What is Done is Done: an Incremental Approach to Semantic Shift Detection

Alfio Ferrara and **Stefano Montanelli** and **Francesco Periti** and **Martin Ruskov**
Department of Computer Science - University of Milan
Via Celoria, 18 - 20133 Milano, Italy
firstname.lastname@unimi.it

Abstract

Contextual word embedding techniques for semantic shift detection are receiving more and more attention. In this paper, we present *What is Done is Done* (WiDiD), an incremental approach to semantic shift detection based on incremental clustering techniques and contextual embedding methods to capture the changes over the meanings of a target word along a diachronic corpus. In WiDiD, the word contexts observed in the past are consolidated as a set of clusters that constitute the "memory" of the word meanings observed so far. Such a memory is exploited as a basis for subsequent word observations, so that the meanings observed in the present are stratified over the past ones.

1 Introduction

The use of contextual embedding techniques is receiving more and more attention in the field of semantic shift detection. In particular, pre-trained models like BERT (Hu et al., 2019; Martinc et al., 2020a), ELMo (Kutuzov and Giulianelli, 2020; Rodina et al., 2020), and XLM-R (Cuba Gyllensten et al., 2020; Rother et al., 2020), are being proposed as promising solutions to capture the different meanings of a target word according to the different contexts in which the word appears throughout a considered diachronic corpus. Such solutions generally employ clustering techniques to aggregate embeddings of a specific word into clusters (Martinc et al., 2020a; Karnysheva and Schwarz, 2020). The idea is that each cluster denotes a specific *word meaning* that can be recognized in the considered documents. In this way, it is possible to analyze the shift of a word meaning/sense by exploiting the evolution of a cluster over time. For instance, an increasing number of elements in a cluster denotes that the associated word meaning is getting frequently adopted. On the opposite, a cluster with a decreasing number of elements over time refers to a word meaning that is getting obsolete. Usually, the corpus is static, meaning that all the documents of the considered time periods are available as one whole, and a single clustering activity is performed over the entire corpus, generating clusters of word meaning with documents of different time periods (Kutuzov et al., 2018; Tahmasebi et al., 2018, 2021). As a result, the time period in which a document is added to the corpus is not taken into account for cluster composition, and this is not completely satisfactory for an appropriate recognition of meaning changes over time. When a dynamic corpus is considered, namely time periods and documents can be progressively added, scalability issues also arise, since the clusters of word meanings need to be re-calculated or updated. As a possible solution, some recent works propose to perform clustering separately for each time period. In this case, the resulting clusters need to be aligned in order to recognize similar word meanings in different, consecutive time periods (Kanjirangat et al., 2020; Montariol et al., 2021). However, solutions based on clustering alignment are not satisfactory as well, since they do not capture the possible evolution pattern of a meaning across different time periods. A recent work proposes an average-based approach to track semantic shift via continuously evolving embeddings (Horn, 2021) computed as a weighted running average (Finch, 2009) of embeddings generated by a contextual model. This method is suitable to be applied on stream data and it is far more scalable than typically cluster-based methods. Nevertheless, it does not allow to analyse which meanings are actually changed.

In this paper, we present *What is Done is Done* (WiDiD), an incremental approach to semantic shift detection based on incremental clustering techniques and contextual embeddings to capture the changes over the meanings of a target word along a diachronic corpus. In WiDiD, we work under the assumption that the documents of the corpus become

Proceedings of the 3rd Workshop on Computational Approaches to Historical Language Change, pages 33 - 43

available as a stream and they are segmented in a sequence of time periods. The word contexts observed in past time periods are consolidated as a set of clusters that constitute the "memory" of the word meanings observed so far. Such a memory is then exploited as a basis for subsequent word observations in the current time period. The idea of WiDiD is that the clusters of word meanings previously created cannot be changed (*what is done is done*), and the word meanings that are observed in the present must be stratified/integrated over the past ones. To enforce scalability, incremental clustering techniques are employed in WiDiD, so that the word embeddings extracted from the documents of the current time period are compared and assimilated into the set of consolidated clusters coming from the past time periods. A comparative evaluation of the proposed WiDiD approach against a reference benchmark is discussed in the paper according to multiple configurations characterized by different clustering algorithms and embedding methods. In particular, we present experiments based on a pre-trained BERT model as well as results obtained from a trained Doc2Vec model, which has been adapted to provide pseudo-contextual word embeddings to extend the conventional static word representations of context-free embedding techniques. As a further contribution of WiDiD, different metrics for semantic shift evaluation of word meanings are defined in the paper and experimental results are provided to discuss their effectiveness.

The paper is organized as follows. In Section 2, the relevant literature is discussed. In Section 3, we present the WiDiD approach. Incremental clustering techniques and semantic shift measures of WiDiD are illustrated in Sections 4 and 5, respectively. Experimental results are discussed in Section 6. Section 7 finally provides our concluding remarks.

2 Related work

Works related to WiDiD are about the use of word embeddings for semantic shift detection by leveraging the idea that semantically-related words are close to each other in the embedding space (Mikolov et al., 2013). In approaches relying on context-free embeddings, independent word vectors defined over different "temporal" vector spaces can be compared after applying an alignment mechanism (Hamilton et al., 2016) such as the Procrustes (Schönemann, 1966). Moreover, recent contextualised architectures are proposed, like ELMo (Peters et al., 2018), BERT (Devlin et al., 2019), and XLM-R (Conneau et al., 2020), which generate dynamic word embeddings according to the use of the words in the input sequences, thus enabling the recognition of different meanings by comparing the context in which words are used throughout the text. The solution proposed in Hu et al. (2019) is one of the first examples based on BERT embeddings to track changes in word meanings and it requires lexicographic supervision, like the use of a reference dictionary (e.g., the Oxford dictionary for the English language) to list the possible word meanings beforehand, thus it is hardly applicable to low-resource languages.

A number of unsupervised approaches based on contextual embeddings are proposed to sidestep the need of lexicographic resources (Schlechtweg et al., 2020; Tahmasebi et al., 2021). In general, these kinds of approaches follow a three-step scheme: i) extraction of embeddings for each occurrence of a target word from a contextual model such as BERT (Hu et al., 2019; Martinc et al., 2020a), ELMo (Kutuzov and Giulianelli, 2020; Rodina et al., 2020), or XLM-R (Cuba Gyllensten et al., 2020; Rother et al., 2020); ii) aggregation of the embeddings with a clustering algorithm like K-Means (Giulianelli et al., 2020; Cuba Gyllensten et al., 2020), Affinity Propagation (Martinc et al., 2020a; Kutuzov and Giulianelli, 2020), or DBSCAN (Rother et al., 2020; Karnysheva and Schwarz, 2020); iii) comparison of the vector distribution over clusters according to time by using a semantic distance measure, like Jensen-Shannon divergence (Martinc et al., 2020a), Entropy Difference (Giulianelli et al., 2020), or Wasserstein Distance (Montariol et al., 2021). The main limitation of applying clustering to word embeddings is the scalability issues about memory consumption and time. As a recent contribution, in Montariol et al. (2021), a scalable and interpretable method is proposed based on merging of similar embeddings to reduce the number of representations to consider for a given word and time slice. Further solutions to overcome scalability issues are provided by Rodina et al. (2020) and Laicher et al. (2021). In particular, they propose to limit the number of embeddings by randomly sampling sentences from each period. The intrinsic time-complexity issues of applying clustering algorithms to embeddings are also addressed in Rother et al. (2020) by reducing the embedding dimensionality. In Martinc et al. (2019),

the contextual embeddings of a word are averaged to generate a single word representation for each time period. In Giulianelli et al. (2020), the average pairwise distance between embeddings of different time periods is calculated. Even if these solutions are more efficient and scalable than clustering, they provide uninterpretable results since multiple word occurrences are collapsed into a single representation, like in context-free embeddings. Most of the *cluster*- and *average*-based approaches estimate the magnitude of semantic shifts ignoring the uncertainty of their estimations. As a result, estimations can be erroneously inflated since the irregularities of word frequencies over time can negatively affect the stability of word embeddings (Zhou et al., 2021; Wendlandt et al., 2018). In this respect, Liu et al. (2021) propose a solution based on the combination of BERT embeddings with permutation-based statistical test and term-frequency thresholding.

Original contribution of WiDiD. With respect to the above solutions, the WiDiD approach is based on incremental clustering techniques applied to contextual word embeddings. In WiDiD, the "memory" of word meanings observed in the past is consolidated in a set of clusters that is not re-calculated in subsequent time periods. As a result, only the word embeddings of the current time period are analyzed with the aim to measure the change with respect to the clusters of past word meanings. This way, it is possible to compare specific word meanings also from a qualitative point of view (i.e., interpretable results) without requiring any alignment mechanisim across time periods. In other words, the stratified layers of clusters over time allow to reconstruct not only the quantity of semantic shift but also the evolution of a word meaning.

3 The WiDiD approach

Consider a diachronic document corpus $\mathcal{C} = C_1 \cup C_2$ where C_2 denotes a set of documents of the time t and C_1 denotes a set of documents cumulatively collected in the $t - n$ time periods prior to t. Given a target word w, the goal of semantic shift detection is to measure how much the meaning(s) of w is changed from C_1 to C_2. The WiDiD approach relies on a contextual embedding model to represent each occurrence of the target word w in a corpus C_j (either C_1 or C_2). We keep track of the word embedding representations collected for w over time by relying on the embedding model E_1 that contains the word vectors computed over C_1. Given this input, we process the new documents in C_2 as follows (see Figure 1).

Document selection. In this step, we select the subset of documents $C_{w,2} \subseteq C_2$ that are relevant for the word w. $C_{w,2}$ is composed by the documents containing the word w. As an alternative, any information retrieval technique suitable for finding relevant documents for a given target can be exploited for the composition of $C_{w,2}$.

Fine tuning. In this step, the model E_1 used to generate the word vectors over C_1 can be optionally updated/fine-tuned into a new model E_2 to take into account the new documents in C_2 (Kim et al., 2014; Giulianelli, 2019). When the observed time t is the initial one, the model E_1 is trained on C_2 or a pre-trained model is used. The WiDiD approach is compatible with any technique for contextual word embedding, that is any method that produces a vector embedding the meaning of a word in a specific document.

Embedding extraction. In this step, we isolate the embedding vectors representing the contextual meaning of the word w. The contextualised embedded representation of the word w in the k-th document of a corpus $C_{w,j}$ is denoted by $e^j_{w,k}$. Then, the representation of the word w in the corpus C_j is defined as:

$$\Phi^j_w = \{e^j_{w,1}, \ldots, e^j_{w,m}\},$$

with m being the number of documents in $C_{w,j}$. As the final output of this step, we have two sets of embedding vectors: Φ^1_w that is produced in the previous iterations of the WiDiD approach over the corpus $C_{w,1}$ and Φ^2_w, produced at the current time t for the corpus $C_{w,2}$.

Clustering. In this step, vectors in $\Phi^1_w \cup \Phi^2_w$ are clustered in order to group vectors representing similar meanings. The set of clusters produced in this step is denoted K_2 and the i-th cluster in K_2 is denoted $\phi_{w,i}$. A distinguishing feature of WiDiD is to perform also the clustering step in an incremental fashion, by updating the clusters K_1 computed in the previous iterations of WiDiD. A more detailed description of the incremental clustering techniques used in WiDiD is given in Section 4. The clusters of K_2 can be classified in three types (see Figure 2). Cluster types (A) and (C) contain vectors that derive from a single corpus, either the past (i.e., C_1) or the current one (i.e., C_2). The cluster type (B) is

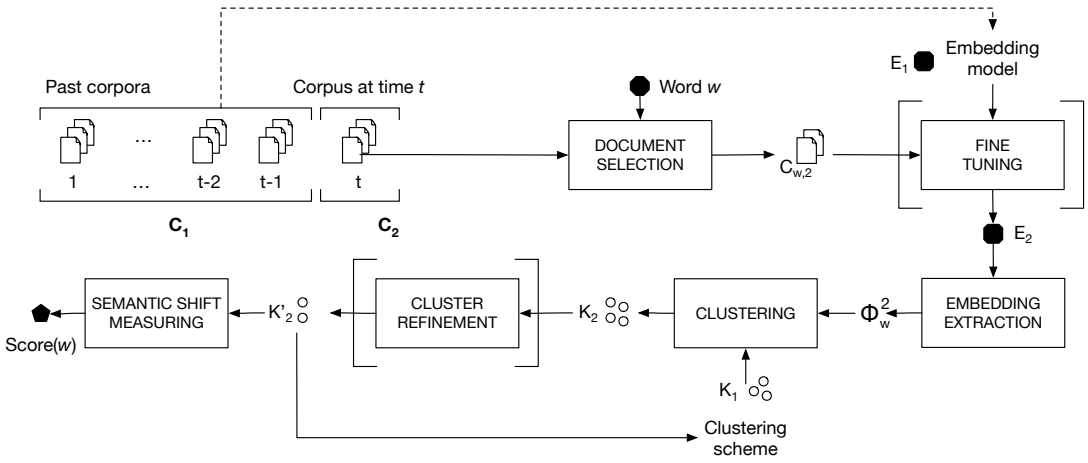

Figure 1: The WiDiD approach

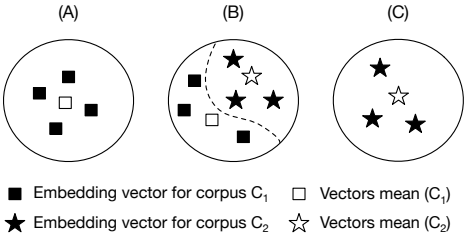

Figure 2: Types of clusters in K_2

a mixture of vectors from the past (corpus C_1) and vectors from the present time (corpus C_2). For each cluster, we compute also the mean μ_i of the vectors that are associated with the same time period (i.e., the same corpus).

Cluster refinement. The cluster set K_2 may contain poorly-informative clusters, such as clusters containing a single vector, or aged information, namely clusters that contain only vectors representing a word meaning observed a long time ago. In order to get rid of poor or aged information, in WiDiD, it is (optionally) possible to perform a cluster refinement step to drop the undesired clusters. We note that this step is also useful to reduce the information available about the past in view of a subsequent execution of WiDiD for the next time period $t+1$. With regard to poorly-informative clusters, we enforce standard cluster pruning techniques that are typically based on a threshold over the cluster size or the average distance of vectors from the cluster centroid (Raskutti and Leckie, 1999). For aged information, the idea of WiDiD is that each cluster is associated with an aging index that measures how recently the cluster has been updated during the incremental clustering process. This index is updated each time a cluster in the cluster set K_1 is upgraded by adding vectors of Φ^2_w (i.e., vectors deriving from the corpus C_2). A threshold over the aging index is then used to decide when an aged cluster should be pruned from K_2. As a result, this is a mechanism to regulate how much memory the WiDiD will keep about the past. The final pruned cluster set is denoted $K_2^{'}$ and will be the basis of the clustering step in the next iteration of WiDiD.

Semantic shift measuring. To evaluate whether a word w exhibits a semantic change between the two corpora C_1 and C_2, we measure the distance between the sets Φ^1_w and Φ^2_w using the clusters in $K_2^{'}$. Further details on how to measure semantic shift are provided in Section 5.

4 Incremental clustering

In WiDiD, we rely on incremental clustering to aggregate contextual embedding vectors that represent similar word meanings into the same cluster. We propose an incremental extension of Affinity Propagation (AP) (Frey and Dueck, 2007), called Affinity Propagation a Posteriori (APP) (see Algorithm 1). Let's call X and X_1, and L and L_1 the embeddings and the cluster labels at time t and $t-1$, respectively. At time $t=1$ the standard AP clustering is performed. At each time $t>1$, for each existing cluster computed at time $t-1$, the data points $x_i \in X_1$ are packed into a single average representation, i.e. the centroid μ of each cluster. The set of the centroids for X_1 is denoted μX_1. Then, the standard AP algorithm is executed on $\mu X_1 \cup X$, with the aim to obtain a new set of temporary labels L_2, i.e., the new assignment of data points to

Algorithm 1 *The APP algorithm*

Input
t: *time step*
X: *data at time step* t
X_1: *data at time step* $t-1$
L_1: *labels at time step* $t-1$
γ: *trim factor*

Output
L, X: *at time step* t

if t == 1 **then**
    L $\leftarrow AP$(X)
    L, X $\leftarrow Trim$(L, X, γ)
    yield L, X

else if t $>$ 1 **then**
    μX$_1$ $\leftarrow Pack$(L$_1$, X$_1$)
    L$_2$ $\leftarrow AP$(μX$_1$ $\cup$ X)
    μL$_1$, L $\leftarrow$ *Split*(L$_2$)
    L$_1$ $\leftarrow UnpackAndUpdate$($\mu$L$_1$, μX$_1$, L$_1$, X$_1$)
    L, X $\leftarrow Trim$(L$_1$ $\cup$ L, X$_1$ $\cup$ X, γ)
    yield L, X
end if

clusters. Such labels are then split in two subsets, μL_1 and L, which contain labels for each average representation in μX_1 and for each data point in X, respectively. Given $\mu L_1, \mu X_1, L_1, X_1$, we unpack the centroids of μL_1 into the corresponding data points X_1 mapping the previous labels L_1 into the new labels of their respective centroids μL_1. Intuitively, clusters from time step $t-1$ can't be changed, in the sense that each point from $t-1$ remain in the same cluster after running AP at time step t. However, each cluster from $t-1$ can be updated with points from t, and new clusters can be created at time step t containing no points from $t-1$. Finally, APP returns $L_1 \cup L$, which is the union of the unpacked and updated L_1 and L. APP includes the notion of aging index to use for cluster refinement, implemented through a trim factor γ. In our current implementation the idea of γ is that clusters containing less than γ percent of the whole set of embeddings $\Phi_w^1 \cup \Phi_w^2$ at time t are assumed to be poorly-informative and thus they are dropped.

5 Semantic shift measuring

Clustering contextual word embeddings for a word w at time t results in a set of k clusters $K^2 = \phi_{w,1}, ..., \phi_{w,k}$ where $\phi_{w,i} \subseteq \Phi_w^1 \cup \Phi_w^2$. In particular, we denote as $\phi_{w,i}^1, \phi_{w,i}^2$ the set of embeddings from Φ_w^1, and Φ_w^2 respectively, enclosed in the $i-$th cluster; formally we define $\phi_{w,i}^1 = \phi_{w,i} \cap \Phi_w^1$ and $\phi_{w,i}^2 = \phi_{w,i} \cap \Phi_w^2$. According to this, in WiDiD, we propose three different aggregation measures to estimate semantic change. Borrowing from Giulianelli (2019), we employ the Jensen-Shannon divergence to measure semantic change leveraging cluster distributions. In addition, we adapt the methods of Martinc et al. (2019) and Kutuzov (2020) for scenarios where embeddings are clustered.

Jensen-Shannon divergence (JSD). The Jensen-Shannon divergence quantify the similarity between two probability distributions using a symmetrization of the Kullback-Leibler divergence.

$$JSD(p_w^1, p_w^2) = H\left(\frac{1}{2}\left(p_w^1 + p_w^2\right)\right) - \frac{1}{2}\left(H(p_w^1) - H(p_w^2)\right) \quad (1)$$

To quantify changes between word senses we create two time-specific cluster distributions p_w^1, p_w^2 as the relative number of cluster members for $t-n$, and t, respectively (Hu et al., 2019). Intuitively, we compute the value related to the $i-$th cluster as:

$$p_{w,i}^j = \frac{|\phi_{w,i}^j|}{|\Phi_w^j|} \quad (2)$$

where $j \in \{1, 2\}$.

Distance between prototype embeddings (PDIS). Recent work used the term *word prototype* to indicate a 'prototypical' representation of the word computed by averaging all its embeddings in a specific temporal sub-corpus (Rodina et al., 2020; Kutuzov, 2020; Martinc et al., 2019). In contrast to this definition, we compute (i) *sense prototypes* $\mu_{w,i}^1, \mu_{w,i}^2$ as the average embedding for each cluster partition $\phi_{w,i}^1, \phi_{w,i}^2$, respectively; and (ii) *word prototypes* M_w^1, M_w^2 as the average embedding of all *sense prototypes* $\mu_{w,i}^1$, and $\mu_{w,i}^2$ respectively. The idea is that computing the average of a smaller set of more significant embeddings, i.e., the *sense prototypes*, can be beneficial to reduce noise in clusters.

The average-based method by Martinc et al. (2019) consists in computing the cosine similarity between the global average embeddings of all embeddings from $t-n$ and t, respectively. We extend this method by computing the cosine distance between M_w^1 and M_w^2.

$$PDIS(M^1, M^2) = 1 - \frac{M^1 \cdot M^2}{\|M^1\| \times \|M^2\|} \quad (3)$$

Difference between prototype embedding diversities (PDIV). The method proposed by Kutuzov (2020) relies on the notion of "embedding diversity" for word prototypes (DIV). We extend this method considering sense prototypes. In particular, we estimate the degree of ambiguity for w in C_1, C_2 as the mean cosine distance d between *sense prototypes* $\mu^j_{w,i}$ and the relative *word prototype* M^j_w. The final result is the absolute difference between the relative coefficients. For the sake of simplicity, let's denote as Ψ^1_w and Ψ^2_w the set of sense prototypes of $\mu^1_{w,i}$, and $\mu^2_{w,i}$ respectively.

$$PDIV(\Psi^1_w, \Psi^2_w) = \left| \frac{\sum_{\mu^1_{w,k} \in \Psi^1_w} d(\mu^1_{w,k}, M^1_w)}{|\Psi^1_w|} - \frac{\sum_{\mu^2_{w,k} \in \Psi^2_w} d(\mu^2_{w,k}, M^2_w)}{|\Psi^2_w|} \right| \quad (4)$$

6 Evaluation of WiDiD

For evaluation of WiDiD, we rely on the Task 1 framework of SemEval-2020. SemEval is a series of international NLP workshops based on a collection of shared tasks in which computational semantic analysis systems designed by different teams are presented and compared. In particular, we focus on SemEval-2020 Subtask 2 where the goal is to consider texts from two distinct time periods and to evaluate the degree of semantic shift of a set of target words (Schlechtweg et al., 2020). In SemEval-2020, the semantic shift degree is measured by the Spearman's rank-order correlation between the semantic shift index (i.e., the ground truth) and the semantic shift assessment computed by a model for each target word in the evaluation set. Our evaluation is performed over the English and Latin corpora of SemEval-2020. A summary view of the considered corpora is provided in Table 1. As proposed in Montariol et al. (2021), in the English corpus, we removed POS tags from both the corpus and the evaluation set.

		Period	Tokens	Corpus	Target Words
SemEval English	C_1	1810 – 1860	6.5M	CCOHA	37
	C_2	1960 – 2010	6.7M		
SemEval Latin	C_1	-200 – 0	65k	LatinISE	40
	C_2	0 – 2000	253k		

Table 1: Period, size, and number of target words for English and Latin corpora of SemEval-2020

6.1 Experimental setup

In the evaluation, the following configurations of WiDiD have been adopted.

Word representations. Pre-trained BERT and trained Doc2Vec models are exploited as embedding models. We use the Transformers library by HuggingFace to extract contextual word embeddings from pre-trained BERT models without performing any fine-tuning stage (Wolf et al., 2020). We use a specific model for each language, namely *bert-base-uncased*[1] for English and *bert-base-multilingual-uncased*[2] for Latin. The models are base versions of BERT with 12 attention layers and a hidden layer of size 768. The only model available for Latin is a multilingual BERT model trained on 104 languages, including Latin.

The acquisition of contextual embeddings is done by feeding the models with text sequences from the corpora in which the target words occur. Sequence embeddings are generated one sequence at a time by summing the last 4 encoder output layers according to Devlin et al. (2019). Finally, given a sequence of size $sequence\ length \times embeddings\ size$, we cut it into pieces to get a separate contextual embedding for each token in the sequence. In this way, we extract token embeddings for each occurrence of a target word in a corpus. Due to the byte-pair input encoding scheme employed by BERT models, some tokens may not correspond to words but rather to word pieces (Sennrich et al., 2016; Wu et al., 2016). Therefore, if a word is split into more than one token, we build a single word embedding by concatenating them.

Pseudo-Word Representations. While BERT-like models generate dynamic embeddings for a word according to their belonging sequences (i.e., documents), Doc2Vec (Le and Mikolov, 2014) produces a static lookup table of word and sequence embeddings only for words and sequences seen during training. We exploit Doc2Vec by computing *pseudo*-contextual word embeddings under the assumption that word occurrences belonging to similar sequences have the same meaning. This means that, given a target word w in the corpus C_j we consider as Φ^j_w the set of sequence embeddings related to sequences where w occurs. For training

[1] https://huggingface.co/bert-base-uncased

[2] https://huggingface.co/bert-base-multilingual-uncased

Doc2Vec models, we use the Gensim library (Rehurek and Sojka, 2011). In particular, we trained word and sequence embeddings of size 100 for 15 epochs, with a window size of 10.

Clustering of embeddings. For the evaluation of WiDiD, we exploit the APP clustering algorithm described in Section 4. Since APP is an extension of the Affinity Propagation (AP) clustering algorithm, we compared the results of APP against the results of AP in the clustering step of the WiDiD approach. In addition, we tested a further incremental extension of AP called IAPNA. IAPNA is an incremental version of AP that has been proposed by Sun and Guo (2014) and it is based on the idea of computing a reasonable assignment for all the data points at the same status. Then, when new points are available, the relationships between the new points and the other points are assigned referring to their nearest neighbors and by updating the responsibility and availability indexes for those points. In particular, we use the scikit-learn (Pedregosa et al., 2011) implementation for standard AP, that we extended for implementing both IAPNA and APP.

Experiments. The following experiments have been executed. We apply the semantic shift measures illustrated in Section 5 (i.e., JSD, PDIS, PDIV) to the clusters of contextual embeddings obtained by using AP, IAPNA, and APP, respectively. Since PDIS and PDIV are extensions of the CD (*Cosine Distance over Word Prototypes*) and DIV (*Difference between Token Embedding Diversities*) measures proposed by Martinc et al. (2019) and Kutuzov (2020), we also consider them as baselines.

6.2 Experimental results

The results of our evaluation are shown in Table 2[3].

Surprisingly, Doc2Vec proved to be a suitable model for semantic shift detection, in both incremental and non-incremental clustering contexts. It performs well, while being smaller and faster than contextual models. In particular, Doc2Vec-based methods achieve the highest result in our experiments on both Latin and English, with correlation coefficient of 0.512 and 0.514, respectively. APP provides top results on both Latin and English, although AP has a slightly higher performance on English.

[3]The source code of our experiments is available under the MIT license at https://github.com/umilISLab/LChange22

On average, both incremental clustering algorithms IAPNA and APP perform well in semantic shift detection compared to the conventional AP clustering. We note that IAPNA and APP have opposite behavior on Latin and English: IAPNA has higher results with BERT embeddings on Latin and Doc2Vec embeddings on English, while APP has higher results with Doc2Vec embeddings on Latin and BERT embeddings on English, respectively. The fact that IAPNA and APP perform differently on different languages is consistent with the literature results (Kutuzov and Giulianelli, 2020).

As a further remark, we note that APP produces a smaller and more reasonable number of clusters compared to both AP and IAPNA. For instance, we observed situations where both AP and IAPNA produce more than 100 clusters, that is rather unrealistic if we assume that a cluster represents a word meaning. On the opposite, in our experiments, the number of APP clusters generally varies between 0 and 30. We also note that APP is sensitive to the aging index. In Table 2, we present the top results obtained with two different values of the aging index (i.e., 0 and 5). Removing clusters containing less than 5% of the embeddings has a positive impact just in some experiments with English, but not with Latin. We plan to further investigate the effects of the aging index in our future work.

About our proposed measures for semantic shift detection (i.e., JSD, PDIS, PDIV), we note that they always perform better than the baselines CD and DIV. We also note that the CD baseline does not work well on Doc2Vec embeddings, while DIV does not work well in all our experiments. On Latin, the highest results are achieved by JSD on both Doc2Vec and BERT embeddings. On English, the top JSD and PDIS results are on Doc2Vec and BERT embeddings, respectively. More experiments are required on PDIV since it performs very differently in the various experiments we performed, and it achieves statistical significance only in four out of twelve experiments (six on Latin, six on English).

Finally, Table 3 provides the best results obtained by other literature approaches for semantic shift detection based on contextual word embeddings over the English and Latin corpora of SemEval-2020. We note that both IAPNA and APP are competitive when compared to the considered literature approaches. The WiDiD scores are above average and slightly below the maxi-

Clustering	Training	Model	Latin (Spearman's coefficients)			English (Spearman's coefficients)		
			JSD	*PDIS*	*PDIV*	*JSD*	*PDIS*	*PDIV*
AP	trained	Doc2Vec	**0.485***	0.229	-0.023	**0.514***	0.139	0.134
	pre-trained	BERT	**0.394***	0.347*	0.236	0.356*	0.326*	**0.406***
IAPNA	trained	Doc2Vec	**0.462***	0.354*	-0.005	0.199	0.322*	**0.336***
	pre-trained	BERT	**0.411***	0.356*	-0.148	0.336*	**0.499***	0.213
APP	trained	Doc2Vec	**0.512_0***	0.337_0*	0.328_0*	**0.333_0***	0.077_0	-0.078_0
	pre-trained	BERT	**0.361_0***	0.210_0	0.036_0	0.302_0°	**0.512_5***	0.370_5*
			CD	*DIV*		*CD*	*DIV*	
	trained	Doc2Vec	0.258°	0.138	-	0.092	0.010	-
	pre-trained	BERT	0.306*	-0.017	-	0.486*	0.168	-

Table 2: Spearman's correlation coefficients over different setups with Latin and English corpora. The asterisks denote statistically significant correlations ($p \leq 0.05$), while degree symbols denote low-level correlations with ($0.05 \leq p \leq 0.1$). The subscript index indicates the value adopted for the aging index. We report in bold the highest scores for each clustering-based method considering BERT and Doc2Vec.

	Clustering	Training	Model	Latin (Spearman's coeff.)	English (Spearman's coeff.)
Beck, 2020	-	pre-trained	BERT	0.343	0.293
Karnysheva and Schwarz, 2020	K-means (English) DBSCAN (Latin)	pre-trained	ELMo	0.177	-0.155
Cuba Gyllensten et al., 2020	K-Means	**pre-trained**	XLM-R	**0.399**	0.209
Rother et al., 2020	HDBSCAN (English) GMMs (Latin)	pre-trained	BERT	0.321	0.512
Kanjirangat et al., 2020	K-means	pre-trained	BERT	0.333	0.159
Laicher et al., 2021	-	**pre-trained**	BERT	N/D	**0.571**
Arefyev and Zhikov, 2020	-	fine-tuned	XLM-R	-0.134	0.299
Kutuzov and Giulianelli, 2020	-	**fine-tuned**	ELMo (English) BERT (Latin)	**0.561**	**0.605**
Montariol et al., 2021	AP	fine-tuned	BERT	0.496	0.456
Pömsl and Lyapin, 2020	-	fine-tuned	BERT	0.464	0.246
Rosin et al., 2021	-	fine-tuned	TinyBERT (English) LatinBERT (Latin)	0.512	0.467
Martinc et al., 2020b	AP	fine-tuned	BERT	0.496	0.436
Liu et al., 2021	-	fine-tuned	BERT	0.304	0.341

Table 3: Spearman's correlation coefficients obtained by different experiments with English and Latin corpora. We report in bold the best scores for pre-trained and fine-tuned models. The hyphens indicate approaches that do not cluster contextual embeddings. N/D indicates that experimental results are not available.

mum scores (in bold). We stress that we obtained these results without fine-tuning, confirming that the idea of using incremental clustering is promising. Compared to other literature approaches based on pre-trained models without fine-tuning, we note that incremental clustering algorithms achieve the highest scores on Latin (0.512 with APP and 0.411 with IAPNA for Doc2Vec and BERT, respectively). Our results on the English corpus come second in the pre-trained ranking (0.512 with APP and 0.499 with IAPNA for Doc2Vec and BERT, respectively) after Laicher et al. (2021). All in all, excluding Laicher et al. (2021) and Kutuzov (2020), our results are the highest of all the considered literature works of Table 3, both on Latin and English.

7 Concluding remarks

In this paper, we presented the WiDiD approach characterized by incremental clustering techniques and contextual word embedding methods. Ongoing work is about the fine-tuning of adopted embedding models to further improve the quality of results. Moreover, we are working on defining cluster analysis techniques. The idea is to exploit the results of semantic shift measures to interpret possible trend patterns over clusters along the time, such as a broad meaning that forks into narrower ones, or a meaning that increases its popularity and vice versa. Further work is about the specification of aging policies to manage the memory of aged embeddings in the cluster evolution.

Acknowledgments

This paper is partially funded by the RECON project within the UNIMI-SEED research programme and by the PSR-UNIMI programme.

References

Nikolay Arefyev and Vasily Zhikov. 2020. BOS at SemEval-2020 Task 1: Word Sense Induction via Lexical Substitution for Lexical Semantic Change Detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 171–179, Barcelona (online). International Committee for Computational Linguistics.

Christin Beck. 2020. DiaSense at SemEval-2020 Task 1: Modeling Sense Change via Pre-trained BERT Embeddings. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 50–58, Barcelona (online). International Committee for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Amaru Cuba Gyllensten, Evangelia Gogoulou, Ariel Ekgren, and Magnus Sahlgren. 2020. SenseCluster at SemEval-2020 Task 1: Unsupervised Lexical Semantic Change Detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 112–118, Barcelona (online). International Committee for Computational Linguistics.

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.

Tony Finch. 2009. Incremental Calculation of Weighted Mean and Variance. *University of Cambridge*, 4(11-5):41–42.

Brendan J Frey and Delbert Dueck. 2007. Clustering by passing messages between data points. *science*, 315(5814):972–976.

Mario Giulianelli. 2019. *Lexical Semantic Change Analysis with Contextualised Word Representations*. Ph.D. thesis, University of Amsterdam - Institute for logic, Language and computation.

Mario Giulianelli, Marco Del Tredici, and Raquel Fern'andez. 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.

William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2016. Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1489–1501, Berlin, Germany. Association for Computational Linguistics.

Franziska Horn. 2021. Exploring Word Usage Change with Continuously Evolving Embeddings. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 290–297, Online. Association for Computational Linguistics.

Renfen Hu, Shen Li, and Shichen Liang. 2019. Diachronic Sense Modeling with Deep Contextualized Word Embeddings: An Ecological View. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3899–3908, Florence, Italy. Association for Computational Linguistics.

Vani Kanjirangat, Sandra Mitrovic, Alessandro Antonucci, and Fabio Rinaldi. 2020. SST-BERT at SemEval-2020 Task 1: Semantic Shift Tracing by Clustering in BERT-based Embedding Spaces. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 214–221, Barcelona (online). International Committee for Computational Linguistics.

Anna Karnysheva and Pia Schwarz. 2020. TUE at SemEval-2020 Task 1: Detecting Semantic Change by Clustering Contextual Word Embeddings. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 232–238, Barcelona (online). International Committee for Computational Linguistics.

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, Baltimore, MD, USA. Association for Computational Linguistics.

Andrey Kutuzov. 2020. *Distributional Word Embeddings in Modeling Diachronic Semantic Change*. Ph.D. thesis, University of Oslo.

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, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Severin Laicher, Sinan Kurtyigit, Dominik Schlechtweg, Jonas Kuhn, and Sabine Schulte im Walde. 2021. Explaining and Improving BERT Performance on Lexical Semantic Change Detection. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 192–202, Online. Association for Computational Linguistics.

Quoc Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1188–1196, Bejing, China. PMLR.

Yang Liu, Alan Medlar, and Dorota Glowacka. 2021. Statistically significant detection of semantic shifts using contextual word embeddings. *CoRR*, abs/2104.03776.

Matej Martinc, Petra Kralj Novak, and Senja Pollak. 2019. Leveraging Contextual Embeddings for Detecting Diachronic Semantic Shift. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4811–4819, Marseille, France. European Language Resources Association.

Matej Martinc, Syrielle Montariol, Elaine Zosa, and Lidia Pivovarova. 2020a. Capturing Evolution in Word Usage: Just Add More Clusters? *CoRR*, abs/2001.06629.

Matej Martinc, Syrielle Montariol, Elaine Zosa, and Lidia Pivovarova. 2020b. Discovery Team at SemEval-2020 Task 1: Context-sensitive Embeddings Not Always Better than Static for Semantic Change Detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 67–73, Barcelona (online). International Committee for Computational Linguistics.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Syrielle Montariol, Matej Martinc, and Lidia Pivovarova. 2021. Scalable and Interpretable Semantic Change Detection. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4642–4652, Online. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Matthew E. 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.

Martin Pömsl and Roman Lyapin. 2020. CIRCE at SemEval-2020 Task 1: Ensembling Context-Free and Context-Dependent Word Representations. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 180–186, Barcelona (online). International Committee for Computational Linguistics.

Bhavani Raskutti and Christopher Leckie. 1999. An evaluation of criteria for measuring the quality of clusters. In *IJCAI: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, volume 99, pages 905–910.

Radim Rehurek and Petr Sojka. 2011. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2).

Julia Rodina, Yuliya Trofimova, Andrey Kutuzov, and Ekaterina Artemova. 2020. ELMo and BERT in Semantic Change Detection for Russian. *CoRR*, abs/2010.03481.

Guy D. Rosin, Ido Guy, and Kira Radinsky. 2021. Time masking for temporal language models. *CoRR*, abs/2110.06366.

David Rother, Thomas Haider, and Steffen Eger. 2020. CMCE at SemEval-2020 Task 1: Clustering on Manifolds of Contextualized Embeddings to Detect Historical Meaning Shifts. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 187–193, Barcelona (online). International Committee 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.

Peter H Schönemann. 1966. A Generalized Solution of the Orthogonal Procrustes Problem. *Psychometrika*, 31(1):1–10.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Leilei Sun and Chonghui Guo. 2014. Incremental Affinity Propagation Clustering Based on Message Passing. *IEEE Transactions on Knowledge and Data Engineering*, 26(11):2731–2744.

Nina Tahmasebi, Lars Borin, and Adam Jatowt. 2018. Survey of Computational Approaches to Lexical Semantic Change. *ArXiv e-prints*, abs/1811.06278.

Nina Tahmasebi, Lars Borin, Adam Jatowt, Yang Xu, and Simon Hengchen, editors. 2021. *Computational Approaches to Semantic Change*. Number 6 in Language Variation. Language Science Press, Berlin.

Laura Wendlandt, Jonathan K. Kummerfeld, and Rada Mihalcea. 2018. Factors Influencing the Surprising Instability of Word Embeddings. 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 2092–2102, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *CoRR*, abs/1609.08144.

Kaitlyn Zhou, Kawin Ethayarajh, and Dan Jurafsky. 2021. Frequency-based Distortions in Contextualized Word Embeddings. *arXiv preprint arXiv:2104.08465*.