



Article

# Vec2Dynamics: A Temporal Word Embedding Approach to Exploring the Dynamics of Scientific Keywords—Machine Learning as a Case Study

Amna Dridi <sup>1,\*</sup>, Mohamed Medhat Gaber <sup>1,2</sup>, Raja Muhammad Atif Azad <sup>1</sup> and Jagdev Bhogal <sup>1</sup>

<sup>1</sup> School of Computing and Digital Technology, Birmingham City University, Birmingham B4 7XG, UK; mohamed.gaber@bcu.ac.uk (M.M.G.); atif.azad@bcu.ac.uk (R.M.A.A.); jagdev.bhogal@bcu.ac.uk (J.B.)

<sup>2</sup> Faculty of Computer Science and Engineering, Galala University, Suez 435611, Egypt

\* Correspondence: amna.dridi@bcu.ac.uk

**Abstract:** The study of the dynamics or the progress of science has been widely explored with descriptive and statistical analyses. Also this study has attracted several computational approaches that are labelled together as the Computational History of Science, especially with the rise of data science and the development of increasingly powerful computers. Among these approaches, some works have studied dynamism in scientific literature by employing text analysis techniques that rely on topic models to study the dynamics of research topics. Unlike topic models that do not delve deeper into the content of scientific publications, for the first time, this paper uses temporal word embeddings to automatically track the dynamics of scientific keywords over time. To this end, we propose *Vec2Dynamics*, a neural-based computational history approach that reports *stability* of *k*-nearest neighbors of scientific keywords over time; the stability indicates whether the keywords are taking new neighborhood due to evolution of scientific literature. To evaluate how *Vec2Dynamics* models such relationships in the domain of Machine Learning (ML), we constructed scientific corpora from the papers published in the Neural Information Processing Systems (NIPS; actually abbreviated *NeurIPS*) conference between 1987 and 2016. The descriptive analysis that we performed in this paper verify the efficacy of our proposed approach. In fact, we found a generally strong consistency between the obtained results and the Machine Learning timeline.

**Keywords:** scientific literature; computational linguistics; temporal word embedding; *k*-NN stability; machine learning



**Citation:** Dridi, A.; Gaber, M.M.; Azad, R.M.A.; Bhogal, J. Vec2Dynamics: A Temporal Word Embedding Approach to Exploring the Dynamics of Scientific Keywords—Machine Learning as a Case Study. *Big Data Cogn. Comput.* **2022**, *6*, 21. <https://doi.org/10.3390/bdcc6010021>

Academic Editors: Renyu Yang, Zhenyu Wen, Xu Wang, Prosanta Gope and Bin Shi

Received: 31 December 2021

Accepted: 15 February 2022

Published: 21 February 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Over the past few years, the computational history of science—as a part of big scholarly data analysis [1]—has grown into a scientific research area that is increasingly being applied in different domains such as business, biomedical, and computing. The surge in interest is due to (i) the explosion of publicly available data on scholarly networks and digital libraries, and (ii) the importance of the study of scientific literature [2], which is continuously evolving. In fact, the recent literature is rich in dealing with the enigmatic question of the dynamics of science [3–12].

Most of these proposed approaches focus on citation counts to detect the popularity and the evolution of research topics [13]. Despite their importance to show the interestingness and the scientific impact of published work, citations fail to analyse the paper content, which could potentially provide more accurate computational history of science. Therefore, some emerging works [4,6,9] relied on topic models to analyse the dynamics of research topics. Topic models generally extract semantics at a document level, which ignores the detection of pairwise associations between keywords, where keywords refer to the words of interest in scientific publications.

Going beyond these approaches, our aim in this paper is to train word embeddings across time in order to obtain a fine-grained computational history of science. For instance, the basic idea of word embedding is words that occur in similar context tend to be closer to each other in vector space. This means that word neighbours should be semantically similar. The choice of word embedding models is justified by the ability they have shown to capture and represent semantic relationships from massive amounts of data [14]. We intend in this study to uncover the dynamics of science by studying the change in pairwise associations between keywords over time. Our examination is, therefore, intended to be descriptive rather than normative following the approaches that studied the computational history of science [4,6,7]. As a matter of fact, no standard has been provided to conduct a comparative study because expert opinions should be taken into consideration in such descriptive analyses [8]. The subject area studied in this work is Computer Science and more precisely *Machine Learning*. The choice of this subject is based on two main reasons: (i) the authors' background in this subject area, and (ii) the fact that Machine Learning has seen remarkable successes in the recent decades and has been applied in different domains including *bioinformatics*, *cyber security*, and *neuroscience*.

To obtain a fine-grained computational history, we introduce *Vec2Dynamics*—a neural-based approach to computational history that uses temporal word embeddings in order to analyse and explore over time the dynamics of keywords in relation with their nearest neighbours. To detect the dynamics of keywords, we learn word vectors across time. Then, based on the similarity measure between keywords embedding vectors, we define the  $k$ -nearest neighbors ( $k$ -NN) of each keyword over successive time windows. The change in stability of  $k$ -NN over time refers to the dynamics of keywords and accordingly the research area.

We evaluate our proposed approach *Vec2Dynamics* with 6562 publications from the Neural Information Processing Systems (NIPS) conference proceedings between the years 1987 and 2016 that we divided into ten time windows of 3-years each. As explained below, we adopt both numerical and visual methods to perform our descriptive analysis and evaluate the effectiveness of *Vec2Dynamics* in tracking the dynamics of scientific keywords.

- *Numerically*, we compute the average stability of  $k$ -nearest neighbours of keywords over all the time windows and detect the stable/unstable periods in the history of Machine Learning.
- *Visually*, we illustrate the advantages of temporal word embeddings that show the evolvement of scientific keywords by drawing *Venn diagrams* showing  $k$ -NN keywords over each two subsequent time periods. All visualisations show that our approach is able to illustrate the dynamics of Machine Learning literature.

The major contributions of this paper are summarized as follows:

1. To detect the dynamics of keywords, word vectors are learned across time. Then, based on the similarity measure between the embedding vectors of keywords, the  $k$ -nearest neighbors ( $k$ -NN) of each keyword are defined over successive timespans.
2. The change in stability of  $k$ -NN over time refers to the dynamics of keywords and accordingly the dynamics of the research area.
3. *Vec2Dynamics* is evaluated in the area of Machine Learning with the NIPS publications between the years 1987 and 2016.
4. Both numerical and visual methods are adopted to perform a descriptive analysis and evaluate the effectiveness of *Vec2Dynamics* in tracking the dynamics of scientific keywords.
5. *Machine Learning timeline* has been proposed as standard for descriptive analysis. A generally good consistency between the obtained results and the machine learning timeline has been found.
6. *Venn diagrams* have been used in this paper for qualitative analyses to highlight the semantic shifts of scientific keywords by showing the evolvement of the semantic neighborhood of scientific keywords over time.

The rest of the paper is organised as follows. Section 2 presents a summary of the existing computational history approaches that focus on the Computer Science research area. Section 3 details the *Vec2Dynamics* framework and its different stages. Section 4 describes the NIPS dataset we have used, and presents and discusses the obtained results. Finally, in Section 5 we conclude the paper and draw future directions.

## 2. Related Work

### 2.1. Computational History of Science: Approaches and Disciplines

Due to the rapid growth of scientific research, the Computational History of Science has emerged [1] recently to understand the scientific evolution. For this reason, several approaches have been presented in the literature relying on different features such as citations, paper content (mainly keywords), or both of them. Accordingly, these approaches can be categorised into three categories based on the features they have been using [15]: bibliometrics-based approaches [3,8,10–12]; *content-based approaches* [4,6,9,16]; and *hybrid approaches* [5,7] that combine both citations and paper content.

These approaches dealing with the computational history of science have been applied to a broad range of disciplines ranging from business, relations and economy [3,10–12,17] to medicine [18], biology [19], geoscience [20], and computer science [4,5,7–9,21–25].

### 2.2. Computational History of Computer Science

This paper concerns a discipline within Computer Science; therefore, it is pertinent to discuss previous works concerning computational history of Computer Science (CS) [15].

#### 2.2.1. Bibliometrics and Hybrid Approaches

Hoonlor et al. [7] did some of the early work that investigated the evolution of CS research. They have analysed data on grant proposals, ACM (<https://dl.acm.org/>, accessed on 12 October 2017) and IEEE (<https://ieeexplore.ieee.org/>, accessed on 12 October 2017) publications using both sequential pattern mining and emergent word clustering. Similarly, Hou et al. [8] studied the progress of research topics using Document-Citation Analysis (DCA) of Information Science (IS). To do so, they used dual-map overlays of the IS literature to track the evolution of IS research based on different academic factors such as H-index, citation analysis and scientific collaboration. In the same context, Effendy and Yap [5] obtained the computational history using the Microsoft Academic Graph (MAG) (<https://www.microsoft.com/en-us/research/project/microsoft-academic-graph/>, accessed on 12 October 2017) dataset. In addition to the citation-basic method, they used a content-based method by leveraging the hierarchical *FoS* (*Field of Study*) given by MAG for each paper to determine the level of interest in any particular research area or topic, and accordingly general publication trends, growth of research areas and the relationship among research areas in CS.

Both the aforementioned approaches to computational history in CS can be classified as hybrid approaches. They use both citation analysis and content analysis to detect research dynamics. The content analysis only studies bursty keywords—that are the new emerging keywords, which become suddenly frequent in a short time—in [7] and fields of studies in [5] without providing any fine-grained analysis of the paper content. Alternatively, they concentrate on citation analysis to uncover citation trends and accordingly the evolution of research areas. While citation counts are accounted to be crucial to assessing the importance of scientific work, they take time to stabilise enough to bring out research trends, which makes this technique more effective for retrospective analysis than descriptive or predictive analysis.

#### 2.2.2. Content-Based Approaches

For the reasons detailed in Section 2.2.1 and because citation-based approaches neglect to delve into the paper content, the work presented in this paper can be classified in the

category of content-based approaches because it provides a fine-grained content analysis of research papers.

Some work on coarse-grained content analysis exists. Glass et al. [26] provided a descriptive analysis of the state of software engineering (SE) research by examining 369 research papers in six leading SE journals over the period 1995–1999. Similarly, Schlagenhauer and Amberg [27] provided a structured, summarised as well as organised overview of research activities conducted on gamification literature by analysing 43 top ranking IS journals and conferences. The authors followed a descriptive analysis using a classification framework with the help of the Grounded Theory approach [28]. Both of these approaches are too coarse-grained in that they follow only a descriptive analysis where no fine-grained content analysis of research papers is provided.

To overcome this problem, some computational approaches have been proposed, namely *topic modeling approaches*. For instance, Hall et al. [6] applied unsupervised topic modeling to the ACL (Association for Computational Linguistics) Anthology (CoLING, EMNLP and ACL) to analyse historical trends in the field of *computational linguistics* from 1987 to 2006. They used Latent Dirichlet Allocation (LDA) to extract topic clusters and examine the strength of each topic over time. In the same context, Anderson et al. [4] have proposed a people-based methodology for computational history that traces authors' flow across topics to detect how some sub-areas flow into the next to form new research areas. This methodology is founded on a central phase of topic modelling that detects papers' topics and identifies the topics the people author in. Mortenson et al. [9] offered a software called *Computational Literature Review (CLR)* that evaluates large volumes of the existing literature with respect to impact, structure and content using LDA topic modeling. They applied their approach to the area of IS and more precisely to the Technology Acceptance Model (TAM) theory. Similarly, Salatino et al. [29] have recently proposed *Augur*, which analyses the temporal relationships between research topics and clusters topics that present dynamics of already established topics in the area of CS.

### 2.3. Limitations

Although the aforementioned approaches [4,6,9] have the intention to conduct a content analysis of research papers by using topic modelling, they still experience some issues that show its limitations. Actually, both the flow of authors across topics and the dynamics of recognised topics need time to take place. Furthermore, topic modelling fails to detect pairwise associations between words, while the investigation of these associations at a local space could lead to the detection of evolving research patterns—topics or keywords—timely.

To overcome the two problems just described, this paper proposes a *temporal word embedding* technique that digs deep into the paper content to instantly detect how the semantic neighbourhood of keywords evolves over time. To the best of our knowledge, only three recent works have explored word embedding techniques to model scientific dynamics. The first work was by He et al. [30] and aimed to track the semantic changes of scientific terms over time in the biomedical area. The second and the third are our recent works that introduced *DeepHist* [31] and *Leap2Trend* [15] respectively, which are word embedding based approaches for computational history applied to Machine Learning publications during the time period 1987–2015. The first approach *DeepHist* detects the converging keywords that may result in trending keywords by computing the acceleration of similarities between keywords over successive timespans. However, the second approach *Leap2Trend* tracks the dynamics of similarities between pairs of keywords, their rankings and respective uprankings (ascents) over time to detect the keywords that start to frequently co-appear in the same context, which may lead to emerging scientific trends.

Therefore, this work employs temporal word embeddings, namely word2vec, to track the dynamics of scientific keywords. Unlike *DeepHist* and *Leap2Trend* that aim to detect only new trends, our current approach aims to track the dynamics of scientific literature in general by detecting the stability of the nearest neighbors keywords over time.

### 3. Vec2Dynamics

In order to understand and uncover the dynamics of scientific literature, we propose *Vec2Dynamics*—a fine-grained text mining approach that follows a temporal paradigm to delve into the content of research papers.

First, our *Vec2Dynamics* digs into the textual content by applying word embedding techniques, namely word2vec [32]. Word2vec has been chosen because it is the long standing word embedding techniques in the area. However, any other word embedding technique could be applied such as *GloVe* [33] and *FastText* [34]. Then, *Vec2Dynamics* grasps dynamic change in interest and popularity of research topics by iteratively applying *k*-NN stability to the keywords/topics of interest over time and accordingly capturing the *recurrent*, *non-recurrent*, *persistent* and *emerging* keywords. Formal definitions of these types of keyword dynamics are given later in this section. We first describe the general architecture of *Vec2Dynamics* and then detail the functionalities of its different stages.

#### 3.1. Vec2Dynamics Architecture

The architecture of *Vec2Dynamics* is depicted in Figure 1. The whole model can be divided into four different stages.

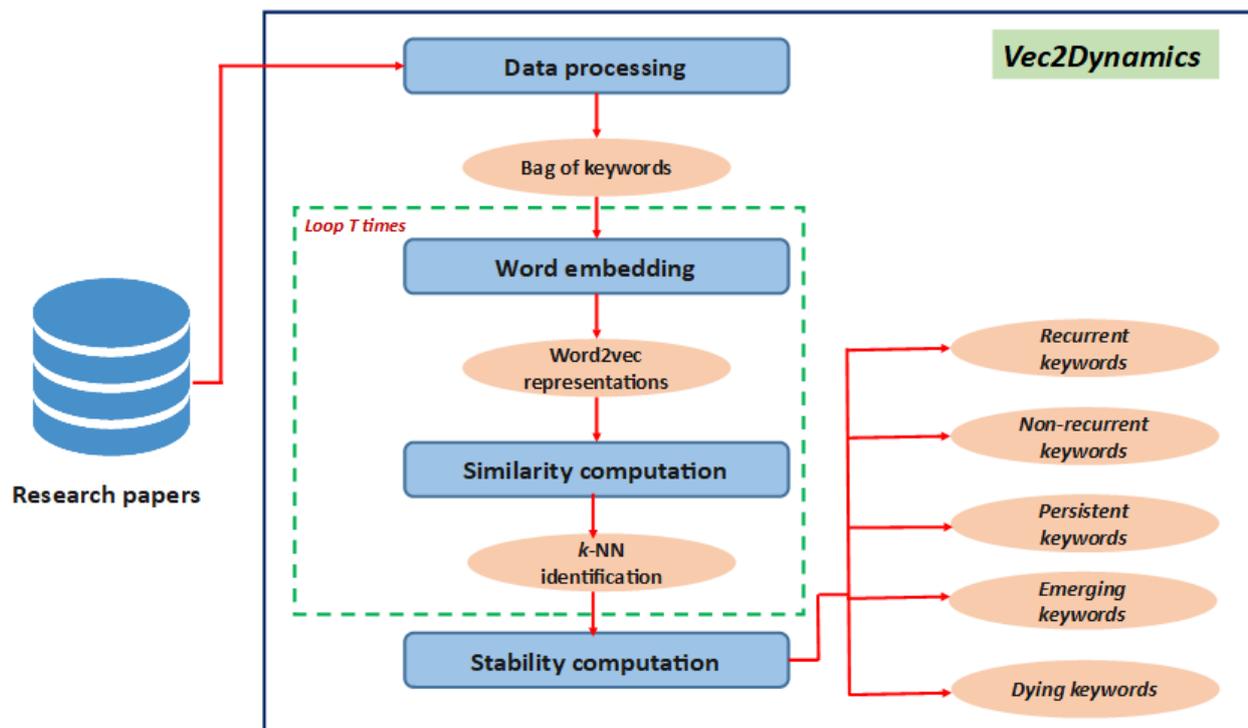


Figure 1. Vec2Dynamics architecture.

- i Data preprocessing. At this stage, we preprocess and clean up the textual content of research papers taking into account the specificity of scientific language. For instance, we consider the frequent use of bigrams in scientific language such as “*information system*”, and “*artificial intelligence*”, and we construct a bag of keywords where keywords are either *unigrams* or *bigrams*. *Unigram* or *1-gram* represents a one-word sequence, such as “*science*” or “*data*”, while a *bigram* or *2-gram* represents a two-word sequence of words like “*machine learning*” or “*data science*”. Data preprocessing consists then of two steps: (a) the removal of stop words, which refer to the words that appear very frequently like “*and*”, “*the*”, and “*a*”; and (b) the construction of bag of words where words are either *unigrams* or *bigrams*. More details on the data preprocessing stage are given in our previous work [15].

- ii Word embedding. At this stage, we adopt the *skip-gram* neural network architecture of the word2vec embedding model [32] to learn word vectors over time. This stage is repeated for each corpus  $P_{t,t=1,\dots,T}$  that corresponds to the corpus of all research papers in the  $t^{\text{th}}$  time window. More details will be given in Section 3.2.
- iii Similarity computation. After generating the vector representation of keywords, we apply *cosine similarity* between embedding vectors to find the  $k$ -nearest neighbors of each keyword. Recall that cosine similarity between two keywords  $w_i$  and  $w_j$  refers to the cosine measure between embedding vectors  $u_{w_i}$  and  $u_{w_j}$  as follows:

$$\text{similarity}(w_i, w_j) = \text{cosine}(u_{w_i}, u_{w_j}) = \frac{u_{w_i} \cdot u_{w_j}}{\|u_{w_i}\| \cdot \|u_{w_j}\|} \quad (1)$$

As with the previous stage, this stage of similarity computation is also repeated at each time window  $t$ .

- iv Stability computation. At this stage, we study the stability of  $k$ -NN of each keyword of interest over time in order to track the dynamics of the scientific literature. To do so, we define a *stability measure* (Equation (4)) that could be computed between the sets of  $k$ -NN keywords over two subsequent time windows  $t$  and  $(t + 1)$ . Based on the obtained stability values, we define four types of keywords/topics: *recurrent*, *non recurrent*, *persistent* and *emerging* keywords. More details will be given in Section 3.3.

### 3.2. Dynamic Word Embedding

*Vec2Dynamics* relies on a central stage of word embedding that learns word vectors in a temporal fashion in order to track the dynamics of scientific literature over time. To this end, we adopt skip-gram architecture of word2vec [32] that aims to predict the *context* given a word  $w_i$ . Note that the *context* is the span of words within a certain range before and after the current word  $w_i$  [35].

#### 3.2.1. Notation

We consider corpora of research papers collected across time. Formally, let  $P$  represent our corpora,  $P = (P_1, P_2, \dots, P_T)$  where each  $P_t$  is the corpus of all papers in the  $t^{\text{th}}$  time window, and  $\mathcal{V} = (w_1, w_2, \dots, w_V)$  the vocabulary that consists of  $V$  words present in  $P$ . It is possible that some  $w_i \in \mathcal{V}$  do not appear at all in some  $P_t$ . This happens because new keywords emerge while some old keywords die, something that is typical of scientific corpora. Let  $V_t$  denote the vocabulary that corresponds to  $P_t$  and  $|V_t|$  denote the corresponding vocabulary size used in training word embeddings at the  $t^{\text{th}}$  time window.

Given this time-tagged scientific corpora, our goal is to find a dense, low-dimensional vector representation  $u_{w_i}^t \in \mathbb{R}^N$ ,  $N \ll V_t$  for each word  $w_i \in V_t$  at each time window  $t = 1, \dots, T$ .  $N$  is the *dimensionality* of word vectors that corresponds to the length of the vector representations of words.

#### 3.2.2. Skip-Gram Model

The architecture of the Skip-Gram model is shown in Figure 2. the main idea of skip-gram is to predict the *context*  $c$  given a word  $w$ . Note that the *context* is a window around  $w$  of maximum size  $L$ . More formally, each word  $w \in W$  and each context  $c \in C$  are represented as vectors  $\vec{w} \in \mathbb{R}^d$  and  $\vec{c} \in \mathbb{R}^d$ , respectively, where  $W = \{w_1, \dots, w_V\}$  is the words vocabulary,  $C$  is the context vocabulary, and  $d$  is the embedding dimensionality. Recall that the vectors parameters are latent and need to be learned by maximising a function of products  $\vec{w} \cdot \vec{c}$ .

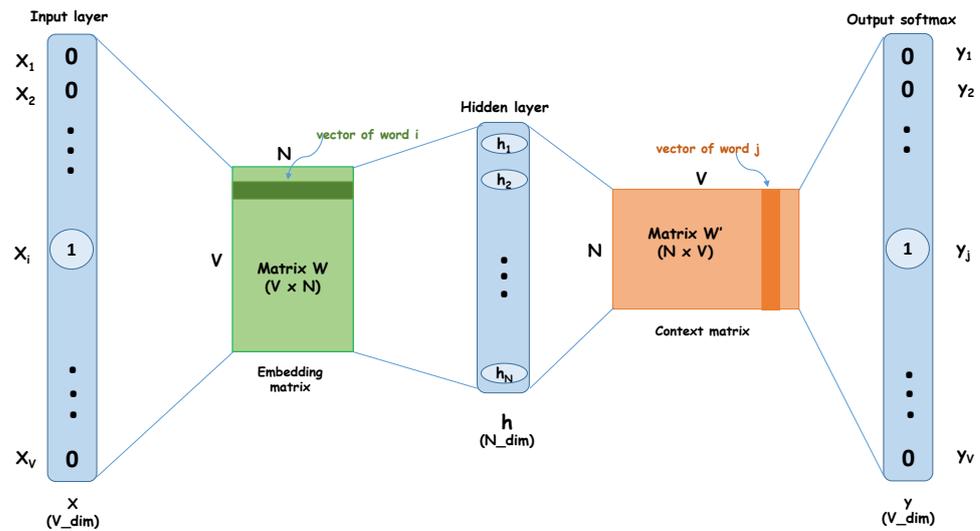


Figure 2. Skip-Gram Architecture,  $V$  here corresponds to  $V_i$  in our text.

More specifically, given the word sequence  $W$  resulted from the scientific corpus, the objective of skip-gram model is to maximise the average log probability:

$$L(W) = \frac{1}{V} \sum_{i=1}^V \sum_{-l \leq c \leq l, c \neq 0} \log Prob(w_{i+c} | w_i) \tag{2}$$

where  $l$  is the context size of a target word. Skip-gram formulates the probability  $Prob(w_c | w_i)$  using a softmax function as follows:

$$Prob(w_c | w_i) = \frac{\exp(\vec{w}_c \cdot \vec{w}_i)}{\sum_{w_c \in W} \exp(\vec{w}_c \cdot \vec{w}_i)} \tag{3}$$

where  $\vec{w}_i$  and  $\vec{w}_c$  are, respectively, the vector representations of target word  $w_i$  and context word  $w_c$ , and  $W$  is the word vocabulary. In order to make the model efficient for learning, the hierarchical softmax and negative sampling techniques are used following Mikolov et al. [36].

The hyperparameters of skip-gram model have been tuned followed our approach detailed in [37]. This outcomes of this approach revealed that the optimal hyperparameters for word2vec in a similar scientific corpora are respectively 200 and 6 for vector dimensionality  $N$  and the context window.

### 3.3. $k$ -NN Stability

After learning word embeddings on the scientific corpora in a temporal fashion, *Vec2Dynamics* uses the stability of  $k$ -nearest neighbors ( $k$ -NN) of word vectors as the objective to measure while tracking the dynamics of research keywords/topics. The  $k$ -NN keywords of a target keyword  $w_i$  correspond to the  $k$  keywords similar to  $w_i$ . Recall that cosine similarity is the measure used to calculate the similarity between two keywords  $w_i$  and  $w_j$ .

#### 3.3.1. Notation

Let  $S_{w_i}^{t-1}$  and  $S_{w_i}^t$  denote respectively the sets of  $k$ -NN of the keyword  $w_i$  over two successive time windows  $t - 1$  and  $t$ . We define  $\Psi_{w_i}^t$  the  $k$ -NN stability of  $w_i$  at the time

window  $t$  as the logarithmic ratio of the intersection between the two sets  $S_{w_i}^{t-1}$  and  $S_{w_i}^t$  to the difference between them. Formally,  $\Psi_{w_i}^t$  is defined as follows:

$$\Psi_{w_i}^t = \log_k \left( \frac{|S_{w_i}^t \cap S_{w_i}^{t-1}|}{0.5 \times |S_{w_i}^t \ominus S_{w_i}^{t-1}|} \right) \tag{4}$$

Recall that the two sets  $S_{w_i}^{t-1}$  and  $S_{w_i}^t$  are equal and their differences are symmetric as  $|S_{w_i}^{t-1}| = |S_{w_i}^t| = k$ , which justifies the multiplication by 0.5 in the denominator.

The  $k$ -NN stability  $\Psi_{w_i}^t$  ranges from  $-1$  to  $+1$ . We refer to  $-1$  ( $\log_k(\frac{1}{k})$ ) where no intersection exists between  $S_{w_i}^{t-1}$  and  $S_{w_i}^t$  ( $|S_{w_i}^t \cap S_{w_i}^{t-1}| = 0$ ) in order to prevent the indeterminate case where the numerator is null. On the other hand, we refer to  $+1$  ( $\log_k(k)$ ) to prevent the indeterminate case where the denominator is null when we have fusion of the two sets  $S_{w_i}^{t-1}$  and  $S_{w_i}^t$ . More formally, we define the stability  $\Psi_{w_i}^t$  as follows:

$$\Psi_{w_i}^t = \begin{cases} \log_k(\frac{1}{k}), & \text{if } S_{w_i}^t \cap S_{w_i}^{t-1} = \emptyset \\ \log_k(k), & \text{if } S_{w_i}^t = S_{w_i}^{t-1} = k \\ \text{Equation (4)}, & \text{otherwise} \end{cases} \tag{5}$$

After computing  $\Psi_{w_i}^t$  corresponding to each keyword of interest  $w_i$  over all time windows  $T$ , we compute the average of stability  $\Psi^t$  of  $n$  selected keywords of interest, where  $n \leq |V_t|$  at a time window  $t$  as follows:

$$\Psi^t = \frac{\sum_{i=1}^n \Psi_{w_i}^t}{n} \tag{6}$$

### 3.3.2. Interpretation

The computation of  $k$ -NN stability is based on the dynamism of the keywords appearing and disappearing in  $S_{w_i}^{t-1}$  and  $S_{w_i}^t$ . We can differentiate five types of keywords based on their dynamism: *recurrent keywords*, *non-recurrent keywords*, *persistent keywords*, *emerging keywords* and *dying keywords*. Algorithm 1 highlights the steps of finding these five types of keywords.

**Definition 1** (Recurrent keyword). *A word  $w_j$  is called recurrent if it appears recurrently in the  $k$ -NN subsequent sets  $S_{w_i}^t$  and  $S_{w_i}^{t_r}$ ,  $t_r \geq 2$ .*

**Definition 2** (Non-Recurrent keyword). *Contrary to recurrent keyword, a non-recurrent keyword does not appear in the  $k$ -NN subsequent sets; it appears in  $S_{w_i}^t$ , but never appears in  $S_{w_i}^{t_r}$ ,  $t_r \geq 2$ .*

**Definition 3** (Persistent keyword). *A word  $w_j$  is persistent if it appears in  $S_{w_i}^t$  and at least in  $S_{w_i}^{t+1}$ .*

**Definition 4** (Emerging keyword). *A word  $w_j$  is called emerging if it appears in the  $k$ -NN set  $S_{w_i}^{t+1} \setminus S_{w_i}^t$ .*

**Definition 5** (Dying keyword). *A word  $w_j$  is called dying if it appears in the  $k$ -NN set  $S_{w_i}^t \setminus S_{w_i}^{t+1}$ .*

We use these definitions to provide a fine-grained analysis of the main streams of keywords based on their appearance/disappearance and frequency of appearance. This helps to fully understand the evolution of scientific keywords over time.

**Algorithm 1:** Finding the types of keywords based on their dynamism.

---

**Input** : set of  $k$ -NN of a keyword  $w_i$ , time windows  $t$  and  $t_r$ , number of nearest neighbors  $k$ , number of keywords of interest  $n$

**Output**: the  $k$ -NN stability  $Psi$ , the class of keyword's dynamism  $C_{w_i}$  {recurrent; non-recurrent; persistent; emerging; dying}

---

```

1 for  $i \leftarrow 1$  to  $T$  do
2    $\Psi_{w_i}^t \leftarrow 0$ ;
3    $\Psi^t \leftarrow 0$ ;
4   for  $j \leftarrow 1$  to  $n$  do
5     if  $(S_{w_i}^t \cap S_{w_j}^{trS} = \emptyset \ \& \ tr = t - 1)$  then
6        $\Psi_{w_j}^t \leftarrow \log_k(\frac{1}{k})$ ;
7     end
8     else if  $(S_{w_i}^t \cap S_{w_j}^{t-1} = \emptyset \ \& \ tr = t - 1)$  then
9        $\Psi_{w_j}^t \leftarrow \log_k(\frac{1}{k})$ ;
10    end
11    else
12       $\Psi_{w_j}^t \leftarrow \log_k(\frac{|S_{w_i}^t \cap S_{w_j}^{t-1}|}{0.5 \times |S_{w_i}^t \oplus S_{w_j}^{t-1}|})$ ;
13    end
14  end
15   $\Psi^t \leftarrow \frac{\sum_{i=1}^n \Psi_{w_i}^t}{n}$ ;
16  if  $(w_i \in S_{w_i}^t \ \& \ w_i \in S_{w_i}^{tr} \ \& \ tr \geq 2)$  then
17     $C_{w_i} \leftarrow$  "recurrent";
18  end
19  if  $(w_i \in S_{w_i}^t \ \& \ w_i \notin S_{w_i}^{tr} \ \& \ tr \geq 2)$  then
20     $C_{w_i} \leftarrow$  "non-recurrent";
21  end
22  if  $(w_i \in S_{w_i}^t \ \& \ w_i \in S_{w_i}^{tr} \ \& \ tr \geq 1)$  then
23     $C_{w_i} \leftarrow$  "persistent";
24  end
25  if  $(w_i \in S_{w_i}^{tr} \setminus S_{w_i}^t \ \& \ tr = t+1)$  then
26     $C_{w_i} \leftarrow$  "emerging";
27  end
28  if  $(S_{w_i}^t \setminus S_{w_i}^{t+1} \ \& \ tr = t+1)$  then
29     $C_{w_i} \leftarrow$  "dying";
30  end
31 end

```

---

#### 4. Experiments

To track the dynamics of the scientific literature in the area of Machine Learning, we evaluate *Vec2Dynamics* on a time-stamped text corpora extracted from NIPS (Neural Information Processing Systems)—recently abbreviated as NeurIPS—conference proceedings. The NIPS conference has been chosen because it is a long standing venue in the area of machine learning with more than 30 editions. This is important for the work proposed in this paper that requires history in publications to perform the computational history of science. NIPS is ranked top 2 conference in the area of machine learning and artificial intelligence, according to *Guide2Research* rankings (<https://www.guide2research.com/topconf/machine-learning>, accessed on 2 February 2021). It comes after *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, which is more specialised in computer vision applications. NIPS is then a more generic venue and covers a wide range of machine learning topics including computer vision. It can

be then considered the top 1 conference in generic machine learning conferences like the *International Conference on Machine Learning (ICML)*, *International Conference on Learning Representations (ICLR)* and *European Conference on Machine Learning and Knowledge Discovery in Databases (ECMLPKDD)* that are ranked top 6, top 17 and top 104, respectively. Finally, aligning with its reputation, the NIPS dataset is made publicly available on Kaggle (<https://www.kaggle.com/benhamner/nips-papers/data>, accessed on 20 October 2017), where the papers database is easily manageable, with the papers' features, such as the title, the abstract and the paper text, which is time-stamped. The availability of this database makes from the task of data collection an efficient task.

We demonstrate that our approach delves into the content of research papers and provides a deep descriptive analysis of the literature of Machine Learning over 30 years by following a temporal embedding and analysing the resulting dynamic  $k$ -NN keywords over time.

#### 4.1. NIPS Dataset

The dataset used in this analysis represents 30 years of NIPS conference papers published between 1987 and 2016, with a total of 6562 papers. The dataset is available on Kaggle (<https://www.kaggle.com/benhamner/nips-2015-papers/data>, accessed on 20 October 2017) and stores information about papers, authors and the relation (papers-authors). The papers database—that defines six features for each paper: the *id*, the *title*, the *event type*, i.e., poster, oral or spotlight presentation, the *PDF name*, the *abstract* and the *paper text*—is used.

The dataset was first preprocessed following the steps described in Section 3.1. Then, in order to study the dynamics of the scientific literature of Machine Learning by tracking the  $k$ -NN of keywords/topics over time, we split the NIPS publications between 1987 and 2016 into ten 3-year time windows. The length of the time window relies on the study conducted by Anderson et al. [4] on the evolution of the scientific topics. Their results showed that the interval of three years was successful to track the flow of scientific corpora. The statistics of the dataset are given in Table 1. This step of dataset splitting into sub-datasets may justify that there are no potential validation threats because for every sub-dataset the approach has been applied independently from the other sub-datasets, which may be considered as a new dataset.

**Table 1.** Statistics of NIPS dataset.

Time Window	Papers	Words	Vocabulary
From 1987 to 1989	288	16,273	9147
From 1990 to 1992	417	465,169	169,728
From 1993 to 1995	453	914,871	166,954
From 1996 to 1998	456	1,387,070	173,341
From 1999 to 2001	499	1,943,821	197,845
From 2002 to 2004	615	2,716,271	264,241
From 2005 to 2007	631	3,595,398	292,681
From 2008 to 2010	807	4,847,535	379,086
From 2011 to 2013	1037	6,501,435	480,440
From 2014 to 2016	1386	8,732,443	610,383

Table 1 shows an increase in the number of papers per 3-years in the period 1987–2016. The average 3-annual growth rate of publications is around 20%, exceeding 33% in the time window 2014–2016. The findings revealed that there are a potential or possible emerging research keywords/topics within the new evolving papers. This could be justified by the constant growth rate of unique words that reaches 29.52% in the timespan 2008–2010, with an average rate of 17.98% overall, excluding the first time window 1987–1989 where the vocabulary size was very small and may bias the result.

#### 4.2. Results and Discussion

We evaluate *Vec2Dynamics* on tracking the dynamics of Machine Learning literature. To do so, we study the impact of the use of temporal word embeddings on tracing the evolution of the main streams of Machine Learning keywords. To this end, NIPS publication—published between 1987 and 2016—have been used.

For each time window  $t$ , we generated a corpus  $P_t$  of all publications published during this time period. Then, after preprocessing as described in Section 4.1, we trained the skip-gram model of word2vec at every  $t$  with the hyperparameters vector dimensionality  $N = 200$  and context window = 6. The choice of these hyperparameters is supported by our previous findings [37] that concluded after investigation that these hyperparameters are optimal for NIPS corpora.

After each training at time window  $t$ , we computed the similarities of *keywords of interest* following Equation (1). The keywords of interest correspond to the top 100 bigrams extracted from the titles of the publications [15]. From these 100 bigrams, we kept only the bigrams that appeared in the highest two levels of the *Computer Science Ontology* (CSO) (<http://skm.kmi.open.ac.uk/cso/>, accessed on 5 September 2019) [38]. These two levels correspond to the upper level of the taxonomy that represent the general terms and categories in Computer Science. This is justified by our aim to keep the keywords as generic as possible reflecting the topics rather than the fine-grained sub-topics and detailed techniques that the ontology illustrates. This restricted our keywords of interest to only 20 bigrams. The choice of the knowledge-based resource (CS ontology) to evaluate the output of word embeddings has been inspired by the work performed by [20] that enhanced word2vec with geological ontology for knowledge discovery in the area of geoscience and minerals.

At every time window  $t$  and for each keyword of interest  $w_i^* \in V_t$ , we selected the  $k$ -NN based on cosine similarity. In this study,  $k$  is set to 10. This choice is motivated by the study performed by Hall et al., on the history trends in *computational linguistics*. Their investigation showed that the set of 10 words was successful to represent each topic.

To select the 10-NN, we followed two steps: first, we took the top 300 similar keywords returned by cosine similarity, and then we filtered out from these 300 keywords the ones that belong to the first or the second level of the CSO aiming to keep the keywords as generic as possible reflecting the topics rather than the fine-grained sub-topics. This aim justifies the choice of 300 neighboring keywords at the beginning. Indeed, this value was chosen experimentally; we took different  $k$  values in the interval  $\{50, 100, 200, 300\}$ . Among these values, only 300 guaranteed the existence of at least 10 nearest neighbors keywords belonging to CSO. Recall that the choice of  $k$  values was limited by the satisfaction of a similarity threshold to fulfill in order to keep the keywords as close as possible. The choice of this threshold, which is 0.37, was based on the work done by Orkphol and Yung [39] on cosine similarity threshold with word2vec.

After defining the  $k$ -NN of each keyword of interest  $w_i^*$  at every time window  $t$ , we computed its  $k$ -NN stability that corresponds to the logarithmic ratio of (a) the intersection between the two sets of 10-NN of  $w_i^*$  over two subsequent time windows to (b) the symmetric difference between them as given per Equation (4). Therefore, the  $k$ -NN stability of  $w_i^*$  at a time window  $t$  corresponds to the output of Equation (4) with the two sets of  $k$ -NN of  $w_i^*$  at  $t$  and  $t - 1$  as inputs. For instance, the  $k$ -NN stability of  $w_i^*$  at the time window 2002–2004 describes how the  $k$ -NN of  $w_i^*$  changed between the time window 1999–2001 and the time window 2002–2004.

Table 2 shows the  $k$ -NN stability of the top 20 bigrams studied in this paper, as well as the average stability per time window as defined in Equation (6). The table does not show the results related to the first two time windows 1987–1989 and 1990–1992 because (i) the first window does not contain the studied keywords of interest, given that the vocabulary size is too small which justifies the difficulty to build bigrams; (ii) the second window refers to the window  $t - 1$  that is used to measure the stability at the time window 1993–1995. The overall results reveal that the  $k$ -NN stability was mostly

negative. For instance, the average stability ranges from  $-0.24$  to  $-0.531$  (which is not surprisingly) tracing the amount of disruption in the field of Machine Learning. The lowest stability was detected in the time period 1996–1998. This suggests that the field may have been more innovative and receptive of new topics/keywords at the time. This is indeed supported by the timeline of Machine Learning ([https://en.wikipedia.org/wiki/Timeline\\_of\\_machine\\_learning](https://en.wikipedia.org/wiki/Timeline_of_machine_learning), accessed on 20 September 2019) [40] that shows an interesting amount of achievements and discoveries in the field during the period of 1995 till 1998 such as the discovery of *Random Forest Algorithm* [41], *Support Vector Machine* [42], *LSTM (Long-Short-Term Memory)* [43] and the achievement of *IBM Deep Blue* in the world champion at chess [44]. However, the highest stability was shown in the time window 2014–2016. At that time, it seemed that the field has reached a certain maturity which makes the  $k$ -NN stability high. To see if this was in fact the case, we analysed the  $k$ -NN of all keywords of interest at this time period. Interestingly, we found that 70% of keywords share one of the following nearest neighbors keywords {"*neural\_network*", "*deep\_learning*", "*big\_data*"}. The shift toward *big data analytics* and *deep learning* is well known in the field in the last years of the analysis. For instance, all discoveries in this period are founded on *deep neural networks* and applied to *big data* such as *Google Brain* (2012) [45], *AlexNet* (2012) [46], *DeepFace* (2014) [47], *ResNet* (2015) [48] and *U-Net* (2015) [49].

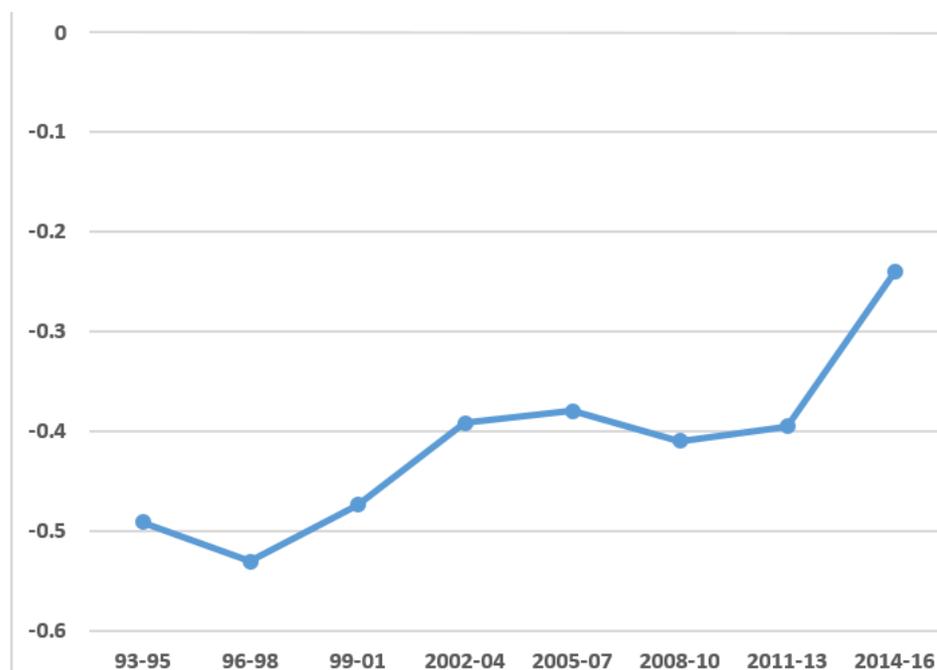
**Table 2.**  $k$ -NN stability of top 20 bigrams, n/a means that the keyword does not exist either in this time window or in the previous time window.

Keywords	93–95	96–98	99–01	02–04	05–07	08–10	11–13	14–16
machine_learning	n/a	-0.6	-0.95	-0.6	-0.6	0	0.176	0
neural_network	-0.6	-0.6	-0.18	-0.95	0	-0.368	-0.18	0
supervised_learning	-0.95	-1	-1	-0.368	-0.368	-0.368	-0.368	-0.18
unsupervised_learning	-0.18	-0.6	0	-0.95	-0.6	-0.368	-0.368	-0.6
reinforcement_learning	-1	0	-0.368	-0.368	-0.18	-0.368	0	0
time_series	-0.48	-0.95	-0.95	-1	-1	-1	-1	-0.95
artificial_intelligence	n/a	n/a	n/a	n/a	-0.3	0	n/a	-0.368
gaussian_process	n/a	n/a	0	0	0	-0.18	-0.368	0.176
semi_supervised	n/a	n/a	n/a	n/a	-0.18	-0.6	-0.6	-0.6
active_learning	n/a	n/a	n/a	-0.18	-1	-1	-1	-1
decision_trees	n/a	-0.6	-0.6	-0.545	-0.95	-0.368	-0.95	0.176
dimensionality_reduction	n/a	n/a	-1	-0.18	-0.368	-0.368	-0.95	-0.6
dynamic_programming	-0.95	0	0.176	-0.368	0	-0.6	-0.6	0
gradient_descent	0.176	-0.18	-0.368	0	-0.368	-0.18	0.602	0.176
hidden_markov	0	-0.6	-0.6	-0.18	-0.18	-0.95	-0.368	-0.368
mutual_information	-0.95	-0.397	-1	-0.6	-1	-0.368	n/a	-0.552
nearest_neighbor	-0.18	-0.18	-0.18	-0.18	-0.18	-0.368	-0.18	0
pattern_recognition	-0.18	-0.95	-0.368	-0.18	n/a	n/a	-0.6	n/a
monte_carlo	-0.6	-0.6	-0.18	-0.18	0.176	0	-0.18	0
graphical_model	n/a	n/a	n/a	0.176	-0.18	-0.368	-0.18	0
<b>Average</b>	-0.491	<b>-0.531</b>	-0.474	-0.392	-0.38	<b>-0.41</b>	-0.395	-0.24

For better interpretation of the rise and decline of  $k$ -NN stability over time, Figure 3 illustrates the average of  $k$ -NN stability over eight time windows. The sharp increase in stability is readily apparent in the last time window as discussed above. On the other hand, we can see the decrease in stability in the period 1996–1998 and then in the period 2008–2010. The former is interpreted previously. However, the later refers to a time window that represented an important time frame for *ImageNet* [50] that was the catalyst for the AI boom of the 21st century. This may justify the decline in stability at that time.

Regarding the time period from 1999 until 2007, Figure 3 shows a steady  $k$ -NN stability. This suggests that the field was stable at that time; it was broadly exploring and applying what has been discovered in the disrupted period 1993–1995. This suggestion is confirmed by Machine Learning timeline that does not report any topics or discoveries having been

prominent at that time except the release of *Torch* [51] which is actually a software library for Machine Learning; it is indeed a tool and does not have anything to do with new topics.



**Figure 3.** *k*-NN average stability over time.

For overall results on NIPS publications, *Vec2Dynamics* shows promising findings in tracking the dynamics of Machine Learning literature. As a proof of evidence, *Vec2Dynamics*, applied to the keyword of interest “*machine\_learning*”, detects interesting patterns as shown in Figure 4. Each figure (Figure 4a–g) depicts the Venn diagram of two subsequent sets of *k*-NN keywords of the keyword “*machine\_learning*”. As these figures show, “*machine\_learning*” seems to have been stabilised significantly over time. For instance, the overlap of the two sets  $S_{machine\_learning}^{t-1}$  and  $S_{machine\_learning}^t$  has increased gradually to pass from only one keyword between the sets of the time periods 1996–1998 and 1999–2001 (Figure 4b) to six keywords between the sets of the time periods 2008–2010 and 2011–2013 (Figure 4f).

In addition, the fine-grained analysis of the main streams of keywords reveals the different types of keywords based on their appearance/disappearance and their frequency of appearance in the subsequent sets of *k*-NN. For instance, in the case of “*machine\_learning*”, the keywords “*computer\_vision*”, “*bioninformatics*”, “*robotics*” and “*economics*” for example are *recurrent* as they appeared recurrently in more than four windows. However, the keyword “*html*” is *non-recurrent* because it appeared only in two windows and then disappeared. On the other hand, “*nlp*” is considered an *emerging* keyword in the time window 2002–2004 due to its first appearance. This is insightful because the field of *natural language processing* has seen a significant progress after the vast quantities of text flooding the World Wide Web in the late 1990s, notably by information extraction and automatic summarising [52]. However, the keyword “*mathematics*” is considered a *dying* keyword as it completely disappeared after the time period 1996–1998. This could be justified by the fact that early Machine Learning approaches and algorithms were developed based on mathematical foundations such as *Bayes’ theorem*, *Markov chains*, *Least Squares*, etc; that is why early Machine Learning researchers have extensively investigated mathematics in their literature comparing to the present ones that focus more on the applications.

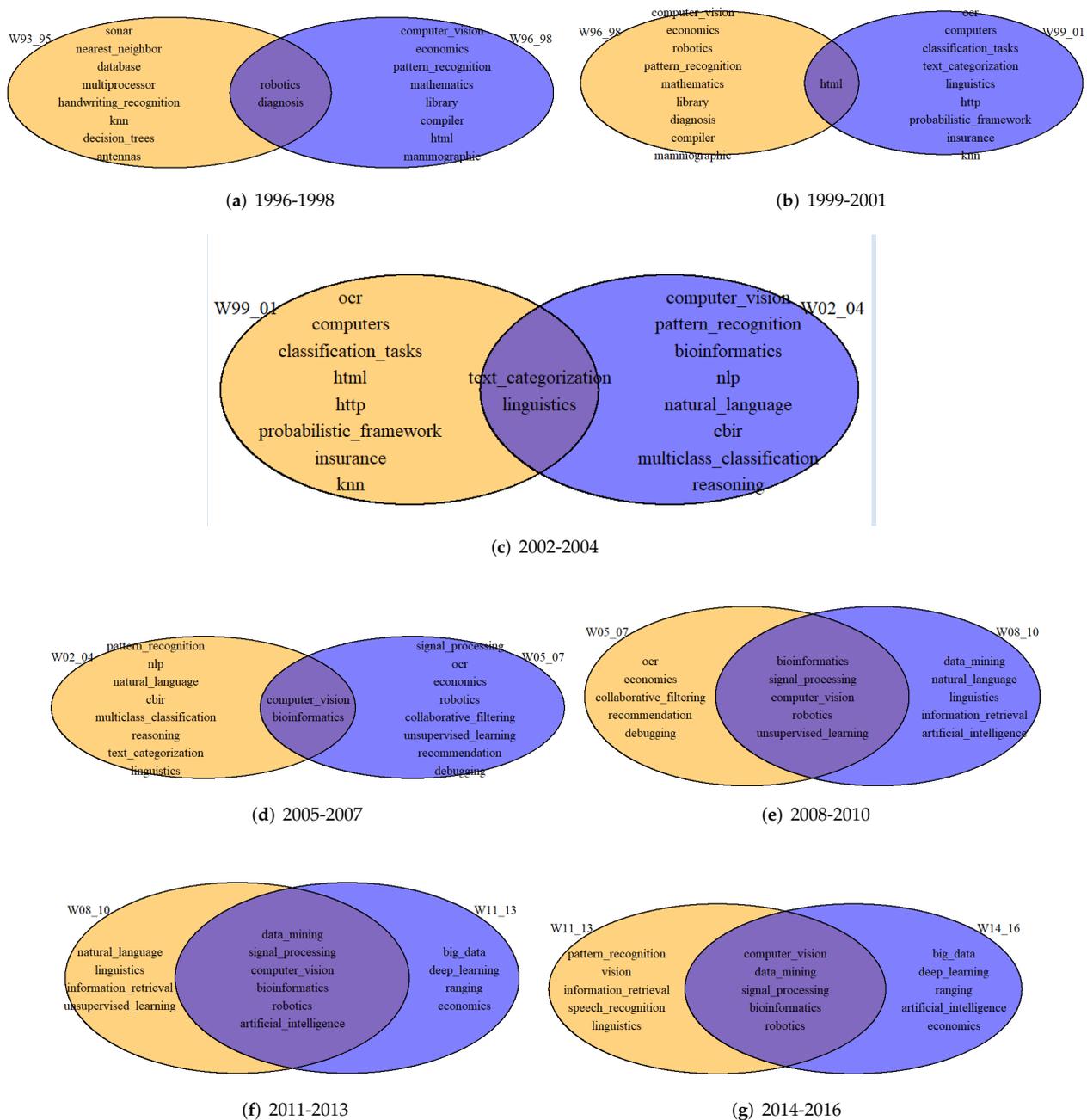


Figure 4. Venn Diagrams of “machine\_learning” in the time frame 1996–2016.

Overall, *Vec2Dynamics* shows a great potential to track and explore the dynamics of Machine Learning keywords/topics over time. Both numerical and visual analyses show the effectiveness of our approach to trace the history of Machine Learning literature exactly as Machine Learning timeline does.

### 5. Conclusions and Future Work

In this study, we have proposed *Vec2Dynamics*, a new approach to analyse and explore the scientific literature. To this end, *Vec2Dynamics* followed an innovative way by leveraging word embedding techniques, namely word2vec, to delve into the paper content and track the dynamics of *k*-nearest neighbors keywords of a keyword of interest. To do so, our approach trained temporal embeddings over ten 3-year time windows. Then, after each

training, it computed the similarities of keywords of interest and accordingly it defined the  $k$ -NN keywords of each keyword of interest. Afterward, it computed the stability of  $k$ -NN over every two subsequent time windows.

By applying the approach in the research area of Machine Learning, we have found numerical and visual evidence of computing the stability of  $k$ -NN keywords over time on tracking the dynamics in science. A research area with growing  $k$ -NN stability is likely to subsequently gain maturity, while the contrary is also true; it refers to an emerging area with new topics becoming more prominent.

As future direction, we plan to apply our approach on different disciplines such as *physics*, *biology* and *medicine*. Then, we plan to explore different clustering techniques, namely *Chameleon* [53], for tracking the dynamism of keywords between different clusters over time in the scientific corpora.

**Author Contributions:** Conceptualization, A.D. and M.M.G.; methodology, A.D. and M.M.G.; software, A.D.; validation, A.D.; resources, A.D.; data curation, A.D.; writing—original draft preparation, A.D.; writing—review and editing, M.M.G.; visualization, A.D.; supervision, M.M.G., R.M.A.A. and J.B. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The dataset used in this study is publicly available on Kaggle at <https://www.kaggle.com/benhamner/nips-2015-papers/data>, accessed on 20 October 2017.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

ACL	Association for Computational Linguistics
CLR	Computational Literature Review
CS	Computer Science
CSO	Computer Science Ontology
DCA	Document-Citation Analysis
FoS	Field of Study
IS	Information Science
$k$ -NN	$k$ - Nearest Neighbors
LDA	Latent Dirichlet Allocation
LSTM	Long-Short-Term Memory
MAG	Microsoft Academic Graph
ML	Machine Learning
NIPS	Neural Information Processing Systems
SE	software engineering
TAM	Technology Acceptance Model

## References

- Xia, F.; Wang, W.; Bekele, T.M.; Liu, H. Big Scholarly Data: A Survey. *IEEE Trans. Big Data* **2017**, *3*, 18–35. [CrossRef]
- Yu, Z.; Menzies, T. FAST2: An intelligent assistant for finding relevant papers. *Expert Syst. Appl.* **2019**, *120*, 57–71. [CrossRef]
- An, Y.; Han, M.; Park, Y. Identifying dynamic knowledge flow patterns of business method patents with a hidden Markov model. *Scientometrics* **2017**, *113*, 783–802. [CrossRef]
- Anderson, A.; McFarland, D.; Jurafsky, D. Towards a Computational History of the ACL: 1980–2008. In Proceedings of the ACL-2012 Special Workshop on Rediscovering 50 Years of Discoveries, Jeju Island, Korea, 10 July 2012; pp. 13–21.
- Effendy, S.; Yap, R.H. Analysing Trends in Computer Science Research: A Preliminary Study Using The Microsoft Academic Graph. In Proceedings of the 26th International Conference on World Wide Web Companion, Perth, Australia, 3–7 April 2017; pp. 1245–1250.
- Hall, D.; Jurafsky, D.; Manning, C.D. Studying the History of Ideas Using Topic Models. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '08, Honolulu, HI, USA, 25–27 October 2008; pp. 363–371.

7. Hoonlor, A.; Szymanski, B.K.; Zaki, M.J. Trends in Computer Science Research. *Commun. ACM* **2013**, *56*, 74–83. [[CrossRef](#)]
8. Hou, J.; Yang, X.; Chen, C. Emerging trends and new developments in information science: A document co-citation analysis (2009–2016). *Scientometrics* **2018**, *115*, 869–892. [[CrossRef](#)]
9. Mortenson, M.J.; Vidgen, R. A Computational Literature Review of the Technology Acceptance Model. *Int. J. Inf. Manag.* **2016**, *36*, 1248–1259. [[CrossRef](#)]
10. Rossetto, D.E.; Bernardes, R.C.; Borini, F.M.; Gattaz, C.C. Structure and evolution of innovation research in the last 60 years: review and future trends in the field of business through the citations and co-citations analysis. *Scientometrics* **2018**, *115*, 1329–1363. [[CrossRef](#)]
11. Santa Soriano, A.; Lorenzo Álvarez, C.; Torres Valdés, R.M. Bibliometric analysis to identify an emerging research area: Public Relations Intelligence. *Scientometrics* **2018**, *115*, 1591–1614. [[CrossRef](#)]
12. Zhang, C.; Guan, J. How to identify metaknowledge trends and features in a certain research field? Evidences from innovation and entrepreneurial ecosystem. *Scientometrics* **2017**, *113*, 1177–1197. [[CrossRef](#)]
13. Taskin, Z.; Al, U. A content-based citation analysis study based on text categorization. *Scientometrics* **2018**, *114*, 335–357. [[CrossRef](#)]
14. Ruas, T.; Grosky, W.; Aizawa, A. Multi-sense embeddings through a word sense disambiguation process. *Expert Syst. Appl.* **2019**, *136*, 288–303. [[CrossRef](#)]
15. Dridi, A.; Gaber, M.M.; Azad, R.M.A.; Bhogal, J. Leap2Trend: A Temporal Word Embedding Approach for Instant Detection of Emerging Scientific Trends. *IEEE Access* **2019**, *7*, 176414–176428. [[CrossRef](#)]
16. Weismayer, C.; Pezenka, I. Identifying emerging research fields: A longitudinal latent semantic keyword analysis. *Scientometrics* **2017**, *113*, 1757–1785. [[CrossRef](#)]
17. Picasso, A.; Merello, S.; Ma, Y.; Oneto, L.; Cambria, E. Technical analysis and sentiment embeddings for market trend prediction. *Expert Syst. Appl.* **2019**, *135*, 60–70. [[CrossRef](#)]
18. Boyack, K.W.; Smith, C.; Klavans, R. Toward predicting research proposal success. *Scientometrics* **2018**, *114*, 449–461. [[CrossRef](#)]
19. Liu, Y.; Huang, Z.; Yan, Y.; Chen, Y. Science Navigation Map: An Interactive Data Mining Tool for Literature Analysis. In Proceedings of the 24th International Conference on World Wide Web, WWW '15 Companion, Florence, Italy, 18–22 May 2015; pp. 591–596.
20. Qiu, Q.; Xie, Z.; Wu, L.; Li, W. Geoscience keyphrase extraction algorithm using enhanced word embedding. *Expert Syst. Appl.* **2019**, *125*, 157–169. [[CrossRef](#)]
21. Alam, M.M.; Ismail, M.A. RTRS: A recommender system for academic researchers. *Scientometrics* **2017**, *113*, 1325–1348. [[CrossRef](#)]
22. Dey, R.; Roy, A.; Chakraborty, T.; Ghosh, S. Sleeping beauties in Computer Science: Characterization and early identification. *Scientometrics* **2017**, *113*, 1645–1663. [[CrossRef](#)]
23. Effendy, S.; Jahja, I.; Yap, R.H. Relatedness Measures Between Conferences in Computer Science: A Preliminary Study Based on DBLP. In Proceedings of the 23rd International Conference on World Wide Web, Seoul, Korea, 7–11 April 2014; pp. 1215–1220.
24. Effendy, S.; Yap, R.H.C. The Problem of Categorizing Conferences in Computer Science. In *Research and Advanced Technology for Digital Libraries*; Fuhr, N., Kovács, L., Risse, T., Nejdl, W., Eds.; Springer: Cham, Switzerland, 2016; pp. 447–450.
25. Kim, S.; Hansen, D.; Helps, R. Computing research in the academy: Insights from theses and dissertations. *Scientometrics* **2018**, *114*, 135–158. [[CrossRef](#)]
26. Glass, R.; Vessey, I.; Ramesh, V. Research in software engineering: An analysis of the literature. *Inf. Softw. Technol.* **2002**, *44*, 491–506. [[CrossRef](#)]
27. Schlagenhauer, C.; Amberg, M. A descriptive literature review and classification framework for gamification in information systems. In Proceedings of the Twenty-Third European Conference on Information Systems (ECIS), Münster, Germany, 26–29 May 2015; pp. 1–15.
28. Martin, P.Y.; Turner, B.A. Grounded Theory and Organizational Research. *J. Appl. Behav. Sci.* **1986**, *22*, 141–157. [[CrossRef](#)]
29. Salatino, A.A.; Osborne, F.; Motta, E. How are topics born? Understanding the research dynamics preceding the emergence of new areas. *PeerJ Comput. Sci.* **2017**, *3*, e119. [[CrossRef](#)]
30. He, J.; Chen, C. Predictive Effects of Novelty Measured by Temporal Embeddings on the Growth of Scientific Literature. *Front. Res. Metrics Anal.* **2018**, *3*, 9. [[CrossRef](#)]
31. Dridi, A.; Gaber, M.M.; Azad, R.M.A.; Bhogal, J. DeepHist: Towards a Deep Learning-based Computational History of Trends in the NIPS. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; pp. 1–8.
32. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed Representations of Words and Phrases and their Compositionality. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 3111–3119.
33. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; Volume 14, pp. 1532–1543.
34. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching Word Vectors with Subword Information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146. [[CrossRef](#)]
35. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. *arXiv* **2013**, arXiv:1301.3781.

36. Mikolov, T.; Yih, W.t.; Zweig, G. Linguistic Regularities in Continuous Space Word Representations. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Atlanta, GA, USA, 9–14 June 2013; pp. 746–751.
37. Dridi, A.; Gaber, M.M.; Azad, R.M.A.; Bhogal, J.  $k$ -NN Embedding Stability for word2vec Hyper-Parametrisation in Scientific Text. In *International Conference on Discovery Science*; Springer: Cham, Switzerland, 2018; pp. 328–343.
38. Osborne, F.; Motta, E. Mining Semantic Relations between Research Areas. In *The Semantic Web—ISWC 2012*; Cudré-Mauroux, P., Heflin, J., Sirin, E., Tudorache, T., Euzenat, J., Hauswirth, M., Parreira, J.X., Hendler, J., Schreiber, G., Bernstein, A., et al., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 410–426.
39. Orkphol, K.; Yang, W. Word Sense Disambiguation Using Cosine Similarity Collaborates with Word2vec and WordNet. *Future Internet* **2019**, *11*, 114. [[CrossRef](#)]
40. Wikipedia. Timeline of Machine Learning. 2022. Available online: [https://en.wikipedia.org/wiki/Timeline\\_of\\_machine\\_learning](https://en.wikipedia.org/wiki/Timeline_of_machine_learning) (accessed on 1 December 2021).
41. Ho, T.K. Random Decision Forests. In Proceedings of the Third International Conference on Document Analysis and Recognition, ICDAR'95, Montreal, QC, Canada, 14–16 August 1995; Volume 1, pp. 278–282.
42. Cortes, C.; Vapnik, V. Support-Vector Networks. *Mach. Learn.* **1995**, *20*, 273–297. [[CrossRef](#)]
43. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]
44. Campbell, M.; Hoane, A.J., Jr.; Hsu, F.H. Deep Blue. *Artif. Intell.* **2002**, *134*, 57–83. [[CrossRef](#)]
45. Le, Q.V.; Ranzato, M.; Monga, R.; Devin, M.; Chen, K.; Corrado, G.S.; Dean, J.; Ng, A.Y. Building High-level Features Using Large Scale Unsupervised Learning. In Proceedings of the 29th International Conference on International Conference on Machine Learning, ICML'12, Edinburgh, UK, 26 June–1 July 2012; Omnipress: Madison, WI, USA, 2012; pp. 507–514.
46. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems, NIPS'12, Lake Tahoe, NV, USA, 3–6 December 2012; Curran Associates Inc.: Red Hook, NY, USA, 2012; Volume 1, pp. 1097–1105.
47. Taigman, Y.; Yang, M.; Ranzato, M.; Wolf, L. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR'14, Columbus, OH, USA, 23–28 June 2014; IEEE Computer Society: Washington, DC, USA, 2014; pp. 1701–1708.
48. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
49. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer: Cham, Switzerland, 2015.
50. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Li, F.F. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
51. Collobert, R.; Bengio, S.; Mariéthoz, J. Torch: A modular machine learning software library. In *Technical Report IDIAP-RR 02-46*; IDIAP: Martigny, Switzerland, 2002.
52. Mani, I.; Maybury, M.T. *Advances in Automatic Text Summarization*; MIT Press: Cambridge, MA, USA, 1999.
53. Karypis, G.; Kumar, V. Chameleon: Hierarchical clustering using dynamic modeling. *Computer* **1999**, *32*, 68–75. [[CrossRef](#)]