

Modelowanie głębszych przekonań przy użyciu sieci neuronowych

Jakub Dajczak, Miłosz Chojnacki, Anton Delinac

July 8, 2022

Wstęp

Celem niniejszego projektu było przygotowanie oprogramowania umożliwiającego analizę głębszych przekonań człowieka oraz tego jak wpływa na nie dostęp do nowych informacji. W naszym przypadku przyjęte strony to osoby popierające szczepienia na covid-19 i osoby będące przeciwko tym szczepieniom. W dalszej części sprawozdania będziemy te dwie grupy nazywać kolejno "proszczenionką" oraz "antyszczepionką".

Wykorzystane biblioteki

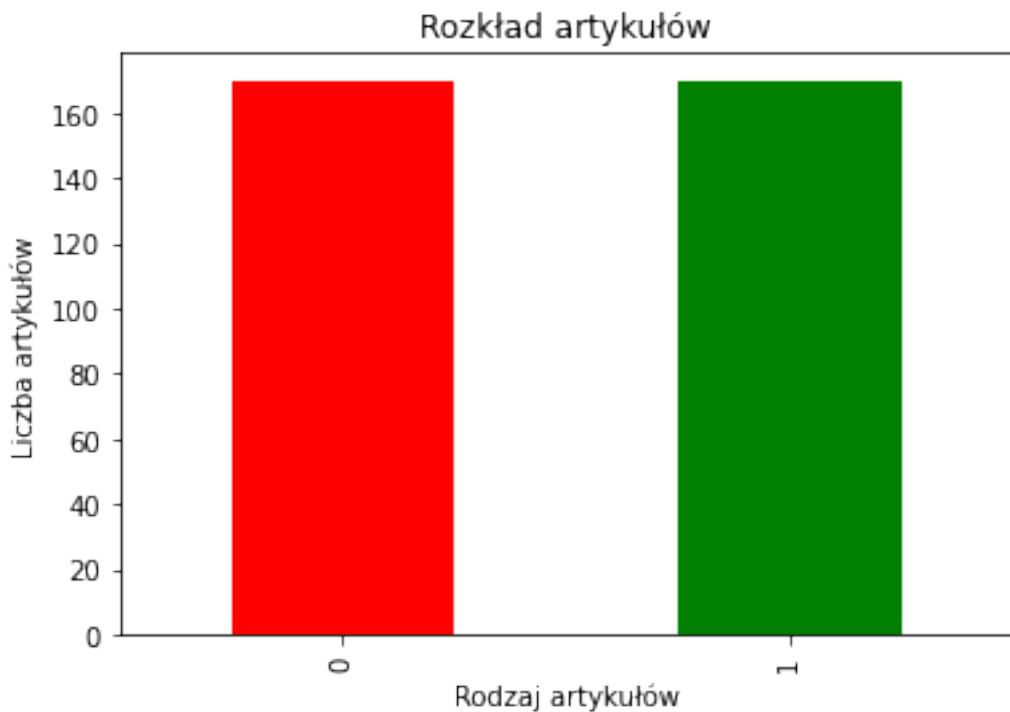
Do pracy przy projekcie posługiwiśmy się głównie biblioteką sklearn która posłużyła nam do przygotowania danych i klasyfikacji oraz matplotlib służącą do wizualizacji danych. Sieć samoorganizującą wykorzystaliśmy z modułu MiniSOM.

Dodatkowo stosowaliśmy wiele stworzonych przez nas funkcji, które pozwoliły nam ułatwić pracę nad projektem.

Dane

Dane do naszego projektu to artykuły, które pozyskaliśmy z artykułów naukowych lub pseudonaukowych oraz z publicznych for gdzie użytkownicy dobrowolnie upubliczniali swoje opinie na temat szczepień covid-19.

Liczba tych artykułów wynosi 340, na które składają się 170 artykułów grupy pro i 170 artykułów grupy anty. Dane zestawiliśmy w pojedynczym pliku .csv, który wczytaliśmy za pomocą biblioteki pandas. W dalszej części sprawozdania artykuły antyszczepionkowe są oznaczane liczbą 0, a artykuły proszczepionkowe liczbą 1.



Przygotowanie danych do klasyfikacji

Z wykorzystaniem biblioteki nltk oczyszczaliśmy artykuły usuwając z nich znaki interpunkcyjne, znaki białe, linki i "stop words". Dodatkowo zmienialiśmy wszystkie litery na małe litery.

Dodatkowo, próbowaliśmy szukać par słów, które występują w artykułach, bazując na już przygotowanych listach słów, jednak nie przyniosło to satysfakcjonujących wyników.

Przykładowe 10 pierwszych artykułów z grupy antyszczepionkowej:

```

0 It's been quite a week. I've been in the media...
1 This week has been eye-opening. Even for me --...
2 Yes, it's true. I beat COVID-19 in 48 hours wi...
3 The COVID-19 vaccines appear to be causing a g...
4 Back in the 1980s, I was a Columbia University...
5 So you got the vaccine because they told you t...
6 I Am A Living Proof That COVID-19 Is Fake This...
7 I have a PhD in virology and immunology. I'm a...
8 How do you convince the world's population to ...
9 Take 15 minutes and listen to this interview w...

```

Name: data, dtype: object

```

0 [quite, week, medium, business, decade, never, ...
1 [week, eye, opening, even, guy, warned, year, ...
2 [yes, true, beat, covid, hour, ivermectin, get...
3 [covid, vaccine, appear, causing, global, heal...
4 [back, columbia, university, student, learning...
5 [got, vaccine, told, get, forced, get, joe, bi...
6 [living, proof, covid, fake, unfiltered, take, ...

```

```
7 [phd, virology, immunology, clinical, lab, sci...
8 [convince, world, population, take, unproven, ...
9 [take, minute, listen, interview, hospital, nu...
Name: data, dtype: object
```

Klasyfikacja artykułów

Wektoryzacja danych

Następnym zadaniem było przygotowanie odpowiedniej klasyfikacji artykułów. Postanowiliśmy wybrać jeden z modeli przestrzeni wektorowych (SVM) - "Worek słów" (Bag of Words). Do wyznaczenia wektorów użyliśmy biblioteki sklearn, a z niej gotowego rozwiązania TfidfVectorizer.

Jest to bardzo popularny algorytm do transformacji tekstu na wartości liczbowe, służące do uczenia maszynowego. TfidfVectorizer jest połączeniem TfidfTransformer i CountVectorizer. CountVectorizer korzystając z przygotowanej przez nas funkcji generuje macierz, która zawiera liczby wystąpień słów w artykułach. TfidfTransformer jest oparty o TF-IDF - Term Frequency Inverse Document Frequency.

$$TFIDF = TF(t, d) * IDF(t)$$

gdzie:

- $TF(t,d)$ - liczba wystąpień słowa t w artykule d
- $IDF(t)$ - logarytm liczby artykułów w których wystąpił słowo t : $\log \frac{n}{1+df(t,d)}$

Wygląd naszej macierzy Bag of Words:

```
['aabduzrw' 'aabys' 'aadhaar' ... 'zoster' 'zuckerberg' 'zurich']
Tf-idf vectorizer
      aabduzrw  aaby    ...  zuckerberg  zurich
0          0.0  0.0    ...          0.0  0.0
1          0.0  0.0    ...          0.0  0.0
2          0.0  0.0    ...          0.0  0.0
3          0.0  0.0    ...          0.0  0.0
4          0.0  0.0    ...          0.0  0.0
...
335         0.0  0.0    ...          0.0  0.0
336         0.0  0.0    ...          0.0  0.0
337         0.0  0.0    ...          0.0  0.0
338         0.0  0.0    ...          0.0  0.0
339         0.0  0.0    ...          0.0  0.0
[340 rows x 14698 columns]
```

Klasyfikacja

Wstępnie naszym głównym klasifikatorem był klasyfikator Bayesa - Naive Bayes. Działał on na poziomie skuteczności 70% - 80%. W przypadku naszego projektu jednak zdecydowaliśmy się na

klasyfikator SVM, którego skuteczność wynosiła ponad 90%. SVM za pomocą funkcji kernelowych - w naszym przypadku funkcji liniowej - oblicza relacje między poszczególnymi artykułami i stara się dopasować płaszczyznę w wielowymiarowej przestrzeni tak, aby jak najlepiej oddzielała ona artykuły jednej klasy od drugiej. Odległości punktów od wyznaczonej płaszczyzny nazywamy wektorami wspierającymi, a marginesem nazywamy odległość najbliższych punktów od płaszczyzny dzielącej.

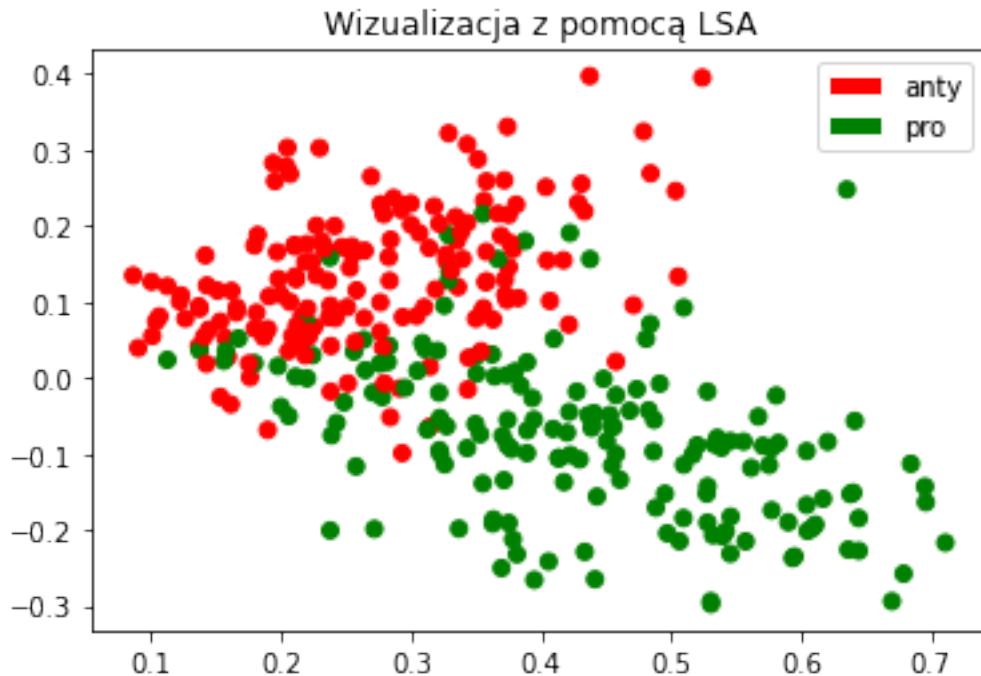
Ostatecznie użyliśmy modelu SGDClassifier z biblioteki sklearn. Jest to wersja algorytmu SVM z wstępny treningiem SGD.

Wizualizacja

Na potrzeby wizualizacji przyjeliśmy, że grupa artykułów antyszpionkowych będzie oznaczona kolorem czerwonym, a grupa proszczeniowa - zielonym.

Wykorzystując PCA (Principal component analysis), klasyfikator SVC oraz LSA (Latent semantic analysis), rzutujemy artykuły na przestrzeń dwuwymiarową.

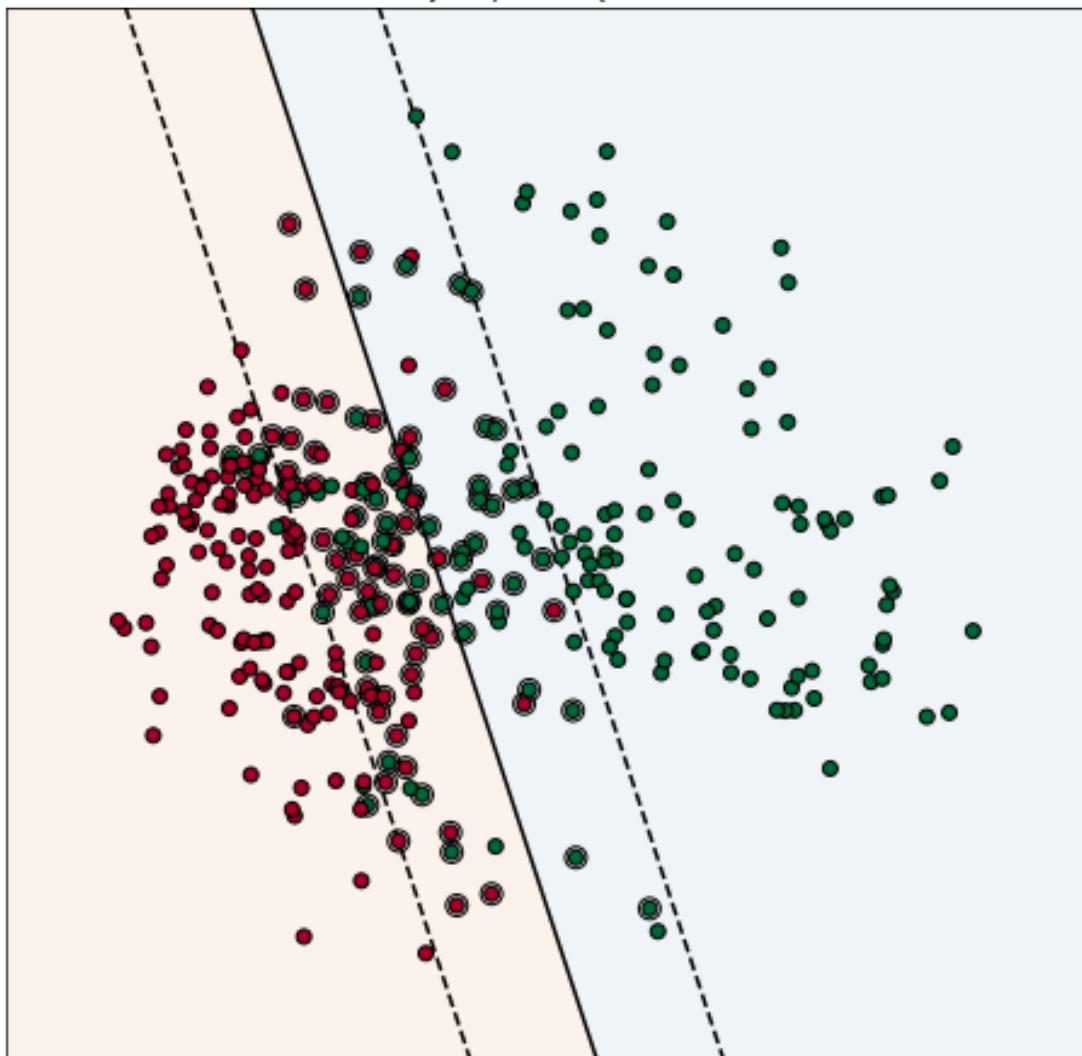
TruncatedSVD - implementacja algorytmu LSA z biblioteki sklearn. Podaliśmy na wejście algorytmu macierz Bag of Words, aby uzyskać kordynaty artykułów w przestrzeni.



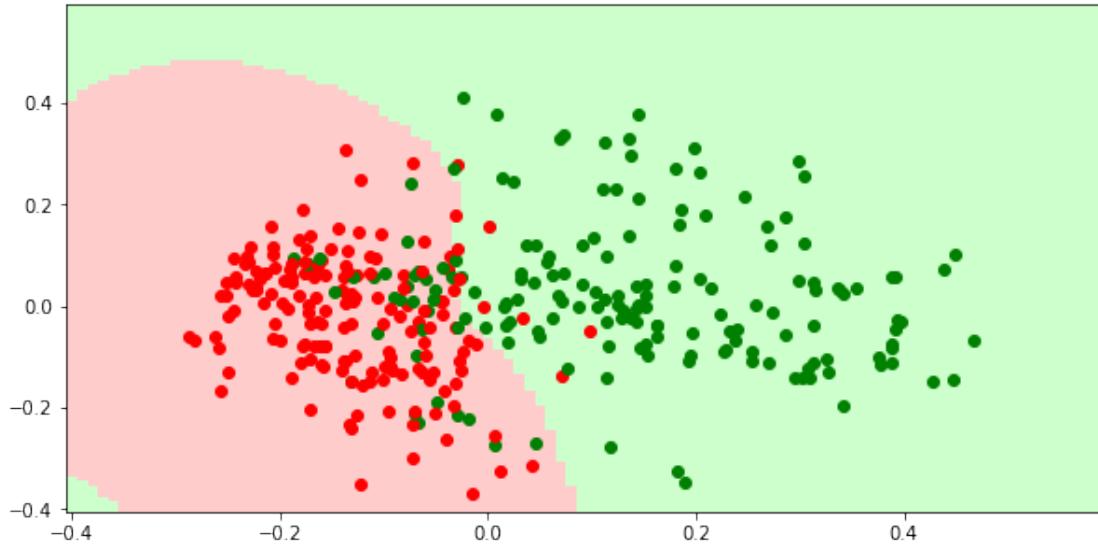
PCA - implementacja algorytmu PCA z biblioteki sklearn. Redukcja wymiarów macierzy w oparciu o wzajemne podobieństwa między artykułami.

SVC - Support Vector Classification. Również z implementacji sklearn. Na pierwszym wykresie korzystając z liniowej klasyfikacji, na drugim z nielinowej.

Wizualizacja z pomocą PCA oraz SVC



Wizualizacja z pomocą PCA oraz SVC



Jak widać na powyższych wykresach granica między artykułami antyszpiegowskimi, a proszepionkowymi jest dobrze widoczna, jednak margines obejmuje niektóre artykuły, z klasy przeciwej.

Wizualizacja za pomocą PCA i SVC jest bardziej czytelna, ale to głównie dzięki zastosowaniu kolorów tła i linii hiperpłaszczyzny.

Wyznaczenie słów kluczowych

Do wyznaczenia słów kluczowych wykorzystaliśmy wyżej już wspominany klasyfikator SGDClassifier, który zapisuje cechy danej klasy oraz odpowiadające im wagie. Napisaliśmy funkcję do pozyskania tych kluczowych słów w oparciu właściwie o wagie klasyfikatora. Dla każdej klasy uzyskaliśmy po 50 słów kluczowych.

	PRO	ANTI
1	barré	injury
2	vaccinated	report
3	safer	tinnitus
4	worker	adverse
5	cell	mass
6	protection	dobbs
7	getting	injection
8	immunocompromised	government
9	additional	ivermectin
10	oxford	mandate
11	horo	wuhan
12	might	jab
13	omicron	death
14	le	kill
15	candidate	lymphocyte
16	seroconversion	wrote
17	two	israel
18	volunteer	medium
19	clinical	vanden
20	illinois	article
21	restaurant	hour
22	travel	fauci
23	biontech	autoimmune
24	protect	pilot
25	infected	lie
26	monitoring	bossche
27	rare	natural
28	approval	ulcer
29	placebo	batch
30	morris	fraiman
31	side	product
32	booster	lied
33	lower	experimental
34	people	harrington
35	chance	paper
36	corbett	fraud
37	participant	biden
38	moderna	doctor
39	safety	snake
40	antigen	innate
41	dose	immunity
42	sarscov	order
43	efficacy	marburg
44	effective	dore
45	dos	aid
46	part	evidence
47	risk	joe
48	trial	patent
49	get	depopulation
50	vaccine	native

Sieć samo-organizująca

Mapa Kohonena zwana również siecią neuronową Kohonena jest najbardziej znaną i najczęściej stosowaną siecią samouczącą się, realizującą zasadę samoorganizacji (SOM). Sieć ta cechuje się tym, że działają w wielowymiarowych przestrzeniach danych wejściowych, w związku z czym warstwa wejściowa zawiera bardzo wiele neuronów. Proces uczenia można podzielić na kilka etapów:

1. Inicjalizacja sieci wagami
2. Wylosowanie danych wejściowych
3. Wybranie wygranego neuronu na bazie odległości cosinusowej (często używa się także odł. euklidesowej)
4. Aktualizacja wag neuronów
5. Powrót do kroku drugiego

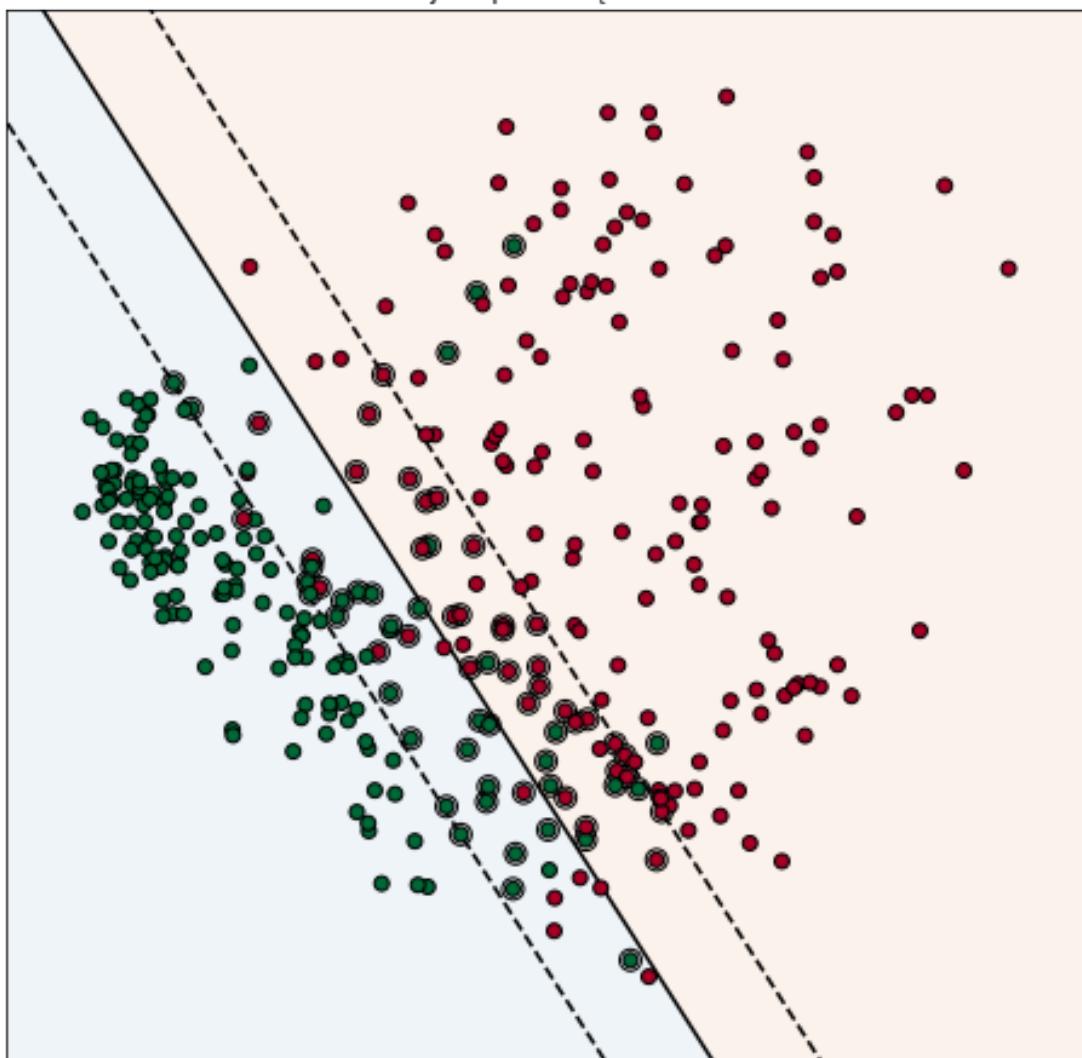
Bardzo ważnym krokiem jest aktualizacja wag neuronów. Wartość o jaką aktualizujemy dany neuron zależy od współczynnika uczenia (learning rate) oraz sąsiedztwa topologicznego z innymi słowami. W dalszej części sprawozdania będziemy analizować wpływ learning rate'u na proces uczenia.

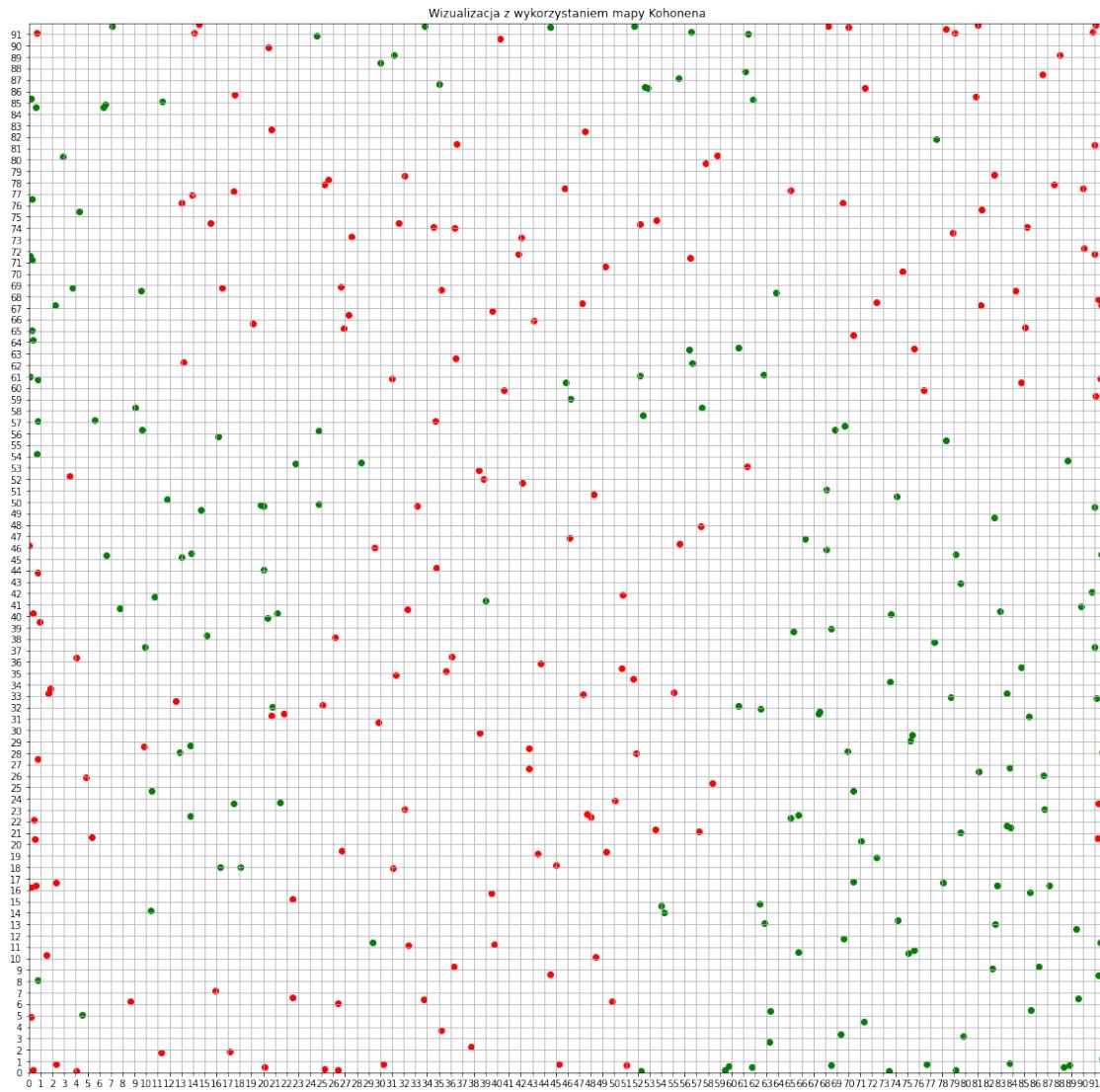
Wektory artykułów

Na wstępny etapie pracy postanowiliśmy zwizualizować nasze artykuły w postaci wektorów słów kluczowych. Tym sposobem pozbywamy się wielu nieistotnych informacji, które nie są potrzebne do analizy. Dla każdego z artykułów zliczamy ilość wystąpień każdego słowa kluczowego. Następnie, dla normalizacji wyników używamy TfifdfTransformer - jednego z etapów już wyżej wspomianego TfifdfVectorizera.

Tak przygotowane wektory używamy tak samo jak przy wizualizacji artykułów przy pomocy PCA i SVC. Efekt jest o wiele lepszy niż w używania pełnych wektorów. W dalszym etapie inicjalizujemy mapę Kohonena i podajemy wektory artykułów do nauki. Po ukończeniu etapu uczenia wyświetlamy wyniki w postaci wykresów.

Wizualizacja z pomocą PCA oraz SVC





Wektory słów kluczowych

Dalszą częścią zadania była wizualizacja wektorów słów kluczowych w zależności od ich sąsiedztwa. Napisaliśmy więc funkcję zliczającą wystąpienia sąsiadujących słów dla każdego słowa kluczowego oraz funkcję generującą słownik, gdzie kluczem jest słowo kluczowe, a wartością słowników słów sąsiadujących z liczbami wystąpień tych słów jako wartości. Z tego też przy podaniu odpowiedniego sąsiedztwa uzyskiwaliśmy wektory słów kluczowych.

Wektory słów kluczowych - symulacja poglądów

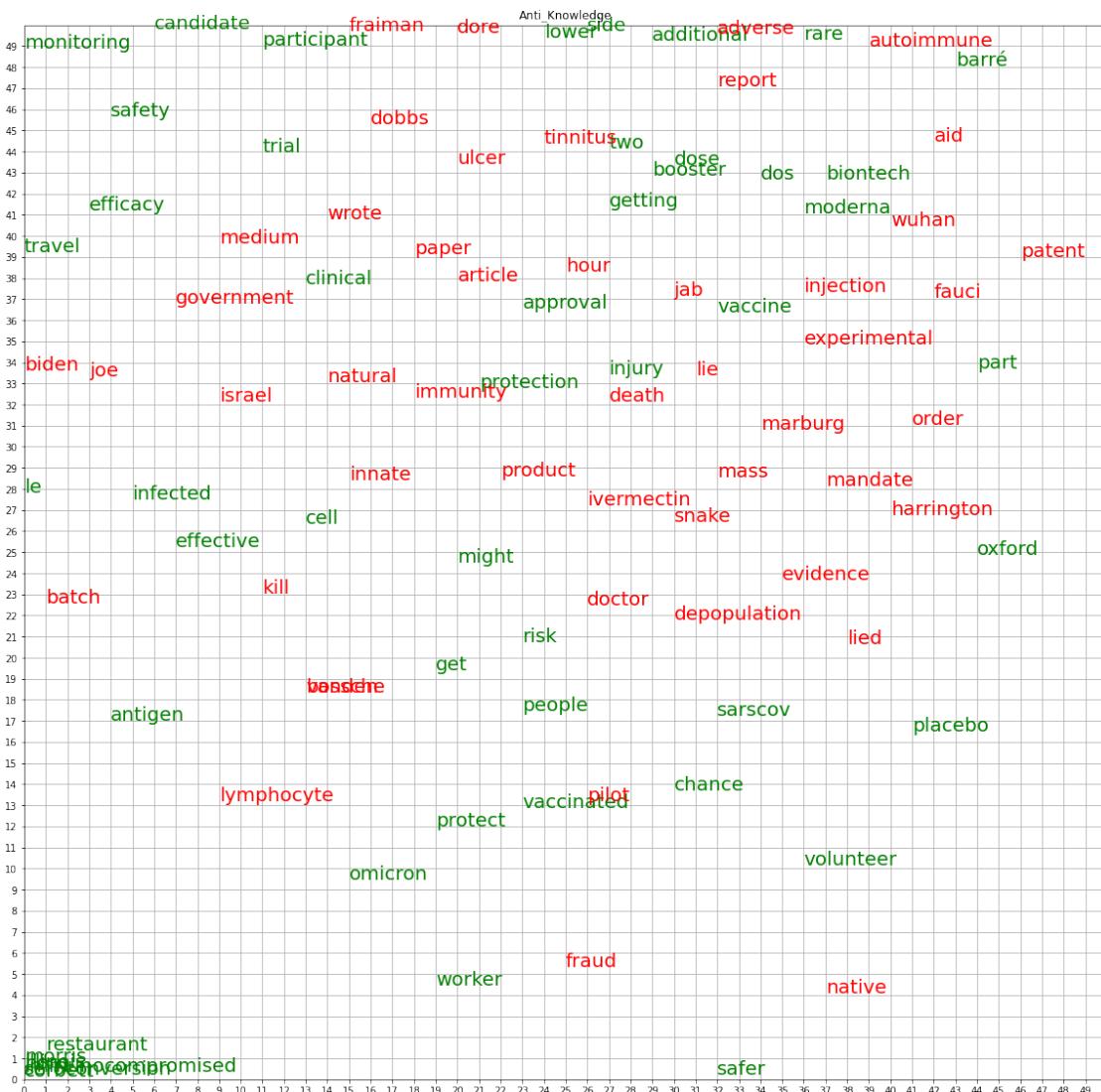
Nastecną częścią zadania była symulacja poglądów. W tym etapie używamy generatora wektorów słów kluczowych, dla którego stopniowo zwiększamy zasób słownictwa który odpowiada kolejnym przeczytanym artykułom. Wybór artykułów jest losowy i jest warunkowany jedynie zbiorem jaki podam na wejście.

Badamy 4 możliwe scenariusze:

- Osoba wierząca w teorię antyszczepionkową zaczyna czytać artykuły zachęcające do szczepień;
 - Osoba przekonana co do skuteczności szczepionek zaczyna czytać artykuły przeciwko szczepieniom;
 - Osoba niezdecydowana zaczyna czytać artykuły zachęcające do szczepień;
 - Osoba niezdecydowana zaczyna czytać artykuły przeciwko szczepieniom.

Dla każdej z trzech osób wstępnie przygotowujemy mapę kohonena oraz uczymy ją 80% wszystkich artykułów z danego zbioru. Przygotowane w mapy zwizualizowane są poniżej:

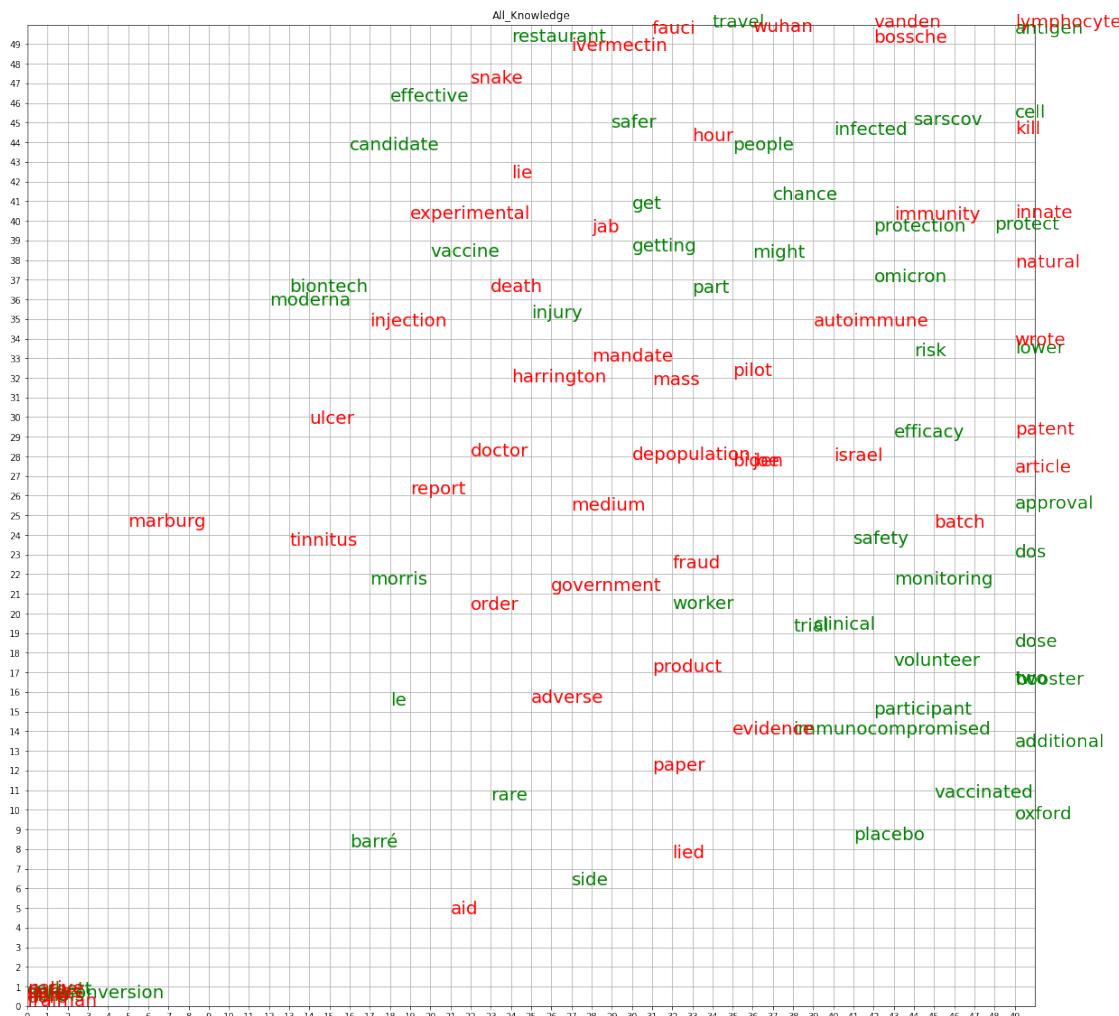
Osoba wierząca w teorię antyszczepionkową



Osoba przekonana co do skuteczności szczepionek



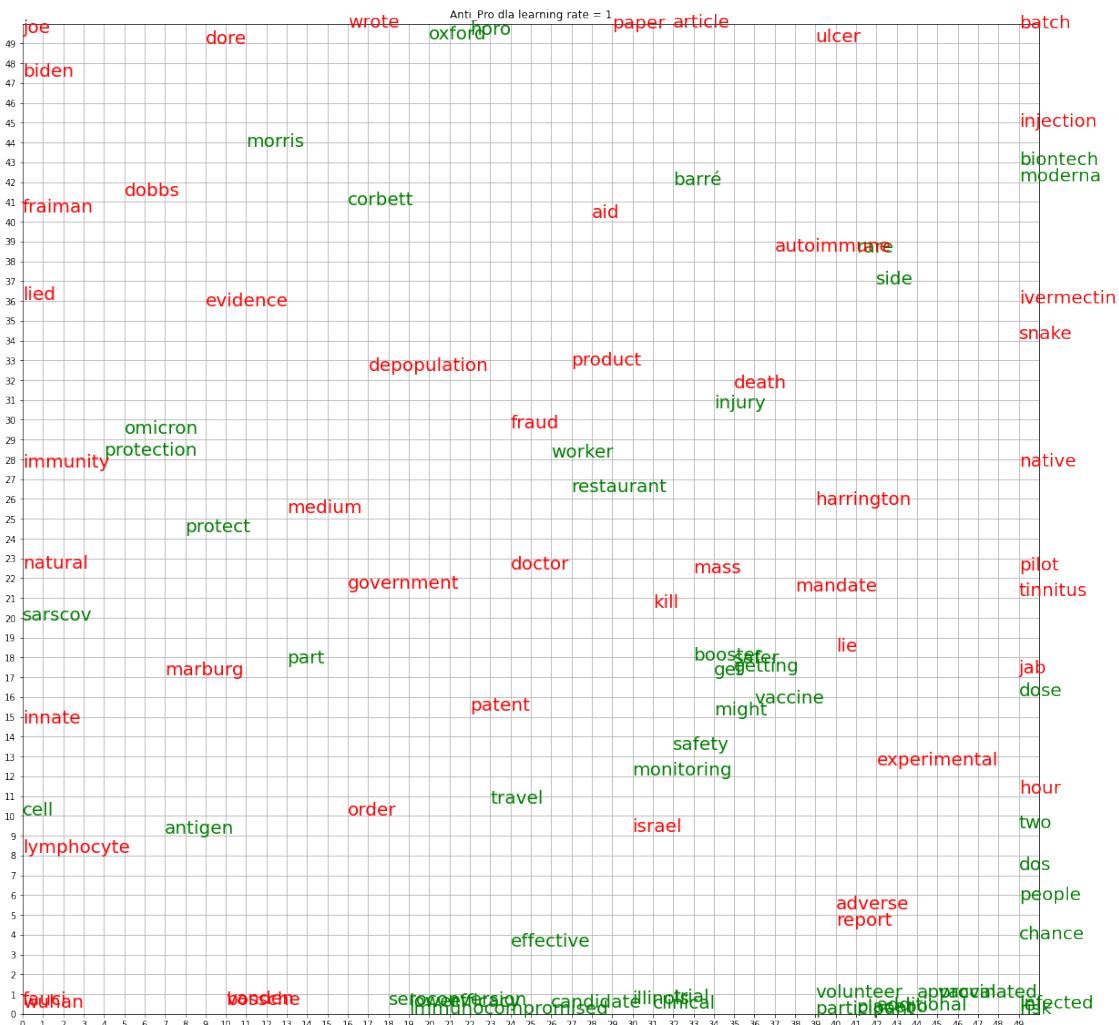
Osoba niezdecydowana



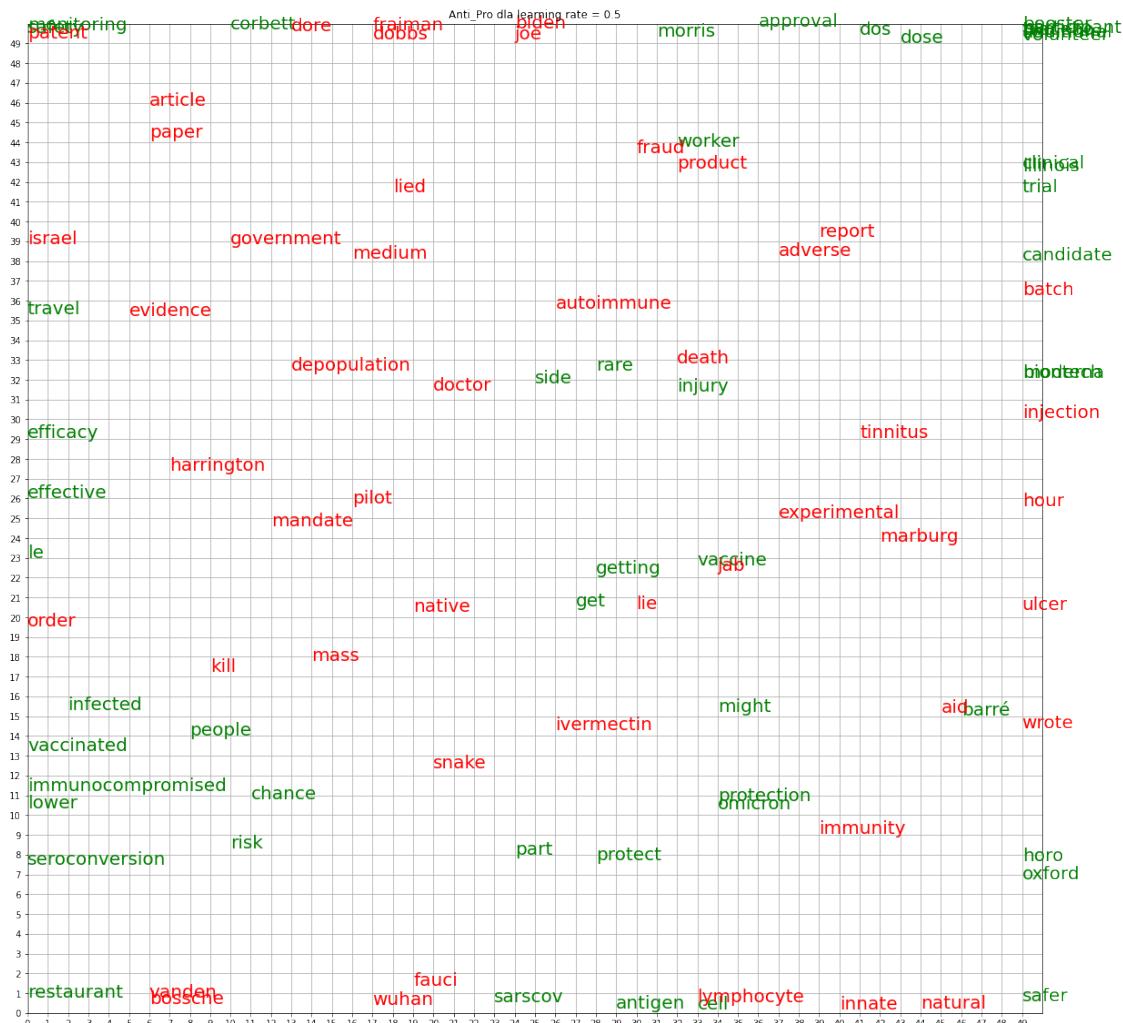
W dalszej części sprawozdania znajdują się wykresy obrazujące wyniki symulacji. Dla każdego przypadku jest przedstawione 7 wykresów. Każdy kolejny wykres jest symulacją osoby z inną podatnością na przeczytane informacje. Pierwszy wykres to symulacja osoby łatwotworowej i szybko zmieniającej poglądy, a każdy następny wykres odpowiada osobie, która jest coraz mniej otwarta na nowe informacje.

Osoba antyszczepionkowa czyta artykuły zachęcające do szczepień

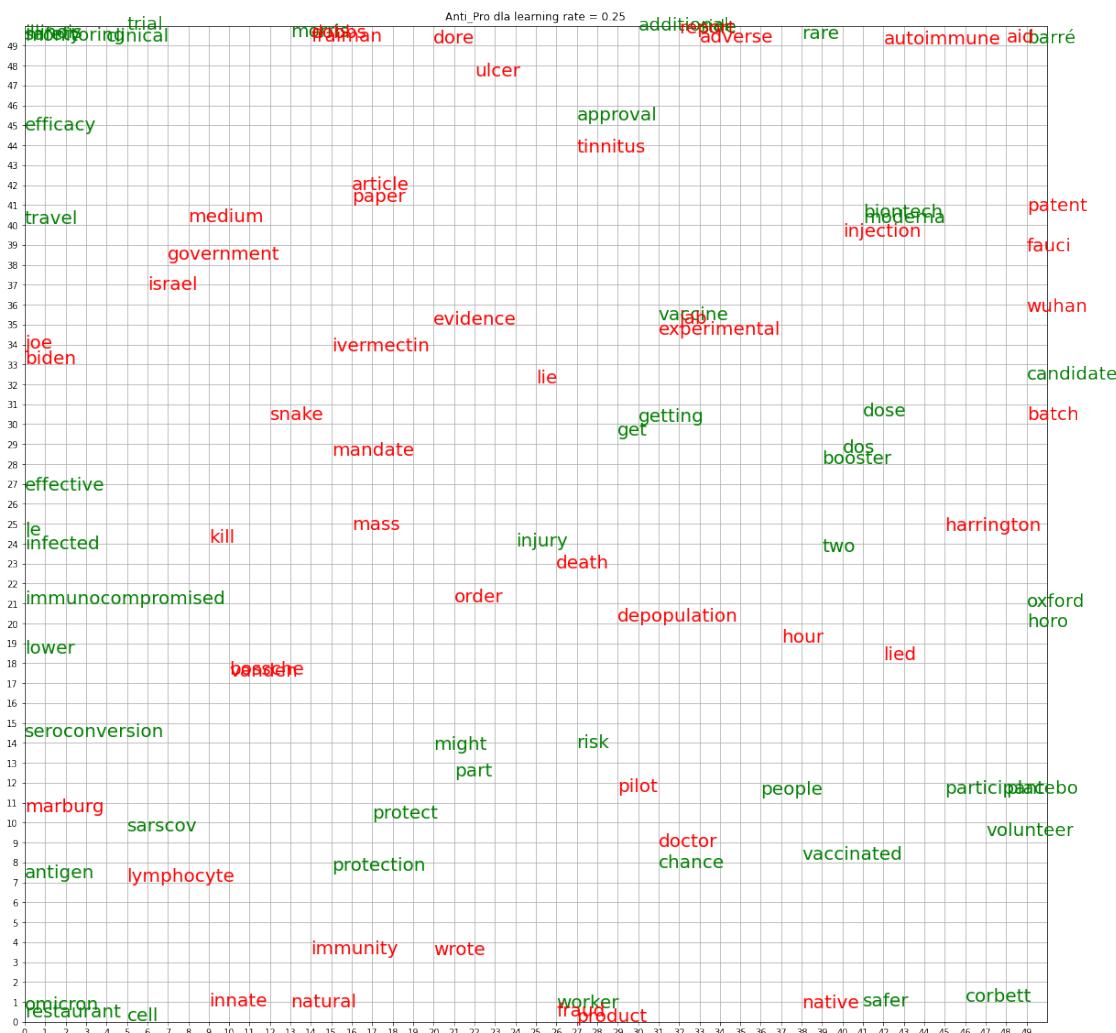
Learning rate = 1.0



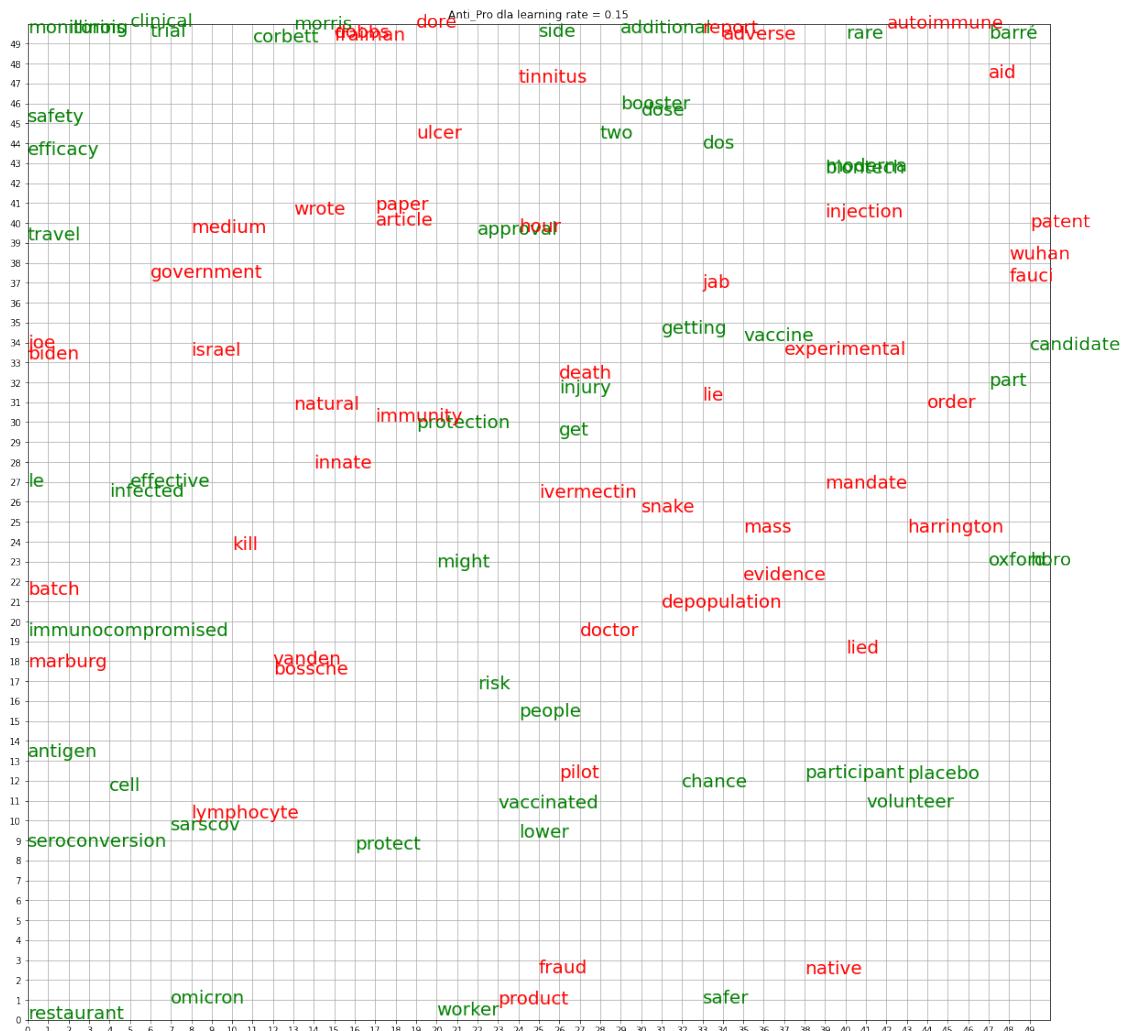
Learning rate = 0.5



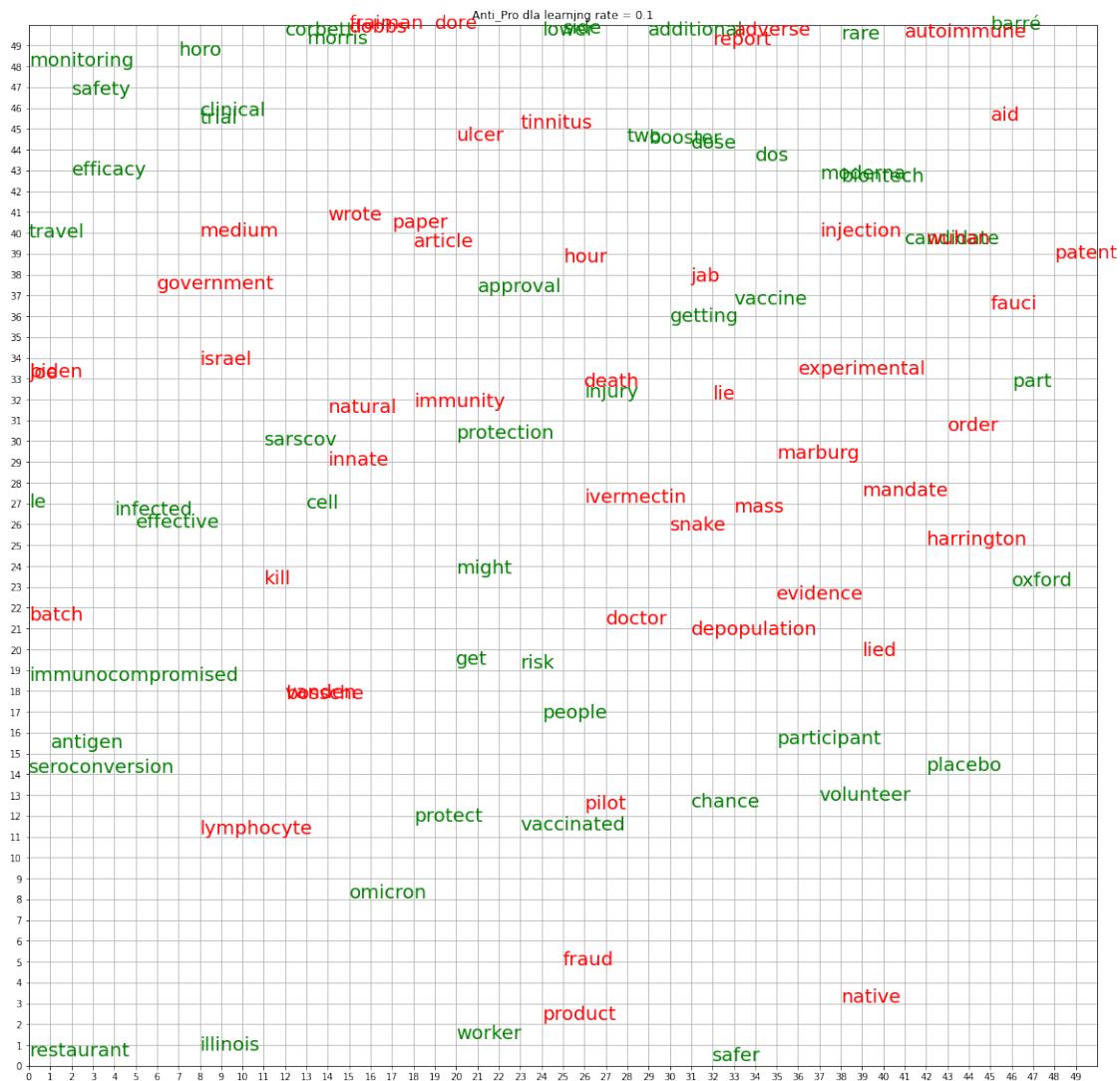
Learning rate = 0.25



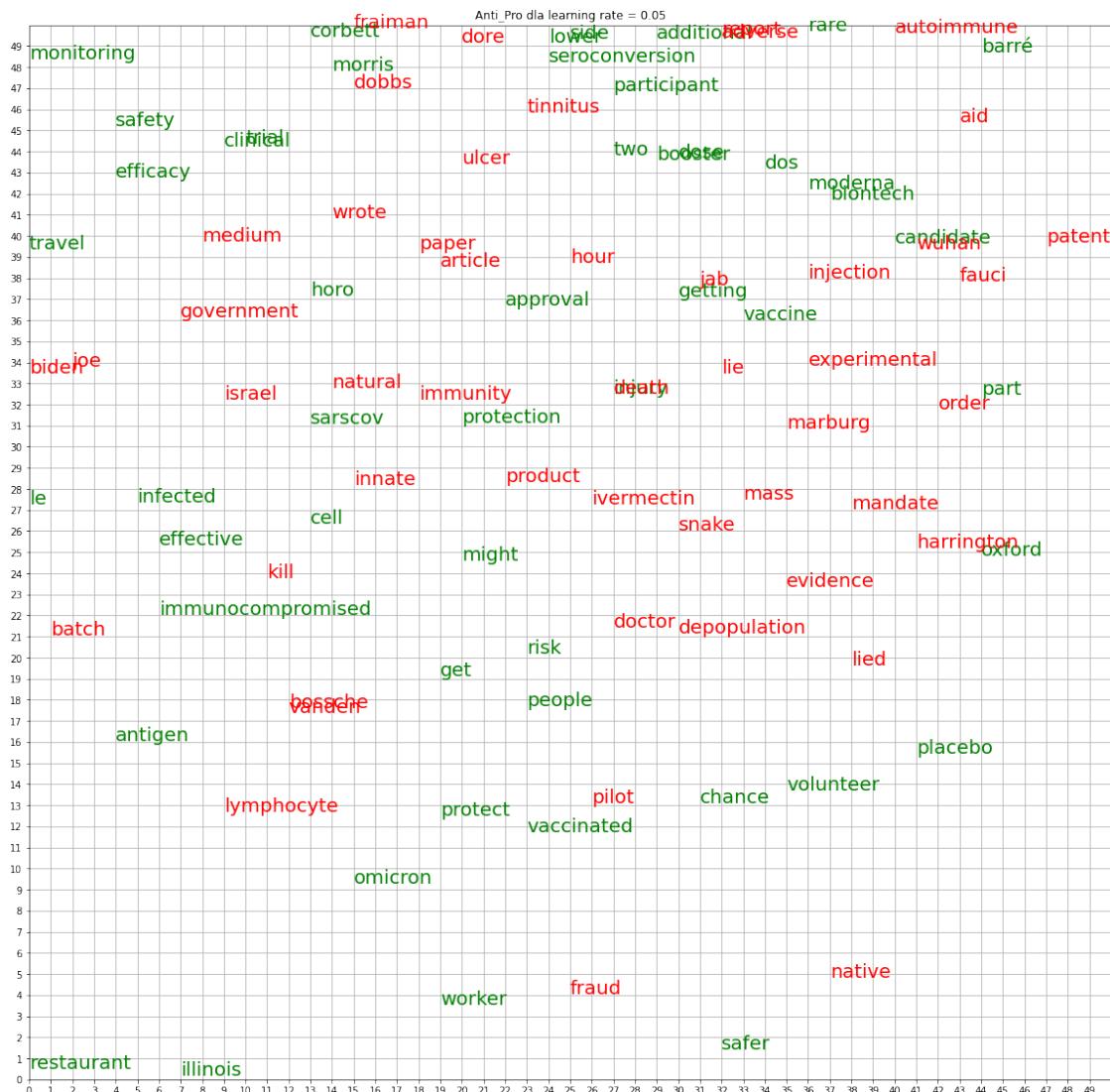
Learning rate = 0.15



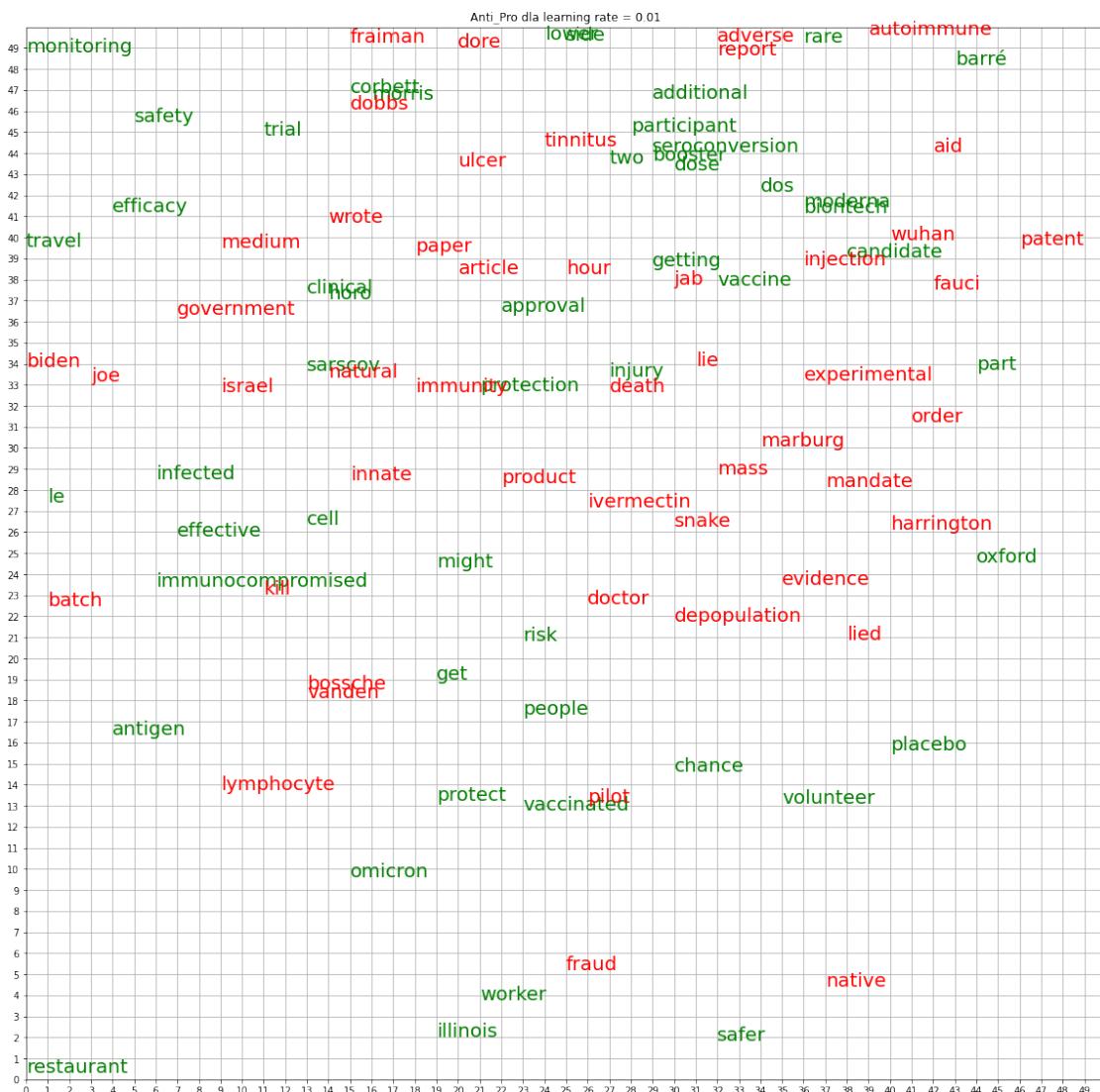
Learning rate = 0.1



Learning rate = 0.05

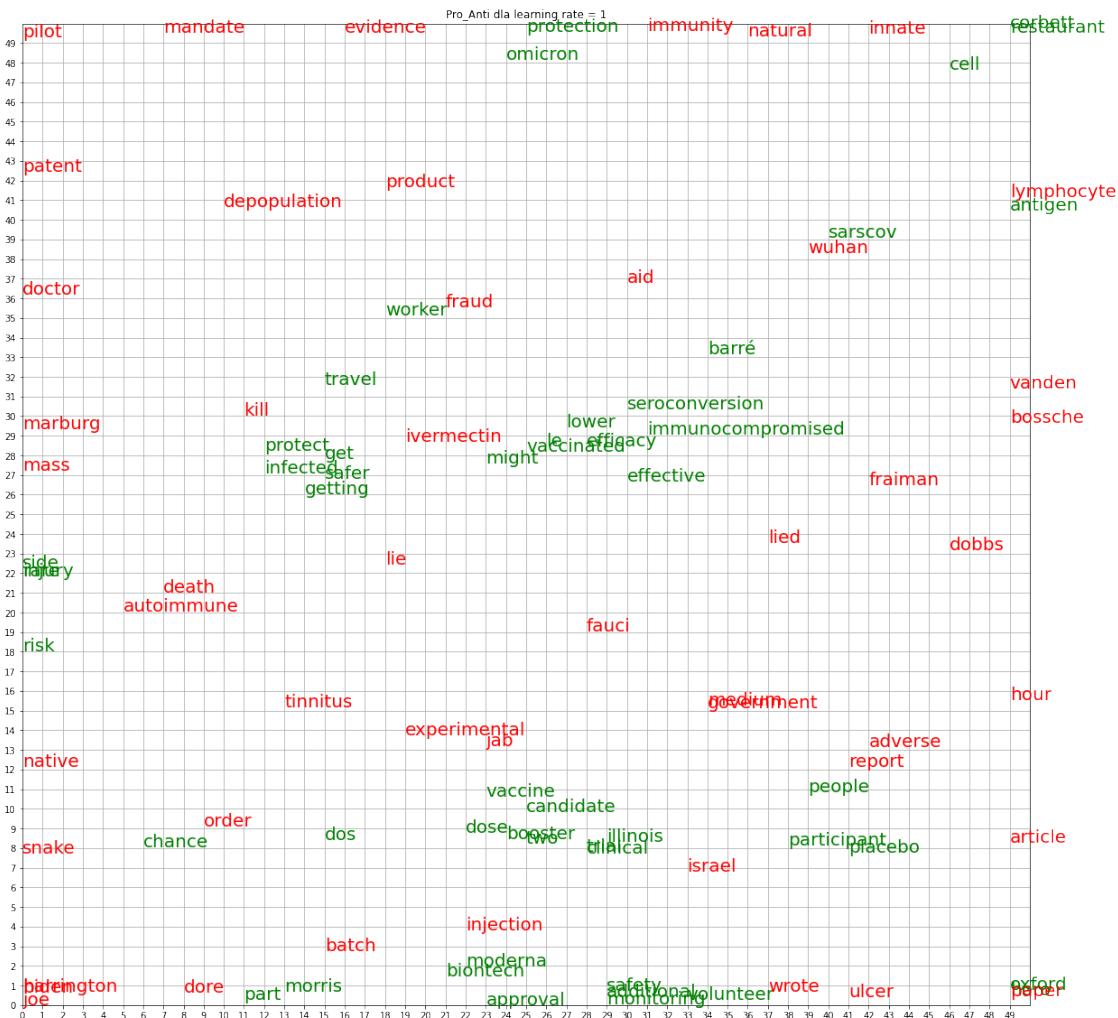


Learning rate = 0.01

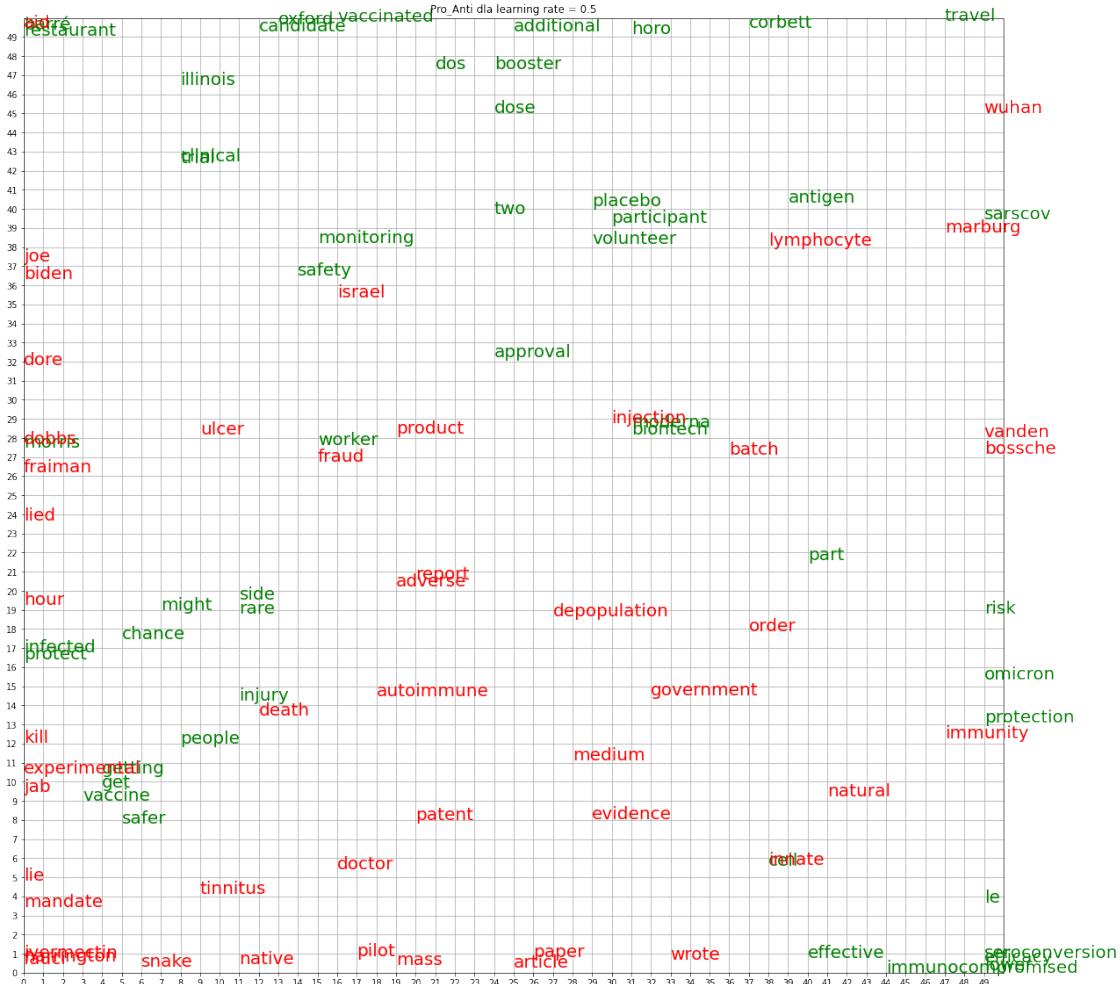


Osoba przekonana co do skuteczności szczepionek czyta artykuły przeciwko szczepieniom

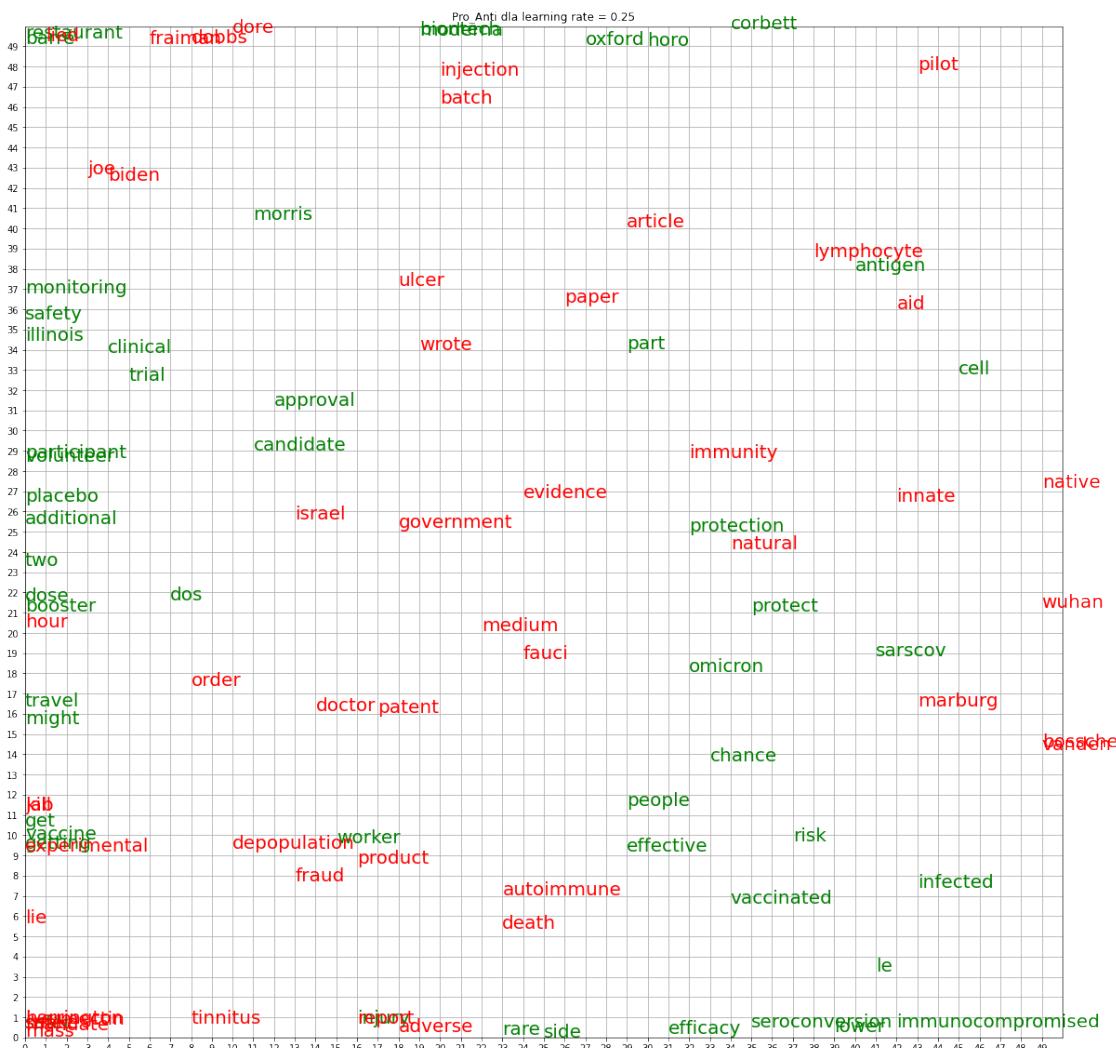
Learning rate = 1.0



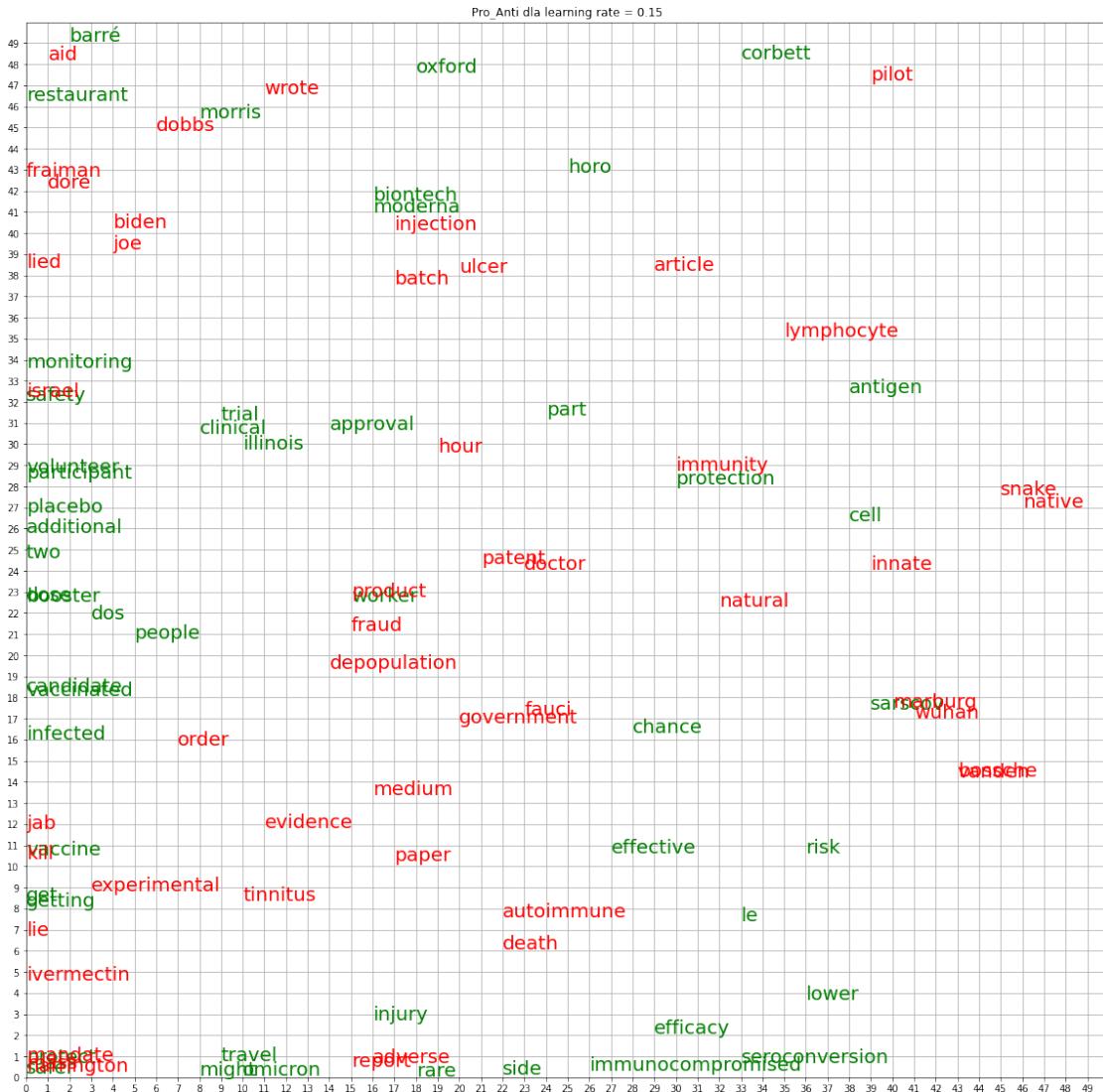
Learning rate = 0.5



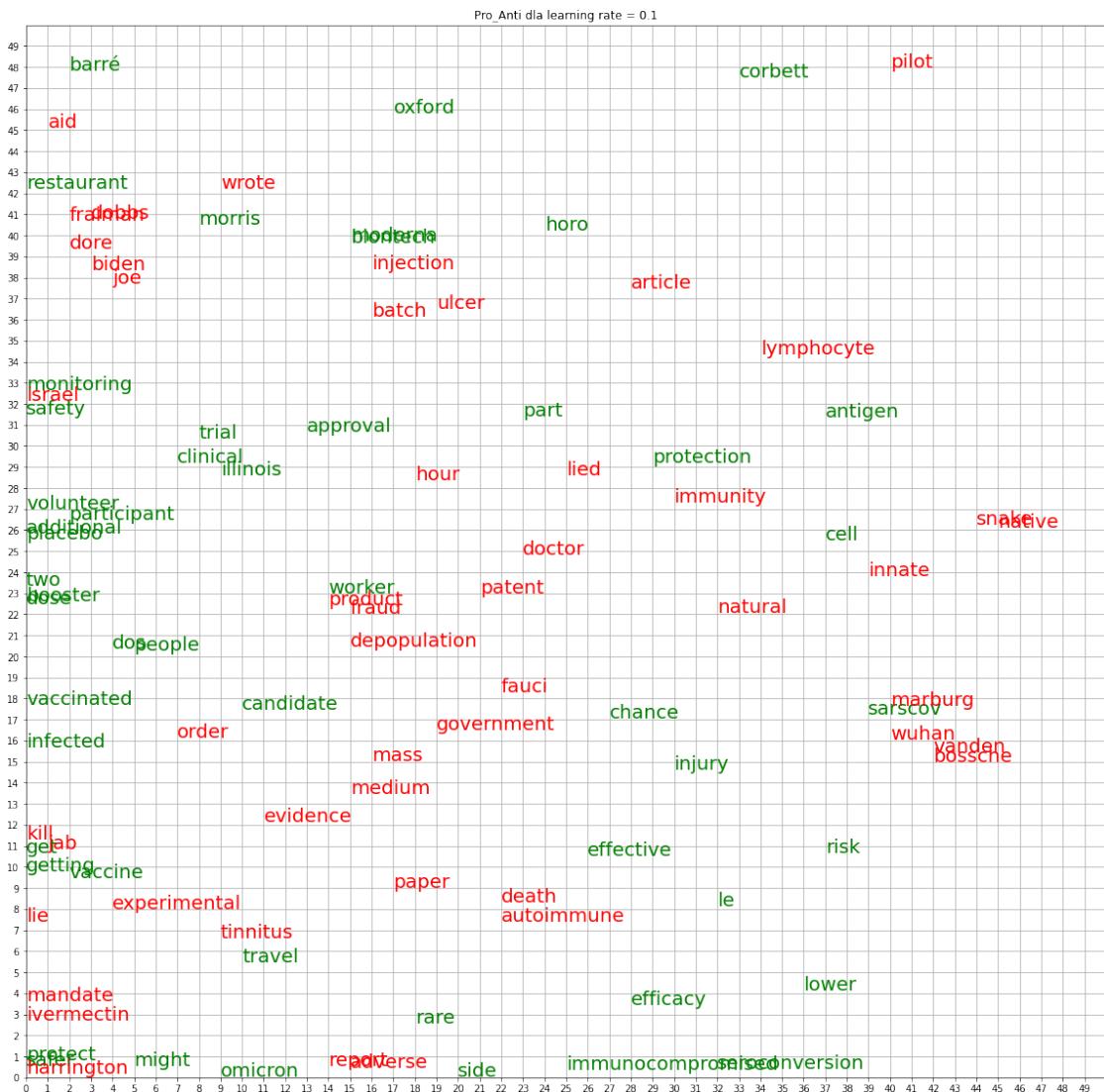
Learning rate = 0.25



Learning rate = 0.15



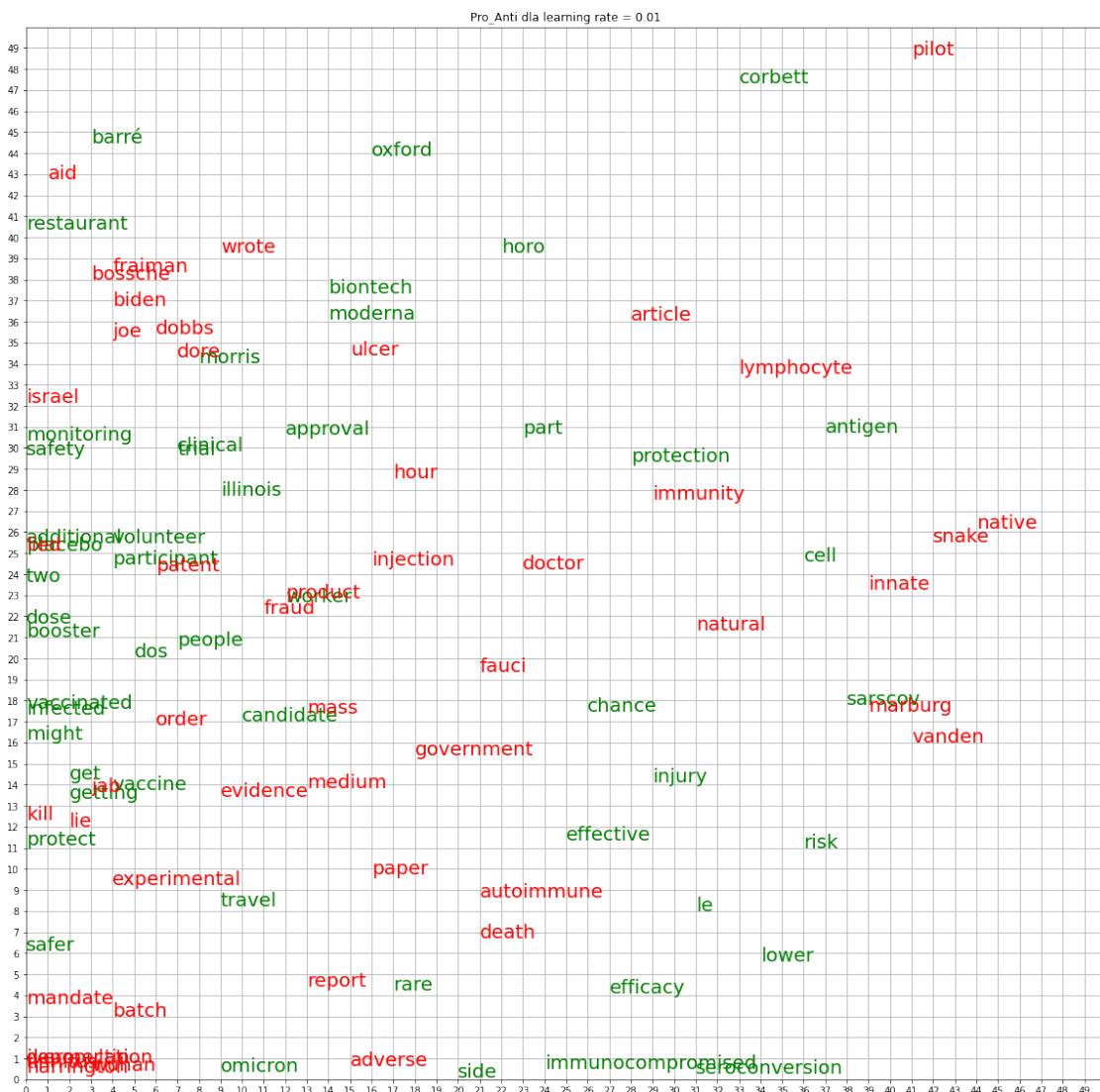
Learning rate = 0.1



Learning rate = 0.05

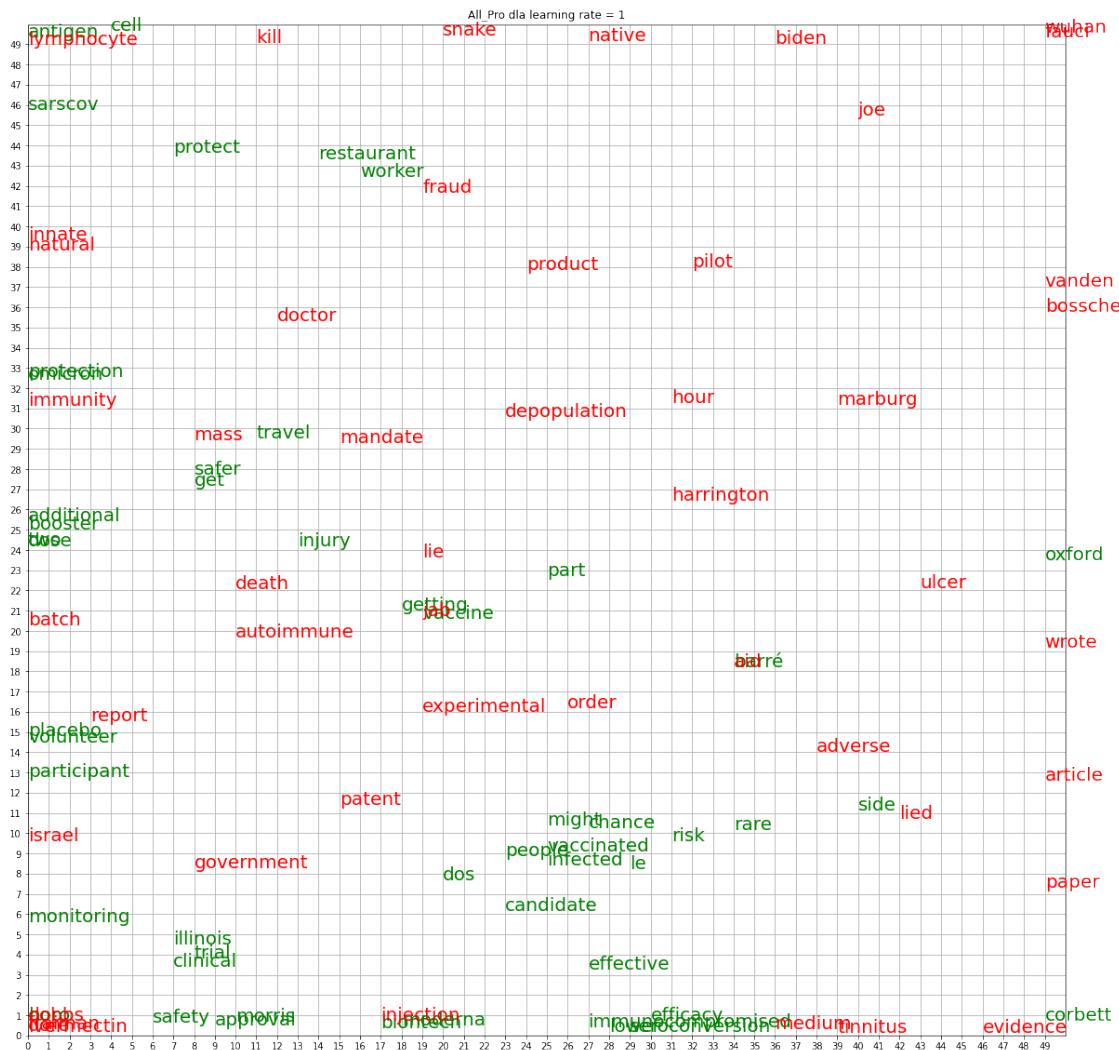


Learning rate = 0.01

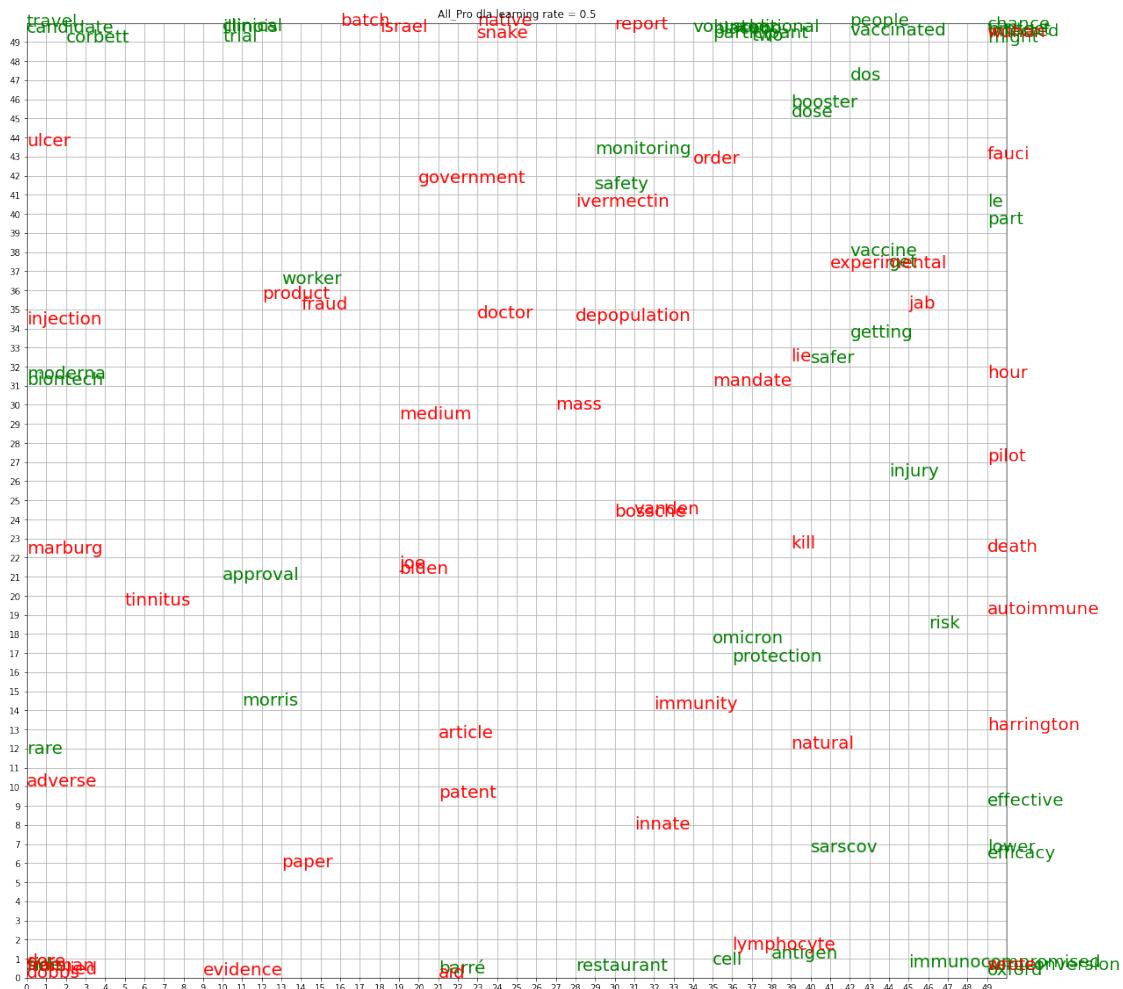


Osoba niezdecydowana czyta artykuły zachęcające do szczepień

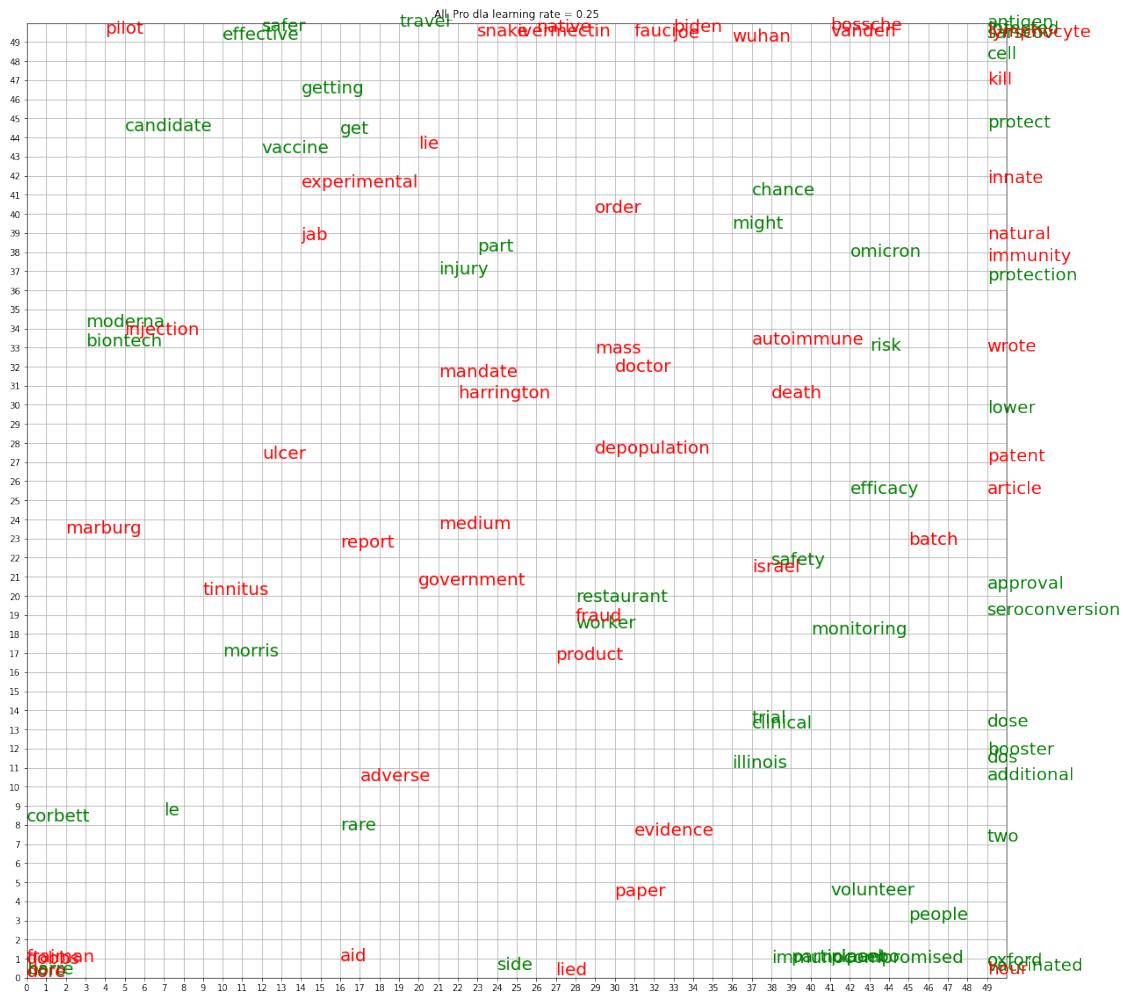
Learning rate = 1.0



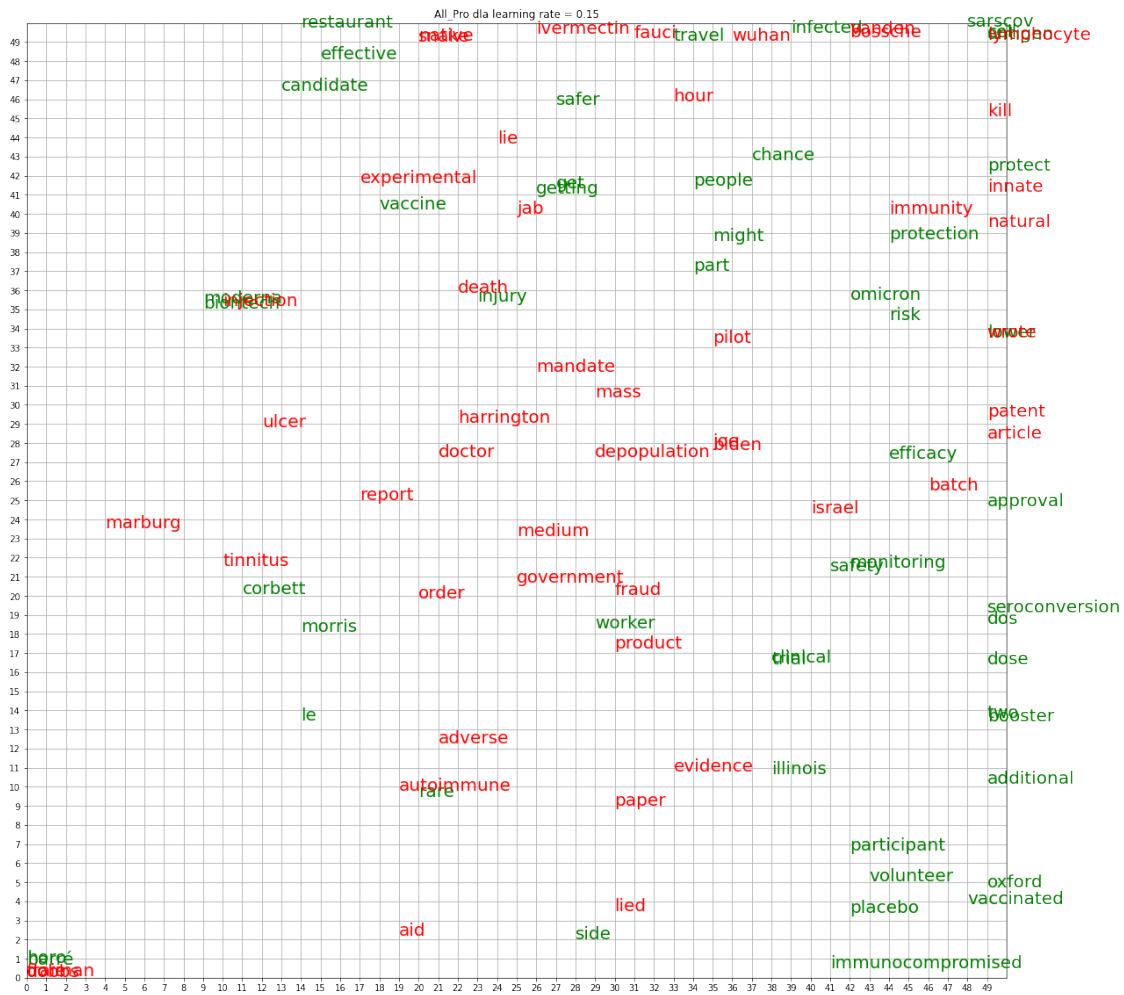
Learning rate = 0.5



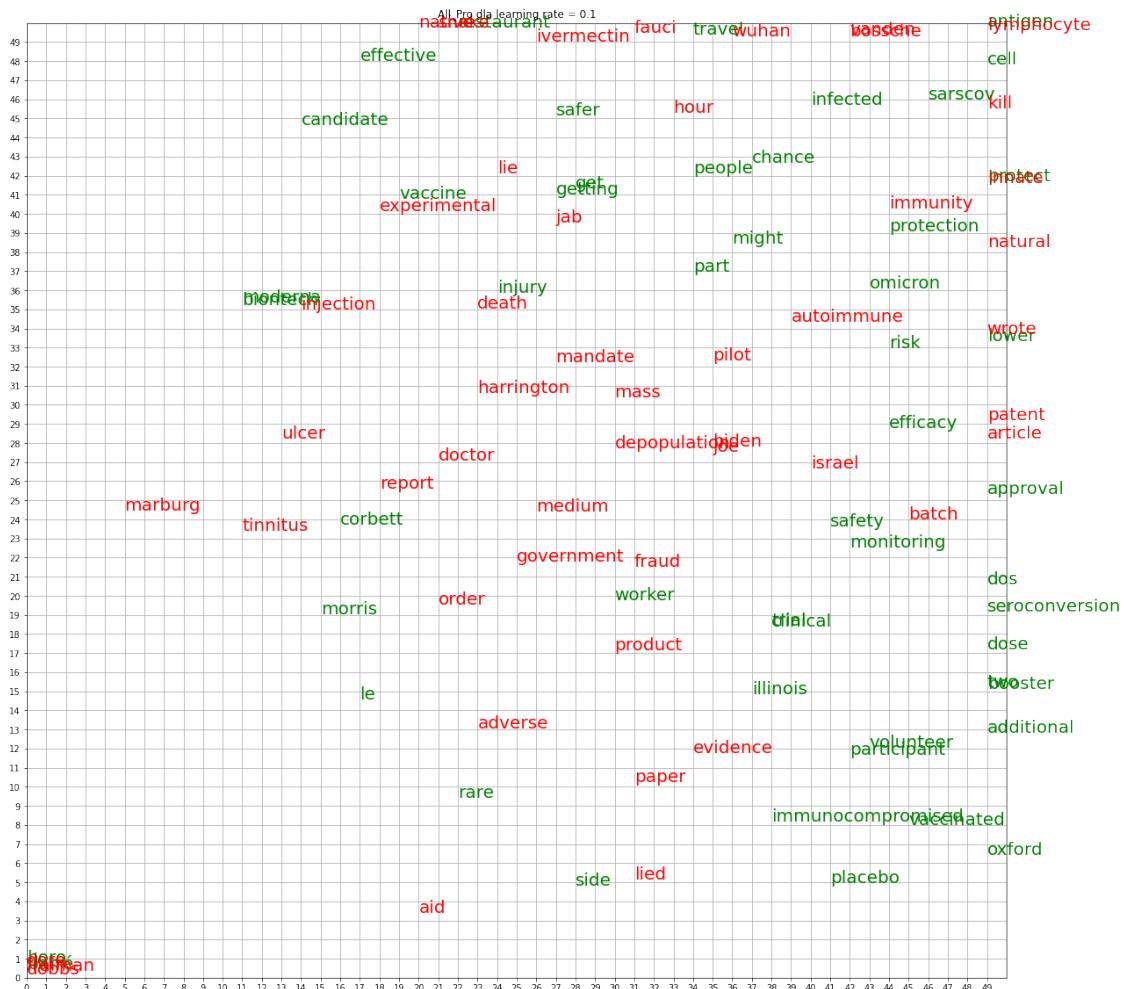
Learning rate = 0.25



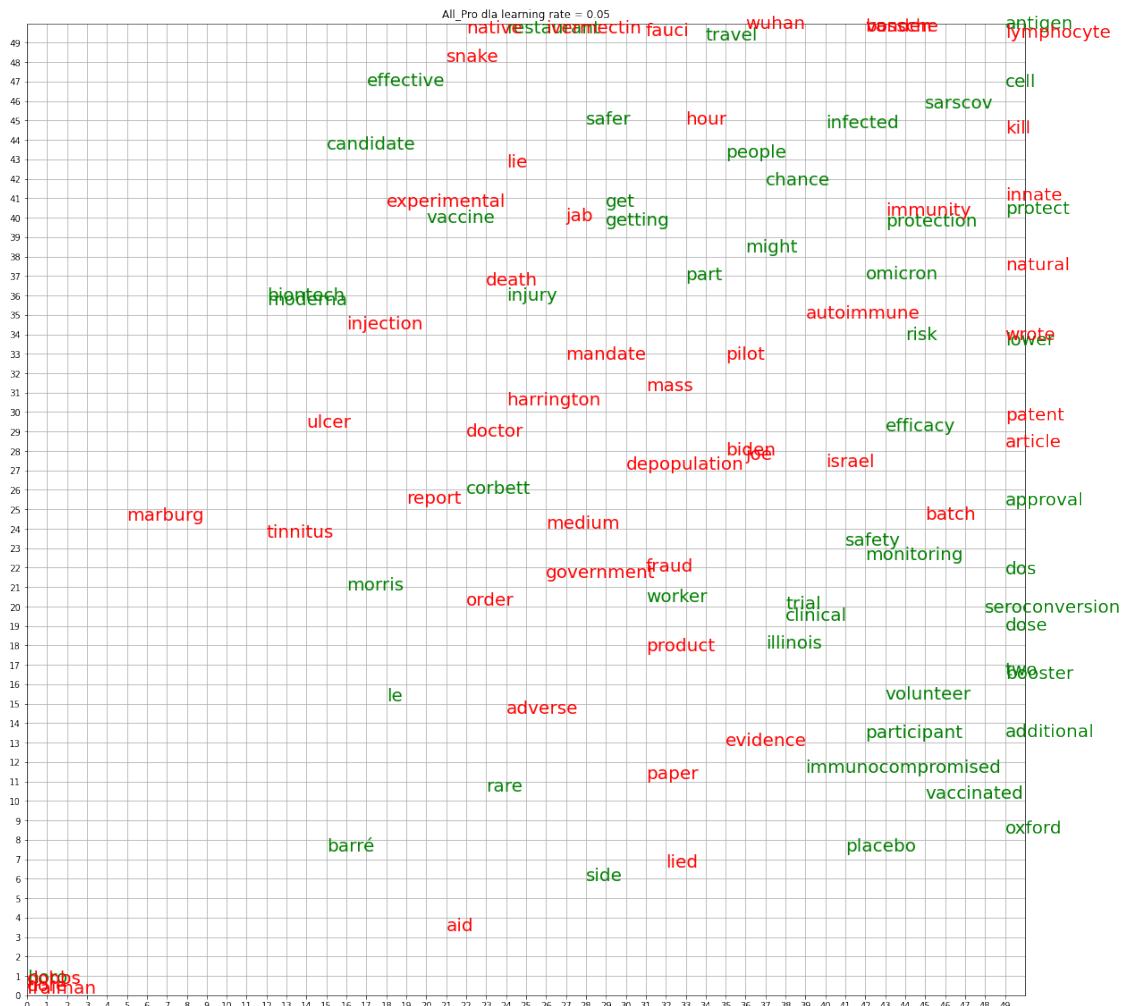
Learning rate = 0.15



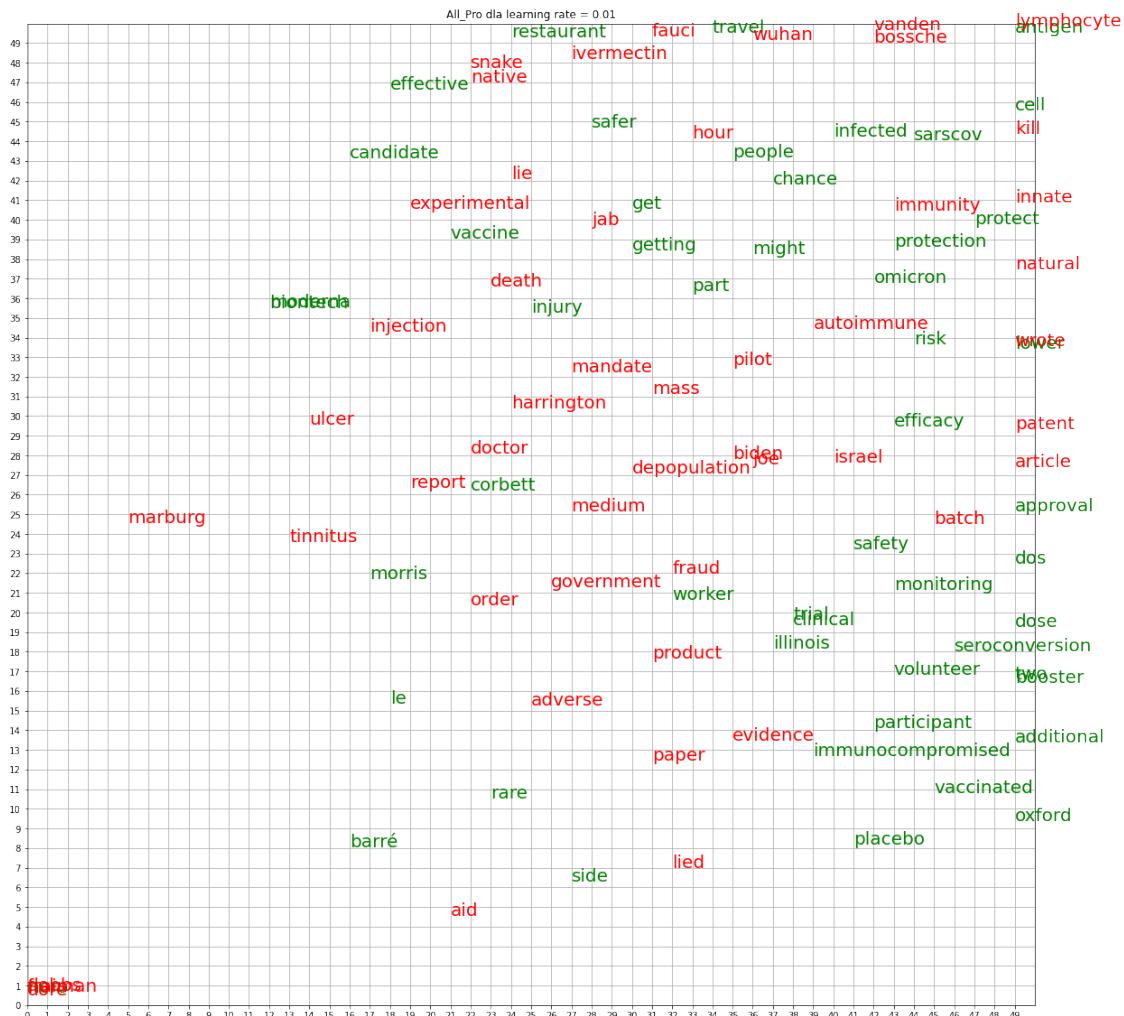
Learning rate = 0.1



Learning rate = 0.05

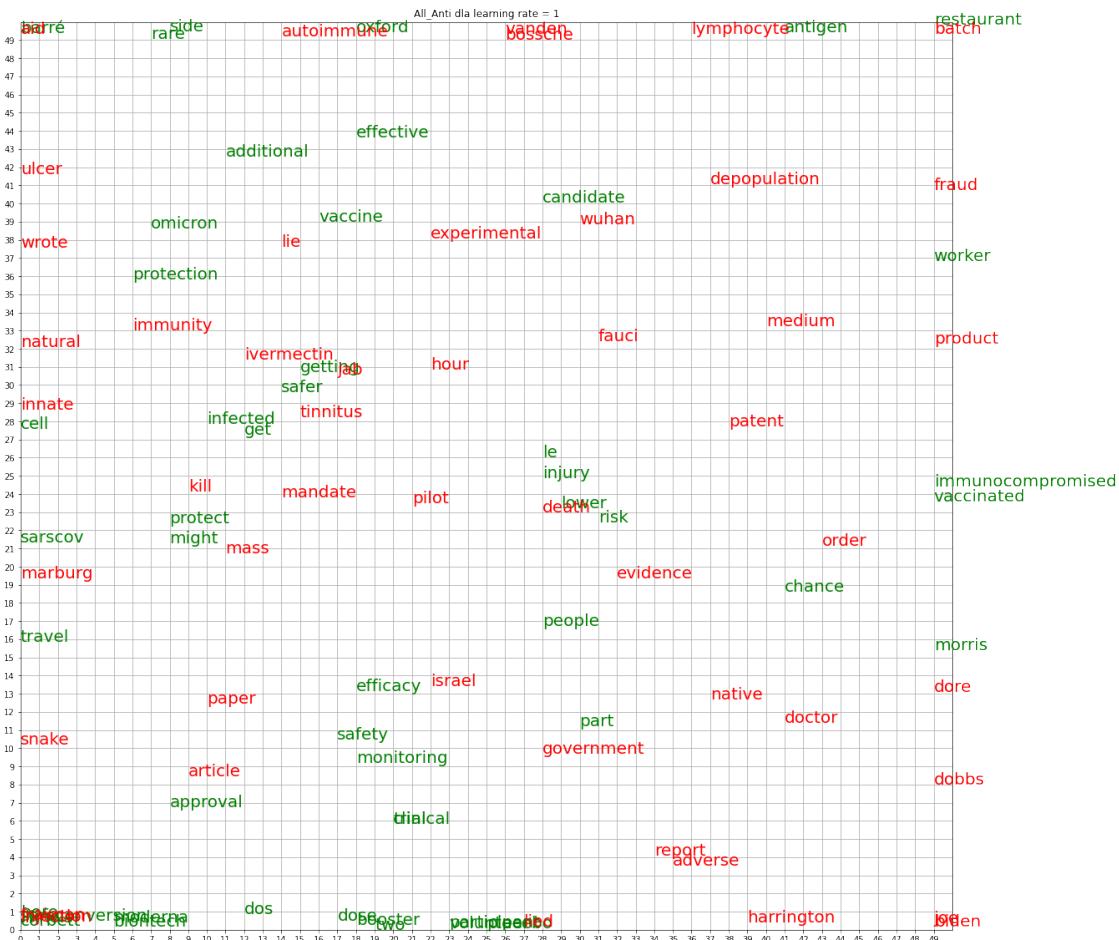


Learning rate = 0.01

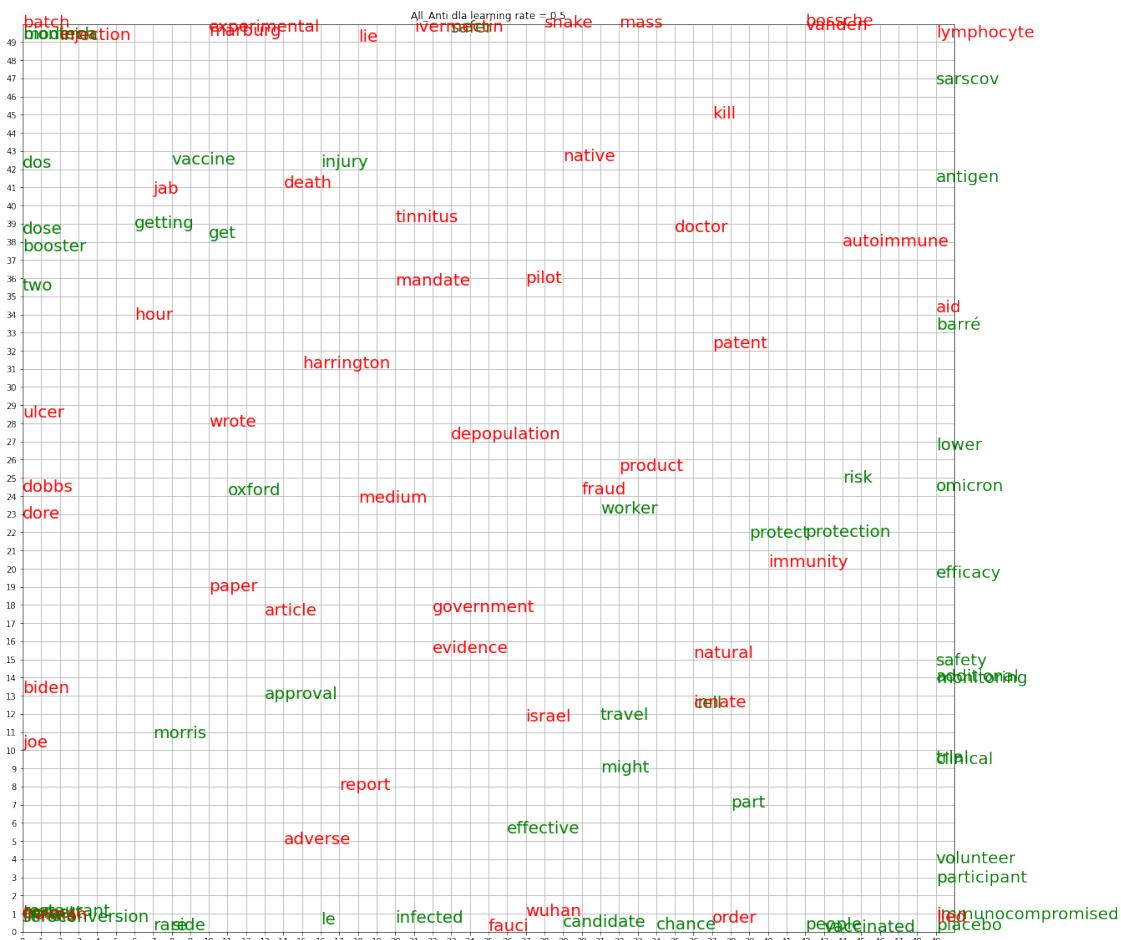


Osoba niezdecydowana zaczyna czytać artykuły przeciwko szczepieniom

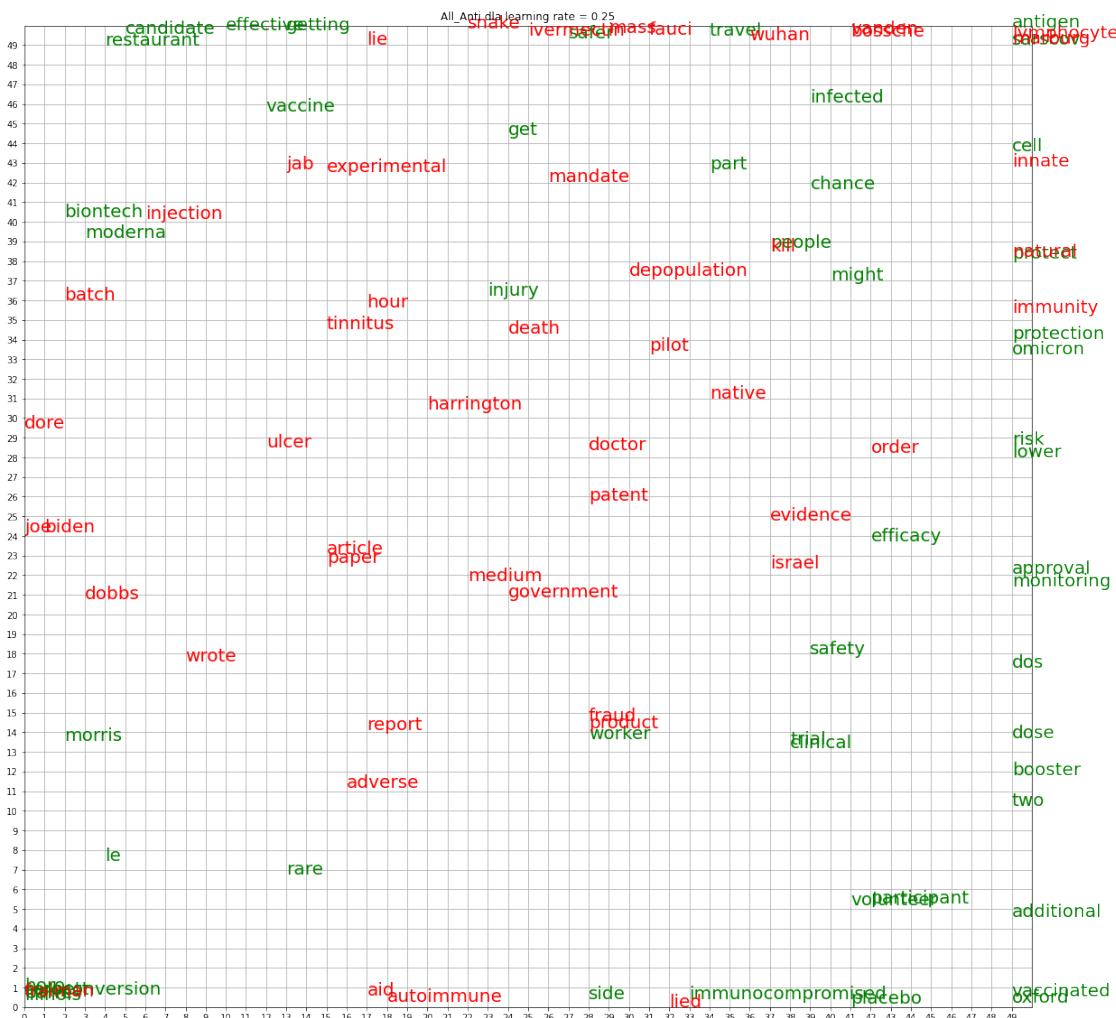
Learning rate = 1.0



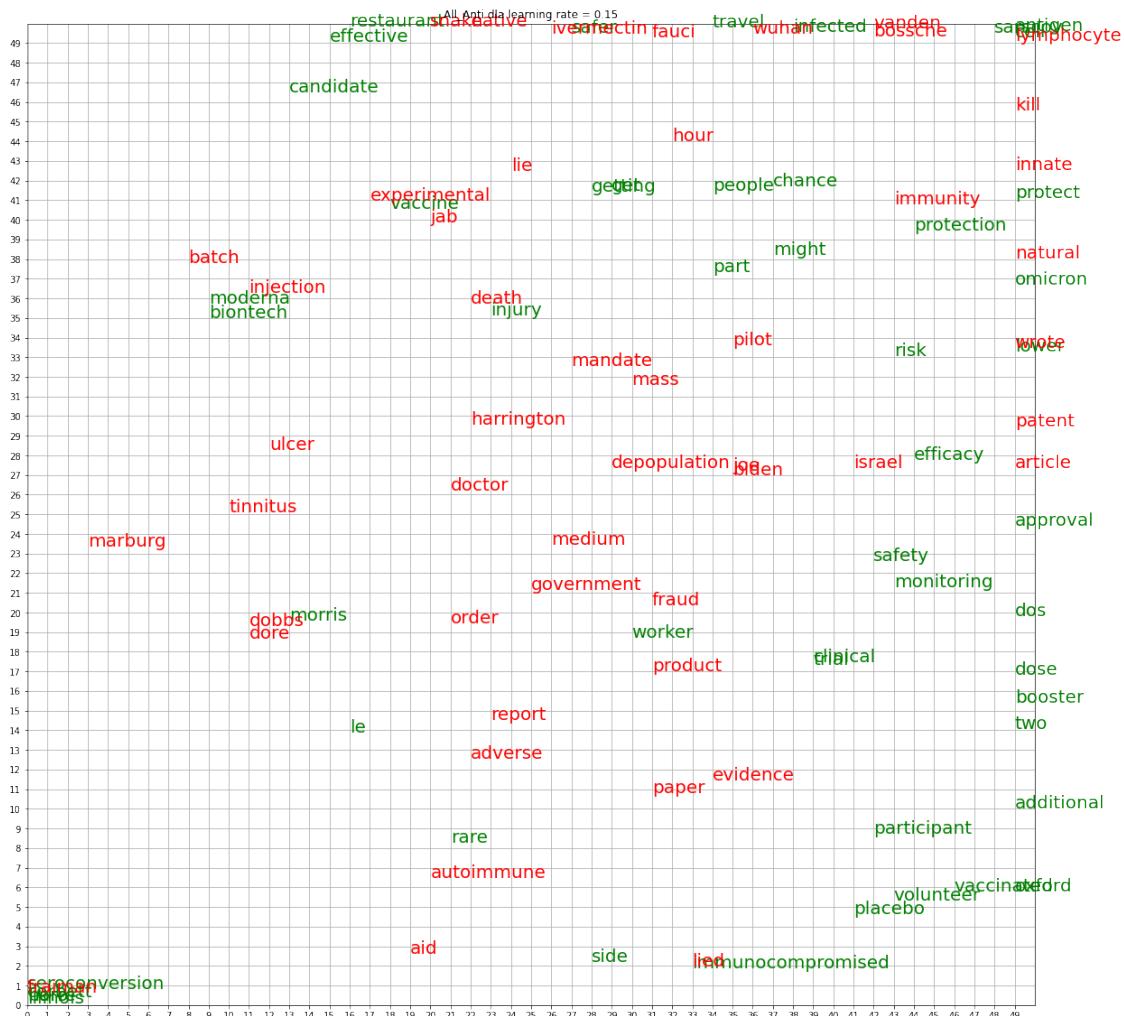
Learning rate = 0.5



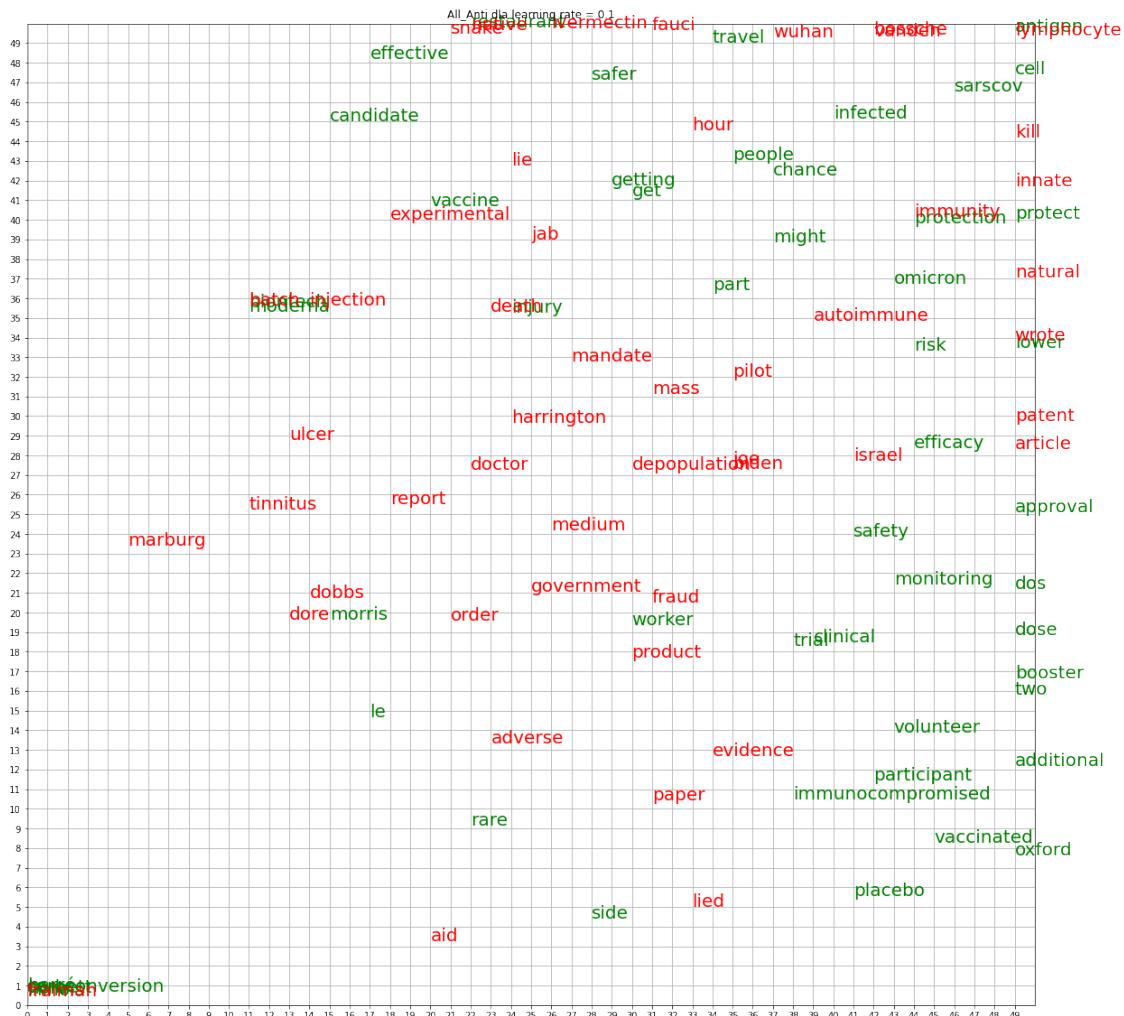
Learning rate = 0.25



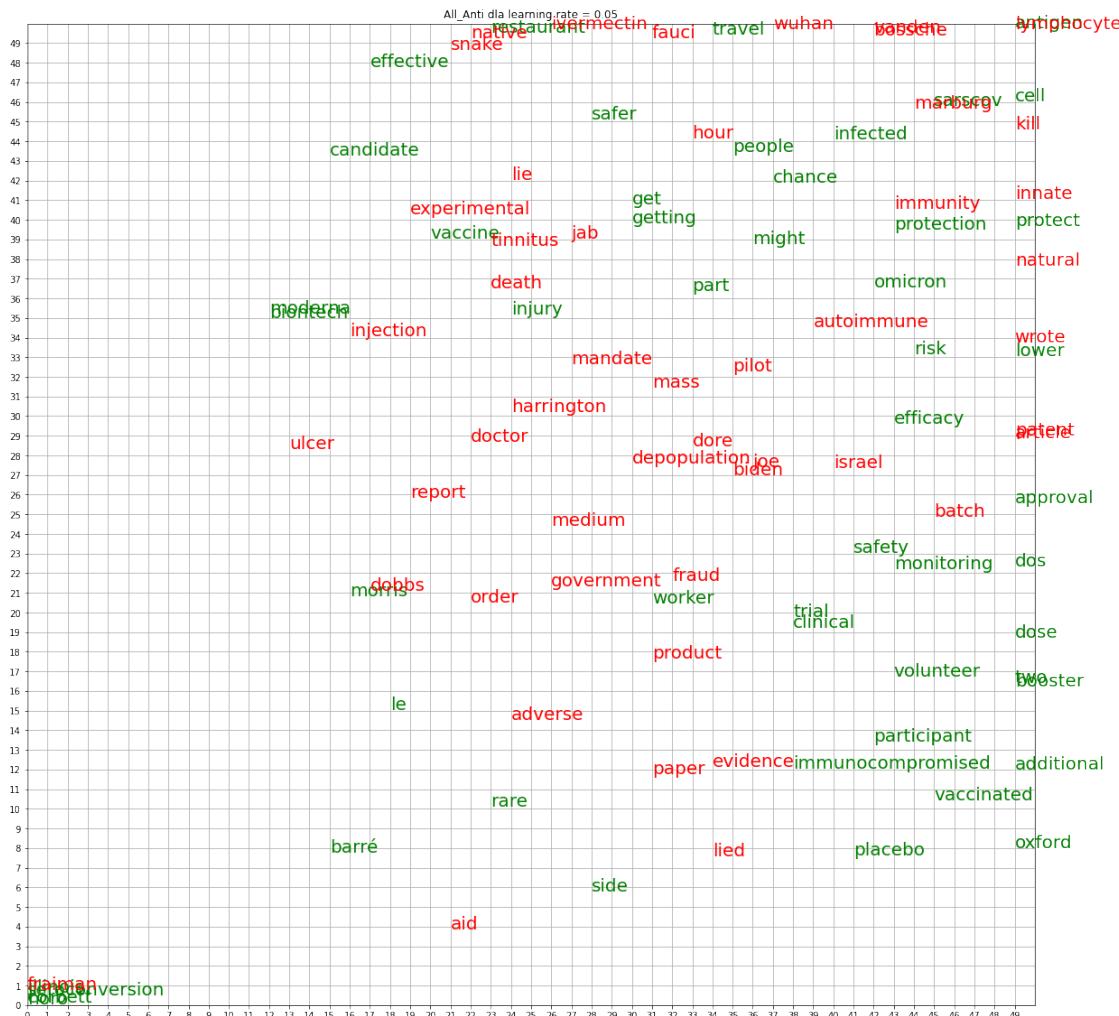
Learning rate = 0.15



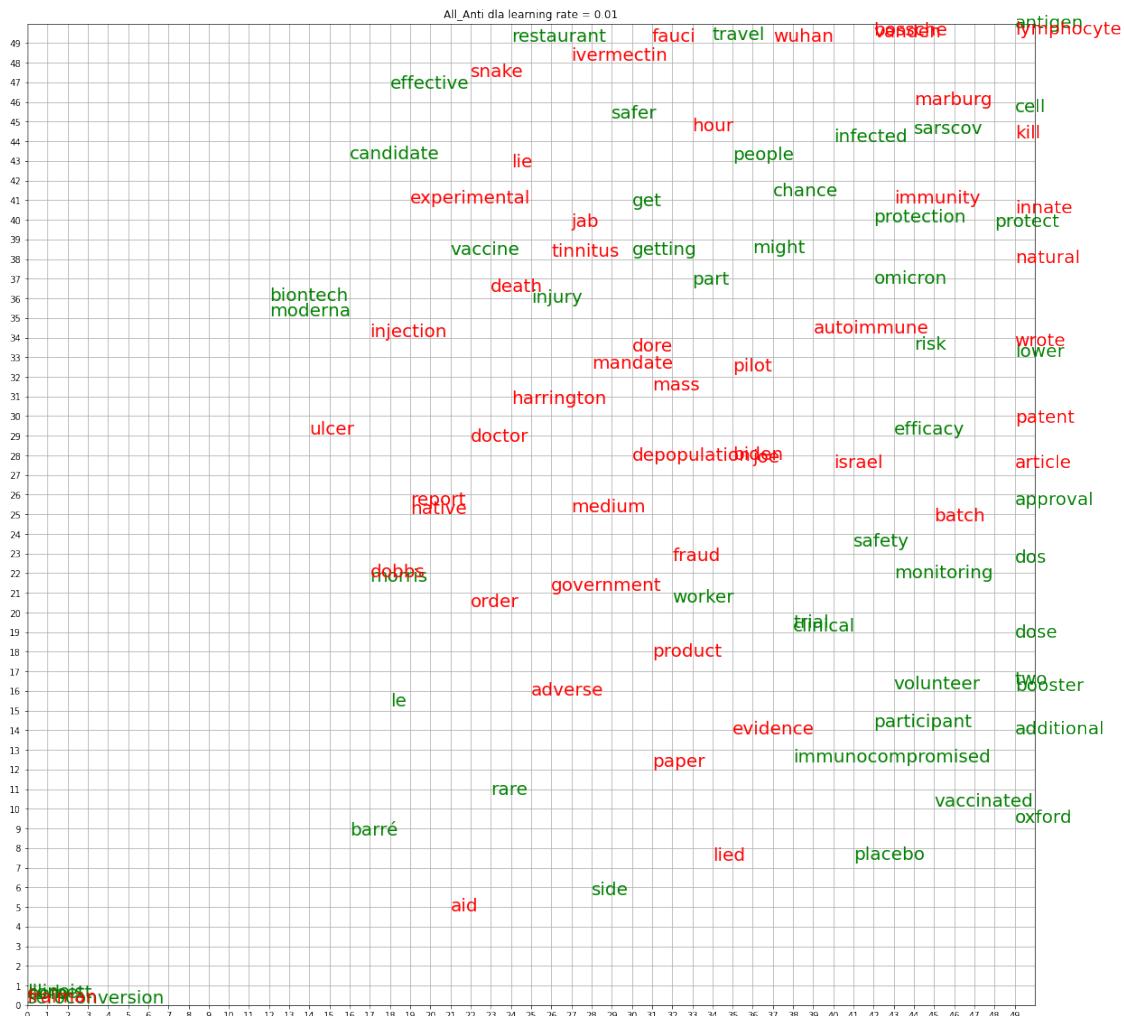
Learning rate = 0.1



Learning rate = 0.05



Learning rate = 0.01



Analiza

Analizując powyższe wykresy, można zauważać jak learning rate wpływa na zmianę opinii badanej osoby. Zbadane wartości współczynnika to 0.01, 0.05, 0.1, 0.15, 0.25, 0.5 oraz 1.

Widzimy, że dla wartości 0.01, po nakarmieniu badanego artykułami z przeciwej grupy, lub w przypadku niezdecydowanego obojętnie jakimi artykułami, słowa kluczowe zbierają się coraz bliżej siebie. Sugeruje to brak zmiany decyzji badanej osoby.

Dla współczynnika 0.05 można zauważać oddalanie się słów kluczowych i zbieranie w obrębie swoich grup. Można to interpretować jako świadome rozpoznawanie podziału idei, gdzie osoba czytając poznaje coraz to bardziej znaczące cechy dla danych grup.

W przypadku współczynnika równego 0.1 lub 0.15 najlepiej widać wpływ uczenia na zmianę stanowiska badanego. Można zauważać jak słowa kluczowe z klasy przeciwej zbliżyły się do głównego miejsca ugrupowania. Taki efekt nastąpił jednak tylko w sytuacjach gdy osoba była początkowo po którejś ze stron. Dla osób niezdecydowanych można jedynie zauważać efekt podobny do poprzedniego punktu.

Przy współczynniku większym lub równym 0.25 tracimy możliwość jednoznacznego odczytania zmian na wykresach. Cechy grupują się, jednak rozprzestrzenione są po całej mapie. Wnioskując, współczynniki równe lub powyżej 0.25 stanowią za dużą podatność na zmianę stanowiska badanego.

Rozszerzenie danych

W dalszej części projektu rozszerzyliśmy liczbę artykułów. Wykorzystaliśmy artykuły zebrane przez grupę studentów (Bogna Lew, Bartosz Strzelecki, Krzysztof Nazar), również pracujących nad modelowaniem głębokich przekonań.

Przygotowali oni 146 artykułów, 72 popierające szczepienia oraz 74 niepopierające. Teksty oraz ich źródła i przynależność do klasy zostały opisane w pliku w formacie json.

Klasyfikacja nowych artykułów

W ramach testu używanego wcześniej klasyfikatora SVM, wykonaliśmy kroswalidację z użyciem nowych danych. Dane z pliku json zostały odpowiednio przygotowane i podane do klasyfikatora.

Do otrzymania wyników wykorzystaliśmy funkcję `classification_report` z biblioteki `sklearn`. Generuje raport o wynikach klasyfikacji oparty na macierzy błędu.

Wynik opisany jest w tabeli, gdzie:

- support - liczba artykułów z klasy wybrane do testu;
- accuracy - procentowy wynik klasyfikacji;
- precision - procentowy wynik klasyfikacji względem artykułów z tej samej klasy;
- recall - procentowy udział prawidłowo przewidzianych przypadków pozytywnych wśród wszystkich przypadków pozytywnych;
- f1-score - średnia harmoniczna pomiędzy czułością (ang. recall) oraz precyzyją (ang. precision)

	precision	recall	f1-score	support
0	0.76	0.69	0.72	74
1	0.71	0.78	0.74	72
accuracy			0.73	146
macro avg	0.74	0.73	0.73	146
weighted avg	0.74	0.73	0.73	146

Jak widać klasyfikacja nowych artykułów jest na poziomie 73% co jest bardzo dobrym wynikiem.

Nauka klasyfikatora z nowymi artykułami

Następnie połączymy wszystkie artykuły do jednego pliku i podzieliliśmy je w stosunku 1:5 na dane testowe i treningowe. Dla nowych danych treningowych wytrenowaliśmy klasyfikator SVM, a następnie wykonaliśmy kroswalidację na danych treningowych, żeby sprawdzić jego skuteczność.

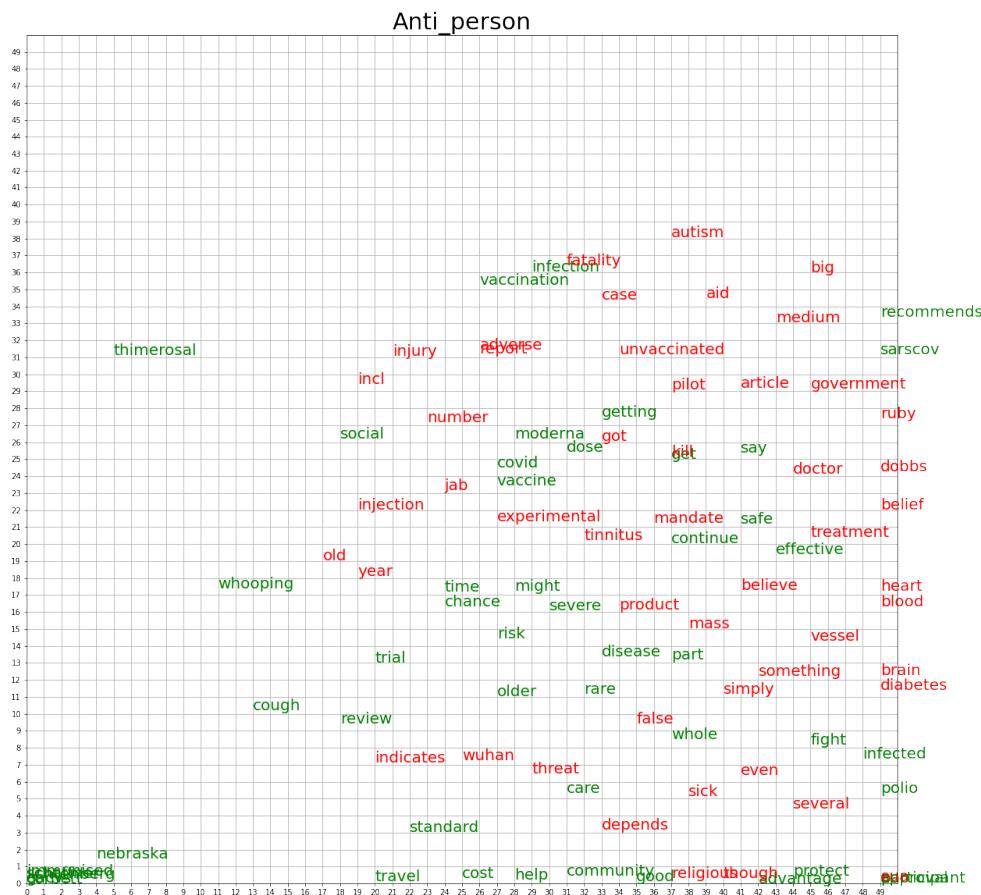
	precision	recall	f1-score	support
0	0.91	0.87	0.89	46
1	0.89	0.92	0.91	52
accuracy			0.90	98
macro avg	0.90	0.90	0.90	98
weighted avg	0.90	0.90	0.90	98

Wyniki na poziomie 90% są bardzo dobre i są na poziomie poprzedniego klasyfikatora.

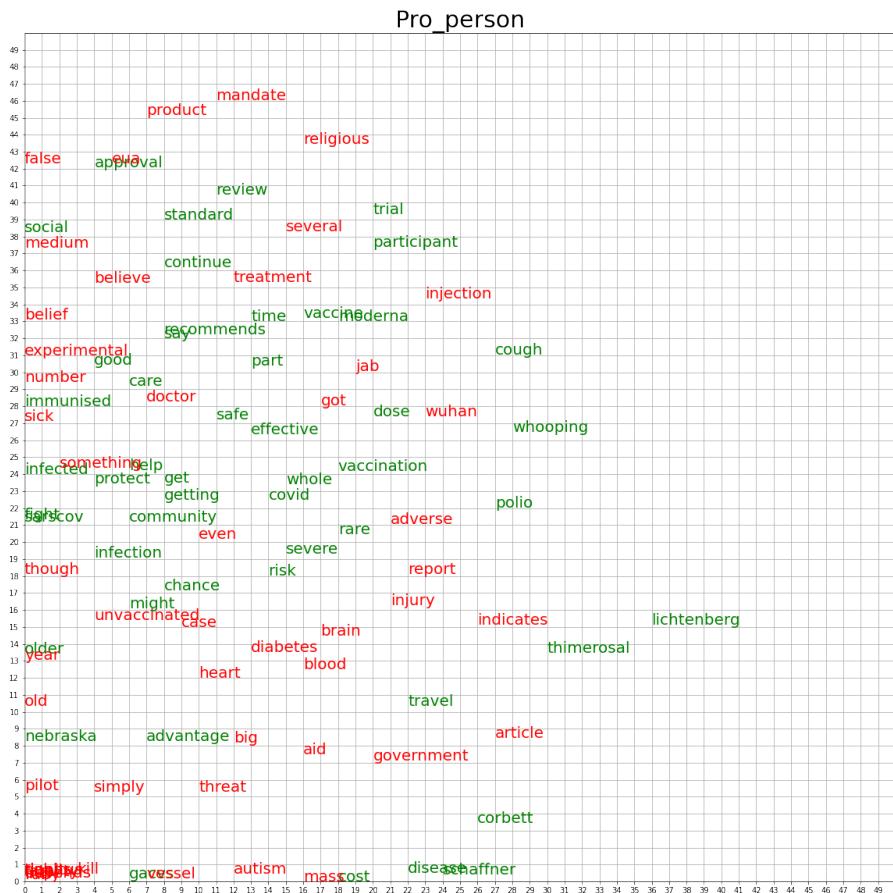
Symulacja poglądów

Jak dla poprzednich danych, przeprowadziliśmy symulację zmiany poglądów pod wpływem, czytania nowych artykułów z drugiej strony. Tak samo jak w poprzednim przypadku z pomocą mapy Kohonena, zasymulowaliśmy wstępne poglądy i wygenerowaliśmy wykresy.

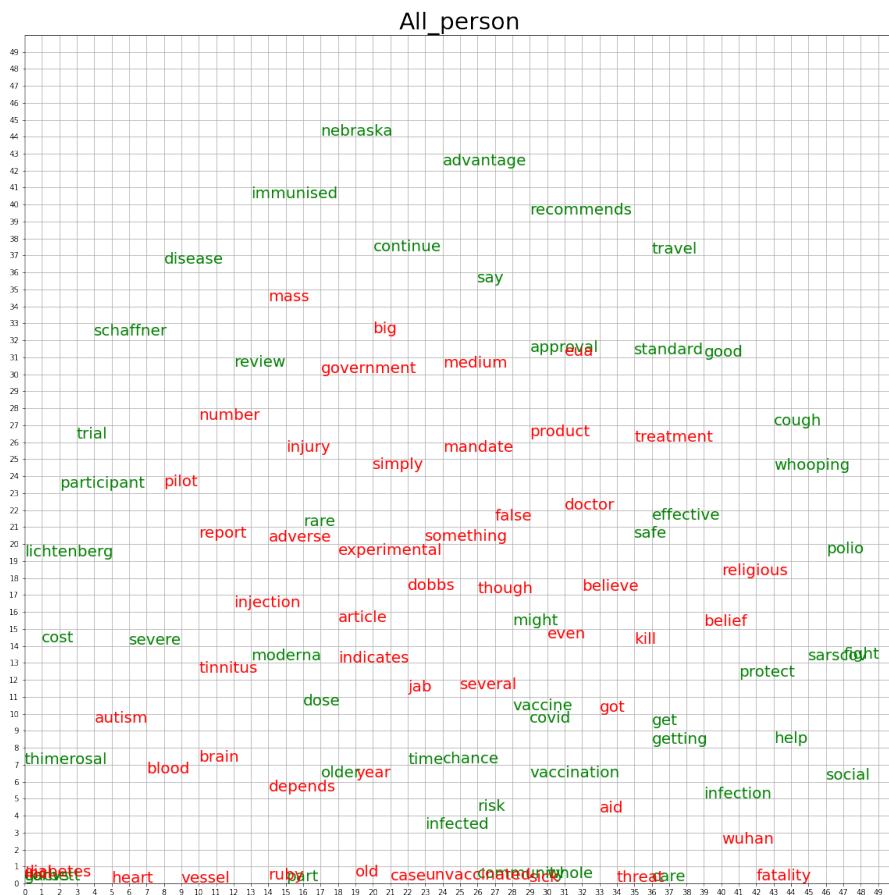
Osoba wierząca w teorię antyszczepionkową



Osoba przekonana co do skuteczności szczepionek

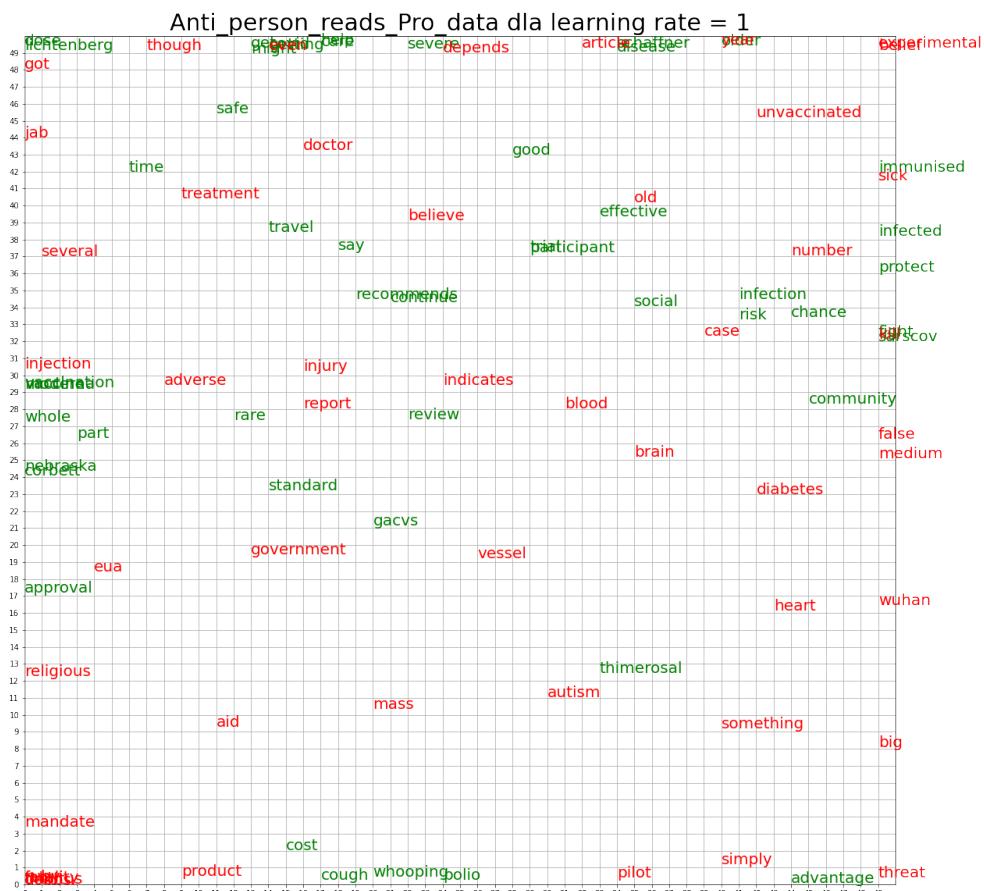


Osoba niezdecydowana

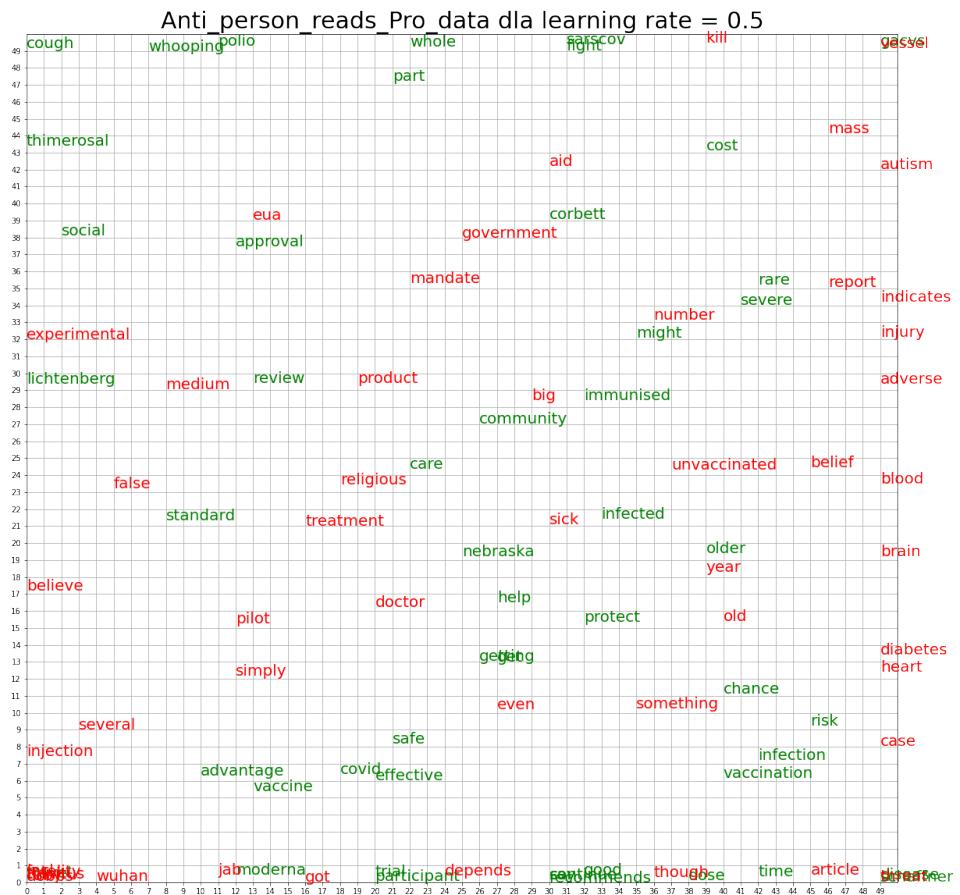


Osoba antyszczepionkowa czyta artykuły zachęcające do szczepień

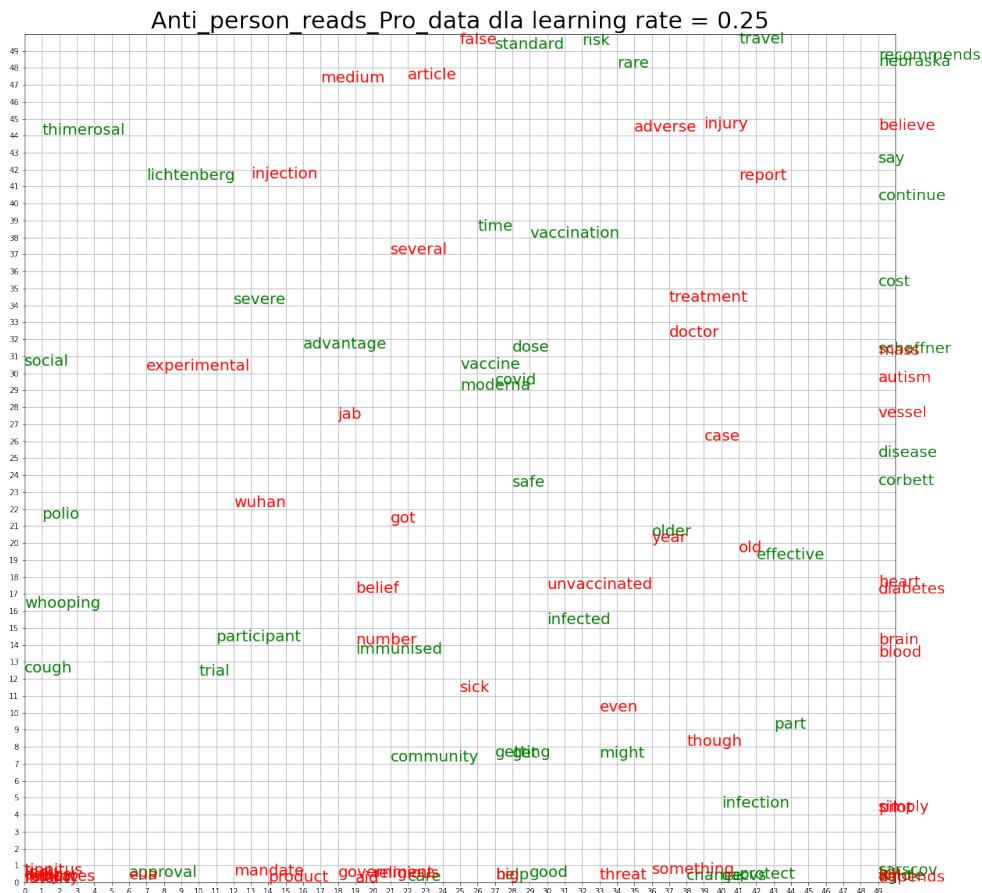
Learning rate = 1.0



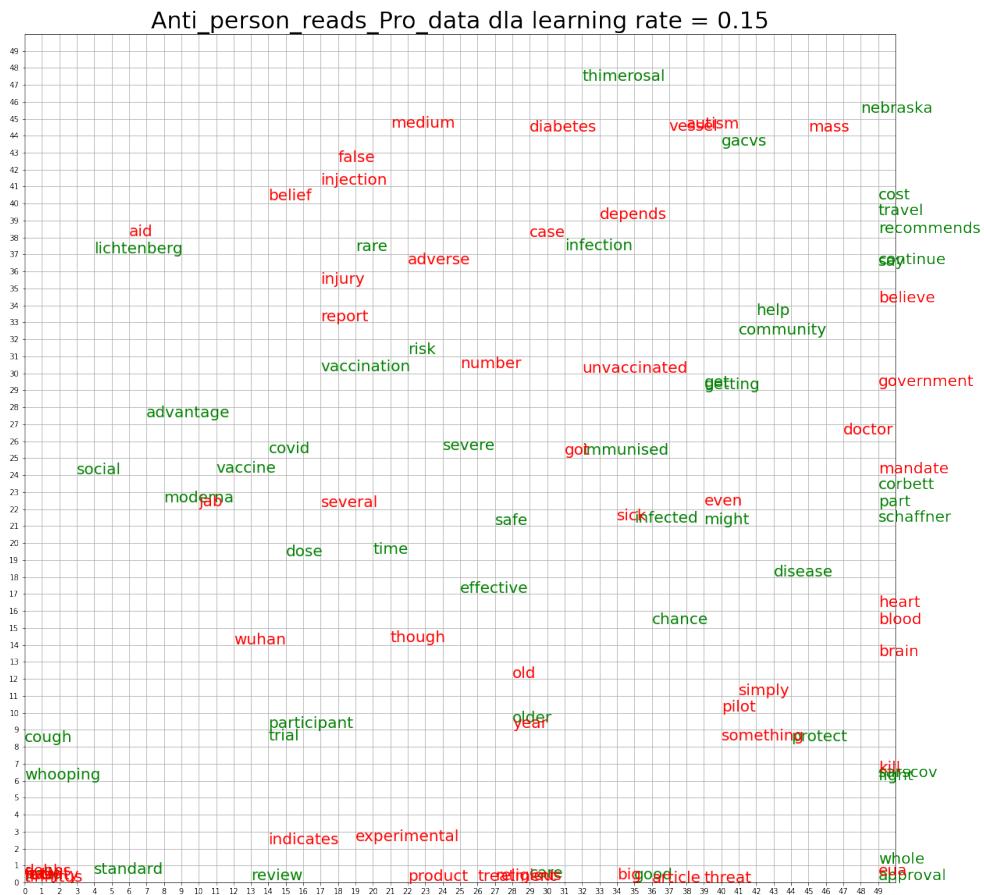
Learning rate = 0.5



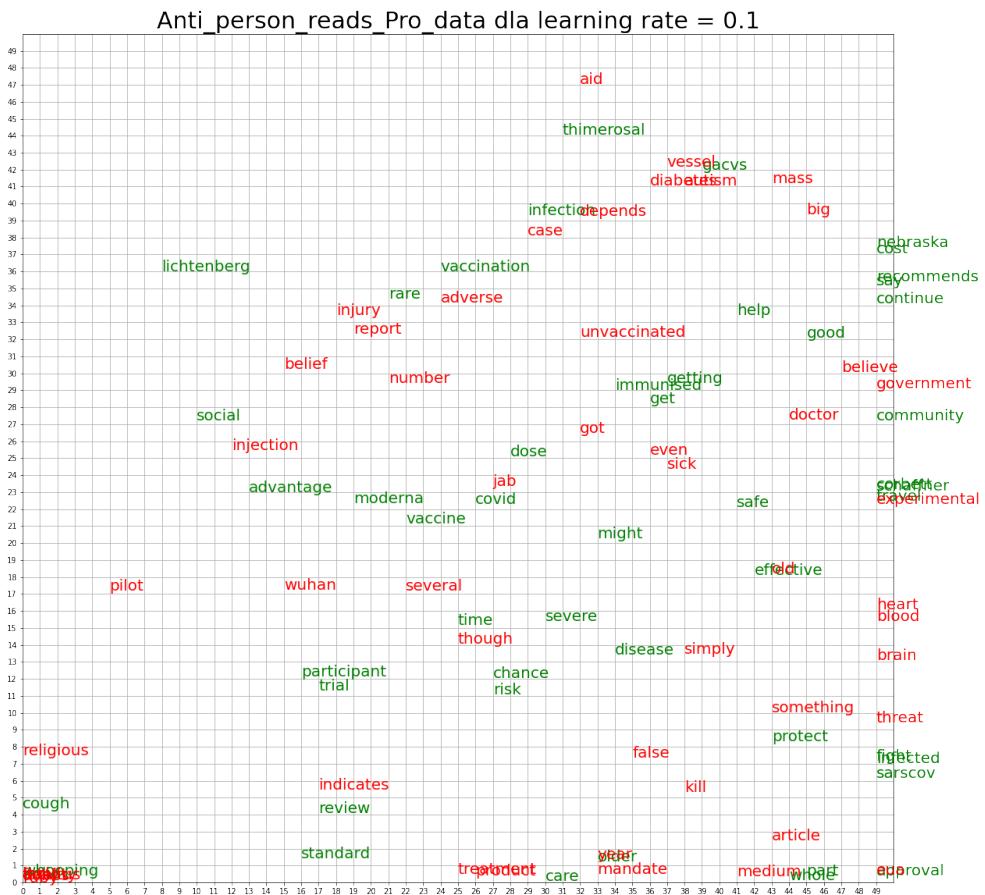
Learning rate = 0.25



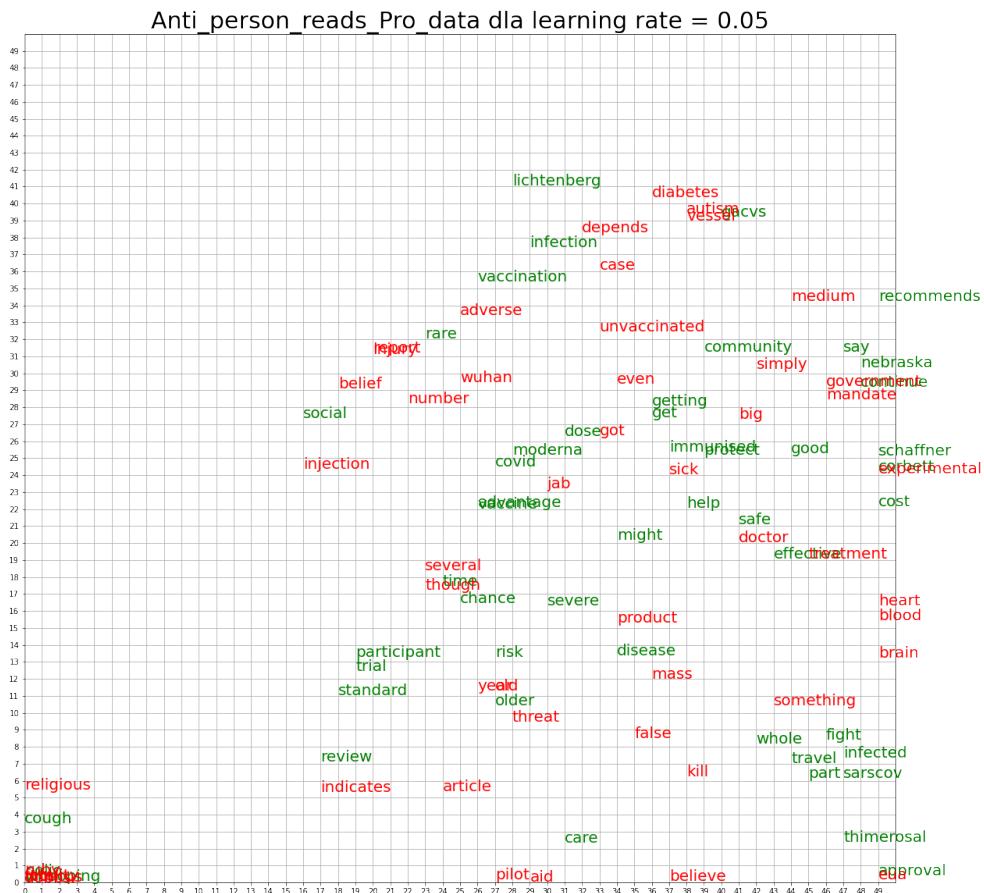
Learning rate = 0.15



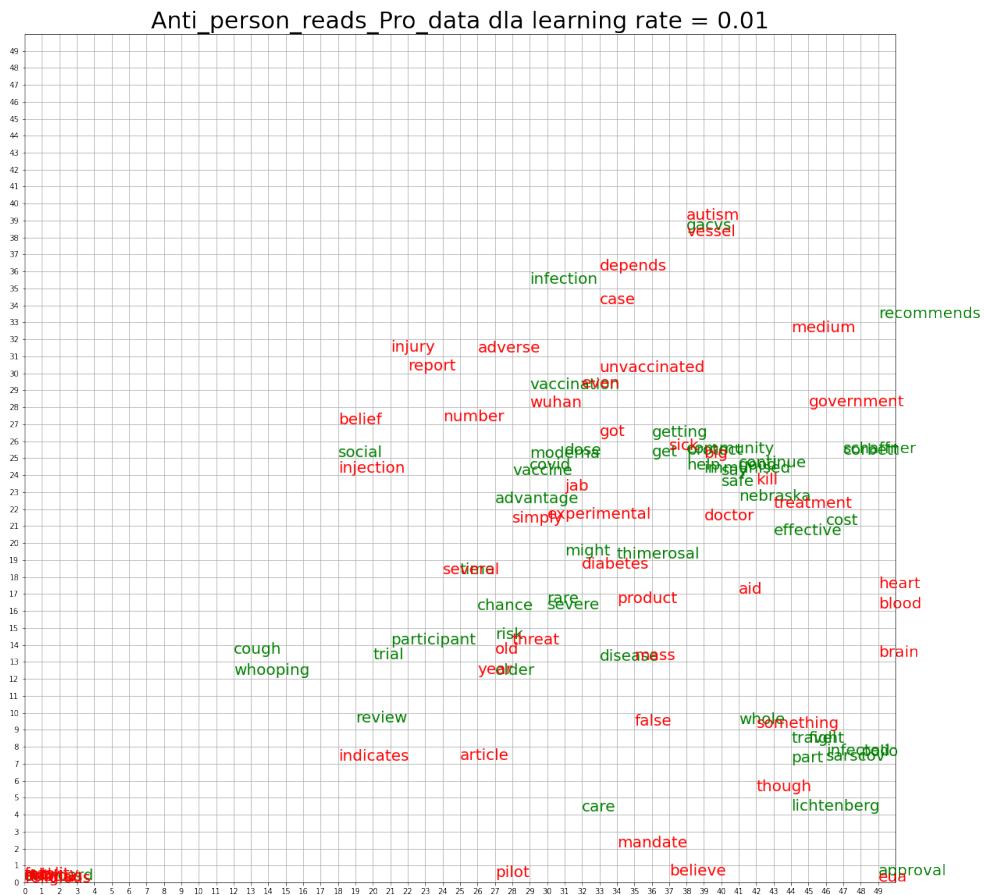
Learning rate = 0.1



Learning rate = 0.05



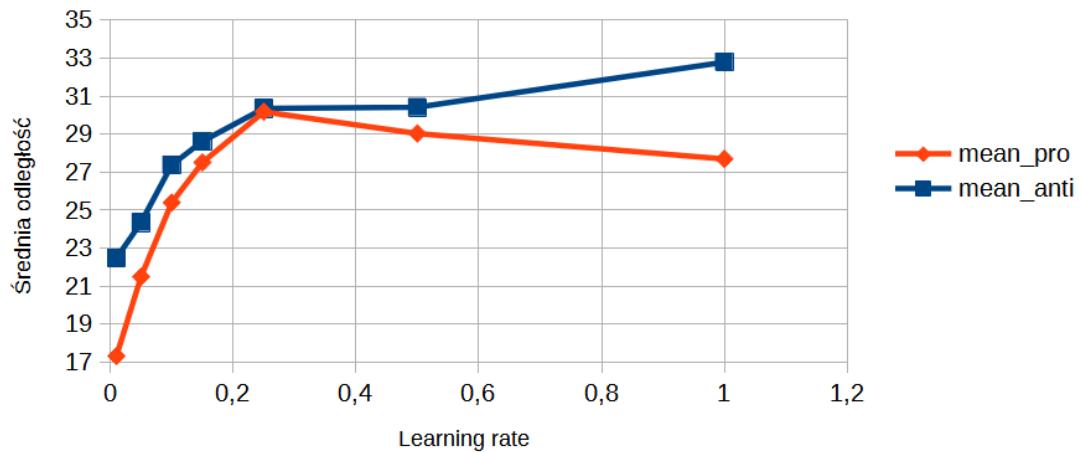
Learning rate = 0.01



LR\Distans	Suma pro	Średnia pro	Suma anty	Średnia anty
knowledge	56590.45340999222	23.30743550658658	45377.308179431304	18.536482099440892
0.01	42282.00271743428	17.32868963829274	54406.54040667965	22.48204149036349
0.05	52585.69153759424	21.49864739885292	59070.58162218958	24.348961921759926
0.1	61977.459945613096	25.400598338366024	66539.31805258655	27.382435412587057
0.15	67247.97667582556	27.515538738062837	69563.36986668759	28.626901179706827
0.25	73707.8329348273	30.18338777019954	73268.13708249846	30.35134096209547
0.5	70795.34318944887	29.038286788125053	73906.79256944568	30.41431792981303
1	67262.24051919853	27.702734974958208	79620.51050952755	32.79263200557148

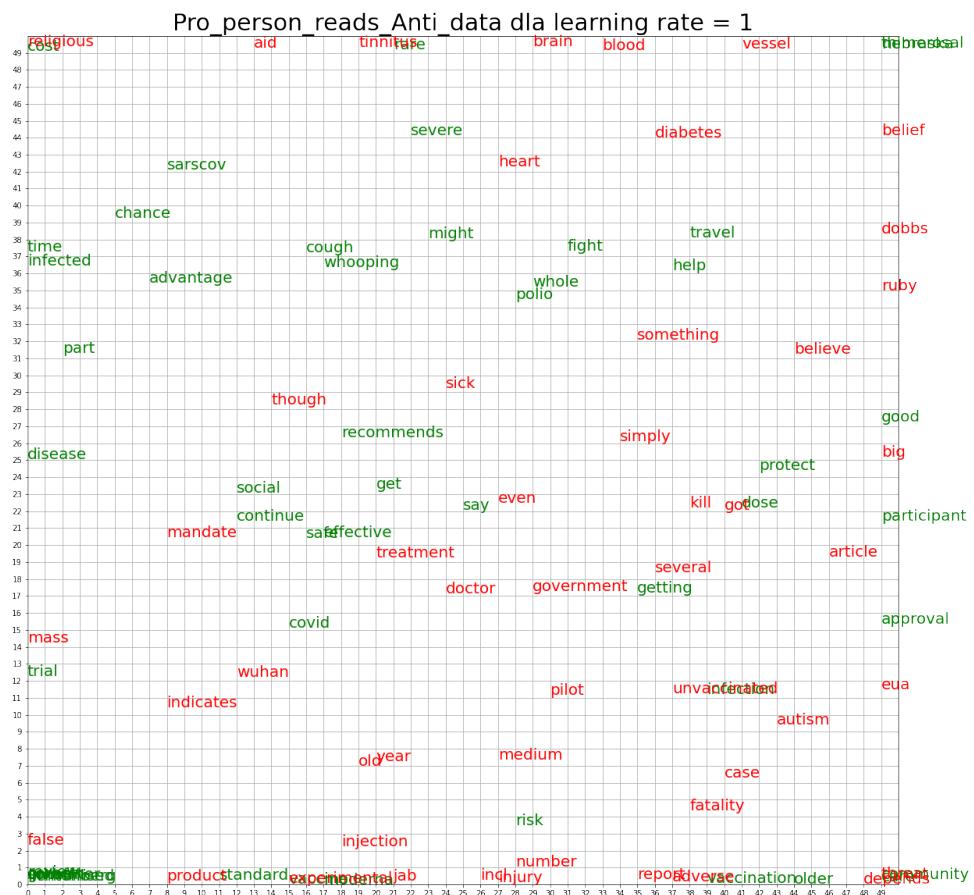
Wykres zależności odległości między słowami od learning rate

Osoba antyszczepionkowa - artykuły proszczepionkowe

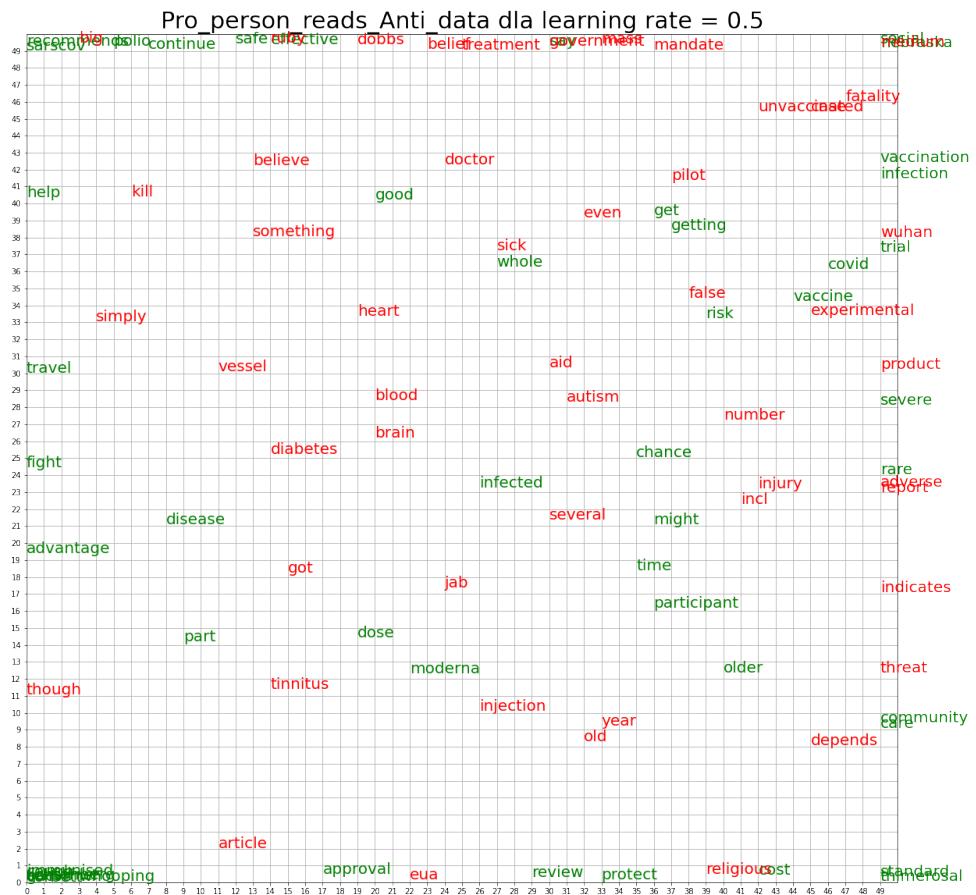


Osoba przekonana co do skuteczności szczepionek czyta artykuły przeciwko szczepieniom

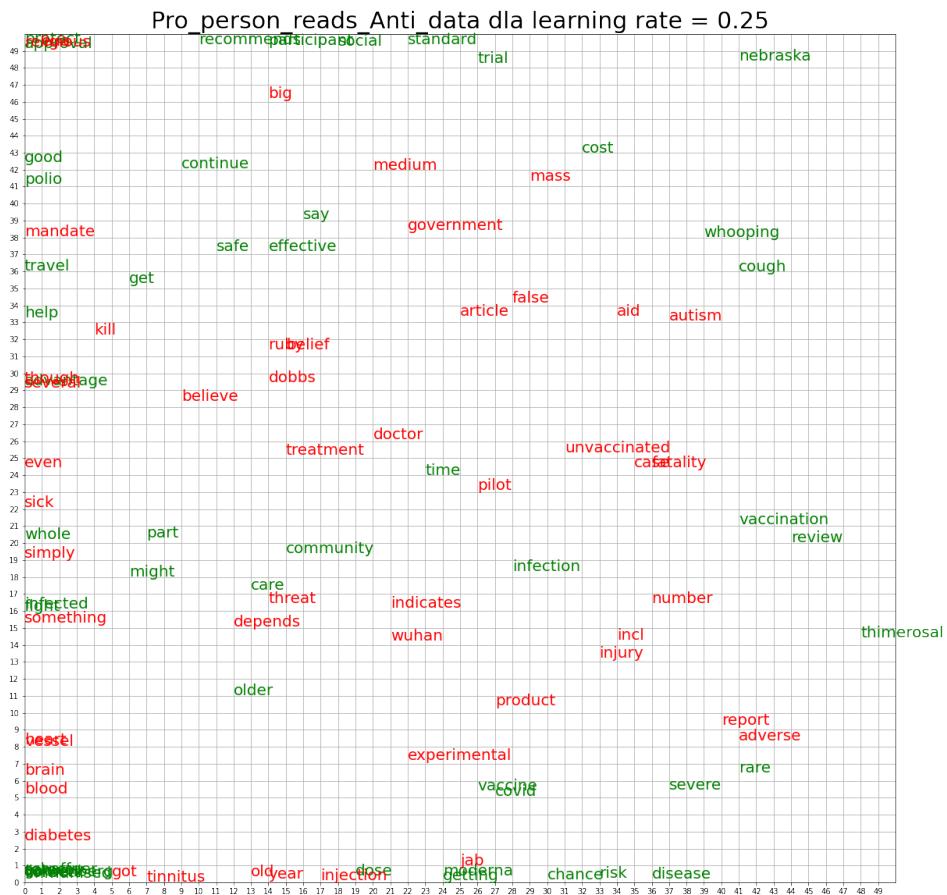
Learning rate = 1.0



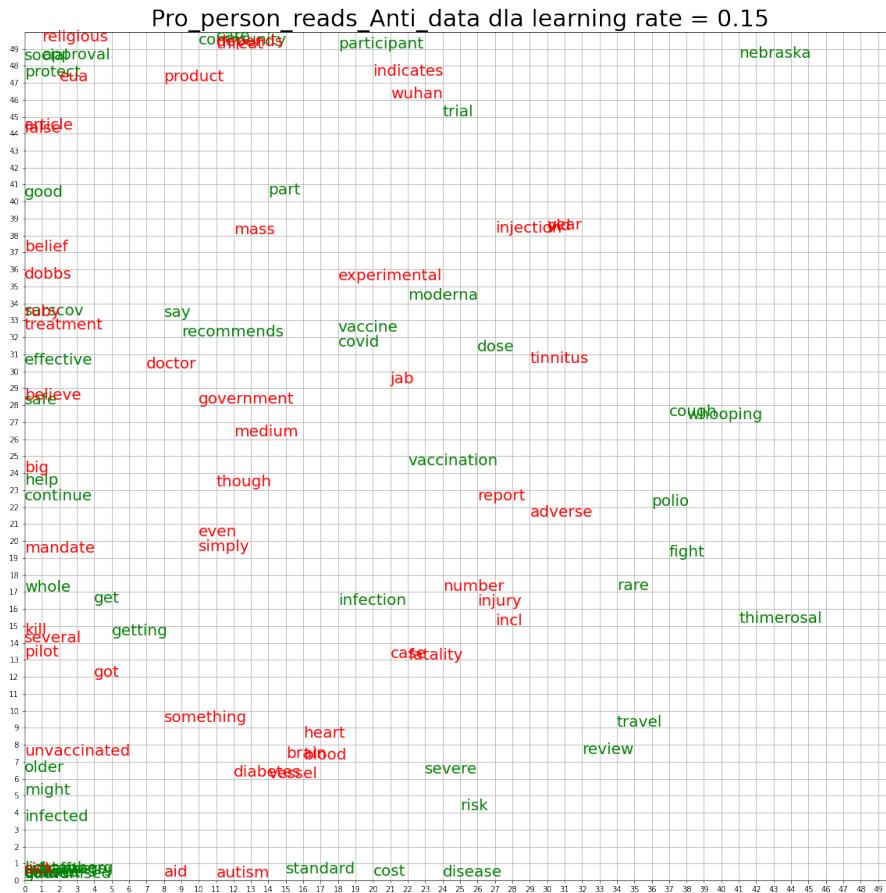
Learning rate = 0.5



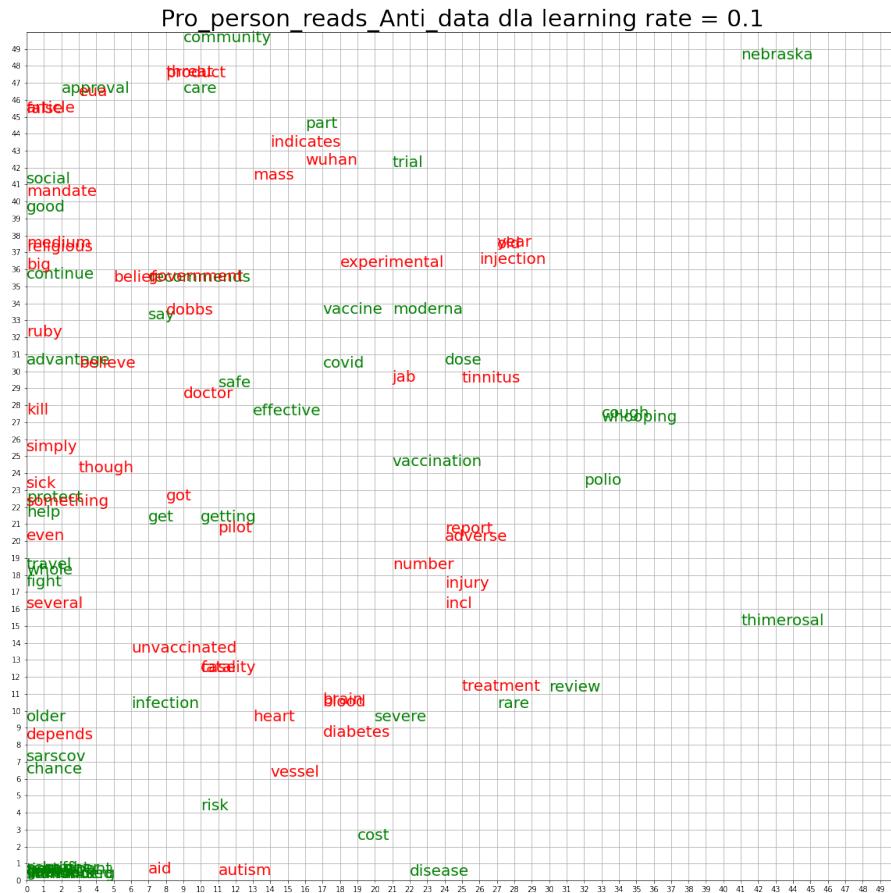
Learning rate = 0.25



Learning rate = 0.15



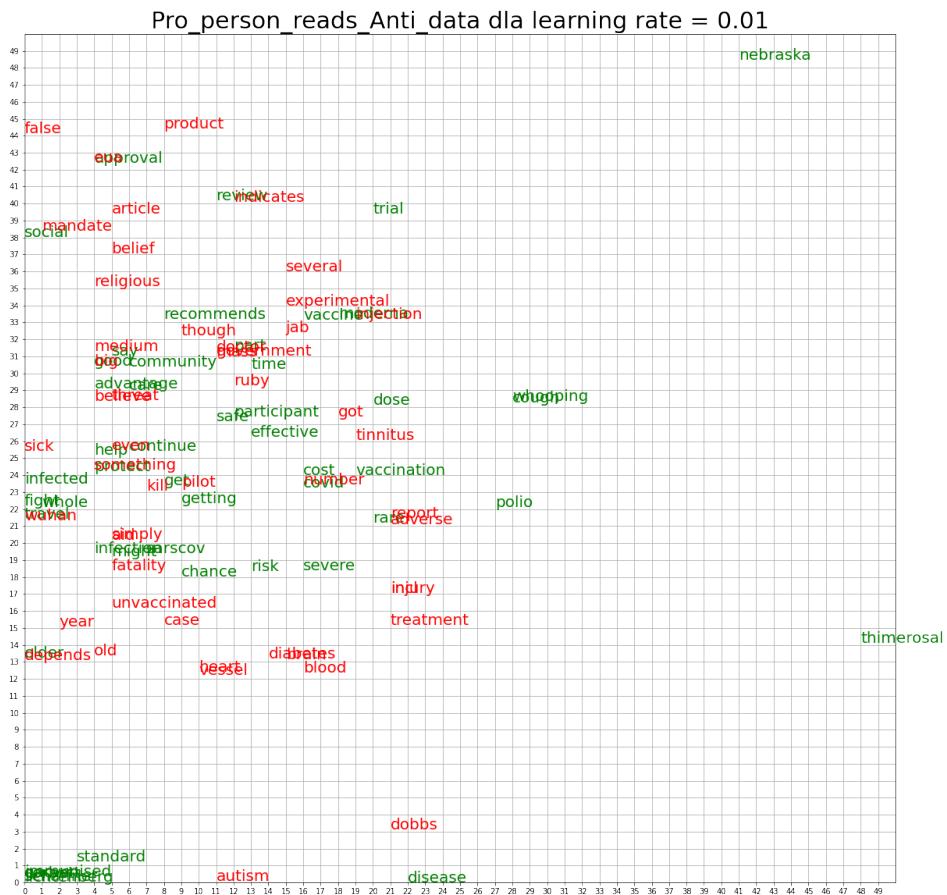
Learning rate = 0.1



Learning rate = 0.05



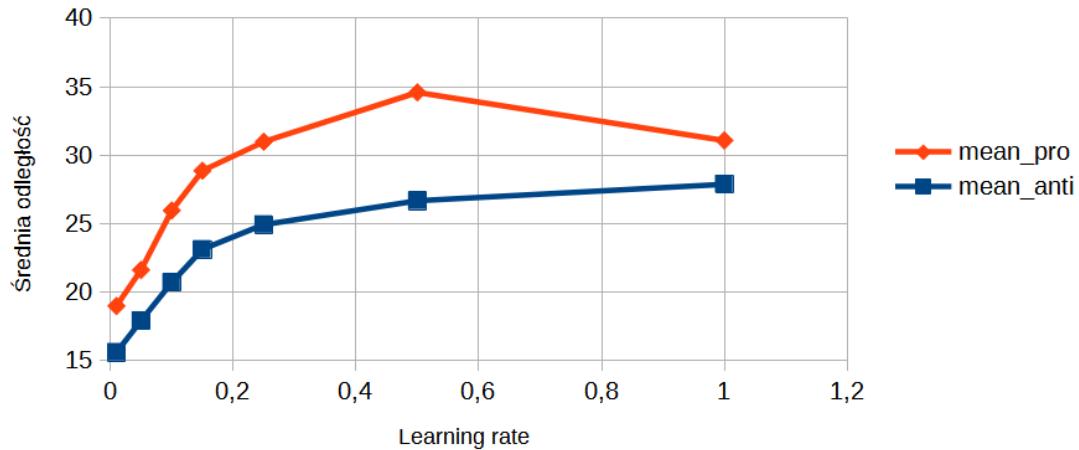
Learning rate = 0.01



LR \ Dystans	Suma pro	Średnia pro	Suma anty	Średnia anty
knowledge	44275.98606196462	18.101384326232473	52085.05710646814	21.522750870441385
0.01	46041.24525783824	18.962621605369954	37965.335689734755	15.585113173125926
0.05	52291.93554093425	21.590394525571533	43812.75380794481	17.89736675161144
0.1	61595.71859463916	25.945964024700572	50375.50254770284	20.679598747004448
0.15	69067.93527592118	28.85043244608236	56400.792566971686	23.096147652322557
0.25	74764.89740261331	30.97137423471968	60929.809679726444	24.90997942752512
0.5	83389.80157446182	34.57288622490125	65230.95085177347	26.646630249907464
1	75053.45676996151	31.06517250412315	68255.3766669517	27.85933741497762

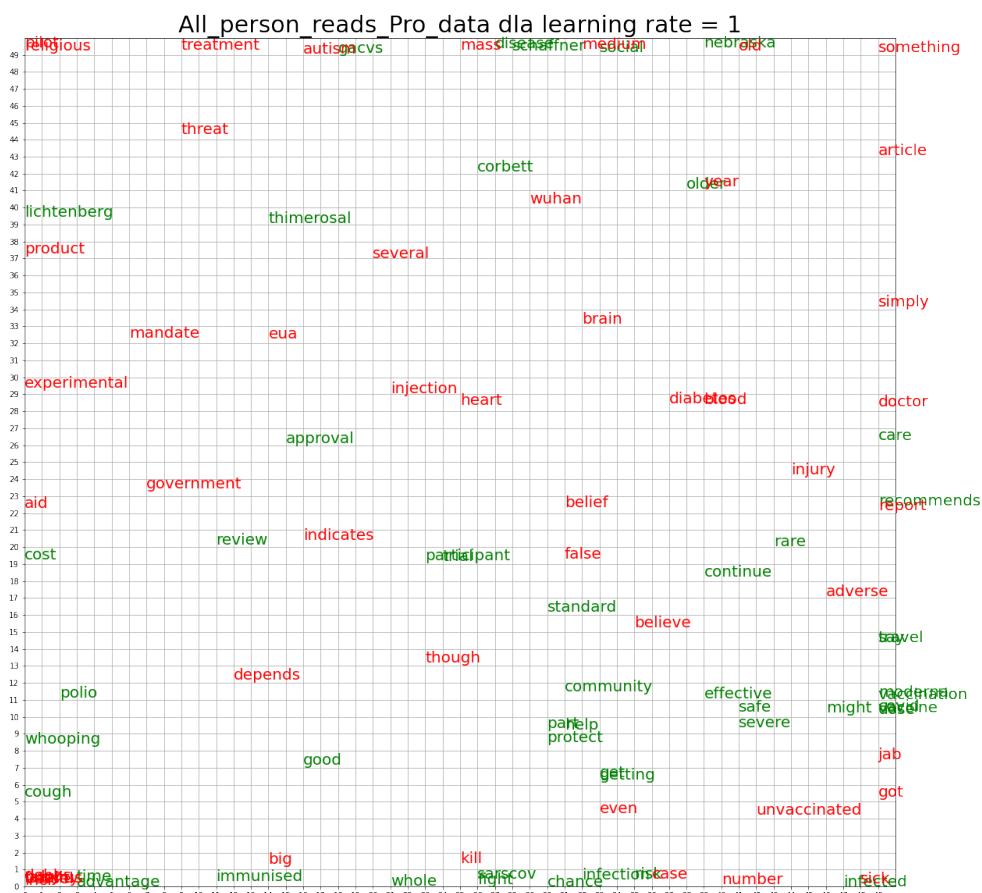
Wykres zależności odległości między słowami od learning rate

Osoba proszczepionkowa - artykuły antyszczepionkowe

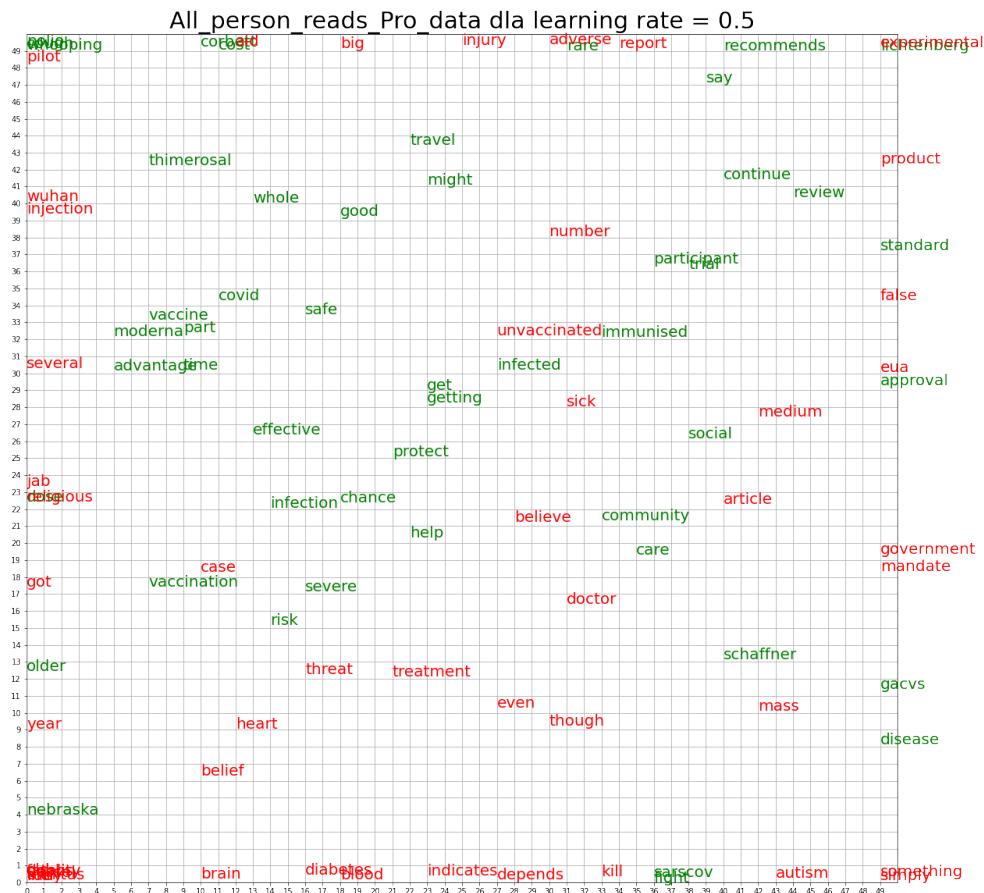


Osoba niezdecydowana czyta artykuły zachęcające do szczepień

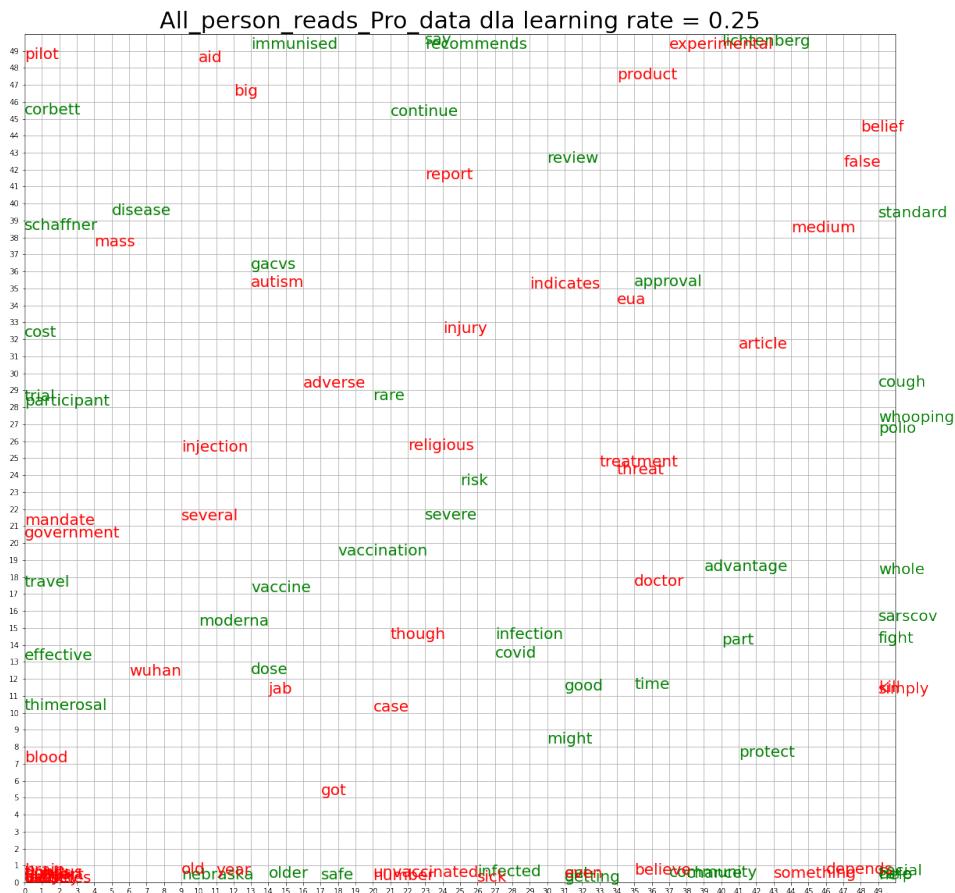
Learning rate = 1.0



Learning rate = 0.5



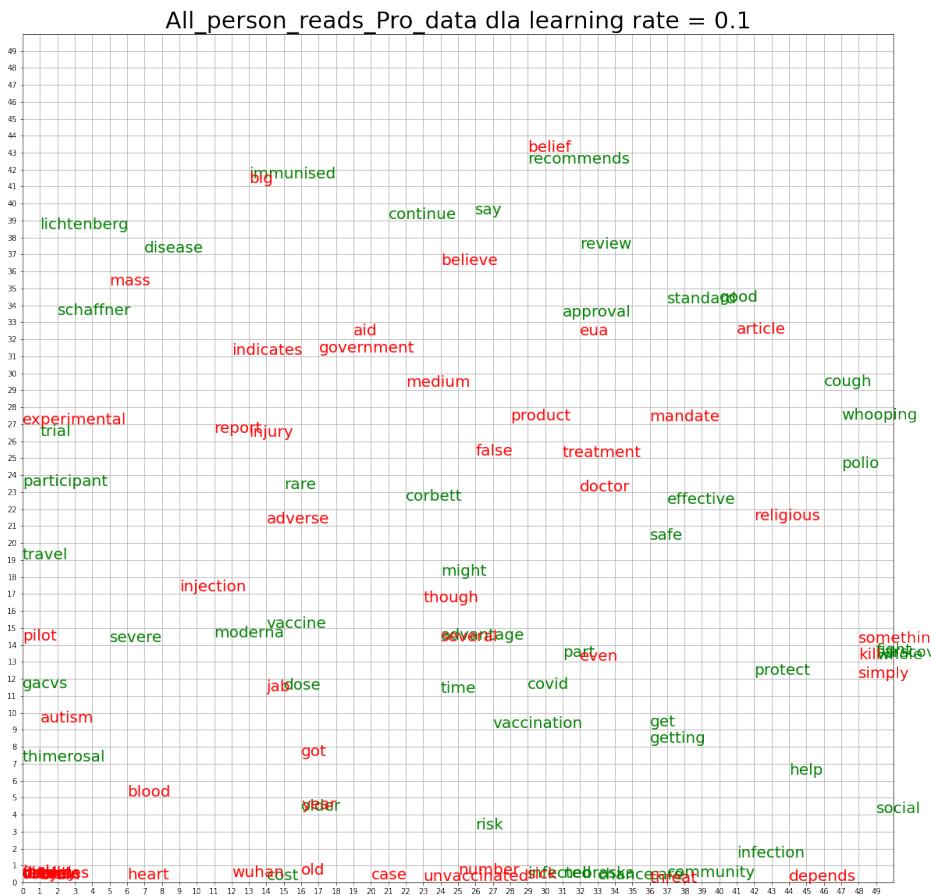
Learning rate = 0.25



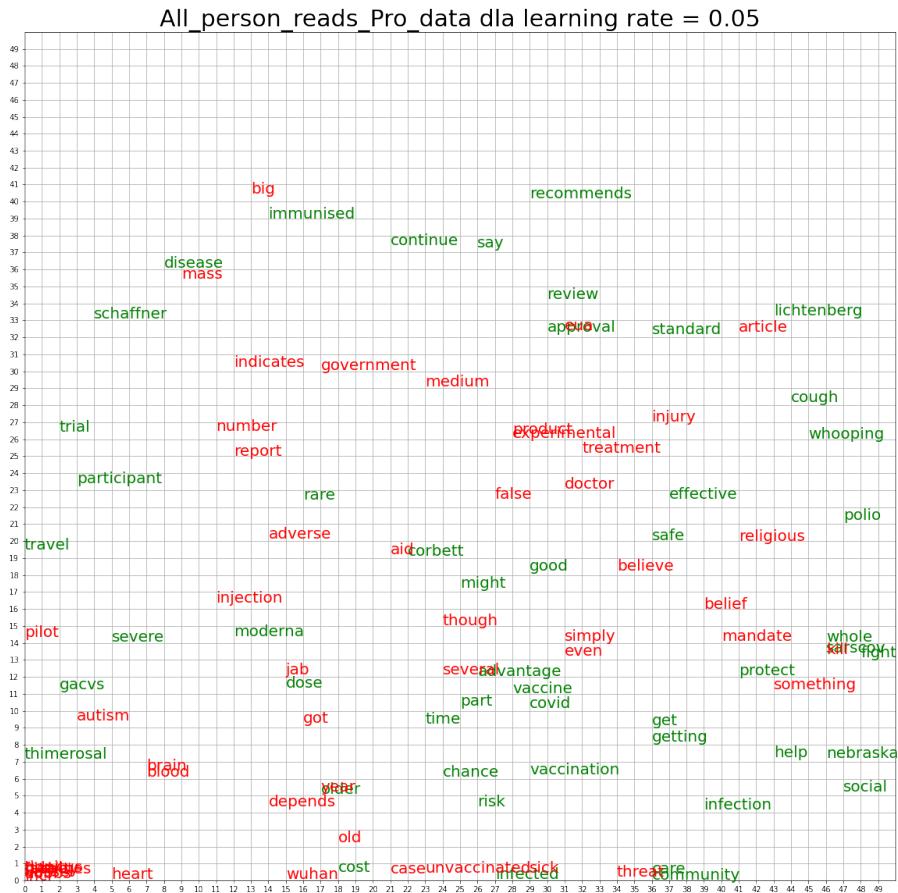
Learning rate = 0.15



Learning rate = 0.1

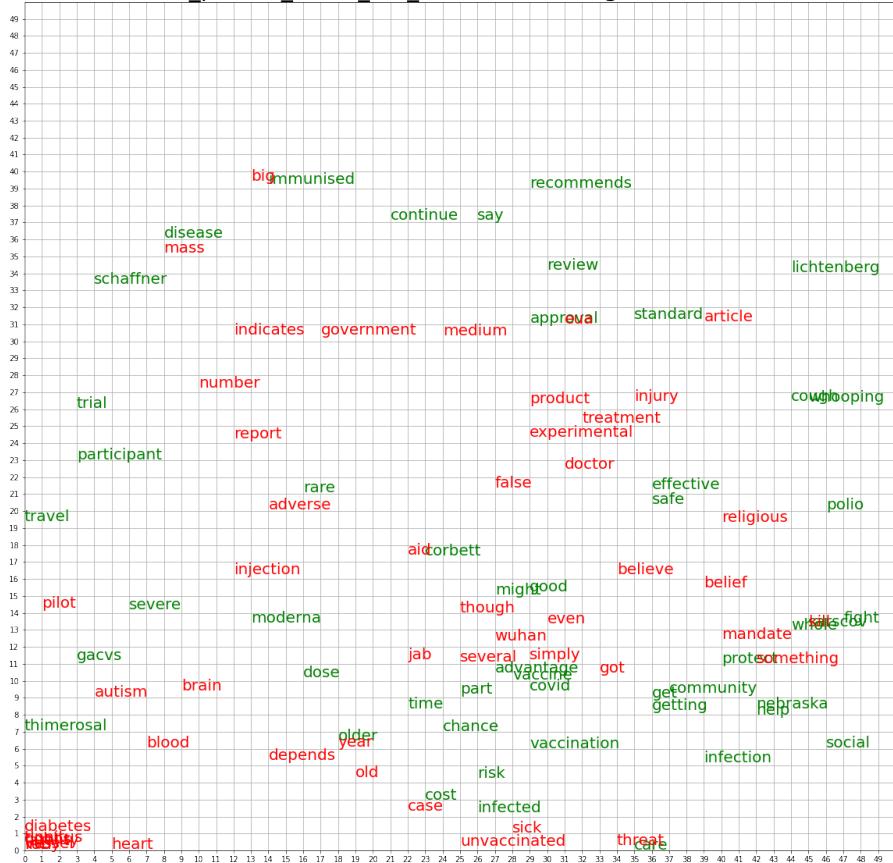


Learning rate = 0.05



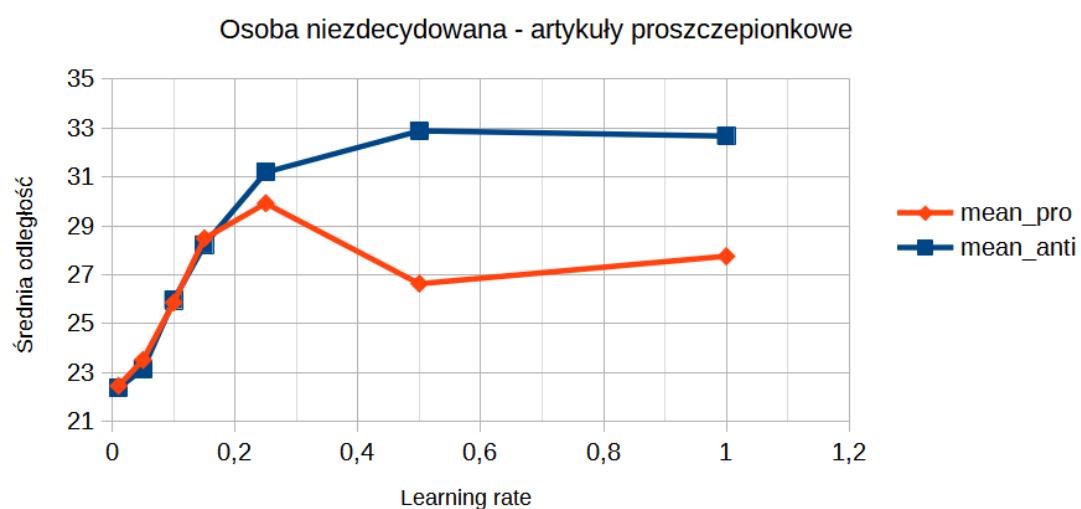
Learning rate = 0.01

All_person_reads_Pro_data dla learning rate = 0.01



LR\ Dystans	Suma pro	Średnia pro	Suma anty	Średnia anty
knowledge	61615.14261972499	25.169584403482432	47325.92368544548	19.332485165623154
0.01	54956.03676972103	22.449361425539635	54102.80470847016	22.356530871268664
0.05	57550.65394364968	23.5092540619484	55621.07807172379	23.136887717023207
0.1	63198.45574760151	25.858615281342676	62497.50973970104	25.954115340407412
0.15	69603.15901237895	28.47919763190628	67560.53621581351	28.22077536165978
0.25	72946.25105182905	29.920529553662448	74549.67317847154	31.192331873837464
0.5	65034.88251621971	26.63181102220299	79110.23775915914	32.88039807113845
1	67542.90160767647	27.749754152701918	78995.10740970238	32.669606042060536

Wykres zależności odległości między słowami od learning rate

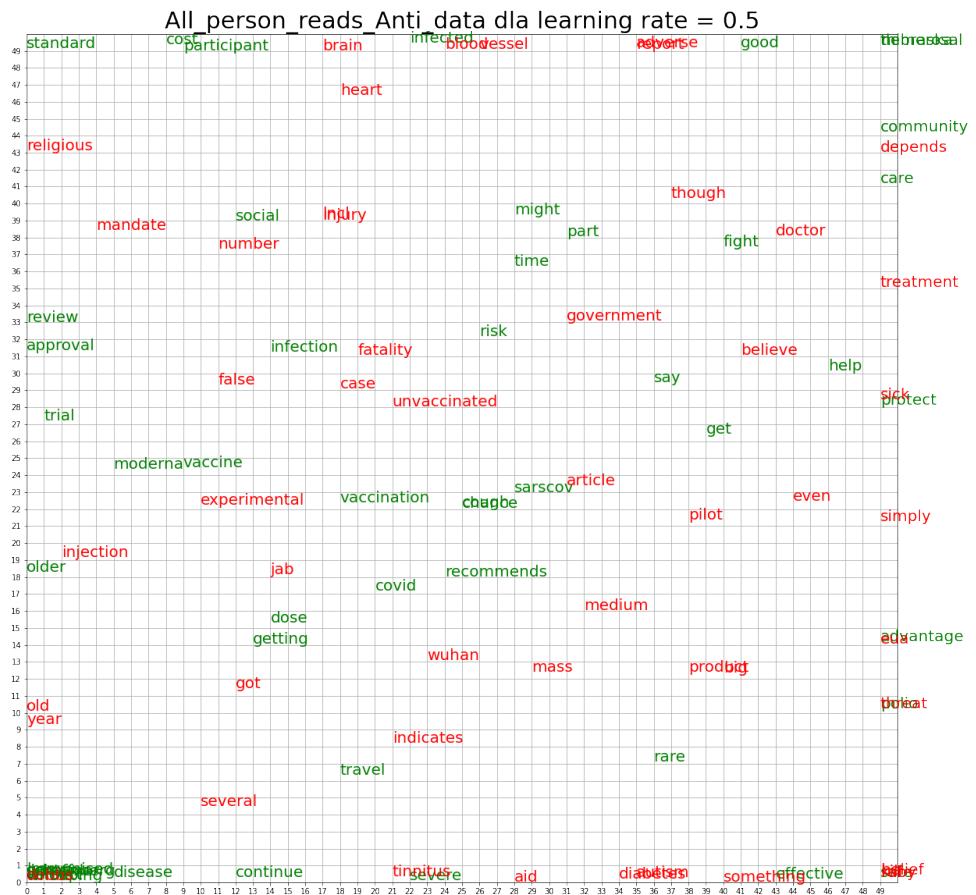


Osoba niezdecydowana zaczyna czytać artykuły przeciwko szczepieniom

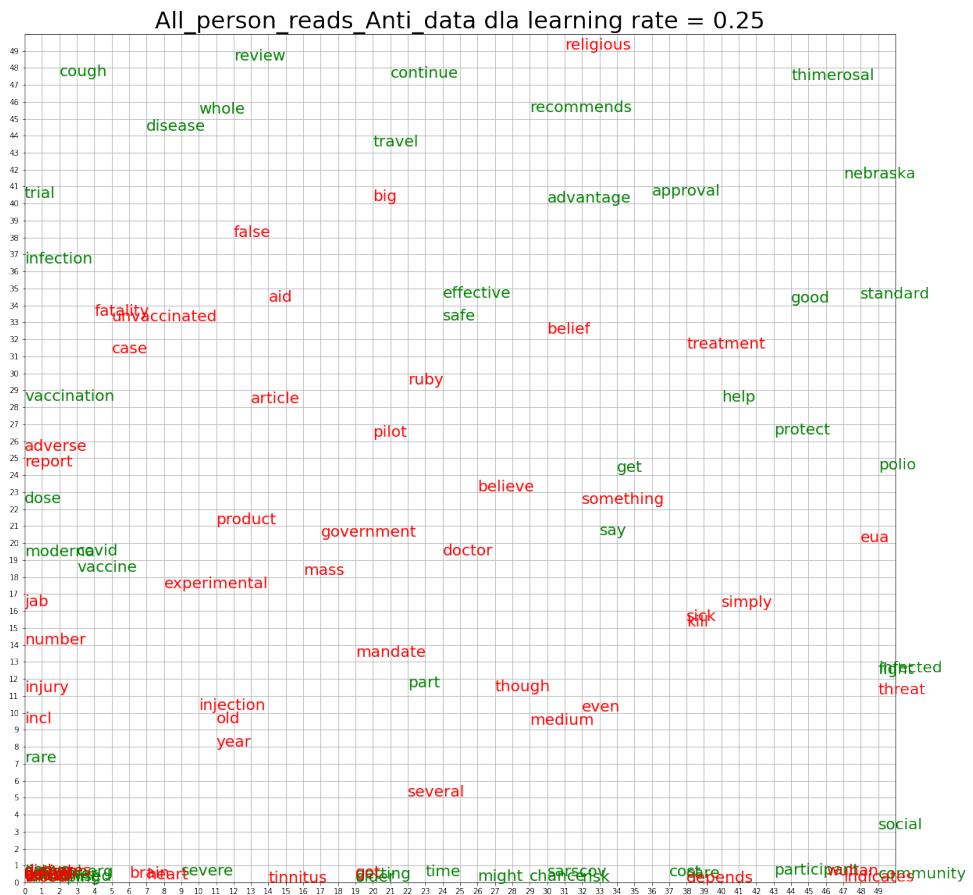
Learning rate = 1.0



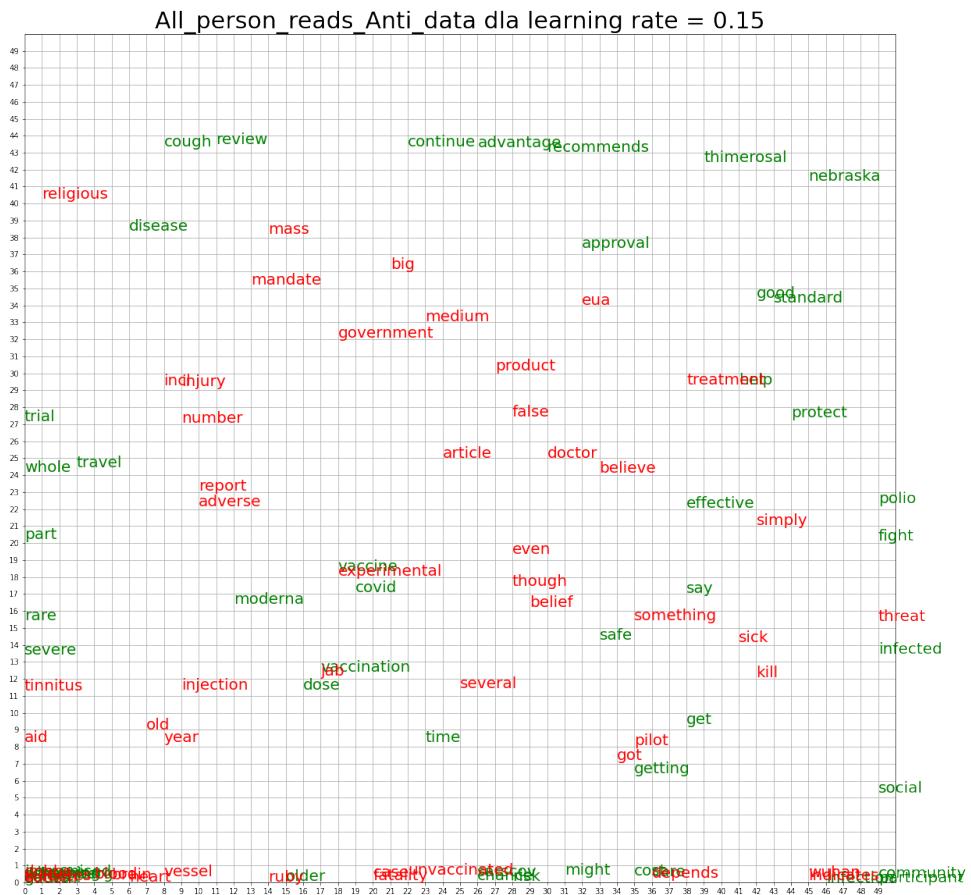
Learning rate = 0.5



Learning rate = 0.25



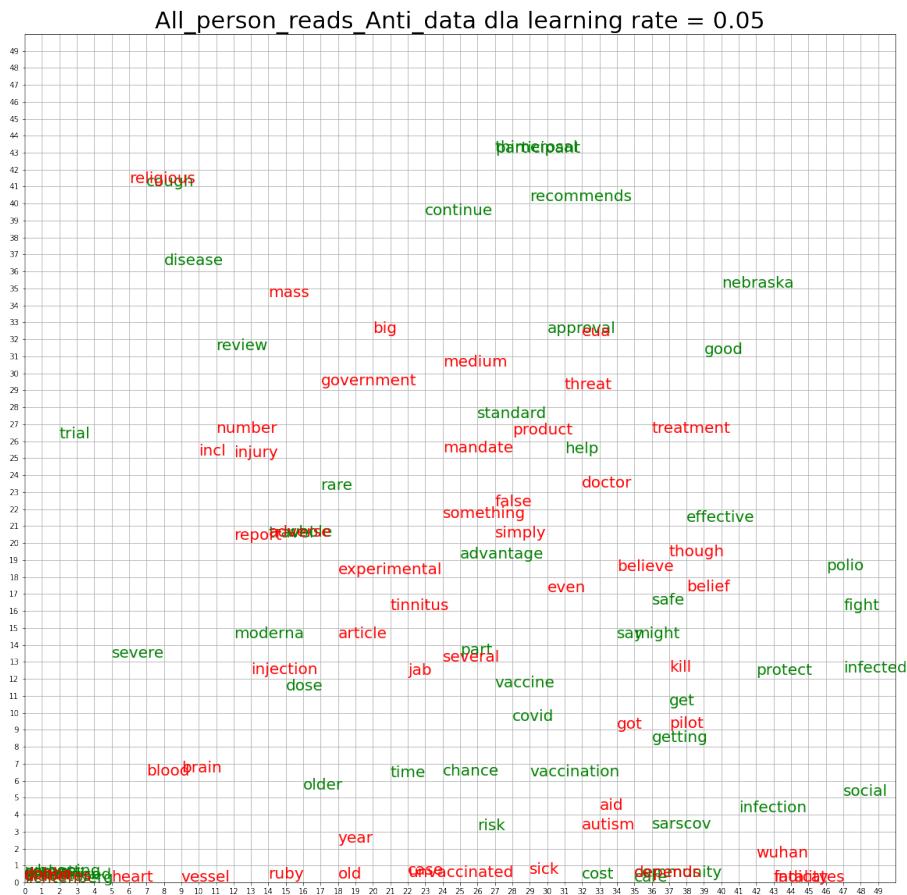
Learning rate = 0.15



Learning rate = 0.1



Learning rate = 0.05

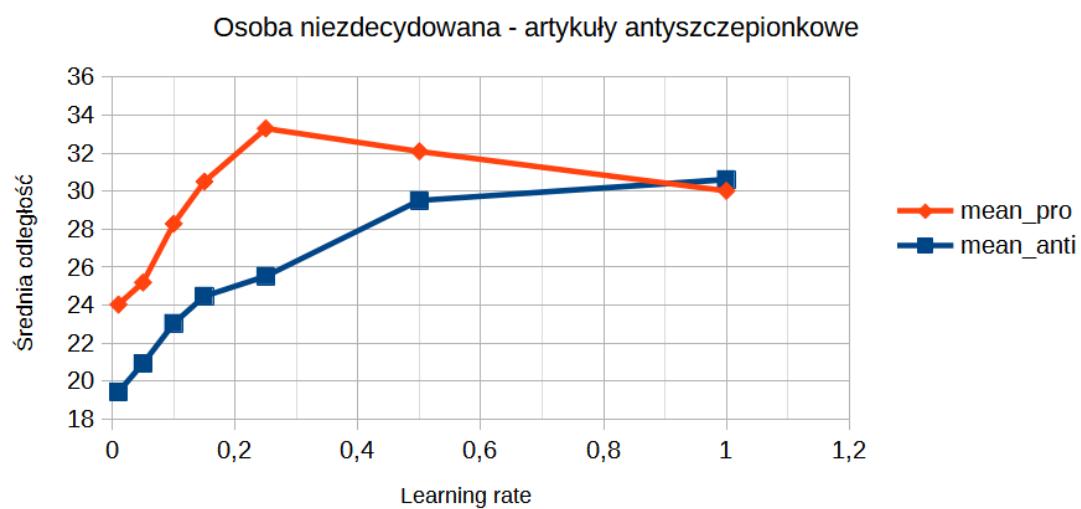


Learning rate = 0.01



LR\Distans	Suma pro	Średnia pro	Suma anty	Średnia anty
knowledge	61615.14261972499	25.169584403482432	47325.92368544548	19.332485165623154
0.01	58023.917640595726	24.03641990082673	47584.54451413297	19.438130928975887
0.05	60878.00552313778	25.197849968186166	51162.215192622825	20.933803270303937
0.1	68467.76230349518	28.29246376177486	56131.90225740378	23.042652815026184
0.15	73701.66900133915	30.50565769922978	59714.70230318776	24.473238648847442
0.25	80458.3955843063	33.302316053106914	61975.1824470377	25.525198701415857
0.5	77155.80217867796	32.09475964171296	71999.98872866922	29.50819210191361
1	72235.0391440425	30.022875787216336	74822.33233639204	30.614702265299517

Wykres zależności odległości między słowami od learning rate



Analiza

#TODO

Wnioski

Podsumowując powyższą analizę, współczynnik uczenia, tzw. learning rate, w samoorganizującej się sieci neuronowej odwzorowuje skłonność człowieka do zmiany swojej opinii przez dostęp do przekonywającej informacji. Dla najmniejszych badanych wartości osoba trzymała się swojego przekonania nie zważając na treść czytanych artykułów. Natomiast dla większych, osobę można uznać za podatną na zmiany pod wpływem otrzymywanych informacji, gdzie wartością graniczną współczynnika uczenia okazała się wartość 0.15.