Klasyfikacja tekstu

Zaawansowane Przetwarzanie Języka Naturalnego

Mateusz Lango

Zakład Inteligentnych Systemów Wspomagania Decyzji
Wydział Informatyki i Telekomunikacji
Politechnika Poznańska

„Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI Tech)”,
projekt finansowany ze środków Programu Operacyjnego Polska Cyfrowa POPC.03.02.00-00-0001/20

Fundusze Europejskie
Polska Cyfrowa

Rzeczpospolita Polska

Unia Europejska
Europejski Fundusz Rozwoju Regionalnego

Ostatnio na Zaawansowanym Przetwarzaniu Języka Naturalnego...

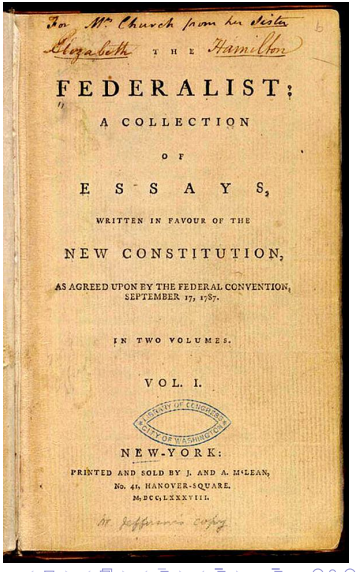
- Zastąpienie bezpośrednich estymat prawdopodobieństwa poprzez systemy uczące się
- Estymacja prawdopodobieństwa dla n-wyrazowego zdania to ile klasyfikacji?
- Uogólnienie modelu klasowego na neuronowy model autoregresywny
- Wprowadzenie macierzy zanurzeń C jako ciągłego odpowiednika dyskretnych klas $C(w)$
- Różne techniki przyśpieszania warstwy softmax, wszechobecnej w klasyfikacji (tekstu)

Agenda

1. Naiwny klasyfikator Bayesa
2. Inżynieria cech dla tekstu
3. Sieci splotowe do klasyfikacji tekstu

Klasyfikacja tekstów

- Klasyfikacja tekstów to problem automatycznego przypisania określonej wartości zmiennej nominalnej do danych tekstowych
- Zastosowania:
  - filtry SPAM
  - automatyczny przydział zgłoszeń do odpowiednich działów (organizacja tzw. „tickets")
  - ustalanie autorstwa
  - identyfikacja języka
  - rekrutacja pracowników
  - automatyczna organizacja dokumentów do zdefiniowanej struktury (kategorii)
  - ocena przejrzystości/dostępności tekstu
  - ...



Najbardziej klasyczny algorytm: Naiwny Bayes

Zwykła postać Naiwnego Bayesa:

$$P(y|\mathrm{x}) = \frac{P(\mathrm{x}|y)P(y)}{\sum_{y \in Y} P(\mathrm{x}|y)P(y)} = \frac{\prod_{i=1}^{d} P(x_i|y)P(y)}{\sum_{y \in Y} \prod_{i=1}^{d} P(x_i|y)P(y)} \propto \prod_{i=1}^{d} P(x_i|y)P(y)$$

Dla tekstów:

$$P(y|\mathrm{w}_1^n) \propto \prod_{i=1}^{n} P(w_i|y)P(y) = P(w_1|y)P(w_2|y)\cdots P(w_n|y)P(y)$$

Najbardziej klasyczny algorytm: Naiwny Bayes

Zwykła postać Naiwnego Bayesa:

$$P(y|\mathrm{x}) = \frac{P(\mathrm{x}|y)P(y)}{\sum_{y \in Y} P(\mathrm{x}|y)P(y)} = \frac{\prod_{i=1}^{d} P(x_i|y)P(y)}{\sum_{y \in Y} \prod_{i=1}^{d} P(x_i|y)P(y)} \propto \prod_{i=1}^{d} P(x_i|y)P(y)$$

Dla tekstów:

$$P(y|\mathrm{w}_1^n) \propto \prod_{i=1}^{n} P(w_i|y)P(y) = P(w_1|y)P(w_2|y)\cdots P(w_n|y)P(y)$$

Najbardziej klasyczny algorytm: Naiwny Bayes

Zwykła postać Naiwnego Bayesa:

$$P(y|\mathrm{x}) = \frac{P(\mathrm{x}|y)P(y)}{\sum_{y \in Y} P(\mathrm{x}|y)P(y)} = \frac{\prod_{i=1}^{d} P(x_i|y)P(y)}{\sum_{y \in Y} \prod_{i=1}^{d} P(x_i|y)P(y)} \propto \prod_{i=1}^{d} P(x_i|y)P(y)$$

Dla tekstów:

$$P(y|\mathrm{w}_1^n) \propto \prod_{i=1}^{n} P(w_i|y)P(y) = \underbrace{P(w_1|y)P(w_2|y)\cdots P(w_n|y)}_{\text{unigramowy, warunkowy model języka}} P(y)$$

Warunkowy model języka

- Warunkowy model języka to... model języka z częścią warunkową.

$$P(w_1^n|y)$$

Wiele problemów możemy modelować jako warunkowe modele języka.

- Generacja podpisu pod obrazkiem
- Tłumaczenie maszynowe
- Podsumowywanie tekstu
- Agent dialogowy
- ... i można go też użyć do klasyfikacji

Inne klasyfikatory

- Skoro Naiwny Bayes używa unigramowego modelu języka - czy nie można od ręki zaproponować innych, analogicznych klasyfikatorów?

$$P(y|w_1^n) \propto P(w_i^n|y)P(y)$$

gdzie $P(w_i^n|y)$ to dowolny model języka wytrenowany osobno na tekstach danej klasy
- Do klasyfikacji tekstów można wykorzystać w zasadzie dowolny algorytm uczący:
  - drzewa decyzyjne (i ich złożenia)
  - klasyfikator najbliższych sąsiadów
  - maszyny wektorów podpierających
  - sieci neuronowe
- Naiwny Bayes często jest mocnym rozwiązaniem bazowym dla „klasycznych" problemów analizy tekstów np. Apache SpamAssassin

Problem

Przy omawianiu algorytmu Naiwnego Bayesa czasami słyszy się, że założenie o warunkowej niezależności cech jest „spełnione" w tekstach, co jest powodem jego dobrego działania w klasyfikacji tekstu. Czy jest to prawda?

Inne klasyfikatory

- Skoro Naiwny Bayes używa unigramowego modelu języka - czy nie można od ręki zaproponować innych, analogicznych klasyfikatorów?

$$P(y|w_1^n) \propto P(w_i^n|y)P(y)$$

gdzie $P(w_i^n|y)$ to dowolny model języka wytrenowany osobno na tekstach danej klasy

- Do klasyfikacji tekstów można wykorzystać w zasadzie dowolny algorytm uczący:
  - drzewa decyzyjne (i ich złożenia)
  - klasyfikator najbliższych sąsiadów
  - maszyny wektorów podpierających
  - sieci neuronowe
- Naiwny Bayes często jest mocnym rozwiązaniem bazowym dla „klasycznych" problemów analizy tekstów np. Apache SpamAssassin

Problem

Przy omawianiu algorytmu Naiwnego Bayesa czasami słyszy się, że założenie o warunkowej niezależności cech jest „spełnione" w tekstach, co jest powodem jego dobrego działania w klasyfikacji tekstu. Czy jest to prawda?

Inne klasyfikatory

- Skoro Naiwny Bayes używa unigramowego modelu języka - czy nie można od ręki zaproponować innych, analogicznych klasyfikatorów?

$$P(y|w_1^n) \propto P(w_i^n|y)P(y)$$

  gdzie $P(w_i^n|y)$ to dowolny model języka wytrenowany osobno na tekstach danej klasy
- Do klasyfikacji tekstów można wykorzystać w zasadzie dowolny algorytm uczący:
  - drzewa decyzyjne (i ich złożenia)
  - klasyfikator najbliższych sąsiadów
  - maszyny wektorów podpierających
  - sieci neuronowe
- Naiwny Bayes często jest mocnym rozwiązaniem bazowym dla „klasycznych" problemów analizy tekstów np. Apache SpamAssassin

Problem

Przy omawianiu algorytmu Naiwnego Bayesa czasami słyszy się, że założenie o warunkowej niezależności cech jest „spełnione" w tekstach, co jest powodem jego dobrego działania w klasyfikacji tekstu. Czy jest to prawda?

Inne klasyfikatory

- Skoro Naiwny Bayes używa unigramowego modelu języka - czy nie można od ręki zaproponować innych, analogicznych klasyfikatorów?

$$P(y|\mathrm{w}_1^n) \propto P(w_i^n|y)P(y)$$

  gdzie $P(w_i^n|y)$ to dowolny model języka wytrenowany osobno na tekstach danej klasy
- Do klasyfikacji tekstów można wykorzystać w zasadzie dowolny algorytm uczący:
  - drzewa decyzyjne (i ich złożenia)
  - klasyfikator najbliższych sąsiadów
  - maszyny wektorów podpierających
  - sieci neuronowe
- Naiwny Bayes często jest mocnym rozwiązaniem bazowym dla „klasycznych" problemów analizy tekstów np. Apache SpamAssassin

Problem

Przy omawianiu algorytmu Naiwnego Bayesa czasami słyszy się, że założenie o warunkowej niezależności cech jest „spełnione" w tekstach, co jest powodem jego dobrego działania w klasyfikacji tekstu. Czy jest to prawda?

Inne klasyfikatory

- Skoro Naiwny Bayes używa unigramowego modelu języka - czy nie można od ręki zaproponować innych, analogicznych klasyfikatorów?

$$P(y|\mathrm{w}_1^n) \propto P(w_i^n|y)P(y)$$

  gdzie $P(w_i^n|y)$ to dowolny model języka wytrenowany osobno na tekstach danej klasy
- Do klasyfikacji tekstów można wykorzystać w zasadzie dowolny algorytm uczący:
  - drzewa decyzyjne (i ich złożenia)
  - klasyfikator najbliższych sąsiadów
  - maszyny wektorów podpierających
  - sieci neuronowe
- Naiwny Bayes często jest mocnym rozwiązaniem bazowym dla „klasycznych" problemów analizy tekstów np. Apache SpamAssassin

Problem

Przy omawianiu algorytmu Naiwnego Bayesa czasami słyszy się, że założenie o warunkowej niezależności cech jest „spełnione" w tekstach, co jest powodem jego dobrego działania w klasyfikacji tekstu. Czy jest to prawda?

Inżynieria cech dla tekstu: „worek słów”

Korpus uczący

	Przykład	Klasa
1	He is a lazy boy. She is also lazy.	+
2	Jurek is a lazy person.	-
3	He is lazy	-
	...	

Macierz X

	she	he	Jurek	is	a	lazy	boy	person	also
1	1	1	0	2	1	2	1	0	1
2	0	0	1	1	1	1	0	1	0
3	0	1	0	1	0	1	0	0	0
					...				

Wada: nie zapisują porządku słów (model unigramowy)

Inżynieria cech dla tekstu: „worek słów”

Korpus uczący

	Przykład	Klasa
1	He is a lazy boy. She is also lazy.	+
2	Jurek is a lazy person.	-
3	He is lazy	-
	...	

Macierz X

	she	he	Jurek	is	a	lazy	boy	person	also
1	1	1	0	2	1	2	1	0	1
2	0	0	1	1	1	1	0	1	0
3	0	1	0	1	0	1	0	0	0
						...			

Wada: nie zapisują porządku słów (model unigramowy)

Inżynieria cech dla tekstu: „worek klas”

Korpus uczący

	Przykład	Klasa
1	He is a lazy boy. She is also lazy.	+
2	Jurek is a lazy person.	-
3	He is lazy	-
	...	

Inżynieria cech dla tekstu: „worek klas”

Korpus uczący

	Przykład	Klasa
1	C_1 is a lazy boy. C_1 is also lazy.	+
2	C_1 is a lazy person.	-
3	C_1 is lazy ...	-

Inżynieria cech dla tekstu: „worek klas”

Korpus uczący

	Przykład	Klasa
1	$C_1\ C_2\ C_3\ C_4\ C_5.$	+
	$C_1\ C_2\ C_6\ C_4.$	
2	$C_1\ C_2\ C_3\ C_4\ C_5.$	-
3	$C_1\ C_2\ C_4$	-
	$\ldots$	

Inżynieria cech dla tekstu: „worek klas”

Korpus uczący

	Przykład	Klasa
1	$C_1\ C_2\ C_3\ C_4\ C_5.$	+
	$C_1\ C_2\ C_6\ C_4.$	
2	$C_1\ C_2\ C_3\ C_4\ C_5.$	-
3	$C_1\ C_2\ C_4$	-
	...	

Macierz X

	C_1	C_2	C_3	C_4	C_5	C_6
1	2	2	1	2	1	1
2	1	1	1	1	1	0
3	1	1	0	1	0	0
			...			

Redukcja cech oraz potencjalnie lepsze uogólnianie wiedzy!

Inżynieria cech dla tekstu: „worek klas”

Korpus uczący

	Przykład	Klasa
1	$C_1\ C_2\ C_3\ C_4\ C_5.$	+
	$C_1\ C_2\ C_6\ C_4.$	
2	$C_1\ C_2\ C_3\ C_4\ C_5.$	-
3	$C_1\ C_2\ C_4$	-
	...	

Macierz X

	C_1	C_2	C_3	C_4	C_5	C_6
1	2	2	1	2	1	1
2	1	1	1	1	1	0
3	1	1	0	1	0	0
			...			

Redukcja cech oraz potencjalnie lepsze uogólnianie wiedzy!

Analogiczne reprezentacje: n-gramy słów, n-gramy klas

Podsumowanie: transfer wiedzy z klasowego modelu języka

1. Wytrenuj klasowy model języka (grupowanie Browna) – uczenie nienadzorowane
2. Zastąp słowa klasami i zbuduj klasyczną reprezentację worka słów
3. Wytrenuj klasyfikator – uczenie nadzorowane

Inżynieria cech dla tekstu: „worek klas”

Korpus uczący

	Przykład	Klasa
1	C_1 C_2 C_3 C_4 C_5.	+
	C_1 C_2 C_6 C_4.	
2	C_1 C_2 C_3 C_4 C_5.	-
3	C_1 C_2 C_4	-
	...	

Macierz X

	C_1	C_2	C_3	C_4	C_5	C_6
1	2	2	1	2	1	1
2	1	1	1	1	1	0
3	1	1	0	1	0	0
			...			

Redukcja cech oraz potencjalnie lepsze uogólnianie wiedzy!

Analogiczne reprezentacje: n-gramy słów, n-gramy klas

Podsumowanie: transfer wiedzy z klasowego modelu języka

1. Wytrenuj klasowy model języka (grupowanie Browna) – uczenie nienadzorowane
2. Zastąp słowa klasami i zbuduj klasyczną reprezentację worka słów
3. Wytrenuj klasyfikator – uczenie nadzorowane

Obsługa rzadkich słów i OOV

- Słowa w których przez przypadek lub celowo zrobiono literówkę nie mają gotowej reprezentacji (nie mają grupy ani miejsca w macierzy)
- Standardowe rozwiązania z modelowania języka: token UNK, pseudo-słowa
- Często agresywniejszy niż w modelowaniu języka pruning tokenów: dlaczego? Pomyśl o przeuczeniu!
- Przy budowaniu reprezentacji pomijamy również najczęstsze tokeny. Dlaczego?
- albo...

Inżynieria cech dla tekstu: „worek k-gramów”

Korpus uczący

	Przykład	Klasa
1	He is a lazy boy. She is also lazy.	+
2	Jurek is a lazy person.	-
3	He is lazy	-
	...	

Macierz X

	He␣	e␣i	␣is	is␣	s␣a	␣a␣	a␣l	␣la	laz	azy
1	1	2	2	2	2	1	1	2	2	2
2	0	0	1	1	1	1	1	1	1	1
3	1	1	1	1	0	0	0	1	1	1
					...					

Nie jest to standardowe, ale dla rozróżnienia: n-gramami nazywam zbitki słów, a k-gramami nazywam zbitki znaków.

Zalety: obsługa literówek, brak OOV, skończona przestrzeń, uwzględnianie które słowa stoją obok siebie, uogólnianie na odmienione słowa

Wady: mniej informatywne cechy, sensowne wartości k to zwykle > 4 a to już $30^5 = 24\,300\,000$ możliwych cech

k-gramy: ogólnie dość dobry pomysł dla systemów NLP

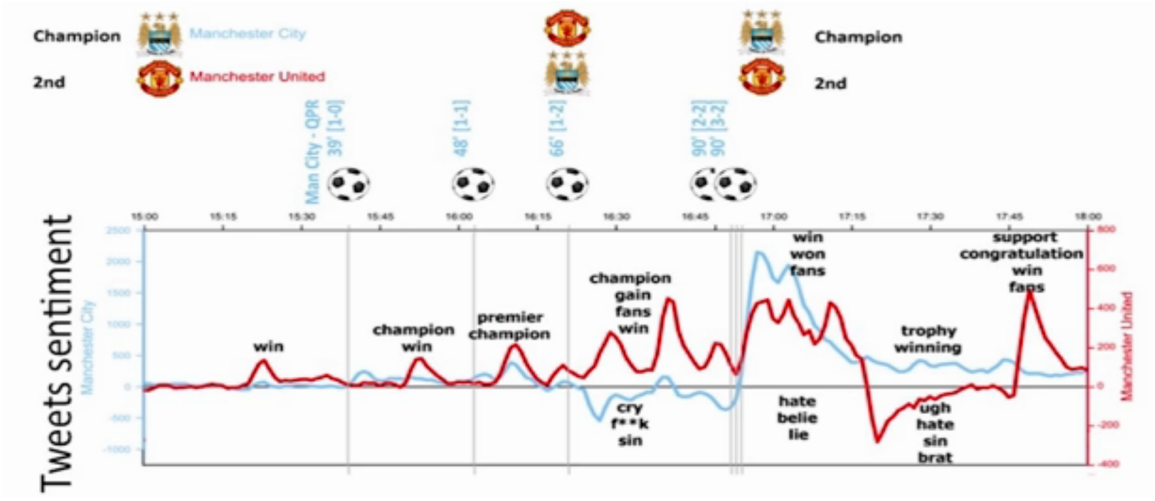
Any learning method powerful enough to understand the world by reading the web ought to find it trivial to learn which strings make words. – Geoffrey Hinton

- Etap wstępnego przetwarzania danych jest często bardzo złożony dla tekstów...
- Modelowanie morfologii fleksyjnej języka „przerabiam", ',przerabiasz",...
- Modelowanie morfologii derywacyjnej języka „prze-robić", „do-robić", „dorabiający"
- Obsługa nazw wieloczłonowych jak np. „Gazeta Wyborcza"
- Dużo prościej jest przetwarzać zmienną dyskretną z setką możliwości (znaki) niż milionem ($|V|$ słów)!

Przykład inżynierii cech: analiza wydźwięku w tweetach

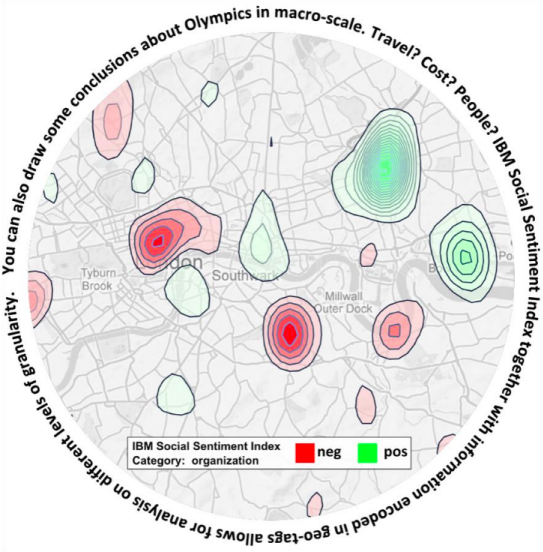
- Analiza wydźwięku to dział NLP zajmujący się wykrywaniem i analizą emocji zawartymi w tekście.
- Na dzisiejszych zajęciach będziemy zajmowali się *klasyfikacją wydźwięku* – czyli problemem klasyfikacji tekstu do klas: negatywny lub pozytywny.
- Alternatywy: skala 3-stopniowa (również klasa neutralny) lub skale od 0 do 5, od 0 do 10.
- Zastosowania
  - Badanie cech niemierzalnych produktów
  - Analizy politologiczne, reklamowe, ...
  - Ocena terapii psychoterapeutycznej

Emocje w Premier League



Slajd prof. Przemysława Biecka

Emocje na Igrzyskach Olimpijskich w Londynie



„Energy of the Nation" project

Analiza firmy IBM

Klasyfikacja wydźwięku

- Klasyfikacja wydźwięku to zwykła klasyfikacja tekstu – o co ten cały szum?

Korpus	TREC	SST-1
Ile klas?	6-klasowe	5-klasowe
Zadanie	przypisanie pytania do typu	wydźwięk
SVM	95% [1]	38%
CNN	93.6%	48%

[1] Dobrze dostrojone + ręcznie zrobione reguły w cechach. Wyniki: Kim (2014), Zhang (2016)

Klasyfikacja wydźwięku: źródła trudności

- Konieczne modelowanie pewnych zjawisk lingwistycznych np. kluczowa rola negacji:
  - „Perhaps one of the most important works of science fiction of the year ... 1Q84 does not **disappoint** ... " –Matt Staggs, Suvudu.com
  - negacja nie zawsze odwraca wydźwięk!
    - great [silnie pozytywne] → not great [słabo negatywne]
    - terrible [silnie negatywne] → not terrible [słabo negatywnie]
- Duża rola kontekstu wypowiedzi
  - „Jestem z tego dumny"vs "Jesteś z tego dumny???"
  - „Schudłem 5 kilogramów"(choroba vs dieta)

A:	Yoga helps make my body flexible, lean & slim.	
A:	**After work overtime for 3 dys, I lose 3 pounds!**	(+)
B:	*I lose 5!*	(+)
C:	Getting poor feedback on a project where you are getting paid very little money for a lot of work.	
C:	**After work overtime for 3 dys, I lose 3 pounds!**	(-)
D:	*I lose 5!*	(-)

Klasyfikacja wydźwięku: źródła trudności

- Konieczne modelowanie pewnych zjawisk lingwistycznych np. kluczowa rola negacji:
  - „Perhaps one of the most important works of science fiction of the year ... 1Q84 **does not disappoint** ... " –Matt Staggs, Suvudu.com
  - negacja nie zawsze odwraca wydźwięk!
    - great [silnie pozytywne] → not great [słabo negatywne]
    - terrible [silnie negatywne] → not terrible [słabo negatywnie]
- Duża rola kontekstu wypowiedzi
  - „Jestem z tego dumny"vs "Jesteś z tego dumny???"
  - „Schudłem 5 kilogramów"(choroba vs dieta)

A:	Yoga helps make my body flexible, lean & slim.	
A:	**After work overtime for 3 dys, I lose 3 pounds!**	(+)
B:	*I lose 5!*	(+)
C:	Getting poor feedback on a project where you are getting paid very little money for a lot of work.	
C:	**After work overtime for 3 dys, I lose 3 pounds!**	(-)
D:	*I lose 5!*	(-)

Analiza wydźwięku w tweetach – potok przetwarzania

- normalizacja
- tokenizacja
- lematyzacja
- usuwanie słów nieinformatywnych (ang. *stop words*)
- usuwanie rzadkich tokenów np. < 5 wystąpień
- grupowanie rzadkich tokenów w pseudo-słowa

Problem

W stosunku do standardowych implementacji tych operacji, jakie zmiany powinny zostać wprowadzone dla analizy wydźwięku w tweetach?

Analiza wydźwięku w tweetach – potok przetwarzania

- normalizacja $\Rightarrow$ np. konwersja „ " czy „<"; wielkie litery NIEEEE!
- tokenizacja $\Rightarrow$ przed hashtagami czy emotikonami często nie ma spacji
- lematyzacja $\Rightarrow$ wzbogacona o korektor pisowni
- usuwanie słów nieinformatywnych (ang. *stop words*) $\Rightarrow$ „not" jest często słowem nieinformatywnym, tutaj jest kluczowe
- usuwanie rzadkich tokenów np. < 5 wystąpień
- grupowanie rzadkich tokenów w pseudo-słowa $\Rightarrow$ specjalne klasy URL, HASHTAG, USERREF

Problem

W stosunku do standardowych implementacji tych operacji, jakie zmiany powinny zostać wprowadzone dla analizy wydźwięku w tweetach?

Analiza wydźwięku w tweetach – potok przetwarzania (2)

Popularne cechy

- n-gramy
- k-gramy
- n-gramy części mowy ($\Rightarrow$ kolejne wykłady)
- przedłużone słowa („baaaaardzo")
- emotikony
- interpunkcja („!!!!")
- słowa pisane drukowanymi literami (BARDZO)
- leksykony emocji (SentiWordNet, Opinion Lexicon, Multi-perspective Question Answering, NRC, ...)

Zanegowane n-gramy

Próba modelowania negacji w przestrzeni cech

- Ręcznie zdefiniowana lista słów negujących: not, never, none, nobody, nowhere, neither
- Cecha jest tworzona dla *n*-gramu w kontekście afirmatywnym i zanegowanym (2 cechy per jeden *n*-gram)
- Kontekst zanegowany zaczyna się od słowa występującego po słowie negującym aż do następnego znaku interpunkcyjnego

The voice quality of this phone is not good, but the battery life is long

The room was very nicely appointed and the bed was sooo comfortable. Even though the bathroom door did not close all the way, it was still pretty private.

Ważność cech w przykładowym systemie analizy wydźwięku

Feature group	Rel. impor. [%]
5 character-gram	26.03
4 character-gram	21.75
3 character-gram	21.74
Brown clusters	6.92
Negated 1-gram	6.62
1-gram + POS	4.24
Negated + 2-gram	3.48
1-gram	2.69
2-gram	1.87
NRC Hashtag Lexicon	1.49
SentiWordNet	1.00
NRC Lexicon	0.93
Opinion Lexicon	0.62
3-gram	0.34

- Po lewej wyniki dla drzew wzmacnianych gradientowo na korpusie tweetów z zadania domowego
- Główne wnioski z tej sekcji:
  - każde dane są inne, specyfika zadania klasyfikacji tekstu może być różna
  - analiza autorstwa, identyfikacja języka również są rozpatrywane jako osobne problemy niż „klasyfikacja tekstu"
  - choć mamy pewne wzorce ogólnych cech jak np. *n*-gramy czy grupy Browna
  - zaprojektowanie wydajnych cech wymaga zbadania problemu, analizy danych i pobrudzenia sobie rączek...

W neuronowym świecie...

- Sieć neuronowa to zwykły klasyfikator – znów możemy wykorzystać cechy worka słów i pozostałe
- Są jednak dwa tematy które mogą być w specjalny sposób obsłużone:
  - W klasycznych klasyfikatorach mogliśmy użyć reprezentacji klas i worka słów. To samo chcielibyśmy zrobić z „klasowym" modelem neuronowym, jednak to sprawia że każde słowo zamieniane jest na *wektor*... i co dalej?

Transfer wiedzy z neuronowego modelu języka

1. Wytrenuj ~~klasowy~~ *neuronowy* model języka *z macierzą zanurzeń* ~~(grupowanie Browna)~~ – uczenie nienadzorowane
2. Zastąp słowa ~~klasami~~ *zanurzeniami* i zbuduj ~~klasyczną~~ reprezentację ~~worka słów~~ (jak to zrobić ???)
3. Wytrenuj klasyfikator – uczenie nadzorowane

- W klasycznych klasyfikatorach, aby przejść z worka słów na worki *n*-gramów potrzebowaliśmy wykładniczej liczby cech, choć pewnie tylko niektóre z nich są przydatne do klasyfikacji. Czy NN może nauczyć się samodzielnie decydować o ekstrakcji odpowiednich *n*-gramów?

W neuronowym świecie...

- Sieć neuronowa to zwykły klasyfikator – znów możemy wykorzystać cechy worka słów i pozostałe
- Są jednak dwa tematy które mogą być w specjalny sposób obsłużone:
  - W klasycznych klasyfikatorach mogliśmy użyć reprezentacji klas i worka słów. To samo chcielibyśmy zrobić z „klasowym" modelem neuronowym, jednak to sprawia że każde słowo zamieniane jest na *wektor*... i co dalej?

Transfer wiedzy z neuronowego modelu języka

1. Wytrenuj ~~klasowy~~ *neuronowy* model języka *z macierzą zanurzeń* ~~(grupowanie Browna)~~ – uczenie nienadzorowane
2. Zastąp słowa ~~klasami~~ *zanurzeniami* i zbuduj ~~klasyczną~~ reprezentację ~~worka słów~~ (jak to zrobić ???)
3. Wytrenuj klasyfikator – uczenie nadzorowane

- W klasycznych klasyfikatorach, aby przejść z worka słów na worki n-gramów potrzebowaliśmy wykładniczej liczby cech, choć pewnie tylko niektóre z nich są przydatne do klasyfikacji. Czy NN może nauczyć się samodzielnie decydować o ekstrakcji odpowiednich n-gramów?

Reprezentacja ciągłego worka słów

- Naturalnym rozszerzeniem reprezentacji worka słów do reprezentacji słów macierzą zanurzeń to reprezentacja ciągłego worka słów (ang. *continous bag-of-words*, *CBOW*, *neural bag-of-words*, *NBOW*)

$$\mathrm{x} = \frac{1}{n} \sum_{i}^{n} \mathrm{C}_{w_i}$$

- Alternatywnie możemy użyć postaci ważonej (ang. *weighted CBOW*, *WCBOW*)

$$\mathrm{x} = \frac{1}{\sum_{j=1}^{n} r_j} \sum_{i=1}^{n} r_i \mathrm{C}_{w_i}$$

  gdzie wagi r_i są określane np. poprzez TF-IDF
- Taką reprezentację możemy wykorzystać dla *dowolnego* klasyfikatora
- Dlaczego jest to „naturalne" rozszerzenie worka słów?

Reprezentacja worka słów jako suma reprezentacji słów

Korpus uczący

	Przykład	Klasa
1	He is a lazy boy. She is also lazy.	+
2	Jurek is a lazy person.	-
3	He is lazy ...	-

Reprezentacja dla pierwszego przykładu

	she	he	Jurek	is	a	lazy	boy	person	also
$\mathbb{1}_{He}$	0	1	0	0	0	0	0	0	0
$\mathbb{1}_{is}$	0	0	0	1	0	0	0	0	0
$\mathbb{1}_{a}$	0	0	0	0	1	0	0	0	0
$\mathbb{1}_{lazy}$	0	0	0	0	0	1	0	0	0
$\mathbb{1}_{boy}$	0	0	0	0	0	0	1	0	0
$\mathbb{1}_{She}$	1	0	0	0	0	0	0	0	0
	...								
$\sum$	1	1	0	2	1	2	1	0	1

Reprezentacje powstają poprzez sumę/średnią reprezentacji poszczególnych słów!

Reprezentacja worka słów jako suma reprezentacji słów

Korpus uczący

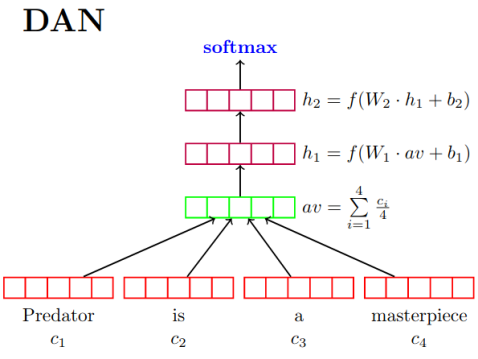
	Przykład	Klasa
1	He is a lazy boy. She is also lazy.	+
2	Jurek is a lazy person.	-
3	He is lazy	-
	...	

Reprezentacja dla pierwszego przykładu

	d_1	d_2	d_3
C_{He}	0.5	-0.2	1.8
C_{is}	-0.33	-0.33	0.1
C_{a}	0.6	-0.6	-0.75
C_{lazy}	-2.5	-0.12	0.98
C_{boy}	0.1	-0.7	0.8
C_{She}	0.51	-0.2	1.5
	...		
$\frac{1}{n}\sum$	0.25	-0.9	0.5

Reprezentacje powstają poprzez sumę reprezentacji słów!

Głęboka Sieć Uśredniająca (ang. *Deep Averaging Network*)



- Głęboka sieć uśredniająca służy do klasyfikacji, ale także do uzyskiwania reprezentacji zdania (po odcięciu ostatniego softmaxa)
- Motywacja: klasyfikator liniowy po CBOW nie uczy się kombinacji cech, więc musimy je wytworzyć dodatkowymi warstwami sieci
- DAN pozwala na utworzenie bardziej złożonych, nieliniowych cech

Technika Word Dropout

- Nieznane słowa zwykle zastępujemy tokenami UNK – w jaki sposób je estymujemy na zbiorze uczącym?
- Zastępując część wyrazów w korpusie poprzez UNK tracimy wiedzę
- Pomysł: w trakcie uczenia losowo zastępujemy słowa UNK uzyskując efekt regularyzacji modelu
- Usuwanie słów odbywa się w trakcie uczenia (w ramach paczki danych) a nie globalnie w ramach zbioru uczącego
- Usuwamy słowo przeciwnie z prawdopodobieństwem jego występowania w zbiorze uczącym

$$\frac{\alpha}{c(w) + \alpha}$$

gdzie α to parametr metody.

- Liczba przykładów uczących... urosła (prawie) wykładniczo...
- „ we always see improvements using this technique" (Iyyer et al., ACL'15)

Technika Word Dropout

- Nieznane słowa zwykle zastępujemy tokenami UNK – w jaki sposób je estymujemy na zbiorze uczącym?
- Zastępując część wyrazów w korpusie poprzez UNK tracimy wiedzę
- Pomysł: w trakcie uczenia losowo zastępujemy słowa UNK uzyskując efekt regularyzacji modelu
- Usuwanie słów odbywa się w trakcie uczenia (w ramach paczki danych) a nie globalnie w ramach zbioru uczącego
- Usuwamy słowo przeciwnie z prawdopodobieństwem jego występowania w zbiorze uczącym

$$\frac{\alpha}{c(w) + \alpha}$$

  gdzie α to parametr metody.
- Liczba przykładów uczących... urosła (prawie) wykładniczo...
- „ we always see improvements using this technique" (Iyyer et al., ACL'15)

Głęboka Sieć Uśredniająca (ang. *Deep Averaging Network*)

Model	RT	SST fine	SST bin	IMDB	Time (s)
DAN-ROOT	—	46.9	85.7	—	**31**
DAN-RAND	77.3	45.4	83.2	88.8	136
DAN	80.3	47.7	86.3	89.4	136
NBOW-RAND	76.2	42.3	81.4	88.9	91
NBOW	79.0	43.6	83.6	89.0	91
BiNB	—	41.9	83.1	—	—
NBSVM-bi	79.4	—	—	91.2	—
RecNN*	77.7	43.2	82.4	—	—
RecNTN*	—	45.7	85.4	—	—
DRecNN	—	49.8	86.6	—	431
TreeLSTM	—	**50.6**	86.9	—	—
DCNN*	—	48.5	86.9	89.4	—
PVEC*	—	48.7	87.8	**92.6**	—
CNN-MC	**81.1**	47.4	**88.1**	—	2,452
WRRBM*	—	—	—	89.2	—

Głęboka Sieć Uśredniająca (ang. *Deep Averaging Network*)

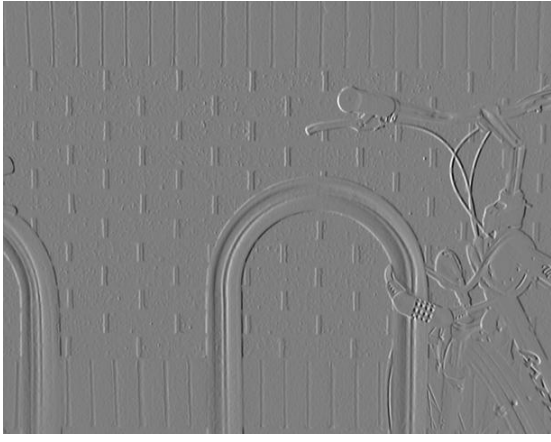
Sentence	DAN	DRecNN	Ground Truth
a lousy movie that's not merely unwatchable, but also unlistenable	negative	negative	negative
if you're not a prepubescent girl, you'll be laughing at britney spears' movie-starring debut whenever it does n't have you impatiently squinting at your watch	negative	negative	negative
blessed with immense physical prowess he may well be, but ahola is simply not an actor	positive	neutral	negative
who knows what exactly godard is on about in this film, but his words and images do n't have to add up to mesmerize you.	positive	positive	positive
it's so good that its relentless, polished wit can withstand not only inept school productions, but even oliver parker's movie adaptation	negative	positive	positive
too bad, but thanks to some lovely comedic moments and several fine performances, it's not a total loss	negative	negative	positive

Splot - przykład

$$G_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix}$$

Splot - przykład

$$G_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix}$$

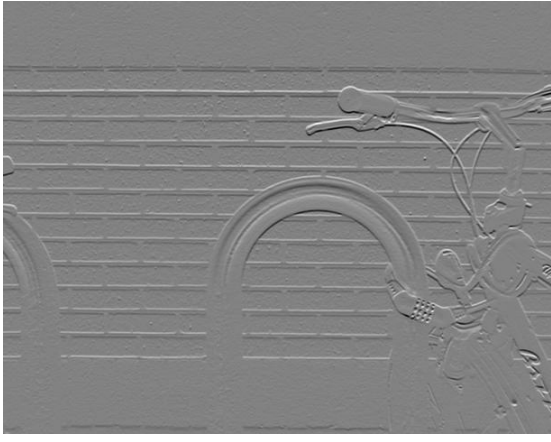


Splot - przykład

$$G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Splot - przykład

$$G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

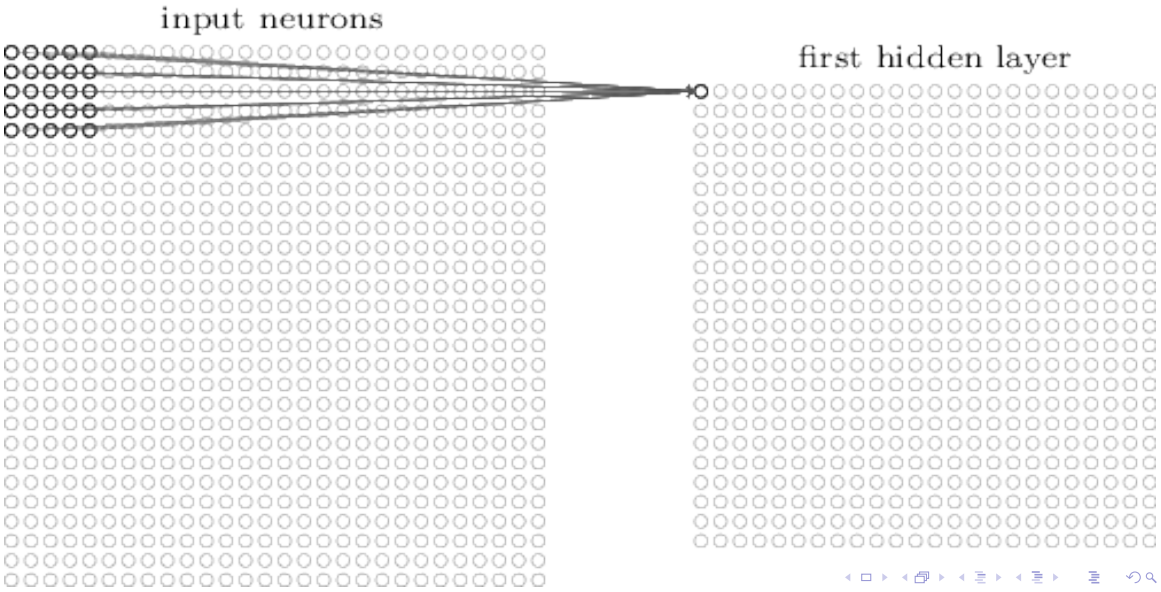


Filtr Sobela

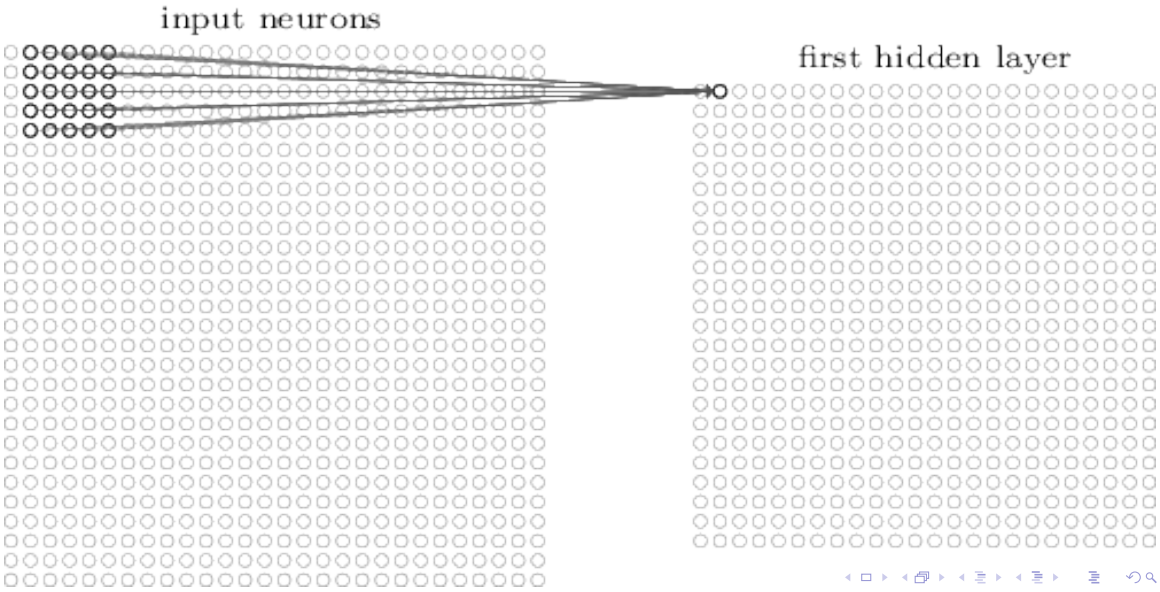
Zastosuj oba wcześniejsze filtry a potem „uśrednij”

$$\sqrt{x^2 + y^2}$$

Sieć splotowa

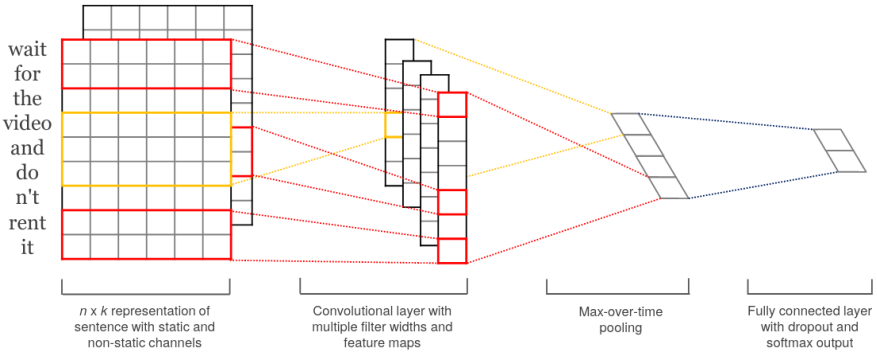


Sieć splotowa



Sieć splotowa dla tekstów

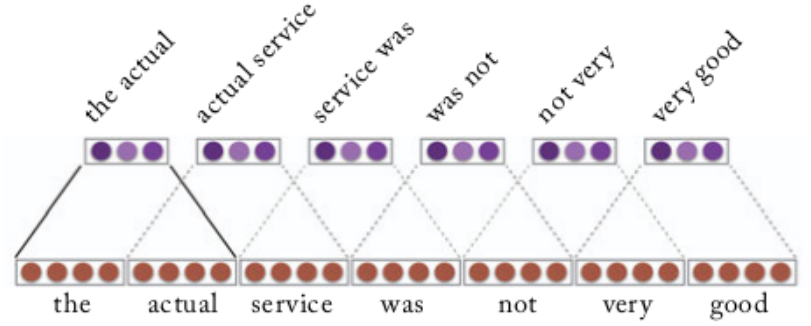
- Sieć splotowa automatycznie wykrywa informatywne dla komputera „kolory” na przetwarzanych obrazach
- Jak stworzyć „obrazek” z tekstu?



Rysunek: Kim, *Convolutional Neural Networks for Sentence Classification*

Sieć splotowa 1D z filtrem o długości 2

- Sieć splotowa automatycznie uczy się potrzebnych *n*-gramów
- W modelu z macierzą zanurzeń będą to tzw. „soft *n*-gram"
- Co więcej sieć może się nauczyć „skip (soft) *n*-grams"
- Dłuższy *n*-gram? Tylko liniowa liczba parametrów! (długość filtra = długość *n*-gramu)
- Parametr stride – czy *n*-gramy się nakładają?

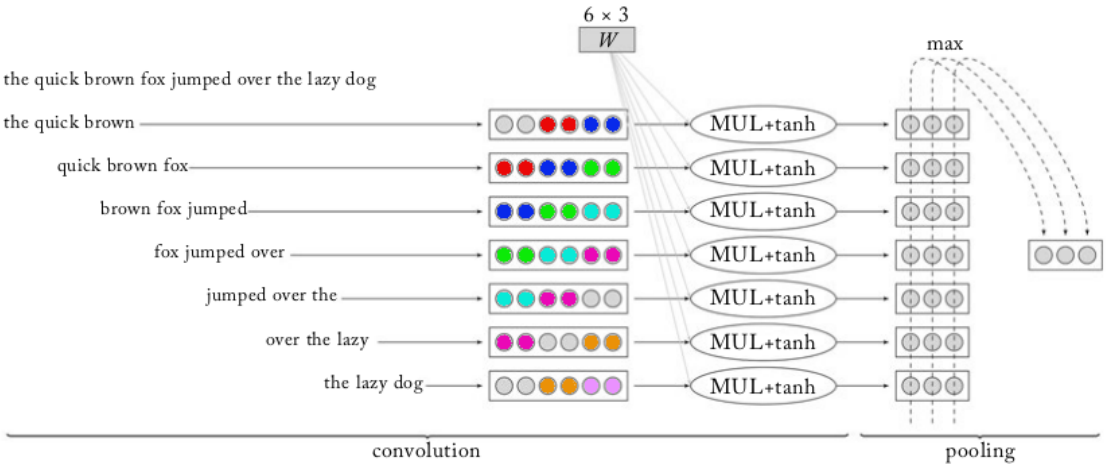


Problem praktyczny: tekst ma różną długość

W przeciwieństwie do obrazów, gdzie zwykle zakładamy że system przetwarza obrazki o danej rozdzielczości, liczbie kolorów i wymiarów, tekst ma różną długość. Jak sobie z tym poradzić?

- ustal stałą długość tekstu i ucinaj resztę, ew. wypełnij STOP tokenami
- stwórz wielowarstwową sieć splotową ze *współdzielonymi* filtrami
- zastosuj sprytną funkcję redukcji (ang. *pooling*)

Redukcja funkcją maksimum wzdłuż czasu (ang. *Max-pooling over time*)

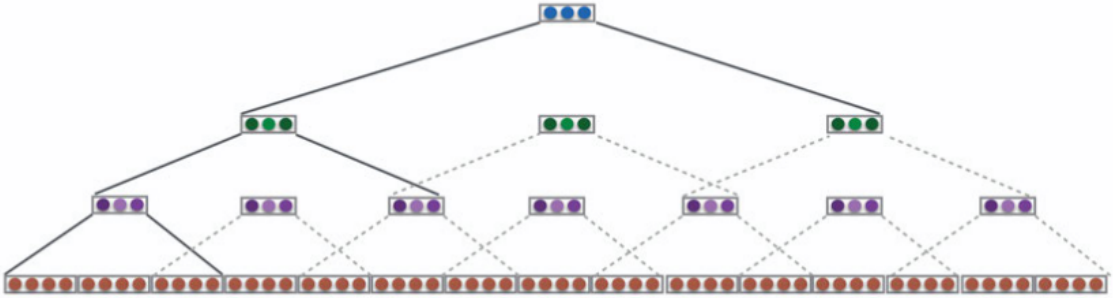


Funkcje redukcji w tekstach

- average pooling over time
- k-Max pooling – wybierz k największych wartości cechy *z zachowaniem kolejności*
- dynamic pooling – podziel sekwencję na kilka fragmentów (nawet 20)

Splot rozszerzany (ang. *dilated convolution*)

Dość popularną techniką jest splot rozszerzany (ang. *dilated convolution*) z filtrem o długości k i stridem $k - 1$.
Bardzo szybko osiąga się długie reprezentacje dokumentów np. $k = 5$ i 8 warstw to > 1000 słów. Również wersje ze współdzieleniem wag.

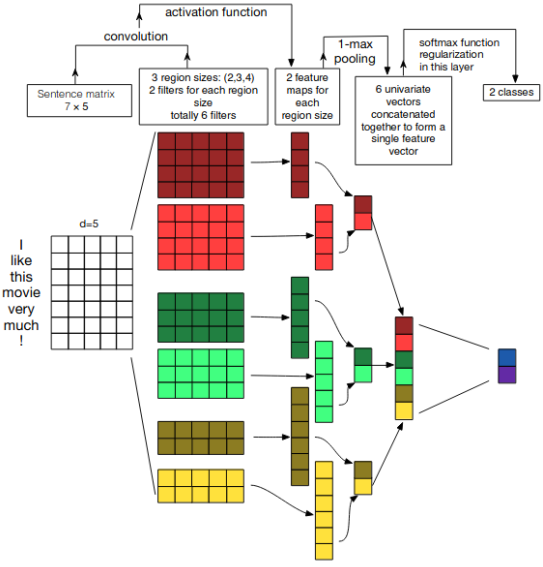


Sieć splotowa do tekstów - przykład prostej architektury

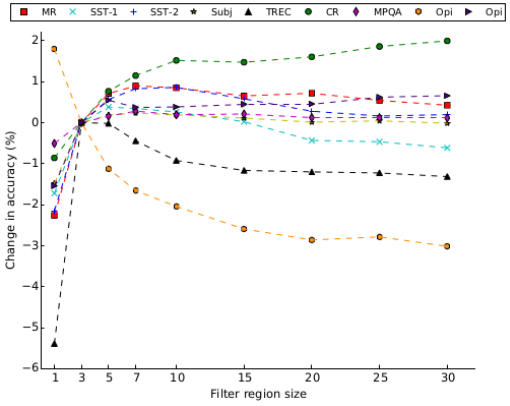
Kim, *Convolutional Neural Networks for Sentence Classification*, 2014

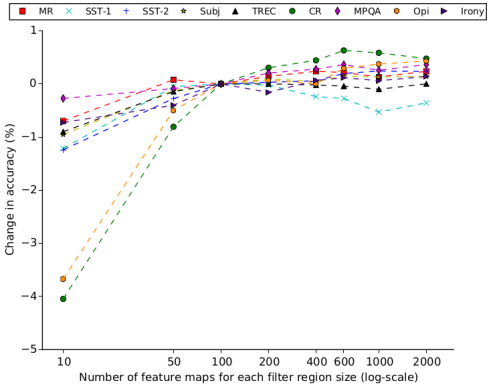
- funkcje aktywacji ReLU, po 100 filtrów o rozmiarach 3, 4 i 5
- jedna warstwa splotu oraz jedna w pełni połączona
- (zwykły) dropout $p = \frac{1}{2}$ tylko w warstwie w pełni połączonej, ze skalowaniem długości gradientu do maksimum 3
- rozmiar paczki: 50, algorytm optymalizacyjny: Adadelta
- rozmiar zanurzeń $d = 300$

Sieć splotowa do tekstów - przykład architektury



Sieć splotowa do tekstów - przykład architektury





(O ile używamy tylko jednej wielkości)

Do zobaczenia!

Fundusze Europejskie
Polska Cyfrowa

Rzeczpospolita Polska

Unia Europejska
Europejski Fundusz Rozwoju Regionalnego