Laboratorium 4 - Podejście strumieniowe do klasyfikacji tekstów

Aleksander Pawłowskia

Politechnika Wrocławska, Wrocław, Polska

1 Omówienie tematu

Na wykładach jak i na poprzednich laboratoriach dowiedzieliśmy się czym jest język naturalny i na czym polega jego przetwarzanie. Mieliśmy okazję najpierw poznać podstawowe zagadnienia przetwarzania wstępnego tekstów. Były nimi między innymi oczyszczanie tekstów z niechcianych znaków czy usuwanie słów nieistotnych dla zadania jakim zajmuje się NLP. Ostatecznie, aby móc zająć się eksperymentami, poznaliśmy czym jest tokenizacja i jaka jest jej rola. Mając tę wiedzę, wykonywaliśmy eksperymenty klasyfikacji tekstów przy pomocy różnych klasyfikatorów, takich jak *MultinomialNB* czy **MLPClassifier** oraz dokonywaliśmy ewaluacji wyników. Nie mniej istotnym poznanym zagadnieniem było znajdowanie podobieństwa między słowami, wykorzystując wyuczony klasyfikator. Eksperyment ten można by łatwo rozszerzyć na znajdowanie podobieństw między całymi tekstami.

Podejściem, którego niestety nie mieliśmy okazji poznać na laboratorium jest strumieniowe podejście do klasyfikacji tekstów. W tym krótkim raporcie omówione zostaną trzy artykuły bazujące na tym podejściu. Pierwszy artykuły przedstawia podejścia przetwarzania strumieni krótkich tekstów. Kolejne dwa omawiają strategię pipeline'u dla ciągłego przetwarzania tekstów.

2 Omówienie artykułów

W artykule autorstwa Peipei Li [1] omówiono kilka metod strumieniowej klasyfikacji tekstów dla dwóch głównych grup: grupa metod wykorzystujących informacje z samego tekstu oraz grupa metod opartych na wykorzystywaniu dodatkowych źródeł informacji w oparciu o informacje znalezione w oryginalnym tekście.

Metodą pozyskującą wiedzę z tekstów było standardowe znajdowanie podobieństw między słowami w celu wykrywania *zdarzeń*, czyli pewnych tematów, których teksty te dotyczyły.

Metody czerpiące wiedzę korzystając z dodatkowych źródeł są ciekawym rozwiązaniem i mogą poprawić jakość grupowania tekstów i przypisywania ich do danych grup. Z eksperymentów wynika, że metody te działają lepiej. W artykule tym skupiono się głównie na badaniu wykrywania dryfu koncepcji dla różnych metod pozyskiwania wiedzy jak i metod detekcji dryfu tematu.

W artykule István'a Pölöskei'a [2] zaproponowano pipeline, którego odpowiednie zaprojektowanie jest niezbędne dla przetwarzania ciągu tekstów. Przetwarzanie tekstów wymaga odpowiedniego przetwarzania wstępnego, aby efektywność tego procesu była wysoka, musi on być częścią dobrze zaprojektowanego *pipeline'u*.

W pracy Rodrigo Agerri'ego [3] zaprojektowana architektura przetwarzania strumieni tekstów została wzbogacona o zrównoleglenie niektórych zadań. Użyto do tego zadania *framework'u* Storm. Następnie porównano działanie tego *framework'u* ze standardowym pipelinem.

Table 3
Performance of the NLP pipeline in different settings: *pipeline* is the basic pipeline used as baseline; *Storm* is the same pipeline executed as a Storm topology; $Storm_2$ represents a Storm pipeline with 2 instances of the WSD module ($Storm_4$ has 4 instances, $Storm_5$ 5, and $Storm_6$ 6).

	Total time	Words/s	Sent/s	Gain (%)
100 documents				
Pipeline	21 m16 s	108.8	4.2	–
Storm	18 m43 s	123.5	4.8	12.0
$Storm_2$	10 m48 s	214.3	8.4	49.3
$Storm_4$	7 m46 s	297.6	11.6	63.5
$Storm_5$	7 m44 s	299.1	11.7	63.7
$Storm_6$	7 m48 s	296.1	11.6	63.3
1000 documents				
Pipeline	3 h15 m16 s	101.2	4.2	–
Storm	2 h50 m21 s	116.0	4.8	12.8
$Storm_2$	1 h40 m37 s	196.5	8.1	48.5
$Storm_4$	1 h14 m25 s	265.6	10.9	61.9
$Storm_5$	1 h10 m45 s	279.3	11.5	63.8
$Storm_6$	1 h11 m37 s	276.0	11.3	63.3

Literatura

1. Peipei Li, Lu He, Haiyan Wang, Xuegang Hu, Yuhong Zhang, Lei Li, and Xindong Wu. Learning from short text streams with topic drifts. *IEEE Transactions on Cybernetics*, 48:2697–2711, 9 2018.
2. Istvan Poloskei. Continuous natural language processing pipeline strategy. pages 221–224. Institute of Electrical and Electronics Engineers Inc., 5 2021.
3. Rodrigo Agerri, Xabier Artola, Zuhaitz Beloki, German Rigau, and Aitor Soroa. Big data for natural language processing: A streaming approach. *Knowledge-Based Systems*, 79:36–42, 2015.