POLITECHNIKA WROCŁAWSKA
WYDZIAŁ INFORMATYKI I ZARZĄDZANIA

Automatyczne wstawianie znaków interpunkcyjnych do tekstu.

Analiza języka naturalnego

Grupa A
Maciej Urbaniak (200842)
Aleksandra Orzechowska (223379)

WROCŁAW 2019

Spis treści

1. Wstęp

W ciągu ostatnich kilku lat badania w obszarze analizy języka naturalnego bardzo się rozwinęły. Język polski również nie jest pomijany w tej analizie, jednak interpunkcja wciąż jest dziedziną która nastręcza wiele trudności. Wstawianie znaków interpunkcyjnych, a w szczególności przecinków, jest jednym z częstszych błędów popełnianych przez samych Polaków, jak i obcokrajowców.

Większość przeanalizowanej literatury opiera się na języku angielskim, natomiast w poniższym raporcie bazowano na regułach interpunkcji z języka polskiego, które często są skomplikowane i trudne do ujednoznacznienia.

1.1. Cel projektu

Celem projektu było automatyczne wstawianie znaków interpunkcyjnych, w szczególności przecinków, z zastosowaniem zasad interpunkcji w języku polskim.

Jako dane początkowe otrzymujemy tekst składający się ze słów, ale jest on pozbawiony znaków interpunkcyjnych z wyjątkiem kropek kończących zdania. Natomiast wejściem do modelu jest pojedyncze zdanie, tzn. że nie rozpatrujemy słów występujących przed rozpoczęciem i po zakończeniu określonego zdania.

W pracy nie uwzględniono jednej z funkcji przecinka, którą jest zapewnienie tekstowi jednoznaczności. Wymagało by to analizy sensu zdania, co znacznie wychodzi poza możliwości zastosowanych narzędzi oraz często zależy od autora.

2. Opis użytych danych

Dane użyte w projekcie pochodzą ze zbioru "1000 novels corpus"(1) oraz strony `http:\wolnelektury.pl`. Książki wybrane zostały przede wszystkim pod kątem posiadania jak najlepszej interpunkcji oraz współczesnego języka. Utwory Bolesława Prusa są pisane prozą i w przeważającej mierze zawierają prosty oraz w miarę współczesny język. Dodatkowo autor posiadał doświadczenie dziennikarskie jak również kronikarskie. Należy zwrócić uwagę, że mogła nastąpić korekta tekstów przez edytora, co mogło wpłynąć na różnicę między różnymi wydaniami książek.

Wybrano następujące książki:

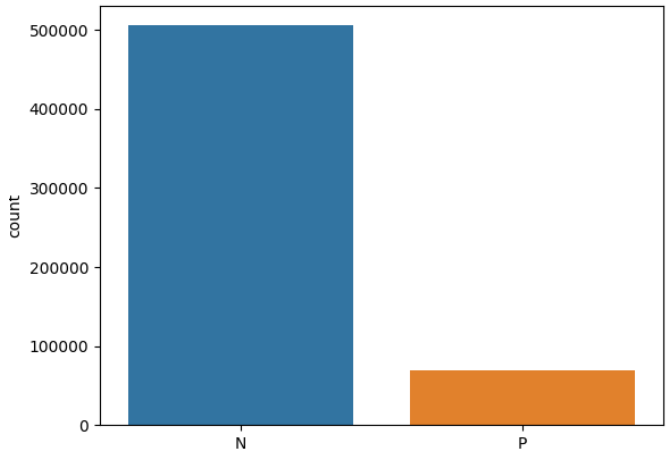
- defoe_przypadki_robinson-crusoe_1868
- deotyma_panienka-z-okienka_1898
- deotyma_zwierciadlana-zagadka_1990
- descartes_rozprawa-o-metodzie_1918
- prus_anielka_1935
- prus_badzmy-ostrozni
- prus_bajka_1935
- prus_bywa-i-tak-na-swiecie
- prus_cienie_1935
- prus_co-sie-z-wielkiej-idei-zrobilo-w-malym-miasteczku
- prus_czego-faust-narobil-w-pewnej-aptece
- prus_dziwna-historja_1935
- prus_echa-muzyczne_1935
- prus_grzechy-dziecinstwa_1935
- prus_kamizelka_1935
- prus_katarynka_1935
- prus_lokator-poddasza
- prus_michalko_1919
- prus_milknace-glosy_1935
- prus_na-pograniczu_1935
- prus_na-saskiej-kepie_1935
- prus_nawrocony_1935

— prus_nowe-prady_1935
— prus_nowy-rok_1935
— prus_ogrod-saski_1935
— prus_omylka_1927
— prus_on
— prus_opowiadanie-lekarza_1935
— prus_orestes-i-pylades_1935
— prus_pan-dudkowski-i-jego-folwark_1935
— prus_pan-wesolowski-i-jego-kij_1935
— prus_piesn-swiata_1935
— prus_placowka_1935
— prus_pod-szychtami_1935
— prus_podwojny-czlowiek_1935
— prus_pojednani_1935
— prus_powiastki-cmentarne_1935
— prus_powracajaca-fala_1935
— prus_przeklete_szczescie_1935
— prus_przygoda-stasia_1930
— prus_przy-ksiezycu_1935
— prus_sen_1935
— prus_sieroca-dola_1935
— prus_slawa_1935
— prus_stara-bajka_1935
— prus_sukienka-balowa_1935
— prus_szkatulka_babki_1935
— prus_wakacje_1935
— prus_w-gorach_1935
— prus_widzenie_1935
— prus_widziadla_1935
— prus_wigilja_1935
— prus_w-walce-z-zyciem_1935
— prus_ze-wspomnien-cyklisty_1935
— prus_z-legend-dawnego-egiptu_1935
— prus_zywy-telegraf_1935

W celu ograniczenia oraz próby zbalansowania zbioru wybrano wyłącznie zdania zawierające chociaż jeden przecinek.

	zbiór oryginalny	zbiór zredukowany
liczba słów	647118	517821
liczba przecinków	68848	68848

Zbiór danych składał się ze słów, które oznaczone były jako te, po których występuję przecinek (P) lub nie (N). Rysunek 2.1 przedstawia liczność każdej z klas. Widać, że pomimo wcześniejszego wstępnego przetworzenia i zbalansowania danych różnica w dystrubucji jest znacząca.



Rysunek 2.1: Dystrybucja klas

3. Przegląd literatury

Pierwsza praca ”Recovering Casing and Punctuation using Conditional Random Fields”(2) opisuje zwycięskie podejście do problemu w konkursie ”ALTA Shared Task 2013”, którym było przywrócenie interpunkcji w ”zdegradowanym” angielskim tekście. Metodologia jaką się posłużyli to CRF(Conditional Random Fields) o cechach takich jak: POS(Part Of Speech), CHUNK(chunking) i NER(Name Entity Recognition) z programu SENNA `https://ronan.collobert.com/senna/`. Uzyskując najlepszy wynik dla przecinków na poziomie 0.597 mierzony miarą F-score.

Drugi dokument ”How do you correct run-on sentences it’s not as easy as it seems.”(3) również opiera się na CRF i zajmuje się korekcją tekstu. Z nowych cech można wymienić: czy wyraz zaczyna się wielką literą oraz konieczność wystąpienia cechy 5-krotnie aby była brana pod uwagę.

Trzecie podejście opisane w ”Correcting comma errors in learner essays, and restoring commas in newswire text.”(4) to użycie modelu CRF i etykietowania sekwencji zdań. Przez klasyfikator pod uwagę brana jest każda spacja między słowami i wstawiany jest przecinek albo nie. Zastosowano następujący zbiór cech: unigram, bigram, trigram, pos_uni, pos_bi, pos_tri, combo, first_combo, bos_dist, eos_dist, prevCC_dist, nextCC_dist. Etykiety POS użyto na oknie o długości 5 słów. Cecha kombinacji (combo) jest unigramem słowo+pos dla każdego słowa. Ostatnie cztery cechy odnoszą się do odległości. Są to następujące odległości: od początku zdania, do końca zdania, z poprzedniej i następnej uzgodnionej koniunkcji. Z danych testowych usunięte zostały wszystkie przecinki, następnie tekst jest tokenizowany i POS etykietowany używając maksymalnej entropii. Każda etykieta jest brana pod uwagę podczas klasyfikacji i decydowane jest czy wstawić przecinek czy nie. Uyzyskane wyniki dla korekcji przecinków wyniósły średnio 89% dla miary precision oraz 25% dla recall. Danymi były eseje napisane przez osoby na różnym stopniu znajomości angielskiego, w tym również rodzimych jego użytkowników.

Czwarta praca ”Deep Learning for Punctuation Restoration in Medical Reports”(5) opisuje odmienne podejście wykorzystujące sieci rekurencyjne. Na początku dane zamieniono na sekwencje tokenów ze znakiem interpunkcji jako etykieta. Przeprowadzono preprocesing tekstu jak np. normalizacje tekstu zamieniając wszystkie cyfry, daty, godzinę, itp. na “D” czy wszystkie litery na małe, przeprowadzając segmentacje. Następnie przeprowadzono redukcję słownika, co spowodowało znaczący spadek w rozmiarze słownika. Model oparty jest na

dwukierunkowej rekurencyjnej sieci neuronowej, B-RNN. B-RNN jest skuteczna w nauce zależności słów znajdujących się daleko od siebie po prawej i lewej stronie. Przesuwne okno zawierające 256 słów jest przekazywane do warstwy jako wektor cech (one-hot vector). Na końcu dodana jest warstwa RNN z mechanizmem uwagi, aby pomóc w uchwyceniu odpowiedniego kontekstu, które pozwolą zdecydować o ustawieniu odpowiedniego znaku interpunkcji. W pracy tej zaimplementowano również krok, w którym rzadkie słowa mapowane są na klasy słów popularnych. Zredukowało to rozmiar słownika i zmniejszyło liczbę parametrów modelu, co było kluczowe dla szybkości modelu. Uzyskany wynik to 83.1% F-score co wydawało nam się dość zaskakujące, dodatkowo długość okna była bardzo duża(256 słów). Z racji skomplikowania tego podejścia nie zdecydowaliśmy się na próbę rozwiązania naszego problemu za pomocą sieci rekurencyjnych.

4. Wykorzystane narzędzia programistyczne

4.1. Toki

Do podziału dokumentów na zdania wykorzystano toki `http://nlp.pwr.wroc.pl/narzedzia-i-zasoby/narzedzia/toki`. Poniższy kod realizuje podział na zdania do nowych linii i usuwa przecinki.

```
for file in books/*.txt
do
    toki-app -S /apps/toki/config/segment.srx -l pl_one \
    --srx-begin-marker="[[[" --srx-end-marker="]]]" < "$file" \
    > sentenced.txt
    sed 's/]]]/\n/g' sentenced.txt | sed 's/[\[[[]//g' \
    | tr -d , > end"${file%%.*}"
done
```

4.2. WCRFT

Do klasyfikacji słów na części mowy (POS tagging) wykorzystano WCRFT `http://nlp.pwr.wroc.pl/redmine/projects/wcrft/wiki/`. Dokładny opis znaczników morfosyntaktycznych znajduje się na stronie Narodowego Korpusu Języka Polskiego `http://nkjp.pl/poliqarp/help/plse2.html`. Poniżej prezentujemy kod źródłowy za pomocą którego dokonaliśmy tagowania.

```
for file in endbooks/*
do
    wcrft-app nkjp_e2.ini -i text "$file" \
    -o ccl > "${file%%.*}".tag
done
```

4.3. CRF

Jako główny model wybrano CRFSUITE `https://python-crfsuite.readthedocs.io/en/latest/`, który zamieniono następnie na pycrfsuite z pakietu sklearn `https://sklearn-crfsuite.readthedocs.io/en/latest/`. Głównie z powodu lepszej dokumentacji oraz dodatkowych funkcji, takich jak np. informacja o cechach mających największy wpływ na klasyfikację do wybranej klasy.

CRF (Conditional random field) oparta jest na grafach skierowanych gdzie wierzchołki to stany a krawędzie są zależnościami pomiędzy stanami. Przy czym modeluje on sekwencję biorąc pod uwagę to iż cechy zależą od siebie oraz korzysta z przyszłych obserwacji do wyuczenia się wzorca. W przeciwieństwie do ukrytych modeli Markov'a które zakładają niezależność cech oraz "conditional Markov model", uwzględniający te zależności, ale nie rozpatrujący przyszłych obserwacji.

5. Badania

W celu użycia klasyfikatora CRF, zdefiniowano zbiór cech, które opisują badane dane i mogą być pomocne we wskazywaniu pozycji przecinka. Poniżej przedstawiono cechy brane pod uwagę podczas projektowania modelu:

— długość okna przesuwnego: liczba słów występujących przed i po badanym słowie,
— słownik zawierający słowa takie jak np. spójniki które mogą wpłynąć na obecność przecinka
— liczba: określa czy słowo jest liczbą
— imiesłów: określa czy dane słowo możemy podejrzewać o bycie imiesłowem
— BOS: początek zdania
— EOS: koniec zdania
— część mowy (POS tagging)
— czy słowo rozpoczyna się wielką literą

5.1. Model podstawowy

Model podstawowy dla którego zmieniamy parametry w poniższych testach został opisany w poniższej tablicy. Wykorzystano wszystkie powyżej opisane cechy przy czym dla słów krańcowych nie wyznaczamy cech opartych o słowniki w celu zaoszczędzenia pamięci. W przypadku wystąpienia słowa ze słownika generujemy dodatkową cechę którą jest ten wyraz. Dalsze testy zostały przeprowadzone z wykorzystaniem kroswalidacji 8-krotnej, ponieważ podział ten uzyskał najlepsze wyniki podczas testów (5.1).

Parametr	Wartość
Algorytm	lbfgs
c1	0.01
c2	0.11
max_iterations	50
Długość okna	5

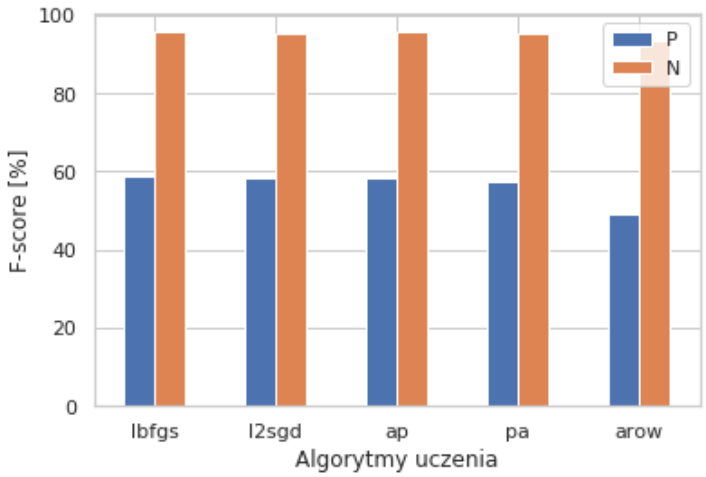
Tablica 5.1: Parametry modelu podstawowego

Rysunek 5.1: Wykres wpływu liczby podziałów (*ang. folds*)

W celu lepszego dobrania modelu przetestowano różne wartości dla parametru określającego algorytm uczenia. W Tablicy 5.2 przedstawiono wyniki dla dostępnych algorytmów. Widoczne jest, że najlepsze wyniki uzyskał algorytm 'lbfgs' (*Limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm*). Z tego powodu był on wykorzystywany podczas analizy innych opisanych poniżej parametrów.

	P			N		
Algorytm	Precision	Recall	F-score	Precision	Recall	F-score
lbfgs	80.63%	46.52%	58.97%	93.08%	98.47%	95.70%
l2sgd	75.48%	47.49%	58.16%	93.17%	97.86%	95.46%
ap	78.60%	46.21%	58.18%	93.05%	98.28%	95.59%
pa	77.99%	45.46%	57.42%	92.95%	98.25%	95.53%
arow	51.32%	46.93%	49.01%	92.85%	93.94%	93.39%

Tablica 5.2: Badanie wpływu zastosowania różnych algorytmów uczenia w CRF



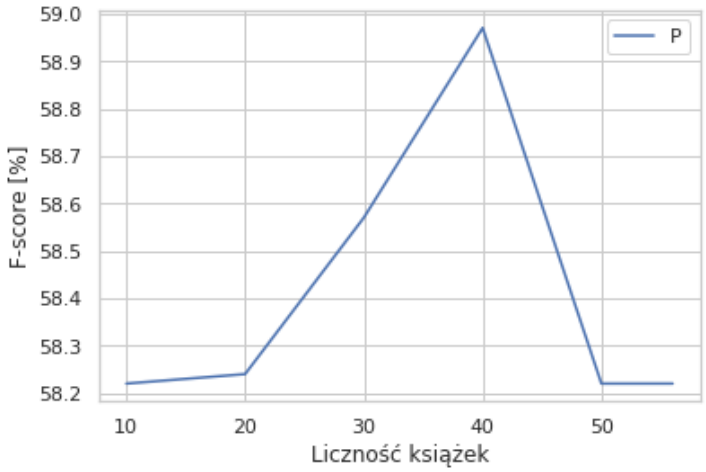
Rysunek 5.2: Wykres wpływu zastosowania różnych algorytmów uczenia

5.2. Wpływ liczności zbioru

W Tablicy 5.3 i na Rysunku 5.3 pokazano wpływ liczności zbioru(liczba książek) na uzyskane wyniki.

	P			N		
Liczba książek	Precision	Recall	F-score	Precision	Recall	F-score
10	80.04%	45.77%	58.22%	93.02%	98.44%	95.65%
20	79.98%	45.85%	58.24%	92.90%	98.40%	95.57%
30	79.86%	46.28%	58.57%	92.99%	98.38%	95.61%
40	80.63%	46.52%	58.97%	93.08%	98.47%	95.70%
50	80.03%	45.78%	58.22%	93.00%	98.44%	95.64%
56	80.04%	45.77%	58.22%	93.02%	98.44%	95.65%

Tablica 5.3: Badanie wpływu liczności książek w zbiorze uczącym na wyniki



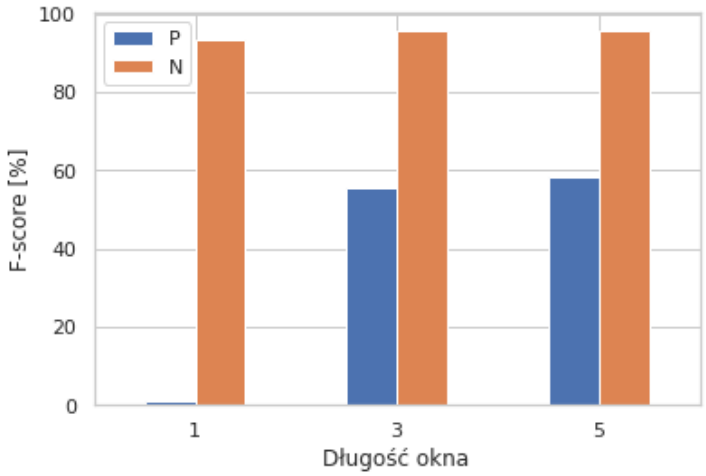
Rysunek 5.3: Wykres wpływu liczności książek na wynik klasy P

5.3. Wpływ długości okna

Długość okna określa dla ilu słów zostaną utworzone cechy. Przy czym jeśli dany wyraz znajdzie się poza zdaniem otrzymuje on tylko jedną cechę: ”BOS” w przypadku gdy jest wcześniej niż początek zdania, a ”EOS” gdy później niż jego koniec(lub jest znakiem kropki kończącej zdanie).

	P			N		
Długość okna	Precision	Recall	F-score	Precision	Recall	F-score
1	67.10%	0.60%	1.2%	88.08%	99.93%	93.63%
3	86.03%	40.91%	55.43%	92.24%	99.09%	95.67%
5	80.04%	45.77%	58.22%	93.02%	98.44%	95.65%

Tablica 5.4: Badanie wpływu długości okna



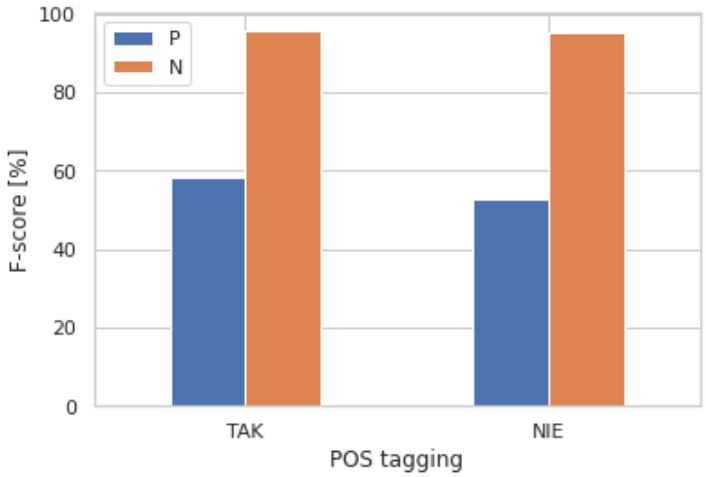
Rysunek 5.4: Wykres wpływu długości okna

5.4. Wpływ części mowy (POS)

Zgodnie z przewidywaniami wykorzystanie tagera do określania części mowy wyrazów pozytywnie wpływa na wyniki klasyfikacji znaku przecinka.

	P			N		
POS tagging	Precision	Recall	F-score	Precision	Recall	F-score
TAK	80.04%	45.77%	58.22%	93.02%	98.44%	95.65%
NIE	78.60%	40.01%	53.02%	92.34%	98.51%	95.33%

Tablica 5.5: Badanie wpływu oznaczenia części mowy (POS tagging)



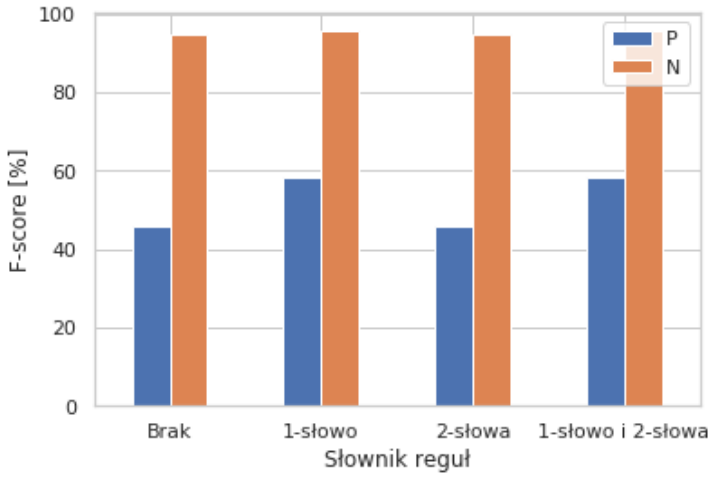
Rysunek 5.5: Wykres wpływu oznaczenia części mowy (POS tagging)

5.5. Wpływ słowników

Stworzenie słownika wyrazów, które podejrzewamy o możliwość wystąpienia w ich sąsiedztwie przecinka ma znacząco pozytywny wpływ na jakość klasyfikacji.

	P			N		
Słownik reguł	Precision	Recall	F-score	Precision	Recall	F-score
Brak	67.97%	34.87%	46.08%	91.68%	97.76%	94.62%
1-słowo tylko	79.82%	45.96%	58.32%	93.04%	98.41%	95.65%
2-słowa tylko	68.27%	34.64%	45.95%	91.66%	97.80%	94.63%
1-słowo i 2-słowa	80.04%	45.77%	58.22%	93.02%	98.44%	95.65%

Tablica 5.6: Badanie wpływu oznaczania słownika reguł interpunkcyjnych z `https://sjp.pwn.pl`



Rysunek 5.6: Wykres wpływu oznaczenia słownika reguł interpunkcyjnych

5.6. Błędne klasyfikacje

W celu lepszego zroumienia jakie błędy popełnia model wybrano zdania ze zbioru testowego, które zostały błędnie sklasyfikowane. Poniżej przedstawiono kilka przykładów.

Błędne: *Walentemu kiedy przyjdzie mi winszować pokaże cierpki grymas gospodarzowi wytoczę proces za to że jeszcze nie dał mi piwnicy...*
Poprawne: *Walentemu kiedy przyjdzie mi winszować, pokaże cierpki grymas, gospodarzowi wytoczę proces za to że jeszcze nie dał mi piwnicy...*

Błędne: *Ja osoba, taka porządna tak sławna ja podpora i współpracownik tylu pism periodycznych*
Poprawne: *Ja osoba taka porządna, tak sławna, ja podpora i współpracownik tylu pism periodycznych*

Błędne: *Gdybym mógł zmiażdżyłbym księżyc, na tabakę ziemię na jakie sto lat cofnąłbym biegu, a słońce zamroził*
Poprawne: *Gdybym mógł, zmiażdżyłbym księżyc na tabakę, ziemię na jakie sto lat cofnąłbym biegu, a słońce zamroził*

Z tych trzech przykładów można już zauważyć, że model nie radzi sobie ze zdaniami złożonymi, bądź gdzie kontekst wypowiedzi jest niejednoznaczny. Przy pierwszym przykładzie, okno o długości 5 nie było prawdopodobnie w stanie objąć dwóch rzeczowników (Walentego i gospodarza). W drugim przykładzie z powodu braku cechy która zapamiętuje wyrazy nie jesteśmy w stanie wykryć powtórzeń (w tym przypadku ”ja”). Natomiast w ostatnim poprawnie wstawiono dwa znaki, ale kwestia interpunkcji zależy również od ”zamysłu” autora i chęci zaakcentowania pewnych treści co jest bardzo trudne do wykrycia.

6. Podsumowanie

Wynikami zbliżyliśmy się do pracy ”Recovering Casing and Punctuation using Conditional Random Fields”(2). Uzyskaliśmy najlepszy wynik równy 58.22%, natomiast w omawianym dokumencie było to 59.7% mierzone miarą F-score. Niestety, są to tylko wartości poglądowe, ponieważ poza innymi językami dla których były przeprowadzone badania różniliśmy się zastosowanymi danymi oraz celem jakim w omawianej pracy było również odtworzenie innych usuniętych znaków. Powodem dla którego porównujemy się z tą pracą jest fakt, że w literaturze nie opisano podobnych badań przeprowadzonych dla interpunkcji w języku polskim.

Jako dodatkowe badanie można było by przebadać wpływ chunkera jako nowej cechy na wyniki zastosowanego modelu, ale należy zwrócić uwagę iż chunkery korzystają ze znaków interpunkcyjnych. Dlatego w zależności od tego czy w tekście występują przecinki zależeć będzie wynik płytkiej analizy. Inne proponowane ulepszenie to wykorzystanie ”Named Entity Recognition”. Można również rozszerzyć badania na określenie wpływu kategorii gramatycznych z wykorzystanego tagera morfosyntaktycznego na wyniki.

Największymi napotkanymi niedogodnościami w modelu CRF były niemożliwość jego ”douczenia” oraz duże zużycie pamięci z powodu tworzenia długich wektorów cech. Co skutkuje tym, że gdy chcemy wprowadzić nowo pozyskane dane, koniecznością staje się przeprowadzenie długotrwałego procesu tworzenia modelu od początku. Dodatkowo ograniczenia sprzętowe sprawiły, że liczność danych, na których testowany był model została znacznie zmniejszona.

Każda część wykorzystana w naszej propozycji rozwiązania problemu automatycznego wstawiania interpunkcji wpływa na końcowy rezultat. W tym przypadku jakość tagera morfosyntaktycznego jest wysoka, ponieważ stworzony został on z myślą o języku polskim, więc w przypadku zmiany języka konieczne jest całkowite przebudowanie struktury rozwiązania i zastosowanie nowych narzędzi.

Innym rozwiązaniem w przypadku automatycznego wstawiania znaków interpunkcyjnych byłoby wybranie sieci rekurencyjnych. Pozwoliłyby one na znaczne rozszerzenie długości okna, ale wymagałyby znacznie powiększonego i poprawnie otagowanego zbioru uczącego.

Bibliografia

[1] Eder, Maciej; Rybicki, Jan; Młynarczyk, Ksenia; et al., 2016, 1000 Novels Corpus, CLARIN-PL digital repository, http://hdl.handle.net/11321/312.

[2] Lui, Marco, and Li Wang. "Recovering casing and punctuation using conditional random fields." Proceedings of the Australasian Language Technology Association Workshop 2013 (ALTA 2013). 2013.

[3] Zheng, Junchao, et al. "How do you correct run-on sentences it's not as easy as it seems." arXiv preprint arXiv:1809.08298 (2018).

[4] Israel, Ross, Joel Tetreault, and Martin Chodorow. "Correcting comma errors in learner essays, and restoring commas in newswire text." Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, 2012.

[5] Salloum, Wael, et al. "Deep learning for punctuation restoration in medical reports." BioNLP 2017 (2017): 159-164.

[6] Strona internetowa "Słownik języka polskiego" https://sjp.pwn.pl (data dostępu 24.12.2018)

[7] Fundacja Nowoczesna Polska, Wolne Lektury, https://wolnelektury.pl/ (data dostępu: 2.12.2018)

Spis rysunków