Wprowadzenie do przetwarzania języka naturalnego
Dokumentacja wstępna

Nachyła Jarosław
Kwiatkowski Jan

May 31, 2023

1 Temat projektu

Pretrenowanie modelu (XLNet i RoBERTa) przystosowanego do określania interpretowalnego podobieństwa semantycznego (iSTS - https://alt.qcri.org/semeval2016/task2/) dwóch zdań w języku angielskim. Odniesienie się do wyników z obronionej pracy magisterskiej/projektu.

2 Zrealizowane założenia

Ze względu na złożoność zadania i napotkane w czasie jego realizacji problemy zaimplementowano wyłącznie model RoBERTa. Zgodnie z tematem projektu dokonano wstępnego uczenia w dwóch wariantach (pretraining na poziomie klasyfikatora oraz nienadzorowany) oraz dostosowano i nauczono model do zadania iSTS.

Do ewaluacji wyników w dwóch wersjach wykorzystano otrzymane skrypty dostępne pod linkiem:
`http://ixa2.si.ehu.eus/stswiki/index.php/Main_Page#Interpretable_STS`.

3 Przegląd literatury

W celu lepszego zapoznania się z zadaniem, uzyskiwanymi wynikami oraz wykorzystanym modelem przeanalizowano różne artykuły oraz materiały, które wymieniono poniżej:

- Praca magisterska, Aleksandra Budzyńska, 2021 - zapoznanie się z zadaniem określania podobieństwa semantycznego (iSTS) oraz modelem BERT.
- Praca magisterska, Ewelina Grudzień, 2018 - rozpoznawanie podobieństwa semantycznego z wykorzystaniem architektur sieci głębokich.
- Wyniki projektów z przedmiotu NLP z poprzednich semestrów.
- Artykuły na temat modelu RoBERTa oraz pretrenowania:

1. https://aclanthology.org/2020.coling-industry.6.pdf - idea dotrenowania.
2. https://arxiv.org/pdf/1907.11692.pdf - opracowanie RoBERTa.

4 Zadanie iSITS

Zadanie iSTS (ang. interpretable semantic textual similarity) polega na wyznaczeniu podobieństwa semantycznego pomiędzy parą fragmentów tekstu (w zadaniu projektowym zdań).

Podobieństwo iSTS składa się z:

- oceny podobieństwa w skali 1-5
- typu podobieństwa - intepretowalności (rozszerzenie w stosunku do zadania STS)

4.1 Oceny iSTS.

Oceny odpowiadają następującej skali:

- 5 - zdania semantycznie identyczne
- 4-3 - zdania podobne, ale nie identyczne
- 2-1 - zdania słabo ze sobą powiązane
- 0 - brak związku znaczeniowego

4.2 Interpretowalość iSTS.

Interpetowalność jest zdefiniowana jako nastepujące typy:

- EQUI - równoznaczne
- OPPO - o przeciwnym znaczeniu
- SPE1 - podobne znaczenie fragmentów, ale pierwszy jest bardziej szczegółowy
- SPE2 - podobne znaczenie fragmentów, natomiast drugi jest bardziej szczegółowy
- SIMI - fragmenty podobne, ale żadne z powyższych
- REL - fragmenty podobne, lecz nie posiadają podobnych atrybutów
- NOALI - dla fragmentu jednego ze zdań nie istnieje odpowiadający mu fragment w drugim zdaniu
- ALIC - podobny do NOALI, ale występuje tylko w przypadku występowania ograniczenia 1:1 w przypisywaniu do siebie fragmentów

5 Wykorzystany model

5.1 RoBERTa

5.1.1 Transformer

Architektura sieci transformer została opublikowana w 2017 roku przez informatyków z Google Brain w pracy: 'Attention Is All You Need'.

Transformer jest rodzajem sieci dekoder-enkoder co oznacza, że pozwala na redukcję informacji wejściowych (enkoder), a następnie odtworzenie ich z tej zredukowanej formy (dekoder). Pozwala to np. na uczenie się reprezentacji danych bez posiadania etykiet.

Transformer jest przystosowany do nauki danych sekwencyjnych np. tekstu, podobnie jak zwykłe sieci rekurencyjne (sekwencyjność - kontekst), które jednak posiadały pewne problemy. Transformer w odróżnieniu od nich rozwiązuje problem powolnego uczenia się i zapominania przez sieć. Wykorzystuje on mechanizm uwagi(ang. attention), który analizuję sekwencje wejściową i w każdym kroku decyduje jak ważne są poszczególne części sekwencji, przypisujac im różne wagi (zamiast tradycyjnych warstw splotowych lub rekurencyjnych).

Jest wiele modeli wykorzystujących architekture transformerów takich jak BERT, GPT, RoBERT czy XLNet, które różnią się liczbą warstw, rozmiarem wektorów, rodzajem danych wejściowych czy sposobem uczenia.

To co wyróżnia architekturę transformera to:

- zrównoleglanie - architektura ta skaluje się bardzo dobrze (45 TB danych uczących nie stanowi problemu),
- kodowanie pozycyjne - zamiast sekwencyjnie przekazywać słowa dla modelu, model uczy się intepretowac porządek jaki przypisano słowom,
- Self-attention - jest to mechanizm w enkoderze, który uczy się dopasowywać pozostałe elementy z sekwencji do poszczególnego elementu (nauka kontekstu).

5.1.2 BERT

Model BERT - Pre-training of Deep Bidirectional Transformers for Language Understanding powstał w 2019 roku. Bazuje na architekturze transformera, jednak wykorzystuje tylko mechanizm enkodera do odczytywania danych wejściowych.

Model BERT, podobnie jak tradycyjny transformer, wykorzystuje warstwy samouwagi i odczytuje całą sekwencję słów jednocześnie. Dzięki temu model potrafi zrozumieć kontekst słowa na podstawie jego otoczenia, czyli lewej i prawej strony. W ten sposób może generować reprezentacje słów, które uwzględniają ich kontekst, co przekłada się na lepsze zrozumienie sensu całego tekstu.

BERT został pretrenowany na zadaniach:

- przewidywanie słów - maskowanie losowych 15% wyrazów w sekwencji i ich przewidywanie
- przewidywania zdań wystepujących po sobie

5.1.3 RoBERTa - ulepszony BERT

Model RoBERTa został opublikowany w 2019 roku w pracy: A Robustly Optimized BERT Pretraining Approach przez Facebook AI.

Model bazuje na implementacji BERTa jednak zostały w nim ulepszone techniki szkolenia dzięki czemu poprawił wyniki pierwotnego modelu i pozwolił na trenowanie na większych zbiorach danych.

W RoBERT-cie zastosowano modyfikacje:

- zwiększenia korpusu do trenowania
- dynamicznego maskowania - maskowanie różnych tokenów tej samej sekwencji w różnych epokach uczenia
- rezygnacja z przewidywania nastepnego zdania podczas pretrenowania

6 Koncepcja projektu

6.1 Uczenie transferowe

Mając pretrenową sieć, modyfikacja architektury dla zadania iSTS wymaga potraktowania modelu RoBERTa jako ekstraktora cech i dodania do niego gęstej sieci - fully-connected służącej zadaniu klasyfikacji.

Nastepnie należy skorzystano z metody transfer learning czyli dotrenowywania ostatnich warstw klasyfikacyjnych sieci przy zamrożeniu części pretrenowanej. Podobna strategia stosowana była w analizowanych przez nas projektach z NLP studentów z poprzednich semestrów.

6.2 Pretrenowanie

6.2.1 Pretrenowanie nienadzorowane

W przypadku pretrenowania nienadzorowanego wykonano nastepujące kroki:

1. Wczytano model RoBERTa.
2. Uruchomiono trenowanie nienadzorowane modelu na wybranym zbiorze używając maskowania wyrazów w zdaniach.
3. Wykonano uczenie transferowe na zbiorach danych służących do zadania iSTS (opisane w 6.1).

6.2.2 Pretrenowanie nadzorowane na poziomie klasyfikacji

W tym przypadku wykonano następujące kroki:

1. Wczytano model RoBERTa.
2. Dodano do modelu sieć gęstą przystosowaną do klasyfikacji na zbiorze STS.
3. Uruchomiono trenowanie nadzorowane(transfer learning) modelu dla zadania STS, na zamrożonych wagach sieci ekstraktora (RoBERTa).
4. Wykonano ucznie transferowe na zbiorach danych służących do zadania iSTS (opisane w 6.1).

7 Dane

7.1 Przygotowanie zbioru do pretrenowania na poziomie klasyfikacji

Skorzystano ze zbioru dla zadania STS - stsb_multi_mt:
`https://huggingface.co/datasets/stsb\_multi\_mt` - zawierającego oryginalne przykłady i tłumaczenia na język angielski tekstów użytych w konkursie SemEval w latach 2012-2017. Łącznie zawiera 8828 par pełnych pełnych zdań wraz z oceną podobieństwa semantycznego bez typów.

Zbiór ten jest wstępnie podzielony na 3 zbiory train, test i dev, jednak do zadania pretrenowania 3 zbiory są zbędne (nie wykonywano ewaluacji na tym

zbiorze). Zbiór testowy został więc połączony ze zbiorem treningowym. Otrzymując podział:

```
train  7128
valid  1500
```

Zbiór oryginalnie był przystosowany do regresji jednak dostosowano go do problemu klasyfikacji. Kolumna z wynikiem podobieństwa w wartościach zmiennoprzecnkowych została zakrąglona do wartości całkowitych:

```
df_data['target'].map(lambda x: int(Decimal(x).to_integral_value()))
```

Zmiana podyktowana była lepszymi wynikami osiąganymi przez klasyfikator w porównaniu do sieci rozwiązujacej zadanie regresji. Dodatkowo zastosowano wagowanie odwrotne które poprawiało wyniki, szczególnie f1_macro (metryka która bardziej karze nierównomierne wyniki w klasach, nawet małolicznych).

7.2 Przygotowanie zbioru do pretrenowania nienadzorowanego.

W przypadku treningu nienadzorowanego użyto tego samego zbioru co w przypadku pretrenowania na poziomie klasyfikacji(STS Benchamrk MT), maskując 15% losowo wybranych tokenów.

Podzielono go na zbiór treningowy i walidacyjny w proporcji 0.8 i 0.2 całości. Zbiór walidaycyjny słuzy do uniknięcia overfittingu przy treningu nienadzorowanym.

7.3 Zbiór do ewaluacji

Do ewaluacji użyto zbiór SemEval z 2015 i 2016 roku dla zadania iSTS zawierający frazy. Zawiera on pary fraz ich podobieństwo(1-5), a także typ podobieństwa semantycznego(1-8). Zbiór zawiera dane pochodzące z 3 źródeł:

- image phrases (970 par treningowych, 943 par testowych),
- headlines(2188 par treningowych, 1172 par testowych),
- answer students(919 par treningowych, 933 par testowych).

7.3.1 Przygotowanie danych

Dane zostały pobrane w postaci plików .tsv, zawierają takie kolumny:

- x1 - zdanie 1
- x2 - zdanie 2
- y_score - wartość score'a podobieństwa [1-5]
- y_type - typ podobienstwa [1-6]

Kolumny x1 i x2 połączono w jeden string rozdzielony kropką i zapisano do kolumny text. Kolumna value używana była jako score. Kolumnę y_type jest zmapowana na wartości liczbowe zgodnie z poniższym:

```
types_map = {'EQUI':0, 'SIMI':1, 'SPE1':2, 'SPE2':3, 'REL':4, 'OPPO':5}
```

W kolumnie y_type dla zbioru treningowego znajdowało się tylko 6 wartości, brakowało NOALI i ALIC. Zatem w zbiorze trenujacym dla score'ów mieliśmy 5 wartości oraz dla typów 6.

Wartości przewidywane dla score i type były niezbalansowane. Wyliczono dla nich odwrotne wagowanie. Dystrubucja dla score'a wygląda nastepująco:

```
5    2447
4      58
3     765
2     571
1     233
```

7.3.2 Podział na zbiory do nauki

Zbiór testowy był oryginalnie wydzielony i podzielony dodatkowo na 3 typy (image phrases, headlines i answer students). Ostatecznie po podziale otrzymano 3 zbiory.

- treningowy - 2352
- walidacyjny - 589
- testowy - 3045

8 Implementacja

8.1 Modyfikacja sieci

Modyfikacja sieci jest dość prosta tzn. model składa się z:

- sieci RoBERTa jako ekstraktora cech
- sieci gęstej jako klasyfikatora

Do pretreningu na poziomie klasyfikacji, zarówno do pretrenowania jak i do uczenia właściwego modelu dla iSTS użyto tej samej architektury sieci.

Poniżej przedstawiono model sieci:

Sieć gęsta zawiera 768 neuronów połączonych z siecią Roberta Fully Conncted(każdy z każdym), a na ich wyjściu funkcję aktywacji ReLU. Zastosowano dropout w celu redukcji przeuczenia usuwając 0.3 neuronów z warstwy. Wyjście w zależności od przewidywania score lub type było o rozmiarze odpowiednio 5 lub 8.

8.2 Pretrenowanie z użyciem klasyfikacji

Implementacja znajduje sie w zamieszczonym notatniku: NLP Projekt STS Preatrain Classification.ipynb.
Idea pretrenowania z użyciem klasyfikacji była następująca:

1. Przygotowano dane zgodne z opisem w podrozdziale 7.1.
2. Wartości przewidywane skwantowano na klasy od 1 do 5 zmieniając zadanie regresji na klasyfkację.

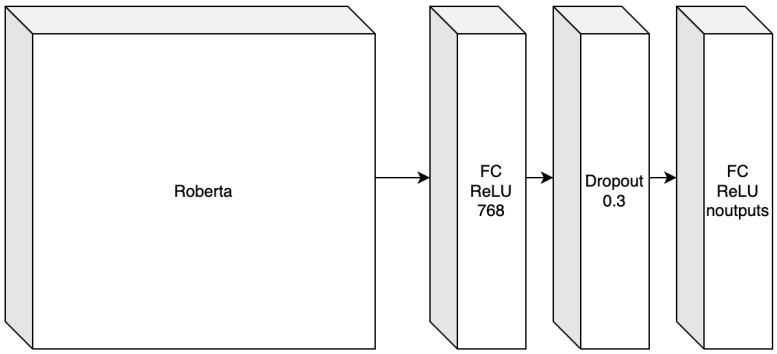

Figure 1: Architektura sieci klasyfikującej z ekstraktorem RoBERTa.

3. Przygotowano model trenujący zgodny z architekturą na rysunku 1.
4. Model trenowano ustawieniami opisanymi poniżej, mierząc stratę na zbiorze walidacyjnym.
5. Ekstraktor danych - RoBERTa zapisano dla najwyższej wartości wyniku na zbiorze walidacyjnym.
6. Ustawienia pretrenowania:
   - learning rate - 1e-5
   - maksymalna liczba epok - 12
   - batch_size - 32

8.3 Pretrenowanie nienadzorowane

W przypadku pretrenowania nienadzorowanego wykonano:

1. Przygotowano dane zgodne z opisem w podrozdziale 7.2.
2. Wartości wyjściowe zostały stworzone jako zamaskowane wartości wejściowe (15%).
3. Model trenowano ustawieniami opisanymi poniżej, mierząc stratę na zbiorze walidacyjnym.
4. Ekstraktor danych - RoBERTa zapisano dla najwyższej wartości wyniku na zbiorze walidacyjnym.
5. Ustawienia pretrenowania:
   - learning rate - 1e-5
   - maksymalna liczba epok - 12
   - batch_size - 32

8.4 Transfer learning

W celu rozwiązania zadania semantycznego wykrywania podobieństwa wagi z wcześniej pretrenowanych modeli wykorzystano w modelu RoBERTa do którego podłączono sieć o odpowiedniej architekturze. Wagi modelu RoBERT zostały zamrożone a uczona była tylko wyjściowa sieć.

Stworzyliśmy dwie 'głowy' jedna służyła do klasyfikacji scoreów a druga do klasyfikacji typów. W pierwszej wersji programu głowy były podpięte do osobnych sieci RoBERTa i modele były tenowoane osobno - załącznik NLP_RoBERTa_semantic_evaluation_ISITS_2heads. Jednak udało nam sie poprawić architekturę i dodaliśmy
trenowanie w jednej sieci - załącznik NLP_RoBERTa_semantic_evaluation_ISITS_1head.

- learning rate - 1e-5
- maksymalna liczba epok - 12
- batch_size - 32

8.5 Metryki

Głównymi metrykami do oceny modelu jakie zastosowano była miara F1. Metryka F1 jest średnią harmoniczną precyzji i czułości. Precyzja to ilość poprawnie sklasyfikowanych przykładów do ilości przykładów przypisanych do danej klasy przez klasyfikator. Natomiast Czułość (Recall) to liczba poprawnie sklasyfikowanych przykładów do liczby przykładów które powinny zostać sklasyfiwkowane.

W naszym przypadku liczone będą 2 wartości metryki F1:

- F1 dla ocen
- F1 dla typów

9 Eksperymenty

9.1 Pretrening nadzorowany na poziomie klasyfikacji.

Pierwszym etapem było wykonanie pretreningu modelu zgodnie z założeniami opisanymi w rozdziale 8. Wyniki poszczególnych wartości funkcji straty dla kolejnych iteracji w trakcie uczenia nadzorowanego przedstawiono na rysunku 2. Wagi modelu pretrnowanego zapisywano dla najwyzszej wartości funkcji straty na zbiorze walidacyjnym.

9.1.1 Wyniki na zbiorze iSTS - pretraining nadzorowany.

Poniżej zamieszczono wyniki dla ewaluacji modelu RoBERTa pretrenowanego w sposób nadzorowanego. Wyniki są rozdzielone na zbiory: Answer Students, Headlines i Images i zawierają raporty dla poszegónych klas, a także zagregowane wyniki metryk F1, Precision, Recalli Accuracy. Dodatkowo dodaliśmy macierze pomyłek okazujące ilościowo poprawnie i niepoprawnie sklasyfikowane przykłady. Głównym problemem jaki możemy zauważyć jest niezbalansowanie klas, które znacząco wpływa na metrykę F1 z makroúrednianiem.

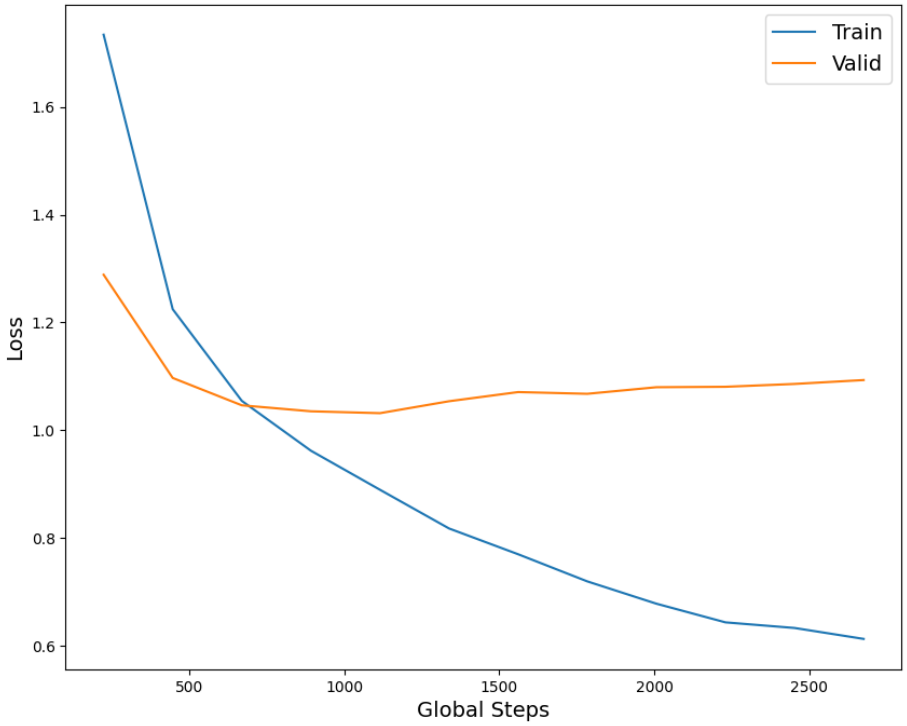

Figure 2: Pretrenowanie z klasyfikatorem modelu RoBERTa. Oś X kroki, Oś Y wartość funkcji straty.

Answer Students,

	precision	recall	f1-score	support
0	0.9600	0.8936	0.9256	564
1	0.7451	0.4935	0.5938	77
2	0.4667	0.4179	0.4409	67
3	0.5673	0.7564	0.6484	78
4	0.7778	0.8660	0.8195	97
5	0.5833	1.0000	0.7368	49
accuracy			0.8176	932
macro avg	0.6834	0.7379	0.6942	932
weighted avg	0.8351	0.8176	0.8192	932

Headlines

	precision	recall	f1-score	support
0	0.9521	0.8976	0.9240	664
1	0.7162	0.7067	0.7114	150
2	0.5610	0.4694	0.5111	98
3	0.5827	0.6916	0.6325	107
4	0.6087	0.8485	0.7089	66
5	0.4500	0.9000	0.6000	10
accuracy			0.8100	1095
macro avg	0.6451	0.7523	0.6813	1095
weighted avg	0.8234	0.8100	0.8135	1095

Images

	precision	recall	f1-score	support
0	0.9596	0.9038	0.9309	499
1	0.7593	0.7235	0.7410	170
2	0.7188	0.7302	0.7244	126
3	0.6714	0.8624	0.7550	109
4	0.7568	0.8000	0.7778	35
5	0.6000	1.0000	0.7500	3
accuracy			0.8397	942
macro avg	0.7443	0.8366	0.7798	942
weighted avg	0.8492	0.8397	0.8424	942

Figure 3: Raporty klasyfikacji dla typów - Pretrenowanie z klasyfikatorem modelu RoBERTa.

Answer Students,

	precision	recall	f1-score	support
0	0.0000	0.0000	0.0000	0
1	0.5385	0.5957	0.5657	47
2	0.6218	0.6789	0.6491	109
3	0.7739	0.7230	0.7476	213
4	0.9359	0.9343	0.9351	563
micro avg	0.8391	0.8391	0.8391	932
macro avg	0.5740	0.5864	0.5795	932
weighted avg	0.8421	0.8391	0.8402	932

Headlines

	precision	recall	f1-score	support
0	0.4483	0.8667	0.5909	15
1	0.4224	0.7000	0.5269	70
2	0.5607	0.4839	0.5195	124
3	0.6541	0.5500	0.5975	220
4	0.9301	0.9189	0.9245	666
accuracy			0.7808	1095
macro avg	0.6031	0.7039	0.6319	1095
weighted avg	0.7938	0.7808	0.7829	1095

Images

	precision	recall	f1-score	support
0	0.5000	1.0000	0.6667	13
1	0.4750	0.6909	0.5630	55
2	0.5758	0.5327	0.5534	107
3	0.7842	0.7079	0.7441	267
4	0.9234	0.9160	0.9197	500
accuracy			0.8015	942
macro avg	0.6517	0.7695	0.6894	942
weighted avg	0.8124	0.8015	0.8040	942

Figure 4: Raporty klasyfikacji dla score'ów - Pretrenowanie z klasyfikatorem modelu RoBERTa.

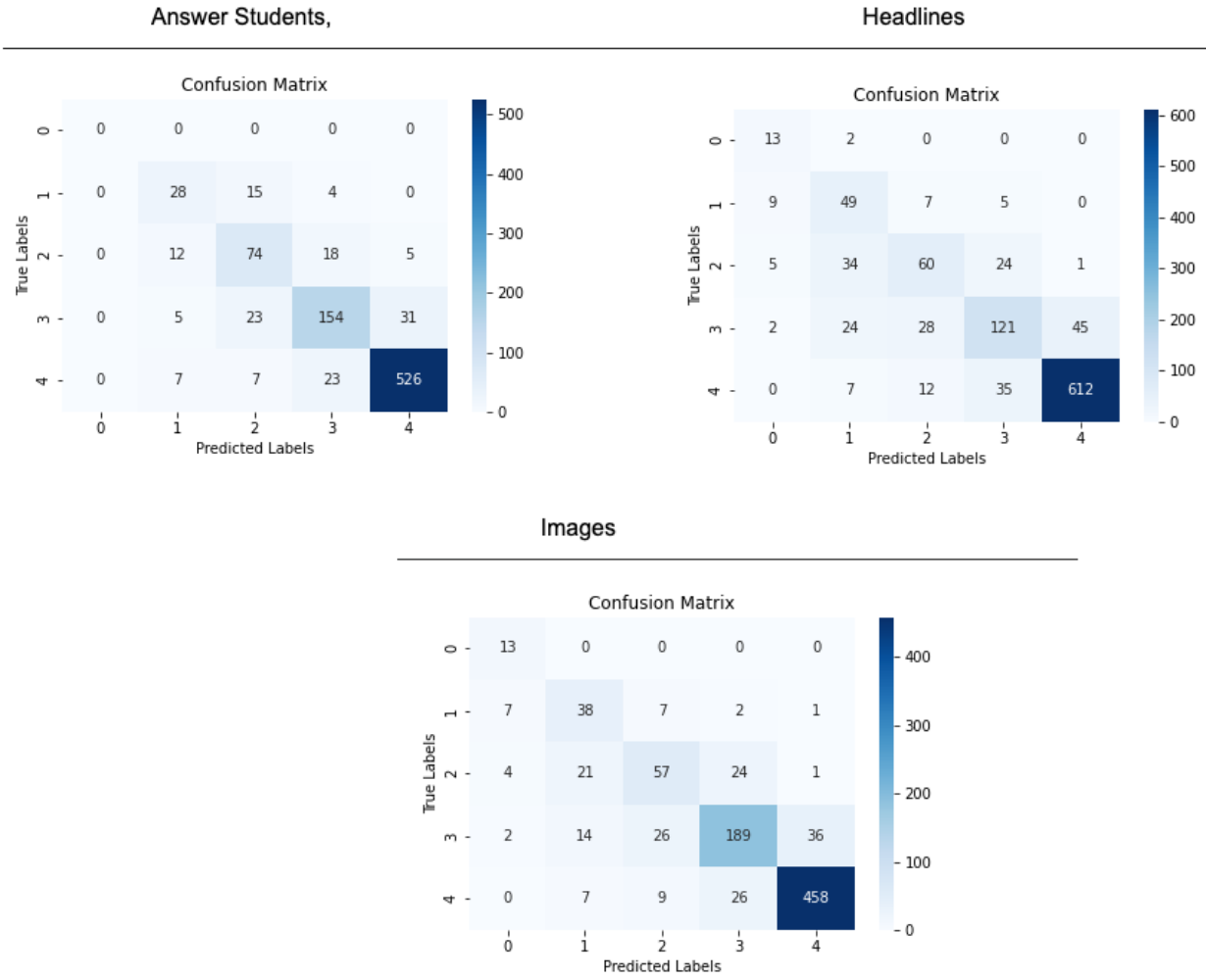

Figure 5: Macierz pomyłek dla score'ów - Pretrenowanie z klasyfikatorem modelu RoBERTa.

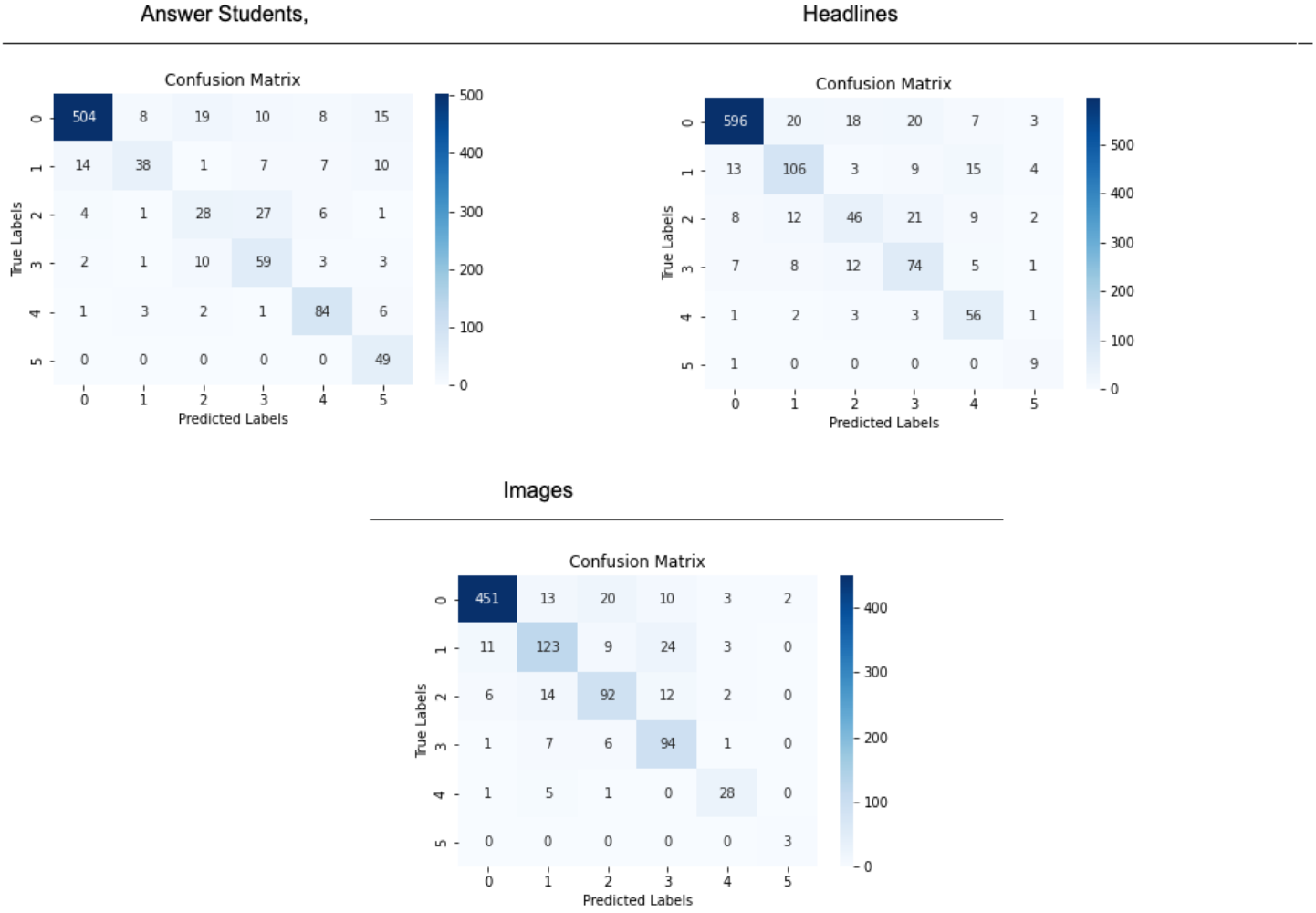

Figure 6: Macierz pomyłek dla typów - Pretrenowanie z klasyfikatorem modelu RoBERTa.

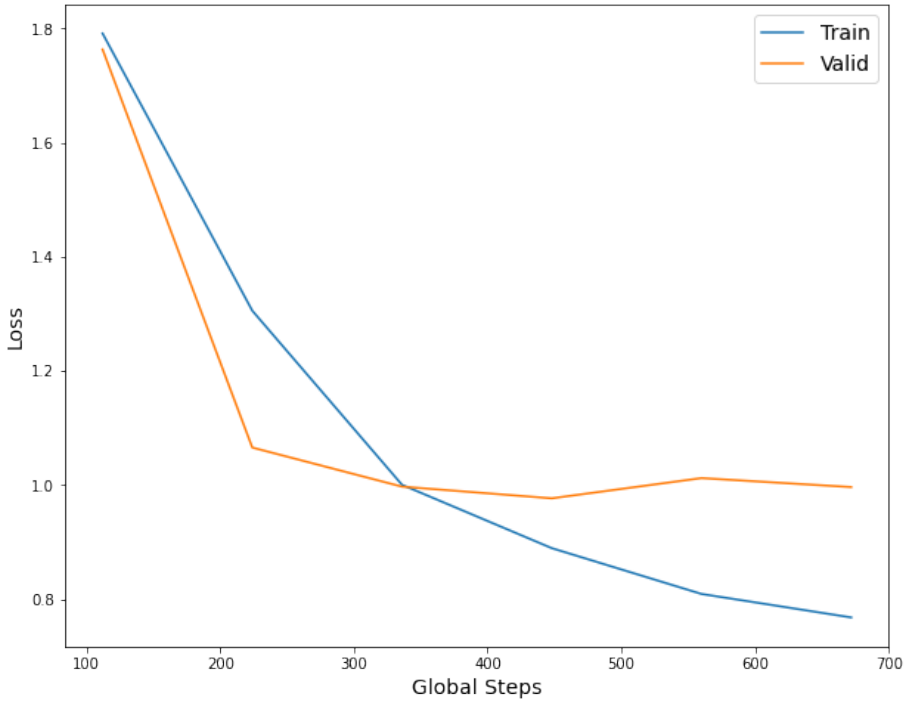

Figure 7: Pretrenowanie nienadzorowane modelu RoBERTa. Oś X kroki, Oś Y wartość funkcji straty.

9.2 Pretrening nienadzorowany.

Dla pretreningu nienadzorowanego operacja trenowania po za zmianą zadania wyglądała podobnie. Podzielono dane na zbiór treningowy i walidacyjny, gdzie zbiór walidacyjny słuzył do zapisu najlepszej wersji modelu w celu uniknięcia przeuczenia. Wyniki poszczególnych wartości funkcji straty dla kolejnych iteracji w trakcie uczenia nadzorowanego przedstawiono na rysunku 7.

9.2.1 Wyniki na zbiorze iSTS - pretraining nienadzorowany.

Poniżej zamieszczono wyniki dla ewaluacji modelu RoBERTa pretrenowanego w sposób nienadzorowany. Wyniki są analogicznie zaprezentowane jak w przypadku pretreningu nadzorowanego, a ogólne wnioski zostały opisane w 9.3.

Answer Students,

	precision	recall	f1-score	support
0	0.9495	0.9007	0.9245	564
1	0.7872	0.4805	0.5968	77
2	0.5424	0.4776	0.5079	67
3	0.5506	0.6282	0.5868	78
4	0.7391	0.8763	0.8019	97
5	0.5517	0.9796	0.7059	49
accuracy			0.8144	932
macro avg	0.6868	0.7238	0.6873	932
weighted avg	0.8307	0.8144	0.8149	932

Headlines

	precision	recall	f1-score	support
0	0.9446	0.8991	0.9213	664
1	0.7042	0.6667	0.6849	150
2	0.5060	0.4286	0.4641	98
3	0.5772	0.6636	0.6174	107
4	0.5446	0.8333	0.6587	66
5	0.6429	0.9000	0.7500	10
accuracy			0.7982	1095
macro avg	0.6533	0.7319	0.6827	1095
weighted avg	0.8097	0.7982	0.8009	1095

Images

	precision	recall	f1-score	support
0	0.9499	0.9118	0.9305	499
1	0.7152	0.6647	0.6890	170
2	0.7131	0.6905	0.7016	126
3	0.6370	0.7890	0.7049	109
4	0.6136	0.7714	0.6835	35
5	0.7500	1.0000	0.8571	3
accuracy			0.8185	942
macro avg	0.7298	0.8046	0.7611	942
weighted avg	0.8265	0.8185	0.8208	942

Figure 8: Raporty klasyfikacji dla typów - Pretrenowanie nienadzorowane modelu RoBERTa.

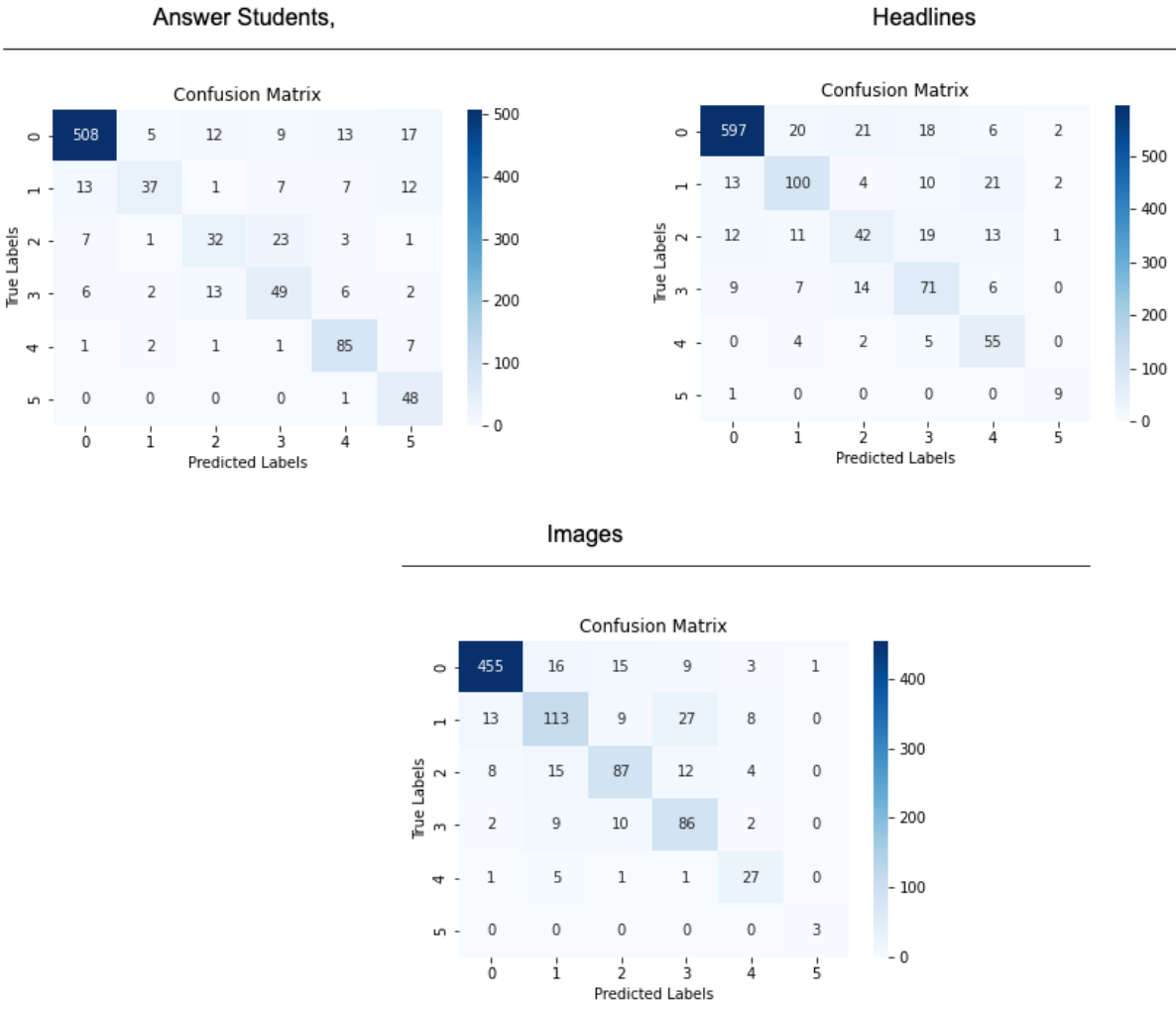

Figure 9: Macierz pomyłek dla typów - Pretrenowanie nienadzorowane modelu RoBERTa.

9.3 Porównanie wyników i wnioski.

Table 1: Zbiór wykorzystany do tworzenia przykładowego drzewa

Model and Dataset	F1 macro types	F1 macro scores
Supervised A.S.	0.694	0.6
Supervised Head.	0.681	0.631
Supervised Img.	0.78	0.69
Non Supervised A.S.	0.687	brak
Non Supervised Head.	0.682	brak
Non Supervised Img.	0.761	brak

Do porównania wyników działania modeli stosowalismy metrykę F1 z makroúrednianiem, która jest odpowiednia jeśli chcemy traktować wszystkie klasy jednakowo co daje lepszy obraz jak model radził sobie z niezbalansowanym zbiorem danych. Wartości makro sa dużo niższe od wartości ang. weigheted, które wyszły powyżej 0.8. Zastosowano podczas nauki w pretrenowaniu jak i w uczeniu transferowym odwrotne wagowanie przekazwane do funkcji straty w celu zniwelowania efektu nizbalansowanych klas w zbiorach, co poprawiło reultaty o 2 punkty procentowe.

Porównując obie metody pretrenowania można wysnuć wniosek, że obie metody dają zbliożone rezultaty, a różnice jakie wystepują w wynikach są w granicach błędu statytycznego - na drugim miejscu po przecinku. Jedynie dla zbioru Images różnica na korzyść pretreningu nadzorowanego o 1.5 % (0.78% do 0.761 %) może wskazywać na minimalnie lepsze rezultaty tej metody. Do obydwu metody pretrenowania zastosowano ten sam zbiór z zadania STS - około 8 tysięcy przykładów, co dla pretreningu nienadzorowanego może okazać się zbyt małym zbiorem. Model RoBERTa jest pretrenowany oryginalnie do przewidywania zamaskowanych tokenów gdzie rozmiar zbioru do trenowania to były gigabajty danych. W porównaniu do kilkutysięcy zdań zdolność generalizacji modelu na niższych wartstwach RoBERTa mogła nie ulec oczekiwanemu polepszeniu. Należało by do pretrenowania użyć wiekszego zbioru.