging*)

- Niech $\Lambda = \{klasa\} \cup A$, gdzie A to zbiór symboli atrybutów, a *klasa* to symbol określający klasę gramatyczną
- $\Lambda_{\mathrm{NKJP}} = \{klasa, \mathrm{nmb}, \mathrm{cas}, \mathrm{gnd}, \mathrm{per} \ldots\}$
- Warstwą L_i nazwiemy dowolny niepusty podzbiór $L_i \subset \Lambda$
- Znakowanie warstwowe: ujednoznacznianie w N przebiegach odpowiadających warstwom; w każdym przebiegu wybierane są wartości atrybutów, które należą do warstwy

MBT

Uczenie pamięciowe (ang. *Memory-Based Learning*, *MBL*)

- Zapamiętujemy wszystkie przypadki uczące
- Klasyfikacja nowego przypadku — znajdowania w bazie przypadku podobnego
- Algorytm k najbliższych sąsiadów (k-NN)
- Zjawiskiem powszechnym w języku są wyjątki, które cechuje pewna regularność (ang. *subregularities*)

Tager pamięciowy MBT (Uniwersytet w Tilburgu)

- Klasyfikacja kolejnych segmentów w zdaniu
- Bez użycia analizy morfologicznej, wszystkie dane pochodzą z korpusu uczącego
- Bez warstw, działanie jednoprzebiegowe
- Podział na słowa znane i nieznane, dwie bazy przypadków

Koncepcja tagera WMBT

Wrocław Memory-Based Tagger

- Dostosowanie tagera MBT do specyfiki języka polskiego
- Dodajemy warstwy: tyle warstw, ile atrybutów + klasa
- Analiza morfologiczna, potem wykreślanie
- Operujemy na zbiorach wartości (wieloznaczność)
- Algorytmy uczenia i działania są bardzo proste

Uczenie tagera WMBT (1)

Dane wejściowe: korpus uczący *korp* oznakowany morfosyntaktycznie.
Każdemu segmentowi przypisany jest wynik analizy morfologicznej (zbiór tagów) oraz jeden z nich wybrany jako właściwy.
Parametry: uporządkowany ciąg atrybutów, cechy
Wyniki działania: bazy przypadków uczących B_a dla $a \in [klasa, atr_1, \ldots, atr_k]$
Algorytm:

for $zdanie \in korp$ **do**
  for $a \in [klasa, atr_1, \ldots, atr_k]$ **do**
    ...
  end for
end for

Uczenie tagera WMBT (2)

for $a \in [klasa, atr_1, \ldots, atr_k]$ **do**
  for $seg \in zdanie$ **do**
    if seg jest niejednoznaczny ze względu na a **then**
      $wek\_cech \leftarrow [f(seg, zdanie)$ **for** $f \in cechy(a)]$
      $decyzja \leftarrow$ prawidłowa wartość a dla segmentu seg
      dodaj do bazy a przykład uczący $(wek\_cech, decyzja)$
      usuń z segmentu seg tagi z nieprawidłową wartością a
    end if
  end for
end for

Generowana jest baza przypadków dla klasy gramatycznej. Potem wykreślane są tagi, które mają inną wartość klasy niż wzorcowa. Przechodzimy do $atr_1 \ldots$

Uczenie tagera WMBT (3)

Pozycja	-1	**0**	$+1$
Tagi	praet:sg:m1:perf	adj:sg:nom:n:pos adj:sg:acc:n:pos subst:sg:dat:f subst:sg:loc:f subst:pl:acc:m3 subst:pl:voc:m3	inf:imperf
Forma	*Kazał*	*kurze*	*ścierać*
Klasa gram.	{praet}	{adj,subst}	{inf}
Przypadek	{}	{nom,dat,acc,loc,voc}	{}
Decyzja dla L_1	praet	subst	inf

Przykład uczący dla segmentu *kurze*, L_1 (*klasa*):

[*Kazał*, *kurze*, *ścierać*, {praet}, {adj,subst}, {inf}, {}, {nom,dat,acc,loc,voc}, {}, subst]

Uczenie tagera WMBT (4)

Pozycja	-1	**0**	$+1$
Tagi	praet:sg:m1:perf	adj:sg:nom:n:pos adj:sg:acc:n:pos subst:sg:dat:f subst:sg:loc:f subst:pl:acc:m3 subst:pl:voc:m3	inf:imperf
Forma	*Kazał*	*kurze*	*ścierać*
Klasa gram.	{praet}	{subst}	{inf}
Przypadek	{}	{dat,acc,loc,voc}	{}
Decyzja dla L_2	sg	sg	-

Przykład uczący dla segmentu *kurze*, L_2 (nmb):

[*Kazał*, *kurze*, *ścierać*, {praet}, {subst}, {inf}, {}, {dat,acc,loc,voc}, {}, sg]

Działanie tagera WMBT (1)

Dane wejściowe: *zdanie* poddane analizie morfologicznej oraz bazy przypadków uczących B_a dla $a \in [klasa, atr_1, \ldots, atr_k]$
Każdemu segmentowi zdania przypisany jest wynik analizy morfologicznej (zbiór tagów).
Parametry: uporządkowany ciąg atrybutów, cechy, parametry klasyfikatora pamięciowego
Wyniki działania: pozostawienie dokładnie jednego tagu per segment w zdaniu
Założenia:

- z każdą bazą B_a skojarzony jest klasyfikator pamięciowy
- klasyfikujemy tylko segmenty niejednoznaczne ze względu na dany atrybut

Działanie tagera WMBT (2)

for $a \in [klasa, atr_1, \ldots, atr_k]$ **do**
  for $seg \in zdanie$ **do**
    if seg jest niejednoznaczny ze względu na a **then**
      $wek\_cech \leftarrow [f(seg, zdanie)$ **for** $f \in cechy(a)]$
      $decyzja \leftarrow$ klasyfikuj (B_a, wek_cech)
      if $decyzja \in$ możliwe wartości a pobrane z tagów przypisanych segmentowi seg **then**
        usuń z segmentu seg tagi, dla których wartość$(a) \neq decyzja$
      end if
    end if
  end for
end for
Wybierz arbitralnie jeden tag, jeśli zostało więcej

Cechy

Cechy proste:

1. Wartości klasy gramatycznej, liczby, rodzaju i przypadka z okna $(-3, -2, -1, 0, +1, +2)$.
2. Formy wyrazowe segmentów z okna $(-3, \ldots, 2)$ filtrowane do $F = 500$ najczęstszych.
3. Sufiksy formy wyrazowej na pozycji 0 o długościach 1–3.
4. Dwa testy na graficzną postać formy wyrazowej na pozycji 0: czy zaczyna się małą literą oraz czy zaczyna się wielką literą.

Testy na uzgodnienie co do liczby, rodzaju i przypadka:

1. między pozycjami -1 i 0 oraz 0 i $+1$
   swoimi ostrymi zębami, swoimi ostrymi zębami
2. testy na *słabe uzgodnienie* między segmentami z okien: $(-2, -1, 0)$, $(-1, 0, +1)$, $(0, +1, +2)$
   jedzenie całkiem smaczne

Implementacja WMBT

Użyto następujących modułów:

1. TiMBL (Tilburg Memory-Based Learner),
2. WCCL (implementacja formalizmu zapisu cech morfosyntaktycznych)
3. Corpus2 (biblioteka obsługująca tagsety, zapis/odczyt korpusów)

Sama implementacja to 421 linijek kodu w Pythonie.
http://nlp.pwr.wroc.pl/redmine/projects/wmbt/wiki
Licencja: GNU GPL 3.0
Na bazie kodu WMBT powstał później tager WCRFT oraz płytki parser IOBBER (GNU LGPL 3.0).

Procedura oceny

1. Ocena na podkorpusie milionowym NKJP 1.0
2. Korpus zawiera wyniki automatycznej analizy morfologicznej oraz ręcznie wybrane tagi uznane za właściwe
3. Tager ma dostęp do zbiorów „możliwych” tagów, jego zadaniem jest wykreślenie niechcianych
4. Miarą oceny jest trafność (wykreślania)
5. Eksperymenty przeprowadzono w oparciu o dziesięciokrotną walidację krzyżową

Wyniki oceny

Tager	*Acc*
MBT	79,31%
PANTERA	92,95%
WMBT	93,00%

Tablica: Porównanie tagerów na podkorpusie milionowym NKJP 1.0.

Tak przeprowadzona ocena została opisana w pracy:

> *Radziszewski, A. i Śniatowski, T. (2011). A memory-based tagger for Polish. W: Proceedings of the 5th Language & Technology Conference, Poznań.*

Analogicznej metody oceny użyto wcześniej w przynajmniej czterech innych publikacjach. Czy wszystko jest jednak w porządku?

Co to jest tager?

Tager to...

1. Moduł ujednoznaczniania analizy morfologicznej
   Wejście: segmenty poddane analizie morfologicznej
   Działanie: wykreślanie niechcianych tagów
2. Moduł znakowania segmentów
   Wejście: „gołe" segmenty (same formy wyrazowe)
   Działanie: przypisywanie tagów
3. Moduł przetwarzający czysty tekst
   Wejście: czysty tekst
   Działanie: wyodrębnienie segmentów, przypisywanie tagów

Oceniamy różne układy i dostajemy różne wyniki.
Kwestia precyzyjnej definicji tagera nie pojawia się w literaturze, za to przyjmuje się „po cichu" różne założenia.

Tager to moduł ujednoznaczniania analizy morfologicznej

Działanie: wykreślanie niechcianych tagów

Oceniamy: trafność wykreślania przy doskonałym słowniku

Zakładane w:

Hajič, J. i Vidová-Hladká, B. (1998). Tagging inflective languages: Prediction of morphological categories for a rich, structured tagset. W: Proceedings of the COLING - ACL Conference, strony 483–490. ACL.

Piasecki, M. (2007). Polish tagger TaKIPI: Rule based construction and optimisation. Task Quarterly, 11(1–2):151–167.

Acedański, S. (2010). A morphosyntactic Brill tagger for inflectional languages. W: Loftsson, H., Rögnvaldsson, E., i Helgadóttir, S., red., Advances in Natural Language Processing, strony 3–14. Springer.

Śniatowski, T. i Piasecki, M. (2011). Combining Polish Morphosyntactic Taggers. W: Proceedings of the 2011 International Joint Conference on Security and Intelligent Information Systems. Springer.

Radziszewski, A. i Śniatowski, T. (2011). A memory-based tagger for Polish. W: Proceedings of the 5th Language & Technology Conference, Poznań.

Tager to moduł ujednoznaczniania analizy morfologicznej

Korpus wzorcowy	Wejście	Wyjście	Dobrze?
w brev:pun prep:acc:nwok prep:loc:nwok	w brev:pun prep:acc:nwok prep:loc:nwok	w prep:loc:nwok	tak
Ramallach ign subst:sg:loc:n	Ramallach ign subst:sg:loc:n	Ramallach subst:sg:loc:n	tak

Trafność wykreślania $Acc = \frac{2}{2} = 100\%$

Forma *Ramallach* jest w rzeczywistości słowem nieznanym. Jeśli oceniamy jedynie zdolność wykreślania, to problemu nie widzimy. Strategią wygrywającą w przypadku słów nieznanych jest niewybieranie tagu `ign`. Ma się to nijak do rzeczywistości.

Tager to moduł znakowania segmentów

Działanie: przypisywanie tagów

Oceniamy: trafność znakowania przy doskonałej segmentacji

Zakładane w:

Dzeroski, S., Erjavec, T., i Zavrel, J. (1999). Morphosyntactic tagging of Slovene: Evaluating taggers and tagsets. Raport nr IJS-DP 8018, Instytut Jožefa Stefana, Lublana, Słowenia.

Schmid, H. i Laws, F. (2008). Estimation of conditional probabilities with decision trees and an application to fine-grained POS tagging. W: Proceedings of COLING 2008, volume 1, strony 777–784. ACL

Acedański, S. i Przepiórkowski, A. (2010). Towards the adequate evaluation of mor- phosyntactic taggers. W: Proceedings of the 23rd International Conference on Com- putational Linguistics (COLING), strony 1–8, Pekin, Chiny.

Daelemans, W., Zavrel, J., Van den Bosch, A., i van der Sloot, K. (2010). MBT: Memory-Based Tagger, version 3.2. Raport nr 10-04, ILK.

Tager to moduł znakowania segmentów

Korpus wzorcowy	Wejście	Wyjście	Dobrze?
w brev:pun prep:acc:nwok prep:loc:nwok	w	w prep:loc:nwok	tak
Ramallach ign subst:sg:loc:n	Ramallach	Ramallach subst:sg:nom:m3	nie

Trafność znakowania $Acc = \frac{1}{2} = 50\%$

Nasz przykładowy tager na wejście dostał nieoznakowane segmenty i nie poradził sobie ze słowem nieznanym.

Uwaga: rozważania dotyczą jedynie wytyczenia granic ocenianego przez nas układu. Tutaj zakładamy, że układ na wejście dostaje nieoznakowane segmenty, ale w środku może używać dowolnych technik, w tym analizatora morfologicznego.

Tager to moduł przetwarzający czysty tekst

Działanie: segmentacja i znakowanie segmentów
Oceniamy: trafność znakowania i segmentację (jak?)
Zakładane w:

> *Karwańska, D. i Przepiórkowski, A. (2011). On the evaluation of two Polish taggers. W: Goźdź-Roszkowski, S., red., The proceedings of Practical Applications in Language and Computers PALC 2009, Frankfurt, Niemcy. Peter Lang.*

Prawdopodobnie pierwsza próba oceny tagerów na czystym tekście (tu: Dębowskiego i TaKIPI). Ale:

- wyniki wyraźnie niższe niż publikowane wcześniej,
- nie podjęto próby wyjaśnienia różnic,
- fakt, że tagery porównywane były na czystym tekście, nie jest w pracy wspomniany.

Różnice w segmentacji (1)

Nie zawsze możliwe jest bezpośrednie porównanie tagów przypisanych danemu segmentowi w wariancie wzorcowym i wariance wyprodukowanym przez tager.

Założenie: korpus wzorcowy i korpus wynikający z oznakowania czystego tekstu zawierają ten sam tekst (pomimo możliwych różnic w podziale na segmenty).

Znaczna część (większość) segmentów z korpusu wzorcowego obecna będzie w korpusie wynikowym w postaci niezmienionej. Pozostałe segmenty z korpusu wzorcowego podlegają zmianie segmentacji.

Różnice w segmentacji (2)

Przykłady teoretycznie możliwych zmian segmentacji:

- [*...*] ↔ [*.*][*.*][*.*]
- [*m.in.*] ↔ [*m*][*.*][*in*][*.*]
- [*człowiek–demolka*] ↔ [*człowiek*][*–*][*demolka*]
- [*dałżebyś*] ↔ [*dał*][*że*][*byś*]
- [*void**][*ptr*] ↔ [*void*][**ptr*]
- [*Lądek Zdrój*] ↔ [*Lądek*] [*Zdrój*] (nie wystąpi w NKJP)

Założenia

1. Niektóre różnice w segmentacji są bardzo istotne
2. Inne są mniej
3. Ciężko jednak sformułować jasne kryteria
4. Trudno też porównywać tagi przypisane innym jednostkom
5. Propozycja (pochodzi od Szymona):
   1. Karzmy jednakowo wszystkie zmiany segmentacji w stosunku do korpusu wzorcowego
   2. Promujemy wysiłek włożony w uzyskanie oczekiwanej segmentacji
   3. Zniechęcamy do „kombinowania z miarami”, by opublikować wysokie wyniki

Dolne i górne ograniczenie trafności (1)

Widełki, w których mieści się rzeczywista trafność tagera. Segmenty z korpusu wzorcowego podlegające zmianie segmentacji nie są analizowane (nie sprawdzamy ich tagów).

1. Dolne ogranicznie trafności: wszystkie takie segmenty traktowane są jako nietrafione.
2. Górne ogranicznie trafności: wszystkie takie segmenty traktowane są jako trafione, niezależnie od przypisanych im tagów.

Dolne ograniczenie jest miarą o charakterze decyzyjnym, górne to hipotetyczna statystyka, której celem jest jedynie pokazanie skali problemu różnic segmentacji.

Dolne i górne ograniczenie trafności (2)

Odwzorowanie $match : \mathbb{N} \to \mathbb{N}$

- przypisuje numerom segmentów niepodlegającym zmianom segmentacji z korpusu wzorcowego numery segmentów w korpusie wynikowym,
- określone jedynie dla segmentów niepodlegających zmianie segmentacji,
- zapis $i \in match$ oznacza, że segment i-ty należy do korpusu wzorcowego oraz nie podlega zmianie segmentacji.

Dolne i górne ograniczenie trafności (3)

Dolne ograniczenie trafności

$$Acc_{lower} = \frac{|\{i : tag(i) = ref(match(i)), i \in match\}|}{N} \quad (2)$$

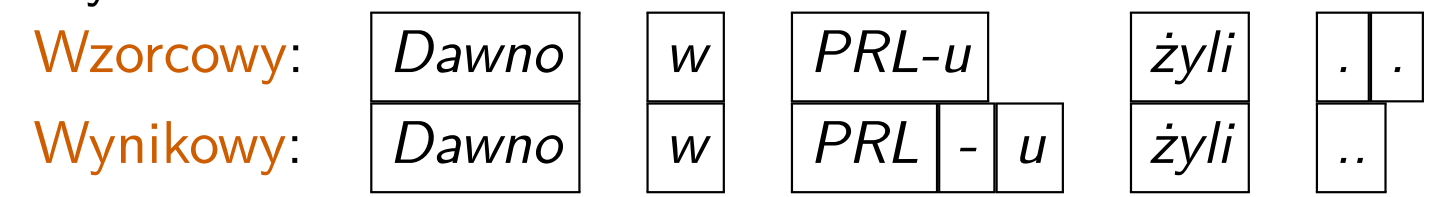
Górne ograniczenie trafności

$$Acc_{upper} = \frac{|\{i : tag(i) = ref(match(i)), i \in match\}| + N_s}{N} \quad (3)$$

$$N_s = |\{i : 0 < i \leq N \wedge i \notin match\}| \quad (4)$$

Dolne i górne ograniczenie trafności (4)

Przykład.



Korpus wzorcowy składa się z sześciu segmentów:

- *Dawno*, *w* oraz *żyli* nie podlegają zmianom segmentacji
- Pozostałe trzy podlegają: segment *PRL-u* został rozbity na trzy, segmenty . . zostały złączone.

Załóżmy, że wszystkim segmentom niepodległym zmianom tager przypisał prawidłowe tagi. Wtedy $Acc_{low} = \frac{3}{6} = 50\%$ oraz $Acc_{upper} = \frac{3+3}{6} = 100\%$.

Wyniki oceny (1)

Tager	Acc	Acc_{lower}	Acc_{upper}	Acc^{K}_{lower}	Acc^{U}_{lower}
MBT	79,31%	79,11%	79,44%	80,30%	40,49%
PANTERA	92,95%	88,79%	89,09%	91,08%	14,70%
WMBT	93,00%	87,50%	87,82%	89,78%	13,57%

Tablica: Porównanie tagerów na podkorpusie milionowym NKJP 1.0.

Acc^{K}_{lower} — tylko dla słów znanych
Acc^{U}_{lower} — tylko dla słów nieznanych

Wyniki oceny (2)

Tager	Acc	Acc_{lower}	Acc_{upper}	Acc^{K}_{lower}	Acc^{U}_{lower}
MBT	79,31%	79,11%	79,44%	80,30%	40,49%
PANTERA	92,95%	88,79%	89,09%	91,08%	14,70%
WMBT	93,00%	87,50%	87,82%	89,78%	13,57%

Tablica: Porównanie tagerów na podkorpusie milionowym NKJP 1.0.

Acc: wartości zbliżone do wcześniej publikowanych

Acc_{lower}: bliższe rzeczywistości

Prawie podwoiliśmy odsetek błędu!

Wyniki oceny (3)

Tager	Acc	Acc_{lower}	Acc_{upper}	Acc^{K}_{lower}	Acc^{U}_{lower}
MBT	79,31%	79,11%	79,44%	80,30%	40,49%
PANTERA	92,95%	88,79%	89,09%	91,08%	14,70%
WMBT	93,00%	87,50%	87,82%	89,78%	13,57%

Tablica: Porównanie tagerów na podkorpusie milionowym NKJP 1.0.

Tagery, które czytają zbiory tagów na wejściu, mogą podejrzeć prawidłowy tag dodany przez lingwistę i zyskać nieuprawnione punkty (mierzone przez Acc).

Acc_{lower} pokazuje oszacowanie rzeczywistego odsetka błędu, sprawdza się dla wszystkich tagerów.

Wyniki oceny (4)

Tager	Acc	Acc_{lower}	Acc_{upper}	Acc^{K}_{lower}	Acc^{U}_{lower}
MBT	79,31%	79,11%	79,44%	80,30%	40,49%
PANTERA	92,95%	88,79%	89,09%	91,08%	14,70%
WMBT	93,00%	87,50%	87,82%	89,78%	13,57%

Tablica: Porównanie tagerów na podkorpusie milionowym NKJP 1.0.

PANTERA i WMBT nie radzą sobie ze słowami nieznanymi.
Problem ten był dotąd niezauważony, prawdopodobnie ze względu na używaną procedurę oceny (Acc).

Wnioski

1. Wpływ metody oceny na obserwowane wyniki jest uderzający
2. Wybór metody oceny powinien być świadomy i jawny
3. Ocena tagera jako narzędzia przetwarzającego czysty tekst jest najlepszym wyborem
   - Uwzględnia błędy wynikłe z niedoskonałości analizatora
   - Uwzględnia błędy segmentacji
   - Działa jednakowo niezależnie od budowy tagera
   - Pozwala docenić jakość analizatora i modułu segmentacji
   - „Normalny użytkownik” ma do otagowania czysty tekst
4. Można taką ocenę przeprowadzić przy pomocy dolnego ograniczenia trafności
5. Warto zająć się słowami nieznanymi

Rozpoznawanie słów nieznanych

Jak tager widzi słowa nieznane?

1. Podczas działania: segmentowi analizator przypisał tylko tag `ign`.
2. Podczas uczenia: segmentowi przypisany jest tag `ign` oraz inny, wybrany jako właściwy.

Modyfikacja algorytmu:

1. Słowom nieznanym przypisujemy zbiór tagów *typowych dla słów nieznanych* pozyskany z danych uczących
2. Podwajamy liczbę baz przypadków: osobne dla znanych, osobne dla nieznanych

Wyniki

Tager	Acc_{lower}	Acc_{upper}	Acc^{K}_{lower}	Acc^{U}_{lower}
PANTERA	88,79%	89,09%	91,08%	14,70%
WMBT bez	87,50%	87,82%	89,78%	13,57%
WMBT z	88,44%	88,76%	89,89%	**41,43%**

Tablica: Wpływ modułu odgadującego nieznane słowa na wyniki tagera pamięciowego. Wyniki tagera wraz z tymże modułem umieszczono w wierszu *„WMBT z”*.

Zbiory tagów w danych uczących

WMBT (TaKIPI, PANTERA, WCRFT, Concraft) korzysta ze zbiorów tagów obecnych w danych uczących.
Należy się spodziewać najlepszych wyników, jeśli ten sam analizator będzie używany podczas działania.

Korpus wzorcowy

```
Dziwne
   dziwna   subst:pl:nom:f
   dziwna   subst:pl:acc:f
   dziwna   subst:pl:voc:f
   dziwny   adj:sg:nom:n:pos
   dziwny   adj:sg:acc:n:pos
   dziwny   adj:pl:nom:m2:pos
   dziwny   adj:pl:acc:m2:pos
   dziwny   adj:pl:nom:m3:pos
   dziwny   adj:pl:acc:m3:pos
   dziwny   adj:pl:nom:f:pos
   dziwny   adj:pl:acc:f:pos
   dziwny   adj:pl:nom:n:pos
   dziwny   adj:pl:acc:n:pos
```

Analizator

```
Dziwne
   dziwny   adj:sg:nom:n:pos
   dziwny   adj:sg:acc:n:pos
   dziwny   adj:sg:voc:n:pos
   dziwny   adj:pl:nom:m2:pos
   dziwny   adj:pl:acc:m2:pos
   dziwny   adj:pl:voc:m2:pos
   dziwny   adj:pl:nom:m3:pos
   dziwny   adj:pl:acc:m3:pos
   dziwny   adj:pl:voc:m3:pos
   dziwny   adj:pl:nom:f:pos
   dziwny   adj:pl:acc:f:pos
   dziwny   adj:pl:voc:f:pos
   dziwny   adj:pl:nom:n:pos
   dziwny   adj:pl:acc:n:pos
   dziwny   adj:pl:voc:n:pos
```

Ponowna analiza morfologiczna danych uczących

Prosty zabieg, którego celem jest lepsze wykorzystanie dostępnych zasobów.

1. Zamieniamy pierwotny korpus uczący do czystego tekstu
2. Przetwarzamy segmenterem i analizatorem → korpus pośredni
3. Synchronizujemy pierwotny korpus z pośrednim → **wynikowy**
   - Podział na segmenty i zdania bierzemy z pierwotnego
   - Uproszczenie: segmenty podlegające zmianom segmentacji bierzemy niezmienione z pierwotnego
   - Pozostałe segmenty (zdecydowana większość): zbiory tagów z pośredniego, tag wybrany z pierwotnego
   - Jeśli tag wybrany nie pojawi się w ogóle w pośrednim, to jest to słowo nieznane. Zostawiamy tylko wybrany tag oraz `ign`, by tager o tym wiedział.

Wyniki

Tager	**Re-analiza**	Acc_{lower}	Acc_{upper}	Acc^{K}_{lower}	Acc^{U}_{lower}
PANTERA	nie	88,79%	89,09%	91,08%	14,70%
	tak	88,99%	89,28%	91,27%	14,74%
WMBT bez	nie	87,50%	87,82%	89,78%	13,57%
	tak	88,75%	89,08%	91,07%	13,62%
WMBT z	nie	88,44%	88,76%	89,89%	41,43%
	tak	**89,71%**	90,04%	91,20%	41,45%

Tablica: Wpływ ponownej analizy morfosyntaktycznej danych uczących na wyniki tagerów.

Dalsze losy tagera

Tager	Acc_{lower}	Acc_{upper}	Acc^{K}_{lower}	Acc^{U}_{lower}
PANTERA	88,99%	89,28%	91,27%	14,74%
WMBT	89,71%	90,04%	91,20%	41,45%
WCRFT	90,80%	91,12%	91,96%	53,21%
Concraft	91,12%	91,44%	92,10%	59,19%

Tablica: Porównanie z nowszymi tagerami

WCRFT — warunkowe pola losowe przypisują etykiety zdaniu, poza tym podobny do WMBT
Concraft — warunkowe pola losowe z ograniczeniami, niewarstwowy

Płytki parsing języka polskiego

Znakowanie fraz (*chunking*)

- Ustalamy z góry zbiór fraz, które nas interesują
- Rozpoznajemy jedynie granice fraz w tekście
- Frazy są płaskie i rozłączne (nie nachodzą na siebie)

Rozpatrywane parsery

1. Spejd wraz z gramatyką napisaną na potrzeby NKJP
2. Klasyfikacja kolejnych segmentów przy pomocy drzew decyzyjnych (DT)
3. J.w., ale uczenie pamięciowe (MBL)
4. Znakowanie zdań przy pomocy warunkowych pól losowych (CRF)

Parsery 2–4 korzystają z zestawu cech tagera WMBT.

Metoda CRF została zaimplementowana w parserze IOBBER (GNU LGPL).

Analogia

Parsery wymagają na wejściu tekstu oznakowanego morfosyntaktycznie.
Podkorpus milionowy NKJP zawiera też płytką anotację składniową. Jak ocenić parsery na tych danych?

- Należy się spodziewać, że będą działać lepiej na danych z NKJP — ręcznie przypisano tagi
- Żeby poznać przybliżenie rzeczywistego działania parsera na tekście nieoznakowanym, musimy na nowo oznakować korpus

Analogia:

- Oceniamy tager na korpusie zawierającym wzorcową segmentację i analizę morfologiczną
- Musimy na nowo oznakować korpus

Trzy metody oceny parserów na NKJP

1. Wzorcowe oznakowanie morfosyntaktyczne z NKJP
2. Wzorcowa segmentacja i analizy morfosyntaktyczna, tager tylko wykreśla
3. Sprowadzamy korpus do czystego tekstu, przetwarzamy.

Algorytm	Wzorcowe	Ujednoznacznienie	Oznakowanie
DT	84,13%	79,96%	77,93%
MBL	85,06%	80,25%	77,99%
CRF	92,31%	88,61%	86,85%
Spejd	87,94%	81,75%	79,82%

Tablica: Trzy metody oceny algorytmów znakowania fraz na korpusie NKJP. Wszystkie podane wyniki są wartościami miary F osiągniętymi w danym teście.

Wnioski (1)

Sposób oceny tagera ma znaczenie

1. Oceniamy różne układy
2. Uzyskujemy różne wyniki
3. Różnice są duże

Warto oceniać cały układ, począwszy od segmentacji

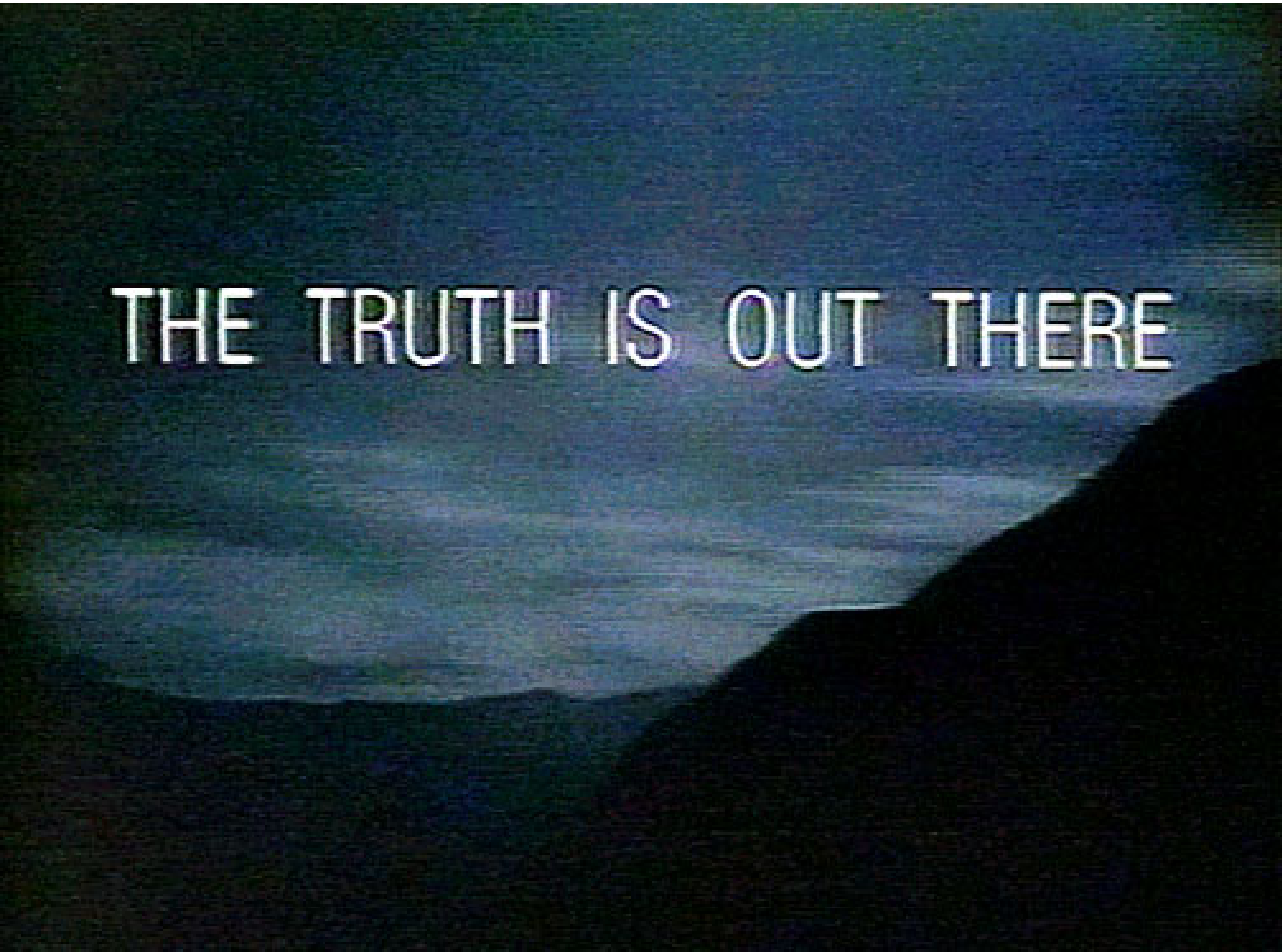
1. Typowy użytkownik dysponuje tekstem nieoznakowanym
2. Zmiany w segmentacji mogą mieć wpływ na zachowanie innych komponentów
3. Możemy ocenić analizator morfologiczny i algorytmy odgadywania
4. Metoda uniwersalna, działa dla każdego tagera

Wnioski (2)

Proponowane miary

1. Dolne ograniczenie trafności
2. Górne ograniczenie trafności

Dostępna implementacja: skrypt `tagger-eval.py` dostarczany razem z biblioteką Corpus2
`http://nlp.pwr.wroc.pl/redmine/projects/corpus2/wiki`


THE TRUTH IS OUT THERE