Sieci neuronowe w NLP

Paweł Rychlikowski

Instytut Informatyki UWr

7 lutego 2019

Sieci neuronowe w NLP

- Założenia: w zasadzie to powienien być osobny przedmiot, tylko dla absolwentów wykładu PJN i SN (może kiedyś...)
- Przyjmiemy sobie dwa cele:
  a) Zrozumieć najprostsze neuronowe modele językowe (ciekawa praca inżynierska?)
  b) Zrozumieć, jak działa sieć osiągająca najlepsze wyniki na zadaniach NLP (ciekawa praca magisterska?)

Uwaga inż/mgr

Oczywiście przenosimy się z angielskiego na polski! W przypadku pierwszej pracy (inżynierskiej) oprócz najprostszych pewnie warto byłoby pójść trochę dalej.

Sieci neuronowe w 5 prostych slajdach

- Wybierzemy absolutne minimum tego, co należy wiedzieć o sieciach neuronowych (dostosowane do naszych celów)
- Oczywiście nie będzie to w pełni kompletna wiedza.

Slajd 1. Neuron

- Neuron to funkcja $f : \mathcal{R}^n \to \mathcal{R}$

$$f(x_1 \dots x_n) = \sigma(\sum_1^n w_i x_i + b)$$

- σ jest jakąś ustaloną funkcją nieliniową, raczej rosnącą, raczej różniczkowalną, na przykład: $\max(0, v)$, albo $\tanh(v)$
- Wygodna (jak za chwilę zobaczymy) jest notacja wektorowo-macierzowa, w niej mamy:

$$f(\mathbf{x}) = \sigma(\mathbf{w}^T \cdot \mathbf{x} + b)$$

Slajd 2. Prosta sieć neuronowa

- Warstwa to funkcja $\mathcal{R}^n \to \mathcal{R}^m$.
- Najbardziej typowa warstwa wyraża się wzorem:

$$L(\mathbf{x}) = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b})$$

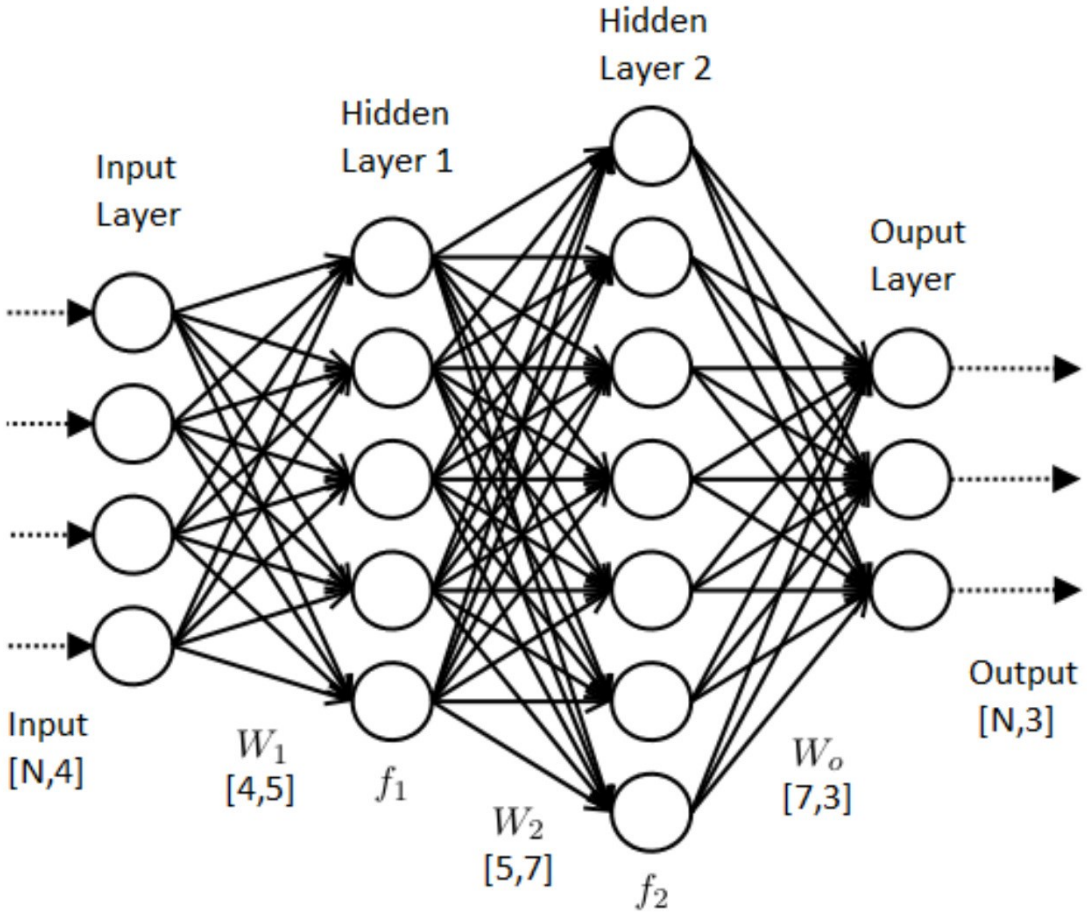
- **Uwaga:** $\mathbf{W}$ jest macierzą wag (złożoną z wektorów wag), a $\sigma(y_1 \dots y_m) = (\sigma(y_1) \dots \sigma(y_m))$

Definicja

Sieć neuronowa typu MLP jest złożeniem warstw (z różnymi macierzami wag dla każdej warstwy).

Slajd 2. Prosta sieć neuronowa

(ten rysunek jest na poprzednim slajdzie, ale byłoby mu tam trochę ciasno)



Źródło: VIASAT (https://medium.com/coinmonks/the-artificial-neural-networks-handbook-part-1-f9ceb0e376b4)

Slajd 3. Uczenie sieci

Zadanie

Danymi jest ciąg $(\mathbf{x}_i, \mathbf{y}_i)$ opisujący porządane zachowanie sieci S oraz architektura tejże sieci (liczba warstw, ich wymiary, funkcja/funkcje σ).
Chcemy tak dobrać parametry ($\mathbf{W_k}$ oraz $\mathbf{b_k}$) żeby dla każdego i

$$S(\mathbf{x}_i) \approx \mathbf{y}_i$$

Funkcja kosztu

Powyższe zadanie formalizujemy jako zadanie znalezienia takich parametrów, że koszt błędów jest jak najmniejszy. Przykładowo, jeżeli wyściem jest liczba, to możemy wybrać:

$$\text{Loss}(\theta) = \sum_{i}^{n} (S_\theta(\mathbf{x}_i) - y_i)^2$$

Slajd 4: Softmax

- Często chcemy, żeby sieć decydowała o jednej z K opcji.
- Rozmywamy ten wybór, prosząc o podanie rozkładu prawdopodobieństwa dla wszystkich K opcji.
- To tzw. Softmax layer, która przypisuje prawdopodobieństwo zależne od wielkości pobudzenia.

Wzór:

$$\text{Sotfmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{d} e^{z_j}}$$

Uwaga

Zastosowania Softmaxu to generowanie tekstu (wybór kolejnego słowa), tagowanie (wybór dostępnego tagu), nasze zadanie z rekonstrukcją (wybór literki w danym miejscu), ...

Slajd 5: one-hot encoding i zanurzenia słów

- Wejściem do sieci może być element ze skończonego zbioru o mocy K (bądź ciąg takich elementów).
- Kodujemy je za pomocą zer i jedynek, w rozrzutny sposób, za pomocą K-elementowych wektorów bitowych z jedną jedynką na odpowiedniej pozycji i $K-1$ zerami.
- Przykładowo: dla alfabetu `abcde` i ciągu `aabee` kodem jest `10000 10000 01000 00001 00001`
- Jeżeli $\mathbf{x}$ jest wektorem typu *one hot*, wówczas $\mathbf{Wx}$ jest wyborem kolumny z macierzy $\mathbf{W}$

Uwaga

Tę kolumnę nazwiemy zanurzeniem słowa s (jeżeli $\mathbf{x}$ ma jedynkę na pozycji odpowiadającej słowu s)

Neuronowe modele językowe

Neuronowy model językowy rozwiązuje zadanie wyznaczenia prawdopodobieństwa kolejnego symbolu (słowa, litery, kawałka słowa) na podstawie poprzednich elementów. Czyli wyznaczamy:

$$P(w_n|w_1 \dots w_{n-1})$$

- Możemy stworzyć w ten sposób model N-gramowy – wejściem będzie wówczas ciąg kodów (one-hot lub policzone gdzie indziej zanurzenia) poprzednich $N-1$ słów.

Uwaga

word2vec można interpretować jako prosty model językowy, przewidujący słowo z innego sąsiedniego (niekoniecznie b.blisko) słowa.

Neuronowe modele językowe (2)

Jak patrzeć dalej wstecz?

Trzy odpowiedzi:

1. Sieci rekurencyjne
2. Sieci konwolucyjne
3. Transformer i mechanizm uwagi

Neuronowe modele językowe (2)

Jak patrzeć dalej wstecz?

Trzy odpowiedzi:

1. Sieci rekurencyjne (RNN, LSTM, Bi-LSTM, GRU)
2. Sieci konwolucyjne (ByteCNN, ...)
3. **Transformer i mechanizm uwagi**

Zobacz pracę: Attention is all you need, Vaswani i inni

Mechanizm uwagi

Będziemy potrzebowali dwóch intuicji (dla prostych operacji na zanurzeniach)

- Co to jest mnożenie wektora zanurzenia przez macierz?
- Co to jest dodawanie wektorów zanurzeń?

Dodawanie wektorów zanurzeń. Intuicja

Fakt

Losowe wektory są do siebie niepodobne. (dla $D = 300$ średnia wartość bezwzględna cosinusa dwóch wektorów to 0.045.)

Definicja

Zapytaniem nazwiemy wektor (o wymiarze D), dla którego chcemy znajdywać najbardziej podobne zanurzenia słów.

Najsłynniejsze zapytanie

king - man + woman daje na szczycie rankingu queen

Dodawanie wektorów zanurzeń (2)

Przeanalizujmy działanie zapytania:

jabłko + *porucznik* + *matematyka* + *łyżka*

Wyniki iloczynów skalarnych z wektorami kąpiel, algebra, gruszka, wątroba.

Dodawanie wektorów zanurzeń (2)

Przeanalizujmy działanie zapytania:

jabłko + *porucznik* + *matematyka* + *łyżka*

Wyniki iloczynów skalarnych z wektorami kąpiel, **algebra**, **gruszka**, wątroba.

Dodawanie wektorów zanurzeń (3)

- Mamy wektory dla form bazowych (b_w)
- W podobny sposób możemy wyznaczyć wektory dla supertagów (t_w)
- **Pytanie:** Czym będzie $b_w + t_w$?

Popatrzmy:

$$(b_x + t_x) \cdot (b_y + t_y) = b_x \cdot b_y + b_x \cdot t_y + t_x \cdot b_y + t_x \cdot t_y$$

Dodawanie wektorów zanurzeń (3)

- Mamy wektory dla form bazowych (b_w)
- W podobny sposób możemy wyznaczyć wektory dla supertagów (t_w)
- **Pytanie:** Czym będzie $b_w + t_w$?

Popatrzmy:

$$(b_x + t_x) \cdot (b_y + t_y) = \mathbf{b_x} \cdot \mathbf{b_y} + b_x \cdot t_y + t_x \cdot b_y + \mathbf{t_x} \cdot \mathbf{t_y}$$

Wyboldowane składniki dominują, pozostałe, jako przypadkowe, mają niewielkie wartości

Dodawanie wektorów zanurzeń (4)

- Dodawanie wektorów *z tej samej bajki* do pewnego stopnia działa jak tworzenie zbiorów tych wektorów
- Dodawanie wektorów z różnych dziedzin działa jak tworzenie *koniunkcji* właściwości

Uwaga

Korzysta z tego FastText, który traktuje słowo jako zanurzenie różnych jego kawałków. Przykładow (nie do końca z FastTextu): supermitologicznej = super + nej + mito + ogiczn

Mnożenie wektorów przez macierz. Intuicje

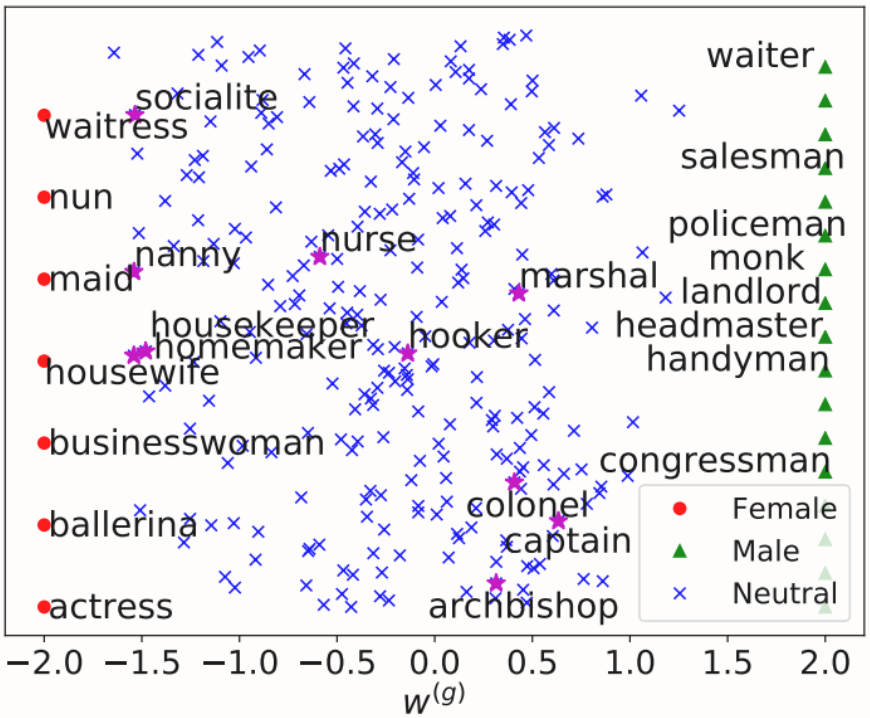
- Chcemy sprawdzić, czy x jest bardziej słoniem czy żyrafą?
- Równoważne alternatywy:
  1. Czy x·słoń $>$ x·żyrafa ?
  2. x·słoń - x·żyrafa > 0 ?
  3. x· (słoń - żyrafa) > 0

Pytanie

Czy są sensowniejsze wektory niż słoń - żyrafa?

Wektory a stereotypy

- Ciekawą osią jest she - he
- Wprowadzamy w ten sposób wymiar męskości i żeńskości słów.
- Przykłady (z pracy *Learning Gender-Neutral Word Embeddings*)



Wektory a stereotypy (2)

Przykładowe zapytania wraz z wynikami:

- doctor - father + mother $\approx$ nurse
- computer_programmer - man + woman $\approx$ homemaker.

Uwaga

Dla nas istotne jest, że mnożenie przez wektor może wprowadzać nowy, użyteczny wymiar dla zanurzenia słowa (a mnożenie przez macierz to zestaw mnożeń przez wektor)

Najprostszy mechanizm uwagi

- Rozważamy zdanie:

  Grażyna i Janusz swoim samochodem z wysokoprężnym silnikiem jechali koło słoni, żyraf i bawołów w Parku Narodowym Serengeti.

- Wyrazy o tym samym kolorze są powiązane:

  Grażyna i Janusz swoim samochodem z wysokoprężnym silnikiem jechali koło słoni, żyraf i bawołów w Parku Narodowym Serengeti.

- Wyznaczamy podobieństwa każdy z każdym, do każdego słowa *domieszowujemy* słowa podobne (tak naprawdę to dodajemy wszystkie, ale z wagą zależną od podobieństwa)

Mechanizm uwagi. Co dalej?

- Alternatywa dla MU, czyli sumowanie wszystkiego, trochę za bardzo rozmywa sygnał.
- Można połączyć z uczeniem macierzy projekcyjnej, przez którą mnożymy zanurzenie słowa.
- Wektory, które porównujemy występują w 3 rolach:
  1. **Query**: ja mam taki wektor, a wy?
  2. **Key**: a my mamy takie, porównajmy, wyznaczmy współczynniki podobieństwa
  3. **Value**: pomnóżmy te współczynniki przez nasze wektory wartości

Te role można rozdzielić!

Mechanizm uwagi użyty w systemie BERT

- Osobne macierze dla Q,K,V
- Kilka **głowic**, czyli zestawów QKV, każda z osobnymi macierzami
- W paru miejscach dodatkowe MLP
- Zabawę powtarzamy kilka razy, tworząc kolejne warstwy wektorów coraz inteligentniejszych zanurzeń wyrazów w kontekście zdania (w każdej warstwie inne parametry).

Dwa zadania dla BERT-a

Zadanie 1. Następstwo zdań

Widzisz dwa zdania, zgadnij czy następują po sobie w tekście, czy są bardziej odległe.

Zadanie 2. Model językowy (wariant Masked LM)

Zgadnij usunięte wyrazy:

> *Niektóre ptaki , szczególnie krukowate i papugowe należą do najbardziej inteligentnych gatunków zwierząt , zdolnych do tworzenia i używania przyrządów pomocniczych , jak i przekazujących tę wiedzę następnym pokoleniom .*

Dwa zadania dla BERT-a

Zadanie 1. Następstwo zdań

Widzisz dwa zdania, zgadnij czy następują po sobie w tekście, czy są bardziej odległe.

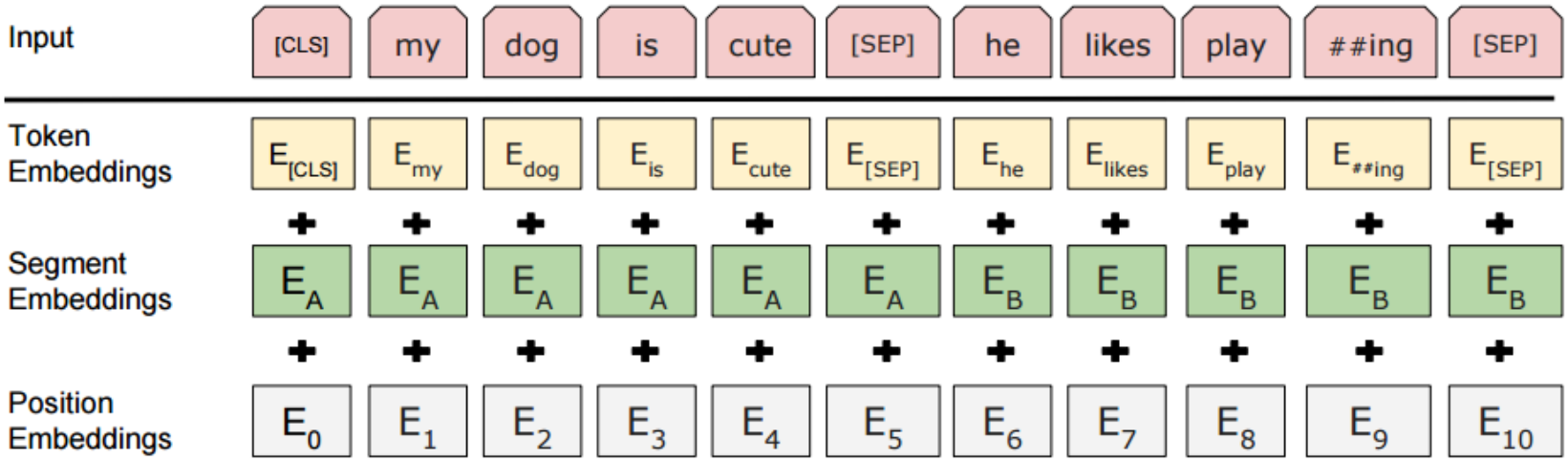
Zadanie 2. Model językowy (wariant Masked LM)

Zgadnij usunięte wyrazy:

> *Niektóre ptaki , szczególnie krukowate i papugowe należą do najbardziej inteligentnych gatunków zwierząt , zdolnych do tworzenia i używania przyrządów pomocniczych , jak i przekazujących tę wiedzę następnym pokoleniom .*

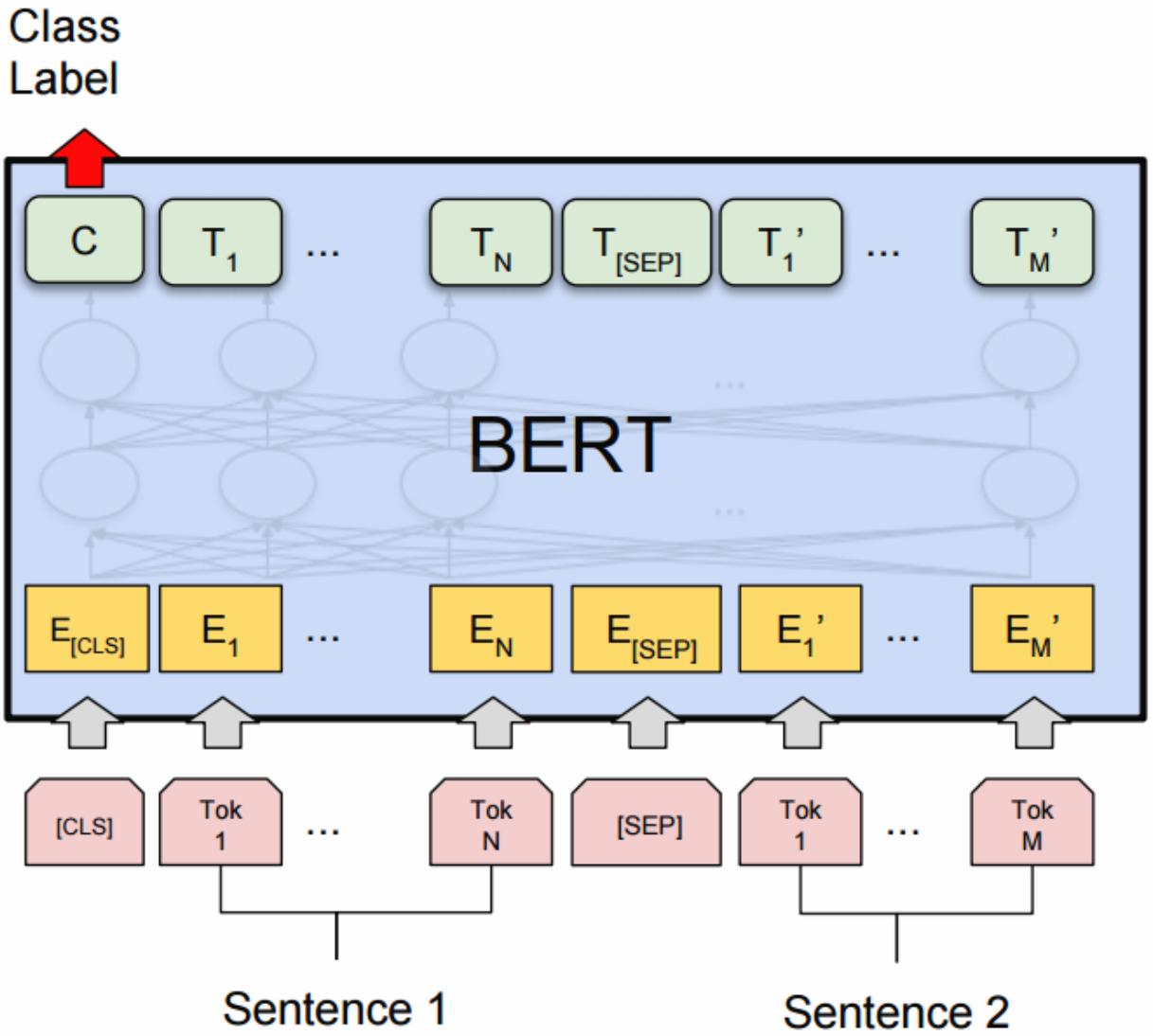
Czy umiemy jakoś te zadania rozwiązać?

Wejście dla BERT-a

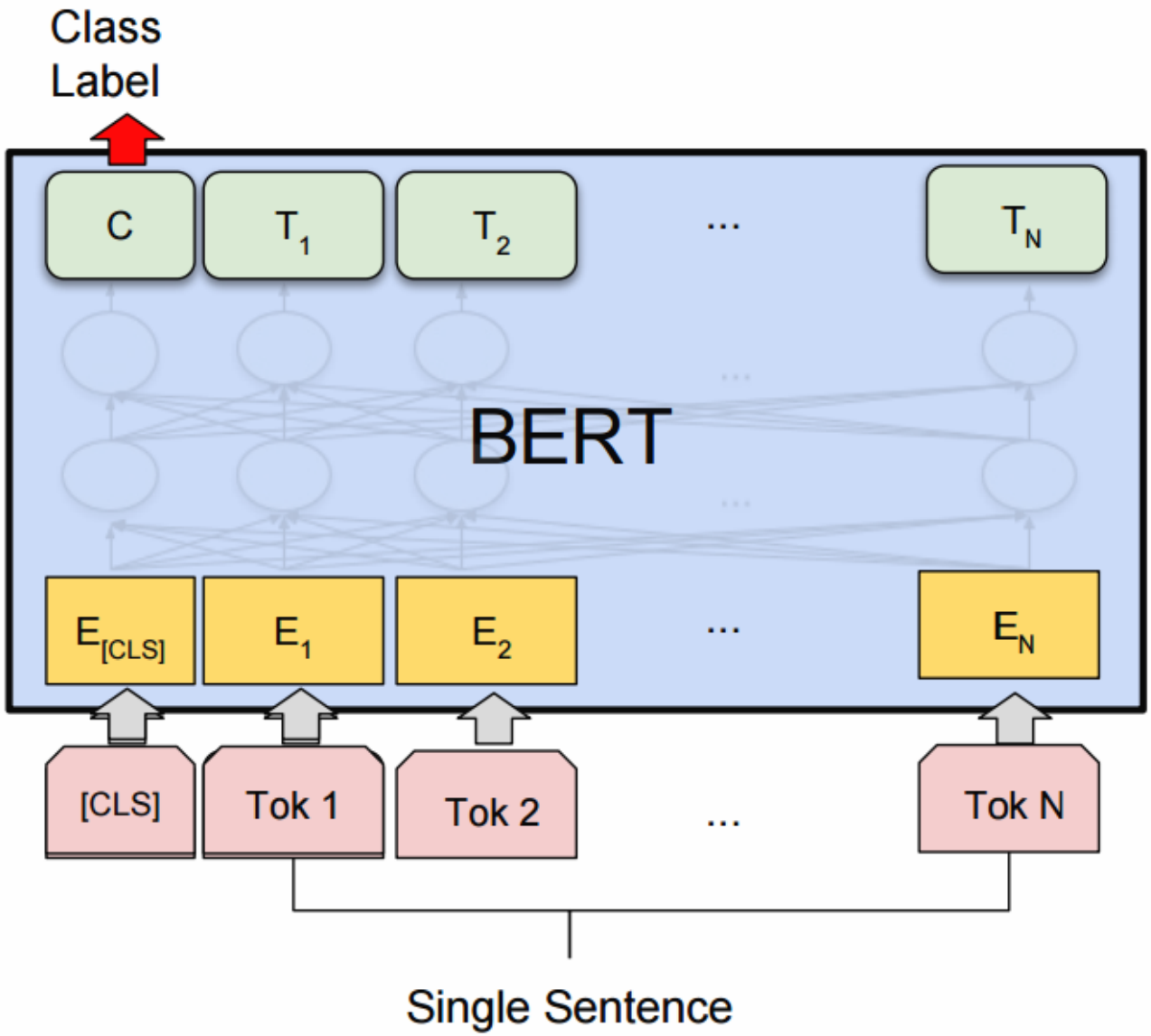


(Źródło tego obrazka i kolejnych: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Jacob Devlin i inni)

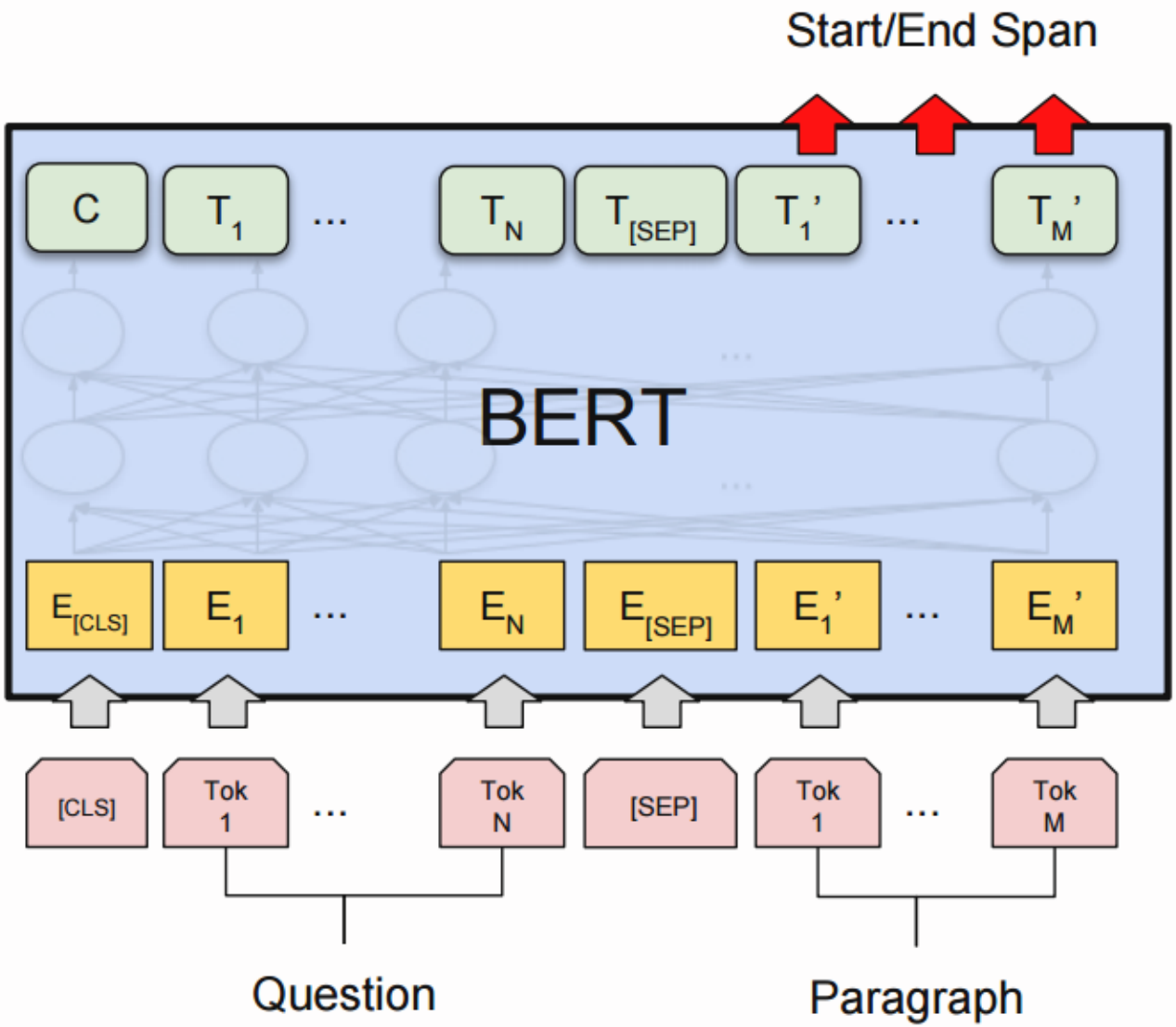
BERT w zadaniu klasyfikacji par zdań



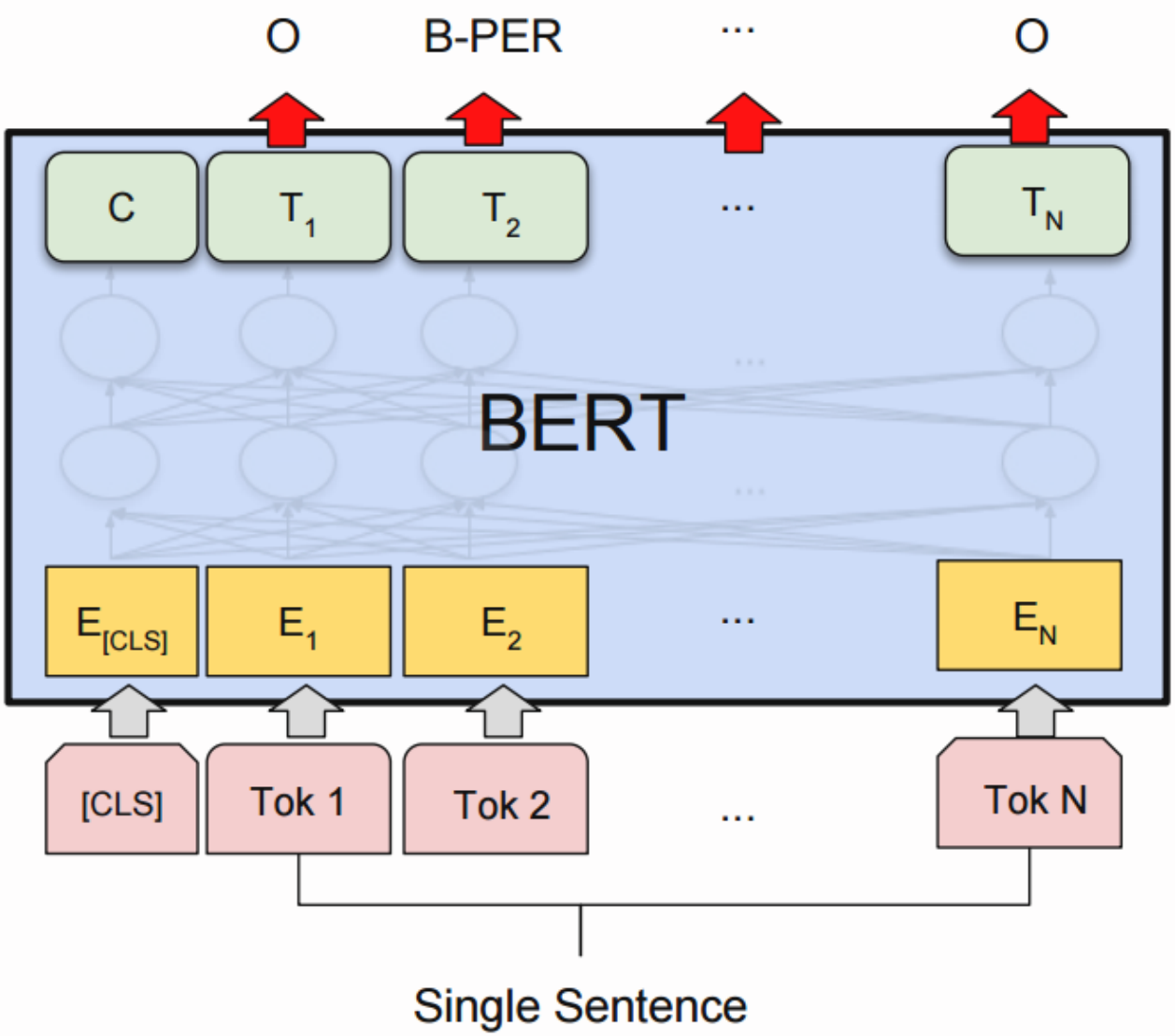
BERT w zadaniu klasyfikacji tekstu



BERT w zadaniu odpowiadania na pytania



BERT w zadaniu tagowania wyrazów



BERT – podsumowanie

- Najpierw uczymy na dużych danych (2 zadania podstawowe), potem douczamy na mniejszych, konkretnych.
- 4 doby uczenia, 16 TPU (dla ustalonej architektury)
- SotA dla bardzo wielu zadań z dziedziny NLP!

A tak przy okazji...

Transformer (czyli podstawa BERT-a) był użyty również w AlphaStar (i pokonał jakiegoś niezłego starcraftera, w wersji Protos vs Protos)

O metrykach

Definicja

Określeniem metryka w NLP oznaczamy metodę oceny rozwiązania zadania.

- Myślimy tu o ewaluacji wewnętrznej (instristic), czyli nieodwołującej się do innego zadania.
- Dwie naturalne metryki (o których mówiliśmy), to:
  1. Accuracy (dokładność)
  2. F_1-score (średnia harmoniczna precyzji i kompletności)

Prosta reguła

Nie używamy Accuracy gdy klasy są mocno niezbalansowane, i algorytm pt. zawsze dawaj najczęstszą miałby zbyt dobry wynik (przykład: dep. parsing bez etykiet, każda para słów to decyzja).

Skala przymiotnikowa

- Stosuje się też skalę przymiotnikową, do oceny wygenerowanych artefaktów
  - (na przykład w syntezie mowy, ale również w systemach dialogowych, czy, potencjalnie, generowaniu poezji)
- Przykład: Mean Opinion Score (MOS), w generacji dźwięku (oryginalnie: kodowanie):
  Stopnie (1-5): zła, słaba, średnia, dobra, znakomita
- Wyciągamy średnią z opinii ludzi oceniających wygenerowane wypowiedzi.

Wariant

Ludzie oceniają co lepsze. Czyli takie jakby A/B testy.

Korelacja

Przykładowe zadanie

Określić, jak bardzo dwa zdania mówią to samo (w jakim stopniu są swoimi parafrazami

- W danych uczących mamy dla każdej pary zdań liczbę od 1 do 5 (przykładowo)
- Często ocenia się korelacje ocen algorytmów z ocenami ludzi (nie skupiając się na poszczególnych trafieniach)

Word Error Rate

Zadanie

Modyfikujemy rekonstruktor polskawy, żeby rozważał również usuwanie i dodawanie spacji (ciąg wzorcowy i wynik algorytmu mają inną długość)

- Inny, bardziej typowy przykład to rozpoznawanie mowy (i sklejanie/rozklejanie wypowiedzianych słów)
- Powszechnie używana miara to Word Error Rate, czyli **odległość edycyjna** między wynikiem otrzymanym a wzorcowym, podzielona przez ... **długość wyniku wzorcowego**.
- Warianty: Phoneme Error Rate (PER), albo Character Error Rate (CER)

BLEU oraz inne francuskie kolorki

Zadanie

Podstawowym zadaniem jest tu **tłumaczenie maszynowe**.

Charakterystyka zadania:

1. Zdanie na wejściu, zdanie na wyjściu.
2. Możliwe wiele wzorcowych odpowiedzi, potencjalnie się od siebie różniących
3. Spodziewamy się, że wielu dobrych odpowiedzi nie będzie we wzorcach (więc chcemy nagradzać za częściową zgodność)

Inny przykład: system dialogowy (zwróćmy uwagę, że spodziewamy się istotnie gorszych ocen).

BLEU oraz inne francuskie kolorki

Pytanie: jak można by mierzyć jakość przekładu?

kandydat: *the the the the the the the*
wzorzec 1: *the cat is on the mat*
wzorzec 2: *there is a cat on the mat*

- Premiowanie (tylko) trafionych słów jest głupie.
- Pierwsza modyfikacja: bierzemy maksymalną liczność każdego słowa (tu 2) i tylko je liczymy (w naszym przykładzie otrzymamy $\frac{2}{7}$
- Uwzględniamy też N-gramy, im większe N, tym większa waga.
- Wybieramy zdanie referencyjne z największą punktacją.
- Oczywiście jest mnóstwo wariantów (na przykład ROUGE)

ROUGE

Miara ROUGE ma różne warianty:

- ROUGE-N – N-gram overlap (na przykład Jackard)
- ROUGE-S – premiuje powtarzanie się prawidłowych skip-gramów
- ROUGE-LCS – premiuje długie ciągi wspólne w tłumaczeniu otrzymanym i wzorcowym
- ROUGE-SU = ROUGE-S +ROUGE-1
- (...) – można sobie wyobrażać różne warianty łączące te klocki

Ogólnie: powszechny jest zarówno konsensus odnośnie stosowania tych miar, jak i ich krytyka :)