DRAFT VERSION, TO BE PUBLISHED

Explaining Drift in Text Data with Document Embeddings

Adrian Wilke[1], Stefan Heindorf[1], Robert Feldhans[2], Barbara Hammer[2], and Axel Ngonga[1]

[1] Paderborn University
[2] Bielefeld University

Abstract. In this paper we present approaches to explain document-based drift. Document-based drift is a special case of general drift (or concept shift), where it is assumed that drift occurs predominantly in single documents and is not distributed in the entirety of documents. Our contributions comprise two major parts. Firstly, we provide a benchmark to evaluate text-document-based drift detection approaches. Secondly, we provide a pipeline (or rather workflow) to evaluate document-based drift detection approaches. In addition, we propose two unsupervised drift detection approaches for text documents represented via embeddings, and explain text-based drift on token-level. The current state of this paper is a draft version.

1 Introduction

This work is structured as follows: In Section 2 we present a benchmark for document-based drift as well as the conducted data cleaning and selection of documents. Sec. 3 introduces a pipeline to generate visual explanations of detected document-based drift. In our application it is based on document embeddings. Sec. 4 presents results of the developed approaches applied to the benchmark.

2 AMORE: A Document-Based Drift Benchmark

The AMORE (Amazon Movie Reviews) benchmark is a collection of document-based benchmark datasets to compare drift explanation approaches. The single benchmark datasets consist of two sets of unlabeled texts. Each first set forms a base, which represents the respective initial setting. In each second set, a subset of the documents semantically differ from the documents of the first set and contain document-based drift. In the following, we describe the single data processing parts used to compute the final benchmark datasets. We publish the datasets after each processed step to enable researchers to build upon the respective dataset.

The source file[3] contains 7,911,684 movie reviews from Amazon. It was published in [1] For each review, the following data fields are available: *product ID*, *user ID*, *profile name*, *helpfulness* (e.g. 9/9) *score* (ranging from 1 to a good score of 5), *time*, *summary* (a short text) and *text* (a long text). In order to assemble text sets based on semantical differences, we limit the data on the fields *summary*, *text* and *score*.

As the scores and text-based content of the underlying user-generated data showed semantic differences (e.g. good score and negative texts), we used a set of positive and negative words to filter the reviews. For this, positive-rated reviews (score 5 or 4) were only included if the number of positive words (each word counted only one time) as well as the general occurrences of positive words (each word occurrence counted multiple words) were higher as the negative ones. For negative reviews (score 1 or 2), the opposite cases were filtered. Neutral ratings (score 3) were also included, but have not been used afterwards. Overall, 5,483,175 reviews were available after the filtering procedure.

Additionally, we deduplicated the reviews based on the summaries (see Fig. 5) and created two distributions to be compared to each other.

3 Drift Explanation Approaches and Evaluation

In order to explain document-based drift, we conducted a pipeline (see Sec. 3.1). In addition, we developed two drift detectors. Based on embeddings, the Polygons Detector (see Sec. 3.2) reduces embedding dimensions and extracts semantical 2D-outliers. The Hyperboxes approach (see Sec. 3.3 uses each embeddings dimension to detect semantical outliers.

3.1 Pipeline

The pipeline is described in Fig. 1.

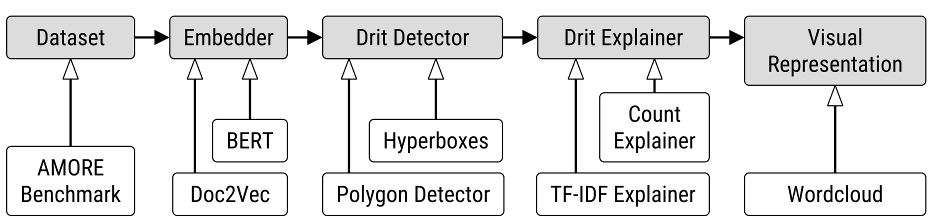


Fig. 1: Explanation and Evaluation pipeline

[3] `https://snap.stanford.edu/data/web-Movies.html`

3.2 Polygons Detector

The 2D-polygons approach is described in Fig. 2.

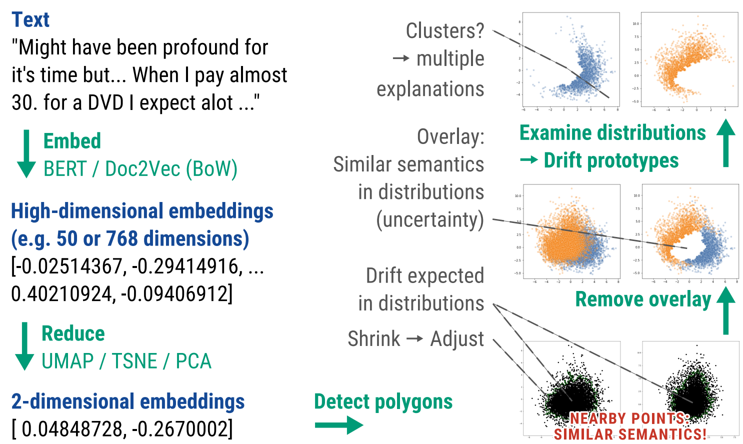


Fig. 2: 2D-polygons approach

3.3 Hyperboxes

The Hyperboxes approach is described in Fig. 3.

Setup: Two sets of embeddings, A and B.

Step 1: Collect values of single dimensions

A_1: `[2,6,3,8,5,...]`
A_2: `[7,4,1,0,9,...]`
$Dim_1(A)$: `[2,7,...]`
$Dim_2(A)$: `[6,4,...]`

Step 2: Create 1-dimensional bounding box for each dimension. Remove outliers (percentiles).

$Dim_1(A)$: `[2,7,7,6,7,9,9,42,9]`
$Dim_2(A)$: `[6,4,1,6,4,6,4,55,6]`
$\rightarrow Box_{Min}(A_1) = 7, Box_{Max}(A_1) = 9$
$\rightarrow Box_{Min}(A_2) = 4, Box_{Max}(A_2) = 6$

Step 3: Get prototypes by checking if values of embeddings B are inside bounding boxes of A.

B_1: `[8,3,...]`
$B_1 \in Box(A_1)$? $7 \leq 8 \leq 9 \rightarrow$ Yes
$B_1 \in Box(A_2)$? $4 \nleq 3 \leq 6 \rightarrow$ No
Score = 1 + 0 + ... < total dimensions
$\rightarrow B_1 \in Prototypes$

Fig. 3: Hyperboxes approach

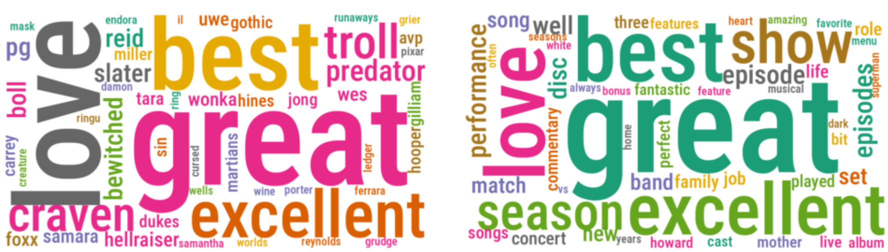
4 Results

We evaluated the Hyperboxes approach on document-level by applying it to BERT and Doc2Vec embeddings and the AMORE-1 benchmark dataset. The Hyperboxes approach using BERT embeddings as source data leads to an accuracy of 0.83 and Doc2Vec embeddings produced an accuracy of 0.58 (see Tab. 1).

Table 1: Evaluation: Detected documents in the Hyperboxes approach

	Doc2Vec	**BERT**
Positives	1.000	1.000
Negatives	9.000	9.000
Detected	4.535	816
TP	691	47
FP	3,844	769
TN	5,156	8,231
Accuracy	0.5847	**0.8278**

Based on the detected documents, we extracted the relevant tokens and created a visualization (see Fig. 4).



(a) BERT embeddings (b) Doc2Vec embeddings

Fig. 4: Explanation of the Hyperboxes approach on token-level

Acknowledgements This work has been supported by the German Federal Ministry of Education and Research (BMBF) within the project EML4U under the grant no 01IS19080B.

References

1. McAuley, J.J., Leskovec, J.: From amateurs to connoisseurs: Modeling the evolution of user expertise through online reviews. p. 897–908. WWW '13 (2013). `https://doi.org/10.1145/2488388.2488466`

5 Appendix

Original dataset

	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	Sum
1	6	191	4.844	19.944	24.221	25.311	25.734	41.016	54.744	49.049	49.521	56.076	59.099	65.343	72.957	81.276	629.332
2	1	262	3.631	17.808	20.320	22.641	24.183	33.117	40.868	37.992	40.205	40.138	39.680	41.430	45.767	47.356	455.399
3	8	442	6.458	30.907	35.395	37.798	43.323	60.489	71.012	66.128	75.239	74.057	73.178	70.279	72.055	74.826	791.594
4	29	797	14.178	73.314	79.152	84.276	90.527	119.160	138.000	135.581	167.632	161.693	149.771	142.000	148.457	150.248	1.654.815
5	64	3.313	49.866	192.002	189.638	198.712	205.916	257.603	308.080	311.252	452.009	412.870	422.403	426.248	465.918	484.650	4.380.544
Σ	108	5.005	78.977	333.975	348.726	368.738	389.683	511.385	612.704	600.002	784.606	744.834	744.131	745.300	805.154	838.356	7.911.684

After semantic cleaning

	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	Sum
1	6	104	3.110	12.698	14.751	15.934	16.611	26.835	35.093	29.878	30.496	33.972	35.420	39.177	43.552	48.432	386.069
2	0	122	1.589	7.820	8.866	10.127	10.742	15.439	19.141	17.152	17.724	17.271	16.641	18.523	20.105	21.036	202.298
3	0	107	1.535	6.152	6.376	6.280	6.917	10.403	12.668	11.947	16.364	16.218	17.691	17.052	18.243	19.266	167.219
4	13	599	10.429	52.130	56.607	58.688	62.256	80.602	93.419	93.912	121.336	117.024	110.548	102.739	109.983	112.609	1.182.894
5	52	2.694	39.621	150.126	148.077	153.886	157.562	194.587	234.448	242.242	370.047	341.264	352.366	353.522	393.641	410.560	3.544.695
Σ	71	3.626	56.284	228.926	234.677	244.915	254.088	327.866	394.769	395.131	555.967	525.749	532.666	531.013	585.524	611.903	5.483.175

After deduplication

	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	Sum
1	2	26	597	2.512	3.015	3.597	3.689	6.643	10.413	9.943	11.125	12.661	14.150	15.822	19.132	21.570	134.897
2	0	30	437	2.162	2.541	3.048	3.364	4.880	7.053	7.050	8.067	8.417	8.846	9.536	11.363	12.041	88.835
3	1	65	880	3.932	4.562	5.064	5.860	8.592	11.420	11.322	13.932	13.944	14.835	14.925	16.796	17.593	143.723
4	4	146	2.166	9.832	11.216	12.257	13.466	19.364	25.958	27.917	37.664	36.838	37.089	36.408	40.392	40.528	351.245
5	14	561	7.266	25.204	26.294	29.576	32.416	46.222	64.445	71.619	108.952	104.455	112.998	113.957	130.571	134.571	1.009.121
Σ	21	828	11.346	43.642	47.628	53.542	58.795	85.701	119.289	127.851	179.740	176.315	187.918	190.648	218.254	226.303	1.727.821

Fig. 5: Number of reviews of cleaning states