



UNIwersYTET
IM. ADAMA MICKIEWICZA
W POZNANIU

Modelowanie języka Sylabus zajęć

Informacje podstawowe

Kierunek studiów Informatyka Specjalność - Jednostka organizacyjna Wydział Matematyki i Informatyki Poziom studiów studia drugiego stopnia poinżynierskie Forma studiów studia stacjonarne Profil studiów profil ogólnoakademicki		Cykl dydaktyczny 2024/25 Kod zajęć 06INFS.44S.01045.24 Języki wykładowe polski Obligatoryjność Fakultatywny Blok zajęciowy Przedmioty specjalnościowe	
Koordynator zajęć	Filip Graliński		
Prowadzący zajęcia	Filip Graliński, Jakub Pokrywka		
Okres Semestr 3	Forma zajęć / liczba godzin / forma zaliczenia - Wykład: 30, Egzamin - Laboratorium: 30, Zaliczenie z oceną		Liczba punktów ECTS 6

Cele kształcenia dla zajęć

Kod	Cel
C1	Nabywanie znajomości podstawowych technik komputerowego przetwarzania tekstów pisanych w języku naturalnym.
C2	Zdobycie wiedzy teoretycznej i praktycznej na temat statystycznych modeli językowych.
C3	Poznanie modeli neuronowych typu "word2vec".
C4	Zdobycie wiedzy teoretycznej i praktycznej na temat neuronowych modeli językowych.
C5	Poznanie zastosowań modeli językowych.
C6	Zdobycie umiejętności wykorzystywania modeli językowych do różnych zastosowań przetwarzania języka naturalnego.

Wymagania wstępne

Podstawowa umiejętność programowania w języku Python.

Efekty uczenia się dla zajęć

Kod	Efekty uczenia się dla zajęć w zakresie	Efekty uczenia się dla kierunku	Metody weryfikacji osiągnięcia efektów uczenia się dla zajęć
Wiedzy - Student/ka:			
W1	Zna podstawowe techniki przetwarzania języka naturalnego (tokenizacja, lowercasing, itp.) i potrafi je zastosować. Potrafi „czyścić” dane tekstowe. Rozumie sposób reprezentacji tekstu w komputerze i problemy związane z wysoką wymiarowością reprezentacji liczbowej tekstu.	INF_K4_W02	Egzamin pisemny, Projekt
W2	Zna definicję modelu językowego, sposobów jego ewaluacji i zastosowania.	INF_K4_W02	Egzamin pisemny
W3	Potrafi wytłumaczyć sposób działania statystycznego modelu językowego. Rozumie pojęcie n-gramu. Rozumie sposób trenowania statystycznego modelu językowego.	INF_K4_W01	Egzamin pisemny
W4	Rozumie podstawy działania sieci neuronowych. Potrafi stworzyć prostą sieć neuronową przy wykorzystaniu biblioteki języka programowania.	INF_K4_W02	Egzamin pisemny, Projekt
W5	Zna modele typu „word2vec”. Potrafi wskazać zastosowania tych modeli, ich zalety i wady. Zna pojęcie reprezentacji numerycznej tekstu („embedding”) i potrafi wyjaśnić różnice reprezentacji rzadkiej i gęstej.	INF_K4_W02	Egzamin pisemny
W6	Potrafi wykorzystywać gotowe modele typu „word2vec”.	INF_K4_W04	Projekt
W7	Rozumie sposób działania neuronowego modelu językowego opartego o prostą sieć „feed-forward”.	INF_K4_W02	Egzamin pisemny
W8	Potrafi zaimplementować i wytrenować prosty model językowy oparty o sieć „feed-forward”.	INF_K4_W04	Projekt

Kod	Efekty uczenia się dla zajęć w zakresie	Efekty uczenia się dla kierunku	Metody weryfikacji osiągnięcia efektów uczenia się dla zajęć
W9	Rozumie sposób działania rekurencyjnego neuronowego modelu językowego. Zna mechanizm uwagi.	INF_K4_W02	Egzamin pisemny
W10	Potrafi zaimplementować i wytrenować rekurencyjny neuronowy model językowy.	INF_K4_W04	Projekt
W11	Rozumie sposób działania neuronowego modelu językowego opartego o sieć typu „transformer”.	INF_K4_W02	Egzamin pisemny
W12	Potrafi zaimplementować i wytrenować model językowy oparty o sieć typu „transformer” przy użyciu gotowych bibliotek.	INF_K4_W04	Projekt
W13	Zna różne algorytmy trenowania modelu językowego typu transformer („masked language model”, „next sentence prediction”). Zna pojęcia „pretraining”, „fine-tuning”, „transfer learning”.	INF_K4_W03	Egzamin pisemny
Umiejętności - Student/ka:			
U1	Potrafi stworzyć statystyczny model językowy bez użycia gotowych bibliotek. Potrafi stworzyć statystyczny model językowy przy pomocy gotowych bibliotek. Umie ewaluować modele językowe.	INF_K4_U01	Projekt
U2	Potrafi wykorzystać wytrenowany model językowy typu „transformer” dla problemu klasyfikacji.	INF_K4_U02	Projekt
U3	Zna alternatywne sposoby ewaluacji neuronowych modeli językowych na podstawie benchmarków „GLUE” i „SuperGLUE”. Potrafi wskazać różnice pomiędzy modelami językowymi typu „transformer”: „BERT”, „RoBERTa”, rodzina modeli „GPT”.	INF_K4_U05	Egzamin pisemny
U4	Potrafi generować tekst przy użyciu modelu językowego typu transformer.	INF_K4_U02	Projekt

Treści programowe dla zajęć

Lp.	Treści programowe dla zajęć	Efekty uczenia się dla zajęć	Formy zajęć
1.	Wprowadzenie do przetwarzania języka naturalnego. Przedstawienie podstawowych metod przetwarzania języka naturalnego. Zaznajomienie z problemami wynikającymi z pracy z tekstem.	W1	Wykład
2.	Zastosowanie metod przetwarzania tekstu w praktyce na podstawie implementacji klasyfikatora tekstu opartego o TF-IDF i regresję logistyczną.	W1	Laboratorium
3.	Modele językowe i ich ewaluacja	W2	Wykład
4.	Statystyczne modele językowe – sposób działania, trenowania, zalety i wady.	W3	Wykład
5.	Trenowanie statystycznego modelu językowego bez użycia gotowych bibliotek. Ewaluacja wytrenowanego modelu językowego	U1	Laboratorium

Lp.	Treści programowe dla zajęć	Efekty uczenia się dla zajęć	Formy zajęć
6.	Trenowanie statystycznego modelu językowego z wykorzystaniem biblioteki „KenLM”.	U1	Laboratorium
7.	Wprowadzenie do sieci neuronowych.	W4	Wykład
8.	Implementacja prostej sieci neuronowej przy wykorzystaniu biblioteki „pytorch”	W4	Laboratorium
9.	Wprowadzenie modeli typu „word2vec” – algorytm trenowania, zalety, wady.	W5	Wykład
10.	Wykorzystanie modeli typu „word2vec” w praktyce.	W6	Laboratorium
11.	Modele językowe oparte o sieć „feed-forward”.	W7	Wykład
12.	Implementacja modelu językowego opartego o sieć „feed-forward”.	W8	Laboratorium
13.	Modele językowe oparte o rekurencyjne sieci neuronowe, sieci typu „GRU” i „LSTM”. Mechanizm uwagi.	W9	Wykład
14.	Implementacja i trenowanie modelu językowego opartego o sieć rekurencyjną przy użyciu biblioteki „pytorch”. Implementacja i trenowanie modelu językowego opartego o sieć rekurencyjną z mechanizmem uwagi przy użyciu biblioteki „pytorch”.	W10	Laboratorium
15.	Modele językowe oparte o sieci neuronowe, sieci typu transformer – wprowadzenie.	W11	Wykład
16.	Implementacja i trenowanie modelu językowego opartego o sieć typu „transformer” przy użyciu biblioteki „pytorch”.	W12	Laboratorium
17.	Modele językowe oparte o sieci neuronowe, sieci typu transformer – metody zaawansowane, zastosowania.	W13	Wykład
18.	Klasyfikacja tekstu za pomocą sieci typu „transformer” przy wykorzystaniu biblioteki fairseq.	U2	Laboratorium
19.	Modele językowe oparte o sieci neuronowe typu transformer- ewaluacja, alternatywne sposoby ewaluacji modeli. Omówienie różnych modeli typu „transformer” („BERT”, „RoBERTa”, rodzina „GPT”).	U3	Wykład
20.	Generowanie tekstu przy użyciu modelu językowego GPT2.	U4	Laboratorium

Informacje dodatkowe

Forma zajęć	Metody i formy prowadzenia zajęć
Wykład	Wykład z prezentacją multimedialną wybranych zagadnień
Laboratorium	Metoda laboratoryjna, Metoda projektu

Forma zajęć	Warunki zaliczenia zajęć
Wykład	Warunkiem przystąpienia do egzaminu jest uzyskanie zaliczenia z laboratoriów. Na końcową ocenę składa się wynik uzyskany na egzaminie. Skala ocen: 1. bardzo dobry (bdb; 5,0) – od 90% punktów, 2. dobry plus (db plus; 4,5) – od 80% punktów, 3. dobry (db; 4,0) – od 70% punktów, 4. dostateczny plus (dst plus; 3,5) – od 60% punktów, 5. dostateczny (dst; 3,0) – od 50% punktów, 6. niedostateczny (ndst; 2,0) – poniżej 50% punktów.
Laboratorium	Na końcową ocenę składa się wynik uzyskany z projektu. Skala ocen: 1. bardzo dobry (bdb; 5,0) – od 90% punktów, 2. dobry plus (db plus; 4,5) – od 80% punktów, 3. dobry (db; 4,0) – od 70% punktów, 4. dostateczny plus (dst plus; 3,5) – od 60% punktów, 5. dostateczny (dst; 3,0) – od 50% punktów, 6. niedostateczny (ndst; 2,0) – poniżej 50% punktów.

Literatura

Obowiązkowa

1. Philipp Koehn. 2010. Statistical Machine Translation. Cambridge University Press. Cambridge.
2. Michael Nelsen. Neural Networks and Deep Learning. <http://neuralnetworksanddeeplearning.com/index.html> (dostęp 30.10.2020)
3. Yoshua Bengio, Réjean Ducharme, Pascal Vincent, Christian Jauvin. 2003. A Neural Probabilistic Language Model. Journal of Machine Learning Research 3 (2003) 1137-1155.
4. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. Advances in Neural Information Processing Systems 26 (NIPS 2013)
5. Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan "Honza" Cernocky, Sanjeev Khudanpur. 2010. Recurrent neural network based language model. INTERSPEECH 2010.
6. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. 2017. Attention Is All You Need. 31st Conference on Neural Information Processing Systems (NIPS 2017).
7. Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. North American Association for Computational Linguistics (NAACL).
8. Jay Alamar. The Illustrated Transformer. <http://jalammar.github.io/illustrated-transformer/> (dostęp 30.10.2020)

Dodatkowa

1. Wang et al. „GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding”
2. Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, Samuel R. Bowman. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. ICLR 2019.
3. Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners.

Nakład pracy studenta i punkty ECTS

Rodzaje zajęć studenta	Średnia liczba godzin* przeznaczonych na zrealizowane rodzaje zajęć
Wykład	30
Laboratorium	30
Przygotowanie do zajęć	20

Czytanie wskazanej literatury	10
Przygotowanie projektu	40
Inne	20
Łączny nakład pracy studenta	Liczba godzin 150
Liczba punktów ECTS	ECTS 6

* godzina (lekcyjna) oznacza 45 minut

Efekty uczenia się dla kierunku

Kod	Treść
INF_K4_U01	Absolwent/ka potrafi zastosować zaawansowaną wiedzę matematyczną do formułowania, analizowania i rozwiązywania złożonych i nietypowych zadań związanych z informatyką
INF_K4_U02	Absolwent/ka potrafi adaptować istniejące oraz tworzyć nowe metody informatyczne do rozwiązywania nieszablonowych problemów praktycznych i teoretycznych
INF_K4_U05	Absolwent/ka potrafi formułować i testować nowe algorytmy i metody rozwiązywania problemów w wybranych obszarach informatyki na potrzeby prowadzenia prac badawczo-rozwojowych z uwzględnieniem aktualnego stanu wiedzy
INF_K4_W01	Absolwent/ka zna i rozumie w pogłębionym stopniu pojęcia z działów matematyki niezbędne do rozwiązywania zaawansowanych problemów w informatyce
INF_K4_W02	Absolwent/ka zna i rozumie współczesny stan badań i tendencje rozwojowe w wiodących obszarach informatyki
INF_K4_W03	Absolwent/ka zna i rozumie w pogłębionym stopniu współczesne metody, narzędzia i technologie informatyczne właściwe dla wybranych obszarów zastosowań niezbędne przy budowie złożonych systemów informatycznych oraz przy prowadzeniu prac badawczo-rozwojowych
INF_K4_W04	Absolwent/ka zna i rozumie zasady rozwiązywania problemów z wykorzystaniem zaawansowanych algorytmów i metod informatycznych