Neural Temporality Adaptation for Document Classification: Diachronic Word Embeddings and Domain Adaptation Models

Xiaolei Huang and **Michael J. Paul**
Department of Information Science
University of Colorado, Boulder, CO 80309, USA
{xiaolei.huang, mpaul}@colorado.edu

Abstract

Language usage can change across periods of time, but document classifiers models are usually trained and tested on corpora spanning multiple years without considering temporal variations. This paper describes two complementary ways to adapt classifiers to shifts across time. First, we show that diachronic word embeddings, which were originally developed to study language change, can also improve document classification, and we show a simple method for constructing this type of embedding. Second, we propose a time-driven neural classification model inspired by methods for domain adaptation. Experiments on six corpora show how these methods can make classifiers more robust over time.

1 Introduction

Language changes and varies over time, which can cause a degradation of performance in natural language processing models over time. For example, document classifiers are typically trained on historical data and tested on future data, where the performance tends to be worse. Recent research has shown that document classifiers can become more stable over time when trained in ways that specifically account for temporal variations (Huang and Paul, 2018; He et al., 2018). We refer to this task of accounting for such variations during training as *temporality adaptation*.

This paper investigates temporality adaptation in two ways. First, we explore how *diachronic* word embeddings, which encode time-varying representations of words, can be used in this setting. Recent research has used diachronic word embeddings to study how language changes over time (Kulkarni et al., 2015; Hamilton et al., 2016; Kutuzov et al., 2018). These studies have shown that shifts in the corpora across time cause changes in word contexts and consequently, changes in the learned representations.

In our study, we further examine these shifts as they relate to important features for document classification. While other research has applied diachronic word embeddings to semantic change detection and validation (Mihalcea and Nastase, 2012; Kim et al., 2014; Kulkarni et al., 2015; Hamilton et al., 2016; Dubossarsky et al., 2017; Yao et al., 2018; Rudolph and Blei, 2018; Rosenfeld and Erk, 2018) and semantic relation analysis (Liao and Cheng, 2016; Szymanski, 2017; Rosin et al., 2017), these types of embeddings have not been studied particularly for the document classification task. We show that neural classifiers which use these embeddings can perform better on future data. As part of this work, we propose a new method for constructing diachronic words embeddings, which we show to be competitive with prior approaches.

Second, we propose a neural classification model that adapts to changes in time using ideas from *domain adaptation*. We previously showed that out-of-the-box domain adaptation techniques can make n-gram classifiers more robust to temporal shifts (Huang and Paul, 2018). We expand this line of work by additionally considering neural adaptation models, which can also take advantage of diachronic word embeddings.

The next section describes our data. We experiment with six English and Chinese datasets from both social media and newspaper sources, spanning varying lengths in time (from several decades to only a few years). We split each dataset into a small set of time intervals, and we define each time interval as a domain. Before presenting our methods for building diachronic word embeddings (Section 3) and our neural model (Section 4), we present empirical analyses of how word usage and word contexts vary over time in our data, to mo-

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4113–4123
Florence, Italy, July 28 - August 2, 2019.

tivate the methods (Section 2). Our main experiments are presented in Section 5, where we experiment with both neural and non-neural classifiers.

2 Time-Varying Corpora

The way people use words to express opinions has been constantly changing over time (Mihalcea and Nastase, 2012; Kulkarni et al., 2015; Hamilton et al., 2016). In this section, we introduce our corpora for this paper and conduct initial analyses on how word usage shifts over time in news articles and social media data in both English and Chinese with respect to the task of classification. We explore the language usage issue from two perspectives: word usage and context shift.

2.1 Data

We retrieved available data sources from previous publications (Zhang et al., 2014; He and McAuley, 2016; Huang and Paul, 2018). Specifically, we use four different sources in in English—Amazon (music reviews), Yelp (restaurant and hotel reviews), Twitter, and economic newspaper articles (Figure Eight Inc., 2015)— and one source in Chinese, Dianping (Meituan-Dianping, 2019).

The *Twitter* data is annotated with binary labels indicating whether the user received a flu vaccine. The *Economy* data is annotated with binary labels indicating if each article relates to the US economy. For the review data (*Amazon, Dianping* and *Yelp*), we encode review scores into three discrete categories: score >3 as positive, $=3$ as neutral, and <3 as negative.

Following Huang and Paul (2018), we group the corpora into several bins of temporal intervals; specifically, non-repeating time intervals spanning one or more years (Table 1). We encode each temporal domain into the discrete time labels, $1, 2, ...T$. One corpus then can be represented as $C = [C_1, C_2, ...C_T]$, where each C_t for $t \in T$ is one temporal slice of the document collection.

2.2 Analysis 1: Word Usage Shift

Document classification models often use feature representations that are derived from words. Therefore, variations in word usage across time will change the distribution of features over time, which can impact the stability of document classifiers (Huang and Paul, 2018). Our goal in this section is to test whether there are temporal variations in our datasets, how strong the effects are,

Dataset	Time intervals
Amazon	1997-99, 2000-02, 2003-05 2006-08, 2009-11, 2012-14
Dianping	2009, 2010, 2011, 2012
Economy	1950-70, 1971-85, 1986-2000, 2001-14
Twitter	2013, 2014, 2015, 2016
Yelp-hotel	2005-08, 2009-11, 2012-14, 2015-17
Yelp-rest	2005-08, 2009-11, 2012-14, 2015-17

Table 1: Corpora spanning multiple time intervals.

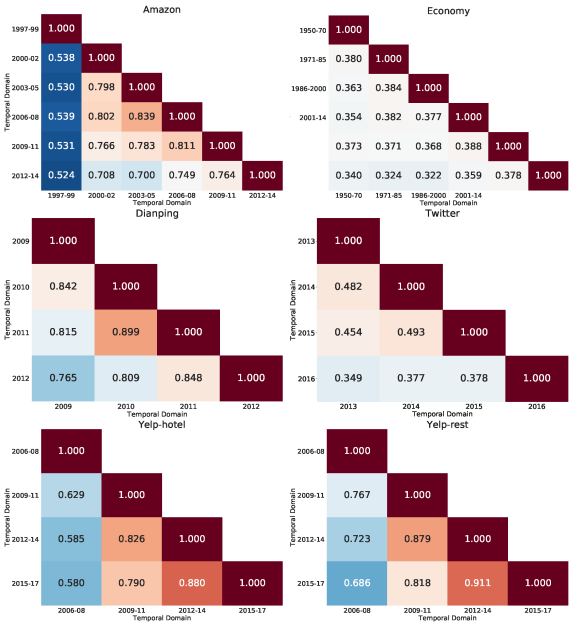

Figure 1: Word usage overlaps between every two time domains. A value of 1 means no variations of top features between two temporal domains, while values less than 1 indicate more temporal variations.

and what patterns exist of word usage shifts. This will help us understand how word usage variations can affect document classifiers.

We consider the word usage as it relates to document classification by measuring the overlap of top word features across time intervals. We rank and select the top 1,000 features for each interval by mutual information. We then calculate the intersection percentage between every two domains; specifically, if S_0 is the set of top features for one temporal domain and S_1 is the set of top features for another attribute, the percent overlap is calculated as $|S_0 \cap S_1|/1000$.

We present the overlaps of word usages across time in Figure 1. The overlap of word usage

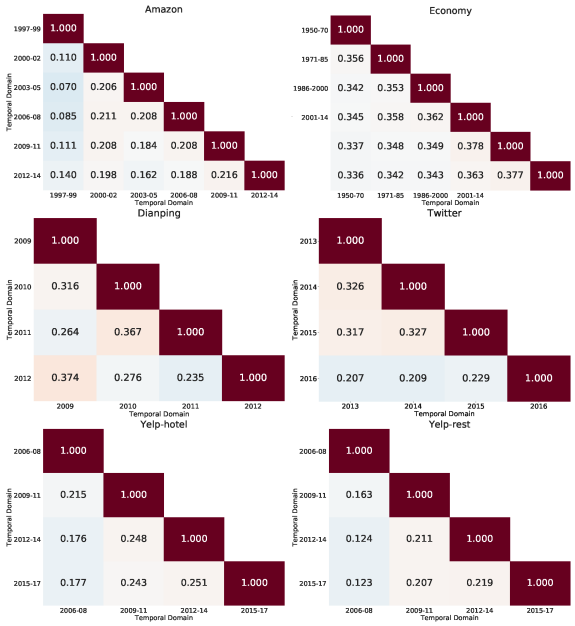

Figure 2: Context overlaps between every two temporal domains. A value of 0 indicates the contexts of top features between two temporal domains have nothing in common, while values away from 0 mean the contexts share more similarities.

between temporal domains varies greatly across different corpora (ranging from 0.322 to 0.911). We observe that closer temporal domains usually have higher overlap while further temporal domains share less overlap. These results thus suggest that the word usage varies over time across many settings.

2.3 Analysis 2: Context Shift

Popular word representations for classification train word embeddings using the *context* of each word (a window around the word, e.g., in skip-gram or continuous bag of words methods) (Mikolov et al., 2013; Bojanowski et al., 2016). Therefore, we seek to understand how semantic contexts of words shift across time, in addition to the words themselves. If we observe a significant context shift, this could lead to inconsistent semantic representations across time.

In the case of context shift, we extract the same unigram features as in the previous section and define word contexts by simulating the word embedding training process via contextual windows. We set a window size of five words and record the words that occur within the context windows. Following the previous section but using the set of words that appear in the context windows, we then calculate the intersection overlap between each pair of time domains.

We show the overlap in the Figure 2. The overlap percentages range from 0.070 to 0.378. We also observe that temporally closer domains share higher percentages of contextual words. The pattern aligns with our observations in the Section 2.2. Since word embeddings rely heavily on contextual information (Mikolov et al., 2013), our observations that contexts have little overlap across different time intervals therefore suggest it will be important to account for temporality in word embeddings.

3 Diachronic Word Embeddings

Standard word embeddings ignore temporal language variations in the data. *Diachronic* word embeddings (Kulkarni et al., 2015) encode temporality into word embeddings to obtain dynamic representations of words. These types of embeddings have been effective in capturing and learning the language usage and semantic shift over time (Kim et al., 2014; Kulkarni et al., 2015; Hamilton et al., 2016; Bamler and Mandt, 2017; Szymanski, 2017; Rudolph and Blei, 2018; Rosenfeld and Erk, 2018; Yao et al., 2018).

To the best of our knowledge, diachronic word embeddings have not been studied in the context of document classification. Since our preliminary analyses in the previous section showed that the top features for document classification vary over time, and the contexts used to train those word embeddings also vary over time, it would make sense to use word representations that can vary over time. In this section, we present a new, simple-to-implement method for constructing diachronic embeddings, and then further analyze temporal shifts in corpora using these embeddings.

3.1 Concatenative Training Approach

Methods to obtain diachronic word embeddings fall into three main directions: incremental training (Kim et al., 2014), alignment transformation (Kulkarni et al., 2015; Hamilton et al., 2016; Yao et al., 2018) and continuous time representations (Rosenfeld and Erk, 2018; Rudolph and Blei, 2018).

In this work, we propose an alternative approach to encoding time into word embeddings. The idea is inspired by the "easy" domain adaptation method (Daume III, 2007), which was shown to be successful at modeling different temporal do-

mains (Huang and Paul, 2018), and can be implemented by simply modifying the input data without modifying the training process. In our approach, words in the training data are concatenated with the name of the time interval, and embeddings are trained using a sub-word sharing framework (Bojanowski et al., 2016). The concatenation step allows for the learning of word representations that are specific to each time interval, while the sub-word framework allows for the learning of general, time-independent representations of each word.

Concretely, we first build domain-specific corpora by adding each document's domain label as a suffix to each word, as shown in Figure 3. In addition to the domain-specific corpora, we retain the original corpus as a domain-independent version. We then train fastText (Bojanowski et al., 2016), a sub-word embedding model, on all of the corpora, We use 3- to 6-grams characters in this study, to provide diverse perspectives to encode time and word representations. This approach learns diachronic word representations by encoding temporality as part of sub-words into the word embedding.

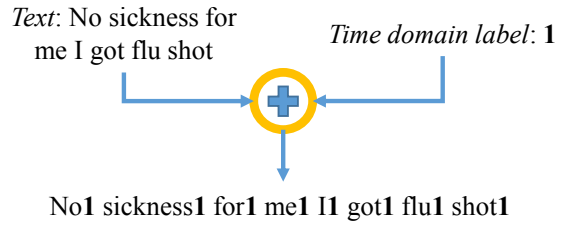

Figure 3: The illustration of building domain corpora. We append the document domain label as a suffix to each word in the document.

FastText learns word embeddings from character n-grams, intended to capture morphological information. As an example example, the word "where1" from time domain 1 using character 3-grams would be encoded in fastText as the following seven parts:

$$< wh, whe, her, ere, re1, e1 >, < where1 >$$

In this way, words with time domain labels can incorporate temporal identities, while the same words with different domain labels will still share close representations because of similar morphological forms. In this way, we encode temporal identity into word representations while still maintaining the connections of the same words across different time domains.

In contrast to prior approaches on diachronic embeddings, this concatenative sub-word approach does not explicitly model the ordering of time information, and it cannot encode, for example, that domains that are close in time should have more similarities than domains that are farther away in time. Despite this limitation, we find experimentally that this approach works competitively, while being simpler to implement and faster to train.

3.2 Analysis 3: Semantic Distribution Shift

Using this approach to constructing diachronic word embeddings, we now consider how these embeddings can be used to further analyze language shift.

The Law of Conformity states a negative correlation between word frequency and meaning change (Hamilton et al., 2016); however, Dubossarsky et al. (2017) show that the word frequency does play an important role in the semantic change, even though a small one. Diachronic embeddings have been used to measure the semantic shift using linear interpolation (regression) (Hamilton et al., 2016). Here, we reexamine this issue from another view, the distance of semantic distributions, which views the word embeddings as semantic distributions and measures how the word embeddings vary across time.

As in Section 2.3, we choose the top 1,000 important words ranked by mutual information, as well as a control group of the 1,000 most frequent words in each corpus. We find that overlap between the 1,000 most important and most frequent words are 0% across every dataset. This suggests that the most frequent words are not predictive for classification. We use our proposed method to train 200-dimensional diachronic word embeddings and extract diachronic word representations for both important and most frequent words, and leave 0s to the words that do not appear within a temporal interval. Finally, we use the Wasserstein distance (Shen et al., 2018) to measure the differences across temporal domains. Wasserstein distance or Earth Mover's distance (Vallender, 1974) measures the distribution differences between source and target domains (Shen et al., 2018), and thus here it measures semantic distribution shifts across time.

We show temporal distribution shifts in Figure 4, and we observe two interesting findings.

First, closer time intervals show less semantic distribution shift, which aligns with our analysis in Sections 2.2 and 2.3. Second, we observe that the frequent words have much smaller semantic distribution shifts than the top features selected by mutual information.

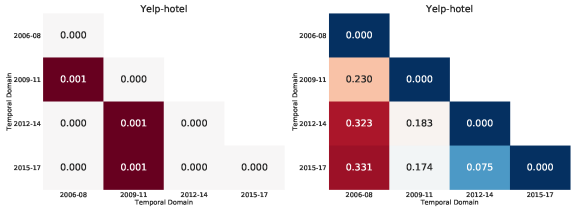

Figure 4: Semantic distribution shifts comparison between the top and most frequent words via Wasserstein distance. We present Yelp-hotel data for the illustration purposes and omit the other data due to space limits. The left is the semantic distribution shift for frequent words, the right is for top words ranked by mutual information. The higher score indicates higher shift.

To verify this second observation statistically, we conduct two-tailed t-test on the Yelp-hotel case to test our null hypothesis that the semantic distribution change is not significant. We separately compare the distribution distances of both frequent and top feature words with 0, which indicates no shift. Finally, the test results show p-value=0.076 for the frequent words and p-value=0.0038 for top feature words. Therefore, we reject the null hypothesis of top feature words at 95% confidence level while we cannot reject the null hypothesis of frequent words.

3.2.1 Comparing Different Ways to Measure Temporal Shifts

We have presented language shifts across time domains based on word usage overlap (Section 2.2), context overlap (Section 2.3), and distribution distance (this section). However, it is not clear if these different metrics are measuring the same information. To understand this further, we calculate the Pearson correlation coefficient to measure the relationships between each pair of metrics. We show the correlations in Table 2. We observe negative correlations between the two overlap measures (higher means more less shift) and distribution distance (lower means less shift), and a positive correlation between word usage and context overlaps. These results show that the three metrics are related, though there are some datasets where the correlations are low.

4 Model for Temporality Adaptation

We construct a document classification model that assumes the language used to describe document categories will evolve over time; for example newer documents may use emoji to express opinions, while older documents would not contain these features. Our goal is to build document classifiers with time-invariant features and are thus robust to language shift.

Our previous work (Huang and Paul, 2018) on temporality adaptation for n-gram classifiers used a domain adaptation approach (Daume III, 2007) where each time interval is treated as a domain. This approach created $T+1$ versions of the feature set, one for each of the T time domains, and one domain-independent feature set. This allows the model to learn which features are associated with specific domains, while the domain-independent parameters can be used for future data. We analogously apply this idea to the neural setting, where we construct $T+1$ different representations, at both the word level (using diachronic word embeddings) and the document level. Moreover, we use a time-driven learning process that models the shift of word representations as a gradual process of adapting representations to new data while starting with old information.

We thus present the **Neural Temporality Adaptation Model (NTAM)** (Figure 5) based on three strategies: diachronic word embeddings (Section 3), $T+1$ views of inputs, and a time-driven learning process. This model can learn language shifts and time invariant representations of documents for classification.

T+1 views of inputs. Analogous to the approach of Daume III (2007) for non-neural classifiers, we create $T+1$ word representations, where T refers to the number of diachronic domain embeddings and 1 refers to a general embedding, which trains word embeddings on the whole corpus without time labels. Our intuition is to use time-specific embeddings to provide documents from different time intervals with different views of semantic meaning. We train diachronic word embeddings using our proposed method via fastText, though we also experiment with other approaches. We initialize the model with the domain-specific embeddings and the general word embedding. The model will encode input documents into $T+1$ views of word representations. The $T+1$ embeddings pro-

Correlation	Amazon	Dianping	Economy	Twitter	Yelp-hotel	Yelp-rest
Usage-DD	-.901*	.160	-.106	.028	-.943*	-.923*
Context-DD	-.989*	-.987*	-.108	.023	-.949*	-.960*
Usage-Context	.926*	-.009	.600*	.979*	.950*	.955*

Table 2: The correlations between word usages overlaps (Usage) and distribution distance (DD) as well as context overlaps (Context) and distribution distance (DD). The star sign (*) indicates p-value is less than .05.

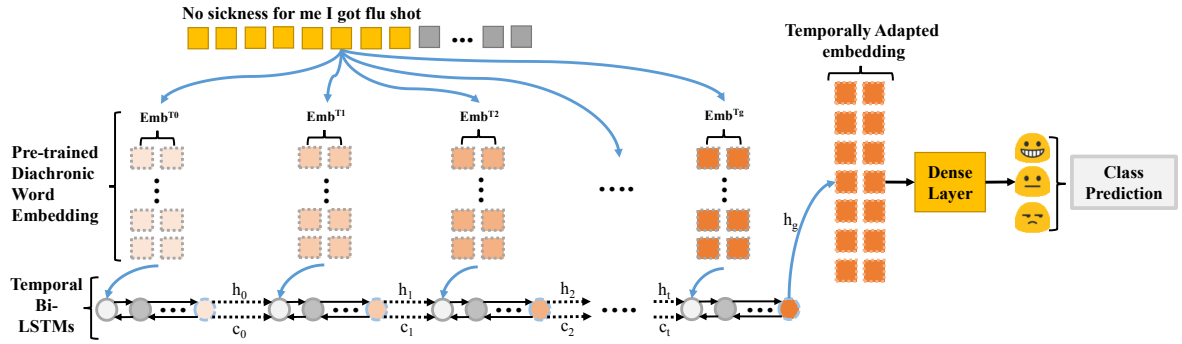

Figure 5: Architecture of the Neural Temporality Adaptation Model (NTAM). NTAM is initialized with T diachronic word embeddings ($Emb^{Tt}, t \in T$) plus one general word embedding (Emb^{Tg}). The hidden state ($h_t, t \in T$) and memory cell (c_t) will excite and initialize the following Bi-LSTM. We feed the final hidden state (h_g, g refers to general domain) to the following learning phase.

vide diverse views of input words, which are fed to the rest of the neural architecture, leaving the model to optimize representations automatically.

Time-driven learning process. To learn temporal variations for document representations, we propose a series of $T + 1$ continuously temporal Bidirectional Long Short Term Memory (Bi-LSTM) models (Hochreiter and Schmidhuber, 1997). The first T Bi-LSTMs correspond to the T time domains and the last Bi-LSTM corresponds to the general view of input documents and outputs the final document representation. Similar to how the diachronic word embeddings encode input words into multiple views of time domains, we use $T + 1$ Bi-LSTMs to learn diachronic views of document representations.

To capture the semantic shifts across time domains, our intuition is to model the dynamic process. The memory mechanism of LSTM fits our need, which optimizes the balance across different time patterns via non-linear computations. While each Bi-LSTM reads through tokens in its own input document, we feed the previous Bi-LSTM's hidden state and memory cell to excite the learning process of the subsequent Bi-LSTM. The final Bi-LSTM learns jointly the previous shift patterns of document representations with the general embedding view of documents and outputs its final document representation h_g.

The final document representation is fed into a dense layer with a non-linear activation function. We use outputs of the dense layer for document class prediction, where we use one-hot encoding to represent document labels and use the softmax function for class predictions. Finally, we use categorical cross-entropy as the loss function.

5 Experiments

We conduct experiments on the task of document classification. We split the data chronologically to simulate the realistic scenario where a classifier is trained on older data and tested on newer data. Thus, the first $T - 1$ time domains are used for training; the last time domain is split into two equal-sized sets for development and testing.

5.1 Preprocessing

We use NLTK (Loper and Bird, 2002) to tokenize the English corpora and the Jieba Python module (Sun, 2012) to segment the Chinese data. We discard reviews that had fewer than 10 tokens. For the Twitter data, we anonymize the data and replace usernames, hyperlinks, and hashtags with "USER", "URL", "HASHTAG" respectively. All other text is lowercased. The final data details are described in Table 3.

Datasets	Train	Dev.	Test
Amazon	59,399	11,880	11,880
Dianping	503,330	83,889	83,889
Economy	4,774	596	596
Twitter	1,632	272	272
Yelp-hotel	20,975	6,993	6,993
Yelp-rest	106,943	35,648	35,648

Table 3: Data statistics of the six corpora. We show the number of documents in each split.

5.2 Implementation and Training

We implement classification models using Keras (Chollet et al., 2015) and scikit-learn (Pedregosa et al., 2011). We select the top 15K words by frequency and set the other words as "unk". The models are trained for 15 epochs with the batch size of 64. Each document is padded to 60 tokens. We set the Bi-LSTM output to 200 dimensions. We choose ReLU (Hahnloser et al., 2000) as the activation function of the dense layer and 0.2 as our default dropout rate (Srivastava et al., 2014). The dense layer outputs 200 dimensions for final document class prediction. We select cross-entropy as our default loss function, and we optimize model parameters via RMSprop (Tieleman and Hinton, 2012) with the learning rate as 0.0001. Unless otherwise stated, we leave the other parameters as defaults.

5.3 Baselines

To ensure fair comparisons, we use the same settings across all models. We compare our proposed model to seven baselines, where three standard classifiers do not perform temporality adaptation.

5.3.1 No Adaptation

LR. We extract 1- and 2-gram features on the corpora with the most frequent 15K features. We then build a logistic regression classifier using `LogisticRegression` from scikit-learn (Pedregosa et al., 2011) with default parameters.

CNN. We implement the Convolutional Neural Network (CNN) classifier described in (Kim, 2014). To keep consistent, we initialize the model with pre-trained word embeddings (Bojanowski et al., 2016) that were trained on the same datasets as the diachronic embeddings. We only keep the 15K most frequent words and replace the rest with an "unk" token. We set model optimizer as Adam (Kingma and Ba, 2014). We keep all other parameter settings as described in the paper.

Bi-LSTM. We build a bi-directional Long Short Term Memory (bi-LSTM) (Hochreiter and Schmidhuber, 1997) classifier to examine the effectiveness of temporal learning process in our proposed model. The classifier is initialized with the pre-trained word embeddings.

5.3.2 Domain Adaptation Models

FEDA. Following Huang and Paul (2018) we adapt for time domains using the "frustratingly easy" domain adaptation (FEDA) method (Daume III, 2007). The feature set is augmented such that each feature has a domain-specific version of the feature for each time domain, as well as a general domain-independent version of the feature. The values of features are set to the original feature value for the domain-independent feature and the domain-specific features that apply to the document, while domain-specific features for documents that do not belong to that domain are set to 0. At test time, we only use the general, domain-independent features. We use the same feature extraction procedures and the same logistic regression classifier as the *LR* baseline.

DANN. We consider the domain adversarial training network (Ganin et al., 2016) (DANN) on the time adaptation task. We re-implement the same network and set domain prediction as predicting the time domain label while keeping the document label prediction as the default. We use the model from the epoch when the model achieves the best result on the development set for the final model.

RCNN & HAN. He et al. (2018) propose an evolving framework to train document classifiers. We re-implement two classifiers, RCNN and HAN with diachronic propagation learning strategy, which achieved the best performances in their paper. The RCNN (Lai et al., 2015) classifier integrates both LSTM and CNN, and the HAN (Yang et al., 2016) classifiers uses hierarchical attention neural architectures. We keep the two models with the same parameters as their open sourced code and initialize the two models with pre-trained 200 dimensional word embeddings (Bojanowski et al., 2016). We apply Adam and RMsprop for RCNN and HAN respectively, because the two optimiz-

ers perform much better on validation sets than the stochastic gradient descent optimizer used in the original paper. The work is close to our work but there are three major differences:

- Time invariance. We train one unified model with diachronic adaptation by using a time-independent representation (the 1 of the $T+1$ representations) to learn a time-invariant classifier that can be used for future data. In contrast, these baselines learn $T-1$ models, where they train one model for each time domain.
- Diachronic word embeddings. Our method uses diachronic word embeddings to encode inputs in $T+1$ different views. The baseline encoding is based on only the current embedding space and therefore might not capture embedding shifts over time.
- Learning process. The baseline learns a weighted sum between the intermediate layer's outputs between the previous model and the current model. In contrast, we deploy the $T+1$ Bi-LSTMs to jointly learn time dependencies across all time intervals.

5.4 Results

The results of our experiments are show in Table 4. Our proposed approach leads to performance improvements over the comparable baselines on most datasets. NTAM has the highest performance on 4 out of 6 datasets, while FEDA has the highest performance on the other 2 (while NTAM is the next best for those 2).

The baselines with domain adaptation generally obtain a small performance boost over the baselines without adaptation on temporality. Among the non-neural models, the adaptation baseline FEDA outperforms the non-adaptation baseline LR on 4 out of 6 datasets. Among the neural models, the best adaptation baseline outperforms the best non-adaptation baseline on 3 out of 6 datasets, with the RCNN generally outperforming the other baselines. This indicates that the temporal factor can potentially improve the performance of document classification, and that domain adaptation is a possible approach to temporality adaptation.

Significance analysis. To verify the improvements of our proposed method NTAM compared to baselines, we conduct a significance analysis to compare our proposed model with the RCNN, which is the closest model to ours. We follow Berg-Kirkpatrick et al. (2012) and bootstrap sample 50 pairs of test datasets with replacement. We keep the same data size as the previous experiments in the Table 4. We then use the same previous parameters and re-conduct the classification experiments. We format the experimental results as two lists of scores. We conduct a paired t-test to test the null hypothesis that our proposed model does not differ significantly from the RCNN. The test presents a significant result with t(95) = 3.258 and p = 0.00119. The result suggests rejecting the null hypothesis at a 95% confidence level.

5.5 Effectiveness of Diachronic Embeddings

Lastly we investigate how diachronic word embeddings affect classifiers. While NTAM used diachronic word embeddings and other baselines did not, we also compare to a version of NTAM initialized with regular word embeddings (to understand whether diachronic embeddings are important to the model's performance), and we also experiment with combining diachronic embeddings with a baseline model (to understand if diachronic embeddings can be used in other classifiers).

We also compare different methods of constructing diachronic word embeddings. In addition to our proposed method in Section 3, which uses subword embeddings via fastText, we consider three other approaches. We use incremental training (Kim et al., 2014) (abbreviated *Incre*, using fastText), linear regression (Kulkarni et al., 2015), implemented in scikit-learn, and Procrustes (Hamilton et al., 2016), implemented in SciPy. We keep the same fastText parameters as in previous experiments and train a word embedding model separately for each time domain, then align the pre-trained embeddings to get final diachronic word embeddings. We then re-run the classification task with the new diachronic word embeddings.

Table 5 shows the absolute percentage improvement in classification performance when using each diachronic embedding compared to a classifier without diachronic embeddings. Overall, diachronic embeddings improve classification models. The diachronic embedding appears to be particularly important for NTAM, improving performance on all 6 datasets with an average increase in performance up to 2.53 points. The RCNN also benefits from diachronic embeddings, but to

Data	Baselines (No adaptation) LR	CNN	Bi-LSTM	Baselines (Adaptation) FEDA	DANN	HAN	RCNN	Our Model NTAM
Twitter	.874	.873	.879	.890	.851	.847	.869	**.898**
Economy	.699	.707	.692	.686	.687	.690	.697	**.711**
Yelp-rest	.818	.756	.787	**.831**	.736	.794	.782	.828
Yelp-hotel	.773	.753	.758	**.811**	.733	.740	.762	.790
Amazon	.778	.762	.771	.782	.686	.748	.782	**.808**
Dianping	.710	.715	.706	.687	.686	.699	.692	**.738**

Table 4: Performance of different models evaluated with weighted F1 scores. For each dataset, the best score is bolded. LR and FEDA are non-neural n-gram models, while the others are neural models.

Data	RCNN Incre	Linear	Procrustes	Subword	NTAM Incre	Linear	Procrustes	Subword
Twitter	-0.7	+1.4	-0.2	-0.8	+1.4	-0.3	+1.7	+3.5
Economy	+0.5	0.0	-0.7	+0.4	-0.3	-1.0	-0.5	+0.3
Yelp-rest	+1.4	+0.1	-1.9	+2.3	+1.9	+1.6	+1.4	+4.3
Yelp-hotel	-1.5	-1.2	-0.5	-0.2	-0.7	-2.0	-1.8	+0.8
Amazon	+0.2	+0.2	-2.0	+0.5	-0.8	-0.7	-0.8	+2.1
Dianping	+0.4	+1.6	+0.7	+1.0	+0.8	+1.8	+3.4	+4.2
Average	0.05	0.35	-0.47	0.53	0.38	-0.10	0.57	2.53
Median	0.30	0.15	-0.60	0.45	0.25	-0.50	0.45	2.80

Table 5: Performance gains of two neural temporality adaptation models when they are initialized by diachronic word embeddings as compared to initialization with standard non-diachronic word embeddings. *Subword* refers to our proposed diachronic word embedding in this paper (Section 3). We report absolute percentage increases in weighted F1 score after applying diachronic word embeddings.

a lesser extent, with an improvement on 4 of the 6 datasets. Comparing the different methods for constructing diachronic embeddings, we find that our proposed subword method works the best on average for both classifiers. The incremental training method also provides improved performance for both classifiers, while the linear regression and Procrustes approaches have mixed results.

6 Conclusion

Our experiments on six corpora covering two languages show that there are shifts in word usage and context over time, and that it is useful to explicitly account for these shifts in representations of words and documents. We have presented a new method for constructing diachronic word embeddings as well as a new model for document classification, which are both shown to be effective for temporality adaptation. We open source our code.[1]

[1] https://github.com/xiaoleihuang/Neural_Temporality_Adaptation

Acknowledgments

This work was supported by the National Science Foundation by award number IIS-1657338. We thank Vivian Lai for her useful feedback.

References

Robert Bamler and Stephan Mandt. 2017. Dynamic word embeddings. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 380–389. JMLR. org.

Taylor Berg-Kirkpatrick, David Burkett, and Dan Klein. 2012. An empirical investigation of statistical significance in nlp. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 995–1005. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

François Chollet et al. 2015. Keras. https://keras.io.

Hal Daume III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263.

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.

Figure Eight Inc. 2015. Data for everyone.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030.

Richard HR Hahnloser, Rahul Sarpeshkar, Misha A Mahowald, Rodney J Douglas, and H Sebastian Seung. 2000. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405(6789):947.

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)*, volume 1, pages 1489–1501.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web (WWW)*, pages 507–517. International World Wide Web Conferences Steering Committee.

Yu He, Jianxin Li, Yangqiu Song, Mutian He, and Hao Peng. 2018. Time-evolving text classification with deep neural networks. In *IJCAI*, pages 2241–2247.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Xiaolei Huang and Michael J Paul. 2018. Examining temporality in document classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 694–699.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751.

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.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Vivek Kulkarni, Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2015. Statistically significant detection of linguistic change. In *Proceedings of the 24th International Conference on World Wide Web*, pages 625–635. International World Wide Web Conferences Steering Committee.

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.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *Twenty-ninth AAAI conference on artificial intelligence*.

Xuanyi Liao and Guang Cheng. 2016. Analysing the semantic change based on word embedding. In *Natural Language Understanding and Intelligent Applications*, pages 213–223. Springer.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ETMTNLP '02, pages 63–70, Stroudsburg, PA, USA. Association for Computational Linguistics.

Meituan-Dianping. 2019. Meituan-dianping official website.

Rada Mihalcea and Vivi Nastase. 2012. Word epoch disambiguation: Finding how words change over time. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 259–263.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

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.

Alex Rosenfeld and Katrin Erk. 2018. Deep neural models of semantic shift. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 474–484.

Guy D Rosin, Eytan Adar, and Kira Radinsky. 2017. Learning word relatedness over time. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1168–1178.

Maja Rudolph and David Blei. 2018. Dynamic embeddings for language evolution. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 1003–1011. International World Wide Web Conferences Steering Committee.

Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. 2018. Wasserstein distance guided representation learning for domain adaptation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

J Sun. 2012. jiebachinese word segmentation tool.

Terrence Szymanski. 2017. Temporal word analogies: Identifying lexical replacement with diachronic word embeddings. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 448–453.

Tijmen Tieleman and Geoffrey Hinton. 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31.

SS Vallender. 1974. Calculation of the wasserstein distance between probability distributions on the line. *Theory of Probability & Its Applications*, 18(4):784–786.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.

Zijun Yao, Yifan Sun, Weicong Ding, Nikhil Rao, and Hui Xiong. 2018. Dynamic word embeddings for evolving semantic discovery. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 673–681. ACM.

Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 83–92. ACM.