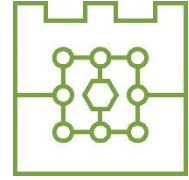




**Politechnika Krakowska
im. Tadeusza Kościuszki**

Wydział Informatyki i Telekomunikacji



Detektor sarkazmu

Karol Kusper

Jakub Kubica

Szymon Jagła

Kraków, 2024

Spis treści

1. Abstrakt	2
2. Wstęp	2
2.1 Cel	2
2.2 Zakres	2
2.3 Metodyka	3
3. Teoria.....	Błąd! Nie zdefiniowano zakładki.
4. Podsumowanie	7
5. Bibliografia.....	8

1. Abstrakt

Sarkazm stanowi jedno z najtrudniejszych zagadnień w przetwarzaniu języka naturalnego (NLP), ponieważ jest subtelną formą wyrażania intencji, w której dosłowne znaczenie wypowiedzi jest sprzeczne z zamierzonym przekazem. Praca ta stanowi przegląd możliwych podejść, które pozwolą stworzyć mniej lub bardziej wydajny detektor sarkazmu.

2. Wstęp

2.1 Cel

Celem tego projektu jest teoretyczne rozeznanie w metodach wykrywania sarkazmu w tekstach, uwzględniając podejścia klasyczne oraz nowoczesne.

2.2 Zakres

W naszym zakresie pracy znalazło się gromadzenie artykułów naukowych, które dotyczą sarkazmu oraz ich analiza. Na platformie Github odbyły się poszukiwania repozytoriów związanych z sarkazmem. Podjęty zostały także próby uruchomienia

projektów w środowisku lokalnym. Wykonana praca zostanie przedstawiona w przystępnej dla oka odbiorcy prezentacji.

2.3 Metodyka

Dołożono wszelkich starań, aby znalezione artykuły nie były starsze niż 5 lat. To pozwoli na przedstawienie w miarę nowoczesnego podejścia.

3. Teoria

3.1 Definicja sarkazmu

Sarkazm to wyrażenie językowe, które często służy do komunikowania przeciwnych intencji w stosunku do tego, co się mówi, zazwyczaj mając na celu wyrażenie czegoś bardzo nieprzyjemnego z zamiarem obrażenia lub wyśmiania [1]. Wrodzona niejednoznaczność wyrażen sarkastycznych sprawia, że ich wykrywanie jest bardzo trudne [1]. Sarkazm to retoryczny sposób wyrażania niechęci lub negatywnych emocji za pomocą różnych konstrukcji językowych, takich jak wyolbrzymienie lub wyśmiewanie [1]. Jest to mieszanka drwiny i fałszywej uprzejmości, mająca na celu nasilenie wrogości, bez jej bezpośredniego wyrażenia [1]. Sarkazm nie zawsze objawia się w ten sam sposób, co sprawia, że nawet ludzie nie zawsze go rozumieją.

3.2 Wykrywanie sarkazmu bez komputera

Jeśli mówimy o wykrywaniu sarkazmu bez użycia komputerów, to może ono polegać na analizie ludzkiej intuicji i interpretacji, a także na kontekście sytuacyjnym lub społecznym, w którym wypowiedź jest użyta. W takim przypadku wykrywanie sarkazmu często opiera się na tonie głosu, zrozumieniu kontekstu kulturowego, społecznego lub sytuacyjnego, cechach charakterystycznych dla wypowiedzi sarkastycznych, takich jak przesada, ironia czy paradoks.

3.3 Wykrywanie sarkazmu z pomocą technologii

Chociaż pewne sygnały, takie jak użycie wykrzykników, przesadnych określeń czy emotikonów, mogą sugerować sarkastyczny ton, nie zawsze są one obecne.

Wczesne metody wykrywania sarkazmu opierały się głównie na z góry ustalonych wzorcach lub regułach, takich jak określone zasady gramatyczne, etykiety czy reguły metaforyczne. Na przykład w tweetach etykiety emocjonalne były wykorzystywane do analizy, czy emocje w reszcie tekstu są z nimi sprzeczne .

W metodach statystycznych mierzono semantyczne korelacje między słowami. Badano, jak bardzo znaczenia słów są ze sobą powiązane, wykorzystując podobieństwo w WordNet. Jest to baza danych językowych, gdzie słowa są powiązane relacjami, takimi jak synonimy, antonimy czy hierarchie pojęciowe (np. pies jest ssakiem) aby wydobywać cechy. Następnie zaczęto stosować tradycyjne metody uczenia maszynowego, takie jak SVM (maszyny wektorów nośnych), drzewa decyzyjne, czy naiwny klasyfikator Bayes'a, który może analizować częstotliwość słów lub fraz w sarkastycznych i niesarkastycznych tekstach, aby obliczyć prawdopodobieństwo, że dane zdanie jest sarkastyczne.

Aby skuteczniej wydobywać cechy sarkastycznych tekstów i poprawić dokładność detekcji, wiele badań zaczęło łączyć różne techniki głębokiego uczenia, takie jak: konwolucyjne sieci neuronowe (CNN), rekurencyjne sieci neuronowe (RNN), mechanizmy uwagi [3].

Wprowadzenie pretrenowanych modeli BERT oraz GPT umożliwiło badaczom poszerzenie zastosowań o analizę emocji i kontekstu, co jeszcze bardziej rozwinęło dziedzinę. Następnie, w zadaniach klasyfikacji tekstów, szeroko stosowano grafowe sieci konwolucyjne (GCN). Na przykład, pewne badanie reprezentowało relacje emocjonalne i składniowe w tekście jako grafy, łącząc w ten sposób informacje z całego tekstu. Aby uniknąć szumów wynikających z ręcznego tworzenia grafów statycznych, zaproponowano metodę iteracyjnego uczenia się grafów niespójności.

3.4 Proponowane podejścia

W materiale [1] proponowane jest następujące podejście:

1. wstępne przetwarzanie danych, czyli konwersji tekstu wejściowego na numeryczną (wektorową) reprezentację, które są wymagane do trenowania modelu głębokiego uczenia np przy użyciu modelu BERT lub GPT.
2. Użycie mechanizmu Multi-Head Self-Attention, dzięki któremu model potrafi równocześnie analizować różne zależności w zdaniu, np. negatywny ton i przesadę (oba typowe dla sarkazmu).
3. bramkowane jednostki rekurencyjne (GRU), klasyfikacja i interpretowalność modelu. GRU pomaga w nauce relacji długozasięgowych między tymi wyróżnionymi słowami i generuje

pojedynczy wektor cech, który koduje całą sekwencję. Na koniec, warstwa w pełni połączona z aktywacją sigmoidalną jest używana do uzyskania końcowego wyniku klasyfikacji.

W materiale [2] wykorzystano różne algorytmy uczenia maszynowego, w tym naiwny klasyfikator Bayes'a, maszyny wektorów nośnych, lasy losowe, rekurencyjne sieci neuronowe oraz konwolucyjne sieci neuronowe.

Zestaw danych z Reddita został załadowany z platformy Kaggle. Zdania zostały tokenizowane za pomocą funkcji `word_tokenize` z Natural Language Toolkit (NLTK). Tokenizowane słowa zostały lematyzowane za pomocą klasy `WordNetLemmatizer` z NLTK. Lematyzacja odnosi się do użycia słownika i analizy morfologicznej słów, mającej na celu usunięcie tylko końcówek fleksyjnych i przywrócenie podstawowej formy słowa

Trick or treating, in general, is just weird [...]", został przekształcony na „tricktreatgeneralweird”.

Przetwarzanie danych - dwa algorytmy, regresja logistyczna (LR) oraz maszyna wektorów nośnych (SVM). Aby uzyskać różne reprezentacje cech, dane zostały poddane przekształceniom TF-IDF (term frequency–inverse document frequency) oraz Word2Vec.

Po przetworzeniu danych, uczenie maszynowe zostało zastosowane do analizy tekstu i klasyfikacji danych jako sarkastyczne lub niesarkastyczne..

W materiale [3] z racji podjęcia się problemu sarkazamu w języku o niskim zasobie (chiński) zaproponowano ramy uczenia przekrojowego oparte na prompt learning z rozszerzaniem danych i uczeniem kontrastowym (PDC), by ulepszyć transfer wiedzy między językiem angielskim.

3.5 Użyte technologie

Model można zaimplementować np. w PyTorch, frameworku do głębokiego uczenia w Pythonie. Do tokenizacji i ekstrakcji wbudowań słów z tekstu wejściowego można skorzystać z publicznie dostępnych zasobów - tokenizatora i wstępnie wytrenowanych wag z innego modelu np. "bert-base-uncased", aby przekształcić słowa na tokeny, a następnie tokeny na wbudowywanie słów. Wbudowania słów w tekście wejściowym są przekazywane przez szereg warstw Multi-Head Self-Attention. Wynik

jest następnie przekazywany przez pojedynczą warstwę GRU w obu kierunkach. Do trenowania modelu można użyć optymalizatora np. Adam.

Innym podejściem może być skorzystanie z dużych modeli językowych np. GPT-4o, Gemini, BERT. Można to zrobić korzystając z API poszczególnych platform udostępniających te modele

W analizie przewidywania sarkazmu dla 10 przykładowych zdań przetestowaliśmy najpopularniejsze modele językowe. Testowane modele to: gpt-4o i Gemini.

Zdania (w nawiasach czy to sarkazm):

“Naprawdę jesteś taki niemądry, czy tylko udajesz?”(tak),

“Ale ładna dziś pogoda, nie sądzisz?”(nie)

„Znowu spóźniłeś się na spotkanie. Twoja punktualność jak zwykle jest niezawodna”(tak),

„Nie udało Ci się? Co za niespodzianka!”(tak),

„Doskonała decyzja! Nie ma nic lepszego niż zaplanować przyjęcie w ogrodzie w dzień ulewy”(tak),

“Uwielbiam włoskie jedzenie, zwłaszcza makarony!”(nie)

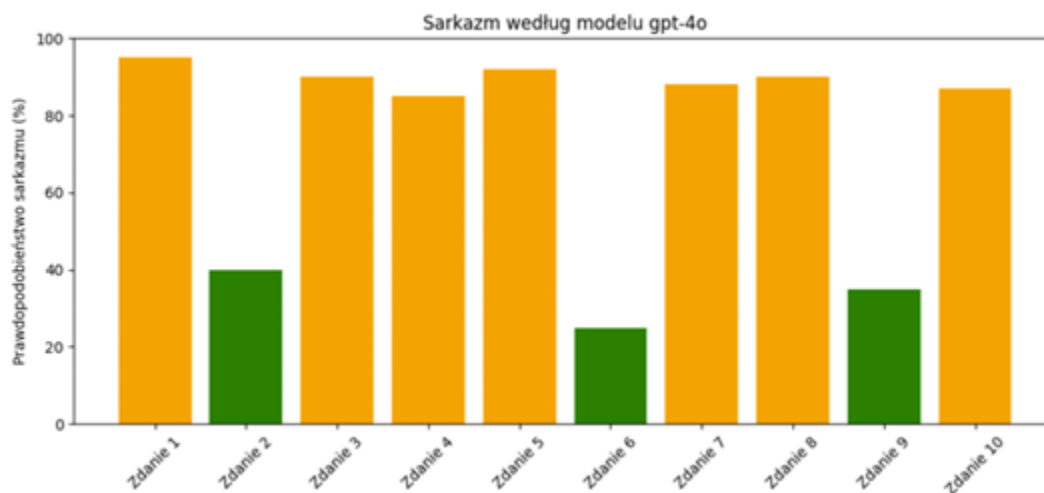
„Oczywiście, każdy z nas jest zgodny, że dwie godziny dodatkowej pracy w piątek wieczorem to najlepszy sposób na rozpoczęcie weekendu”(tak),

„Niesamowite! Jesteś specjalistą w każdej dziedzinie”(tak),

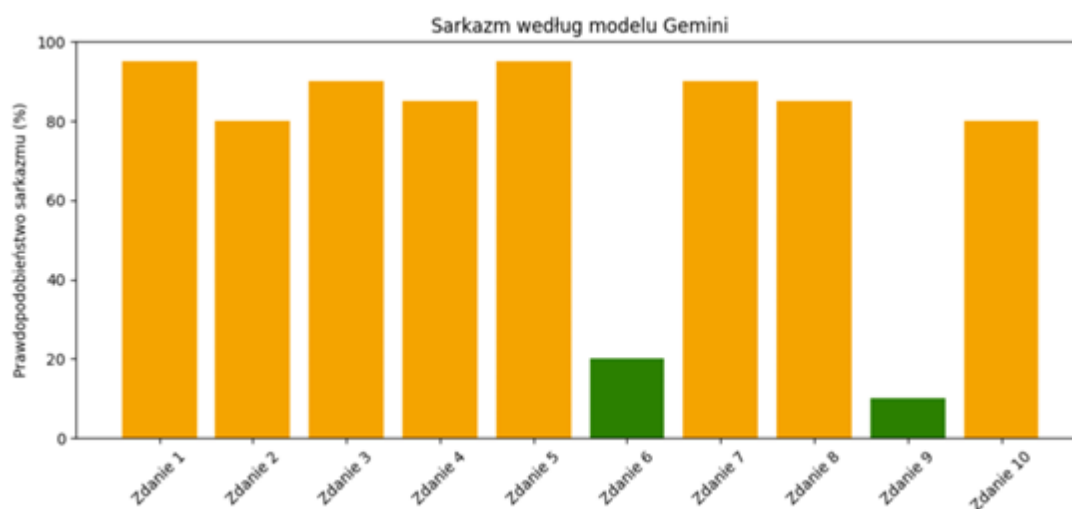
“Nie lubię rzodkiewki, ale ogórki i marchew już tak”(nie)

„Oczywiście, zrobię Ci kawę, bo właśnie miałam wychodzić do pracy”(tak)

Większość zdań zawierających sarkazm była poprawnie rozpoznana z wysokim poziomem pewności (Rys. 1, Rys. 2).



Rys. 1 Sarkazm według modelu gpt-4o (Opracowanie własne)



Rys. 2 Sarkazm według modelu Gemini (Opracowanie własne)

Zastosowanie detektora sarkazmu

Wraz z eksplozją użycia internetu, wykrywanie sarkazmu w komunikacjach online, takich jak media społecznościowe, fora dyskusyjne, chatboty i strony e-commerce, stało się kluczowe dla analizy opinii, analizy sentymentu oraz identyfikowania cyberprzemocy i dezinformacji [1]

4. Podsumowanie

Sarkazm, charakteryzujący się celowym używaniem słów w celu wyrażenia przeciwieństwa ich dosłownego znaczenia, stanowi złożoną formę komunikacji.

Niemniej jednak, to rozeznanie tematu przedstawia solidną ramę do utworzenia narzędzia, które pozwoli na wykrycie. Poprzez dokładne identyfikowanie przypadków sarkazmu, można przyczynić się do lepszego zrozumienia złożonych wzorców komunikacji w interakcjach online i wspomóc bardziej efektywne podejmowanie decyzji [1].

5. Bibliografia

[1] <https://aclanthology.org/2021.wassa-1.4.pdf>

[2] <https://www.emerald.com/insight/content/doi/10.1108/IDD-01-2023-0002/full/pdf?title=sarcasm-detection-in-online-comments-using-machine-learning13.10.2024>

[3] <https://www.mdpi.com/2079-9292/13/11/2163/pdf?version=1717568255>