The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20)

# TRENDNERT: A Benchmark for Trend and Downtrend Detection in a Scientific Domain

**Alena Moiseeva, Hinrich Schütze**
Center for Information and Language Processing (CIS), LMU Munich, Germany
alena@cis.lmu.de; inquiries@cislmu.org

**Abstract**

Computational analysis and modeling of the evolution of trends is an important area of research in Natural Language Processing (NLP) because of its socio-economic impact. However, no large publicly available benchmark for trend detection currently exists, making a comparative evaluation of methods impossible. We remedy this situation by publishing the benchmark TRENDNERT, consisting of a set of gold trends and downtrends and document labels that is available as an unrestricted download, and a large underlying document collection that can also be obtained for free. We propose Mean Average Precision (MAP) as an evaluation measure for trend detection and apply this measure in an investigation of several baselines.

## 1 Introduction

Science changes and evolves rapidly: novel research areas emerge, while others fade away. Keeping pace with these changes is challenging. Therefore, recognizing and forecasting *research trends* is of great importance for researchers and academic publishers as well as for funding agencies, companies and journalists. The task of trend detection is often solved by domain experts who use special tools to investigate the data and get useful insights from it, e.g., through data visualization. However, manual analysis of large amounts of data can be time-consuming, and domain experts are expensive. Also, the overall increase of research publications (e.g., on Google Scholar, PubMed, DBLP) in the past decade makes an approach based on human domain experts increasingly difficult whereas automated approaches become a more desirable solution for the task of trend detection (Kontostathis et al. 2004).

At its inception, a new research topic often does not attract a lot of attention from the scientific community and is represented by just a few publications. He and Chen (2018) attribute the emergence of a trend to some triggering event, e.g., the publication of an article in a high-impact journal. Then the research topic starts to grow faster and becomes a trend (He and Chen 2018). We adopt this as our definition of *trend* in this paper: a trend is a research topic that has a strongly increasing number of publications in a particular time interval. We define a *downtrend* as the converse: a downtrend is a research topic that has a strongly decreasing number of publications in a particular time interval. Downtrends can be as important to detect as trends, e.g., for a funding agency that is planning its budget.

There has been a lot of work on detecting trends in textual data, both in NLP and machine learning. However, despite the overall importance of trend analysis, no large publicly available *benchmark* for trend detection currently exists, making a *comparative evaluation* of methods impossible. Most of the prior work has used proprietary data or small datasets; or evaluation measures were not directly tied to the objectives of trend detection (Gollapalli and Li 2015).

We remedy this situation by publishing the benchmark TRENDNERT.[1] The benchmark consists of the following:

1. A set of gold trends compiled based on an extensive analysis of the metaliterature;
2. A set of labeled documents, where each document is assigned to a trend, a downtrend or to a third class, the *flat topics*;
3. An evaluation script to compute Mean Average Precision as the measure for the accuracy of trend detection algorithms;

We make the benchmark available as an unrestricted download.[2] The underlying research corpus can be obtained for free from Semantic Scholar (Ammar et al. 2018).[3]

### 1.1 Outline and Contributions

In this work, we address the challenge of benchmark creation for (down)trend detection. This paper is structured as follows: Section 2 discusses related work. Section 3 introduces our model for corpus creation and its components. In Section 4, we present the crowdsourcing procedure for labeling the benchmark. Section 5 introduces our evaluation measure. Section 6 describes the proposed baseline for trend



[1] The name is a concatenation of *trend* and its ananym *dnert* because it supports evaluation of both trends and downtrends.

[2] https://doi.org/10.17605/OSF.IO/DHZCT

[3] https://api.semanticscholar.org/corpus/

detection, our experimental setup, and final results. Finally, we present the details of the distributed benchmark in Section 7 and conclude in Section 8.

Our contributions are as follows:

- TRENDNERT is the *first publicly available benchmark* for (down)trend detection.
- TRENDNERT is based on a collection of more than a million documents. It is among the largest that has been used for trend detection and therefore offers a *realistic setting* for developing trend detection algorithms.
- TRENDNERT addresses the task of detecting both trends and *downtrends*. To the best of our knowledge, the task of downtrend detection has not been addressed before.

## 2 Related Work

In addition to growth, two other characteristics of trends that have been proposed are *novelty* and *utility*, and some prior work has used this more narrow definition of trend (Tu and Seng 2012; Small, Boyack, and Klavans 2014; Rotolo, Hicks, and Martin 2015; He and Chen 2018). Unfortunately, novelty and utility are much harder to quantify automatically than growth. For this reason, we define trends in terms of *growth* only in this paper, i.e., a trend is a research topic that has a strongly increasing number of publications in a particular time interval.

Prior work on analyzing topics and their evolution over time can be classified according to the primary source of information it employs: text, citations and key phrases.

**Text-based analysis:** There has been a great deal of prior work on text-based analysis for trend detection and evolution (Blei and Lafferty 2006; Wang and McCallum 2006; Hall, Jurafsky, and Manning 2008; Gollapalli and Li 2015; Gupta et al. 2018; Schein et al. 2019). However, our focus in this paper is not to develop new algorithms or models, but instead to create a benchmark. We have therefore selected a simple baseline text-based analysis method for trend detection.

**Citation-based analysis** is the second popular direction that is considered to be effective for trend identification (Le, Ho, and Nakamori 2005; Small 2006; He et al. 2009; Shibata et al. 2008; Shibata, Kajikawa, and Takeda 2009). This method assumes that citations in their various forms (bibliographic citations, co-citation networks, citation graphs) indicate meaningful relationships between topics and uses citations to model the topic evolution in the scientific domain. Nie and Sun (2017) utilize this approach along with *Latent Dirichlet Allocation* (LDA) and *k-means* clustering to identify research trends. The authors first use LDA to extract features and determine the optimal number of topics. Then they use k-means to obtain thematic clusters, and finally, they compute *citation functions* to identify the changes of clusters over time. Though the main drawback of citation-based analysis is that there are many trend detection scenarios (e.g., research news articles or research blog posts) in which citations are not available and the approach is therefore not applicable.

**Keyphrase-based analysis:** This approach is based on *keyphrase information* extracted from research papers. A keyphrase is interpreted as a representative for a single research topic. For example, Asooja et al. (2016) utilize the Saffron system (Monaghan et al. 2010) to extract keywords from LREC proceedings and then forecast trends based on regression models. A potential problem with this approach is that there is no clear correspondence between topics and keyphrases. Keyphrases are noisy, ambiguous (e.g., *Java* as an island vs. *Java* as a programming language), and many topics may not correspond to a single keyphrase because there are several equivalent names for a topic.

## 3 Corpus Creation

In this section, we describe the methodology we use to create the TRENDNERT benchmark.

### 3.1 Underlying Data

We use a subset of the Open Research Corpus provided by Semantic Scholar. It contains more than one million papers published mostly between 2000 and 2015 in about 5000 computer science journals and conference proceedings.[4] Each document in the collection consists of title, keyphrases, abstract and other metadata (e.g., venue, year).

### 3.2 Stratification

The distribution of papers over time in the underlying dataset is skewed: years before 2000 have less than 1000 documents per year (see Figure 1).[5] During our experiments, we found that clustering the entire collection or a random sample does not support high-quality identification of (down)trend candidates, because more weight is given to later years than to earlier years (see Section 6.2). Therefore, the first step in the benchmark creation was the generation of a *stratified sample* of the original document collection. To this end, we randomly select $10,000$ documents for each year between 2000 and 2015 for an overall sample size of $160,000$.

### 3.3 Document Representations & Clustering

As we mentioned before, our main focus in this paper is not to develop new algorithms or models, but instead to create a benchmark. We have therefore selected an algorithm based on k-means clustering as a simple baseline method for trend detection. In traditional document clustering (Manning, Raghavan, and Schütze 2008), documents are usually represented as bag-of-words (BOW) feature vectors that however have a major weakness: they ignore the semantics of words (Le and Mikolov 2014). Still, the recent work in representation learning and in particular *doc2vec* – the method proposed by Le and Mikolov (2014) – can provide

[4] It also contains a small number of neuroscience articles.

[5] The number of papers for the year 2016 is small because of the underlying dataset that was provided by Semantic Scholar at the beginning of our work in 2016.

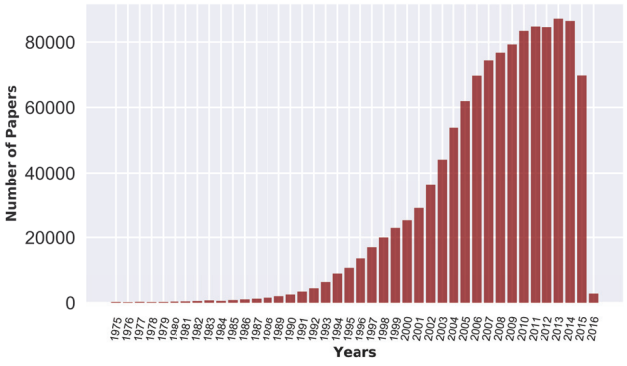



Figure 1: Overall distribution of papers in our dataset. Years 1975 to 2016.

representations to document clustering that overcome this weakness. The *doc2vec*[6] algorithm is inspired by techniques for learning word vectors (Mikolov et al. 2013) and is able to capture semantic regularities in document collections. It is an unsupervised approach for learning continuous distributed vector representations for pieces of text (i.e., paragraphs, documents). This approach maps texts into a vector space such that semantically similar documents are assigned similar vector representations (e.g., an article about *genomics* is closer to an article about *gene expression* than to an article about *fuzzy sets*). Other work has already successfully applied this type of document representations for topic modeling, combining them with both LDA and clustering approaches (Moody 2016; Dieng, Ruiz, and Blei 2019; Xie and Xing 2013; Curiskis et al. 2019).

In our work we run *doc2vec* on the stratified sample of $160,000$ papers and represent each document as a length-normalized vector (i.e., as a document embedding). These vectors are then clustered into $k = 1000$ clusters using the scikit-learn[7] implementation of *k-means* (MacQueen 1967) with default parameters. The combination of document representations with a clustering algorithm is conceptually simple and interpretable. Also, the comprehensive comparative evaluation of topic modeling methods utilizing document embeddings performed by Curiskis et al. (2019) showed that *doc2vec* feature representations with *k-means* clustering outperform several other methods[8] on three evaluation measures (Normalized Mutual Information, Adjusted Mutual Information, and Adjusted Rand Index).

We run 10 trials of stratification and clustering, resulting in 10 different clusterings. We do this to protect against the variability of clustering and because we do not want to rely on a single clustering for proposing (down)trends for the benchmark.

[6]We use the implementation provided by Gensim: https://radimrehurek.com/gensim/models/doc2vec.html

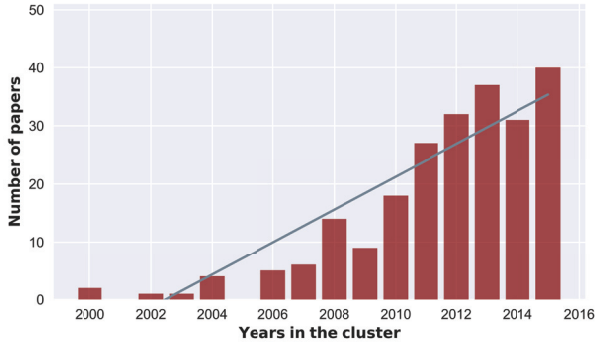
[7]https://scikit-learn.org/stable/

[8]hierarchical clustering, k-medoids, NMF and LDA

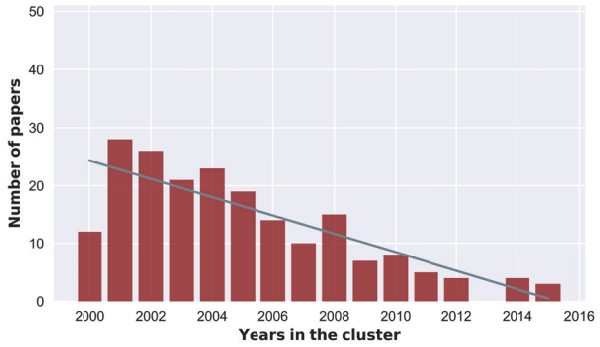
### 3.4 Trend & Downtrend Estimation

Recall our definitions of trend and downtrend. A trend (resp. downtrend) is a research topic that has a strongly increasing (resp. decreasing) number of publications in a particular time interval.

Linear regression is a widely used technique in trend analysis (Hess, Iyer, and Malm 2001) that can identify changes over time in a variable. It correlates the measurements of the variable to the times at which they occurred. Here, we use linear regression to identify trend and downtrend candidates in a given set of clusters. We count the number of documents per year for a cluster and then estimate the slope of the best fitting line. Clusters are then ranked according to the resulting slope and the $n = 150$ clusters with the largest (resp. smallest) slope are selected as *trend* (resp. *downtrend*) *candidates*. Thus, our definition of a single-clustering (down)trend candidate is a cluster with extreme ascending or descending slope. Figure 2 gives two examples of (down)trend candidates.




(a) A trend candidate (positive slope); Topic: Sentiment and Emotion Analysis (using Social Media Channels)




(b) A downtrend candidate (negative slope); Topic: XML

Figure 2: An example of trend and downtrend candidates according to the computed slope.

### 3.5 (Down)trend Candidate Validation

We run ten rounds of clustering (see. Section 3.3), resulting in ten different clusterings. We then estimate the

*(down)trend candidates* according to the procedure described in Section 3.4, so that for each of the ten clusterings, we obtain 150 trend and 150 downtrend candidates.

For our benchmark, among all the (down)trend candidates across the ten clusterings, we want to keep only those that are *consistent* across clusterings. To obtain such consistent (down)trend candidates, we employ the *Jaccard coefficient*. We define an equivalence relation $\sim_R$: two candidates (i.e., clusters) are equivalent if their Jaccard coefficient is $\geq \tau_\sim$, where $\tau_\sim = 0.5$. We compute the equivalence classes of $\sim_R$ over the ten clusterings. Finally, we define an equivalence class as an equivalence class trend candidate (resp. equivalence class downtrend candidate) if it contains trend (resp. downtrends) candidates in at least half of the clusterings. This procedure gave us in total 110 equivalence class trend candidates and 107 equivalence class downtrend candidates that we then annotate for our benchmark.[9]

# 4 Benchmark Annotation

To annotate the equivalence class (down)trend candidates obtained from our model we used the *Figure-Eight*[10] platform. It provides an integrated quality-control mechanism and a training phase before the actual annotation (Kolhatkar, Zinsmeister, and Hirst 2013), which helps disqualify poorly performing annotators.

We design our annotation task as follows. A worker is shown a *descriptor* of a particular candidate along with the list of possible *(down)trend tags* (i.e., "curated" names). The descriptors of candidates are obtained through the following procedure:

- First, we examine the papers assigned to a candidate cluster.
- Then we extract keywords and titles of the papers assigned to the cluster. Semantic Scholar provides keywords and titles as a part of the metadata in the underlying collection (see Section 3.1).
- Finally, we compute the top 15 most frequent keywords and randomly pick 15 titles. These keywords and titles are the descriptor of the candidate shown to crowd-workers.

We create the (down)trend tags as follows. Based on keywords, titles and metadata such as publication venue, cluster candidates are manually labeled by graduate employees (i.e., domain experts in the computer science domain). Examples of (down)trend tags created this way are *"XML"* and *"Sentiment and Emotion Analysis (using Social Media Channels)"*.

In the crowdsourcing interface, the descriptor of a (down)trend candidate is shown on the right. The (down)trend tags are shown on the left. They are mixed and presented in random order. The temporal presentation order of candidates (i.e., descriptors) is also random. The worker is asked to pick from the list of (down)trend tags the tag that matches the descriptor best. Even if there are several tags that are a possible match, the worker must choose one. If there is no matching tag, the worker can select the option *other*. Three different workers evaluate each (down)trend candidate (i.e., descriptor), and the final label is the majority label. If there is no majority label, we present the (down)trend candidate repeatedly to the workers until there is a majority.

## 4.1 Inter-annotator Agreement

To measure the quality of annotation, we compute the inter-annotator agreement – a measure of how well two (or more) annotators agree. We use two measures of inter-annotator agreement: Krippendorff's $\alpha$ and the Figure-Eight Internal Agreement Rate.

**Krippendorff's $\alpha$** is a reliability coefficient developed to measure the agreement between observers (i.e., annotators) (Krippendorff 2011). We choose Krippendorff's $\alpha$ as an inter-annotator agreement measure because it – unlike other specialized coefficients – is a generalization of several reliability criteria and applies to situations like ours where we have more than two annotators and a given annotator only annotates a subset of the data (i.e., we have to deal with missing values).

For our gold standard, Krippendorff's $\alpha$ for the 217 annotated down(trend) candidates (represented as descriptors) is $\alpha = 0.798$. According to Landis and Koch (1977), this value corresponds to a substantial level of agreement.

**Figure-Eight Internal Agreement Rate.** Figure-Eight provides an agreement score $c$ for each annotated unit $u$, which is based on the majority vote of the trusted workers. The score has been proven to perform well compared to classic metrics (Kolhatkar, Zinsmeister, and Hirst 2013). For our gold standard, the average $c$ for the 217 annotated (down)trend candidates is 0.799.

Since both Krippendorff's $\alpha$ and the Figure-Eight Internal Agreement Rate are quite high, we consider the obtained annotations for the benchmark to be of good quality.

## 4.2 Gold Trends

We then compiled a list of computer science trends for the year 2016 based on our analysis of twelve survey publications (Augustin 2016; Frot 2016; Brooks 2016; Reese 2015; Markov 2015; Zaino 2016; Rivera 2016; Harriet 2015; IEEE Computer Society 2016; Nessma 2015; Meyerson and Mariette 2016; Ankerholz 2016). We identified a total of 31 trends; see Table 1. Since there is variation as to the name of a specific trend, we created a standard name for each. Of the 31 trends, 28 (90%) were found as trend candidates during crowdsourcing[11]. We searched for the three missing trends using a variety of techniques: inspection of non-trends/downtrends; random browsing of documents not

[9] Note that the overall number of 217 refers to the number of the (down)trend candidates and not to the number of documents. Each candidate contains hundreds of documents.

[10] formerly called CrowdFlower

[11] The first author verified this based on her judgment of equivalence of one of the 31 gold trend names in the literature with one of the trend tags.

assigned to trend candidates; and keyword searches. Two of the missing trends do not seem to occur in the collection at all: *Human Augmentation* and *Food and Water Technology*. The third missing trend is *Cryptocurrency and Cryptography*; it does occur in the collection, but the occurrence rate is very small (ca. $0.0001$). We do not consider these three trends in the rest of the paper.

In summary, based on our analysis of the meta-literature, we identified $28$ gold trends that also occur in our collection. We will use them as the *gold standard* that *trend detection algorithms should aim to discover*.

## 5 Evaluation Measure for Trend Detection

Whether something is a trend is a graded notion. For this reason, we adopt an evaluation measure that is based on a ranking of trend candidates: *Mean Average Precision (MAP)*. The score represents the average of the precision values of ranks of correctly identified (and non-redundant) trends.

We define trend detection as computing a ranked list of sets of documents (i.e., papers), where each set is a trend candidate. We refer to documents as *trendy*, *downtrendy* or *flat* depending on which category they are assigned to in our benchmark. A trend candidate is called *trendy*, *downtrendy* or *flat*, depending on which of the three classes constitutes the plurality of documents. We count a trend candidate $c$ as a correct recognition of gold trend $t$ if it meets the following conditions:

- $c$ is trendy;
- $t$ is the largest trend in $c$;
- $|t \cap c|/|c| \geq \rho$, where $\rho$ is a coverage parameter;
- $t$ was not earlier recognized in the list.

This criterion give us a true positive (recognition of a gold trend) or false positive (otherwise) for each position of the ranked list, and a precision score for each trend that was found. Precision for a trend that was not found is $0$. We then compute MAP; see Table 2 for an example.

## 6 Trend Detection Baselines

In this section, we investigate the impact of four different configuration choices on the performance of trend detection by way of ablation.

### 6.1 Configuration Choices

**Abstracts vs. Full text.** Our subset of data collection[12] contains both abstracts and full text documents. Our initial expectation was that the full text of a document contains more information than the abstract and should be a better basis for trend detection. This is one of the configuration choices we test in the ablation. We apply our proposed method on full text (document content without abstract) and abstract (only abstract) collections separately and observe the results.

[12]The underlying dataset was provided by Semantic Scholar at the beginning of our work in 2016.

**Stratification.** Due to the uneven distribution of documents over the years, the last years (with increased volume of scientific publications) may get too much influence on results compared to earlier years. To investigate this, we conduct an ablation experiment in which we compare: (i) randomly sampled $160,000$ documents from the entire collection and (ii) stratified sampling. In stratified sampling, we select $10,000$ documents for each of the years $2000$ – $2015$, resulting in a sampled collection of $160,000$.

**Measure of Growth: Length $L_t$ of Interval.** Clusters are ranked by *trendiness*, and the resulting ranked list is evaluated. To measure the *trendiness* or growth of topics over time we fit a line to an interval $\{(i, n_i\}|i_0 \leq i < i_0 + L_t\}$ by linear regression, where $L_t$ is the length of the interval, $i$ is one of the 16 years $(2000, \ldots, 2015)$ and $n_i$ is the number of documents that were assigned to cluster $c$ in that year.

As a simple default baseline, we apply the regression to half the entire interval, i.e., $L_t = 8$. There are nine such intervals in our 16-year period. As the final measure of growth for the cluster, we take the *max* of the nine individual slopes. To determine how much this configuration choice affects our results, we also test a linear regression over the entire time span, i.e., $L_t = 16$. In this case, there is a single interval.

**Clustering Method.** We consider *Gaussian Mixture Models (GMM)* and *k-means*. We use GMM with *spherical* covariance, where each component has the same variance. For both, we proceed as described in Section 3.3.

### 6.2 Experimental Setup and Results

We conducted experiments on the Semantic Scholar corpus and evaluated a ranked list of trend candidates against the benchmark. We used MAP as the primary evaluation measure. As a secondary evaluation measure, we compute recall at $50$ (R@50), which estimates the percentage of gold trends found in the list of $50$ highest-ranked trend candidates. We found that the configuration $(0)$ in Table 3 works best. We then conducted ablation experiments to determine the importance of the configuration choices.

**Comparing (0) and (1),**[13] we see that abstracts $(A)$ are a better representation for our trend detection baseline than full text $(F)$. This can be explained by the fact that an abstract is a summary of a scientific paper that covers only the central points, and thus it is semantically very concise and rich. In contrast, the full text contains many parts that are secondary (e.g., future work, related work) and thus may be an inferior representation of the overall meaning of the document.

**Comparing (0) and (2),** we observe that stratification $(y)$ clearly improves results compared to the setting with unstratified data $(n)$, i.e., randomly sampled data. We would

[13]Numbers (0), (1), (2), (3), (4) denote the configuration choices in the ablation Table 3.

| Computer Science Trends | |
|---|---|
| Autonomous Agents and Systems | Natural Language Processing |
| Autonomous Vehicles | Open Source |
| Big Data Analytics | Privacy |
| Bioinformatics and (Bio) Neuroscience | Quantum Computing |
| Biometrics and Personal Identification | Recommender Systems and Social Networks |
| Cloud Computing and Software as a Service | Reinforcement Learning |
| Cryptocurrency and Cryptography* | Renewable Energy |
| Cyber Security | Robotics |
| E-business | Semantic Web |
| Food and Water Technology* | Sentiment and Emotion Analysis |
| Game-based Learning | Smart Cities |
| Games and (Virtual) Augmented Reality | Supply Chains and RFIDs |
| Human Augmentation* | Technology for Climate Change |
| Machine/Deep Learning | Transportation and Energy |
| Medical Advances and DNA Computing | Wearables |
| Mobile Computing | |

Table 1: 31 topics identified as computer science trends for 2016 in the media. 28 of these trends were identified as trends in crowdsourcing, and thus are covered in our benchmark. The three topics marked with asterisk ($*$) were not found.

| **Gold Trends** | **Gold Downtrends** |
|---|---|
| Cloud Computing | Compilers |
| Bioinformatics | Petri Nets |
| Sentiment Analysis | Fuzzy Sets |
| Privacy | Routing Protocols |

| | *Trend Candidate* | $\lvert t \cap c \rvert / \lvert c \rvert$ | *tp/fp* | $P$ |
|---|---|---|---|---|
| $c_1$ | Cloud Computing | 0.60 | tp | 1.00 |
| $c_2$ | Privacy | 0.20 | fp | 0.50 |
| $c_3$ | Cloud Computing | 0.70 | fp | 0.33 |
| $c_4$ | Bioinformatics | 0.55 | tp | 0.50 |
| $c_5$ | Sentiment Analysis | 0.95 | tp | 0.60 |
| $c_6$ | Sentiment Analysis | 0.59 | fp | 0.50 |
| $c_7$ | Bioinformatics | 0.41 | fp | 0.43 |
| $c_8$ | Compilers | 0.60 | fp | 0.38 |
| $c_9$ | Petri Nets | 0.80 | fp | 0.33 |
| $c_{10}$ | Privacy | 0.33 | fp | 0.30 |

Table 2: Example of proposed MAP evaluation (with $\rho = 0.5$): MAP is the average of the precision values 1.0 (Cloud Computing), 0.5 (Bioinformatics), 0.6 (Sentiment Analysis), and 0.0 (Privacy), i.e., MAP = $2.1/4 = 0.525$. True positive: *tp*, False positive: *fp*, Precision: *P*.

expect the effect of stratification to be even stronger for collections in which the distribution of documents over time is more skewed than in ours.

**Comparing (0) and (3),** the length of the interval is an important configuration choice: as we expected, the 16 year interval is too long and the 8 year interval seems to be a better choice, especially if one aims to find short-term trends.

**Comparing (0) and (4),** we see that topics obtained from k-means and from GMM have similar performance. However, GMM, which takes variance into account and estimates a soft assignment, performs slightly better.

## 7 Benchmark Description

We now describe the contents of the TRENDNERT benchmark. It contains the following components.

1. Two files containing information about documents, one for documents assigned to (down)trends and one for documents assigned to flat topics, i.e., topics that are neither trends nor downtrends;
2. A script for generating document hash codes;
3. A file mapping hash codes to our internal IDs;
4. A script for computing MAP for a list of (down)trend candidates (default setting is $\rho = .25$);

The two files containing information about documents give the following information for each document.

- Paper ID: the internal ID we use for documents in the Semantic Scholar collection;
- Cluster ID: unique ID of the equivalence class in which the document occurs;
- Label: (down)trend tag assigned by crowd-workers (e.g., *Cyber Security*, *Fuzzy Sets and Systems*);
- Type of (down)trend candidate: trend ($T$), downtrend ($D$) or flat topic ($F$);
- Hash ID[14] of each paper from the Semantic Scholar collection;

## 8 Conclusion

With this work we release TRENDNERT – the *first publicly available* benchmark for (down)trend detection that offers a realistic setting for evaluating trend detection algorithms.

[14]MD5-hash of: First Author + Title + Year

| | (0) | (1) | (2) | (3) | (4) |
|---|---|---|---|---|---|
| document part | A | **F** | A | A | A |
| stratification | y | y | **n** | y | y |
| $L_t$ | 8 | 8 | 8 | **16** | 8 |
| clustering | $GMM^{sph}$ | $GMM^{sph}$ | $GMM^{sph}$ | $GMM^{sph}$ | $\mathbf{k}\mu$ |
| **MAP** | .36 (.03) | .07 (.19) | .30 (.03) | .32 (.04) | .34 (.21) |
| **R@50 avg** | .50 (.012) | .25 (.018) | .50 (.016) | .46 (.018) | .53(.014) |
| **R@50 max** | .61 | .37 | .53 | .61 | .61 |

Table 3: Ablation results. Standard deviations are in parentheses. *GMM* clustering performed with spherical (sph.) covariance. *K-means* clustering is denoted as k$\mu$ in the ablation table. $\rho = .25$

TRENDNERT also supports *downtrend detection* – an important problem that was not addressed before. We evaluate a number of baselines for trend detection on TRENDNERT. We find that, for these baselines, *stratification* improves trend detection if the distribution of documents is skewed and that *abstract-based* representations perform better than fulltext representations for the models included in our experiments.

**Acknowledgments.** We gratefully acknowledge funding from the European Research Council (grant 740516) and Deutsche Forschungsgemeinschaft (grant DFG SCHU 2246/8-2, SPP 1335). We would like to thank Stefan Rüd for his help with the crowdsourcing setup, and Annemarie Friedrich and the anonymous reviewers for their valuable comments. Finally, we owe gratitude to Semantic Scholar for making the document collection available that we use in this paper.

## References

Ammar, W.; Groeneveld, D.; Bhagavatula, C.; Beltagy, I.; Crawford, M.; Downey, D.; Dunkelberger, J.; Elgohary, A.; Feldman, S.; Ha, V.; et al. 2018. Construction of the literature graph in semantic scholar. *arXiv preprint arXiv: 1805. 02262*.

Ankerholz, A. 2016. 2016 future of open source survey says open source is the modern architecture. https://www.linux.com/news/2016-future-open-source-survey-says-open-source-modern-architecture.

Asooja, K.; Bordea, G.; Vulcu, G.; and Buitelaar, P. 2016. Forecasting emerging trends from scientific literature. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, 417–420.

Augustin, J. 2016. Emergin science and technology trends: A synthesis of leading forecast. http://www.defenseinnovationmarketplace.mil/resources/2016_SciTechReport_16June2016.pdf.

Blei, D. M., and Lafferty, J. D. 2006. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, 113–120. ACM.

Brooks, C. 2016. 7 top tech trends impacting innovators in 2016. http://innovationexcellence.com/blog/2015/12/26/7-top-tech-trends-impacting-innovators-in-2016/.

Curiskis, S. A.; Drake, B.; Osborn, T. R.; and Kennedy, P. J. 2019. An evaluation of document clustering and topic modelling in two online social networks: Twitter and reddit. *Information Processing & Management*.

Dieng, A. B.; Ruiz, F. J.; and Blei, D. M. 2019. Topic modeling in embedding spaces. *arXiv preprint arXiv:1907.04907*.

Frot, M. 2016. 5 trends in computer science research. https://www.topuniversities.com/courses/computer-science-info-rmation-systems/5-trends-computer-science-research.

Gollapalli, S. D., and Li, X. 2015. Emnlp versus acl: Analyzing nlp research over time. In *EMNLP*, 2002–2006.

Gupta, P.; Rajaram, S.; Schütze, H.; and Andrassy, B. 2018. Deep temporal-recurrent-replicated-softmax for topical trends over time. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 1079–1089. New Orleans, Louisiana: Association for Computational Linguistics.

Hall, D.; Jurafsky, D.; and Manning, C. D. 2008. Studying the history of ideas using topic models. *In Proceedings of the EMNLP*.

Harriet, T. 2015. Privacy will hit tipping point in 2016. http://www.cnbc.com/2015/11/09/privacy-will-hit-tipping-point-in-2016.html.

He, J., and Chen, C. 2018. Predictive effects of novelty measured by temporal embeddings on the growth of scientific literature. *Frontiers in Research Metrics and Analytics* 3:9.

He, Q.; Chen, B.; Pei, J.; Qiu, B.; Mitra, P.; and Giles, L. 2009. Detecting topic evolution in scientific literature: How can citations help? In *Proceedings of the 18th ACM conference on Information and knowledge management*, 957–966. ACM.

Hess, A.; Iyer, H.; and Malm, W. 2001. Linear trend analysis: a comparison of methods. *Atmospheric Environment* 35(30):5211 – 5222. Visibility, Aerosol and Atmospheric Optics.

IEEE Computer Society. 2016. Top 9 computing technology trends for 2016. https://www.scientificcomputing.com/news/2016/01/top-9-computing-technology-trends-2016.

Kolhatkar, V.; Zinsmeister, H.; and Hirst, G. 2013. Annotating anaphoric shell nouns with their antecedents. In

*Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, 112–121.

Kontostathis, A.; Galitsky, L. M.; Pottenger, W. M.; Roy, S.; and Phelps, D. J. 2004. A survey of emerging trend detection in textual data mining. In *Survey of text mining*. Springer. 185–224.

Krippendorff, K. 2011. Computing krippendorff's alpha-reliability. *Annenberg School for Communication (ASC) Departmental Papers* 43.

Landis, J. R., and Koch, G. G. 1977. The measurement of observer agreement for categorical data. *biometrics* 159–174.

Le, Q. V., and Mikolov, T. 2014. Distributed representations of sentences and documents. In *ICML*, volume 14, 1188–1196.

Le, M.-H.; Ho, T.-B.; and Nakamori, Y. 2005. Detecting emerging trends from scientific corpora. *International Journal of Knowledge and Systems Sciences* 2(2):53–59.

MacQueen, J. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, 281–297. Oakland, CA, USA.

Manning, C. D.; Raghavan, P.; and Schütze, H. 2008. *Introduction to Information Retrieval*. Cambridge University Press. Web publication at informationretrieval.org.

Markov, I. 2015. 13 of 2015's hottest topics in computer science research. https://www.forbes.com/sites/quora/2015/04/22/13-of-2015s-hottest-topics-in-computer-science-research/#fb0d46b1e88d.

Meyerson, B., and Mariette, D. 2016. These are the top 10 emerging technologies of 2016. http://www3.weforum.org/docs/GAC16_Top10_Emerging_Technologies_2016_report.pdf.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Monaghan, F.; Bordea, G.; Samp, K.; and Buitelaar, P. 2010. Exploring your research: Sprinkling some saffron on semantic web dog food. In *Semantic Web Challenge at the International Semantic Web Conference*, volume 117, 420–435. Citeseer.

Moody, C. E. 2016. Mixing dirichlet topic models and word embeddings to make lda2vec. *arXiv preprint arXiv:1605.02019*.

Nessma, J. 2015. Top 10 hottest research topics in computer science. http://www.pouted.com/top-10-hottest-research-topics-in-computer-science/.

Nie, B., and Sun, S. 2017. Using text mining techniques to identify research trends: A case study of design research. *Applied Sciences* 7(4):401.

Reese, H. 2015. 7 trends for artificial intelligence in 2016: 'like 2015 on steroids'. http://www.techrepublic.com/article/7-trends-for-artificial-intelligence-in-2016-like-2015-on-steroids/.

Rivera, M. 2016. Is digital game-based learning the future of learning? https://elearningindustry.com/digital-game-based-learning-future.

Rotolo, D.; Hicks, D.; and Martin, B. R. 2015. What is an emerging technology? *Research Policy* 44(10):1827–1843.

Schein, A.; Linderman, S. W.; Zhou, M.; Blei, D. M.; and Wallach, H. M. 2019. Poisson-randomized gamma dynamical systems. *CoRR* abs/1910.12991.

Shibata, N.; Kajikawa, Y.; Takeda, Y.; and Matsushima, K. 2008. Detecting emerging research fronts based on topological measures in citation networks of scientific publications. *Technovation* 28(11):758–775.

Shibata, N.; Kajikawa, Y.; and Takeda, Y. 2009. Comparative study on methods of detecting research fronts using different types of citation. *Journal of the American Society for information Science and Technology* 60(3):571–580.

Small, H.; Boyack, K. W.; and Klavans, R. 2014. Identifying emerging topics in science and technology. *Research Policy* 43(8):1450–1467.

Small, H. 2006. Tracking and predicting growth areas in science. *Scientometrics* 68(3):595–610.

Tu, Y.-N., and Seng, J.-L. 2012. Indices of novelty for emerging topic detection. *Information processing & management* 48(2):303–325.

Wang, X., and McCallum, A. 2006. Topics over time: a non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 424–433. ACM.

Xie, P., and Xing, E. P. 2013. Integrating document clustering and topic modeling. *arXiv preprint arXiv:1309.6874*.

Zaino, J. 2016. 2016 trends for semantic web and semantic technologies. http://www.dataversity.net/2017-predictions-semantic-web-semantic-technologies/.