Wykorzystanie sieci neuronowych do upraszczania tekstów w języku polskim

(Neural networks in text simplification problem for polish language)

Michał Doros

Praca licencjacka

Promotor: dr Paweł Rychlikowski

Uniwersytet Wrocławski
Wydział Matematyki i Informatyki
Instytut Informatyki

1 września 2022

Streszczenie

Praca bada różnice ilościowe języków przyjętych jako polski prosty oraz polski normalny. Następnie podejmuje się zdefiniowania miary prostoty oraz stworzenia klasyfikatora. Finalnie bada, jak z zadaniem upraszczania radzą sobie różne metody – zarówno heurystyczne jak i wykorzystujące sieci neuronowe

---

The paper investigates the quantitative differences between languages defined as simple Polish and normal Polish. It then undertakes the task of defining a measure of simplicity and creating a classifier. Finally, it examines how different methods – both heuristic and those using neural networks deal with the task of simplification.

Spis treści

Rozdział 1.

Wprowadzenie

Język polski jest powszechnie uznawany za język trudny. Narzędzie zmniejszające poziom skomplikowania tekstu tak, aby był on zrozumiały dla średniozaawansowanego użytkownika, przy jednoczesnym zachowaniu przekazywanej informacji, pozwalałoby na tworzenie ciekawych materiałów do nauki, ułatwiłoby zrozumienie komunikatów obcokrajowcom, mogłoby też pomóc rodzimym użytkownikom zrozumieć przekaz nadmiernie skomplikowanych dokumentów lub oficjalnych wypowiedzi.

W poniższej pracy podejmuję próbę zdefiniowania, co to znaczy, że język jest prosty oraz ilościowego odróżnienia takiego języka od występującego powszechnie. Na podstawie poczynionych obserwacji proponuję miarę prostoty języka. W dalszej części pracy implementuję różne metody upraszczania i badam jakie dają rezultaty w oparciu o stworzoną metrykę.

Rozdział 2.

Badanie prostego tekstu

Głównym celem tego rozdziału jest zdefiniowanie miary prostoty. Chcemy, aby wyniki otrzymywane z jej zastosowania były spójne z subiektywnym odczuciem rodzimych użytkowników języka. Miara spełniająca tę własność pozwala w ścisły i ilościowy sposób zdefiniować język prosty, co ma kluczowe znaczenie przy rozwijaniu modeli upraszczających teksty oraz badaniu ich efektywności.

Aby tego dokonać najpierw zdefiniuję język prosty, a następnie zaproponuję różnego rodzaju statystyki pozwalające odróżnić taki język od języka normalnego. Na koniec wybiorę najlepsze z tych statystyk, do wytrenowania klasyfikatorów, które posłużą później jako wspomniana wyżej miara prostoty.

2.1. Definicja

Jako pierwowzór języka prostego w tej pracy przyjmiemy język, którym pisane są książki dla dzieci od 6 do 8 oraz 9 do 12 roku życia. Są to najniższe kategorie wiekowe, w których książki składają się już głównie z tekstu i są na tyle obszerne, aby zawierać reprezentatywną próbkę języka zrozumiałego dla młodych czytelników. Dodatkowo ich dostępność jest duża, co ułatwia stworzenie odpowiedniego korpusu.

2.2. Korpusy

Stworzony na potrzeby pracy korpus języka prostego składa się z około 50 książek dla dzieci dostępnych w internecie, które zawierają w sumie około 200 tysięcy zdań. Książki z tego zbioru były pisane przez 8 różnych autorów i są głównie o tematyce przygodowej, obyczajowej oraz fantasy.

Jako korpusu kontrolnego, reprezentującego język normalny, nieuproszczony, będę używać korpusu POLEVAL[11] składającego się 23 milionów przetasowanych zdań w języku polskim pochodzących z różnych źródeł oraz statystyk unigramowych

Narodowego Korpusu Języka Polskiego[12][1], składających się z par (słowo, liczba wystąpień w pełnym korpusie).

2.3. Narzędzia

Do badań języka wykorzystany został język Python (wersja 3) wraz z biblioteką spaCy[2]. Narzędzie to dostarcza model języka polskiego pozwalający m.in. na podział dokumentu na tokeny i zdania, policzenie zanurzeń słów, lematyzację, rozpoznawanie części mowy oraz bardziej zaawansowaną analizę składniową. Wyjaśnienie i omówienie wymienionych technik znajduje się w kolejnych sekcjach.

2.3.1. Zanurzenia słów

Zanurzenia słów są reprezentacją słownictwa dokumentu, która jest w stanie uchwycić kontekst w jakim dane słowa występują. W najprostszej wersji dokonuje się tego poprzez przypisanie każdemu słowu pojedynczego wektora z $\mathbb{R}^n$ (gdzie n wynosi zazwyczaj od 100 do kilkuset) Idea jest taka, że słowa, które występują razem, mają zbliżone znaczenie, oraz że słowa o zbliżonym znaczeniu powinny znaleźć się blisko siebie. O współrzędnych wektorów można też myśleć jak o poziomie nasilenia pewnych abstrakcyjnych cech słowa – objawia się to tym, że zanurzenia spełniają relacje typu $V_{mama} - V_{tata} + V_{krol} \approx V_{krlowa}$ gdzie V_a oznacza wektor przypisany słowu 'a'.

W pionierskiej pracy[1] do wytrenowania takich zanurzeń użyto płytkiej sieci neuronowej, jednak są też podejścia, które jej nie wymagają[2]. Technika ta rozwiązuje wiele problemów wcześniej stosowanych kodowań m. in. rozmiar wektorów jest ustalony i mały niezależnie od rozmiaru korpusu, zachowujemy informację o kontekście i sieci neuronowe uczą się dużo szybciej korzystając z takiej reprezentacji. Dodatkowo można ją wykorzystać do badania poziomu bliskości słów i to z tej własności najbardziej będziemy korzystać.

Zanurzenia używane w pracy zostały wytrenowane na podstawie finalnych korpusów2.8. przy pomocy biblioteki FastText[5].

2.3.2. Lematyzacja

Lematyzacja jest procesem sprowadzania słów do ich formy podstawowej(słownikowej) np. 'jestem' do 'być'. W odróżnieniu od stemmingu – sprowadzania odmienionego

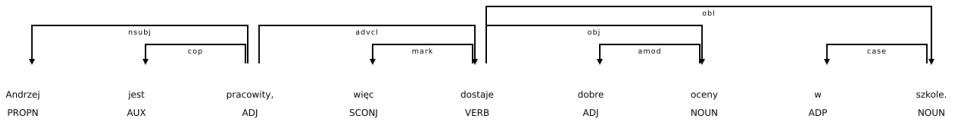
[1] Wydaje się, że NKJP, który był opracowany przez lingwistów jest bardziej wiarygodnym źródłem od korpusu POLEVAL, ale niestety udostępniane są jedynie statystyki n-gramowe, a nie pełen tekst

[2] https://spacy.io/

słowa do jego tematu – algorytm lematyzacji uwzględnia kontekst zdania, aby rozróżnić formy dwuznaczne np. słowo 'wina' może występować w zdaniach 'to moja wina' i 'dolej mi więcej wina' i powinno zostać zlematyzowane odpowiednio jako 'wina' i 'wino'.

2.3.3. Drzewo rozbiorów

W 2016r. zespół językoznawców stworzył standard ujednoliconych zależności gramatycznych [4]. Zależności te przyjmują formę relacji binarnej, w której argumentami są słowo główne i słowo zależne. Przetworzenie zdania zgodnie z relacjami w naturalny sposób transformuje je w drzewo zależności lub też inaczej – drzewo rozbiorów. Dla przykładu zdanie 'Andrzej jest pracowity, więc dostaje dobre oceny w szkole.' ma następujący rozbiór: Korzeniem drzewa jest 'pracowity', od którego



zależy podmiot (nsubj) 'Andrzej' i złożenie (advcl) 'dostaje'. Oba dalsze rzeczowniki – 'oceny' i 'szkole' – dają więcej informacji do słowa 'dostaje', z kolei 'dobre' modyfikuje, więc i zależy od słowa 'oceny'.

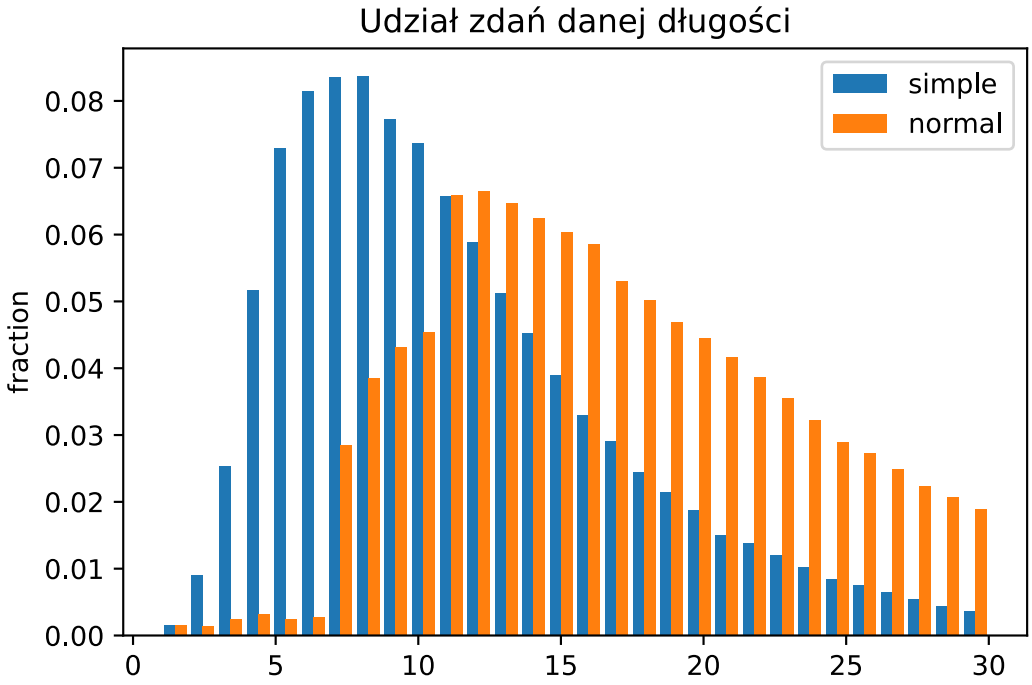
Jak widać poziom skomplikowania zdania znajduje swoje odzwierciedlenie w powstałym drzewie, co spróbujemy wykorzystać do uproszczenia tekstu. Do znalezienia rozbiorów wykorzystam bibliotekę spaCy.

2.3.4. Morfologia i tagowanie

Same słowa również niosą dużo informacji gramatycznych. W zależności od części mowy może to być fleksja, czas, rodzaj, liczba itd. Części mowy (czasem z pewnymi rozszerzeniami) nazywamy tagiem, natomiast pozostałe cechy morfologią danego słowa.

Informacje te są przydatne ponieważ na podstawie przewagi w liczbie wystąpień jednych form nad drugimi można określić powszechność danej formy. Dodatkowo można ich użyć również do odtworzenia poprawnej formy lematu otrzymanego przy zamianie słownictwa.

Do zdeterminowania morfologii i tagu danego słowa wykorzystam bibliotekę spaCy. Skorzystam też z korpusu mapującego[3] lematy do słowa w odpowiedniej formie gramatycznej, na podstawie morfologii.



Rysunek 2.1: histogram długości zdań

2.4. Podstawowe statystki korpusów

[3]

2.4.1. Długość słów i zdań

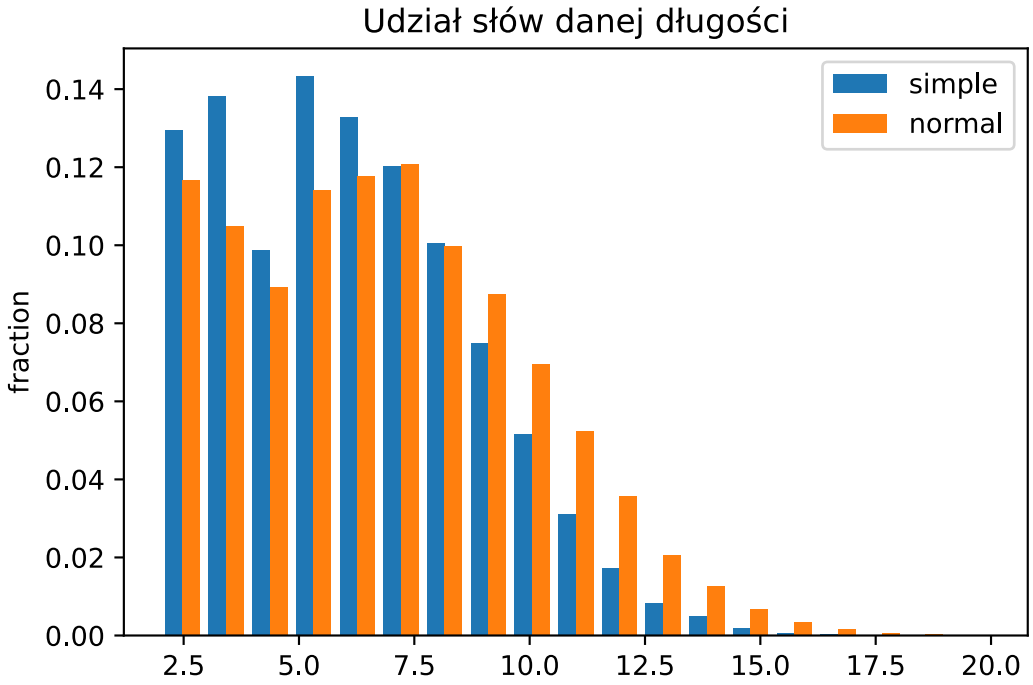
Badanie zdań zaczniemy od podstawowych statystyk odnośnie ich długości. Wydaje się, że zdania proste powinny być krótkie. Hipotezę tę potwierdzają wyniki pierwszych obliczeń, gdzie średnia długość jest o 77% większa wśród zdań normalnych.

Z wykresu2.1 wyraźnie widać, że wśród zdań krótkich dominują zdania proste, ponadto ich rozkład szybko zanika wraz ze wzrostem długości zdań, ustępując zdaniom normalnym.

Mniej wyraźną, ale również znaczącą różnicę możemy zaobserwować przy długości słów2.2. Tutaj średnia w normalnym języku jest o 15% większa.

Dokładniejsza inspekcja wykresu pokazuje przewagę słów do 6 liter w języku prostym i znaczną przewagę dłuższych – 9 i więcej – w języku normalnym. Badanie ilościowe pokazuje, że rzeczywiście udział słów o długości 9 i więcej jest o ponad

[3] kod źródłowy do poniższych eksperymentów dostępny jest na https://github.com/DorosMichal/SimpleLanguage



Rysunek 2.2: histogram długości słów

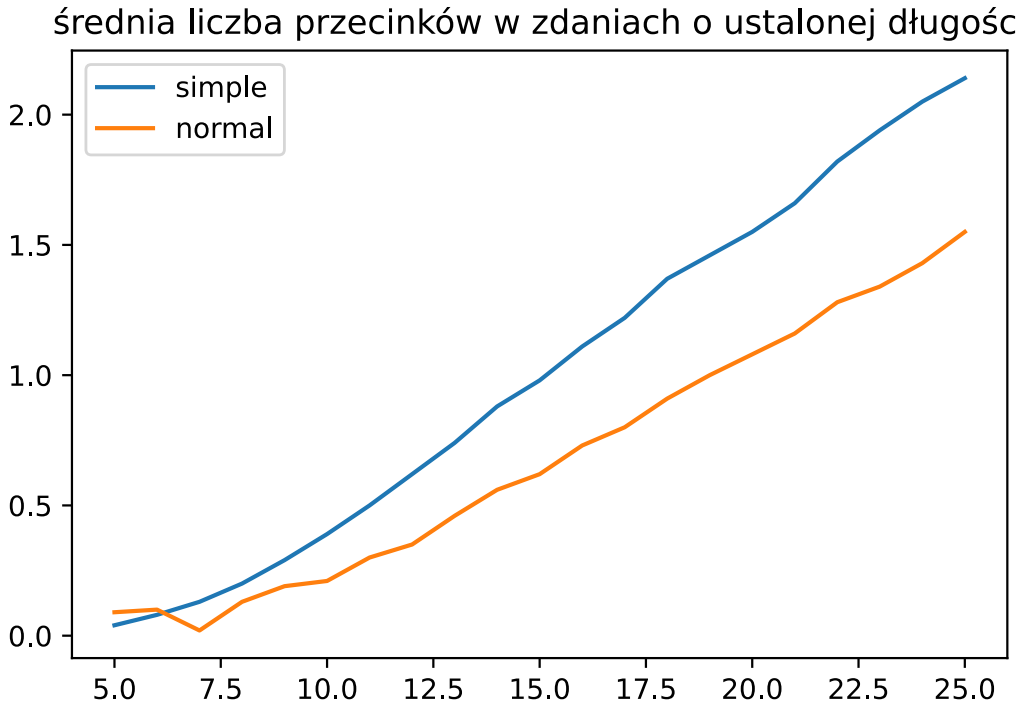
50% wyższy w zdaniach normalnych (18,5%) w porównaniu ze zdaniami prostymi (12%).

2.4.2. Interpunkcja

Następnym krokiem będzie sprawdzenie, jak często różne znaki interpunkcyjne występują w tekście w zależności od jego trudności. W tym celu porównamy ich średnią liczbę wystąpień w zdaniach tej samej długości. Okazuje się, że tylko przecinek i myślnik wykazują widoczne zależności.

Przeciwnie do intuicji autora, przecinków w zdaniach uproszczonych jest więcej2.3. Manualna inspekcja sugeruje, że spowodowane jest to częstym występowaniem zdań podrzędnie złożonych. Rzeczywiście ilościowa analiza drzew składniowych (2.3.3. pokazuje, że zależności związanych ze zdaniami podrzędnymi jest o ok. 10% więcej w korpusie uproszczonym.

Większa liczba myślników w zdaniach prostych prawdopodobnie wynika z tego, że teksty korpusów pochodzą ze znacząco różnych źródeł – teksty proste są literaturą, w której występuje dużo dialogów oraz przeniesień, które stanowią dwa główne zastosowania myślnika w języku polskim. Udział literatury w tekstach normalnych jest znacznie mniejszy.



Rysunek 2.3:

2.5. Statystyki występowania słów

2.5.1. Ranking na bazie korpusów języka prostego i normalnego

Kolejnym punktem badania będzie trudność słownictwa. W tym eksperymencie utożsamimy trudność słowa z częstością jego występowania - intuicyjnie ta prosta miara wydaje się dobrze oddawać zasób słownictwa jakim dysponują osoby uczące się obcego języka. W oparciu o używane korpusy sporządzony zostanie ranking słów w odpowiednich formach – niezmieniona, wszystko z małych liter, lemat, lemat z tagiem części mowy – i sprawdzone zostanie jaka część zdań daje się w pełni zrozumieć, kiedy znamy tylko najbardziej popularne N słów z rankingu.

Dla każdej formy liczba zrozumiałych zdań prostych jest znacząco wyższa (średnio 2,4 raza) dla małych N (3-10tys.) i wyrównuje się w okolicach 100 tysięcy znanych słów. Trend ten zachowuje się również, gdy ograniczymy się tylko do zdań, których długość liczona w słowach jest większa od 5 oraz od 10.

Następnym eksperymentem jest policzenie jaki procent słów występujących w zdaniu mieści się w słowach łatwych. Tutaj wyniki dla wszystkich form okazały się tylko nieznacząco lepsze od zdań normalnych, co sugeruje, że na bardzo wysoką przewagę w poprzednim eksperymencie wpływ miała długość zdań – rzeczywiście ograniczenie się do zdań tej samej długości zmniejsza przewagę zdań prostych z kilkukrotnej do zaledwie 10-40%.

2.5.2. Ranking na bazie korpusu NKJP

W powyższych eksperymentach rankingi słów brały się bezpośrednio z używanych korpusów co mogło mieć wpływ na wyniki. Zbadamy teraz jaki wpływ będzie miało użycie niezależnego korpusu NKJP. Warto jednak zauważyć, że prawdopodobnie korpus ten mógł w znacznym stopniu korzystać z łatwo dostępnych źródeł, takich jak transkrypcje obrad sejmu czy Wikipedia, podobnie jak korpus POLEVALa, co może faworyzować zdania normalne. Zgodnie z oczekiwaniami ogólne zrozumienie się zmniejszyło, tak samo, jak przewaga języka prostego, ale nadal dla słowników o rozmiarach od 3 do 10 tys. zrozumiałych zdań w języku prostym było więcej średnio 60%. Sytuacja ta zmienia się, kiedy w przypadku NKJP ograniczymy się do zdań tej samej długości – sprawia to, że zdania normalne są zrozumiałe nawet 2 razy częściej. Taki wynik może być spowodowany wspomnianym wyżej podobieństwem korpusów, jednak ustalenie tego wymagałoby dokładniejszej analizy.

2.6. Analiza gramatyczna

Kolejnym aspektem języka, który ma wpływ na jego postrzeganą trudność, jest jego budowa gramatyczna.

2.6.1. Części mowy

SpaCy oferuje podstawową i rozszerzoną klasyfikację części mowy. Dla obu sprawdzamy, które tagi charakteryzują się największą dysproporcją w częstości występowania między językami. Analiza ta pokazuje, znaczące dysproporcje w występowaniu konkretnych części mowy np. zaimki i czasowniki występują dwa razy częściej w języku prostym. Tagi rozszerzone pokazują za to, że rzeczowniki w dopełniaczu są 5 razy bardziej prawdopodobne w języku normalnym.

Tagi takie jak 'SYM' – symbole czy 'X' – znaki pochodzenia obcego lub 'BREV' – skróty, występują dużo częściej w języku normalnym niż uproszczonym, mimo że nie stanowią rzeczywistej trudności – zostaną więc pominięte w dalszej analizie.

2.6.2. Morfologia

Dokładniejsza analiza gramatyczna pozwala dla każdego słowa określić jego morfologię. Analogicznie do eksperymentów na słowach, można przeprowadzić testy, ile zdań będziemy rozumieli, znając N pierwszych 'kształtów' słów. Dla małych wartości – do 60 znanych 'kształtów' – język prosty osiąga ponad dwukrotną przewagę w liczbie zrozumiałych zdań.

2.6.3. Drzewa rozbioru

Drzewa rozbioru – zapewniając bardziej całościowe patrzenie na zdanie – mogą dostarczać wartościowych informacji o trudności zdania. W eksperymentach patrzyłem na cechy takie jak głębokość drzewa, szerokość przy korzeniu i średnią odległość w zdaniu między zależnymi słowami – przykładowo mając zdanie 'A B C D' i zależność A→D, ma ona odległość 3.

Ta ostatnia miara okazała się najbardziej informatywna uzyskując w zdaniach normalnych wynik o 30% wyższy. Pozostałe eksperymenty również uzyskiwały średnio wynik wyższy o ok. 20% dla języka normalnego.

2.7. Klasyfikator

2.7.1. Miary

W oparciu o powyższe eksperymenty ze zrozumieniem zdań, można zaproponować następujące miary pochodne, które mogą być bardziej użyteczne dla klasyfikatora prostych zdań niż bezpośrednie wyniki eksperymentów.

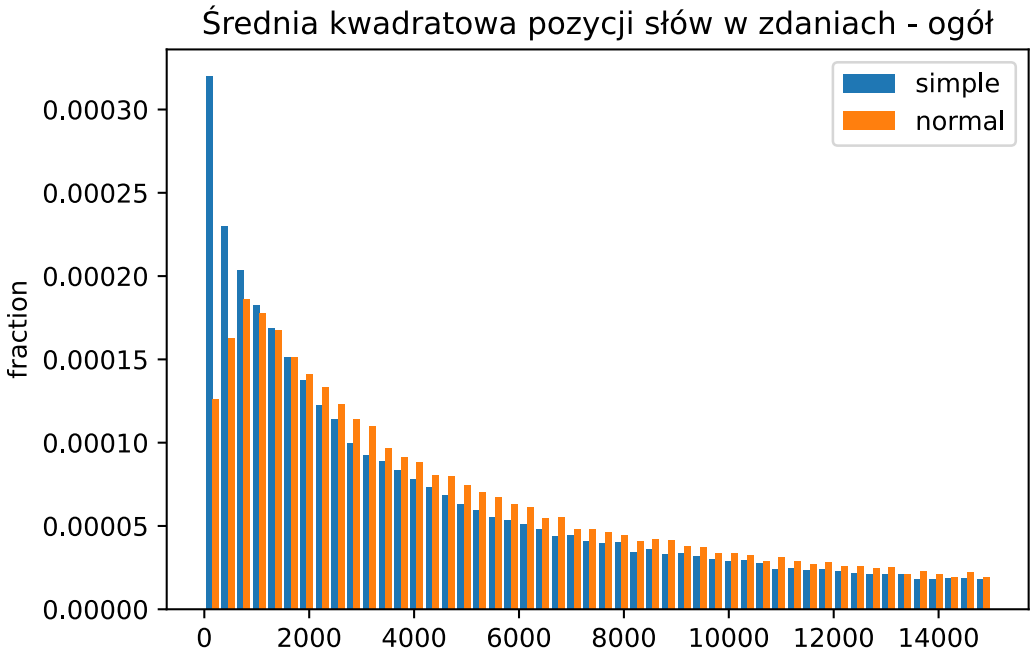
Zamiast zero-jedynkowo badać, czy zdanie jest zrozumiałe, co w znacznym stopniu zależało od jego długości, można dla każdego zdania policzyć średnią pozycję jego słów w rankingu. Okazuje się, że to podejście uwydatnia różnice między językiem prostym i standardowym – w szczególności pokazuje, że w języku prostym zdań bardzo prostych – uzyskujących średnią pozycję słowa na poziomie kilkuset – jest dwukrotnie więcej. Dodatkowo, można zwiększyć 'karę' za trudne słowa, korzystając ze średnich wyższego stopnia, co jeszcze bardziej uwydatnia różnice.

2.7.2. Wybór cech

Korzystając z informacji zebranych w trakcie analizy można dokonać wstępnego wyboru cech dla klasyfikatora, który będzie określać czy dane zdanie jest 'proste'. Zbierając wyniki wszystkich przeprowadzonych eksperymentów otrzymujemy 119 wartości (wiele z nich to zliczone i znormalizowane tagi lub zależności gramatyczne danego typu). Ten zestaw został dalej zredukowany do 20 najbardziej znaczących cech przy użyciu testu chi-kwadrat.

2.7.3. Modele i ewaluacja

Korzystając z wyłonionych cech oraz z biblioteki sklearn (konkretnie modeli LogisticRegression, GaussianNB, RandomForestClassifier) wytrenowane zostały 3 modele klasyfikujące – regresja logistyczna, naiwny klasyfikator bayesowski o rozkładzie gaussowskim oraz las losowy.



Rysunek 2.4: histogram średnich kwadratowych pozycji słów dla zdań w korpusach

	train	test
regresja	64.4	64.2
naive bayes	62.9	62.8
las losowy	92.8	92.0

Tabela 2.1: Wyniki klasyfikatorów

Regresja stanowi punkt odniesienia i osiągnęła wynik 64.2% na zbiorze testowym.

Klasyfikatora bayesowskiego uzyskał delikatnie gorsze wyniki na poziomie 62.9%, co jest zrozumiałe ponieważ wiele cech jest, choć częściowo, zależna od siebie, a ich rozkłady są skrzywione, tylko zbliżone do gaussowskiego.

Zdecydowanie najlepszy i w pełni zadowalający wynik 92% na zbiorze testowym uzyskał las losowy z głębokością drzew ograniczoną do 10 (bez tego ograniczenia las przetrenowywał się i osiągał wynik 99.9% na zbiorze treningowym).

2.8. Eksperymenty z innymi korpusami normalnymi

Podczas badań wyników klasyfikatora pojawiło się podejrzenie, że cechy, których nauczył się model, nie odpowiadają w głównej mierze za subiektywną trudność zdań, a bardziej za różnice w rodzajach tekstów użytych w korpusach. Aby przetestować tę hipotezę, został stworzony korpus składający się z około 400 tys. zdań

pochodzących z popularnych książek kierowanych głównie do dorosłych czytelników (autorów takich jak R. Mróz, S. King, R. R. Martin). Ku zaskoczeniu autora, po powtórzeniu powyższych eksperymentów dla takiego korpusu okazało się, że jest on ilościowo nierozróżnialny od korpusu prostego, który – przypomnę – został stworzony na bazie książek dla dzieci. Nasuwają się dwa możliwe wyjaśnienia zaistniałej sytuacji:

1. zaproponowane eksperymenty mogą nie spełniać swojego zadania i nie wychwytywać różnić w trudności zdań, lub
2. popularne książki są po prostu pisane łatwym, przystępnym dla wszystkich językiem, bardzo zbliżonym do języka używanego w książkach dla dzieci, a różnica wiekowa wynika jedynie z przekazywanej treści

Aby wykluczyć pierwszą możliwość stworzony został jeszcze jeden korpus, tym razem składający się z 50 tys. zdań pochodzących z książek uznawanych za literaturę ambitną (autorów tj. Tokarczuk, Gombrowicz, Reymont, Kapuściński). W tym wypadku widoczna była znacząca różnica we wszystkich eksperymentach, co sugeruje, że 2. wyjaśnienie jest bardziej prawdopodobnym wytłumaczeniem zaistniałej sytuacji.

Biorąc to pod uwagę, wydaje się, że możliwe były dwa dalsze kierunki:

- kontynuowanie z przyjętym korpusem i próba stworzenia rozwiązania zbliżającego dowolny tekst do lekkiej prozy albo
- przyjęcie nowej definicji języka prostego i trudnego.

Po konsultacji z promotorem w dalszej części pracy przyjęty został 1. scenariusz m.in. z powodu znaczących trudności natury ekonomicznej pozyskiwania dużego korpusu mało popularnych i trudno dostępnych tekstów literatury pięknej.

Wytrenowany klasyfikator został użyty do przetworzenia całego korpusu POLEVAL i stworzenia finalnych zbiorów języka prostego i normalnego o liczebności odpowiednio 4.8 i 18.2 miliona zdań.

Rozdział 3.

Heurystyczne systemy upraszczające

3.1. Wstęp teoretyczny

W tej części pracy podjęta została próba wykorzystania klasycznych metod analizy języka do stworzenia prostych mechanizmów upraszczających. Uzyskane wyniki pozwalają określić punkt odniesienia dla bardziej zaawansowanych metod, a także dostarczają narzędzi upraszczających, których wytrenowanie i używanie, w porównaniu z metodami neuronowymi, jest tanie obliczeniowo.

3.2. Modele upraszczające i wyniki ich działania

Najprostszy, bazowy model polega na upraszczaniu słownictwa jeden do jednego bazując na wytrenowanych zanurzeniach. Ustalamy słowa proste jako N najbardziej popularnych słów w korpusie prostym i podmieniamy każde słowo spoza tego zbioru na słowo proste o najbliższym zanurzeniu. Podejście to ma trzy oczywiste wady:

1. rozważa konkretne formy gramatyczne zamiast lematów – słowa, których rzadsze formy nie znalazły się w zbiorze słów prostych są podmieniane na swoje bardziej popularne lecz niepoprawne formy gramatyczne

2. dokonując zamiany na synonim nie dba o poprawność gramatyczną podstawianych słów – można by mieć nadzieję, że najbliższa forma gramatyczna prostszego słowa będzie taka sama, jednak eksperymenty wskazują, że jest to przypadkowe i nie należy na tym polegać.

3. potencjalnie skomplikowana struktura zdania pozostaje niezmieniona

Następne podejścia spróbują odnieść się do tych problemów. Pomysł na pierwsze dwa jest następujący – możemy zlematyzować cały korpus i wytrenować nowe zanurzenia znające tylko lematy. Następnie upraszczać zdanie w następujący sposób:

1. badamy i zapisujemy morfologię oryginalnego zdania
2. lematyzujemy zdanie i dokonujemy podstawień lematów na prostsze odpowiedniki
3. szukamy właściwej formy dla odpowiednika porównując stopień zgodności znaczników morfologicznych dostępnych form z morfologią oryginału

Inspekcja tak tłumaczonych zdań wskazuje, że zaproponowane podejście znacząco poprawia poprawność gramatyczną zdań jednak sam dobór słownictwa nie zawsze odpowiada oczekiwanemu – zgodnie z naturą zanurzeń pojawiają się słowa, często występujące w tych samych kontekstach jednak mające zupełnie inne znaczenia np. 'gwałt' jest tłumaczony na 'samobójstwo', 'kamienny' na 'drewniany' itp., dodatkowo słowa czasem są tłumaczone na zaprzeczenia np. 'moralny' na 'niemoralny'. Tłumaczenie pojedynczych słów traci też kontekst, przez co jest narażone na zamianę 'z art. 33 ust' na 'z art.33 twarzy'. Możliwymi rozwiązaniami do dalszego sprawdzenia jest zastosowanie heurystyk zapobiegających takim zamianom, uwzględnienie kontekstu poprzez dodanie wektorów słów okolicznych oraz wspomaganie się słownikiem synonimów. Dodatkowo można też sprawdzić wyniki w zależności od sposobu definiowania słów prostych – zaskakująco przy aktualnej metodzie w zbiorze słów prostych znalazło się np. słowo 'wcześniejszy', ale już nie 'wcześniej', które jest tłumaczone na 'poprzednio' – zjawisku temu po części winna jest niedoskonała lematyzacja.

Do próby rozwiązania problemu 3. użyte zostały drzewa rozbiorów – opierając się na intuicji, że im głębsze drzewo tym bardziej skomplikowane zdanie, pierwszą wypróbowaną heurystyką było obcinanie drzewa do ustalonej lecz potencjalnie zależnej od zdania głębokości. To podejście nie jest jednak dobre ponieważ:

1. nie dbamy o poprawność gramatyczną otrzymanego obcięcia
2. w zdaniach wielokrotnie złożonych nadmiernie faworyzuje ono pierwsze zdania proste zachowując wszystkie szczegóły, a usuwając ważne części zdań dalszych. Efektywnie degeneruje się to do brania prefiksu zdania.

Analizując rozkłady można zauważyć, że niektóre relacje takie jak 'fixed' – ustalone związki wyrazowe, 'expl:pv' – zaimki zwrotne, 'case' – przyimki, a także niektóre rodzaje spójników czy relacja oznaczająca podmiot są bardzo wrażliwe na usuwanie i w większości przypadków, po takim zabiegu zdanie przestaje być gramatycznie poprawne. Można więc ręcznie wybrać zbiór relacji, których nie obcinamy, nawet po przekroczeniu dozwolonej głębokości. Zastosowanie tego podejścia sprawia, że wynikowe zdania są w większości poprawne gramatycznie.

Universal Dependency[4] wyróżnia niektóre relacje jako odpowiadające za złożenia współrzędne i podrzędne. Intuicyjnie, zdania współrzędne powinny być analizowane na podobną głębokość niezależnie od tego w jakiej kolejności występują w zdaniu, natomiast zdania podrzędne wydają się mniej ważne. Zamiast dopuszczać stosunkowo dużą głębokość od początku, spróbujemy modyfikować ją tak aby algorytm odpowiadał powyższej intuicji. To podejście daje bardzo zadowalające efekty np.

> Przypomnę tutaj Wysokiej Izbie, że właśnie to stosunkowo niewielkie biuro – zaledwie w ciągu 12 miesięcy – było w stanie zrealizować ogromny ogólnopolski program badawczy związany z 20. rocznicą wprowadzenia stanu wojennego.

Jest upraszczane do

> Przypomnę tutaj Wysokiej Izbie, że to niewielkie biuro w ciągu 12 miesięcy – było w stanie zrealizować program związany z 20 rocznicą wprowadzenia stanu.

Jedyna usterka powyższego zdania to obcięty związek wyrazowy 'stan wojenny'. Jednak takie zachowanie zostało spowodowane błędnym zakwalifikowaniem tej zależności przez parser, jako zwykły przymiotnik. Aby jeszcze poprawić wyniki można próbować zastosować heurystyki typu:

- jeśli trafiamy na datę to wypisujemy całą,
- zostawiamy następujące po sobie słowa pisane wielkimi literami (zazwyczaj nazwy własne),
- jeśli trafiamy na wartość numeryczną to zostawiamy ją (numer, kwota, inny rodzaj danych)

Ostatecznie można połączyć obie formy upraszczania otrzymując następujące efekty:

> Dlatego uważam, że właśnie w Komisji Edukacji, Nauki i Młodzieży, a może bardziej w podkomisji młodzieży, powinniśmy na bieżąco analizować wydawanie publicznych pieniędzy, powinniśmy analizować merytoryczne programy.

> Dlatego uznam, że w Komisji, a może bardziej w komisji młodzieży powinniśmy na bieżąco oceniać wydawanie pieniędzy, powinniśmy oceniać programy.

uproszczenie	średni wynik trudności zdania
oryginał	0.706
struktura - ustalona głębokość	0.647
struktura - głębokość zależna od długości zdania	0.619
struktura - głębokość postępująca	0.674
zmiana słownictwa	0.569
finalny model	0.517

Tabela 3.1: Wyniki metod upraszczających względem przyjętej miary prostoty

Kolejny przykład pokazuje oryginalne zdanie, zdanie uproszczone zaproponowanym systemem i zdanie poprawione powyższymi heurystykami:

> W odpowiedzi na interpelację nr 3262 z dnia 5 lutego 2000 r. posła na Sejm RP pana Ryszarda Brejzy w sprawie powstania w Polsce warunków do monopolizacji rynku materiałów termoizolacyjnych w wyniku wejścia w życie rozporządzenia ministra spraw wewnętrznych i administracji z dnia 22 września 1999 r., zmieniającego rozporządzenie w sprawie szczegółowego zakresu i formy audytu energetycznego oraz algorytmu oceny opłacalności przedsięwzięcia termomodernizacyjnego, a także wzorów kart audytu energetycznego (DzU nr 79, poz. 900), uprzejmie przedstawiam następujące wyjaśnienia.

> W odpowiedzi na zapytanie nr z dnia posła na Sejm pana Ryszarda w sprawie powstania warunków do monopolizacji rynku materiałów, uprzejmie przedstawiam następujące wyjaśnienia.

> W odpowiedzi na zapytanie nr 3262 z dnia 5 lutego 2000 r. posła na Sejm RP pana Ryszarda Brejzy w sprawie powstania warunków do monopolizacji rynku materiałów, uprzejmie przedstawiam następujące wyjaśnienia.

Po zastosowaniu tych uproszczeń do losowej próbki 300 zdań z korpusu normalnego okazało się, że tylko około 30% ulega uproszczeniu. Poniżej przedstawiam średni poziom uproszczenia na podstawie wyników zwracanych przez klasyfikator prostoty (im większy wynik tym zdanie trudniejsze).

Rozdział 4.

Neuronowe systemy upraszczające

Poprzednie podejścia są ograniczone do podmieniania i usuwania tekstu. Heurystyki próbujące upraszczać język w bardziej wyrafinowany sposób np. poprzez zamianę skomplikowanych struktur gramatycznych na ich prostsze, lecz znacząco inne odpowiedniki, takie jak zamiana strony biernej na aktywną lub rozbicie skomplikowanych zdań wielokrotnie złożonych na kilka prostszych, szybko robią się bardzo skomplikowane, szczególnie w językach o luźnym szyku takich jak polski. Narzędziem, które może sprostać takiemu poziomowi złożoności są sieci neuronowe.

4.1. Tłumaczenie maszynowe

Jednym z naturalnych sposobów myślenia o zadaniu upraszczania tekstu jest zaanonsowane w poprzednich rozdziałach, dobrze zbadane, zadanie tłumaczenia maszynowego, w którym językiem źródłowym jest język polski normalny, a docelowym język polski prosty.

Celem każdego neuronowego modelu tłumaczącego(NMT) jest przyjęcie zdania w języku źródłowym jako wejścia i zwrócenie tego zdania przetłumaczonego na inny język jako wyjścia. Aby tego dokonać trzeba umieć zamienić zdanie na abstrakcyjną reprezentację, która może posłużyć jako wejście do sieci, a później zmapować tę sekwencję na odpowiednie tłumaczenie. Do pierwszej części bardzo dobrze nadają się omawiane wcześniej zanurzenia słów. Główny problem z drugą polega na tym, że odpowiednie sekwencje bardzo rzadko są tych samych długości np. poprawnym tłumaczeniem 'Lubię jeździć na rowerze' na angielski jest 'I like cycling' – trzeba więc odpowiednio zmotywować sieć aby nie tłumaczyła słów bezpośrednio na ich angielskie odpowiedniki produkując 'Like riding on bike'.

4.1.1. Architektura koder-dekoder

Aktualnie najlepiej radzącymi sobie, z tego typu zadaniem, modelami są te oparte o architekturę koder-dekoder. Pomysł polega na tym, aby stworzyć pewną abstrakcyjną, stosunkowo małą reprezentację znaczenia danego zdania. Reprezentacja ta, zazwyczaj w postaci n-wymiarowego wektora rzeczywistego, jest jedynym środkiem komunikacji między koderem i dekoderem. Zmusza to model to bardziej holistycznego patrzenia na tłumaczone zdanie oraz zachęca do używania naturalnych dla danego języka konstrukcji wyrażających dane znaczenie.

Mapowanie seq2seq

Żeby sieć była w stanie dobrze tłumaczyć zdania, musi uwzględniać kontekst. Innymi słowy, wczytując n-ty token w zdaniu musi pamiętać co przeczytała wcześniej. Takie zachowanie próbują symulować rekurencyjne sieci neuronowe (RNN). Sieć taka jako swoje wejście w momencie n przyjmuje n-ty token oraz swoje wyjście (zwane też ukrytym stanem) z momentu n-1, zachowując w ten sposób jakąś informację z poprzednich tokenów. Specjalnym typem RNN współcześnie używanym w modelach tłumaczących są sieci typu LSTM[15] (long-short term memory) oraz GRU[16] (gated recurrent unit) – obie rozbudowują powyższą ideę o jeszcze dodatkową komórkę, która w zamyśle ma służyć jako pamięć długotrwała. W każdej iteracji usuwane są z niej rzeczy, które sieć uzna już za niepotrzebne, a dodawane są informacje pozyskane z aktualnego wejścia, które mogą się kiedyś przydać.

Mimo przekazywanej pamięci w praktyce przy dłuższych zdaniach sieci te zapominają niektóre informacje. Dodatkowo nie wszystkie informacje zmieszczą się w komórce pamięci - wektorze o stałej długości. Rozwiązaniem tych problemów jest mechanizm uwagi. Idea jest taka, że zamiast przesyłać zawartość pamięci przez kolejne komórki, możemy popatrzeć na raz na wszystkie ukryte stany w całej sekwencji i na ich podstawie stworzyć wektor kontekstowy.

4.1.2. Zastosowanie obecnych modeli do upraszczania

Na dzień powstania pracy nie udało mi się znaleźć żadnych gotowych modeli wykonujących takie tłumaczenia dla języka polskiego. Zanim spróbowałem więc taki stworzyć postanowiłem przetestować dwa inne podejścia.

Identyczność jako metoda upraszczania

Tłumaczenie zdań na inny język jest procesem stratnym. Pojawia się przypuszczenie, że podwójne tłumaczenie z polskiego na angielski i z powrotem ma szansę

zamienić bardziej skomplikowane słowa i zwroty na ich prostsze odpowiedniki. Przeprowadzony eksperyment polegał wykorzystaniu komercyjnego modelu tłumaczącego DeepL do przetłumaczenia losowej próbki 590 zdań z korpusu języka polskiego normalnego. Tylko dla 37 zdań przetłumaczona wersja jest identyczna z oryginałem. Ok. 25% zostało zmienionych w bardzo niewielkim stopniu osiągając BLEU[1] na poziomie powyżej 0.7 (np. zmiana jednego słowa na inną formę). W pozostałych przypadkach doszło do zmian, z których znaczna część jest pewną formą uproszczenia.

> Nakłady finansowe związane ze sportem w znacznej części zaczęły przekraczać nasz budżet domowy i zostałam zmuszona do złożenia pozwu o podwyższenie alimentów

> Nakłady finansowe związane ze sportem zaczęły w dużym stopniu przekraczać nasz budżet domowy i byłam zmuszona złożyć pozew o podwyższenie alimentów.

Prezentacja możliwości na przykładzie DeepL + GPT3

Korzystając z tego, że najlepsze modele tłumaczące są w stanie przetłumaczyć zdanie w tą i z powrotem zachowując sens i poprawność, można spróbować wykorzystać do upraszczania zaawansowane modele językowe języka angielskiego umiejscawiając je w środku tego procesu. W tym przykładzie wykorzystam GPT3[13] – jeden z czołowych modeli językowych. Korzystając z tego samego zdania co poprzednio, wpisując do GPT3 'write simplified version of this sentence', a następnie tłumacząc wynik przy pomocy DeepL, otrzymujemy

> Nakłady finansowe na sport zaczęły przekraczać nasz budżet domowy, więc złożyłem pozew o podwyższenie alimentów.

Jak widzimy efekty mogą być bardzo dobre (nie tylko subiektywnie – wartość metryki prostoty poprawiła się o 15%), jednak w procesie przejścia między językami tracimy np. typową dla polskiego odmianę czasownika przez rodzaje – tutaj

'zmuszona złożyć pozew' ⇾ 'forced to sue' ⇾ 'I sued' ⇾ 'złożyłem pozew'

Próbując osiągnąć w pełni zadowalające wyniki dla języka polskiego powinniśmy więc spróbować stworzyć dedykowany model.

[1] Jedną z przyjętych metryk jakości tłumaczenia jest BLEU (Bi-Lingual Evaluation Understudy)[6]. Porównuje ona zdanie przetłumaczone maszynowo z istniejącymi tłumaczeniami ludzkimi. Jest oparta na precyzji – dokładniej patrzy jaki procent n-gramów w zdaniu przetłumaczonym maszynowo występuje w zdaniach referencyjnych. Przyjmuje więc wartości od 0 do 1.

4.1.3. Duże korpusy równoległe

Główny problem z powyższą architekturą użytą w obu tych modelach, polega na tym, że jest ona duża, co przekłada się na znaczące koszty trenowania (szacowane w milionach dolarów dla wykorzystanych wyżej modeli), a także potrzebę posiadania ogromnych korpusów. W przypadku modeli tłumaczących, klasyczne trenowanie odbywa się na podstawie par (zdanie, ludzkie tłumaczenie). Dla różnych języków takie korpusy powstają w sposób naturalny np. przy okazji przekładu dokumentów czy książek. Jeśli jednak chodzi o upraszczanie zdań, ilość dostępnych danych jest bardzo ograniczona. W przypadku języka angielskiego największym dostępnym korpusem jest Simple Wikipedia składająca się z 218 000 artykułów pisanych prostym językiem (Nie ma jednak gwarancji, że artykuł ten będzie równoległym uproszczeniem oryginalnego artykułu). Niestety w przypadku języka polskiego nie udało mi się znaleźć żadnych takich zbiorów. Jednym z możliwych pomysłów na pozyskanie takich danych byłoby przetłumaczenie korpusów angielskich korzystając z tłumaczenia maszynowego, jednak ze względu na rozmiar wiązałoby się to ze znacznymi kosztami. Dodatkowo nie ma gwarancji, że tak pozyskany korpus nie będzie generował tych samych problemów co wyżej.

4.1.4. Tłumaczenie maszynowe bez nadzoru

Alternatywnym podejściem jest tłumaczenie bez nadzoru. Pomysł wykorzystuje architekturę koder-dekoder, więc znowu będziemy tworzyć wspólną przestrzeń znaczeniową, ale tłumaczenie będzie odbywało się w dwie strony (i w zależności od implementacji będziemy mieli parę koder i dekoder dla każdego języka lub wspólny koder i dwa dekodery). Jednak teraz nie dysponujemy równoległymi danymi – do utworzenia wspólnej reprezentacji można wykorzystać następujące 4 kroki:

1. wyrównujemy zanurzenia – zamiast mieć dwie osobne przestrzenie zanurzeń słów, mapujemy je we wspólną przestrzeń, tak żeby swoje tłumaczenia były blisko siebie – można to zrobić używając tylko jednojęzycznych korpusów (czyli bez słowników wielojęzycznych), co opisuje praca [10]

2. uczymy model w podobny sposób jak autokoder odszumiający – chcemy, aby zakodował specjalnie uszkodzone zdanie (poprzez usunięcie lub zamianę słów) i odkodował je w poprawnej wersji (wszystko w obrębie tego samego języka)

3. tłumaczymy zdanie s w języku źródłowym na zdanie t w języku docelowym używając naszego modelu (w pierwszej iteracji można zacząć od naiwnego tłumaczenia słowo na słowo), następnie dodajemy szum do zdania t i próbujemy je przetłumaczyć z powrotem tak, aby dostać zdanie wejściowe s.

4. stosujemy dodatkową sieć zwaną dyskryminatorem, która próbuje odróżnić kodowanie z języka źródłowego i kodowanie z języka docelowego. Chcemy zapewnić, żeby kodowania miały takie same rozkłady (intuicyjnie wyglądały tak

samo niezależnie od języka) więc trenując model dążymy do tego, żeby oszukać dyskryminator.

Podejście to zostało zaprezentowane na konferencji ICLR 2018[8] i zgodnie z pracą zapewnia rezultaty porównywalne do modeli nadzorowanych trenowanych na korpusach 100tys. równoległych zdań (wymaga jednak niesparowanych korpusów o rozmiarze rzędu 10milionów zdań). W tej pracy podjąłem próbę zreprodukowania wyników korzystając z innej pracy z tej samej konferencji[**?**], która korzysta z modelu o mniejszej złożoności oraz pomija ostatni krok, lecz również daje lepsze niż punkt odniesienia rezultaty.

Próba nauczenia

W przypadku tego samego języka wyrównanie słownictwa jest wyjątkowo łatwe. Korzystając z korpusów przygotowanych w 2. rozdziale wytrenowałem zanurzenia dla wszystkich zdań z obu korpusów, a następnie rozdzieliłem je dla słów występujących tylko w jednym z nich. Dodatkowo ograniczyłem rozmiar słownictwa do 30tys. dla każdego z języków – w języku prostym poprzez wybranie najbardziej popularnych słów, w języku normalnym poprzez wzięcie słów występujących w jednostajnie wylosowanych zdaniach.

Korzystając z kodu dostarczonego przez autorów wytrenowałem dwa modele, pierwszy 3-krotnie zmniejszony do rozmiarów, które pozwalały na trenowanie go przy ograniczonym dostępie do zasobów obliczeniowych. Ponieważ tak stworzony model tłumaczył wszystkie zdania wejściowe na ustalone, przypadkowe słowo powtórzone kilka razy, podjąłem drugą próbę trenując model w konfiguracji i z parametrami zalecanymi przez autora. Niestety przy drugim podejściu efekt był taki sam.

Nie byłem w stanie sprawdzić czy kod dostarczony przez autora pracy rzeczywiście działa poprawnie na oryginalnym zadaniu, toteż kwestia tego czy zaproponowane podejście działa w przypadku upraszczania języka pozostaje otwarte do dalszych badań.

Rozdział 5.

Rezultaty i dalsze prace

Praca spełniła swoje założenia definiując miarę prostoty, a następnie proponując metody upraszczania, które poprawiały wynik tej miary na testowanych zdaniach. W trakcie pisania pracy pojawiło się jednak sporo nowych pytań i pomysłów do dalszego zbadania:

1. Praca ta określiła język prosty jako język książek dla dzieci w wieku do 12 lat – jak się jednak okazało2.8. być może nie jest to wystarczający poziom prostoty – dysponując większymi zasobami można spróbować stworzyć wystarczająco duży korpus na zamówienie lub zgromadzić książki dla młodszych dzieci, które niestety trudno jest pozyskać w dużej liczbie za darmo.

2. Rezultaty uzyskane zaproponowanymi modelami być może można poprawić aplikując do nich bardziej granularne heurystyki3.2..

3. korzystając z przygotowanego korpusu i wytrenowanych zanurzeń można próbować powtórzyć wyniki innych badaczy dotyczące tłumaczenia bez nadzoru.

Bibliografia

[1] Tomas Mikolov et al (2013) *Distributed Representations of Words and Phrases and their Compositionality* https://arxiv.org/pdf/1310.4546.pdf

[2] Jeffrey Pennington el al(2014) GloVe: Global Vectors for Word Representation https://nlp.stanford.edu/pubs/glove.pdf

[3] Miłkowski M. (2010). Developing an open-source, rule-based proofreading tool.

*[4] Nivre, J., M.-C. de Marneffe, F. Ginter, Y. Goldberg, J. Hajic, C. D. Manning, R. McDonald, S. Petrov, S. Pyysalo, N. Silveira, R. Tsarfaty, and D. Zeman. (2016) Universal Dependencies v1: A Multilingual Treebank Collection*https://aclanthology.org/L16-1262/

[5] FastText documentation https://radimrehurek.com/gensim/models/fasttext.html

[6] Kishore Papineni, Salim Roukos, Todd Ward, Wei-Jing Zhu (2002) Bleu: a Method for Automatic Evaluation of Machine Translation https://aclanthology.org/P02-1040/

[7] Mikel Artetxe, Gorka Labaka Eneko Agirre (2018) UNSUPERVISED NEURAL MACHINE TRANSLATION https://arxiv.org/pdf/1710.11041.pdf

[8] Guillaume Lample et al (2018) UNSUPERVISED MACHINE TRANSLATION USING MONOLINGUAL CORPORA ONLY https://arxiv.org/pdf/1711.00043.pdf

[9] Ian Goodfellow. Nips (2016) Generative adversarial networks.

[10] Mikel Artetxe, Gorka Labaka, and Eneko Agirre (2018) A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings https://aclanthology.org/P18-1073/

[11] http://2018.poleval.pl/index.php/tasks#task3

[12] http://zil.ipipan.waw.pl/NKJPNGrams

[13] Tom B. Brown et al (2020) Language Models are Few-Shot Learners https://arxiv.org/abs/2005.14165

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin (2017) Attention Is All You Need `https://arxiv.org/abs/1706.03762?context=cs`

[15] Sepp Hochreiter; Jürgen Schmidhuber (1995) Long Short Term Memory

[16] Cho, Kyunghyun; van Merrienboer, Bart; Bahdanau, DZmitry; Bengio, Yoshua (2014) On the Properties of Neural Machine Translation: Encoder-Decoder Approaches