Back to the Future

Sequential Alignment of Text Representations

Bjerva, Johannes; Kouw, Wouter M. ; Augenstein, Isabelle

Published in:
Proceedings of the 34th AAAI Conference on Artificial Intelligence

Publication date:
2020

Document version
Peer reviewed version

Back to the Future – Sequential Alignment of Text Representations

Johannes Bjerva,[1*] **Wouter M. Kouw,**[1,2*] **Isabelle Augenstein**[1]
[1]Department of Computer Science, University of Copenhagen, Copenhagen, Denmark
[2]Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, the Netherlands
bjerva@di.ku.dk, w.m.kouw@tue.nl, augenstein@di.ku.dk

Abstract

Language evolves over time in many ways relevant to natural language processing tasks. For example, recent occurrences of tokens 'BERT' and 'ELMO' in publications refer to neural network architectures rather than persons. This type of temporal signal is typically overlooked, but is important if one aims to deploy a machine learning model over an extended period of time. In particular, language evolution causes data drift between time-steps in sequential decision-making tasks. Examples of such tasks include prediction of paper acceptance for yearly conferences (regular intervals) or author stance prediction for rumours on Twitter (irregular intervals). Inspired by successes in computer vision, we tackle data drift by sequentially aligning learned representations. We evaluate on three challenging tasks varying in terms of time-scales, linguistic units, and domains. These tasks show our method outperforming several strong baselines, including using all available data. We argue that, due to its low computational expense, sequential alignment is a practical solution to dealing with language evolution.

Introduction

As time passes, language usage changes. For example, the names 'Bert' and 'Elmo' would only rarely make an appearance prior to 2018 in the context of scientific writing. After the publication of BERT (Devlin et al. 2018) and ELMo (Peters et al. 2018), however, usage has increased in frequency. In the context of named entities on Twitter, it is also likely that these names would be tagged as PERSON prior to 2018, and are now more likely to refer to an ARTEFACT. As such, their part-of-speech tags will also differ. Evidently, evolution of language usage affects natural language processing (NLP) tasks, and as such, models based on data from one point in time cannot be expected to generalise to the future.

In order to become more robust to language evolution, data should be collected at multiple points in time. We consider a dynamic learning paradigm where one makes predictions for data points from the current time-step given labelled data points from previous time-steps. As time increments, data points from the current step are labelled and new unlabelled data points are observed. This setting occurs in NLP in, for instance, the prediction of paper acceptance to conferences (Kang et al. 2018) or named entity recognition from yearly data dumps of Twitter (Derczynski, Bontcheva, and Roberts 2016). Changes in language usage cause a data drift between time-steps and some way of controlling for the shift between time-steps is necessary.

In this paper, we apply a domain adaptation technique to correct for shifts. Domain adaptation is a furtive area of research within machine learning that deals with learning from training data drawn from one data-generating distribution (source domain) and generalising to test data drawn from another, different data-generating distribution (target domain) (Kouw and Loog 2019). We are interested in whether a sequence of adaptations can compensate for the data drift caused by shifts in the meaning of words or features across time. Given that linguistic tokens are embedded in some vector space using neural language models, we observe that in time-varying dynamic tasks, the drift causes token embeddings to occupy different parts of embedding space over consecutive time-steps. We want to avoid the computational expense of re-training a neural network every time-step. Instead, in each time-step, we map linguistic tokens using the same pre-trained language model (a "BERT" network (Devlin et al. 2018)) and align the resulting embeddings using a second procedure called subspace alignment (Fernando et al. 2013). We apply subspace alignment sequentially: find the principal components in each time-step and linearly transform the components from the previous step to match the current step. A classifier trained on the aligned embeddings from the previous step will be more suited to classify embeddings in the current step. We show that sequential subspace alignment (SSA) yields substantial improvements in three challenging tasks: paper acceptance prediction on the PeerRead data set (Kang et al. 2018); Named Entity Recognition on the Broad Twitter Corpus (Derczynski, Bontcheva, and Roberts 2016); and rumour stance detection on the RumourEval 2019 data set (Gorrell et al. 2018). These tasks are chosen to vary in terms of domains, timescales, and the granularity of the linguistic units. In addition to evaluating SSA, we include two technical contributions as we extend the method both to allow for time series of unbounded

*JB and WMK contributed equally to this work.

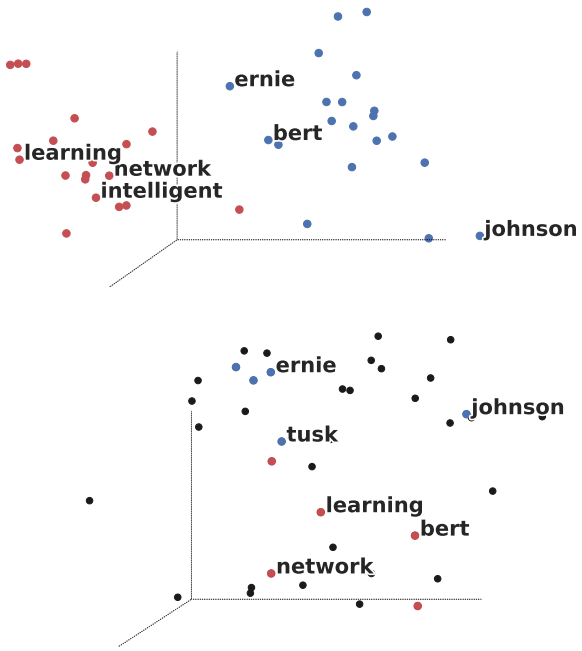

Figure 1: Example of a word embedding at t_{2017} vs t_{2018} (blue=PERSON, red=ARTEFACT, black=UNK). Source data (top, t_{2017}), target data (bottom, t_{2018}). Note that at t_{2017}, 'bert' is a PERSON, while at t_{2018}, 'bert' is an ARTEFACT.

length and to consider instance similarities between classes. The best-performing SSA methods proposed here are semi-supervised, but require only between 2 and 10 annotated data points per class from the test year for successful alignment. Crucially, the best proposed SSA models outperform baselines utilising more data, including the whole data set.

Subspace Alignment

Suppose we embed words from a named entity recognition task, where ARTEFACTs should be distinguished from PERSONs. Figure 1 shows scatterplots with data collected at two different time-points, say 2017 (top; source domain) and 2018 (bottom; target domain). Red points are examples of ARTEFACTs embedded in this space and blue points are examples of PERSONs. We wish to classify the unknown points (black) from 2018 using the labeled points (red/blue bottom) from 2018 and the labeled points from 2017 (red/blue top).

As can be seen, the data from 2017 is not particularly relevant to classification of data from 2018, because the red and blue point clouds do not match. In other words, a classifier trained to discriminate red from blue in 2017 would make a lot of mistakes when applied directly to the data from 2018, partly because words such as 'bert' have changed from being PERSONs to being ARTEFACTs. To make the source data from 2017 relevant – and reap the benefits of having more data – we wish to *align* source and target data points.

Unsupervised Subspace Alignment

Unsupervised alignment extracts a set of bases from each data set and transforms the source components such that they match the target components (Fernando et al. 2013). Let $C_{\mathcal{S}}$ be the principal components of the source data X_{t-1} and $C_{\mathcal{T}}$ be the components of the target data set X_t. The optimal linear transformation matrix is found by minimising the difference between the transformed source components and the target components:

$$\begin{aligned} M^* &= \arg\min_M \|C_{\mathcal{S}} M - C_{\mathcal{T}}\|_F^2 \\ &= \arg\min_M \|C_{\mathcal{S}}^\top C_{\mathcal{S}} M - C_{\mathcal{S}}^\top C_{\mathcal{T}}\|_F^2 \\ &= \arg\min_M \|M - C_{\mathcal{S}}^\top C_{\mathcal{T}}\|_F^2 \;=\; C_{\mathcal{S}}^\top C_{\mathcal{T}}\,, \end{aligned} \quad (1)$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Note that we left-multiplied both terms in the norm with the same matrix $C_{\mathcal{S}}^\top$ and that due to orthonormality of the principal components, $C_{\mathcal{S}}^\top C_{\mathcal{S}}$ is the identity and drops out. Source data X_{t-1} is aligned to target data by first mapping it onto its own principal components and then applying the transformation matrix, $X_{t-1} C_{\mathcal{S}} M^*$. Target data X_t is also projected onto its target components, $X_t C_{\mathcal{T}}$. The alignment is performed on the d largest principal components, i.e. a *subspace* of the embedding. Keeping d small avoids the high computational expense of eigendecomposition in high-dimensional data.

Unsupervised alignment will only match the total structure of both data sets. Therefore, global shifts between domains can be accounted for, but not local shifts. Figure 1 is an example of a setting with local shifts, i.e. red and blue classes are shifted differently. Performing unsupervised alignment on this setting would fail. Figure 2 (left middle) shows the source data (leftmost) aligned to the target data (rightmost) in an unsupervised fashion. Note that although the total data sets roughly match, the classes (red and blue ellipses) are not matched.

Semi-Supervised Subspace Alignment

In semi-supervised alignment, one performs subspace alignment *per class*. As such, at least 1 target label per class needs to be available. However, even then, with only 1 target label per class, we would only be able to find 1 principal component. To allow for the estimation of more components, we provisionally label all target samples using a 1-nearest-neighbour classifier, starting from the given target labels. Using pseudo-labelled target samples, we estimate d components.

Now, the optimal linear transformation matrix for each class can be found with an equivalent procedure as in Equation 1:

$$M_k^* \;=\; \arg\min_M \|C_{\mathcal{S},k} M - C_{\mathcal{T},k}\|_F^2 \;=\; C_{\mathcal{S},k}^\top C_{\mathcal{T},k}\,. \quad (2)$$

Afterwards, we transform the source samples of each class X_{t-1}^k through the projection onto class-specific components $C_{\mathcal{S},k}$ and the optimal transformation: $X_{t-1}^k C_{\mathcal{S},k} M_k^*$. Additionally, we centre each transformed source class on the corresponding target class. Figure 2 (right middle) shows

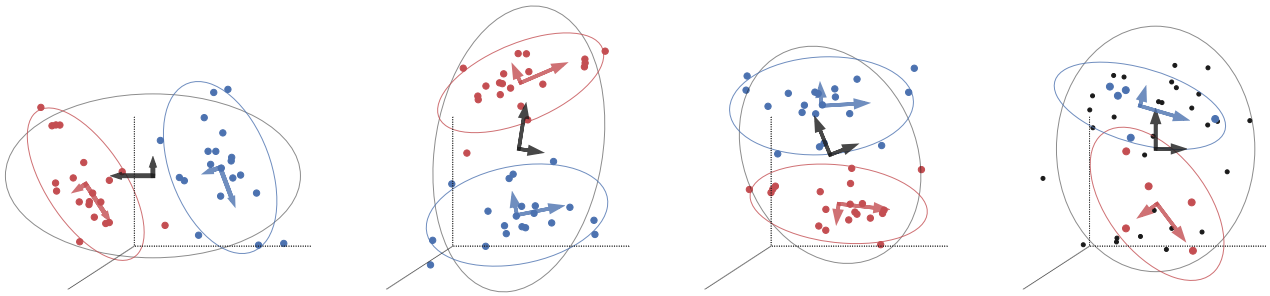

Figure 2: Illustration of subspace alignment procedures. Red vs blue dots indicate samples from different classes, arrows (black for total data and red vs blue for each class) indicate scaled eigenvectors of the covariance matrix (error ellipses indicate regions within 2 standard deviations). (Leftmost) Source data, fully labeled. (Left middle). Unsupervised subspace alignment: the total principal components from the source data (black arrows in leftmost figure) have been aligned to the total principal components of the target data (black arrows in rightmost figure). (Right middle) Semi-supervised subspace alignment: the class-specific principal components of the source data (red/blue arrows from leftmost figure) have been aligned to the class-specific components of the target data (red/blue arrows from the rightmost figure). Note that unsupervised alignment fails to match the red and blue classes across domains, while semi-supervised alignment succeeds. (Rightmost) Target data, with few labeled samples per class (black dots are unlabeled samples).

the source documents transformed through semi-supervised alignment. Now, the classes match the target data classes.

Extending SSA to Unbounded Time

Semi-supervised alignment allows for aligning *two* time steps, t_1 and t_2, to a joint space $t'_{1,2}$. However, when considering a further alignment to another time step t_3, this can not trivially be mapped, since the joint space $t'_{1,2}$ necessarily has a lower dimensionality. Observing that two independently aligned spaces, $t'_{1,2}$ and $t'_{2,3}$, *do* have the same dimensionality, we further learn a new alignment between the two, resulting in the joint space of $t'_{1,2}$ and $t'_{2,3}$, namely $t''_{1,2,3}$. While there are many ways of joining individual time steps to a single joint space, we approach this by building a binary branching tree, first joining adjacent timesteps with each other, and then joining the new adjacent subspaces with each other.

Although this is seemingly straight-forward, there is no guarantee that $t'_{1,2}$ and $t'_{2,3}$ will be coherent with one another, in the same way that two word embedding spaces trained with different algorithms might also differ in spite of having the same dimensionality. This issue is partially taken care of by using semi-supervised alignment which takes class labels into account when learning the 'deeper' alignment t''. We further find that it is beneficial to also take the similarities between samples into account when aligning.

Considering Sample Similarities between Classes

Since intermediary spaces, such as $t'_{1,2}$ and $t'_{2,3}$, do not necessarily share the same semantic properties, we add a step to the semi-supervised alignment procedure. Given that the initial unaligned spaces do encode similarities between instances, we run the k-means clustering algorithm ($k = 5$) to give us some course-grained indication of instance similarities in the original embedding space. This cluster ID is passed to SSA, resulting in an alignment which both attempts to match classes across time steps, in addition to instance similarities. Hence, even though $t'_{1,2}$ and $t'_{2,3}$ are not necessarily semantically coherent, an alignment to $t''_{1,2,3}$ is made possible.

Experimental Setup

In the past year, several approaches to pre-training representations on language modelling based on transformer architectures (Vaswani et al. 2017) have been proposed. These models essentially use a multi-head self-attention mechanism in order to learn representations which are able to attend directly to any part of a sequence. Recent work has shown that such contextualised representations pre-trained on language modelling tasks offer highly versatile representations which can be fine-tuned on seemingly any given task (Peters et al. 2018; Devlin et al. 2018; Radford et al. 2018; 2019). In line with the recommendations from experiments on fine-tuning representations (Peters, Ruder, and Smith 2019), we use a frozen BERT to extract a consistent task-agnostic representation. Using a frozen BERT with subsequent subspace alignment allows us to avoid re-training a neural network each time-step while still working in an embedding learned by a neural language model. It also allows us to test the effectiveness of SSA without the confounding influence of representation updates.

Three Tasks. We consider three tasks representing a broad selection of natural language understanding scenarios: paper acceptance prediction based on the PeerRead data set (Kang et al. 2018), Named Entity Recognition (NER) based on the Broad Twitter Corpus (Derczynski, Bontcheva, and Roberts 2016), and author stance prediction based on the RumEval-19 data set (Gorrell et al. 2018). These tasks were chosen so as to represent i) different textual domains, across ii) differ-

ing time scales, and iii) operating at varying levels of linguistic granularity. As we are dealing with dynamical learning, the vast majority of NLP data sets can unfortunately not be used since they do not include time stamps.

Paper Acceptance Prediction

The PeerRead data set contains papers from ten years of arXiv history, as well as papers and reviews from major AI and NLP conferences (Kang et al. 2018).[1] From the perspective of evaluating our method, the arXiv sub-set of this data set offers the possibility of evaluating our method while adapting to ten years of history. This is furthermore the only subset of the data annotated with both timestamps and with a relatively balanced accept/reject annotation.[2] As arXiv naturally contains both accepted and rejected papers, this acceptance status has been assigned based on Sutton and Gong who match arXiv submissions to bibliographic entries in DBLP, and additionally defining acceptance as having been accepted to major conferences, and not to workshops. This results in a data set of nearly 12,000 papers, from which we use the raw abstract text as input to our system. The first three years were filtered out due to containing very few papers. We use the standard train/test splits supplied with the data set.

Kang et al. show that it is possible to predict paper acceptance status at major conferences at above baseline levels. Our intuition in applying SSA to this problem, is that the topic of a paper is likely to bias acceptance to certain conferences *across time*. For instance, it is plausible that the likelihood of a neural paper being accepted to an NLP conference before and after 2013 differs wildly. Hence, we expect that our model will, to some extent, represent the topic of an article, and that this will lend itself nicely to SSA.

Model

We use the pre-trained BERT-BASE-UNCASED model as the base for our paper acceptance prediction model. Following the approach of Devlin et al., we take the final hidden state (i.e., the output of the transformer) corresponding to the special `[CLS]` token of an input sequence to be our representation of a paper, as this has aggregated information through the sequence (Figure 3). This gives us a d-dimensional representation of each document, where $d = 786$. In all of the experiments for this task, we train an SVM with an RBF kernel on these representations, either with or without SSA depending on the setting.

Experiments & Results

We set up a series of experiments where we observe past data, and evaluate on present data. We compare both unsupervised and semi-supervised subspace alignment, with several strong baselines. The baselines represent cases in which we have access to more data, and consist of training our model on either **all** data (i.e. both past and future data), on the **same** year as the evaluation year, and on the **previous** year. In our alignment settings, we only observe data from the previous year, and apply subspace alignment. This is a different task than presented by Kang et al., as we evaluate paper acceptance for papers in the present. Hence, our scores are not directly comparable to theirs.

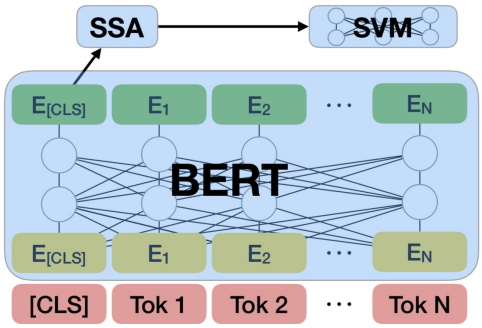

Figure 3: Paper acceptance model (BERT and SSA).

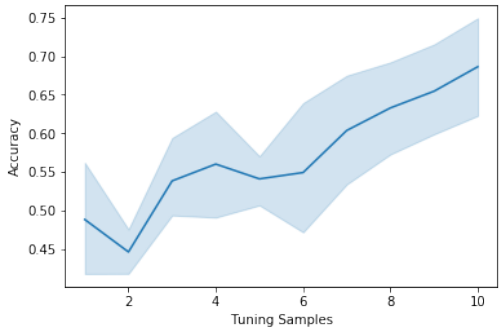

Figure 4: Tuning semi-supervised subspace alignment on PeerRead development data (95% CI shaded).

One parameter which significantly influences performance, is the number of labelled data points we use for learning the semi-supervised subspace alignment. We tuned this hyperparameter on the development set, finding an increasing trend. Using as few as 2 tuning points per class yielded an increase in performance in some cases (Figure 4).

Our results are shown in Table 1, using 10 tuning samples per class. With unsupervised subspace alignment, we observe relatively unstable results – in one exceptional case, namely testing on 2010, unsupervised alignment is as helpful as semi-supervised alignment. Semi-supervised alignment, however, yields consistent improvements in performance across the board. It is especially promising that adapting from past data outperforms training on all available data, as well as training on the actual in-domain data. This highlights the importance of controlling for data drift due to language evolution. It shows that this signal can be taken advantage of to increase performance on present data with only a small amount of annotated data. We further find that using several past time steps in the Unbounded condition is generally helpful, as is using instance similarities in the alignment.

[1] https://github.com/allenai/PeerRead

[2] The NIPS selection, ranging from 2013-2017, only contains accepted papers. The other conferences contain accept/reject annotation, but only represent single years.

Test year	All	Same	Prev	Unsup.	Semi-sup.	Unsup. Unb.	S. Unb.	S. Unb. w/Clst
2010	61.77	67.64	35.29	**70.59**	**70.59**	70.58	**70.59**	**70.59**
2011	61.77	58.82	55.88	14.71	**72.35**	24.71	**72.35**	**72.35**
2012	56.25	56.25	58.75	50.00	72.50	45.00	**72.80**	72.30
2013	67.54	56.14	58.78	76.31	78.07	72.31	78.97	**79.03**
2014	50.53	51.64	51.64	36.88	68.03	31.88	69.03	**69.45**
2015	57.83	54.05	54.05	49.19	58.37	41.19	**59.97**	59.93
2016	58.89	57.36	57.36	50.61	61.04	38.61	**63.04**	**63.04**
2017	56.04	58.24	58.24	68.13	63.73	58.13	68.73	**69.80**
avg	58.82	57.52	53.75	52.05	68.09	47.80	69.44	**69.56**

Table 1: Paper acceptance prediction (acc.) on the PeerRead data set (Kang et al. 2018). Abbreviations represent Unsupervised, Semi-supervised, Unsupervised Unbounded, Semi-supervised Unbounded, and Semi-supervised Unbounded with Clustering.

Named Entity Recognition

The Broad Twitter Corpus contains tweets annotated with named entities, collected between the years 2009 and 2014 (Derczynski, Bontcheva, and Roberts 2016). However, as only a handful of tweets are collected before 2012, we focus our analysis on the final three years of this period (i.e. two test years). The corpus includes diverse data, annotated in part via crowdsourcing and in part by experts. The inventory of tags in their tag scheme is relatively small, including Person, Location, and Organisation. To the best of our knowledge no one has evaluated on this corpus either in general or per year, and so we cannot compare with previous work.

In the case of NER, we expect the adaptation step of our model to capture the fact that named entities may change their meaning across time (e.g. the example with "Bert" and "BERT" in Figure 1). This is related to work showing temporal drift of topics (Wang and McCallum 2006).

Model

Since casing is typically an important feature in NER, we use the pre-trained BERT-BASE-CASED model as our base for NER. For each token, we extract its contextualised representation from BERT, before applying SSA. As Devlin et al. achieve state-of-the-art results without conditioning the predicted tag sequence on surrounding tags (as would be the case with a CRF, for example), we also opt for this simpler architecture. The resulting contextualised representations are therefore passed to an MLP with a single hidden layer (200 hidden units, ReLU activation), before predicting NER tags. We train the MLP over 5 epochs using the Adam optimiser (Kingma and Ba 2014).

Experiments & Results

As with previous experiments, we compare unsupervised and semi-supervised subspace alignment with baselines corresponding to using all data, data from the same year as the evaluation year, and data from the previous year. For each year, we divide the data into 80/10/10 splits for training, development, and test. Results on the two test years 2013 and 2014 are shown in Table 2. In the case of NER, we do not observe any positive results for unsupervised subspace alignment. In the case of semi-supervised alignment, however, we find increased performance as compared to training on the previous year, and compared to training on all data. This shows that learning an alignment from just a few data points can help the model to generalise from past data. However, unlike our previous experiments, results are somewhat better when given access to the entire set of training data from the test year itself in the case of NER. The fact that training on only 2013 and evaluating on the same year does not work well can be explained by the fact that the amount of data available for 2013 is only 10% of that for 2012. The identical results for the unbounded extension is because aligning from a single time step renders this irrelevant.

SDQC Stance Classification

The RumourEval-2019 data set consists of roughly 5500 tweets collected for 8 events surrounding well-known incidents, such as the Charlie Hebdo shooting in Paris (Gorrell et al. 2018).[3] Since the shared task test set is not available, we split the training set into a training, dev and test part based on rumours (one rumour will be training data with a 90/10 split for development and another rumour will be the test data, with a few samples labelled). For Subtask A, tweets are annotated with stances, denoting whether it is in the category `Support`, `Deny`, `Query`, or `Comment` (SDQC).

Each rumour only lasts a couple of days, but the total data set spans years, from August 2014 to November 2016. We regard each rumour as a time-step and adapt from the rumour at time t-1 to the rumour at time t. We note that this setting is more difficult than the previous two due to the irregular time intervals. We disregard the rumour `ebola-essien` as it has too few samples per class.

Model

For this task, we use the same modelling approach as described for paper acceptance prediction. This method is also suitable here, since we simply require a condensed representation of a few sentences on which to base our temporal adaptation and predictions. In the last iteration of the task, the winning system used hand-crafted features to achieve a high performance (Kochkina, Liakata, and Augenstein 2017). Including these would complicate SSA, so we opt for

[3] http://alt.qcri.org/semeval2019/index.php?id=tasks

Test year	All	Same	Prev	Unsup.	Semi-sup.	Unsup. Unb.	S. Unb.	S. Unb. w/Clst
2013	62.95	42.24	54.16	42.25	63.82	42.25	63.82	**63.95**
2014	72.77	77.76	59.53	50.43	73.67	50.43	73.67	**78.75**
avg	67.86	60.00	56.85	46.34	68.75	46.34	68.75	**71.35**

Table 2: NER (F1 score) on the Broad Twitter Corpus (Derczynski, Bontcheva, and Roberts 2016).

this simpler architecture instead. We use the shorter timescale of approximately weeks rather than years as rumours can change rapidly (Kwon, Cha, and Jung 2017).

Experiments & Results

In this experiment, we start with the earliest rumour and adapt to the next rumour in time. As before, we run the following baselines: training on all available labelled data (i.e. all previous rumours and the labelled data for the current rumour), training on the labelled data from the current rumour (designated as 'same') and training on the labelled data from the previous rumour. We perform both unsupervised and semi-supervised alignment using data from the previous rumour. We label 5 samples per class for each rumour.

In this data set, there is a large class imbalance, with a large majority of `comment` tweets and few `support` or `deny` tweets. To address this, we over-sample the minority classes. Afterwards, a SVM with RBF is trained and we test on unlabelled tweets for the current rumour. Table 3 shows the performance of the baselines and the two alignment procedures. As with the previous tasks, semi-supervised alignment generally helps, except for in the `charliehebdo` rumour.

Analysis and Discussion

We have shown that sequential subspace alignment is useful across natural language processing tasks. For the PeerRead data set we were particularly successful. This might be explained by the fact that the topic of a paper is a simple feature for SSA to pick up on, while being predictive of a paper's acceptance chances. For NER, on the other hand, named entities can change in less predictable ways across time, proving a larger challenge for our approach. For SDQC, we were successful in cases where the tweets are nicely clustered by class. For instance, where both rumours are about terrorist attacks, many of the `support` tweets were headlines from reputable newspaper agencies. These agencies structure tweets in a way that is consistently dissimilar from `comments` and `queries`.

The effect of our unbounded time extension boosts results on the PeerRead data set, as the data stretches across a range of years. In the case of NER, however, this extension is excessive as only two time steps are available. In the case of SDQC, the lack of improvement could be due to the irregular time intervals, making it hard to learn consistent mappings from rumour to rumour. Adding instance similarity clustering aids alignment, since considering sample similarities across classes is important over longer time scales.

Example of Aligning Tweets

Finally, we set up the following simplified experiment to investigate the effect of alignment on SDQC data. First, we consider the rumour `charliehebdo`, where we picked the following tweet:

Support:
`France: 10 people dead after shooting at HQ of satirical weekly newspaper #CharlieHebdo, according to witnesses <URL>`

It has been labeled to be in support of the veracity of the rumour. We will consider the scenario where we use this tweet and others involving the `charliehebdo` incident to predict author stance in the rumour `germanwings-crash`. Before alignment, the following 2 `germanwings-crash` tweets are among the nearest neighbours in the embedding space:

Query:
`@USER @USER if they had, its likely the descent rate wouldve been steeper and the speed not reduce, no ?`

Comment:
`@USER Praying for the families and friends of those involved in crash. I'm so sorry for your loss.`

The second tweet is semantically similar (both are on the topic of tragedy), but the other is unrelated. Note that the news agency tweet differs from the comment and query tweets in that it stems from a reputable source, mentions details and includes a reference. After alignment, the `charliehebdo` tweet has the following 2 nearest neighbours:

Support:
`@USER: 148 passengers were on board #GermanWings Airbus A320 which has crashed in the southern French Alps <URL>`

Support:
`Report: Co-Pilot Locked Out Of Cockpit Before Fatal Plane Crash <URL> #Germanwings <URL>`

Now, both neighbours are of the `support` class. This example shows that semi-supervised alignment maps source tweets from one class close to target tweets of the same class.

Test year	All	Same	Prev	Unsup.	Semi-sup.	Unsup. Unb.	S. Unb.	S. Unb. w/Clst
ottawashooting	31.51	23.67	30.77	30.77	**31.88**	28.37	30.68	30.88
prince-toronto	36.27	23.37	34.46	34.46	**40.32**	31.36	39.12	39.52
sydney-siege	32.34	27.17	41.23	41.23	**43.60**	33.23	43.50	43.54
charliehebdo	38.51	31.67	**35.73**	**35.73**	33.76	33.71	32.70	32.61
putinmissing	28.33	22.38	34.53	34.53	**36.11**	31.95	35.10	35.81
germanwings-crash	29.38	22.01	44.79	44.79	**44.84**	40.30	44.88	44.80
illary	29.24	25.81	37.53	37.53	**40.08**	34.10	39.30	38.95
avg	31.13	25.16	37.00	37.00	**38.65**	33.29	37.90	38.02

Table 3: F1 score in SDQC task of RumourEval-2019 (Gorrell et al. 2018)

Limitations

A necessary assumption in subspace alignment is that classes are clustered in the embedding space: most embedded tokens should lie closer to *other* embedded tokens of the *same* class than to embedded tokens of another class. If this is not the case, then aligning based on a few labelled samples of class k does not imply that the embedded source tokens are aligned to other target points of class k. This assumption is violated if, for instance, people only discuss one aspect of a rumour on day one and discuss several aspects of a rumour simultaneously on day two. One would observe a single cluster of token embeddings for supporters of the rumour initially and several clusters at a later time-step. Note that there is no unique solution for aligning a single cluster to multiple clusters.

Additionally, if those few samples labeled in the current time-step (for semi-supervised alignment) are falsely labeled or their label is ambiguous (e.g. a tweet that could equally be labeled as QUERY or DENY), then the source data could be aligned to the wrong point cloud. It is important that the few labeled tokens actually represent their classes. This is a common requirement in semi-supervised learning and is not specific to sequential alignment of text representations.

Related Work

The temporal nature of data can have a significant impact in natural language processing tasks. For instance, Kutuzov et al. compare a number of approaches to diachronic word embeddings, and detection of semantic shifts across time. For instance, such representations can be used to uncover changes of word meanings, or senses of new words altogether (Gulordava and Baroni 2011; Heyer, Holz, and Teresniak 2009; Michel et al. 2011; Mitra et al. 2014; Wijaya and Yeniterzi 2011). Other work has investigated changes in the usage of parts of speech across time (Mihalcea and Nastase 2012). Yao et al. investigate the changing meanings and associations of words across time, in the perspective of language change. By learning time-aware embeddings, they are able to outperform standard word representation learning algorithms, and can discover, e.g., equivalent technologies through time. Lukeš and Søgaard show that lexical features can change their polarity across time, which can have a significant impact in sentiment analysis. Wang and McCallum show that associating topics with continuous distributions of timestamps yields substantial improvements in terms of topic prediction and interpretation of trends. Temporal effects in NLP have also been studied in the context of scientific journals, for instance in the context of emerging themes and viewpoints (Blei and Lafferty 2006; Sipos et al. 2012), and in terms of topic modelling on news corpora across time (Allan, Gupta, and Khandelwal 2001). Finally, in the context of rumour stance classification, Lukasik et al. show that temporal information as a feature in addition to textual content offers an improvement in results. While this previous work has highlighted the extent to which language change across time is relevant for NLP, we present a concrete approach to taking advantage of this change. Nonetheless, these results could inspire more specialised forms of sequential adaptation for specific tasks.

Unsupervised subspace alignment has been used in computer vision to adapt between various types of representations of objects, such as high-definition photos, online retail images and illustrations (Fernando et al. 2013).Alignment is not restricted to linear transformations, but can be made nonlinear through kernelisation (Aljundi et al. 2015). An extension to semi-supervised alignment has been done for images (Yao et al. 2015), but not in the context of classification of text embeddings or domain adaptation on a sequential basis.

Conclusions

In this paper, we introduced sequential subspace alignment (SSA) for natural language processing (NLP), which allows for improved generalisation from past to present data. Experimental evidence shows that this method is useful across diverse NLP tasks, in various temporal settings ranging from weeks to years, and for word-level and document-level representations. The best-performing SSA method, aligning sub-spaces in a semi-supervised way, outperforms simply training on all data with no alignment.

Acknowledgements

WMK was supported by the Niels Stensen Fellowship.

References

Aljundi, R.; Emonet, R.; Muselet, D.; and Sebban, M. 2015. Landmarks-based kernelized subspace alignment for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 56–63.

Allan, J.; Gupta, R.; and Khandelwal, V. 2001. Temporal

summaries of news topics. In *International Conference on Research and Development in Information Retrieval*, 10–18.

Blei, D. M., and Lafferty, J. D. 2006. Dynamic topic models. In *International Conference on Machine Learning*, 113–120.

Derczynski, L.; Bontcheva, K.; and Roberts, I. 2016. Broad Twitter Corpus: A diverse named entity recognition resource. In *International Conference on Computational Linguistics*, 1169–1179.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*.

Fernando, B.; Habrard, A.; Sebban, M.; and Tuytelaars, T. 2013. Unsupervised visual domain adaptation using subspace alignment. In *IEEE International Conference on Computer Vision*, 2960–2967.

Gorrell, G.; Bontcheva, K.; Derczynski, L.; Kochkina, E.; Liakata, M.; and Zubiaga, A. 2018. RumourEval 2019: Determining rumour veracity and support for rumours. *arXiv:1809.06683*.

Gulordava, K., and Baroni, M. 2011. A distributional similarity approach to the detection of semantic change in the google books N-gram corpus. In *Workshop on Geometrical Models of Natural Language Semantics*, 67–71.

Heyer, G.; Holz, F.; and Teresniak, S. 2009. Change of topics over time-tracking topics by their change of meaning. *International Conference on Knowledge Discovery and Information Retrieval* 9:223–228.

Kang, D.; Ammar, W.; Dalvi, B.; van Zuylen, M.; Kohlmeier, S.; Hovy, E.; and Schwartz, R. 2018. A dataset of peer reviews (PeerRead): Collection, insights and NLP applications. In *North American Chapter of the Association for Computational Linguistics*.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv:1412.6980*.

Kochkina, E.; Liakata, M.; and Augenstein, I. 2017. Turing at SemEval-2017 task 8: sequential approach to rumour stance classification with Branch-LSTM . In *International Workshop on Semantic Evaluation*, 475–480.

Kouw, W. M., and Loog, M. 2019. A review of domain adaptation without target labels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Kutuzov, A.; Øvrelid, L.; Szymanski, T.; and Velldal, E. 2018. Diachronic word embeddings and semantic shifts: a survey. In *International Conference on Computational Linguistics*, 1384–1397.

Kwon, S.; Cha, M.; and Jung, K. 2017. Rumor detection over varying time windows. *PLoS ONE* 12(1):e0168344.

Lukasik, M.; Srijith, P.; Vu, D.; Bontcheva, K.; Zubiaga, A.; and Cohn, T. 2016. Hawkes processes for continuous time sequence classification: an application to rumour stance classification in Twitter. In *Annual Meeting of the Association for Computational Linguistics*, volume 2, 393–398.

Lukeš, J., and Søgaard, A. 2018. Sentiment analysis under temporal shift. In *Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 65–71.

Michel, J.-B.; Shen, Y. K.; Aiden, A. P.; Veres, A.; Gray, M. K.; Pickett, J. P.; Hoiberg, D.; Clancy, D.; Norvig, P.; Orwant, J.; et al. 2011. Quantitative analysis of culture using millions of digitized books. *Science* 331(6014):176–182.

Mihalcea, R., and Nastase, V. 2012. Word epoch disambiguation: Finding how words change over time. In *Annual Meeting of the Association for Computational Linguistics*, volume 2, 259–263.

Mitra, S.; Mitra, R.; Riedl, M.; Biemann, C.; Mukherjee, A.; and Goyal, P. 2014. That's sick dude!: Automatic identification of word sense change across different timescales. In *Annual Meeting of the Association for Computational Linguistics*, volume 1, 1020–1029.

Peters, M.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. In *North American Chapter of the Association for Computational Linguistics*, volume 1, 2227–2237.

Peters, M.; Ruder, S.; and Smith, N. A. 2019. To tune or not to tune? Adapting pretrained representations to diverse tasks. *CoRR* abs/1903.05987.

Radford, A.; Narasimhan, K.; Salimans, T.; and Sutskever, I. 2018. Improving language understanding by generative pre-training.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; ; and Sutskever, I. 2019. Language models are unsupervised multitask learners.

Sipos, R.; Swaminathan, A.; Shivaswamy, P.; and Joachims, T. 2012. Temporal corpus summarization using submodular word coverage. In *International Conference on Information and Knowledge Management*, 754–763.

Sutton, C., and Gong, L. 2017. Popularity of arXiv. org within Computer Science. *arXiv:1710.05225*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, 5998–6008.

Wang, X., and McCallum, A. 2006. Topics over time: a non-Markov continuous-time model of topical trends. In *International Conference on Knowledge Discovery and Data Mining*, 424–433. ACM.

Wijaya, D. T., and Yeniterzi, R. 2011. Understanding semantic change of words over centuries. In *International Workshop on Detecting and Exploiting Cultural Diversity on the Social Web*, 35–40.

Yao, T.; Pan, Y.; Ngo, C.-W.; Li, H.; and Mei, T. 2015. Semi-supervised domain adaptation with subspace learning for visual recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2142–2150.

Yao, Z.; Sun, Y.; Ding, W.; Rao, N.; and Xiong, H. 2018. Dynamic word embeddings for evolving semantic discovery. In *International Conference on Web Search and Data Mining*, 673–681.