scientific data

OPEN

DATA DESCRIPTOR

DUKweb, diachronic word representations from the UK Web Archive corpus

Adam Tsakalidis[1,2], Pierpaolo Basile[3], Marya Bazzi[1,4,5], Mihai Cucuringu[1,5] & Barbara McGillivray[1,6] ✉

Lexical semantic change (detecting shifts in the meaning and usage of words) is an important task for social and cultural studies as well as for Natural Language Processing applications. Diachronic word embeddings (time-sensitive vector representations of words that preserve their meaning) have become the standard resource for this task. However, given the significant computational resources needed for their generation, very few resources exist that make diachronic word embeddings available to the scientific community. In this paper we present DUKweb, a set of large-scale resources designed for the diachronic analysis of contemporary English. DUKweb was created from the JISC UK Web Domain Dataset (1996–2013), a very large archive which collects resources from the Internet Archive that were hosted on domains ending in '.uk'. DUKweb consists of a series word co-occurrence matrices and two types of word embeddings for each year in the JISC UK Web Domain dataset. We show the reuse potential of DUKweb and its quality standards via a case study on word meaning change detection.

Background & Summary

Word embeddings, dense low-dimensional representations of words as real-number vectors[1], are widely used in many Natural Language Processing (NLP) applications, such as part-of-speech tagging, information retrieval, question answering, sentiment analysis, and are employed in other research areas, including biomedical sciences[2] and scientometrics[3]. One of the reasons for this success is that such representations allow us to perform vector calculations in geometric spaces which can be interpreted in semantic terms (i.e. in terms of the similarity in the meaning of words). This follows the so-called distributional hypothesis[4], according to which words occurring in a given word's context contribute to some aspects of its meaning, and semantically similar words share similar contexts. In Firth's words this is summarized by the quote "You shall know a word by the company it keeps"[5].

Vector representations of words can take various forms, including count vectors, random vectors, and word embeddings. The latter are nowadays most commonly used in NLP research and are based on neural networks which transform text data into vectors of typically 50–300 dimensions. One of the most popular approaches for generating word embeddings is word2vec[1]. A common feature of such word representations is that they are labour-intensive and time-consuming to build and train. Therefore, rather than training embeddings from scratch, in NLP it is common practice to use existing pre-trained embeddings which have been made available to the community. These embeddings have typically been trained on very large web resources, for example Twitter, Common Crawl, Gigaword, and Wikipedia[6,7].

Over the past few years NLP research has witnessed a surge in the number of studies on diachronic word embeddings[8,9]. One notable example of this emerging line of research is[10], where the authors proposed a method for detecting semantic change using word embeddings trained on the Google Ngram corpus[11] covering 8.5 hundred billion words from English, French, German, and Chinese historical texts. The authors have released the trained word embeddings on the project page[12]. The embeddings released in[10] have been successfully used in subsequent studies[13,14] and over time further datasets of diachronic embeddings have been made available to the scientific community. The authors of[15] released word2vec word embeddings for every 5 year-period, trained on the 10 million 5-grams from the English fiction portion of the Google Ngram corpus[16]. The authors of[17] have released different versions of word2vec embeddings trained on the Eighteenth-Century Collections Online

[1]The Alan Turing Institute, London, United Kingdom. [2]Queen Mary University of London, London, United Kingdom. [3]University of Bari, Bari, Italy. [4]University of Warwick, Coventry, United Kingdom. [5]University of Oxford, Oxford, United Kingdom. [6]King's College London, London, United Kingdom. ✉e-mail: barbara.mcgillivray@kcl.ac.uk

(ECCO-TCP corpus), covering the years 1700–1799[18]. These include embeddings trained on five twenty-year periods for 150 million words randomly sampled from the "Literature and Language" section of this corpus. Another set of diachronic word embeddings was released as part of a system for diachronic semantic search on text corpora based on the Google Books Ngram Corpus (English Fiction and German subcorpus), the Corpus of Historical American English, the Deutsches Textarchiv 'German Text Archive' (a corpus of ca. 1600–1900 German), and the Royal Society Corpus (containing the first two centuries of the Philosophical Transactions of the Royal Society of London)[19] (http://jeseme.org/help.html, last accessed 27/11/2020).

The only example of trained word diachronic embeddings covering a short and recent time period are available in[20] and were built following the methodology described in[21]. The authors trained monthly word embeddings from the tweets available via the Twitter Streaming API from 2012 to 2018 and comprising over 20 billion word tokens.

The word embeddings datasets surveyed in this section are useful resources for researchers conducting linguistic diachronic analyses or developing NLP tools that require data with a chronological depth. However, more steps are needed in order to process these embeddings further.

We present DUKweb, a rich dataset comprising diachronic embeddings, co-occurrence matrices, and time series data which can be directly used for a range of diachronic linguistic analysis aimed an investigating different aspects of recent language change in English. DUKweb was created from the JISC UK Web Domain Dataset (1996–2013), a very large archive which collects resources from the Internet Archive hosted on domains ending in '.uk'. DUKweb consists of three main components:

1. co-occurrences matrices for each year built by relying on the original text extracted from the JISC UK Web Domain Dataset;
2. a set of different word embeddings (Temporal Random Indexing and word2vec) for each year;
3. time series of words' change in representation across time.

DUKweb can be used for several time-independent NLP tasks, including word similarity, relatedness, analogy, but also for temporally dependent tasks, such as semantic change detection (i.e., tracking change in word meaning over time), a task which has received significant attention in recent years[22,23].

The main innovative features of our contribution are:

- Variety of object types: we release the full package of data needed for diachronic linguistic analyses of word meaning: co-occurrence matrices, word embeddings, and time series.
- Size: we release word vectors trained on a very large contemporary English diachronic corpus of 1,316 billion word occurrences spanning the years 1996–2013; the total size of the dataset is 330GB;
- Time dimension: the word vectors have been trained on yearly portions of the UK web archive corpus corpus, which makes them ideally suited for many diachronic linguistic analyses;
- Embedding types: we release count-based and prediction-based types of word embeddings: Temporal Random Indexing vectors and word2vec vectors, respectively, and provide the first systematic comparison between the two approaches;

None of the other existing datasets offer researchers all the above features. The surveyed datasets are based on corpora smaller than the UK Web archive JISC dataset, the biggest one being 850 billion words vs. 1316 billion words[11]. Moreover, only the Twitter embedding resource[20] was specifically built to model recent language change (2012–2018). On the other hand, recent workshops on semantic change detection[22,23] study the semantic change across the two distinct time periods and therefore lack the longitudinal aspect of our resources. In addition to being based on a much larger corpus with a longitudinal component, DUKweb can be readily used to study semantic change in English between 1996 and 2013 and therefore to investigate the effects of various phenomena such as the expansion of the World Wide Web or social media on the English language. Finally, DUKweb offers researchers a variety of object types and embedding types; as we show in our experiments, these capture the semantic change of different words and can therefore be leveraged in conjunction for further analysis in future work.

Methods

The flow chart in Fig. 1 illustrates the process from corpus creation to the construction of DUKweb. In the next sub-sections we provide details about how the three parts of the dataset were built.

Source data.

We used the JISC UK Web Domain Dataset (1996–2013)[24], which collects resources from the Internet Archive (IA) that were hosted on domains ending in '.uk', and those that are required in order to render '.uk' pages. The JISC dataset contains resources crawled by the IA Web Group for different archiving partners, the Web Wide crawls and other miscellaneous crawls run by IA, as well as data donations from Alexa (https://www.alexa.com) and other companies or institutions, therefore we do not have access to all the crawling configurations used by the different partners. The dataset contains not only HTML pages and textual resources, but also video, images and other types of files.

The JISC dataset is composed of two parts: the first part contains resources from 1996 to 2010 for a total size of 32TB; the second part contains resources from 2011–2013 for a total size of 30TB. The JISC dataset cannot be made generally available, but can be used to generate derived datasets (like DUKweb).

Text extraction and pre-processing.

The first step in the creation of DUKweb consisted in processing the JISC web archive in order to extract its textual resources. For this purpose, we extracted the text from resources

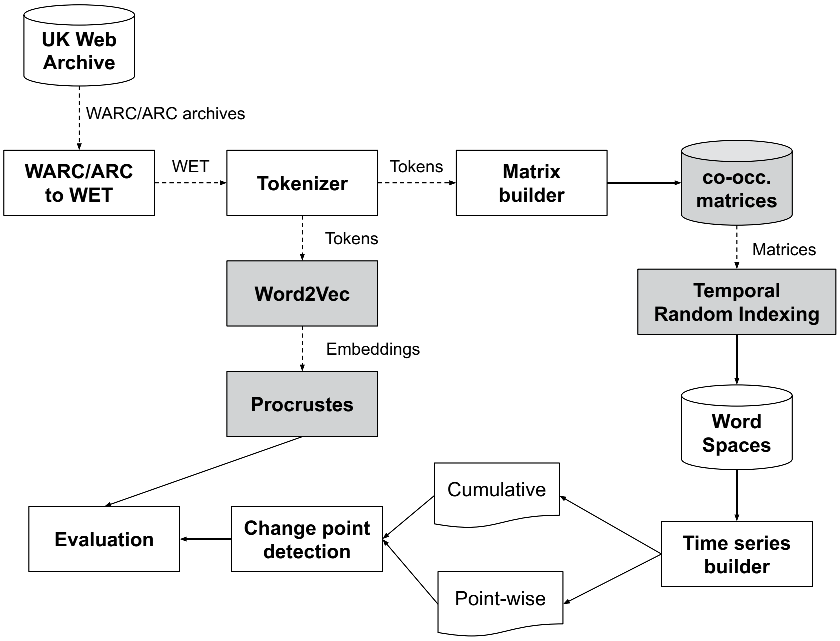

Fig. 1 Flowchart of the creation of DUKweb.

such as TXT files and parsed HTML pages. We used the jsoup library (https://jsoup.org/) for parsing HTML pages. The original JISC dataset contains files in the ARC and WARC formats, standard formats used by IA for storing data crawled from the web as sequences of content blocks. The WARC format (https://commoncrawl.org/2014/04/navigating-the-warc-file-format/, last accessed 27/11/2020) is an enhancement of the ARC format supporting a range of features including metadata and duplicate events. We converted the ARC and WARC archives into the WET format, a standard format for storing plain text extracted from ARC/WARC archives. The output of this process provided about 5.5TB of compressed WET archives.

The second step consisted in tokenizing the WET archives. For this purpose, we used the StandardAnalyzer (https://lucene.apache.org/core/7_3_1/core/index.html), last accessed 27/11/2020. provided by the Apache Lucene 4.10.4 API (https://lucene.apache.org/core/, last accessed 27/11/2020). This analyzer also provides a standard list of English stop words. The size of the tokenized corpus is approximately 3TB, with a vocabulary size of 29 million tokens and about 1200 billion occurrences. We did not apply any further text processing steps such us lemmatization or stemming because our aim was to build a language independent pipeline.

Co-occurrence matrices. The co-occurrence matrices of DUKweb store co-occurrence information for each word token in the JISC dataset processed as described in section ‘Text extraction and pre-processing’. For the construction of co-occurrence matrices, we focused on the 1,000,000 most frequent words.

In order to track temporal information, we built a co-occurrence matrix for each year from 1996 to 2013. Each matrix is stored in a compressed text format, with one row for each token, where each row contains the token and the list of tokens co-occurring with it. Following standard practice in NLP, we extracted co-occurrence counts by taking into account a window of five words to the left and five words to the right of the target word[25–27].

Word embeddings. We constructed semantic representations of the words occurring in the processed JISC dataset by training word embeddings for each year using two approaches: Temporal Random Indexing and the word2vec algorithm (i.e., skip-gram with negative sampling). The next subsections provide details of each approach.

Temporal random indexing. The first set of word embeddings of DUKweb was trained using Temporal Random Indexing (TRI)[28–30]. We further developed the TRI approach in three directions: 1) we improved the system to make it possible to process very large datasets like the JISC UK Web Domain Dataset; 2) we introduced a new way to weigh terms in order to reduce the impact of very frequent tokens; 3) compared to our previous work on the same topic[30], we proposed a new “cumulative” approach for building word vectors.

The idea behind TRI is to build different word spaces for each time period under investigation. The peculiarity of TRI is that word vectors over different time periods are directly comparable because they are built using the same random vectors. TRI works as follows:

1. Given a corpus C of documents and a vocabulary V of terms (V contains the terms that we want to analyse, typically, the top n frequent terms) extracted from C, the method assigns a random vector r_i to each term $t_i \in V$. A random vector is a vector that has values in the set $\{-1, 0, 1\}$ and is sparse, with few non-zero elements randomly distributed along its dimensions. The sets of random vectors assigned to all terms in V are near-orthogonal;

2. The corpus C is split into different time periods T_k using temporal information, for example the year of publication;
3. For each period T_k, a word space WS_k is built. Each of the terms of V occurring in T_k is represented by a semantic vector. The semantic vector sv_i^k for the i-th term in T_k is built as the sum of all the random vectors of the terms co-occurring with t_i in T_k. Unlike the approach proposed in[30], the sv_i^k is not initialized as a zero vector, but as the the semantic vectors sv_i^{k-1} built in the previous period. Using this approach we are able to collect semantic features of the term across time. If the sv_i^{k-1} is not available, (the sv_i^{k-1} is not available when the term T_i appears for the first time in T_k) the zero vector is used. When computing the sum, we apply some weighting to the random vector. To reduce the impact of very frequent terms, we use the weights $\sqrt{\frac{th \times C_k}{\#t_i^k}}$, where C_k is the total number of occurrences in T_k and $\#t_i^k$ is the number of occurrences of the term T_i in T_k. The parameter th is set to 0.001.

This way, the semantic vectors across all time periods are comparable since they are the sum of the same random vectors.

Time series. For each term t_i DUKweb also contains a time series $\Gamma(t_i)$, which can be used to track a word's meaning change over time. The time series are sequences of values, one value for each time period, and represent the semantic shift of that term in the given period. We adopt several strategies for building the time series. The baseline approach is based on term log-frequency, where each value in the series is defined as $\Gamma_k(t_i) = \log\left(\frac{\#t_i^k}{C_k}\right)$.

In addition to the baseline, we devised two other strategies for building the time series:

point-wise: $\Gamma_k(t_i)$ is defined as the cosine similarity between the semantic vector of t_i in the time period k, sv_i^k, and the semantic vector of t_i in the previous time period, sv_i^{k-1}. This way, we capture semantic change between two time periods;
cumulative: we build a cumulative vector $sv_i^{C_{k-1}} = \sum_{j=0}^{k-1} sv_i^j$ and compute the cosine similarity of this cumulative vector and the vector sv_i^k. The idea behind this approach is that the semantics of a word at point k-1 depends on the semantics of the word in all the previous time periods. The cumulative vector is the vector sum of all the previous word vectors[31].

Skip-gram with negative sampling. The second approach we followed for generating word representations is based on the of skip-gram with negative sampling (SGNS) algorithm[32]. The skip-gram model is a two-layer neural network that aims at predicting the words (context) surrounding a particular word in a sentence. The training process is performed on a large corpus, where samples of {context, word} pairs are drawn by sliding a window of N words at a time. The resulting word vectors can appropriately represent each word based on the context in which it appears so that the distance between similar words is small and the analogy of word pairs like (*king*, *queen*) and (*uncle*, *aunt*) is maintained. Over the past few years SGNS has been widely employed and its efficiency has been demonstrated in several studies on semantic change[10,22,27,33,34].

We split the pre-processed corpus into yearly bins, as in the case of TRI, and train one language model per year (we refrain from using the years 1996–1999 for SGNS, due to their considerably smaller size compared to the rest of the years in our processed collection, which could result into noisy word representations). Our skip-gram model then learns a single 100-dimensional representation for each word that is present at least 1,000 times on each year independently, i.e. 3,910,329 words. We used the implementation of skip-gram with negative sampling as provided in gensim (https://radimrehurek.com/gensim/), using 5 words as our window size and training for 5 epochs for each year while keeping the rest of hyperparameters on their default values. (In previous work[27], we selected the 47.8 K words occurring in all years in both our corpus and in the entry list of the Oxford English Dictionary. Importantly, very common words (e.g., "facebook") that appeared only after a certain point in time are not included in our previous analysis.)
Orthogonal procrustesIn contrast to TRI, a drawback of training independent (i.e., one per year) SGNS models is the fact that the resulting word vectors are not necessarily aligned to the same coordinate axes across different years[10]. In particular, SGNS models may result in arbitrary linear transformations, which do not affect pairwise cosine-similarities within-years, but prevent meaningful comparison across years.

To align the semantic spaces, we follow the Procrustes analysis from[10]. Denote by $W^{(t)} \in \mathrm{R}^{n \times m}$ the matrix of word embeddings in year t. The orthogonal Procrustes problem[35] consists of solving

$$\min_{Q} \left\| W^{(t)}Q - W^{(t+1)} \right\|_F, \quad \text{subject to } Q^T Q = I, \tag{1}$$

where $Q \in \mathbb{R}^{m \times m}$ is the rotation matrix and I is the $m \times m$ identity matrix. We align the word embedding in year t to their respective embeddings in year $t+1$ by finding a translation, rotation, and scaling of $W^{(t)}$ that minimizes its distance to $W^{(t+1)}$ as measured by the Frobenius norm. The optimisation problem in (1) can be solved using the singular value decomposition of $W^{(t)}(W^{(t+1)})^T$. We can then use the cosine distance between the vector representation of a word across aligned embeddings as a measure of the semantic displacement of that word.

An alternative approach would be to initialise the embeddings of the year $t+1$ with the resulting representations of the year t[15]. However, this would demand sequential – and thus longer – processing of our data and it is not clear whether the resulting representations capture the semantics of the words more effectively, as demonstrated in recent work on semantic change detection[21]. Another potential approach to consider stems from the

```
linux   swapping   4   google   173   xp   454   manufacturer
237   job   64   install   255   security   137   cgi   47
operating   705   host   69   performance   44   sharing
56...
```

Fig. 2 Example of co-occurrence matrix for the word *linux* (year 2011) in DUKweb.

Year	Vocabulary Size	#co-occurrences	File Size
1996	454,751	1,201,630,516	645.6MB
1997	711,007	17,244,958,174	2.7GB
1998	704,453	10,963,699,018	2.4GB
1999	769,824	32,760,590,881	3.6GB
2000	847,318	107,529,345,578	5.8GB
2001	911,499	197,833,301,500	9.2GB
2002	945,565	274,741,483,798	11GB
2003	992,192	539,189,466,798	14GB
2004	1,040,470	975,622,607,090	18.2GB
2005	1,060,117	793,029,668,228	16.9GB
2006	1,076,523	721,537,927,839	16.7GB
2007	1,093,980	834,261,488,677	18.1GB
2008	1,105,511	1,067,076,347,615	19.6GB
2009	1,105,901	481,567,239,481	14.15GB
2010	1,125,201	778,111,567,761	16.7GB
2011	1,145,990	1,092,441,542,978	18.9GB
2012	1,144,764	1,741,038,554,999	20.6GB
2013	1,044,436	393,672,000,378	8.9GB

Table 1. Statistics about the co-occurrences matrices in DUKweb. The first column shows the year, the second column contains the size of the vocabulary for that year in terms of number of word types, the third column contains the total number of co-occurrences of vocabulary terms for that year, and the last column shows the size (compressed) of the co-occurrence matrix file.

generalized orthogonal Procrustes problem which simultaneously considers all the available embeddings at once and aims to optimally identify an *average embedding* which is simultaneously close to all the input embeddings[36]. This contrasts with our approach where only pairwise Procrustes alignments are performed.

Time series. We construct the time series of a word's similarity with itself over time, by measuring the cosine distance of its aligned representation in a certain year (2001–2013) from its representation in the year 2000. Recent work[21] has demonstrated that the year used for reference matters (i.e., $W^{(t)}$ in Eq. 1). We intuitively opted to use the year 2000 as our referencing point, since it is the beginning of our time period under examination. In order to construct time series in a consistent manner, we only use the representations of the words that are present in each year.

Data Records

This section describes each data record associated with our dataset, which is available on the British Library repository (https://doi.org/10.23636/1209)[37].

Co-occurrences matrices. The first part of our dataset consists of the co-occurrences matrices. We built a co-occurrence matrix from the pre-processed JISC dataset for each year from 1996 to 2013. Each matrix is stored in a compressed text format, with one row per token. Each row reports a token and the list of tokens co-occurring with it. An example for the word *linux* is reported in Fig. 2, which shows that the token *swapping* co-occurs 4 times with *linux*, the word *google* 173 times, and so on.

Table 1 reports the size of the vocabulary and the associated compressed file size, for each year. The total number of tokens considering only the terms in our vocabulary is 1,316 billion.

Word embeddings. The second part of the dataset contains word embeddings built using TRI and word2vec. Both embeddings are provided in the GZIP compressed textual format, with one file for each year. Each file stores a word embedding for each line, the line starts with a word followed by the corresponding embedding vector entries separated by spaces, for example:

```
dog 0.0019510963-0.033144157 0.033734962...
```

	TRI		SGNS	
Year	Vocabulary Size	File Size	Vocabulary Size	File Size
1996	454,751	284.9MB	—	—
1997	711,007	904.8MB	—	—
1998	704,453	823.3MB	—	—
1999	769,824	1.1GB	—	—
2000	847,318	1.5GB	235,428	114.7MB
2001	911,499	1.9GB	407,074	198.2MB
2002	945,565	2.1GB	571,419	277.8MB
2003	992,192	2.4GB	884,393	430.4MB
2004	1,040,470	2.7GB	1,270,804	619.1MB
2005	1,060,117	2.7GB	1,202,899	585.4MB
2006	1,076,523	2.7GB	1,007,582	490.6MB
2007	1,093,980	2.8GB	1,124,179	548.2MB
2008	1,105,511	2.9GB	1,173,870	572.2MB
2009	1,105,901	2.6GB	671,940	327.6MB
2010	1,125,201	2.8GB	1,183,907	576.8MB
2011	1,145,990	2.9GB	1,309,804	637.7MB
2012	1,144,764	3.0GB	1,607,272	784.0MB
2013	1,044,436	1.9GB	587,035	285.6MB

Table 2. Statistics about the vocabulary in terms of overall number of words and (compressed) file size, per year and per method (TRI, SGNS).

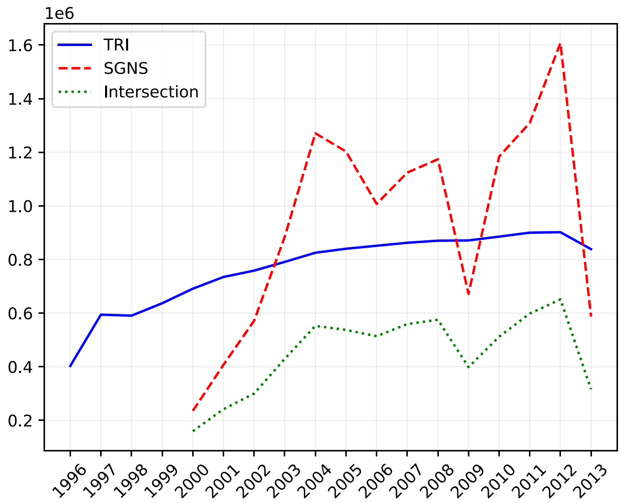

Fig. 3 Number of words included in the TRI and SGNS representations contained in DUKweb, along with the size of their intersected vocabulary size, per year.

Table 2 shows the (compressed) file size of each vector space. For TRI, the number of vectors (terms) for each year is equal to the vocabulary size of co-occurrences matrices as reported in Table 1. The TRI vector dimension is equal to 1,000, while the number of no-zero elements in random vector is set to 10. Finally, the parameter *th* is set to 0.001, and TRI vectors are built by using the code reported in section 'Code Availability'. For SGNS, the total number of words represented in any year is 3,910,329. Finally, the chart in Fig. 3 shows the intersected vocabulary between the two methods. The number of total terms contained in the intersection is 47,871. We also release a version of TRI embeddings built by taking into account only the terms contained in the intersection. In order to perform a fair evaluation, in our experiments we only take into account the intersected vocabulary.

Time series. The last part of the dataset contains a set of time series in CSV format computed using the different strategies described in the previous sections. For each time series we extract a list of change points. The chart on Fig. 4(a) shows the time series for different words that have acquired a new meaning after the year 2000, according to the Oxford English Dictionary. Similarly, Fig. 4(b) shows the time series of the cosine distances of the same words that result after Orthogonal Procrustes is applied on SGNS as described in Section 'Skip-gram with Negative Sampling'. In particular, we first align the embeddings matrices of a year *T* with respect to the year

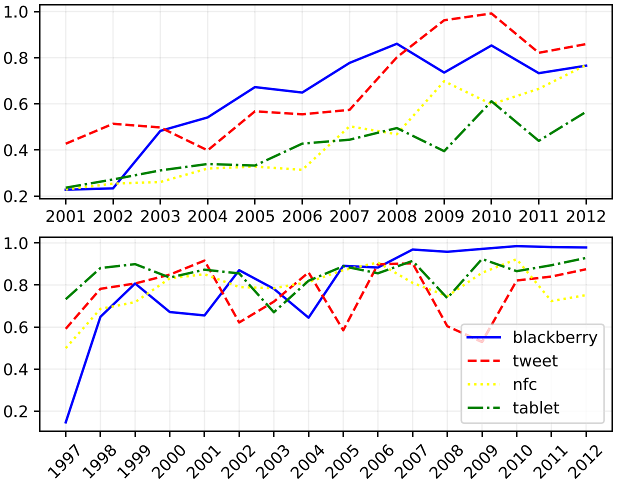

Fig. 4 Time series based on TRI (below) and SGNS (above) of four words whose semantics have changed between 2001–2013 according to the Oxford English Dictionary (i.e., they have acquired a new meaning during this time period).

2000 and then we measure the cosine distance of the aligned word vectors. As a result, this part of our dataset consists of 168,362 tokens and their respective differences from the year 2000 during the years 2001–2013. We further accompany these with the time series consisting of the number of occurrences of each word in every year.

Figure 4 shows that the semantic similarity of the word "blackberry" decreases dramatically in 2004, which corresponds to the change point year detected by the semantic change detection algorithm. On the other hand, Fig. 4(b) shows that the four words are moving away with time from their semantic representation in the year 2000. We also find examples of cases of semantic shifts corresponding to real-world events in SGNS representations: for example, the meaning of the word "tweet" shifted more rapidly after Twitter's foundation (2006), whereas the lexical semantics of the word "tablet" mostly shifted in the year 2010, at the time when the first mass-market tablet – iPad – was released.

Furthermore, as we showed in our previous work[30], it is possible to analyze the neighborhood of a word (e.g. "blackberry") in different years (e.g. in 2003 and 2004) in order to understand the semantic change it underwent. The list of neighborhoods can be computed as the most similar vectors in a given word space. Similarly, in the case of SGNS we can measure the cosine similarity between the word representations in different years.

Dataset summary. Overall, our dataset consists of the following files:

- **D-**YEAR**_merge_occ.gz**: co-occurrences matrix for each year.
- YEAR**.csv.zip**: The SGNS word embeddings during the year YEAR, as described in section 'Skip-gram with Negative Sampling'. There are 14 files, one for each year between 2000 and 2013, and the size of each file is shown in Table 2.
- **D-**YEAR**_merge.vectors.txt.gz**: TRI embeddings for each year in textual format.
- **D-**YEAR**_merge.vectors**: TRI embeddings for each year in binary format. This vectors can be directly used with the TRI tool. (https://github.com/pippokill/tri)
- **timeseries**: four files containing the timeseries for the words based on (a) their counts, as extracted from SGNS (count_timeseries.tsv), (b) the cosine distances from their representations in the year 2000 based on SGNS (w2v_dist_timeseries.tsv) and (c) the time series generated via TRI-based pointwise and cumulative approaches (ukwac_all_point.csv, ukwac_all_cum.csv respectively).
- **vectors.elemental**: TRI embeddings for the whole vocabulary in binary format.

Technical Validation

We perform two sets of experiments in order to assess the quality of the embeddings generated via TRI and SGNS. In the first set, our goal is to measure the embeddings' ability to capture semantic properties of words, i.e. analogies, similarities and relatedness levels. In the second set, we aim at exploring to what extent the two types of contextual vectors capture the change in meaning of words.

Static Tasks: Word-level Semantics. In this set of tasks we examine the ability of the word vectors to capture the semantics of the words associated to them. We work on three different sub-tasks: (a) word analogy, (b) word similarity and (c) word relatedness detection.

Tasks description. Word analogy. *Word analogy detection* is the task of identifying relationships between words in the form of "w_a is to w_b as w_c is to w_d", where w_i is a word. In our experiments, we make use of the widely employed dataset which was created by Mikolov and colleagues[32] and which contains these relationships in different categories (we list four categories, with one example for each of them):

- Geography, e.g. *Paris is to France as Madrid is to Spain*.

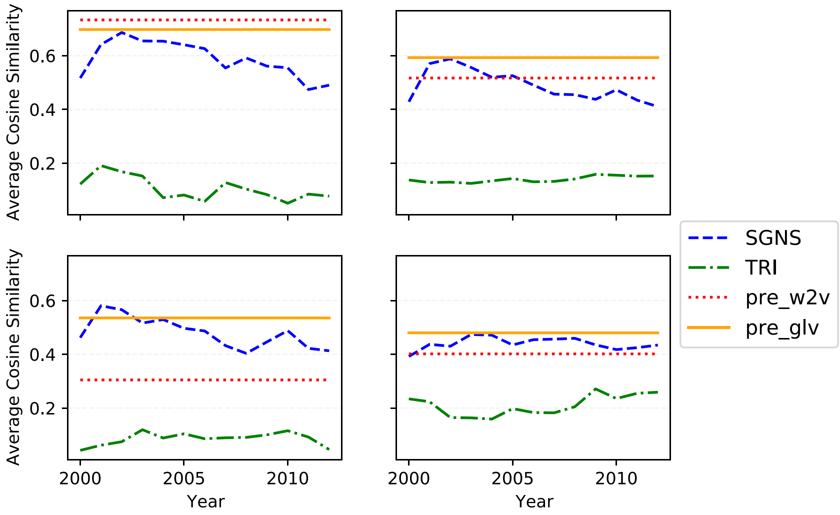

Fig. 5 Results of the SGNS and TRI embeddings and the two baselines models on the Word Analogy task for the four categories: Family, Grammar, Geography and Currency.

- Currencies, e.g. *Peso is to Argentina as euro is to Europe*.
- Family, e.g. *Boy is to girl as policeman is to policewoman*.
- Grammar rules, e.g. *Amazing is to amazingly as apparent is to apparently*.

Given the vectors $[v_a, v_b, v_c, v_d]$ of the words $[w_a, w_b, w_c, w_d]$ respectively, and assuming an analogous relationship between the pairs $[w_a, w_b]$ and $[w_c, w_d]$ in the form described above, previous work has demonstrated that in SGNS-based embeddings, $v_a - v_b \approx v_c - v_d$. We perform this task by measuring the cosine similarity s_c that results when comparing the vector v_c of the word w_c against the vector resulting from $v_a - v_b + v_d$. We apply this method to all the words in the examples of the word analogy task set and we average across all examples. Higher average cosine similarity scores indicate a better model.

TRI-based embeddings are not suitable for this kind of task due to the nature of the vector space. TRI is not able to learn the linear dependency between vectors in the space, which may be due to the Random Projection that preserves all distance/similarity measures based on L2-norm, but it distorts the original space and does not preserve the original position of vectors. We try to simulate analogy by using vector orthogonalization as the negation operator[38]. In particular, the vector sum $v_b + v_d$ is orthogonalized with respect to the vector v_a. For each word vector the cosine similarity is computed against the vector obtained as the result of the orthogonalization, then the word with the highest similarity is selected.

We perform experiments using our TRI and SNGS word vectors. For comparison purposes, we also employ the word2vec pre-trained embeddings generated in[32] as well as the GloVe embeddings released in[39]. These are well-established sources of word vectors that have been trained on massive corpora of textual data and have been employed in an extremely large body of research work across multiple NLP tasks. In particular, for word2vec we employ the 300-dimensional word vectors that were trained on Google news (pre_{w2v})[32], whereas for GloVe we use the 100-dimensional vectors trained on Twitter (pre_{glv})[39]. As opposed to SGNS and TRI, pre_{w2v} and pre_{glv} are temporally independent (i.e., there is one representation of each word through time). To allow for a fair comparison, the analysis described here and in the following sections is on the intersected vocabulary across all years, i.e. on the set of words occurring in all years in our corpus.

Word similarity. In the *word similarity* task, we are interested in detecting the similarity between two words. We employ the dataset in[40], which contains examples of 203 word pairs, along with their similarity score, as provided by two annotators. We only use the 187 word pairs that consist of words present in the intersected vocabulary, as before. We deploy the following experimental setting: given the word vectors w_a and w_b of a pair of words, we measure their cosine similarity $sim(w_a, w_b)$; subsequently, we measure the correlation between $sim(w_a, w_b)$ and the ground truth. Higher values indicate a better model. We compare performance against the baselines pre_{w2v} and pre_{glv}.

Word relatedness. Our final experiment in this section involves the task of identifying the level of relatedness between two words. For example, the word "computer" is much more closely related to the word "software" than to the word "news". We employ the word relatedness dataset introduced by[40], which contains 252 examples of word pairs, along with their associated relatedness score. As in the previous tasks, we only use the 232 examples that are present in our intersected vocabulary; we deploy the same experimental setting as in the case of the Word Similarity task and compare against the two previously introduced baselines (pre_{w2v}, pre_{glv}).

Results

Word analogy. The results are displayed in Fig. 5. pre_{glv} outperforms our models in almost all cases. *SGNS* achieves comparable performance to pre_{glv} and pre_{w2v}, especially during the first years; comparably. *TRI* performs poorly since it is not able to learn the linear dependency between vectors in the space, which may be due to the *Random Projection* that preserves all distance/similarity measures based on L2-norm, but it distorts the original

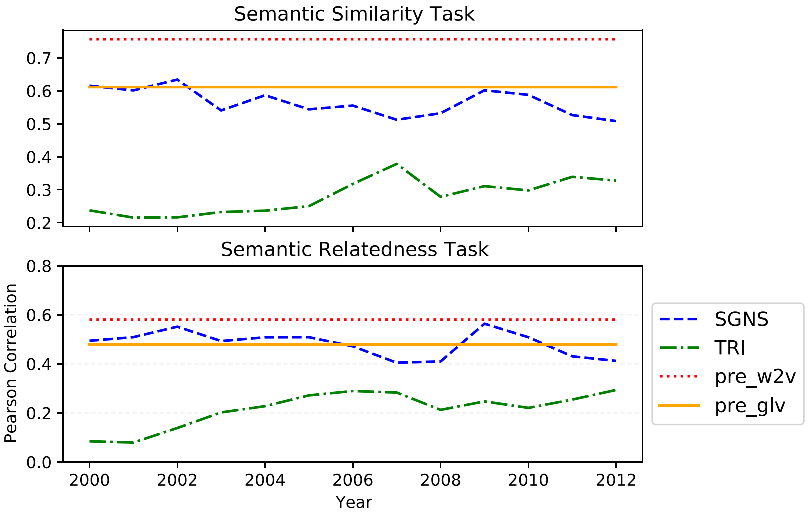

Fig. 6 Results of the SGNS and TRI embeddings and the two baselines models on the Word Similarity and Word Relatedness tasks.

space and does not preserve the original position of vectors. pre_{w2v}, pre_{glv} and *SGNS* better capture relationships related to "Family" and "Grammar" than "Currency" in our experiments.

Word relatedness and word similarity. Figure 6 shows the results on the word similarity and word relatedness tasks. Here pre_{w2v} achieves the highest average Pearson correlation score across all years (0.76 and 0.58 for the two tasks, respectively). *SGNS* performs competitively (0.57 vs 0.61, on average) and outperforms in most cases the pre_{glv} baseline for the case of semantic similarity and relatedness, respectively. Importantly, its performance is again consistent across time, ranging from 0.51 to 0.63 and from 0.40 to 0.56 for the two respective tasks. TRI performs again more poorly and slightly more inconsistently than *SGNS*, with its (across years) evaluation score ranging from 0.22 to 0.38 (average: 0.28) and from 0.08 to 0.29 (average: 0.22).

The results presented so far show that, overall, the *SGNS* and *TRI* embeddings do not outperform the two baselines (pre_{w2v} and pre_{glv}) consisting of static word representations in temporally independent tasks. This is partially attributed to the facts that our resources are built on large-scale, yet potentially noisy content and are also restricted primarily to British English, which could impose some geographical and linguistic biases. However, both representations (*SGNS*, *TRI*) can be effectively utilised for a dynamic, i.e. temporally dependent, task which cannot be dealt with static word representations, as we show in the next section.

Dynamic task: Semantic change detection. The major contribution of the *TRI* and *SGNS* embeddings of DUKweb consists in their temporal dimension. To exploit this aspect, we measure their ability to capture lexical semantic change over time.

Experimental setting. We use as ground truth 65 words that have changed their meaning between 2001 and 2013 according to the Oxford English Dictionary[27]. We define the task in a time sensitive manner: given the word representations in the year 2000 and in the year *X*, our aim is to find the words whose lexical semantics have changed the most. We vary *X* from 2001 to 2013, so that we get clearer insights on the effectiveness of the word representations. Since our ground truth consists of words that are present in the OED, we also limit our analysis to the 47,834 words that are present both in the OED and in the TRI/SGNS vocabularies.

Models. We employ four variants of Orthogonal Procrustes-based methods operating on *SGNS* from our prior work[27,41] and two models operating on *TRI*, as follows:

- $SGNS_{pr}$ employs the Orthogonal Procrustes (OP) alignment to align the word representations in the year *X* based on those for the year 2000;
- $SGNS_{pr(a)}$ applies OP in two passes: during the first pass, it selects the most stable (*"anchor"*) words (i.e., those whose aligned representations across years have the highest cosine similarity); then, it learns the alignment between the representations in the years 2000 and *X* using solely the anchor words;
- $SGNS_{pr(d)}$ applies OP in several passes executed in two rounds: during the first round, it selects the most stable (*"diachronic anchor"*) words across the full time interval; then, it learns the alignment between the representations in the years 2000 and *X* using solely the diahcronic anchor words;
- TRI_c and TRI_p exploit time-series built by the cumulative and point-wise approach, respectively (see Section 'Temporal Random Indexing').

Evaluation. In all of the *SGNS* models, we rank the words on the basis of the cosine similarity of their representations in the aligned space, such that words in the higher ranks are those whose lexical semantics has

SGNS$_{pr}$	**SGNS$_{pr(a)}$**	**SGNS$_{pr(d)}$**	**TRI$_c$**	**TRI$_p$**
cloud	sars	eris	tweet	root
sars	fap	ds	qe	purple
tweet	trending	follow	parmesan	blackberry
trending	eris	blw	event	tweet
fap	tweet	fap	sup	follow
tweeter	preloading	unlike	status	eta
like	chugging	chugging	grime	prep
preloading	bloatware	roasting	prep	grime
bloatware	tweeter	even	trending	status
parmesan	parmesan	parmesan	tomahawk	tomahawk

Table 3. Examples of easy-to-predict (top-5) and hard-to-predict (bottom-5) words by our SNGS and TRI models.

Model	**2001**	**2002**	**2003**	**2004**	**2005**	**2006**	**2007**	**2008**	**2009**	**2010**	**2011**	**2012**	**2013**
SGNS$_{pr}$	36.33	34.03	30.90	26.85	29.29	27.16	26.88	28.60	**25.16**	27.21	25.69	**25.83**	29.13
SGNS$_{pr(a)}$	36.83	32.30	31.67	27.23	27.27	**26.31**	**25.25**	28.15	26.54	**27.14**	27.59	30.42	28.78
SGNS$_{pr(d)}$	**33.09**	**28.32**	**28.91**	**23.17**	**25.13**	27.99	30.38	**25.60**	27.17	28.96	**25.08**	27.84	**24.83**
TRI$_c$	54.65	51.22	56.64	55.63	50.90	55.98	60.96	58.03	58.94	56.59	59.00	45.96	45.72
TRI$_p$	56.77	59.79	54.81	54.61	53.22	54.39	55.44	55.12	59.76	59.22	53.22	46.30	50.07

Table 4. Average rank of a semantically shifted word; lower scores indicate a better model.

Model	**2001**	**2002**	**2003**	**2004**	**2005**	**2006**	**2007**	**2008**	**2009**	**2010**	**2011**	**2012**	**2013**
SGNS$_{pr}$	**16.92**	23.08	21.54	26.15	**35.38**	30.77	**29.23**	32.31	**38.46**	29.23	32.31	**36.92**	32.31
SGNS$_{pr(a)}$	15.38	**24.62**	23.08	23.08	30.77	**35.38**	**29.23**	32.31	33.85	**32.31**	33.85	26.15	32.31
SGNS$_{pr(d)}$	15.38	18.46	**27.69**	**29.23**	**35.38**	23.08	27.69	**36.92**	30.77	23.08	**35.38**	18.46	**33.85**
TRI$_c$	7.69	12.31	6.15	3.08	12.31	7.69	7.69	7.69	6.15	7.69	4.62	13.85	12.31
TRI$_p$	12.31	4.62	7.69	10.77	10.77	4.62	13.85	7.69	3.08	4.62	12.31	21.54	13.85

Table 5. Recall at 10% of our different SGNS and TRI models.

changed the most. For the *TRI*-based models, the Mean Shift algorithm[42] is used for detecting change points in the time series consisting of the cosine similarity scores between the representations of the same word in each year covered by the corpus. For each detected change point, a p-value is estimated according to the confidence level obtained by the bootstrapping[43] approach proposed in[42], then words are ranked according to the p-value in ascending order. Finally, we run the evaluation using the recall at 10% of the size of the dataset as well as the average rank (μ_r, scaled in [0, 1]) of the 65 words with altered semantics. Higher recall-at-k and lower μ_r scores indicate a better performance.

Results. Tables 4 and 5 present the results of our models in terms of μ_r and recall-at-k, respectively. In both cases, the *SGNS*-based approaches perform better than *TRI*: on average, the best-performing SGNS-based model achieves 27.42 in μ_r (SGNS$_{pr(a)}$) and 29.59 in recall-at-k (*SGNS$_{pr}$*). The difference compared to TRI is attributed to their ability to better capture the contextualised representation of the words in our corpus. Nevertheless, TRI has recently achieved state-of-the-art performance on semantic change detection in the Swedish language[22]. Furthermore, despite their superior performance in this task, the Procrustes- and SGNS-based approaches have the shortcoming that they operate on a common vocabulary across different years; thus, words that have appeared at a certain year cannot be detected in these variants – a drawback is not present in the case of TRI.

Finally, we inspect the semantically altered words that have been detected by each model. Table 3 displays the most "obvious" and challenging examples of semantic change, as ranked on a per-model basis. It becomes evident that the two different word representations better capture the changes of different words. This is attributed to the different nature of the two released word representations. Incorporating hybrid approaches operating on multiple embedding models could be an important direction for future work in this task.

Usage Notes

The DUKweb datasets can be used for various time-independent tasks, as demonstrated in this article. Their major application is for studying how word meaning changes over time (i.e., *semantic change*) in a computational and linguistic context. The instructions on how to run our code for the experiments as well as for further downstream tasks have been made publicly available (see Code Availability). Access to the original JISC UK

Web Domain Dataset (1996–2013) collection[24] may be sought by contacting the UK Web Archive (https://doi.org/10.5259/ukwa.ds.2/1).

Code availability

The creation of the described datasets requires several steps, each step is performed by a different software. All the software is freely available, in particular:

- the code for the processing of the JISC UK Web Domain Dataset for producing both the WET and tokenized files: https://github.com/alan-turing-institute/UKWebArchive_semantic_change;
- the software for building both co-occurrences matrices and TRI: https://github.com/alan-turing-institute/temporal-random-indexing;
- the code for the experiments can be found at https://github.com/alan-turing-institute/DUKweb;
 For information on our input data, refer to: https://data.webarchive.org.uk/opendata/ukwa.ds.2/.

Received: 15 February 2021; Accepted: 27 August 2021;
Published online: 15 October 2021

References

1. Mikolov, T., Chen, K., Corrado, G. & Dean, J. Efficient estimation of word representations in vector space. *Proceedings of Workshop at the International Conference on Learning Representations* (2013).
2. Zhang, Y. *et al*. BioWordVec, improving biomedical word embeddings with subword information and MeSH. *Scientific Data* **6**(52) (2019).
3. Chinazzi, M., Gonçalves, B. Zhang, Q. & Vespignani, A. Mapping the physics research space: A machine learning approach. *EPJ Data Science* **8** (2019).
4. Lenci, A. Distributional semantics in linguistic and cognitive research. *Italian journal of linguistics* **20**(1), 1–31 (2008).
5. Firth, J.R. *Papers in Linguistics 1934–1951* (Oxford University Press, 1957)
6. Bojanowski, P., Grave, E., Joulin, A. & Mikolov, T. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* **5**, 135–146 (2017).
7. Cieliebak, M., Deriu, J., Uzdilli, F. & Egger, D. A Twitter Corpus and Benchmark Resources for German Sentiment Analysis. *Proceedings of the 4th International Workshop on Natural Language Processing for Social Media*, 45–51 (2017).
8. Kutuzov, A., Øvrelid, L., Szymanski, T. & Velldal, E. Diachronic word embeddings and semantic shifts: A survey. *Proceedings of the 27th International Conference on Computational Linguistics*, 1384–1397 (2018).
9. Tahmasebi, N., Borin, L. & Jatowt, A. Survey of computational approaches to lexical semantic change. *Computational approaches to semantic change* **6**, 1 (2021).
10. Hamilton, W.L., Leskovec, J. & Jurafsky, D. Diachronic word embeddings reveal statistical laws of semantic change. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 1489–1501 (2016).
11. Lin, Y., Michel, J.-B., Aiden Lieberman, E., Orwant, J., Brockman, W. & Petrov, S. Syntactic annotations for the Google Books Ngram corpus. *Proceedings of ACL, System Demonstrations*, 169–174 (2012).
12. Hamilton, W.L., Leskovec, J. & Jurafsky, D. *HistWords: Word Embeddings for Historical Text* https://nlp.stanford.edu/projects/histwords/ (2016).
13. Garg, N., Schiebinger, L., Jurafsky, D. & Zou, J. Word Embeddings Quantify 100 Years of Gender and Ethnic Stereotypes. *PNAS* **115**(16) (2017).
14. Hamilton, W.L., Leskovec, J. & Jurafsky, D. Cultural shift or linguistic drift? Comparing two computational measures of semantic change. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2116–2121 (2016).
15. Kim, Y., Chiu, Y.I., Hanaki, K., Hegde, D. & Petrov, S. Temporal Analysis of Language through Neural Language Models. *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, pp. 61–65 (2014).
16. *Google Books Ngram Viewer* http://storage.googleapis.com/books/ngrams/books/datasetsv2.html (2010).
17. Grayson, S., Mulvany, M., Wade, K., Meaney, G. & Greene, D. Novel2Vec: Characterising 19th Century Fiction via Word Embeddings. *24th Irish Conference on Artificial Intelligence and Cognitive Science*, 20–21 (2016).
18. Heuser, R. Word2Vec Models for Twenty-year Periods of 18 C (ECCO, "Literature and Language"). *Internet Archive* https://archive.org/details/word-vectors-18c-word2vec-models-across-20-year-periods (2016)
19. Hellrich, J. & Hahn, U. Exploring Diachronic Lexical Semantics with JeSemE. *Proceedings of ACL 2017, System Demonstrations*, pp. 31–36 (2017).
20. Shoemark, P., Ferdousi, L. F., Nguyen, D., Scott, H. & McGillivray, B. Monthly word embeddings for Twitter random sample (English, 2012–2018). *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Zenodo* https://doi.org/10.5281/zenodo.3527983 (2019).
21. Shoemark, P., Ferdousi, L.F., Nguyen, D., Scott, H. & McGillivray, B. Room to Glo: A systematic comparison of semantic change detection approaches with word embeddings. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 66–76 (2019).
22. Schlechtweg, D., McGillivray, B., Hengchen, S., Dubossarsky, H. & Tahmasebi, N. SemEval-2020 Task 1: Unsupervised Lexical Semantic Change Detection. *Proceedings of the 14th International Workshop on Semantic Evaluation*, 1–23 (2020).
23. Basile, P., Caputo, A., Caselli, T., Cassotti, P. & Varvara, R. Overview of the EVALITA 2020 Diachronic Lexical Semantics (DIACR-Ita) Task. *Proceedings of the 7th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA)* (2020).
24. JISC, the Internet Archive: JISC UK web domain dataset (1996–2013). https://doi.org/10.5259/ukwa.ds.2/1 (2013).
25. Levy, O. & Goldberg, Y. Dependency-based word embeddings. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 302–308 (2014).
26. Zhao, Z., Liu, T., Li, S., Li, B. & Du, X. Ngram2vec: Learning improved word representations from ngram co-occurrence statistics. *Proceedings of the 2017 conference on empirical methods in natural language processing*, 244–253 (2017).
27. Tsakalidis, A., Bazzi, M., Cucuringu, M., Basile, P. & McGillivray, B. Mining the UK Web Archive for Semantic Change Detection. *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP)*, 1212–1221 (2019).
28. Basile, P., Caputo, A. & Semeraro, G. Analysing word meaning over time by exploiting temporal random indexing. First Italian Conference on Computational Linguistics (Pisa University Press, 2014).
29. Caputo, A., Basile, P. & Semeraro, G. Temporal random indexing: A system for analysing word meaning over time. *Italian Journal of Computational Linguistics* **1**(1), 55–68 (2015).
30. Basile, P. & McGillivray, B. Exploiting the Web for Semantic Change Detection. *International Conference on Discovery Science*, 194–208 (Springer-Verlag, 2018).
31. Mitchell, J. & Lapata, M. Composition in distributional models of semantics. *Cognitive Science* **34**(8), 1388–1429 (2010).

32. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. & Dean, J. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing systems* 26, 3111–3119 (2013).
33. Kulkarni, V., Al-Rfou, R., Perozzi, B. & Skiena, S. Statistically significant detection of linguistic change. *Proceedings of the 24th International Conference on World Wide Web*, 625–635 (2015).
34. Schlechtweg, D., Hatty, A., del Tredici, M. & Schulte im Walde, S. A Wind of Change: Detecting and Evaluating Lexical Semantic Change across Times and Domains. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 732–746 (2019).
35. Schönemann, P. H. A generalized solution of the orthogonal procrustes problem. *Psychometrika* **31**(1), 1–10 (1966).
36. Pumir, T., Singer, A. & Boumal, N. The generalized orthogonal Procrustes problem in the high noise regime. *Information and Inference: A Journal of the IMA* **10**(3), 921–954 (2021).
37. Basile, P. & Tsakalidis, A. DUKweb (Diachronic UK web). *British Library* https://doi.org/10.23636/1209 (2020).
38. Widdows, D. Orthogonal negation in vector spaces for modelling word-meanings and document retrieval. *Proceedings of the 41st annual meeting of the Association for Computational Linguistics*, 136–143 (2003).
39. Pennington, J., Socher, R. & Manning, C. D. Glove: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543 (2014).
40. Agirre, E., *et al.* A Study on Similarity and Relatedness Using Distributional and WordNet-based Approaches. *Proceedings of the International Conference on North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, 19–27 (2009).
41. Tsakalidis, A. & Liakata, M.: Sequential Modelling of the Evolution of Word Representations for Semantic Change Detection. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 8485–8497 (2020).
42. Taylor, W.A. Change-point analysis: A powerful new tool for detecting changes. *Taylor Enterprises, Inc.* (2000).
43. Efron, B. & Tibshirani, R. J. *An introduction to the bootstrap* (CRC press, 1994).

Acknowledgements

This work was supported by The Alan Turing Institute under the EPSRC grant EP/N510129/1 and the seed funding grant SF099.

Author contributions

A.T. conducted the experiments for the SGNS embeddings and wrote sections Skip-gram with Negative Sampling, Word embeddings, Time series and Technical Validation. P.B. conducted the experiments for the TRI embeddings and contributed to sections Background & Summary, Source Data, Text extraction and pre-processing, Co-occurrence matrices, Temporal Random Indexing, Data Records/Co-occurrences matrices, Time series, Dataset Summary. M.B. co-supervised this work, contributed to the design of the experiments and contributed to subsection Skip-gram with Negative Sampling (Orthogonal Procrustes). M.C. co-supervised this work, contributed to the design of the experiments and contributed to subsection Skip-gram with Negative Sampling (Orthogonal Procrustes). B.M.c.G. was the main supervisor of this work, contributed to the design of the experiments and wrote section Background & Summary. All authors reviewed and approved the final manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to B.M.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.