Scalable and Interpretable Semantic Change Detection

Syrielle Montariol[*]
LISN - CNRS, Univ. Paris-Saclay
Societé Générale
syrielle.montariol@limsi.fr

Matej Martinc[*]
Jozef Stefan Institute
matej.martinc@ijs.si

Lidia Pivovarova
University of Helsinki
lidia.pivovarova@helsinki.fi

Abstract

Several cluster-based methods for semantic change detection with contextual embeddings emerged recently. They allow a fine-grained analysis of word use change by aggregating embeddings into clusters that reflect the different usages of the word. However, these methods are unscalable in terms of memory consumption and computation time. Therefore, they require a limited set of target words to be picked in advance. This drastically limits the usability of these methods in open exploratory tasks, where each word from the vocabulary can be considered as a potential target. We propose a novel scalable method for word usage-change detection that offers large gains in processing time and significant memory savings while offering the same interpretability and better performance than unscalable methods. We demonstrate the applicability of the proposed method by analysing a large corpus of news articles about COVID-19.

1 Introduction

Studying language evolution is important for many applications, since it can reflect changes in the political and social sphere. In the literature, the study of language evolution either focuses on long-term changes in the meaning of a word, or on more common short-term evolutionary phenomena, such as the word suddenly appearing in a new context, while keeping its meaning unchanged in a lexicographic sense. We refer to all types of language evolution—short- or long-term, with or without meaning change—as word usage change, a broad category that includes semantic change, but also any shifts in the context in which a word appears.

Recent studies (Giulianelli et al., 2020; Martinc et al., 2020a) show that clustering of contextual embeddings could be a proxy for word usage change: if clusters, which in theory capture distinct word usages, are distributed differently across time periods, it indicates a possible change in word's context or even loss or gain of a word sense. Thus, the cluster-based approach offers a more intuitive interpretation of word usage change than alternative methods, which look at the neighborhood of a word in each time period to interpret the change (Gonen et al., 2020; Martinc et al., 2020b) and ignore the fact that a word can have more than one meaning. The main limitation of the cluster-based methods is the scalability in terms of memory consumption and time: clustering is applied to each word in the corpus separately and all occurrences of a word need to be aggregated into clusters. For large corpora with large vocabularies, where some words can appear millions of times, the use of these methods is severely limited.

To avoid the scalability issue, cluster-based methods are generally applied to a small set of less than a hundred manually pre-selected words (Giulianelli et al., 2020; Martinc et al., 2020a). This drastically limits the application of the methods in scenarios such as identification of the most changed words in a large corpus or measuring of usage change of extremely frequent words, since clustering of all of word's contextual embeddings requires large computational resources. One way to solve the scalability problem using contextual embeddings is to *average* a set of contextual representations for each word into a single static representation (Martinc et al., 2020b). Averaging, while scalable, loses a lot on the interpretability aspect, since word usages are merged into a single representation.

The method we propose in this paper tackles scalability and interpretability at the same time. The main contributions of the paper are the following:

- A *scalable* method for contextual embeddings clustering that generates interpretable representations and outperforms other cluster-based methods.
- A method of measuring word usage change between periods with the *Wasserstein distance*. As far as we are aware, this is the first paper leverag-

[*] These authors contributed equally.

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4642–4652
June 6–11, 2021. ©2021 Association for Computational Linguistics

ing optimal transport for lexical semantic change detection.
- A *cluster filtering* step, which balances the deficiencies of clustering algorithms and consistently improves performance.
- An *interpretation pipeline* that automatically labels word senses, allowing a domain expert to find the most changing concepts and to understand *how* those changes happened.

The practical abilities of our method are demonstrated on a large corpus of news articles related to COVID-19, the Aylien Coronavirus News Dataset[1]. We compute the degree of usage change of almost 8,000 words, i.e., all words that appear more than 50 times in every time slice of the corpus, in the collection of about half a million articles in order to find the most changing words and interpret their drift[2].

2 Related Work

Diachronic word embedding models have undergone a surge of interest in the last two years with the successive publications of three articles dedicated to a literature review of the domain (Kutuzov et al., 2018; Tahmasebi et al., 2018; Tang, 2018). Most approaches build static embedding models for each time slice of the corpus and then make these representations comparable by either employing *incremental updating* (Kim et al., 2014) or *vector space alignment* (Hamilton et al., 2016b). The alignment method has proved superior on a set of synthetic semantic drifts (Shoemark et al., 2019) and has been extensively used (Hamilton et al., 2016b; Dubossarsky et al., 2017) and improved (Dubossarsky et al., 2019) in the literature. The recent SemEval Task on Unsupervised lexical semantic change detection has shown that this method is most stable and yields the best averaged performance across four SemEval corpora (Schlechtweg et al., 2020).

Yet another approach (Hamilton et al., 2016a; Yin et al., 2018) is based on comparison of neighbors of a target word in different time periods. This approach has been recently used to tackle the scalability problem (Gonen et al., 2020).

In all these methods, each word has only one representation within a time slice, which limits the sensitivity and interpretability of these techniques.

The recent rise of contextual embeddings such as BERT (Devlin et al., 2019) and ELMO (Peters et al., 2018) introduced significant changes to word representations. Contextual embeddings can be used for usage change detection by aggregating the information from the set of token embeddings. This can be done either through averaging of all vectors within a time slice and then computing averaged vector similarity (Martinc et al., 2020b), by computing a pairwise distance between vectors from different time slices (Kutuzov and Giulianelli, 2020), or by clustering all token representations to approximate its set of senses (Giulianelli et al., 2020). The analysis in this paper derives from this last set of methods, which demonstrate a higher performance than static embeddings methods at least on some datasets (Martinc et al., 2020a).

Automatic semantic shift detection has been used for text stream monitoring tasks, such as event detection (Kutuzov et al., 2017) viewpoint analysis (Azarbonyad et al., 2017) or monitoring of rapid discourse changes during crisis events (Stewart et al., 2017). None of these applications use clustering techniques and, as far as we are aware, only Martinc et al. (2020b) uses contextual embeddings for news stream analysis. In this paper we demonstrate the large potential of contextual embeddings for the *interpretable* tracking of short-term changes in word usage, which has a practical application for crisis-related news monitoring.

3 Scalability and Interpretability Limitations of Previous Methods

The main motivation for this research are the scalability or interpretability issues of previous methods for word usage change detection. The ones using contextual embeddings are either interpretable but unscalable (Giulianelli et al., 2020; Martinc et al., 2020a) or scalable but uninterpretable (Martinc et al., 2020b). The scalability issues of interpretable methods can be divided into two problems.

Memory consumption: Giulianelli et al. (2020) and Martinc et al. (2020a) apply clustering on all embeddings of each target word. This procedure becomes unfeasible for large sets of target words or if the embeddings need to be generated on a large corpus, since too many embeddings need to be saved into memory for further processing. To give an example, single-precision floating-point in Python requires 4 bytes of memory. Each contextual embedding contains 768 floats (Devlin et al.,

[1] https://blog.aylien.com/free-coronavirus-news-dataset/

[2] The code can be found at https://github.com/matejMartinc/scalable_semantic_shift

2019), leading each embedding to occupy 3072 bytes[3]. To use the previous methods on the Aylien Coronavirus News Dataset, which contains 250M tokens, about 768 Gb RAM would be necessary to store the embeddings for the entire corpus. If we limit our vocabulary to the 7,651 words that appear at least 50 times in every time slice and remove the stopwords (as we do in this work), we still need to generate contextual embeddings for 120M tokens, which is about 369 Gb of RAM.

Complexity of clustering algorithms: For the complexity analyses, we denote by d the dimension of the embedding, k is the number of clusters and n is the number of contextual embeddings, i.e., the number of word occurrences in the corpus. The time complexity of the affinity propagation algorithm (the best performing algorithm according to Martinc et al. (2020a)) is $O(n^2td)$, with t being the predefined maximum number of iterations of the data point message exchange. The time complexity of the simpler k-means algorithm[4] can be stated as $O(tknd)$, where t is the number of iterations of Lloyd's algorithm (Lloyd, 1982). As an example, consider the word *coronavirus*, which appears in the Aylien corpus about 1,2M times. For k-means with $k = 5$ and a maximal number of iterations set to 300 (the Scikit library default), about $300 * 5 * 1,300,000 * 768 \approx 1.5 \times 10^{12}$ operations are conducted for the clustering. With affinity propagation with the maximum number of iterations set to 200 (the default), clustering of the word *coronavirus* would require $1,300,000^2 * 200 * 768 \approx 2.6 \times 10^{17}$ operations, which is impossible to conduct in a reasonable amount of time on a high end desktop computer.

Contextual Embeddings Method with Interpretability Limitations: The averaging approach (Martinc et al., 2020b) eliminates the scalability problems: token embeddings for each word are not collected in a list but summed together in an element-wise fashion, which means that only 768 floats need to be saved for each word in the vocabulary. The averaged word representation is obtained for each time slice by dividing the sum by the word count. A single embedding per word is saved, leading to only 23.5 Mb of RAM required to store the embeddings for 7,651 words. These representations loose on the interpretability aspect, since all word usages are merged into a single averaged representation. It makes the method inappropriate for some tasks such as automatic labelling of word senses, and in some cases affects the overall performance of the method (Martinc et al., 2020a).

4 Methodology

Our word usage change detection pipeline follows the procedure proposed in the previous work (Martinc et al., 2020a; Giulianelli et al., 2020): for each word, we generate a set of contextual embeddings using BERT (Devlin et al., 2019). These representations are clustered using k-means or affinity propagation and the derived cluster distributions are compared across time slices by either using Jensen-Shannon divergence (JSD) (Lin, 2006) or the Wasserstein distance (WD) (Solomon, 2018). Finally, words are ranked according to the distance measure, assuming that the ranking resembles a relative degree of usage shift.

The primary contributions of this work lay in the embedding generation step, which improves the scalability of the method, and in leveraging WD to compute the distance between clusters. We also propose post-processing steps, which domain experts could use for the interpretation of results. We now describe the pipeline in more details.

4.1 Embeddings Generation

We use a pre-trained BERT model for each language of the evaluation corpora[5]. All models have 12 attention layers and a hidden layer of size 768. We fine-tune them for domain adaptation on each corpus as a masked language model for 5 epochs. Then, we extract token embeddings from the fine-tuned models. Each corpus is split into time slices. The models are fed 256 tokens long sequences in batches of 16 sequences at once. We generate sequence embeddings by summing the last four encoder output layers of BERT, following Devlin et al. (2019). Next, we split each sequence into 256 subparts to obtain a separate contextual embedding of size 768 for each token. Since one token does not necessarily correspond to one word due to byte-

[3] If we ignore the additional memory of a Python container—e.g., a Numpy list or a Pytorch tensor—required for storing this data.

[4] Here we are referring to the Scikit implementation of the algorithm employed in this work: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html.

[5] For German: bert-base-german-cased (https://deepset.ai/german-bert, for English: bert-base-uncased model, for Latin: bert-base-multilingual-uncased model from the `huggingface` library, for Swedish: bert-base-swedish-uncased (https://github.com/af-ai-center/SweBERT).

pair tokenization, we average embeddings for each byte-pair token constituting a word to obtain embeddings for each occurrence of a word.

Next, after obtaining a contextual embedding vector for each target word in a specific sequence, we decide whether this vector should be *saved* to the list or *merged* with one of the previously obtained vectors for the same word in the same time slice. To improve the scalability, we limit the number of contextual embeddings that are kept in the memory for a given word and time slice to a predefined threshold. The threshold of 200 was chosen empirically from a set of threshold candidates (20, 50, 100, 200, 500) and offers a reasonable compromise between scalability and performance. The new vector is merged if it is too similar—i.e., a duplicate or a near-duplicate—to one of the saved vectors or if the list already contains a predefined maximum number of vectors (200 in our case).

More formally, we add the new embedding e_{new} to the list of word embeddings $L = \{e_i, ..., e_n\}$ if:

$$|L| < 200 \quad \& \quad \forall e_i \in L\colon s(e_{\text{new}}, e_i) < 1 - \varepsilon$$

where s is the cosine similarity and ε is a threshold set to 0.01.

If $|L| \geq 200$ or if any vector in the list L is a near duplicate to e_{new}, we find a vector e_m in the list which is the closest to e_{new} in terms of cosine similarity:

$$e_m = \arg\max_{e_i \in L} \; s(e_i, e_{\text{new}})$$

This element e_m is then modified by summing it with e_{new}:

$$e_m \leftarrow e_m + e_{\text{new}}$$

The number of summed-up elements for each of the 200 groups in the list is stored besides their summed-up representations. Once the model has been fed with all the sequences in the time slice, the final summed-up vector is divided by this number to obtain an averaged embedding.

By having only 200 merged word embeddings per word per time slice, and by limiting the vocabulary of the corpus to 7,651 target words, we require up to 4.7 Gb of space for each time slice, no matter the size of the corpus. While this is still 200 times more space than if the averaging method was used (Martinc et al., 2020b), the conducted experiments show that the proposed method nevertheless keeps the bulk of the interpretability of the less scalable method proposed by Giulianelli et al. (2020), and offers competitive performance on several corpora.

4.2 Clustering

After collecting 200 vectors for each word in each time slice, we conduct clustering on these lists to extract the usage distribution of the word at each period. Clustering for a given word is performed on the set of all vectors from all time slices jointly.

We use two clustering methods previously applied for this task, namely k-means used in Giulianelli et al. (2020) and affinity propagation in Martinc et al. (2020a). The main strength of affinity propagation is that the number of clusters is not defined in advance but inferred during training. The clustering is usually skewed: a limited number of large clusters is accompanied with many clusters consisting of only a couple of instances. Thus, affinity propagation allows to pick out the core senses of a word. K-means tends to produce more even clusters. Appearance of small clusters that contain only few instances and do not represent a specific sense or usage of the word is nevertheless relatively common, since BERT is sensitive to syntax and pragmatics, which are not necessarily relevant for usage change detection. Another limitation of the k-means algorithm is that the number of clusters needs to be set in advance. This means that if the number of actual word usages is smaller than a predefined number of clusters, k-means will generate more than one cluster for each word usage.

To compensate for these deficiencies, we propose an additional *filtering and merging* step. A cluster is considered to be a legitimate representation of a usage of the word, if it contains at least 10 instances[6]. We compute the average embedding inside each cluster, and measure the cosine distance (1 - cosine similarity) between the average embeddings in each pair of legitimate clusters for a given word. If the distance between two clusters is smaller than a threshold, the clusters are merged. The threshold is defined as $avg_{cd} - 2 * std_{cd}$, where avg_{cd} is the average pairwise cosine distance between all legitimate clusters and std_{cd} is the standard deviation of that distance. This merging procedure is applied recursively until the minimum distance between the two closest clusters is larger than the threshold. After that, the merging proce-

[6] The threshold of 10 was derived from the procedure for manual labelling employed in the SemEval Task (Schlechtweg et al., 2020), where a constraint was enforced that the specific sense is attested at least 5 times in a specific time period in order to contribute word senses. We set the overall threshold of 10, which roughly translates to 5 per time period, since all of our test corpora (besides Aylien) contain two time periods.

dure is applied to illegitimate clusters (that contain less than 10 instances), using the same threshold. Illegitimate clusters could be added into one of the legitimate clusters or merged together to form a legitimate cluster with more than 10 instances. If there is no cluster that is close enough to be merged with, the illegitimate cluster is removed.

4.3 Change Detection and Interpretation

After the clustering procedure described above, for each word in each time slice, we extract its cluster distribution and normalise it by the word frequency in the time slice. Then target words are *ranked* according to the usage divergence between successive time slices, measured with the JSD or the WD[7]. If a ground-truth ranking exists, the method can be evaluated using the Spearman Rank Correlation to compare the true and the outputted ranking. In the exploratory scenario, the ranking is used to detect the most changing words and then investigate the most unevenly distributed clusters over time for the interpretation of the change.

JSD has been used for semantic shift detection in several recent papers, e.g. (Martinc et al., 2020a; Giulianelli et al., 2020; Kutuzov and Giulianelli, 2020). Since this is the first paper applying WD for this purpose, we describe it in more details.

The motivation for using the WD (Solomon, 2018) is to take into account the position of the clusters in the semantic space when comparing them. The JSD leverages semantic information encoded in the embeddings indirectly, distilled into two time-specific cluster distributions that JSD receives as an input. In addition to cluster distributions, WD accesses characteristics of the semantic space explicitly, through a matrix of cluster averages (obtained by averaging embeddings in each cluster) of size $T \times k \times 768$, where k is a number of clusters, T is a number of time slices and 768 is the embedding dimension.

This setup is a classical problem that can be solved using optimal transport (Peyré et al., 2019). We denote with μ_1 and μ_2 the sets of k average embedding points in the two vector spaces, and with c_1 and c_2 the associated clusters distributions. Thus, c_1 and c_2 are histograms on the simplex (positive and sum to 1) that represent the weights of each embedding in the source (μ_1) and target (μ_2) distributions. The task is to quantify the effort of moving one unit of mass from μ_1 to μ_2 using a chosen cost function, in our case the cosine distance. It is solved by looking for the transport plan γ, which is the minimal effort required to reconfigure c_1's mass distribution into that of c_2. The WD is the sum of all travels that have to be made to solve the problem:

$$\mathrm{WD}(c_1, c_2) = \min_{\gamma} \sum_{i,j} \gamma_{i,j} M_{i,j}$$
$$\text{with } \gamma 1 = c_1;\ \gamma^{\intercal} 1 = c_2;\ \gamma \geq 0$$

Where $M \in \mathbb{R}^{+}_{m \times n}$ is the cost matrix defining the cost to move mass from μ_1 to μ_2. We use the cosine similarity s, with $M = 1 - s(\mu_1, \mu_2)$.

Interpretation. Once the most changing words are detected, the next step is to understand *how* they change between two time slices by interpreting their clusters of usages.

Cluster distributions can be used directly to identify the clusters that are unevenly distributed across a time dimension. However, a cluster itself may consist of several hundreds or thousands of word usages, i.e. sentences. Interpreting the underlying sense behind each cluster by manually looking at the sentences is time-consuming. To reduce human work, we extract the most discriminating words and bigrams for each cluster: by considering a cluster as a single document and all clusters as a corpus, we compute the term frequency - inverse document frequency (tf-idf) score of each word and bigram in each cluster. The stopwords and the words appearing in more than 80% of the clusters are excluded to ensure that the selected keywords are the most discriminant. Thus, a ranked list of keywords for each cluster is obtained and top-ranked keywords are used for the interpretation of the cluster.

5 Evaluation

We use six existing manually annotated datasets for evaluation. The first dataset, proposed by Gulordava and Baroni (2011), consists of 100 English words labelled by five annotators according to the level of semantic change between the 1960s and 1990s[8]. To build the dataset, the annotators evaluated semantic change using their intuition, without looking at the context. This procedure is problematic since an annotator may forget or not be aware of a particular sense of the word.

[7] Using the POT package https://pythonot.github.io/.

[8] In order to make the proposed approach comparable to previous work, we remove four words that do not appear in the BERT vocabulary from the evaluation dataset, same as in Martinc et al. (2020a).

	COHA	**SE English**	**SE Latin**	**SE German**	**SE Swedish**	**DURel**	**Avg. all**
METHODS NOT USING CLUSTERING							
SGNS + OP + CD	0.347	0.321	0.372	**0.712**	**0.631**	**0.814**	**0.533**
Nearest Neighbors	0.310	0.150	0.273	0.627	0.404	0.590	0.392
Averaging	0.349	0.315	**0.496**	0.565	0.212	0.656	0.432
NON-SCALABLE CLUSTERING METHODS							
k-means 5 JSD	0.508	0.189	0.324	0.528	0.238	0.560	0.391
aff-prop JSD	**0.510**	0.313	0.467	0.436	-0.026	0.542	0.374
INTERPRETABLE SCALABLE METHODS							
Without filtering or merging of clusters							
k-means 5 JSD	0.430	0.316	0.358	0.508	0.073	0.658	0.390
aff-prop JSD	0.394	0.371	0.346	0.498	0.012	0.512	0.355
k-means 5 WD	0.372	0.360	0.450	0.514	0.316	0.607	0.437
aff-prop WD	0.369	**0.456**	0.397	0.421	0.264	0.484	0.399
With filtering and merging of clusters							
k-means 5 JSD	0.448	0.318	0.374	0.519	0.073	0.649	0.397
aff-prop JSD	0.403	0.348	0.408	0.583	0.018	0.712	0.412
k-means 5 WD	0.382	0.375	0.466	0.520	0.332	0.628	0.451
aff-prop WD	0.352	0.437	0.488	0.561	0.321	0.686	0.474

Table 1: Spearman Rank Correlation between system output rankings and ground truth rankings for various datasets. “SE” stands for SemEval.

The organizers of the recent SemEval-2020 Task 1— Unsupervised Lexical Semantic Change Detection (Schlechtweg et al., 2020)—employed another approach: the annotators had to decide whether a pair of sentences from different time periods convey the same meaning of the word (Schlechtweg and Schulte im Walde, 2020). For each of the four languages—German, English, Latin and Swedish—senses were manually annotated by labeling word senses in a pair of sentences drawn from different time periods. All SemEval-2020 Task 1 corpora contain only two periods and the sentences are shuffled and lemmatized. The lexical semantic change score is defined as the difference between word sense frequency distributions in the two time periods and measured by the Jensen-Shannon Distance (Lin, 2006).

The DURel dataset (Schlechtweg et al., 2018) is composed of 22 German words, ranked by semantic change by five annotators between two time periods, 1750–1799 and 1850–1899. Similarly to SemEval, the ranking was build by evaluating the relatedness of pairs of sentences from two periods.

In order to conduct usage change detection on the target words proposed by Gulordava and Baroni (2011), we fine-tune the English BERT-base-uncased model and generate contextual embeddings on the Corpus of Historical American English (COHA)[9]. We only use data from the 1960s to the 1990s (1960s has around 2.8M and 1990s 3.3M words), to match the manually annotated data. For the SemEval Task 1 evaluation set, we fine-tune the BERT models and generate contextual embeddings on the four corpora provided by the organizers of the task, English (about 13.4M words), German (142M words), Swedish (182M words) and Latin (11.2M words). Finally, we fine-tune BERT and generate embeddings on the German DTA corpus (1750–1799 period has about 25M and 1850–1899 has 38M tokens)[10].

The results are shown in Table 1. We compare our scalable approach with the *non-scalable clustering* methods used by Giulianelli et al. (2020) and Martinc et al. (2020a). *Averaging* (Martinc et al., 2020b) is the less interpretable method described in Section 3. *SGNS + OP + CD* (Schlechtweg et al., 2019) refers to the state-of-the-art semantic change detection method employing non-contextual word embeddings: the Skip-Gram with Negative Sampling (SGNS) model is trained on two periods independently and aligned using Orthogonal Procrustes (OP). Cosine Distance (CD) is used to compute the semantic change. The *Nearest Neighbors* method (Gonen et al., 2020) also uses SGNS embeddings.

[9] https://www.english-corpora.org/coha/

[10] https://www.ims.uni-stuttgart.de/en/research/resources/experiment-data/durel/

For each period, a word is represented by its top nearest neighbors (NN) according to CD. Semantic change is measured as the size of the intersection between the NN lists of two periods.

On average, the proposed scalable clustering with filtering and merging of clusters leads to a higher correlation with gold standard than the standard non-scalable clustering methods: the best method (aff-prop WD) achieving a Spearman correlation with the gold standard of 0.474 compared to the best non-scalable k-means 5 JSD achieving the Spearman correlation of 0.391. The method also outperforms averaging and NN, though it is outperformed by a large margin by the SGNS+OP+CD, achieving the score of 0.533.

The best performing clustering algorithm differs for different datasets. On average, affinity propagation only outperforms k-means when filtering and merging of clusters is employed. The effect of the filtering on k-means is positive on average but the difference is thin, as the number of clusters is low.

WD leads to better results than JSD on most of the corpora where averaging outperforms clustering, the only exception is DURel. An extreme example is the Swedish SemEval dataset, where the clustering with JSD performs particularly poorly: using the WD, which takes into account the average embeddings on top of cluster distributions, greatly increases the correlation with the gold standard. On the contrary, on COHA where averaging performs poorly in comparison to clustering, WD is under-performing.

6 Use Case: Aylien COVID-19 Corpus

The combination of scalable clustering with the interpretation pipeline opens new opportunities for diachronic corpus exploration. In this section, we demonstrate how it could be used to analyze the Aylien Coronavirus News Dataset. The corpus contains about 500k news articles related to COVID-19 from January to April 2020[11], unevenly distributed over the months (160M words in March, 41M in February, 35M in April and 10M in January). We split the corpus into monthly chunks and apply our scalable word usage change detection method.

6.1 Identification of the Top Drifting Words

The scalable method allows to perform embeddings extraction and clustering for all words in the corpus. We extract the top words with the highest average WD between the successive months to conduct a deeper analysis. We exclude words that appear less than 50 times in each month to avoid spurious drifts due to words having too few occurrences in a time slice. However, some drifts due to corpus artefacts remain, in particular dates such as *'2019-20'*. Thus, words containing numbers and one-letter words are also removed.

1	diamond	6	tag
2	king	7	paramount
3	ash	8	lynch
4	palm	9	developers
5	fund	10	morris

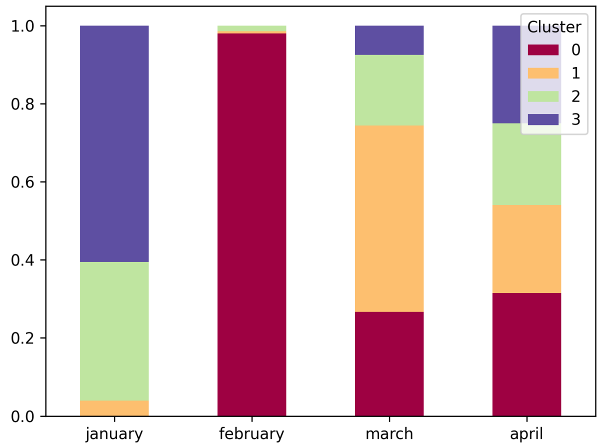
Table 2: Top 10 most changed words in the corpus according to a monthly-averaged WD of k-means ($k = 5$) cluster distributions.

In Table 2 we present the top 10 most drifting words extracted using k-means with k=5 and ranked according to the average WD across the four months[12]. Among them, the word *diamond* is related to the cruise ship “Diamond Princess”, which suffered from an outbreak of COVID-19 and was quarantined for several weeks. The word *king*, which is the second most changing word, is related to the King county, Washington, where the first confirmed COVID-19 related death in the USA appeared, and to the Netflix show “Tiger King”, which was released in March. Thus, the primary context for this word changed several times, which is reflected in our results. Other words are mostly constituent words in named entities, related e.g., to an American Society of Hematology (ASH) Research Collaborative's Data Hub, which is capturing data on subjects tested positive for COVID-19.

The results suggest that the model does what it is meant to do: for most words in the list it is possible to find an explanation why its usage changed during the beginning of 2020. The list contains many proper names or proper name constituents, which could be either desirable or undesirable property, depending on research goals. Some work focuses specifically on proper names (Hennig and Wilson, 2020), since they could be a good proxy to shifts in socio-political situations. On the other hand, if

[11] We used an older version of the corpus. Currently the data from May are also available.

[12] This is a rather arbitrary procedure: one can imagine that a domain expert would prefer a different frequency threshold or focus more on a given month. The most time-consuming part is embedding extraction. Once this is done, clustering and keyword extraction can be done as many times as necessary.



#	Keywords
0	diamond princess, cruise ship, princess cruise, japanese, tested positive, confirm, ship diamond
1	neil diamond, comic, sweet caroline, trump, song, diamond said, comic book,
2	diamond hill, hill capital, diamond jubilee, diamond mountain, league postponed, portfolio, athletics
3	diamond industry, black diamond, jewellery, hong kong, diamond ring, surat diamond, india

Figure 1: Cluster distributions per month and top keywords for each cluster for word *diamond*.

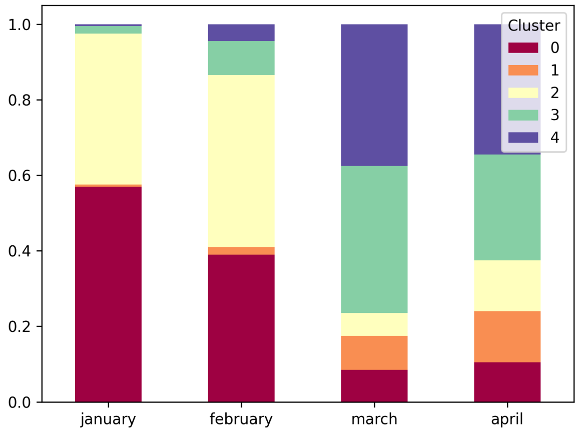
the focus of the study are shifts in more abstract concepts, then proper names could be filtered out before the embedding generation stage by employing named entity recognition tools.

6.2 Interpretation of the Usage Change

The interpretation pipeline, described in Section 4.3, is illustrated in figures 1 and 2. We focus on two words, *diamond* and *strain*, to show the various phenomena that can be detected. *Diamond* is the top drifting word in the entire vocabulary (see Table 2); it can be both a common noun and an entity, inducing usage drift when the entity appears in the newspapers after events with high media coverage. *Strain* is the 38th word with the highest drift overall, and the 15th highest between February and March 2020. It has several different senses whose usage vary across time following the events in the news. We cluster their vector representations from the Aylien corpus using k-means with $k = 5$ and apply the cluster filtering and merging step. Then, using tf-idf on unigrams and bigrams, we extract a set of keywords for each cluster to interpret the variations of their distribution.

The keywords and cluster distributions for the word *diamond* can be found in Figure 1. One of the clusters was removed at the filtering step, as it had less than 10 embeddings inside, and no other cluster was close enough. A clear temporal tendency is visible from the cluster distribution in Figure 1: a new major usage appears in February, corresponding to the event of the quarantined cruise ship (Cluster 0); this association is revealed by the keywords for this cluster. Moreover, the WD between January and February, when the outbreak happened, is 0.337; it is also very high between February and March (0.342). It reflects the large gap between the cluster distributions, first with the appearance of Cluster 0 in February that made the other usages of the word *diamond* in the media almost disappear, and then the reappearance of other usages in March, when the situation around the cruise ship gradually normalized. Cluster 1, that appears in March, is related to Neil Diamond's coronavirus parody of the song “Sweet Caroline" which was shared mid-March on the social media platforms and received a lot of attention in the US. Cluster 3 is related to the diamond industry; it is much less discussed as soon as the pandemic breaks out in February. Finally, Cluster 2 deals with several topics: Diamond Hill Capital, a US investment company, and the Wanda Diamond League, an international track and field athletic competition which saw most of its meetings postponed because of the pandemic. This last cluster shows the limitations of our clustering: it is complex to identify and differentiate all the usages of a word perfectly.

The keywords and cluster distributions for the word *strain* can be found in Figure 2. This is a polysemic word with two main senses in our corpus: as the variant of a virus or bacteria (biological term) and as “a severe or excessive demand on the strength, resources, or abilities of someone or something” (Oxford dictionary). Clusters 1, 3 and 4, which roughly match the second sense of the word (strain on healthcare systems in cluster 4, financial strain in cluster 3 and strain on resources and infrastructure in cluster 1), grow bigger across time, while clusters 0 and 2, which match the first sense of the word (e.g., new virus strain), shrink. This behavior underlines the evolution of the concerns related to the pandemic in the newspapers.



#	Keywords
0	strain coronavirus, new strain, city wuhan, novel strain, strain virus, chinese city
1	strain health, strain resources, stream, network infrastructure, international resources, likely strain
2	new strain, acute respiratory, 2019 ncov, respiratory syndrome, severe acute, identified humans
3	financial strain, feeling strain, strain coronavirus, economic strain, signs strain, strain said
4	ease strain, putting strain, strain health, reduce strain, care system, strain hospitals

Figure 2: Cluster distributions per month and top keywords for each cluster for word *strain*.

7 Conclusion

We proposed a scalable and interpretable method for word usage change detection, which outperforms the non-scalable contextual embeddings-based methods by a large margin. The new method also allows completely data-driven analysis of word sense dynamic in large corpora, which was impossible to conduct with unscalable methods. This opens new opportunities in both language change studies and text stream monitoring tasks. In this paper we focused on the latter application by analysing a large corpus of COVID-19 related news.

The method is outperformed by the state-of-the-art SGNS+OP+CD method. We hypothesise that this can be connected with the fact that the sentences in all but one evaluation corpus (COHA) are shuffled, meaning that BERT models cannot leverage the usual sequence of 512 tokens as a context, but are limited to the number of tokens in the sentence. We will explore this hypothesis in the future.

Despite achieving lower performance than the SGNS+OP+CD method, we nevertheless argue that our method offers a more fine-grained interpretation than methods based on non-contextual embeddings, since it accounts for the fact that words can have multiple meanings. The cluster-based technique returns a degree of change and a set of sentence clusters for each word in the corpus, roughly corresponding to word senses or particular usages. For this reason, the approach can be used for detection of new word usages and for tracing how these usages disappear, as we have shown in Section 6. Even more, word usages and their distributions over time could be linked with real-word events by labeling sentence clusters with a set of cluster-specific keywords.

Overall, we observe a large disparity between results on different evaluation corpora. This is in line with the results of the Semeval 2020 task 1 (Schlechtweg et al., 2020), where none of the best-performing methods was able to achieve the best result on all corpora. In practice, different methods focus on different aspects of word usage change: Averaging and SGNS+OP+CD focus on average variation of word usage, hiding the intra-period diversity. When it comes to clustering, JSD-based method detects the appearance or disappearance of a given usage, even a minor one. The WD-based method, using information from both the cluster distribution and the embeddings vectors, represents a compromise between the averaging and the JSD-based methods.

In this paper we follow the general approach in semantic shift detection literature and apply our analysis on the raw text. However, our results demonstrate that at least news monitoring applications would benefit from the application of the traditional text processing pipeline, in particular the extraction of named entities and dates. This will be addressed in the future work.

Acknowledgements

This work has been supported by the European Union Horizon 2020 research and innovation programme under grants 770299 (NewsEye) and 825153 (EMBEDDIA), the project Computer-assisted multilingual news discourse analysis with contextual embeddings (CANDAS, J6-2581), and Project Development of Slovene in the Digital Environment (RSDO).

References

Hosein Azarbonyad, Mostafa Dehghani, Kaspar Beelen, Alexandra Arkut, Maarten Marx, and Jaap Kamps. 2017. Words are malleable: Computing semantic shifts in political and media discourse. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1509–1518.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Haim Dubossarsky, Simon Hengchen, Nina Tahmasebi, and Dominik Schlechtweg. 2019. Time-out: Temporal referencing for robust modeling of lexical semantic change. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 457–470, Florence, Italy. Association for Computational Linguistics.

Haim Dubossarsky, Daphna Weinshall, and Eitan Grossman. 2017. Outta control: Laws of semantic change and inherent biases in word representation models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1136–1145. Association for Computational Linguistics.

Mario Giulianelli, Marco Del Tredici, and Raquel Fernández. 2020. Analysing lexical semantic change with contextualised word representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3960–3973, Online. Association for Computational Linguistics.

Hila Gonen, Ganesh Jawahar, Djamé Seddah, and Yoav Goldberg. 2020. Simple, interpretable and stable method for detecting words with usage change across corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 538–555, Online. Association for Computational Linguistics.

Kristina Gulordava and Marco Baroni. 2011. A distributional similarity approach to the detection of semantic change in the google books ngram corpus. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 67–71. Association for Computational Linguistics.

William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2016a. Cultural shift or linguistic drift? comparing two computational measures of semantic change. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2116–2121, Austin, Texas. Association for Computational Linguistics.

William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2016b. Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1489–1501.

Felix Hennig and Steven Wilson. 2020. Diachronic embeddings for people in the news. In *Proceedings of the Fourth Workshop on Natural Language Processing and Computational Social Science*, pages 173–183.

Yoon Kim, Yi-I Chiu, Kentaro Hanaki, Darshan Hegde, and Slav Petrov. 2014. Temporal analysis of language through neural language models. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, pages 61–65.

Andrey Kutuzov and Mario Giulianelli. 2020. UiO-UvA at SemEval-2020 task 1: Contextualised embeddings for lexical semantic change detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 126–134, Barcelona (online). International Committee for Computational Linguistics.

Andrey Kutuzov, Lilja Øvrelid, Terrence Szymanski, and Erik Velldal. 2018. Diachronic word embeddings and semantic shifts: a survey. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1384–1397. Association for Computational Linguistics.

Andrey Kutuzov, Erik Velldal, and Lilja Øvrelid. 2017. Tracing armed conflicts with diachronic word embedding models. In *Proceedings of the Events and Stories in the News Workshop*, pages 31–36, Vancouver, Canada. Association for Computational Linguistics.

J. Lin. 2006. Divergence measures based on the shannon entropy. *IEEE Trans. Inf. Theor.*, 37(1):145–151.

Stuart Lloyd. 1982. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137.

Matej Martinc, Syrielle Montariol, Elaine Zosa, and Lidia Pivovarova. 2020a. Capturing evolution in word usage: Just add more clusters? In *Companion Proceedings of the Web Conference 2020*, WWW '20, page 343–349, New York, NY, USA. Association for Computing Machinery.

Matej Martinc, Petra Kralj Novak, and Senja Pollak. 2020b. Leveraging contextual embeddings for detecting diachronic semantic shift. In *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, pages 4811—4819.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Gabriel Peyré, Marco Cuturi, et al. 2019. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607.

Dominik Schlechtweg, Anna Hätty, Marco Del Tredici, and Sabine Schulte im Walde. 2019. A wind of change: Detecting and evaluating lexical semantic change across times and domains. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 732–746, Florence, Italy. Association for Computational Linguistics.

Dominik Schlechtweg, Barbara McGillivray, Simon Hengchen, Haim Dubossarsky, and Nina Tahmasebi. 2020. SemEval-2020 task 1: Unsupervised lexical semantic change detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1–23, Barcelona (online). International Committee for Computational Linguistics.

Dominik Schlechtweg and Sabine Schulte im Walde. 2020. Simulating lexical semantic change from sense-annotated data. *CoRR*, abs/2001.03216.

Dominik Schlechtweg, Sabine Schulte im Walde, and Stefanie Eckmann. 2018. Diachronic usage relatedness (DURel): A framework for the annotation of lexical semantic change. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 169–174, New Orleans, Louisiana. Association for Computational Linguistics.

Philippa Shoemark, Farhana Ferdousi Liza, Dong Nguyen, Scott Hale, and Barbara McGillivray. 2019. Room to Glo: A systematic comparison of semantic change detection approaches with word embeddings. In *Proceedings of EMNLP-IJCNLP 2019*, pages 66–76, Hong Kong, China. Association for Computational Linguistics.

Justin Solomon. 2018. Optimal transport on discrete domains.

Ian Stewart, Dustin Arendt, Eric Bell, and Svitlana Volkova. 2017. Measuring, predicting and visualizing short-term change in word representation and usage in VKontakte social network. In *Eleventh international AAAI conference on web and social media*.

Nina Tahmasebi, Lars Borin, and Adam Jatowt. 2018. Survey of computational approaches to diachronic conceptual change. *CoRR*, 1811.06278.

Xuri Tang. 2018. A state-of-the-art of semantic change computation. *Natural Language Engineering*, 24(5):649–676.

Zi Yin, Vin Sachidananda, and Balaji Prabhakar. 2018. The global anchor method for quantifying linguistic shifts and domain adaptation. In *Advances in neural information processing systems*, pages 9412–9423.